IACR Transactions on Symmetric Cryptology
ISSN 2519-173X, Vol. 2022, No. 2, pp. 322–350. DOI:10.46586/tosc.v2022.i2.322-350

Vectorial Decoding Algorithm for Fast Correlation Attack and Its Applications to Stream Cipher Grain-128a

Zhaocun Zhou[1,3], Dengguo Feng[2] and Bin Zhang[1]

[1] Trusted Computing and Information Assurance Laboratory, Institute of Software, Chinese Academy of Sciences, Beijing, China
zhouzhaocun@126.com,martin_zhangbin@hotmail.com

[2] State Key Laboratory of Computer Science, Institute of Software, Chinese Academy of Sciences, Beijing, China
fengdg@263.net

[3] University of Chinese Academy of Sciences, Beijing, China

Abstract. Fast correlation attack, pioneered by Meier and Staffelbach, is an important cryptanalysis tool for LFSR-based stream cipher, which exploits the correlation between the LFSR state and key stream and targets at recovering the initial state of LFSR via a decoding algorithm. In this paper, we develop a vectorial decoding algorithm for fast correlation attack, which is a natural generalization of the original binary approach. Our approach benefits from the contributions of all correlations in a subspace. We propose two novel criteria to improve the iterative decoding algorithm. We also give some cryptographic properties of the new FCA which allows us to estimate the efficiency and complexity bounds. Furthermore, we apply this technique to the well-analyzed stream cipher Grain-128a. Based on a hypothesis, an interesting result for its security bound is deduced from the perspective of iterative decoding. Our analysis reveals the potential vulnerability for LFSRs over matrix ring and also for nonlinear functions with biased multidimensional linear approximations such as Grain-128a.

Keywords: Linear approximation · Fast correlation attack · Iterative decoding · Grain-128a

1 Introduction

Stream ciphers are a widely used class of symmetric-key cryptosystems. A key stream sequence is generated from the initial state derived from the key. The plaintext is encrypted by XORing with the key stream of the same length.

Linear feedback shift register (LFSR) based stream ciphers form an important class of stream cipher system, in which one or more LFSRs are often used. LFSRs could be defined over different algebraic structures, such as finite fields and matrix rings. Besides LFSR, these ciphers usually adopt a nonlinear filter function or a finite state automaton (FSM) with a nonlinear function. The history of these ciphers can be traced back to decades ago, e.g., LILI-128 [CDF+02], the SNOW family [EJ00, EJ03, UEA06, EJMY19] and the Grain family etc.

The Grain family includes three well-known stream ciphers: Grain-128a [ÅHJM11], Grain-128 [HJMM06] and Grain-v1 [HJM07]. Grain-v1 is in the eSTREAM portfolio and Grain-128a is standardized by ISO/IEC [29115]. All the members of the Grain family share a similar structure. Several lightweight ciphers proposed recently also adopt similar

Received: 2021-11-23 Revised: 2022-03-01 Accepted: 2022-05-01 Published: 2022-06-10

structures [AM15, AHMN13, MAM16]. However, the Grain family is reported to be vulnerable to fast correlation attacks (FCA) in CRYPTO 18 [TIM$^+$18]. After that, the same FCA approach is applied to Grain-like small state stream ciphers such as Plantlet, Fruit-v2 and Fruit-80 [WLLM19].

FCA is pioneered by Meier and Staffelbach in 1989 [MS89]. Generally speaking, FCA exploits the correlation between the key stream and the state or the outputs of LFSR. The problem of recovering the initial state of LFSR is transformed into a decoding problem. The linear part of the stream cipher is treated as a linear code, and the nonlinear part of the stream cipher is treated as noise. According to the differences in decoding strategies, these FCA approaches can be roughly divided into two classes.

The first class adopts a one-pass decoding algorithm. For example, the FCA adopts convolution codes and Viterbi decoding algorithm [JJ99b], which is improved by turbo codes [JJ99a]. Another FCA adopts maximum likelihood decoding on a reduced set of information bits [CJS00]. The parity-checks are usually folded to eliminate partial bits. List decoding and polynomial reconstruction can also be applied in FCA [MFI02, JJ00]. An important improvement is accelerating the parity-check evaluations by fast Walsh-Hadamard transform (FWHT) [CJM02]. This technique is applied in cryptanalysis of the stream cipher E0 [LV04]. It was later generalized to extension fields and applied to stream cipher SNOW 2.0 [ZXM15]. A recent improvement of FCA is based on commutative property and applied to Grain family [TIM$^+$18].

The second class adopts a probabilistic iterative decoding algorithm. After Meier and Staffelbach's original FCA, low-density parity-check code (LDPC) is introduced into FCA to improve the iterative decoding algorithm [CT00]. There are many related works in this area, such as [ÅLHJ12, CT00, Gol01, CGD96, GH05, MG91, MG93]. Intuitively, iterative decoding algorithm seems to be more powerful, as their decoding abilities are closer to Shannon's bound [Sha48]. However, compared with the FCA decoding by information set, it has some inconveniences. Firstly, it is usually very hard to describe its properties by mathematical language, e.g., the relationship among the number of parity-checks, the decoding ability and the noise distribution. Thereby, it is hard to derive a clear time/space/memory complexity as expected, sometimes even for toy ciphers. Secondly, although multidimensional linear approximations may have advantages in cryptanalysis stream ciphers [ZXM15], it still lacks a convenient iterative decoding algorithm to work with the multidimensional linear approximation. For these reasons, the application of the FCA based on an iterative algorithm to modern stream ciphers is very limited. In addition, in terms of iterative algorithms, even though some principles are discussed to avoid the iterative process being trapped into a "tie" state too early [MG91, CGD96], how to get rid of the tie state is also need to be considered.

Our Contributions

Firstly, we propose a vectorial iterative decoding algorithm for fast correlation attack, which generalizes Meier and Staffelbach's original FCA very naturally. The vectorial approach benefits from a multidimensional linear approximation, while the binary version only exploits a binary linear approximation. Moreover, the quantity of parity-checks fitting for the vectorial algorithm is much more than that for binary algorithms. We propose two novel criteria to improve the iterative decoding process. One is proposed for breaking the tie state, which may be also applied for some binary iterative algorithms. We also perform a scaled experiment to verify the validity of the vectorial algorithm and perform another experiment to verify the generality of the idea for breaking a tie state.

Secondly, we give some cryptographic properties for the first iteration via distribution approximations, which allows us to describe the relationship between the decoding efficiency and the noise distribution. We also give two propositions that involve the relationship between the number of parity-checks, the noise distribution and the data complexity. The first one illustrates the number of parity checks needed when the noise distribution is fixed. In addition, by the first proposition, we show that the vectorial algorithm may have theoretical advantages in data complexity for some cases. Given the number of parity-checks and the noise distribution, the second one shows the length of data needed to correct errors. These results reveal some theoretical constraints of the complexity, which provide us a more clear profile of the complexities of the FCA based on an iterative algorithm.

Finally, we apply those results to the well-analyzed stream cipher Grain-128a. In the first step, we construct a multidimensional linear approximation by bundling up the linear approximations proposed in [TIM+18]. In the second step, based on a hypothesis there are parity-checks with two taps or with a special form, we give a data complexity estimation by the proposed theoretical bounds for the vectorial algorithm. The result shows maybe its potential security margin is lower than we thought from the perspective of vectorial iterative decoding. Our analysis reveals the potential vulnerability for LFSRs over matrix ring and also for nonlinear functions with biased multidimensional linear approximations.

Outline

The rest of the paper is organized as follows. Section 2 is preliminary. Section 3 describes the details of the vectorial decoding algorithm and the scaled experiments. In section 4, we propose some cryptographic properties. How to apply the new FCA to Grain-128a is explained in section 5. Section 6 shows the limitations and open problems. Finally, we conclude the paper.

2 Preliminaries

2.1 Notations and Definitions

Some notations are introduced for convenience.

- Given 2 binary row vectors $\boldsymbol{x} = (x_1, \ldots, x_n) \in \mathbb{F}_2^n$ and $\boldsymbol{y} = (y_1, \ldots, y_n) \in \mathbb{F}_2^n$, their inner product is denoted by $\boldsymbol{x} \cdot \boldsymbol{y} = \oplus_{i=1}^n x_i y_i$. The Hamming weight of $\boldsymbol{x}$ are denoted by $wt(\boldsymbol{x})$.

- Let $F : \mathbb{F}_2^m \to \mathbb{F}_2^n$ denote a vectorial Boolean function. A binary linear approximation of F with m-bit input mask $\boldsymbol{u} = (u_1, \ldots, u_m)$ and n-bit output mask pair $\boldsymbol{v} = (v_1, \ldots, v_n)$ can be represented by $\boldsymbol{u}\cdot\boldsymbol{x}\oplus\boldsymbol{v}\cdot F(\boldsymbol{x})$. When we have $1 < a \leq m+n$ linearly independent mask pair $(\boldsymbol{u}_1, \boldsymbol{v}_1), \ldots, (\boldsymbol{u}_a, \boldsymbol{v}_a)$, a vectorial (or multidimensional) linear approximation is denoted by $U\boldsymbol{x} \oplus VF(\boldsymbol{x})$, where the i-th row of (U, V) is $(\boldsymbol{u}_i, \boldsymbol{v}_i)$, $\boldsymbol{x}$ are treated as a column vector unless otherwise stated.

- Linear correlation is used to measure the bias of a binary linear approximation. Let $e(\boldsymbol{x}) = \boldsymbol{u} \cdot \boldsymbol{x} \oplus \boldsymbol{v} \cdot F(\boldsymbol{x})$, the correlation of the binary linear approximation is defined by $c(\boldsymbol{u}, \boldsymbol{v}) = c(e) = 2^{-m} (\#\{\boldsymbol{x} : e(\boldsymbol{x}) = 0\} - \#\{\boldsymbol{x} : e(\boldsymbol{x}) = 1\})$. Similarly, let $\boldsymbol{e}(\boldsymbol{x}) = U\boldsymbol{x} \oplus VF(\boldsymbol{x})$, $\boldsymbol{w}$ is an r bits binary linear mask, the correlation of linear approximation with mask pair $(\boldsymbol{w}U, \boldsymbol{w}V)$ is

$$c(\boldsymbol{w}) = 2^{-m} (\#\{\boldsymbol{x} : \boldsymbol{w} \cdot \boldsymbol{e}(\boldsymbol{x}) = 0\} - \#\{\boldsymbol{x} : \boldsymbol{w} \cdot \boldsymbol{e}(\boldsymbol{x}) = 1\}),$$

  where $\boldsymbol{w}$ is treated as a row vector.

- Let $X \sim P$ denote a discrete random variable follows distribution P and takes values in $\mathbb{F}_2^m$, Its probability density function $p(\boldsymbol{x})$ is denoted by $\left(p_{(0,\ldots,0)}, \ldots, p_{(1,\ldots,1)}\right) = (\Pr(X = (0, \ldots, 0)), \ldots, \Pr(X = (1, \ldots, 1)))$.

- Let $\boldsymbol{a} \in \mathbb{F}_2^m$ denote a binary vector. There is an integer $a = \sum_{i=0}^{m-1} a_{i+1} 2^i$ corresponding to $\boldsymbol{a}$. For convenience, we alternatively use them if there is no ambiguity in the context, especially as a subscript. For example, for a probability density function $\big(p_{(0,\ldots,0)}, \ldots, p_{(1,\ldots,1)}\big)$, we mean the same thing when denote it by $(p_0, \ldots, p_{2^m-1})$.
- Let $M_m(\mathbb{F}_2)$ denote the $m \times m$ matrix ring over $\mathbb{F}_2$. Given a LFSR with rank d and m-bit cell, its generator is denoted by $L(x) = E + C_1 x + C_2 x^2 + \cdots + C_d x^d \in M_m(\mathbb{F}_2)[x]$, where C_d is nonsingular and E is the identity matrix. The number of information bits of $L(x)$ are denoted by $k = d \times m$. If $L(x) \in \mathbb{F}_{2^m}[x]$, it can also be mapped into $GL_m(\mathbb{F}_2)[x]$.
- Give 2 positive integers a and b with $gcd(a, b) = 1$. The b-cyclotomic coset modulo a containing i is denoted by $\mathcal{C}_i = \{i, ib, \ldots, ib^{r-1}\} \mod a$, where r is the smallest positive integer such that $ib^r \cong i \mod a$. The minimal integer in $\mathcal{C}_i$ is called coset header and denoted by $\bar{i}$. All coset headers form a set $\mathcal{R}_{b,a}$.
- Given 2 vectors $\boldsymbol{a} = (a_1, \ldots, a_n) \in \mathbb{R}^n$ and $\boldsymbol{b} = (b_1, \ldots, b_n) \in \mathbb{R}^n$, The notation $\boldsymbol{a} \succ \boldsymbol{b}$ implies that there is at least one $1 \leq j \leq n$ satisfying $a_j > b_j$, while $\preceq$ has reverse meaning.

Walsh-Hadamard Transform

Walsh-Hadamard transform is a spectral tool widely used in cryptanalysis. Let $X \sim P$ denote a discrete random variable which take values in $\mathbb{F}_2^m$. The Walsh-Hadamard transform of X is defined by

$$\mathcal{W}(X)_{\boldsymbol{w}} = 2^{-m} \sum_{\boldsymbol{x} \in \mathbb{F}_2^m} p_{\boldsymbol{x}} (-1)^{\boldsymbol{w} \cdot \boldsymbol{x}}.$$

Since Walsh-Hadamard transform is a linear operator for XOR, let $X = X_1 \oplus X_2 \oplus \cdots \oplus X_k$, we can efficiently compute probability distribution of X with the help of the convolution property

$$p_{\boldsymbol{x}} = \mathcal{W}^{-1}(\mathcal{W}(X_1) \times \cdots \times \mathcal{W}(X_k))_{\boldsymbol{x}}.$$

Square Euclid Imbalance

Relative entropy (or KullbackâĂŞLeibler divergence) is used to measure the difference between two probability distributions P and Q, i.e.,

$$D\left(p(\boldsymbol{x}) \parallel q(\boldsymbol{x})\right) = \sum_{\boldsymbol{x}} p_{\boldsymbol{x}} \log \frac{p_{\boldsymbol{x}}}{q_{\boldsymbol{x}}}.$$

If $p(\boldsymbol{x})$ is close to $q(\boldsymbol{x})$, i.e., $p_{\boldsymbol{x}} = q_{\boldsymbol{x}} + \epsilon(\boldsymbol{x})$, the relative entropy could be approximated by $D\left(p(\boldsymbol{x}) \parallel q(\boldsymbol{x})\right) \approx \frac{1}{2} \sum_x \frac{(p_{\boldsymbol{x}} - q_{\boldsymbol{x}})^2}{q_{\boldsymbol{x}}} + O\left(\epsilon^3(\boldsymbol{x})\right)$. The summation term is usually called capacity, and denoted by $C(p \parallel q)$. Square Euclid Imbalance (SEI) is defined to be the capacity between a probability distribution and uniform distribution, i.e.,

$$\Delta(p(\boldsymbol{x})) = 2^m \sum_{\boldsymbol{x}} \left(p_{\boldsymbol{x}} - \frac{1}{2^m}\right)^2. \tag{1}$$

The following theorem reveals the relationship between SEI and linear correlation.

Theorem 1 ([BJV04]). *Let $X \in \mathbb{F}_2^m$ be a random variable with density function $p_{\boldsymbol{x}}$, then its SEI*

$$\Delta(p(\boldsymbol{x})) = \sum_{\boldsymbol{w}} \hat{\epsilon}^2(\boldsymbol{w}) = \sum_{\boldsymbol{w} \neq \boldsymbol{0}} c^2(\boldsymbol{w}),$$

where $\epsilon(\boldsymbol{x}) = p_{\boldsymbol{x}} - 2^{-m}$, $\hat{\epsilon}(\boldsymbol{w})$ denotes the FWHT of $\epsilon(\boldsymbol{x})$. For convenience, we use $\Delta(p)$ if $\boldsymbol{x}$ is well known in the context, or $\Delta(X)$ if the random variable X with density function $p(\boldsymbol{x})$ is clear. Particularly, we have $c^2(e) = \Delta(p)$ when $m = 1$.

Parity-Check and Characteristic Polynomial

A parity-check corresponds to an equation that fulfills the LFSR output sequence $\boldsymbol{x}_t$. For example, it is well known that any multiples of $L(x) \in \mathbb{F}_{2^m}[x]$ is a parity-check. Usually, only those very sparse parity-checks with a low degree are exploited in FCA.

Let set $\mathcal{H}(\tau+1, d)$ denote all parity-checks with $\tau + 1$ taps and degree at most d, abbreviated by $\mathcal{H}$ without ambiguity. The available parity-checks at position n denoted by $\mathcal{H}^{(n)} \subseteq \mathcal{H}$. Suppose a parity-check for sequence $\boldsymbol{x}_t$ is denoted by

$$G_n \boldsymbol{x}_t + \cdots + G_1 \boldsymbol{x}_{t+n-1} + E \boldsymbol{x}_{t+n} = 0, \tag{2}$$

where G_n is nonsingular. Its characteristic polynomial is denoted by $F_n(x) = \det(Ex + A) = \det(\sum_{i=0}^{n} G_{n-i} x^i)$, where A denotes the companion matrix

$$A = \begin{pmatrix} 0 & E & 0 & 0 & \cdots & 0 \\ 0 & 0 & E & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 0 & E \\ G_n & G_{n-1} & G_{n-2} & \cdots & G_2 & G_1 \end{pmatrix}.$$

2.2 A Brief Description of Original FCA

Meier and Staffelbach's original FCA includes a precomputation phase and a decoding phase.

Precomputation Phase

Let LFSR's generator polynomial $L(x) \in \mathbb{F}_2[x]$. The purpose of the precomputation phase is to find sufficient very sparse parity-checks with a low degree, which is a hard open problem. One way recommended by Zeng [ZYR91] is evaluating logarithms in finite fields of characteristic 2. It is rather efficient to find low weight multiples, but the degree is not promised to be low. Another way is by extended K-tree algorithm based on general birthday collision [NS15]. The extended k-tree algorithm can be used to find low-weight multiples of a polynomial with not-so-large degrees with flexible parameters.

Decoding Phase

The decoding phase targets to recover the initial state of LFSR from key stream. Suppose we have found sufficient suitable parity-checks $x_n \oplus a_l^{(n)} = 0$, where $a_l^{(n)}$ is the sum of τ taps $a_l^{(n)} = \sum_{k=1}^{\tau} x_{n-l_k}$. The check value is $z_n \oplus b_l^{(n)}$, where $b_l^{(n)} = \sum_{k=1}^{\tau} z_{n-l_k}$ is the sum of τ key stream bits corresponding to x_{n-l_k}. The nonlinear part of a stream cipher is modeled as a binary symmetric channel (BSC), the crossover probability is $p = \Pr[x_n \oplus z_n = 1]$. The critical part of the decoding phase is calculating a posterior probability (APP) with prior distribution symbol by symbol. Suppose that the check values are all 0 for a subset $\mathcal{H}_0 \subseteq \mathcal{H}$, then by Bayes' formula,

$$p^* = \frac{p \prod_{l \in \mathcal{H}_0} (1 - s_l) \prod_{l \in \mathcal{H} \setminus \mathcal{H}_0} s_l}{p \prod_{l \in \mathcal{H}_0} (1 - s_l) \prod_{l \in \mathcal{H} \setminus \mathcal{H}_0} s_l + (1-p) \prod_{l \in \mathcal{H} \setminus \mathcal{H}_0} (1 - s_l) \prod_{l \in \mathcal{H}_0} s_l},$$

where each $s_i = s(p_{l_1}, \ldots, p_{l_\tau}) = \Pr[a_l^{(n)} = b_l^{(n)}]$ depends on the probability of τ symbols involved in parity-check. Moreover, s_l can be calculated recursively in the BSC model

$$s(p_{l_1}, \ldots, p_{l_\tau}) = p_{l_\tau} s(p_{l_1}, \ldots, p_{l_{\tau-1}}) + (1 - p_{l_\tau})(1 - s(p_{l_1}, \ldots, p_{l_{\tau-1}})).$$

The specific process is depicted in Algorithm 1. For more details, we refer to the original paper [MS89].

Algorithm 1 Meier and Staffelbach's binary iterative decoding Algorithm B

```
Input: A key stream sequence $\boldsymbol{z}$ of length $N$ and $\mathcal{H}$.
 1. Calculate the probability threshold $p_{thr}$ and quantity threshold $N_{thr}$.
 2. For round $r \in \{1, 2, \ldots\}$ do
 3.    For iteration $i$ from 1 to a small integer do
 4.       Calculate APP $p^*$ from priori probability $p$, assign $p^*_n = p_n$ for all position $n$.
 5.       If $N_w \geq N_{thr}$ where $N_w = |\{n|p_n > p_{thr}\}|$ then, break; EndIf
 6.    EndFor
 7.    Complement the bits of $\boldsymbol{z}$ with $p_n > p_{thr}$.
 8.    Reset all positions to initial probability $p$.
 9.    If $\boldsymbol{z}$ satisfies all parity-checks then, break; EndIf
10. EndFor
11. Terminate with $\boldsymbol{x} = \boldsymbol{z}$.
```

3 Fast Correlation Attack Based on Vectorial Iterative Decoding Algorithm

3.1 Channel Model

The FCA based on binary linear approximations usually deploys the binary symmetric channel (BSC). Similarly, when the transmitted w-bit word is $\boldsymbol{x}$, and received word is $\boldsymbol{z} = \boldsymbol{x} \oplus \boldsymbol{e}$, we can model it as the symmetric channel (SC). Its transition matrix has the following properties. Each row is a permutation of another row, and so as to columns. Moreover, the sum of each row equals 1 by the definition of SC. SC can be treated as an extended BSC. Its channel capacity is $C = w - H(\boldsymbol{r})$, where $\boldsymbol{r}$ denotes a row of the transition matrix.

Suppose we have a linear approximation with dimension m, i.e.,

$$\bigoplus_{\substack{i\in\{1,\ldots,\#\mathcal{T}_x\}\\ j(i)\in\mathcal{T}_x}} U_i\boldsymbol{x}_{j(i)} \oplus \bigoplus_{\substack{i\in\{1,\ldots,\#\mathcal{T}_z\}\\ j(i)\in\mathcal{T}_z}} V_i\boldsymbol{z}_{j(i)} = \boldsymbol{e}. \tag{3}$$

where $\mathcal{T}_x$ and $\mathcal{T}_z$ are sets of indexes related to linear approximation, all U_i and V_i are $m \times w$ matrices over $\mathbb{F}_2$, both $\boldsymbol{x}_{j(i)}$ and $\boldsymbol{z}_{j(i)}$ are w-bit vectors, $\boldsymbol{e}$ is a m-bit noise. Similarly as BSC, the channel noise vector $\boldsymbol{e}$ is XORed to $\bigoplus_{i\in\{1,\ldots,\#\mathcal{T}_x\},j(i)\in\mathcal{T}_x} U_i\boldsymbol{x}_{j(i)}$, and the output is $\bigoplus_{i\in\{1,\ldots,\#\mathcal{T}_z\},j(i)\in\mathcal{T}_z} V_i\boldsymbol{z}_{j(i)}$, see Fig. 3.1.

Remark 1. When we are discussing a generic multidimensional linear approximation, we can always obtain a linear approximation with form $U\boldsymbol{x}' \oplus V\boldsymbol{z}'$, i.e., only including one input vector $\boldsymbol{x}'$ and one output vector $\boldsymbol{z}'$, for example, by rewriting $\boldsymbol{x}'$ to a larger input vector of dimension $w \times \#\mathcal{T}_x$. Thus the rank of U becomes larger than those U_i. However, although we are interesting to those linear approximations with large dimensions and large SEI, the SEI is hard to always increase sufficiently as the dimension increases. Thus we pick multidimensional linear approximation with form (3) as a generic form.

3.2 Checking Parity with Vectorial Noise

Let $l \in \mathcal{H}^{(n)}$ denote a specific parity-check:

$$l : E\boldsymbol{x}_n \oplus G_1\boldsymbol{x}_{n-1} \oplus \cdots \oplus G_n\boldsymbol{x}_{n-d} = \boldsymbol{0}. \tag{4}$$

where E is the $w \times w$ identity matrix, and all G_i are $w \times w$ square matrices, which is a common case for the LFSR over the finite field $\mathbb{F}_2^w$ or the matrix ring $M_w(\mathbb{F}_2)$.

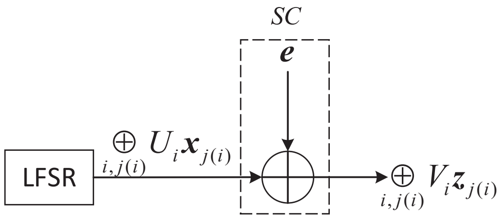
SC
$\boldsymbol{e}$
LFSR
$\bigoplus_{i,j(i)} U_i \boldsymbol{x}_{j(i)}$
$\bigoplus_{i,j(i)} V_i \boldsymbol{z}_{j(i)}$

Figure 1: Channel model for VFCA

Since all U_i in (3) are $m \times w$ matrices, we multiply (4) with U_i, and we acquire that

$$U_i E \boldsymbol{x}_{n+j(i)} \oplus U_i G_1 \boldsymbol{x}_{n-1+j(i)} \oplus \ldots \oplus U_i G_d \boldsymbol{x}_{n-d+j(i)} = \mathbf{0}.$$

In order to check parity over matrix ring, we require that for each G_k, there is a $m \times m$ matrix G'_k such that $U_i G_k = G'_k U_i, \forall i \in \{1, \ldots, \#\mathcal{T}_x\}$. Here U_i, G_k, G'_k are $m \times w$, $w \times w$ and $m \times m$ matrices respectively. Thus we have

$$E(U_i \boldsymbol{x}_{n+j(i)}) \oplus G'_1(U_i \boldsymbol{x}_{n-1+j(i)}) \oplus \ldots \oplus G'_d(U_i \boldsymbol{x}_{n-d+j(i)}) = \mathbf{0}, i \in \{1, \ldots, \#\mathcal{T}_x\}.$$

According to (3), we can sum them up, and check parity by substituting those $U_i \boldsymbol{x}_{n+j(i)}$, $\ldots$, $U_i \boldsymbol{x}_{n-d+j(i)}$ with the observed values. Thus we have

$$\bigoplus_{i=0}^{d} G'_i \left(\bigoplus_{j=1}^{\#\mathcal{T}_x} U_j \boldsymbol{x}_{n-i+k(j)} \right) = \bigoplus_{i=0}^{d} G'_i \left(\bigoplus_{j=1}^{\#\mathcal{T}_z} V_j \boldsymbol{z}_{n-i+k'(j)} \right) \oplus \bigoplus_{i=0}^{d} G'_i \boldsymbol{e}_{n-i}, \tag{5}$$

where $k(j) \in \mathcal{T}_x$, $k'(j) \in \mathcal{T}_z$, $G'_0 = E$, and $\boldsymbol{e}_{n-i}$ is a m-bit noise vector. Consequently, the purpose is to determine $\boldsymbol{e}_{n-i}$ of each position, when observing $\bigoplus_{j=1}^{\#\mathcal{T}_z} V_j \boldsymbol{z}_{n-i+k'(j)}$.

This process can be done for all parity-checks in $\mathcal{H}^{(n)}$. Notice that the approach here is generic. When the parity-checks and linear approximations have special forms, a more efficient checking approach is feasible, see section 5.2. To describe the effect of these parity-checks, we divide them into two sets. Let H_I include those parity-checks whose coefficients are all E, while H_{II} includes the rest. They are called type I and type II parity-checks respectively, which play different roles in the iterative decoding phase.

Notice that there is no need that all $G_i = E$ as in linear distinguishing attack in large alphabets [YJM20], which is expected to have a very high degree. For example, the degree of those special parity-checks with weight 4 of SNOW 3G is expected to be 2^{172}.

3.3 Vectorial Iterative Decoding Algorithm

3.3.1 Iterative Process

In this subsection, we consider how to extract information from a noisy sequence by a vectorial iterative decoding algorithm. Firstly, we try to generalize the original Algorithm B, then improve the iterative criteria.

Let $\#\mathcal{H}^{(n)} = h$ denote the number of parity-checks with $\tau + 1$ taps at position (or clock) n. Let $\boldsymbol{e}_1 \ldots \boldsymbol{e}_N$ denote the sequence of noises, and $\boldsymbol{z}'_1 \ldots \boldsymbol{z}'_N$ denote the derived sequence from key stream $\boldsymbol{z}_1 \ldots \boldsymbol{z}_N$ by $\bigoplus_{i \in \{1,\ldots,\#\mathcal{T}_z\}, j(i) \in \mathcal{T}_z} V_i \boldsymbol{z}_{j(i)}$. The initial priori distribution P is the same for each $\boldsymbol{e}_n$, which is derived by linear approximation. Let $p_\zeta^{(n)} = \Pr[\boldsymbol{e}_n = \zeta, \zeta \in \mathbb{F}_2^m]$ denote its density function, then the APP $p_\zeta^{*(n)}$ could be

Algorithm 2 Calculate the nominator

```
Input: priori p.d $p_\zeta^{(n)}$
Let priori probability distribution $\boldsymbol{p}^{(n)} = (p_0, p_1, \ldots, p_{2^m-1})$.
For each parity-check $l \in \mathcal{H}^{(n)}$ do
    Calculate distribution $\boldsymbol{p}(l)$ of $\sum_{i=1}^{\tau} G'_{l_i} \boldsymbol{e}_{n-l_i}$ by FWHT and convolution property.
    Permute $p(l)_{\boldsymbol{x}} \leftarrow p(l)_{\boldsymbol{x} \oplus \zeta}$, $\boldsymbol{x} \in \mathbb{F}_2^m$.
End For.
Multiply corresponding coordinate together of all these $\boldsymbol{p}(l)$.
```

Algorithm 3 Vectorial iterative decoding

```
Input: The sequence $\boldsymbol{z'}$ of length $N$ derived from key stream,
       The sequence of noises $\boldsymbol{e}$ with initial p.d. $\boldsymbol{p}$,
       The parity-checks set $\mathcal{H}$ with $\tau + 1$ taps.
parameters: Maximal rounds $R$, maximal iterations $T$ and minimal gap $G$ to infuse new noises.
1.  $\boldsymbol{pri} \leftarrow \boldsymbol{p}$, $\boldsymbol{E}^{glb} = (E_1^{glb}, \ldots, E_{2^m-1}^{glb}) \leftarrow \boldsymbol{0}$.
2.  For $r = 1, 2, \ldots, R$ do
3.      $\boldsymbol{E}^{rnd} = (E_1^{rnd}, \ldots, E_{2^m-1}^{rnd}) \leftarrow \boldsymbol{0}$, $\zeta \leftarrow \boldsymbol{0}$.
4.      For $i = 1, 2, \ldots, T$ do
5.          $\boldsymbol{E}^{itr} = (E_1^{iter}, \ldots, E_{2^m-1}^{iter}) \leftarrow \boldsymbol{0}$.
6.          For $n = 1, 2, \ldots, N$ do
7.              Compute $\boldsymbol{app}$ from $\boldsymbol{pri}$ by equation (6).
8.              If $p_j^{(n)} > p_0^{(n)}$ then $E_j^{itr} \leftarrow E_j^{itr} + 1/N, j \in \{1, 2, \ldots, 2^m - 1\}$. End If.
9.          End For.
10.         If $\boldsymbol{E}^{itr} \succ \boldsymbol{E}^{rnd}$ then $\boldsymbol{E}^{rnd} \leftarrow \boldsymbol{E}^{itr}$, $\boldsymbol{pri} \leftarrow \boldsymbol{app}$. End If.
11.         If $\boldsymbol{E}^{itr} \preceq \boldsymbol{E}^{rnd}$ or $i = T$ then
12.             If $\boldsymbol{E}^{itr} = \boldsymbol{0}$ then return failed.
13.             else if $||\boldsymbol{E}^{rnd} - \boldsymbol{E}^{glb}|| < G$ then reset $\boldsymbol{z'} \leftarrow \boldsymbol{z'} \oplus \boldsymbol{n}$, break.
14.             else $\boldsymbol{E}^{glb} \leftarrow \boldsymbol{E}^{rnd}$, select $\zeta$ that maximizes $E_{int(\zeta)}^{rnd} + E_{int(\zeta)}^{itr}$, break. End If.
15.         End If.
16.     End For.
17.     If $\zeta \neq \boldsymbol{0}$ then complement all positions of $\boldsymbol{z'}$ such that $p_\zeta > p_{\boldsymbol{0}}$ with $\zeta$. End If.
18.     If $\boldsymbol{z'}$ satisfies all parity-checks then return success. End If.
19.     Reset $\boldsymbol{pri} \leftarrow \boldsymbol{p}$.
20. End For.
21. Terminate.
```

computed by Bayes's formula.

$$
\begin{aligned}
p_\zeta^{*(n)} &= \Pr\left[\boldsymbol{e}_n = \zeta | \text{when observed check values } (\boldsymbol{c}_1, \boldsymbol{c}_2, \ldots, \boldsymbol{c}_h)\right] \\
&= \frac{p_\zeta^{(n)} \prod_{l \in \mathcal{H}^{(n)}} \Pr[\bigoplus_{i=1}^{\tau} G'_{l_i} \boldsymbol{e}_{n-l_i} = \boldsymbol{c}_l \oplus E\zeta]}{\bigoplus_\eta p_\eta^{(n)} \prod_{l \in \mathcal{H}^{(n)}} \Pr[\bigoplus_{i=1}^{\tau} G'_{l_i} \boldsymbol{e}_{n-l_i} = \boldsymbol{c}_l \oplus E\eta]}.
\end{aligned} \tag{6}
$$

We always assume $\boldsymbol{e}_{n-l_i}$ are independent and all parity-checks are orthogonal. As $\zeta \in \mathbb{F}_2^m$ runs over the alphabet, $\Pr[\sum_{i=1}^{\tau} G'_{l_i} \cdot \boldsymbol{e}_{n-l_i} = \boldsymbol{c}_l + E \cdot \zeta]$ can be calculate by convolution property and FWHT. Thus, the nominator and denominator can be computed by Algorithm 2.

The vectorial iterative decoding algorithm is listed in Algorithm 3. In line 1, the prior distribution is initialized with the noise distribution $\boldsymbol{p}$, and a global empirical vector $\boldsymbol{E}^{glb}$ is initialized with $\boldsymbol{0}$. In line 3 of the main loop, the round empirical vector $\boldsymbol{E}^{rnd}$ and the coin ζ are all set to be zero. In line 5 of the iteration loop, an iteration empirical vector $\boldsymbol{E}^{itr}$ is set to be zero. For each symbol, we compute the $\boldsymbol{app}$ and increase the empirical vector $\boldsymbol{E}^{itr}$, see line 7 and 8. If $\boldsymbol{E}^{itr}$ is still increasing, then we assign $\boldsymbol{E}^{rnd}$ with $\boldsymbol{E}^{itr}$ and $\boldsymbol{pri}$ with $\boldsymbol{app}$, and continue the iteration, see line 10. Otherwise, if $\boldsymbol{E}^{rnd}$ is close to $\boldsymbol{E}^{glb}$, i.e., the algorithm may correct very few errors, we inject an appropriate noise sequence and break the current loop, which is a new criterion, see line 13. Otherwise, we choose a coin ζ which is likely to correct more errors and break the current loop in line 14. The complements in Algorithm 3 are applied on the derived sequence $\boldsymbol{z'}$. The n-th position $\boldsymbol{z}'_n$ is changed to $\boldsymbol{z}'_n \oplus \zeta$ when the noise $\boldsymbol{e}_n$ is determined to be ζ and the complement is performed. If $\boldsymbol{z'}$ satisfies all parity-checks at the end, we just deduce that all $\boldsymbol{e}_i = \boldsymbol{0}$. Thus with the help of LFSR's feedback polynomial, the initial state of LFSR can be recovered. The criteria that are used to break the iterative loop and trigger a resetting process are the main factors affecting the speed of convergence [CGD96, MG91].

3.3.2 Iterative Criteria

The criteria used in the vectorial algorithm could be summarized in two points.

Criterion 1. Passing through sufficient iterations before breaking up and resetting, which corresponds to line 7-10 and 14. More specifically, if new ***app*** strengthens the empirical complement effect and iterations are less than maximal, then continue iteration by Bayes's rule. Otherwise, select the complement coin which has the potential largest empirical complement effect.

Criterion 2. When the empirical complement effect is weak from the previous round to the current round, a sequence of very biased noises is infused in order to break the tie caused by the self-combination property of LFSR. The noises' SEI is required to be appropriate, neither very large to counteract the previous decoding work nor very small to break the tie.

These criteria are motivated by scaled experiments when parity-checks are not so many, and proposed for different purposes. On one hand, notice that if the loop is broken when achieves a preset threshold as in Algorithm B, it is easier to be triggered in the earlier rounds than in the later rounds. However, when a complement is performed very early without passing through enough iterations, it will pull the algorithm into a tie state very early and weaken the decoding efficiency. A tie state represents that the decoding algorithm reaches the point where the iterations fail to improve the correction of the key stream sequence. To improve this, Criterion 1 is proposed to avoid converging to a tie state too early. We hope it will help to correct errors as many as possible in each of the early rounds.

On the other hand, notice that once the algorithm entered into a tie state, i.e., the number of right complements and the wrong complements are almost equal, the correcting effect is very weak. However, If we only reset the iterative process, the experiments show that the new process will enter into the tie state again without profits. The main reason is that the noise errors are no longer independent with the parity-checks in higher rounds. Therefore, a new sequence of biased noises is XORed to the sequence $\boldsymbol{z'}$ to get out of the trap. We hope it will improve the convergency to some extent.

Intuitively, a tie state is likely to appear, when it is close to the bound of decoding ability. Let $\boldsymbol{e'} = \boldsymbol{e}'_0, \boldsymbol{e}'_1, \ldots, \boldsymbol{e}'_{N-1}$ denote the current noise sequence after many rounds. In higher rounds, the check result maybe indicate the $\boldsymbol{e}'_i$ is a swing error for many $i \in \{0, 1, \ldots, N-1\}$, i.e., either to be $\boldsymbol{0}$ or ζ. Complementing those $\boldsymbol{e}'_i$ with $\boldsymbol{e}'_i \oplus \zeta$ will have no profits. However, injecting noise $\boldsymbol{n}$ maybe weaken the dependency between $\boldsymbol{e'}$ and the parity-checks. The new sequence $\boldsymbol{e'} \oplus \boldsymbol{n}$ maybe far away from the original $\boldsymbol{e'}$ in the sense of the check result. Thus it may pull the algorithm out of the tie state when both the number of parity-checks and the SEI of initial noise are not too small. Thus we require that the SEI of the infused noise is not too small, or it will counteract all previous iterative correcting processes. Meanwhile, the SEI of the infused noise should not be too large, or the pull will be too small to get out of the trap, as the changes for $\boldsymbol{e'}$ are not enough in this case. As for the binary case, i.e., $\boldsymbol{e'}$ is a binary sequence, the difference is that a tie state maybe satisfy that almost half of parity-checks hold. Thereby, it seems

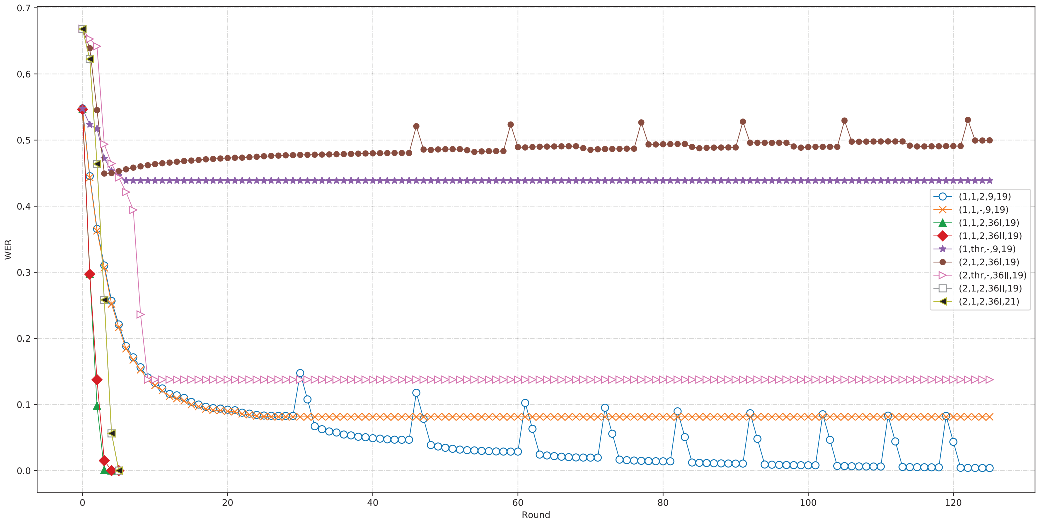

Figure 2: Several vectorial iterative decoding curves of scaled experiments

that the idea may be applied to some binary iterative algorithms. The experiment results in the next subsection also provide an evidence.

Remark 2. Regardless of the differences in criteria, the original FCA proposed by Meier et. al. can be treated as a special case of the new FCA with dimension $m = 1$. The coefficient matrices of LFSR degenerate to scalar elements in $\mathbb{F}_2$. Therefore, the commutative condition for coefficient matrices of parity-checks is not needed to be considered. The multidimensional linear approximation degenerates to a binary linear approximation, and ζ must be 1.

3.4 Scaled Experiments and A Discussion for Comparison

3.4.1 Small Scale Experiments

In this section, we perform scaled experiments to verify the vectorial iterative algorithm and the idea of Criterion 2. The first scaled experiments show the validity of the vectorial iterative algorithm. The second experiment shows that the idea of Criterion 2 may be applied to other binary iterative algorithms.

The first experiment settings are as follows. The generator polynomial of LFSR is $g(x) = x^{16} + x^{15} + x + \alpha \in \mathbb{F}_{2^2}[x]$, where α is the primitive element of $\mathbb{F}_{2^2}$. The output of LFSR at time t is $\boldsymbol{x}_t$. The noise stems from a SC channel instead of nonlinear part of a stream cipher. The target is recovering LFSR output sequence $\boldsymbol{x}_1\boldsymbol{x}_2\ldots\boldsymbol{x}_N$ from noisy sequence $\boldsymbol{z}_1\boldsymbol{z}_2\ldots\boldsymbol{z}_N = (\boldsymbol{x}_1\boldsymbol{x}_2\ldots\boldsymbol{x}_N) \oplus (\boldsymbol{e}_1\boldsymbol{e}_2\ldots\boldsymbol{e}_N)$.

We tweak the parameters such as channel capacity, the number of parity-checks and the infused noises to verify the word-error ratio (WER) after iterating a number of rounds. Specifically, the density functions of 2 priori distributions P_1 and P_2 are $(0.45, 0.25, 0.2, 0.1)$ and $(0.33, 0.25, 0.22, 0.20)$ respectively. The length of data is $N = 2^{19}$ or 2^{21} key stream words. The number of parity-checks with $\tau = 2$ are $h = 9$, $h_I = 36$ or $h_{II} = 36$. The results of experiment are illustrated in 2. For example, the curve $(1, 1, 2, 9, 19)$ denotes the result derived by parameters P_1, $h = 9$, $N = 2^{19}$ with Criteria 1 and 2. The curve $(2, \text{thr}, -, 36II, 19)$ denotes the result derived by parameters P_2, $h_{II} = 36$, $N = 2^{19}$ with threshold criterion like Algorithm B.

Some observations could be induced from Figure 2. Firstly, comparing the curve $(1, 1, 2, 9, 19)$ with $(1, 1, 2, 36I, 19)$, we see that the speed of convergence increases with the number of parity-checks when channel capacity is fixed. Secondly, comparing the curve $(1, 1, 2, 9, 19)$ with $(1, 1, -, 9, 19)$, we see that Criterion 2 indeed improves the convergence. The algorithm will enter into a tie state at about the 29th round without Criterion 2. However, it can't improve convergency, when the SEI is very small, see $(3, 1, 2, 36I, 19)$. This result verifies our above statements. Thirdly, from the curves $(1, 1, -, 9, 19)$ and $(1, thr, -, 9, 19)$, it seem that Criterion 1 improves the convergence too. Finally, it seems that more data also improves the convergence from $(2, 1, 2, 36I, 19)$ $(2, 1, 2, 36I, 21)$. Notice that $(2, 1, 2, 36I, 19)$ seems be worse than $(2, 1, 2, 36II, 19)$. The reason is that the length of key stream $N = 2^{19}$ is not sufficiently large comparing with the degrees. Therefore, the average feasible parity-checks for both the head and tail segments of the key stream in $(2, 1, 2, 36I, 19)$ are less than in $(2, 1, 2, 36II, 19)$.

As stated in Section 3.3.2, the new Criterion 2 may help to improve some other binary algorithms. We take Algorithm B in [MS89] and MIPD Algorithm in [CGD96] as examples to perform another scaled experiment. Thereby, we have 4 binary algorithms, i.e., the Algorithm B, the MIPD Algorithm and their modified versions with Criterion 2. The parameters of the experiments are as follows. The length of the LFSR over $\mathbb{F}_2$ is 32-bit. The noised bit z_t is derived by XORing the output of LFSR x_t with noise bit e_t. The probability of error $\Pr(e_t = 1) = 0.378$ for the Algorithm B and its modified version, while the value is 0.371 for MIPD and its modified version. For both cases, we use 9 parity-checks with 3 taps and 2^{20} bits data (key stream). For Algorithm B, we inject a noise sequence whenever N_w is still very small after the iterations. For the MIPD Algorithm, we inject a noise sequence whenever the number of holding parity-checks changes is very small. Figure 3 illustrates the first 175 rounds of bit-error ratio (BER) for the 4 algorithms. Noticed that the way we embed Criterion 2 into the original algorithms may not be optimal. However, Criterion 2 still improves the convergence in both cases. The results imply that Criterion 2 maybe also work for some other binary algorithms when they approach the decoding boundary. It seems that Criterion 2 has better performance in Algorithm B than in the MIPD algorithm. The main reason is that the complements are performed in each iteration, which means that the infused noises may be eliminated slowly. Thus we injected a slight noise sequence. However, the Algorithm B complements z_t after some iterations in each round, which means the infused noises may be cleared fast. Thereby, in order to make the improvement more obvious to be illustrated in a figure, we choose a slightly smaller initial probability in comparison MIPD Algorithm with its modified version.

3.4.2 A Discussion about the Potential Advantages

Although the new criteria may be effective, we can not directly compare the vectorial algorithm with a binary algorithm. The reason is that it is hard to compare them under equal status. Firstly, as the same number of vectorial parity-checks and binary parity-checks does not mean the equivalent comparison condition, it is hard to choose the two numbers to make the two algorithms stand at the same starting line so far as we know. Secondly, the corresponding concept is the bit error ratio (BER) in the binary case instead of the word error ratio (WER). The vectorial algorithm aims at small WER, while the binary one aims at small BER. Small BER does not strictly mean small WER and vice versa. These differences prevent us from giving a precise and fair comparison.

However, since the purpose of proposing the vectorial decoding algorithm is to deploy the multidimensional linear approximation, whose SEI may be significantly larger than that of the binary linear approximation, we can observe some cases where the vectorial algorithm may have advantages. For example, let the l-bit LFSR is defined in $\mathbb{F}_{2^m}$, where l is a multiple of m, the SEI of the multidimensional linear approximation is $2^{-\gamma}$. Since the number of parity-checks should not be smaller than $\left(\frac{2^m-1}{2^m}\right)^{\tau-1} 2^{\frac{(2m+\gamma)(\tau-1)}{2}}$, i.e., the equation (18) in Section 4.2.1, we need at least $2^{\gamma/2}(2^m - 1)$ parity-checks with 3 taps. Thus the length N of data needed satisfies $(2^m - 1)^2 2^{-l} \binom{N}{2} \approx 2^{\gamma/2}(2^m - 1)$ by a birthday

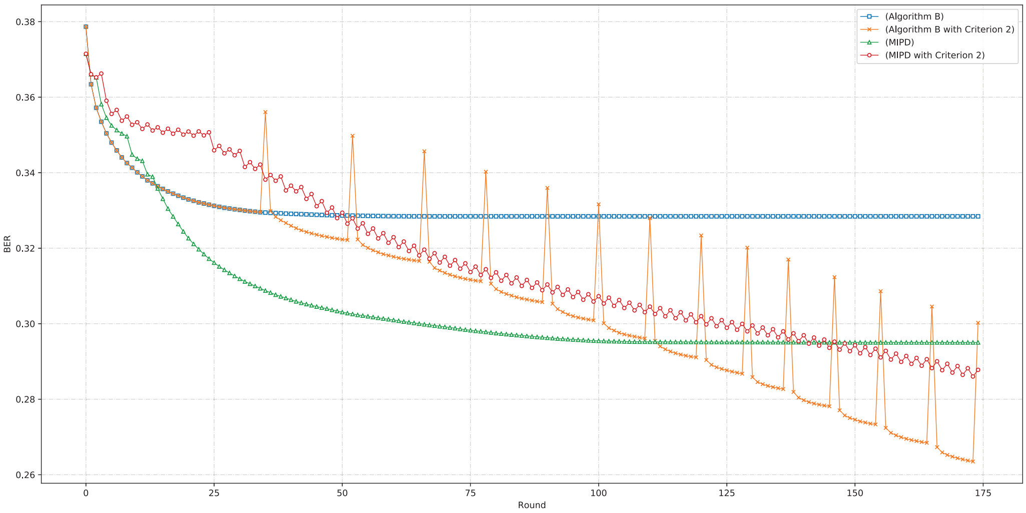

Figure 3: An example for comparing Criterion 2

collision, which means $N \approx 2^{(\gamma+2l+2)/4}/\sqrt{2^m-1}$. When $m = 1$, $N \approx 2^{(\gamma+2l+2)/4}$. For the vectorial case, N seems to be smaller than the binary case, because that $m > 1$ and γ is expected to be smaller than the binary case. For more details about these relationships, we refer to Section 4.2.1.

4 Some Cryptographic Properties for Vectorial Iterative Algorithm

4.1 Some Statistical Properties of the Iteration

4.1.1 Convergence Property

It is necessary to figure out the convergence property when iteratively computing APP. Intuitively, we hope that APP $p_\zeta^{*(n)}$ increases when noise variable $\boldsymbol{e}_n = \zeta$ and decreases when $\boldsymbol{e}_n \neq \zeta$. Its expected value is computed as follows.

$$\begin{aligned} E_0[p_\zeta^{*(n)}] =& E[p_\zeta^{*(n)} | \boldsymbol{e}_n = \zeta] \\ =& \sum_{(\boldsymbol{c}_1,\dots,\boldsymbol{c}_h)} \frac{p_\zeta^{(n)} \left(\prod_{l\in\mathcal{H}^{(n)}} \Pr[\sum_{i=1}^{\tau} G'_{l_i}\boldsymbol{e}_{l_i} = \boldsymbol{c}_l + E\zeta]\right)^2}{\sum_\zeta p_\zeta^{(n)} \prod_{l\in\mathcal{H}^{(n)}} \Pr[\sum_{i=1}^{t} G'_{l_i}\boldsymbol{e}_{l_i} = \boldsymbol{c}_l + E\zeta]}, \end{aligned}$$

$$\begin{aligned} E_1[p_\zeta^{*(n)}] =& E[p_\zeta^{*(n)} | \boldsymbol{e}_n \neq \zeta] \\ =& \sum_{\zeta'\neq\zeta} \sum_{(\boldsymbol{c}_1,\dots,\boldsymbol{c}_h)} \frac{p_\zeta^{(n)} \prod_{l\in\mathcal{H}^{(n)}} \Pr[\sum_{i=1}^{\tau} G'_{l_i}\boldsymbol{e}_{l_i} = \boldsymbol{c}_l + E\zeta]}{\sum_\zeta p_\zeta^{(n)} \prod_{l\in\mathcal{H}^{(n)}} \Pr[\sum_{i=1}^{\tau} G'_{l_i}\boldsymbol{e}_{l_i} = \boldsymbol{c}_l + E\zeta]} \\ & \frac{p_{\zeta'}^{(n)} \prod_{l\in\mathcal{H}^{(n)}} \Pr[\sum_{i=1}^{\tau} G'_{l_i}\boldsymbol{e}_{l_i} = \boldsymbol{c}_l + E\zeta']}{1 - p_\zeta^{(r)}}. \end{aligned}$$

And we conclude that $E[p^{*(n)}] = p_\zeta E_0[p^{*(n)}] + (1-p_\zeta)E_1[p^{*(n)}] = p_\zeta$.

Example 1. Let the generator polynomial of LFSR $L(x) \in \mathbb{F}_{2^2}[x]$ with degree 16. We get the increasing and decreasing ratios in Table 1 when exploits 3 type I parity-checks with 3

Table 1: An example of increasing and decreasing ratio

x	0	1	2	3
p_x	0.4500	0.2500	0.2000	0.1000
E'_0/p^*	1.02618712	1.00117564	1.02744428	1.10462318
E'_1/p^*	0.97857418	0.99960812	0.99313893	0.98837520
E_0/p^*	1.03907892	1.06836181	1.16004050	1.19334394
E_1/p^*	0.96802634	0.97721273	0.95998988	0.97851734

taps. The second row is a priori probability distribution P. $E_0[p^*]/[p^*]$ and $E_1[p^*]/[p^*]$ denote the increasing and decreasing ratio. Particularly, E'_0/p^* and E'_1/p^* denote the case only considering the number of holding parity-checks. Both cases meet our expectations.

4.1.2 Decoding Efficiency

In algorithm B, a threshold N_{thr} is computed to promote the efficiency of the complements. It is determined by the intersection point of two shrunk normal distributions, In the multidimensional case, the intersection point becomes an intersection curve (surface). The threshold reflects the correcting ability of the first iteration in the binary case. Although we do not need such a threshold to promote efficiency in the vectorial case, it still reflects the decoding efficiency from the first iteration. Thus we discuss how to estimate the correcting ability by measuring the volume of the intersection area in this subsection.

Let N_ζ^{thr} denote this threshold corresponding to ζ. Without loss of generality, we assume that the priori probability distribution P of noise sequence $\boldsymbol{e}_1 \dots \boldsymbol{e}_N$ s.t. $p_0 \geq p_1 \geq \dots \geq p_{2^m-1} > 0$. Suppose that a random variable $X \sim P$, we require that the distribution of new random variable $G'_{l_i}X$ still has 0 as the maximal value point [1]. It surely holds when G'_{l_i} is nonsingular. This requirement may reduce the number of available parity-checks, but it simplifies the analysis of the effect of parity-checks.

Let $X_1, \dots, X_\tau$ denote τ independent random variables all follow P. Let Q denote the distribution of their linear combination $\sum_{i=1}^{\tau} G'_{l_i} X_i$. Thus Q still has 0 as its maximal value point, which could be deduced from the convolution property and Walsh-Hadamard transform. Particularly, if all $G'_{l_i} = E$, Q preserves the order of P, i.e., $q_0 \geq q_1 \geq \dots \geq q_{2^m-1} > 0$.

The approach to calculate N_ζ^{thr} is inspired by the fact p_ζ^* is large when more check values appear to be ζ. Let $q_{\boldsymbol{c}} = \Pr[\sum_{i=1}^{\tau} G'_{l_i} \boldsymbol{e}_{n-l_i} = \boldsymbol{c}]$ denote the probability that the τ taps sum to be $\boldsymbol{c}$ for parity-check l. Obviously, $q_{\boldsymbol{c}}$ depends on the individual parity-check. This phenomenon makes it very complicated to calculate the threshold N_ζ^{thr}. To simplify the calculation, we divide all parity-checks into two sets $\mathcal{H}_I$ and $\mathcal{H}_{II}$ according to its coefficients, then deal with them separately.

The set $\mathcal{H}_I$ includes all parity-checks whose coefficients are all identity. For this class, $q_{\boldsymbol{c}}$ is obviously independent of parity-checks. Let $\#\mathcal{H}_I = h_I$, the probability the current noise $\boldsymbol{e} = \zeta$ and x_i check values equal $i, i \in \{0, \dots, 2^m - 1\}$ is as follows [2]

$$p_\zeta q(x_0, \dots, x_{2^m-1}, \zeta) = p_\zeta \frac{h_I!}{x_0! \dots x_{2^m-1}!} \prod_{i=0}^{2^m-1} q_{i \oplus \zeta}^{x_i}, \tag{7}$$

where $x_{2^m-1} = h_I - \sum_{i=0}^{2^m-2} x_i$.

[1] Minimal value point is similar. We assume that p_0 is minimal instead.

[2] Actually, check values are vectors in $\mathbb{F}_2^m$, here we use integers i to denote the same thing.

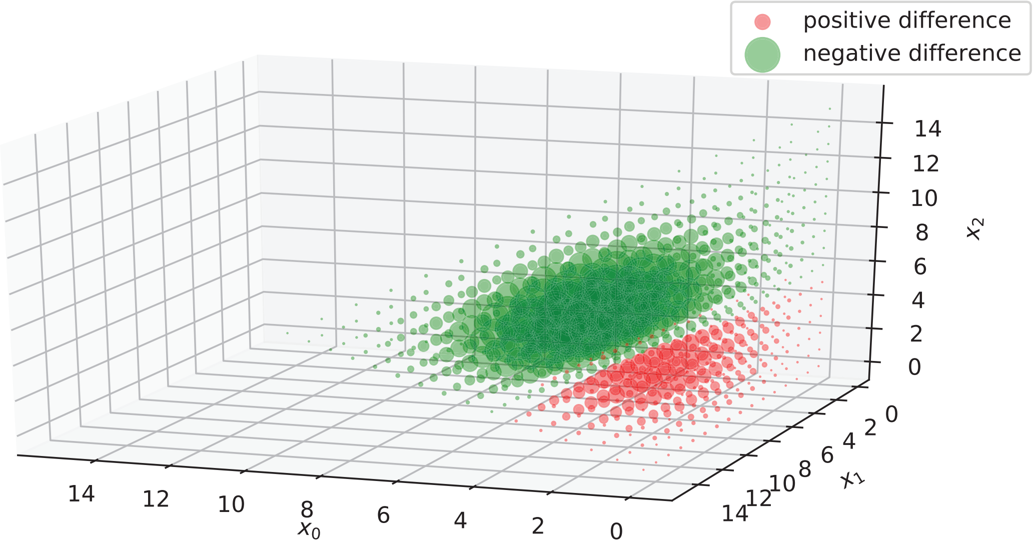

Figure 4: An example for the difference distribution

Obviously, $\boldsymbol{x} = (x_0, \ldots, x_{2^m-1})$ follows multinomial distribution $\text{Multi}(h_I, \boldsymbol{q}_\zeta)$ with parameter $\boldsymbol{q}_\zeta = (q_\zeta, \ldots, q_{2^m-1\oplus\zeta})$. Its density function is denoted by $q(\boldsymbol{x}, \zeta)$. For convenience, we introduce notations

$$\boldsymbol{q}_\zeta^{\boldsymbol{x}} = \prod_{i=0}^{2^m-1} q_{i\oplus\zeta}^{x_i}, \binom{h_I}{\boldsymbol{x}} = \frac{h_I!}{x_0! \ldots x_{2^m-1}!}.$$

Let $\mathcal{A}(\zeta)$ be a subset of all possible $\boldsymbol{x}$. Once we complement those noises with $\zeta \neq 0$ when the vectors in $\mathcal{A}(\zeta)$ are observed, the expected number of correctly complemented noises and erroneously complemented noises are respectively

$$N \times W(P, \mathcal{A}(\zeta), \zeta) = N \sum_{\boldsymbol{x}\in\mathcal{A}(\zeta)} p_\zeta q(\boldsymbol{x}, \zeta), N \times W(P, \mathcal{A}(\zeta), 0) = N \sum_{\boldsymbol{x}\in\mathcal{A}(\zeta)} p_0 q(\boldsymbol{x}, 0), \quad (8)$$

where N denote the length of data. All the other cases of complements are neutral. Thereby, the number of actual corrected positions is the difference

$$N \times I(P, \mathcal{A}(\zeta), \zeta, 0) = N \times W(P, \mathcal{A}(\zeta), \zeta) - N \times W(P, \mathcal{A}(\zeta), 0). \quad (9)$$

Given P and $\mathcal{H}_I$, if we can find a set $\mathcal{A}(\zeta)$ maximizing $I(P, \mathcal{A}(\zeta), \zeta, 0)$, then the expected number of actual corrected positions of each complement should be maximized. Firstly, we observe that the means of the two multinomial distributions are $h_I\boldsymbol{q}_\zeta$ and $h_I\boldsymbol{q}_0$ respectively. Therefore, similar as the binomial case, there is a set $\mathcal{A}(\zeta)$ of $\boldsymbol{x}$ in which $I(P, \mathcal{A}(\zeta), \zeta, 0)$ takes non-negative value.

Since given $\boldsymbol{x}$, $I(P, \mathcal{A}(\zeta), \zeta, 0)$ and $p_\zeta^* - p_0^*$ have the same sign, it is equivalent to find $\mathcal{A}(\zeta)$ such that $p_\zeta^* - p_0^* > 0$ for each $\boldsymbol{x} \in \mathcal{A}(\zeta)$, that is to determine the region $\mathcal{A}(\zeta)$ such that

$$\delta(\zeta, 0) = p_\zeta q(\boldsymbol{x}, \zeta) - p_0 q(\boldsymbol{x}, 0) > 0, \boldsymbol{x} \in \mathcal{A}(\zeta). \quad (10)$$

Example 2. Let initial distribution P and LFSR be the same as in Example 1, and $h_I = 15$. The difference $\delta(\zeta, 0)$ is illustrated in Fig. 4. The green circles denote the negative $\delta(\zeta, 0)$, while the red circles denote the positive $\delta(\zeta, 0)$. The size of circle represents the relative value of $|\delta(\zeta, 0)|$. The non-negative and the negative area are separated. The region $\mathcal{A}(\zeta)$ corresponds to those $\boldsymbol{x}$ which derives red circles.

Table 2: Direct computation and normal approximation for $I(p, \mathcal{A}(1), 1, 0)$

h_I	40	80	200	400
direct computation	0.0686	0.1138	0.1835	0.2266
normal approximation	0.0707	0.1148	0.1841	0.2267

When h is small, it is feasible to evaluate N_ζ^{thr} by exhaustively searching. The threshold N_ζ^{thr} can be determined by

$$N_\zeta^{thr} = N \left(\sum_{\boldsymbol{x} \in \mathcal{A}(\zeta)} \sum_{\eta \in \mathbb{F}_2^m} p_\eta q(\boldsymbol{x}, \eta) \right). \tag{11}$$

The time complexity is about $O(2^m \binom{h_I + 2^m}{2^m})$.

When h_I is large and $\boldsymbol{q}$ is not near the boundary of the parameter space, multivariate normal distribution approximation is suitable. Multi$(h_I, \boldsymbol{q})$ could be approximated by $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ with density function

$$\frac{1}{\sqrt{(2\pi)^{2^m-1} |\boldsymbol{\Sigma}|}} \exp \left(-\frac{1}{2} (\boldsymbol{x} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\boldsymbol{x} - \boldsymbol{\mu}) \right),$$

where superscript T denotes transposition, mean vector $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$ are determined by Multi$(h_I, \boldsymbol{q})$. Therefore, the area $\mathcal{A}(\zeta)$ maximizing the multiple integral

$$I(P, \mathcal{A}(\zeta), \zeta, 0) \approx \int_{\mathcal{A}(\zeta)} \left(p_\zeta \mathcal{N}(\boldsymbol{\mu}_\zeta, \boldsymbol{\Sigma}_\zeta) - p_0 \mathcal{N}(\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0) \right) d\boldsymbol{x} \tag{12}$$

should be part of a hypercube with dimension $2^m - 2$ that restricted by the $2^m - 1$ coordinate plane and two surfaces

$$\begin{aligned} &\Omega_1 : \sum_i^{2^m-2} x_i = h_I, \\ &\Omega_2 : \frac{1}{2} \left((\boldsymbol{x} - \boldsymbol{\mu}_0)^T \boldsymbol{\Sigma}_0^{-1} (\boldsymbol{x} - \boldsymbol{\mu}_0) \right) - \frac{1}{2} \left((\boldsymbol{x} - \boldsymbol{\mu}_\zeta)^T \boldsymbol{\Sigma}_\zeta^{-1} (\boldsymbol{x} - \boldsymbol{\mu}_\zeta) \right) - \ln \frac{p_0}{p_\zeta} = 0. \end{aligned} \tag{13}$$

Notice that Ω_2 is a quadratic form in the real field, the multiple integral (12) can be computed by repeated integral. Once $\mathcal{A}(\zeta)$ is determined, the threshold can be calculated by volume integral

$$N \sum_{\eta \in \mathbb{F}_2^m} \int_{\mathcal{A}(\eta)} \mathcal{N}(\boldsymbol{\mu}_\eta, \boldsymbol{\Sigma}_\eta) d\boldsymbol{x}. \tag{14}$$

Example 3. Let the probability distribution P and LFSR be the same as in Example 1. To illustrate this multivariate normal approximation, $I(P, \mathcal{A}(1), 1, 0)$ is computed by two methods and depicted in Table 2. In order to simplify the integral, we could even slightly adequate the boundary of A without fluctuating the result much.

When the parity-checks stem from $H_{II} = \mathcal{H} \backslash \mathcal{H}_I$, q_c depends on individual parity-check. Thus when the probability value peak is q_0, we introduce a symmetric multinomial distribution Q' to simulate the influences of type II parity-checks, which parameter is

$$q_0' = q_0, q_1' = \cdots = q_{2^m-1}' = \frac{1 - q_0'}{2^m - 1}. \tag{15}$$

Then the calculation is similar as for $\mathcal{H}_I$. According to the size of $\mathcal{H}_I$ and $\mathcal{H}_{II}$, we could estimate N_ζ^{thr} by combine $\mathcal{H}_I$ and $\mathcal{H}_{II}$ together. The multinomial distribution is replaced by Multi$(h_I, \boldsymbol{q}_\zeta)$Multi$(h_{II}, \boldsymbol{q}_\zeta')$ in this case.

Table 3: Theoretical and empirical value of N_ζ^{thr}/N

No. of parity-checks (h_I, h_{II})	ζ	theoretical	empirical		
			$N = 2^{19}$	$N = 2^{20}$	$N = 2^{21}$
(36,0)	1	0.277133	0.227242	0.250517	0.264012
	2	0.253926	0.242359	0.246835	0.249339
	3	0.200412	0.164480	0.181245	0.190250
(18,18)	1	0.297959	0.251286	0.270056	0.279394
	2	0.260769	0.220915	0.238914	0.248543
	3	0.167968	0.125576	0.144096	0.154273
(0,138)	1	0.376058	0.360392	0.364783	0.368026
	2	0.325561	0.321800	0.332389	0.338674
	3	0.221771	0.198662	0.213513	0.221388

Example 4. To verify the validity of these approximations, with the same P and LFSR as in Example 1, we compute the theoretical ratio of N_ζ^{thr}/N and the empirical ratio by the ratio where $p_\zeta^* > p_0^*$. Table 3 depicts that our estimations are very precise.

We also give some direct properties from the point view of information theory in Appendix B, which maybe imply some relationships among the decoding efficiency, the initial noise distribution and the number of the parity-checks.

4.2 Two bounds Related to Cryptanalysis Complexity

As the number of parity-checks h influences the decoding complexity. We focus on the property of the first iteration in the first round, which seems to be the critical part by the previous section, and discuss how to deduce some theoretical bounds for h as well as key stream length N.

4.2.1 A Bound Derived from Decoding Codes

Similarly, as Proposition 1 in [CS91], in order to perform error-corrected iterative decoding, the lower bounds of h should satisfy that there exists at least a ζ such that $p_\zeta^* > p_0^*$. It is summarized as follows.

Proposition 1. *If iterative decoding is feasible, then there is at least one $\zeta \in \{1, 2, \ldots, 2^m - 1\}$ such that $p_\zeta q(\boldsymbol{x}, \zeta)/(p_0 q(\boldsymbol{x}, 0)) > 1$. Particularly, when P, Q and Q' are multinomial probability distributions as before, then $\zeta = 2^m - 1$ and*

$$\frac{p_\zeta}{p_0} > \left(\frac{q_\zeta}{q_0}\right)^{h_I} \left(\frac{q'_\zeta}{q'_0}\right)^{h_{II}}. \tag{16}$$

Proof. Since if $p_\zeta q(\boldsymbol{x}, \zeta)/(p_0 q(\boldsymbol{x}, 0)) \leq 1$ holds for all ζ, then p_i^* converges to 0 or becomes ambiguous during the iterations, i.e., $p_0^* = p_i^*$ is one of the largest. The decoding algorithm won't work.

Particularly, when the probability values of P and Q (or Q') are in order as stated before, and all values of parity-checks are ζ, obviously we have

$$\frac{p_\zeta \boldsymbol{q}_\zeta^{\boldsymbol{x}}}{p_0 \boldsymbol{q}_0^{\boldsymbol{x}}} \leq \frac{p_\zeta q_\zeta^{h_I} q_\zeta^{h_{II}}}{p_0 q_0^{h_I} q_0^{h_{II}}}.$$

□

Table 4: Two probability distributions P and P'

x	0	1	$\dots$	2^m-2	2^m-1
$p_x - 2^{-m}$	$2^{-\frac{2m+\gamma}{2}}$	$-2^{-\frac{2m+\gamma}{2}}$	$\dots$	$2^{-\frac{2m+\gamma}{2}}$	$-2^{-\frac{2m+\gamma}{2}}$
$p'_x - 2^{-m}$	$2^{-\frac{2m+\gamma}{2}}$	ε	$\dots$	ε	ε

Remark 3. Though the ratio $\eta(\zeta, 0)$ has large value when all check values are ζ, The lower bound for h given in Proposition 1 may be loose, as the probability that all check values are ζ is small.

A lower bound for N could be derived through Proposition 1. For example, when generator polynomial $L(x) \in \mathbb{F}_{2^m}[x]$, the number of parity-checks h and the key stream length N shall satisfy that $\binom{N}{\tau}(2^m-1)^\tau \approx h2^k$.

As an application of Proposition 1, we consider an example of two special probability distributions. Since when $\Delta(\boldsymbol{e}) = 2^{-\gamma}$, it is expected that there are probability values around $2^{-m} \pm 2^{-\frac{2m+\gamma}{2}}$ in practice [YJM20], the distributions P and P' in Table 4 may be useful, where ε denotes $(2^{-\frac{2m+\gamma}{2}})/(2^m-1)$ [3].

When $2^{-\gamma/2}$ is relatively small to 1, by Taylor's formula, we have

$$\frac{p_{2i+1}}{p_0} =\approx 1 - 2^{\frac{-\gamma+2}{2}}, \frac{p'_i}{p'_0} =\approx 1 - \frac{2^m}{2^m-1}2^{-\frac{\gamma}{2}}.$$

Furthermore, by the convolution property, when each parity-check has $\tau+1, \tau \geq 2$ taps, we have

$$\frac{q_{2i+1}}{q_0} = \frac{1-2^{-\frac{\tau\gamma}{2}}}{1+2^{-\frac{\tau\gamma}{2}}} \approx 1 - 2^{\frac{-\tau\gamma+2}{2}}.$$

Hence, by Proposition 1, the number of type I parity-checks for P is

$$1 - 2^{\frac{-\gamma+2}{2}} \geq \left(1 - 2^{\frac{-\tau\gamma+2}{2}}\right)^{h_I} \Rightarrow h_I \geq 2^{\frac{(\tau-1)\gamma}{2}}. \tag{17}$$

For the case of P', the general term formula of distributions convolution could be deduced by its recursion formula, i.e.,

$$q'_0 = 2^{-m} + \frac{2^{m(\tau-1)}}{(2^m-1)^{\tau-1}}2^{-\frac{2m+\gamma}{2}\tau}, q'_i = 2^{-m} - \frac{2^{m(\tau-1)}}{(2^m-1)^\tau}2^{-\frac{2m+\gamma}{2}\tau}.$$

Thus we have

$$\frac{q'_i}{q'_0} \approx 1 - \frac{2^{m(\tau+1)}}{(2^m-1)^\tau}2^{-\frac{2m+\gamma}{2}\tau},$$

which means

$$1 - \frac{2^m}{2^m-1}2^{\frac{-\gamma}{2}} \geq \left(1 - \frac{2^{m(\tau+1)}}{(2^m-1)^\tau}2^{-\frac{2m+\gamma}{2}\tau}\right)^h \Rightarrow h \geq \left(\frac{2^m-1}{2^m}\right)^{\tau-1} 2^{\frac{(2m+\gamma)(\tau-1)}{2}}. \tag{18}$$

Notice that type I and II parity-checks are not distinguished in the case of P'.

The FCA mainly benefits from the increased SEI. More specifically, according to Theorem 1, there are 2^m-1 binary linear approximations contributing to the SEI of linear approximation with dimension m. Notice that there are other distributions, e.g., P'' with $p''_0 = 2^{-m} - 2^{-\frac{2m+\gamma}{2}}$, while the other value points are all the same. This case is similar with P' except that 0 is the minimal value point.

[3] Since the SEI of P' is less than $2^{-\gamma}$, the number of parity-checks h needed in practice may be smaller than that derived from P'.

4.2.2 A Bound Derived from the Practical Corrected Errors

In this part, we discuss how to deduce a bound from the number of expected positions with $p_\zeta^* > p_0^*, \zeta \neq 0$.

Let us consider the sets $\mathcal{A}(i), i \in \{1, 2, \ldots, 2^m - 1\}$ for multinomial distributions. Since $\mathcal{A}(i)$ may intersect with each other, the way of computing threshold in section 4.1.2 can't be directly applied. Thereby, we introduce some new sets: $\mathcal{A}'(i) = \mathcal{A}(i) - \mathcal{A}(i) \cap (\bigcup_{j=1}^{i-1} \mathcal{A}(i))$, That is $\mathcal{A}(i)$ excluding all elements that are included in previous sets $\mathcal{A}(i), i \in \{1, 2, \ldots, i\}$. Let M_i' denote the summation of probability values over set $\mathcal{A}'(i)$, more specifically,

$$\sum_{\zeta=1}^{2^m-1} M_\zeta' = \sum_{\zeta=1}^{2^m-1} p_\zeta \sum_{\boldsymbol{x} \in \mathcal{A}'(\zeta)} q(\boldsymbol{x}, \zeta). \tag{19}$$

It is reasonable to require that $\sum_{\zeta=1}^{2^m-1} M_\zeta' > 1$ after the first iteration. Then the succeeding iterations may trigger more positions with $p_\zeta^* > p_0^*$. This phenomenon may be the main advantage that soft decision decoding algorithms have.

Summing up the probability values in multinomial distributions is inconvenient. Though multivariate normal distribution approximation could also be used as before when h is large, the integral may not be easy to evaluate in practice, as the integral area $\mathcal{A}'(\zeta)$ is very complicated. Since symmetric distribution Q' simulates the iterative process very well, we could deduce boundaries for $\mathcal{A}'(\zeta)$ using $\mathrm{Multi}(h, \boldsymbol{q}')$. The following results show how to estimate M_ζ' in this case.

Proposition 2. *For multinomial probability distribution Multi*$(h, \boldsymbol{q}')$*, we have*

$$M_\zeta' = \sum_{l=h_b}^{h} \binom{h}{l} (1 - \sum_{i=0}^{\zeta} q_{i\oplus\zeta}')^{h-l} \sum_{(x_0,\ldots,x_\zeta) \in \mathcal{B}(\zeta)} \binom{l}{x_0, \ldots, x_\zeta} \prod_{i=0}^{\zeta} q'^{x_i}_{i\oplus\zeta}, 1 \le \zeta < 2^m,$$

where $\mathcal{B}(\zeta)$ *is constrained by* $\sum_{i=1}^{\zeta} x_i = l$*,* $x_\zeta - x_0 \ge h_b$ *and* $x_i - x_0 \le h_b, 1 \le i < \zeta$*.*

Particularly, when $\sum_{i=0}^{\zeta} q_{i\oplus\zeta}'$ *is small and* $hq_i' \le h_b$*, the expected number of positions with* $p_\zeta^* > p_0^*$ *in the first iteration are dominated by those small* l*.*

Proof. Since $q_1' = \cdots = q_{2^m-1}'$, we have

$$\begin{aligned} M_\zeta' &= \sum_{\boldsymbol{x} \in \mathcal{A}'(\zeta)} \binom{h}{\boldsymbol{x}} \boldsymbol{q}'^{\boldsymbol{x}}_\zeta \\ &= \sum_{l=h_b}^{h} \binom{h}{l} (1 - \sum_{i=0}^{\zeta} q_{i\oplus\zeta}')^{h-l} \sum_{(x_0,\ldots,x_\zeta) \in \mathcal{B}(\zeta)} \binom{l}{x_0, \ldots, x_\zeta} \prod_{i=1}^{\zeta} q'^{x_i}_{i\oplus\zeta}. \end{aligned}$$

By Proposition 1, we deduce that there is a minimal positive integer h_b such that $\delta(\zeta, 0) > 0$ when $x_\zeta - x_0 \le h_b$. Furthermore, $x_i - x_0 < h_b$ should holds for all $0 < i < \zeta$ to exclude the points in $\mathcal{A}'(i)$. Therefore, when $p_\zeta^* > p_0^*$, $(x_0, \ldots, x_\zeta) \in \mathcal{A}'(\zeta)$ must satisfy that

$$\begin{cases} x_i \ge 0, & 0 \le i \le \zeta, \\ x_i - x_0 < h_b, & 0 < i < \zeta, \\ x_\zeta - x_0 \ge h_b, & \\ x_0 + \cdots + x_\zeta < h & . \end{cases}$$

When h is not small and $\sum_{i=0}^{\zeta} q_{i\oplus\zeta}'$ is not high, multidimensional distribution $\mathrm{Multi}(h, \boldsymbol{q}_\zeta')$ could be approximated by $\zeta+1$ independent Poisson distributions with means $\lambda_{i\oplus\zeta} = hq_{i\oplus\zeta}'$, i.e.,

$$\Pr(X = \boldsymbol{x}) \approx \sum_{\mathcal{A}'(\zeta)} \prod_{i=0}^{\zeta} \frac{\lambda_{i\oplus\zeta}^{x_i}}{x_i!} e^{-\lambda_{i\oplus\zeta}} = \frac{\lambda_0^{x_\zeta}}{x_\zeta!} e^{-\lambda_0} \frac{\lambda_\zeta^{x_0+\cdots+x_{\zeta-1}}}{x_0! \ldots x_{\zeta-1}!} e^{-\zeta\lambda_\zeta}. \tag{20}$$

As $\lambda_i \leq h_b$, the maximal value of $\Pr(X = \boldsymbol{x})$ is when $\sum_{i=0}^{\zeta} x_i$ is small, i.e., when l is small. □

Proposition 2 gives us a hint that the value corresponding small l dominate M'_i. When ζ is not very large, M'_ζ could be approximated by partial summation for small l close to the boundary. Obviously, $M'_{i\neq 0}$ are monotone non-increasing sequence.

When $\zeta = 1$, there is another elegant way to estimate M'_1 by Skellam distribution. Let $Y_0 \sim \mathrm{Pois}(\lambda_1)$ and $Y_1 \sim \mathrm{Pois}(\lambda_0)$, we know that their difference $K = Y_1 - Y_0$ follows Skellam distribution with following probability density function.

$$p(k, \lambda_\zeta, \lambda_0) = e^{-\lambda_\zeta - \lambda_0} \left(\frac{\lambda_\zeta}{\lambda_0}\right)^{k/2} I_{|k|}(2\sqrt{\lambda_1 \lambda_0}),$$

where $I_{|k|}$ is the modified Bessel function of the first kind. Obviously, $M'_1 = Np_\zeta \Pr(K > h_b)$ since a boundary line is $x_1 \geq x_0 + h_b$ by Proposition 2.

5 Application to Grain-128a

In this section, we apply our new techniques to stream cipher Grain-128a. We assume the cryptanalysis is under the known-plaintext scenario. Since the output is directly used as key stream and the plaintext never participates in updating internal states, this assumption is reasonable for Grain-128a.

5.1 A Brief Description of Grain-128a

Grain-128a includes a 128-bit LFSR cascaded with a 128-bit NFSR. Let $s^{(t)} = (s_t, s_{t+1}, \ldots, s_{t+127})$ and $b^{(t)} = (b_t, b_{t+1}, \ldots, b_{t+127})$ denote their internal states at time t. The output y_t of the pre-output function at time t is represented by

$$y_t = h(s^{(t)}, b^{(t)}) \oplus s_{t+93} \oplus b_{t+2} \oplus b_{t+15} \oplus b_{t+36} \oplus b_{t+45} \oplus b_{t+64} \oplus b_{t+73} \oplus b_{t+89},$$

where $h(s^{(t)}, b^{(t)})$ is defined as

$$\begin{aligned} h(s^{(t)}, b^{(t)}) =& h(b_{t+12}, s_{t+8}, s_{t+13}, s_{t+20}, b_{t+95}, s_{t+42}, s_{t+60}, s_{t+79}, s_{t+94}) \\ =& b_{t+12}s_{t+8} \oplus s_{t+13}s_{t+20} \oplus b_{t+95}s_{t+42} \oplus s_{t+40}s_{t+79} \oplus b_{t+12}b_{t+95}s_{t+94}. \end{aligned}$$

The feedback bits of LFSR and NFSR are computed by

$$\begin{aligned} s_{t+128} =& s_t \oplus s_{t+7} \oplus s_{t+38} \oplus s_{t+70} \oplus s_{t+81} \oplus s_{t+96}, \\ b_{t+128} =& s_t \oplus b_t \oplus b_{t+26} \oplus b_{t+56} \oplus b_{t+91} \oplus b_{t+96} \oplus \\ & b_{t+3}b_{t+67} \oplus b_{t+11}b_{t+13} \oplus b_{t+17}b_{t+18} \oplus b_{t+27}b_{t+59} \oplus \\ & b_{t+40}b_{t+48} \oplus b_{t+61}b_{t+65} \oplus b_{t+68}b_{t+84} \oplus \\ & b_{t+22}b_{t+24}b_{t+25} \oplus b_{t+70}b_{t+78}b_{t+82} \oplus b_{t+88}b_{t+92}b_{t+93}b_{t+95}. \end{aligned}$$

Key stream bit $z_t = y_t$ in the stream cipher mode, while $z_t = y_{2w+2t}$ in the authenticated mode, where w is the tag size. The overall structure of Grain-128a is depicted in Fig. 5.

5.2 Constructing Multidimensional Linear Approximations and Checking Parity

In [TIM+18], the authors proposed a family of linear approximations of Grain-128a by pilling up different clocks to eliminate the linear terms of the NFSR, which forms are

$$\begin{aligned} \oplus_{i\in\mathbb{T}_z} y_{t+i} \approx& \oplus_{i\in\mathbb{T}_z} s_{t+i+93} \oplus \oplus_{j\in\mathbb{A}} s_{t+j} \oplus_{i\in\mathbb{T}_z} \langle \Lambda_i[1-3], (s_{t+i+8}, s_{t+i+13}, s_{t+i+20}) \rangle \\ & \oplus \langle \Lambda_i[5-8], (s_{t+i+42}, s_{t+i+60}, s_{t+i+79}, s_{t+i+94}) \rangle, \end{aligned} \tag{21}$$

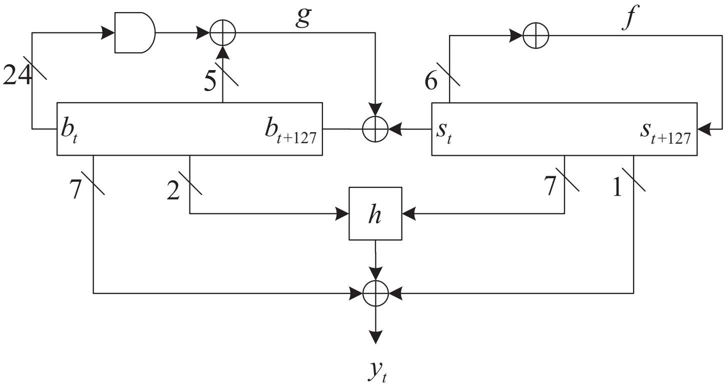

Figure 5: The overall schematic of Grain-128a

where $\mathbb{A} = \{2, 15, 36, 45, 64, 73, 89\}, \mathbb{T}_z = \{0, 26, 56, 91, 96, 128\}$, Λ_i is a 9-bit binary linear mask, two bits $\Lambda_i[0, 4]$ are fixed.

According to [TIM+18], an assignment of $\Lambda_i[1-3]$ and $\Lambda_i[5-8]$ will completely determine the correlation of h function, when $\Lambda_i[0, 4]$ is fixed. For a specific $i \in \mathbb{T}_z$, there are only 64 possible $\Lambda_i[0, 4], i \in \mathbb{A}$ such that the correlation of Eq. (21) is nonzero. Hence, the linear correlation value of (21) can be deduced by summing up all these 64 $\Lambda_i[0, 4], i \in \mathbb{T}_z$. Meanwhile, there are 2^6 values of $\Lambda_i[1-3, 5-8]$ of a specific $i \in \mathbb{T}_z$ with the correlation of h function is nonzero. For example, when $\Lambda_i[1-3, 5-8] = 0000000, \forall i \in \mathbb{T}_z$, the correlation of (21) is about $\pm 2^{-57.0454}$. For more details of these linear approximations, we refer to [TIM+18].

In this paper, we reuse these linear approximations but in a new way by bundling them up. Firstly, we choose 42 linear approximations which $\Lambda_i[1-3, 5-8], i \in \mathbb{T}_z$ has form

$$(\Lambda_0[1-3, 5-8], \Lambda_{26}[1-3, 5-8], \ldots, \Lambda_{128}[1-3, 5-8]) = (0, \ldots, 0, 1, 0, \ldots, 0),$$

i.e., $\Lambda_i[1-3, 5-8], i \in \mathbb{T}_z$ as a group of standard basis. Then a linear approximation with dimension $9 \leq m \leq 42$ can be established as follows

$$E(\boldsymbol{x}_t + \boldsymbol{u}_t) + E\boldsymbol{y}_t = \boldsymbol{e}_t, \tag{22}$$

where E is a $m \times m$ identity matrix in $\mathbb{F}_2$. $\boldsymbol{e}_t$ is noise vector, and

$$\begin{aligned}
\boldsymbol{x}_t &= (\ldots, s_{t+i+8}, s_{t+i+13}, s_{t+i+20}, s_{t+i+42}, s_{t+i+60}, s_{t+i+79}, s_{t+i+94}, \ldots), \\
\boldsymbol{u}_t &= \left(\sum_{i \in \mathbb{A} \bigcup \mathbb{T}_z} s_{t+i}, \sum_{i \in \mathbb{A} \bigcup \mathbb{T}_z} s_{t+i}, \ldots, \sum_{i \in \mathbb{A} \bigcup \mathbb{T}_z} s_{t+i} \right), \\
\boldsymbol{y}_t &= \left(\sum_{i \in \mathbb{T}_z} y_{t+i}, \sum_{i \in \mathbb{T}_z} y_{t+i}, \ldots, \sum_{i \in \mathbb{T}_z} y_{t+i} \right), \\
\boldsymbol{e}_t &= (e_t, e_{t+1}, \ldots, e_{t+m-1}).
\end{aligned}$$

Any even Hamming weight linear combination of Eq. (22) will generate a linear approximation without $\sum_{i \in \mathbb{A} \bigcup \mathbb{T}_z} s_{t+i}$ and $\sum_{i \in \mathbb{T}_z} y_{t+i}$, which correlation would be treated as 0. As for odd linear combinations, it is still required that any of $\Lambda_i[1-3, 5-8], i \in \mathbb{T}_z$ will not deduce a zero correlation for h function. Therefore, we can construct a multidimensional linear approximation with dimension $9 \leq m \leq 42$, which consists of $2^{m-1-6} = 2^{m-7}$ linear approximations with correlation $\pm 2^{-57.0454}$. By Theorem 1, its SEI $\Delta(e_t) = 2^{m-121.0908}$.

As s_t is a m-sequence, shifting and summation sequence $s'_{t+c'_j} = s_{t+c_j} + \sum_{i\in\mathbb{A}\bigcup\mathbb{T}_z} s_{t+i}$ is also a a m-sequence with same generator polynomial as s_t. Let vectorial sequence $\boldsymbol{x}'_t = (s'_{t+c'_1}, \ldots, s'_{t+c'_m})$, since shift offsets $c'_j, 1 \le j \le m$ have large difference, the parity-checks with $\tau = 1$ are not all ruled out.

Since $\boldsymbol{x}'_t$ runs over $\mathbb{F}_2^m \backslash \{\boldsymbol{0}\}$, there is at most one parity-check with $\tau = 1$ for each $0 < n \le N/m$. In order to increase the occurrence possibility for parity-check with $t = 1$, several redundant binary linear approximations with nonzero correlation could be added into the subspace. The dimension increases but SEI is almost unchanged. Therefore, the maximal probability value should decrease.

Another way is exploiting a kind of special parity-checks with $\tau > 1$. In order to avoid the great loss of SEI while implementing convolution, we play a trade-off trick when special parity-checks are feasible. For example, suppose we have h special parity-checks as follows.

$$G_{n,1}\boldsymbol{x}'_{t-d_{n,1}} + \sum_{i=1}^{a} G_{n-i,1}\boldsymbol{x}'_{t-d_i} + E\boldsymbol{x}'_t = 0, \ldots, G_{n,h}\boldsymbol{x}'_{t-d_{n,h}} + \sum_{i=1}^{a} G_{n-i,h}\boldsymbol{x}'_{t-d_i} + E\boldsymbol{x}'_t = 0.$$

Notice that all of them involve vector variables $\boldsymbol{x}'_t, \boldsymbol{x}'_{t-d_1}, \ldots, \boldsymbol{x}'_{t-d_a}$ except for the last variable $\boldsymbol{x}'_{t-d_{n,j}}$ Let $D_{n-i,j} = G_{n-i,j} + G_{n-i,1}$, $1 \le i \le a$, denote the coefficient difference between the j-th and the 1-st equation. Let $\sum_{i=1}^{a} D_{n-i,j}\boldsymbol{x}'_{t-d_i} = \boldsymbol{\delta}_j$ denote the difference value. Moreover, we require that $\boldsymbol{\delta}_j$ satisfies some restrictions.

Since we have $h-1$ groups of linear equations with coefficients $(D_{n-1,j}, \ldots, D_{n-a,j})$, we require that those linear equation groups have the same solution subspace S with large dimension, for example, with dimension $am-1$ or $am-2$, which implies that the rank of $(D_{n-1,j}, \ldots, D_{n-a,j})$ may be 1 or 2. Thus when $(\boldsymbol{x}'_t, \boldsymbol{x}'_{t-d_1}, \ldots, \boldsymbol{x}'_{t-d_a}) \in S$, all $\boldsymbol{\delta}_j = \boldsymbol{0}$. Otherwise, $\boldsymbol{\delta}_j \ne \boldsymbol{0}$ are likely different. Thus we have

$$\begin{aligned}
G_{n,1}\boldsymbol{x}'_{t-d_{n,1}} + \boldsymbol{0} + \sum_{i=1}^{a} G_{n-i,1}\boldsymbol{x}'_{t-d_i} + E\boldsymbol{x}'_t = 0,\\
G_{n,2}\boldsymbol{x}'_{t-d_{n,2}} + \boldsymbol{\delta}_2 + \sum_{i=1}^{a} G_{n-i,1}\boldsymbol{x}'_{t-d_i} + E\boldsymbol{x}'_t = 0,\\
\ldots,\\
G_{n,r}\boldsymbol{x}'_{t-d_{n,h}} + \boldsymbol{\delta}_h + \sum_{i=1}^{a} G_{n-i,1}\boldsymbol{x}'_{t-d_i} + E\boldsymbol{x}'_t = 0.
\end{aligned}$$

Then the APP for $\sum_{i=1}^{a} G_{n-i,1}\boldsymbol{e}_{t-d_i} + E\boldsymbol{e}_t$ could be evaluated by total probability theorem according to whether all of $\boldsymbol{\delta}_j$ are $\boldsymbol{0}$. The initial state is recovered from observed values $\boldsymbol{z}_t$ of the error-corrected positions, i.e.,

$$\sum_{i=1}^{a} G_{n-i,1}(\boldsymbol{x}'_{t-d_i} + \boldsymbol{e}_{t-d_i}) + E(\boldsymbol{x}'_t + \boldsymbol{e}_t) = \sum_{i=1}^{a} G_{n-i,1}\boldsymbol{x}'_{t-d_i} + E\boldsymbol{x}'_t = \boldsymbol{z}_t.$$

The dimension of linear approximation is not changed but the APP converges slower. Thus the decoding ability decreases when dimension of S decreasing. However, the constraints for parity-checks is relaxed.

With these techniques, a fast correlation attack could be performed with these special parity-checks and multidimensional linear approximations in (22).

5.3 Complexity Estimation

In this section, we estimate some theoretical bounds for Grain-128a, which would bring us a new perspective on its security margin.

Let the SEI $\Delta(\boldsymbol{e}_t) = 2^{-\gamma}$, dimension $m = 42$. According to results in [YJM20], we can assume $p_0 = 2^{-m} + 2^{-\frac{2m+\gamma}{2}}$ be the maximal probability value. This implies that all other points are very close to 2^{-m}. Thereby, the probability distribution P stemming from SEI is close to symmetric distribution. To simplify the process of estimating the expected number of positions with $p^*_\zeta > p^*_0$, we need the following hypothesis.

Hypothesis 1. *There are at least 2 parity-checks with two taps, or there are more special parity-checks as stated in the previous section.*

Suppose we have h special parity-checks corresponding to a solution subspace of dimension $am-1$ as stated above. Let $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_h$ denote the check values, $\boldsymbol{\gamma} = (\gamma_0, \ldots, \gamma_{2^m-1})$ and $\boldsymbol{\gamma}' = (\gamma'_0, \ldots, \gamma'_{2^m-1})$ denote the frequency of values in $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_h$ and $\boldsymbol{v}_1, \boldsymbol{v}_2 \oplus \boldsymbol{\delta}_2, \ldots, \boldsymbol{v}_h \oplus \boldsymbol{\delta}_h$ respectively. There are two events that may deduce $p^*_\zeta > p^*_0$: event A denotes that $\boldsymbol{\gamma} \in \mathcal{A}'(\zeta)$, while event B denotes that $\boldsymbol{\gamma}' \in \mathcal{A}'(\zeta)$. For simplicity, we only consider that when A occurs, then we have

$$\begin{aligned} M'_\zeta =& \frac{1}{2} p_\zeta \left(\sum_{\boldsymbol{\gamma} \in \mathcal{A}'(\zeta)} \binom{h}{\boldsymbol{\gamma}} \prod_i p_{i \oplus \zeta}^{\gamma_i} + \sum_{\boldsymbol{\gamma} \in \mathcal{A}'(\zeta)} \binom{h}{\boldsymbol{\gamma}'} \prod_i p_{i \oplus \zeta}^{\gamma'_i} \right), \\ M'_0 =& \frac{1}{2} p_0 \left(\sum_{\boldsymbol{\gamma} \in \mathcal{A}'(\zeta)} \binom{h}{\boldsymbol{\gamma}} \prod_i p_i^{\gamma_i} + \sum_{\boldsymbol{\gamma} \in \mathcal{A}'(\zeta)} \binom{h}{\boldsymbol{\gamma}'} \prod_i p_i^{\gamma'_i} \right). \end{aligned}$$

The first term denotes the probability that current noise symbol is ζ or 0, when the frequency vector $\boldsymbol{\gamma} \in \mathcal{A}'(\zeta)$ and all $\boldsymbol{\delta}_j = \mathbf{0}$. The second term corresponds to when the frequency vector $\boldsymbol{\gamma} \in \mathcal{A}'(\zeta)$ but many $\boldsymbol{\delta}_j \neq \mathbf{0}$, $2 \leq j \leq h$. Thus the observed vector is $\boldsymbol{\gamma}'$. Since γ'_i are likely different, it is reasonable to assume that the second terms of M'_ζ and M'_0 are close. To simplify the evaluation, we only consider the first term.

Table 5 in Appendix A depicts the approximation of M'_ζ($\frac{1}{2}$ is neglected). M'_1 is estimated by two methods: Skellam distribution and summation for small l. The two estimations are very close to each other. Let $D'_i = M'_i - M'_0$ denote the difference. We also compute the summation $\sum_{i=1}^{2^{36}} M'_i$ and the difference summation $\sum_{i=1}^{2^{36}} D'_i$. For example, when $h = h_b = 2$, the expected key stream length $N > 2^{48+42+1} = 2^{91}$. As P is symmetric, it seems there is no need to evaluate every probability value of APP distribution. Therefore, we use the key stream length N multiplied by the number of parity-checks h as time complexity. For the other case when there are at least 2 parity-checks with two taps, there is no probability loss caused by the trade-off. The complexity estimation is similar.

6 Discussion and Open Problems

The analysis of the vectorial iterative decoding algorithm is very complicated, there are still several open problems needed further study.

Firstly, it seems we cannot directly compare the vectorial decoding algorithm with a binary algorithm. The main reason is that we do not know how to put them in the same start line. For example, how many parity-checks means they have equal status. Although we point out the potential theoretical advantages for some cases in Section 3.4, it is still an open problem for the generic case.

Secondly, the other theoretical properties of the vectorial algorithm are still not clear. For example, the time complexity is estimated by the key stream length multiplied by the number of parity-checks. There are lots of redundant computations. However, we have no idea whether FWHT acceleration technique could be applied in this case.

Thirdly, the main difficulties are figuring out the existence of the special parity-checks and proposing an efficient algorithm to generate suitable parity-checks in matrix rings instead of finite fields. In this paper, on one hand, we don't know whether Hypothesis 1 is realistic. Thus the estimation for M_i' and D_i' of Grain-128a is not very concrete. In Appendix C, we attempt to discuss some necessary conditions for the existence of special parity-checks by observations. However, we don't know the results for the generic case as it seems to be a difficult problem. Perhaps it is related to classifying all the sequences generated by the LFSRs over a matrix ring, which needs a lot of future work. On the other hand, we didn't study how to generate parity-checks in a matrix ring. Thus the complexity of the precomputation phase is skipped over. We leave them as open problems.

7 Conclusion

In this paper, a vectorial iterative decoding algorithm for FCA is proposed. Two novel criteria are given to break the tie and improve the decoding efficiency. The original binary FCA proposed by Meier and Staffelbach is a special case of our FCA with dimension 1. We describe some cryptographic properties of its statistical model, decoding efficiency, etc. Based on the statistical property of the first iteration, we estimate the number of needed parity-checks and the bound of expected key stream length from the perspective of iterative decoding. We also perform a scaled experiment to verify the validity of the vectorial iterative decoding algorithm and the generality of the iterative criterion.

Moreover, we apply it to stream cipher Grain-128a. We construct a multidimensional linear approximation with large SEI by bundling up those binary linear approximations proposed in CRYPTO 18. We also give a trade-off approach to use special parity-checks with more than 2 taps. Consequently, based on a hypothesis, we give an estimation of the potential security margin of Grain-128a from the point view of vectorial probabilistic iterative decoding.

References

[29115] ISO/IEC: JTC1: ISO/IEC 29167-13. Information technology - Automatic identification and data capture techniques – Part 13: Crypto suite Grain-128A security services for air interface communications. 2015.

[ÅHJM11] Martin Ågren, Martin Hell, Thomas Johansson, and Willi Meier. Grain-128a: A new version of grain-128 with optional authentication. *Int. J. Wirel. Mob. Comput.*, 5(1):48–59, 2011.

[AHMN13] Jean-Philippe Aumasson, Luca Henzen, Willi Meier, and María Naya-Plasencia. Quark: A lightweight hash. *J. Cryptol.*, 26(2):313–339, 2013.

[ÅLHJ12] Martin Ågren, Carl Löndahl, Martin Hell, and Thomas Johansson. A survey on fast correlation attacks. *Cryptogr. Commun.*, 4(3-4):173–202, 2012.

[AM15] Frederik Armknecht and Vasily Mikhalev. On lightweight stream ciphers with shorter internal states. In Gregor Leander, editor, *Fast Software Encryption - 22nd International Workshop, FSE 2015, Istanbul, Turkey, March 8-11, 2015, Revised Selected Papers*, volume 9054 of *Lecture Notes in Computer Science*, pages 451–470. Springer, 2015.

[BJV04] Thomas Baignères, Pascal Junod, and Serge Vaudenay. How far can we go beyond linear cryptanalysis? In Pil Joong Lee, editor, *Advances in Cryptology - ASIACRYPT 2004*, pages 432–450, Berlin, Heidelberg, 2004. Springer Berlin Heidelberg.

[CDF$^+$02] A. Clark, Ed Dawson, J. Fuller, J. Golić, H. J. Lee, William Millan, S. J. Moon, and L. Simpson. The lili-ii keystream generator. In Lynn Batten and Jennifer Seberry, editors, *Information Security and Privacy*, pages 25–39, Berlin, Heidelberg, 2002. Springer Berlin Heidelberg.

[CGD96] Andrew J. Clark, Jovan Dj. Golic, and Ed Dawson. A comparison of fast correlation attacks. In Dieter Gollmann, editor, *Fast Software Encryption, Third International Workshop, Cambridge, UK, February 21-23, 1996, Proceedings*, volume 1039 of *Lecture Notes in Computer Science*, pages 145–157. Springer, 1996.

[CJM02] Philippe Chose, Antoine Joux, and Michel Mitton. Fast correlation attacks: An algorithmic point of view. In Lars R. Knudsen, editor, *Advances in Cryptology — EUROCRYPT 2002*, pages 209–221, Berlin, Heidelberg, 2002. Springer Berlin Heidelberg.

[CJS00] Vladimir V. Chepyzhov, Thomas Johansson, and Ben J. M. Smeets. A simple algorithm for fast correlation attacks on stream ciphers. In Bruce Schneier, editor, *Fast Software Encryption, 7th International Workshop, FSE 2000, New York, NY, USA, April 10-12, 2000, Proceedings*, volume 1978 of *Lecture Notes in Computer Science*, pages 181–195. Springer, 2000.

[CS91] Vladimir V. Chepyzhov and Ben J. M. Smeets. On a fast correlation attack on certain stream ciphers. In Donald W. Davies, editor, *Advances in Cryptology - EUROCRYPT '91, Workshop on the Theory and Application of of Cryptographic Techniques, Brighton, UK, April 8-11, 1991, Proceedings*, volume 547 of *Lecture Notes in Computer Science*, pages 176–185. Springer, 1991.

[CT00] Anne Canteaut and Michaël Trabbia. Improved fast correlation attacks using parity-check equations of weight 4 and 5. In Bart Preneel, editor, *Advances in Cryptology - EUROCRYPT 2000, International Conference on the Theory and Application of Cryptographic Techniques, Bruges, Belgium, May 14-18, 2000, Proceeding*, volume 1807 of *Lecture Notes in Computer Science*, pages 573–588. Springer, 2000.

[EJ00] Patrik Ekdahl and Thomas Johansson. SNOW-A new stream cipher. In *Proceedings of first open NESSIE workshop, KU-Leuven*, pages 167–168, 2000.

[EJ03] Patrik Ekdahl and Thomas Johansson. A new version of the stream cipher SNOW. In Kaisa Nyberg and Howard Heys, editors, *Selected Areas in Cryptography*, pages 47–61, Berlin, Heidelberg, 2003. Springer Berlin Heidelberg.

[EJMY19] Patrik Ekdahl, Thomas Johansson, Alexander Maximov, and Jing Yang. A new SNOW stream cipher called SNOW-V. *IACR Trans. Symmetric Cryptol.*, 2019(3):1–42, 2019.

[Ger61] Murray Gerstenhaber. On the number of nilpotent matrices with coefficients in a finite field. *Illinois Journal of Mathematics*, 5(2):330 – 333, 1961.

[GH05] Jovan Dj. Golic and Philip Hawkes. Vectorial approach to fast correlation attacks. *Des. Codes Cryptogr.*, 35(1):5–19, 2005.

[Gol01] Jovan Dj. Golic. Iterative optimum symbol-by-symbol decoding and fast correlation attacks. *IEEE Trans. Inf. Theory*, 47(7):3040–3049, 2001.

[GX94] Guang Gong and Guozheng Xiao. Synthesis and uniqueness of m-sequences over GF(q^{n}) as n-phase sequences over GF(q). *IEEE Trans. Commun.*, 42(8):2501–2505, 1994.

[HJM07] Martin Hell, Thomas Johansson, and Willi Meier. Grain: A stream cipher for constrained environments. *Int. J. Wirel. Mob. Comput.*, 2(1):86–93, 2007.

[HJMM06] Martin Hell, Thomas Johansson, Alexander Maximov, and Willi Meier. A stream cipher proposal: Grain-128. In *Proceedings 2006 IEEE International Symposium on Information Theory, ISIT 2006, The Westin Seattle, Seattle, Washington, USA, July 9-14, 2006*, pages 1614–1618. IEEE, 2006.

[JJ99a] Thomas Johansson and Fredrik Jönsson. Fast correlation attacks based on turbo code techniques. In Michael J. Wiener, editor, *Advances in Cryptology - CRYPTO '99, 19th Annual International Cryptology Conference, Santa Barbara, California, USA, August 15-19, 1999, Proceedings*, volume 1666 of *Lecture Notes in Computer Science*, pages 181–197. Springer, 1999.

[JJ99b] Thomas Johansson and Fredrik Jönsson. Improved fast correlation attacks on stream ciphers via convolutional codes. In Jacques Stern, editor, *Advances in Cryptology - EUROCRYPT '99, International Conference on the Theory and Application of Cryptographic Techniques, Prague, Czech Republic, May 2-6, 1999, Proceeding*, volume 1592 of *Lecture Notes in Computer Science*, pages 347–362. Springer, 1999.

[JJ00] Thomas Johansson and Fredrik Jönsson. Fast correlation attacks through reconstruction of linear polynomials. In Mihir Bellare, editor, *Advances in Cryptology — CRYPTO 2000*, pages 300–315, Berlin, Heidelberg, 2000. Springer Berlin Heidelberg.

[LV04] Yi Lu and Serge Vaudenay. Faster correlation attack on bluetooth keystream generator E0. In Matthew K. Franklin, editor, *Advances in Cryptology - CRYPTO 2004, 24th Annual International CryptologyConference, Santa Barbara, California, USA, August 15-19, 2004, Proceedings*, volume 3152 of *Lecture Notes in Computer Science*, pages 407–425. Springer, 2004.

[MAM16] Vasily Mikhalev, Frederik Armknecht, and Christian Müller. On ciphers that continuously access the non-volatile key. *IACR Trans. Symmetric Cryptol.*, 2016(2):52–79, 2016.

[MFI02] Miodrag J. Mihaljevi, Marc P. C. Fossorier, and Hideki Imai. Fast correlation attack algorithm with list decoding and an application. In Mitsuru Matsui, editor, *Fast Software Encryption*, pages 196–210, Berlin, Heidelberg, 2002. Springer Berlin Heidelberg.

[MG91] Miodrag J. Mihaljevic and Jovan Dj. Golic. A comparison of cryptanalytic principles based on iterative error-correction. In Donald W. Davies, editor, *Advances in Cryptology - EUROCRYPT '91, Workshop on the Theory and Application of of Cryptographic Techniques, Brighton, UK, April 8-11, 1991, Proceedings*, volume 547 of *Lecture Notes in Computer Science*, pages 527–531. Springer, 1991.

[MG93] Miodrag J. Mihaljević and Jovan Dj. Golić. Convergence of a bayesian iterative error-correction procedure on a noisy shift register sequence. In Rainer A. Rueppel, editor, *Advances in Cryptology — EUROCRYPT' 92*, pages 124–137, Berlin, Heidelberg, 1993. Springer Berlin Heidelberg.

[MS89] Willi Meier and Othmar Staffelbach. Fast correlation attacks on certain stream ciphers. *J. Cryptol.*, 1(3):159–176, 1989.

[NS15] Ivica Nikolić and Yu Sasaki. Refinements of the k-tree algorithm for the generalized birthday problem. In Tetsu Iwata and Jung Hee Cheon, editors, *Advances in Cryptology – ASIACRYPT 2015*, pages 683–703, Berlin, Heidelberg, 2015. Springer Berlin Heidelberg.

[Sha48] Claude Elwood Shannon. A mathematical theory of communication. *The Bell System Technical Journal*, 27:379 – 423, 623 – 656, 1948.

[TIM+18] Yosuke Todo, Takanori Isobe, Willi Meier, Kazumaro Aoki, and Bin Zhang. Fast correlation attack revisited. In Hovav Shacham and Alexandra Boldyreva, editors, *Advances in Cryptology – CRYPTO 2018*, pages 129–159, Cham, 2018. Springer International Publishing.

[UEA06] IA UEA2&UIA. Specification of the 3gpp confidentiality and integrity algorithms uea2& uia2. document 2: Snow 3g specifications. version: 1.1. etsi, 2006.

[WLLM19] Shichang Wang, Meicheng Liu, Dongdai Lin, and Li Ma. Fast correlation attacks on Grain-like small state stream ciphers and cryptanalysis of plantlet, fruit-v2 and fruit-80. Cryptology ePrint Archive, Report 2019/763, 2019.

[YJM20] Jing Yang, Thomas Johansson, and Alexander Maximov. Spectral analysis of ZUC-256. *IACR Trans. Symmetric Cryptol.*, 2020(1):266–288, 2020.

[ZXM15] Bin Zhang, Chao Xu, and Willi Meier. Fast correlation attacks over extension fields, large-unit linear approximation and cryptanalysis of snow 2.0. In Rosario Gennaro and Matthew Robshaw, editors, *Advances in Cryptology – CRYPTO 2015*, pages 643–662, Berlin, Heidelberg, 2015. Springer Berlin Heidelberg.

[ZYR91] Kencheng Zeng, C. H. Yang, and T. R. N. Rao. An improved linear syndrome algorithm in cryptanalysis with applications. In Alfred J. Menezes and Scott A. Vanstone, editors, *Advances in Cryptology-CRYPTO' 90*, pages 34–47, Berlin, Heidelberg, 1991. Springer Berlin Heidelberg.

A Estimation of M_i'

Table 5 depicts some estimatied values about the data complexity.

Table 5: Estimation of some M_i' with $m = 42$

$\log_2(h)$	$\log_2(D_1)$	$\log_2(M_1')$		$\log_2(\sum_{i=1}^{2^{36}} M_i')$	$\log_2(\sum_{i=1}^{2^{36}} D_i')$
		`summation`	`Skellam`		
1	-122.5454	-84.0004	-83.0000	-47.9999	-86.5435
2	-119.9605	-81.4150	-81.0000	-45.4151	-83.9722
3	-117.7381	-79.1926	-79.0000	-43.1943	-81.7714
4	-115.6385	-77.0931	-77.0000	-41.1209	-79.7206
5	-113.5912	-75.0458	-75.0000	-39.0876	-77.7676
6	-111.5682	-73.0227	-73.0000	-37.1719	-75.9413
7	-109.5567	-71.0113	-71.0000	-35.4574	-74.3296
8	-107.5511	-69.0056	-69.0000	-34.0809	-73.0173
9	-105.5483	-67.0028	-67.0000	-33.0023	-71.9597
10	-103.5469	-65.0014	-65.0000	-32.0000	-70.9649

B Some Information Theory Properties

In this subsection, we discuss some properties from the point view of information theory. Suppose the noises are independent and the parity-checks are linearly independent, the relative entropy between Multi$(h, \boldsymbol{q}_0)$ with density function $q(\boldsymbol{x})$ and Multi$(h, (2^{-m}, \ldots, 2^{-m}))$ with density function $u(\boldsymbol{x})$ is

$$D(q \parallel u) = H(q, u) - H(q) = h \sum_{i=0}^{2^m-1} q_i \log \frac{q_i}{2^{-m}} = h(m - H(\boldsymbol{q}_0)). \tag{23}$$

That is the relative entropy is the number of parity-checks times the SEI of probability distribution Q.

Secondly, we hope that the right corrected positions are as many as possible in the complement process. Now we think about the sum of relative entropy between Multi$(h, \boldsymbol{q}_c)$ and Multi$(h, \boldsymbol{q}_0)$ for all $c \neq 0$, and we have

Proposition 3. *Let $q_c(\boldsymbol{x})$ and $q_0(\boldsymbol{x})$ be density functions of Multi$(h, \boldsymbol{q}_c)$ and Multi$(h, \boldsymbol{q}_c)$ respectively, then*

$$\sum_{c \neq 0} D(q_c(\boldsymbol{x}) \parallel q_0(\boldsymbol{x})) = -h \log \prod_{i=0}^{2^m-1} q_i - h2^m H(\boldsymbol{q}_0).$$

Proof.

$$\begin{aligned}\sum_{c \neq 0} D(q_c(\boldsymbol{x}) \parallel q_0(\boldsymbol{x})) =& \sum_{c \neq 0} h(\sum_{i=0}^{2^m-1} q_{i \oplus c} \log q_{i \oplus c} - \sum_{i=0}^{2^m-1} q_{i \oplus c} \log q_i) \\ =& - h(2^m - 1) H(\boldsymbol{q}_0) - h \sum_{i=0}^{2^m-1} \sum_{c \neq 0} q_{i \oplus c} \log q_i \\ =& - h(2^m - 1) H(\boldsymbol{q}_0) - h \sum_{i=0}^{2^m-1} (1 - q_i) \log q_i \\ =& - h \log \prod_{i=0}^{2^m-1} q_i - h2^m H(\boldsymbol{q}_0).\end{aligned}$$

□

This tells us when the probability distribution of noises approaches uniform distribution, the total relative entropy converges to 0.

C A Discussion for Sparse Parity-checks

Since sparse parity-checks have large advantages while checking parity, we are interested in these parity-checks with 2 or 3 taps, and those special parity-checks stated in Section 5.2. In this section, we give some miscellaneous observations about them.

Let $\boldsymbol{x}_t = (x_{t+c_1}, x_{t+c_2}, \ldots, x_{t+c_m}), c_1 < \ldots < c_m$ denotes the output at time t of LFSR with generator polynomial $L(x) \in M_m(\mathbb{F}_2)[x]$. Each coordinate sequence is a m-sequence $x_1 x_2 \ldots$ left shifting c_i times, and its minimal polynomial $f(x) \in \mathbb{F}_2[x]$ has degree k. Particularly, when shift vector $(c_1, c_2, \ldots, c_m)$ satisfies special condition, it becomes a LFSR over extension field $\mathbb{F}_{2^m}$ [GX94]. Though parity-checks with 2 taps have very large advantages, unfortunately, the following direct observations imply that they must satisfy some necessary conditions.

Proposition 4. *Let $\boldsymbol{x}_t = (x_{t+c_1}, x_{t+c_2}, \ldots, x_{t+c_m})$ be as stated above, we have*

- *If $c_m - c_1 + m - 1 < k$, then there is no parity-check with $\tau = 1$.*
- *Given two parity-checks with $\tau = 1$, $G\boldsymbol{x}_t + E\boldsymbol{x}_{t+d_1} = 0$, $G'\boldsymbol{x}_t + E\boldsymbol{x}_{t+d_2} = 0$, if $d_1 = d_2$ and $\boldsymbol{x_t}$ run over all values in $\mathbb{F}_2^m \backslash \{\boldsymbol{0}\}$, then $G = G'$. If $d_1 \neq d_2$, then $\gcd(d_1, d_2) > k - m$.*
- *Assume that the sequence $\boldsymbol{x}$ is periodic, if there are two special parity-checks as stated in Section 5.2, let $G\boldsymbol{x}_t + \ldots + G'\boldsymbol{x}_{t+d} = 0$ denote their sum, then both the head G and the tail G' must be invertible.*

Proof. 1. Let $G\boldsymbol{x}_t + E\boldsymbol{x}_{t+d} = 0$ be a parity-check. Since i-th row of A and E forms a check polynomial f_i with nonzero constant for x_t, then $f|f_i$. As G is nonsingular, there must be two different check polynomials $f_i(x)$ and $f_j(x)$. That means $f_i + f_j$ also forms a check polynomial, but $c_m - c_1 + m - 1 < k$ means a polynomial with degree less than k could be deduced, which is impossible.

2. When $d_1 = d_2$, it is deduced that $(G + G')\boldsymbol{x}_t = 0$ for all $\boldsymbol{x}_t$, When $\boldsymbol{x_t}$ run over all values in $\mathbb{F}_2^m \backslash \{\boldsymbol{0}\}$, then we have $G = G'$.

When $d_1 < d_2$, we could deduce another linearly dependent parity-check

$$G'(G^{-1}\boldsymbol{x}_t + E\boldsymbol{x}_{t+d_2-d_1}) = 0.$$

Therefore, according to Euclid long division algorithm, there is a G^* which satisfies

$$G^*\boldsymbol{x}_t + E\boldsymbol{x}_{t+gcd(d_1,d_2)} = 0.$$

Since there are k information bit of LFSR, then $gcd(d_1, d_2) \geq k - m$.

3. As the sum of two different parity-checks still satisfies the sequence $\boldsymbol{x}$, and $\boldsymbol{x}$ is periodic, then we must have both G and G' are invertible. □

These observations imply that parity-checks with 2 taps may be rare, but it doesn't mean none, even though the key stream length needed may be large. For example, when all $(x_{t+c_1}, x_{t+c_2}, \ldots, x_{t+c_m})$ are only in a subspace of $\mathbb{F}_2^m$, and $c_m - c_1 + m - 1$ is large. Once a parity-check is found, more could be constructed by sliding and adding together.

Alternatively, the parity-checks with 2 taps may be indirectly constructed for some very special cases. For example, assume that we can find a parity-check with 4 taps, which has the following form

$$E\boldsymbol{x}_t + G\boldsymbol{x_{t-a}} + G'\boldsymbol{x}_{t-b-a} + G'G\boldsymbol{x}_{t-b-2a} = 0.$$

Moreover, if it happens that the multidimensional linear approximation has the form

$$U\boldsymbol{x}_t \oplus UG\boldsymbol{x}_{t-a} \oplus \bigoplus_{i \in \{1,\ldots,\#\mathcal{T}_z\}, j(i) \in \mathcal{T}_z} V_i \boldsymbol{z}_{t+j(i)} = \boldsymbol{e},$$

Thus we have a parity-check with 2 taps $E\boldsymbol{y_t} + G'\boldsymbol{y}_{t-b-a} = 0$ for a new sequence $\boldsymbol{y}_t = E\boldsymbol{x_t} + G\boldsymbol{x}_{t-a}$. When $\boldsymbol{x}$ is a binary m-sequence with period $2^n - 1$, the above 4 taps parity-check with small $b + 2a$ does not exist, when $\gcd(a, 2^n - 1) = 1$. The reason is that b must be a multiple of the period of sequence $\boldsymbol{y}$, whose period is also $2^n - 1$. Thus if there is a low degree one, then at least $\gcd(a, 2^n - 1)$ is large.

Besides that, we have the following observations. If a parity-check satisfies sequence $\boldsymbol{x}$, then its characteristic polynomial $F_n(x) \in \mathbb{F}_2[x]$ has $f(x)$ as a factor. Since $G_n = G, G_1 = \ldots = G_{n-1} = 0$, then $F_n(x) = \det(Ex^n + G)$, the number of choices for matrix G and n is

$(N/m-1)|GL_m(\mathbb{F}_2)|$. Let $\mathcal{S}=\{F_n(x): 1\leq nm \leq N\}$ denote all possible characteristic polynomials. For convenience, we introduce a map sending $F_n(x)\in\mathcal{S}_G$ to $\mathbb{F}_2[x]$.

$$\phi: \mathcal{S}_G \to \mathbb{F}_2[x]$$
$$F_n(x)=\det(Ex^n+G)\to F(x)=\det(Ex+G).$$

Since $F(x)$ is the characteristic polynomial of invertible matrix G, the number of different $F(x)$ is 2^{m-1}. Suppose that $F(x)=f_1^{n_1}\ldots f_v^{n_v}$, where all f_i are distinct irreducible polynomials of degree d_i, it has been proved that the number of G with given $F(x)$ is $\theta(F(x))$ [Ger61], i.e.,

$$\theta(F(x))=\frac{2^{m^2-m}\prod_{i=1}^{m}(1-2^{-i})}{\prod_{i=1}^{v}\prod_{j=1}^{n_i}(1-2^{-jd_i})}.$$

We also know that $F_{n_1}(x)=F_{n_2}(x)^{2^i}$ for some $i>0$ when n_1 and n_2 are in the same 2-cyclotomic coset $\mathcal{C}_{\bar{n}}$ modulo $ord(f)=2^k-1$. And the size of set $\mathcal{F}=\{F_{\bar{n}}(x): 1<nm<N\}$ is bounded by $N/(km)<\#\mathcal{F}\leq\sum_{d|k}\mu(d)\sum_{i=1}^{k/d}2^i$, where $\mu(\cdot)$ is Möbius function.

For the case $\tau\geq 2$, there are about $(N/m-1)|GL_m(\mathbb{F}_2)|(2^{m^2}-1)$ choices for the two coefficients and n. An upper bound of $\#\mathcal{S}$ is the number of conjugacy classes of T in $GL_{nm}(\mathbb{F}_2)$, which is roughly about $2^{nm}-\sum_{i=\lfloor nm/3\rfloor}^{\lfloor (nm-1)/2\rfloor}2^i$. We believe it is much more than $(2^m-1)^2$ when $L(x)\in\mathbb{F}_{2^m}$.