Cryptanalysis of an E0-like Combiner with Memory

Yi Lu[1] and Serge Vaudenay[2]

[1] School of Physical and Mathematical Sciences
Nanyang Technological University, Singapore
[2] EPFL, CH-1015 Lausanne, Switzerland
http://lasecwww.epfl.ch

Abstract. In this paper, we study an E0-like combiner with memory as the keystream generator. First, we formulate a systematic and simple method to compute correlations of the FSM output sequences (up to certain bits). An upper bound of the correlations is given, which is useful to the designer. Second, we show how to build either a uni-bias-based or multi-bias-based distinguisher to distinguish the keystream produced by the combiner from a truly random sequence, once correlations are found. The data complexity of both distinguishers is carefully analyzed for performance comparison. We show that the multi-bias-based distinguisher outperforms the uni-bias-based distinguisher only when the patterns of the largest biases are linearly dependent. The keystream distinguisher is then upgraded for use in the key-recovery attack. The latter actually reduces to the well-known Maximum Likelihood Decoding (MLD) problem given the keystream long enough. We devise an algorithm based on Fast Walsh Transform (FWT) to solve the MLD problem for any linear code with dimension L and length n within time $O(n + L \cdot 2^L)$. Meanwhile, we summarize a design criterion for our E0-like combiner with memory to resist the proposed attacks.

Keywords. stream cipher, combiner, Bluetooth, E0, correlation

1 Introduction

To protect confidentiality, stream ciphers are often used in the constrained environment (e.g. high speed, minimal area, limited power supply, low power consumption). For this reason, wireless encryption often uses stream ciphers (e.g. A5/1 in GSM, E0 in Bluetooth, RC4 in WEP).

Many stream ciphers are based on Linear Feedback Shift Registers (i.e. LFSRs [41]). They use different mechanisms such as the irregular clocking, the nonlinear combination function or the nonlinear filtering function to destroy the fatally weak property of LFSRs: linearity. We call them by clock-controlled generators, nonlinear combiners and nonlinear filter generators respectively.

As one of the mainstream attacks on LFSR-based stream ciphers, correlation attack was first introduced by Siegenthaler [50] to attack the nonlinear combiners. The basic idea is to "divide and conquer" when the keystream output is correlated to the individual LFSR output sequence due to the poor choice of the combining function. That is, instead of the naive exhaustive search on all possible combinations of the initial states of the component LFSRs, we only perform an exhaustive search for the initial state of each individual LFSR independently and test the correlation between each LFSR output sequence and the keystream. The optimum (deterministic) Maximum Likelihood Decoding (MLD) strategy yields the answer for the initial state of the LFSR. This idea can be applied to attack nonlinear filter generators (e.g. [20, 23, 47, 51]).

Apparently, the time complexity of the basic correlation attack [50] grows exponential in the length of the LFSR, which is impractical for a long LFSR. As a matter of fact, in coding theory, the MLD problem for linear codes, according to [5], was shown to be NP-complete (see [21] for definition). The focus of cryptographers has been on the general problem where the individual LFSR may be arbitrarily long. In order to speed up the attack for the general setting, Meier and Staffelbach [38, 39] used the probabilistic iterative decoding strategy to refine the basic correlation attack into a so-called "fast correlation attack" to reconstruct each individual LFSR. A critical factor for the efficiency of the fast correlation attack is the novel use of the multiple polynomial of the LFSR's feedback polynomial with low weight (and low degree). This fast correlation attack of [38, 39] was improved by a series of variant fast correlation attacks (e.g. [10, 13, 44–46, 55]). Recently, various (still probabilistic) decoding techniques have proved very successful to further improve the performance of the fast correlation attack (e.g. [8, 9, 11, 12, 27–29, 42, 43]).

As a new emerging short-range wireless radio standard with low power consumption, Bluetooth [6] uses the stream cipher E0. It is a combiner with memory and actually a variant of the summation generator [48]. In this paper, we propose an E0-like combiner with memory as the stream cipher. A systematic computation method is formulated to calculate correlations of the FSM output sequences (up to certain bits) by a recursive expression. Furthermore, we give an upper bound of the correlations, which is useful to the designer. Prior to our work, correlation properties of combiners with one-bit memory, and with m-bit memory were studied in [40], and [22] respectively. As they considered correlations of a general form, the length of the correlation pattern is restricted to be rather small for the analysis. By comparison, as we restrict ourselves to a spe-

cial class of correlations (i.e. correlations of the FSM output sequence), we are able to investigate those correlations with the sequence length of much a wider range. This is quite an important result, since the search of a correlation as large as possible constitutes one of the crucial tasks for efficient correlation attacks on LFSR-based stream ciphers.

When correlations are found, we can build either a uni-bias-based or multi-bias-based distinguisher to distinguish the keystream produced by the combiner from a truly random sequence. We apply the concept of convolution to the data complexity analysis of the multi-bias-based distinguisher that uses all the correlations. Based on the theory of [4], we show that the multi-bias-based distinguisher outperforms the uni-bias-based distinguisher only when the patterns of the largest biases are linearly dependent.

The keystream distinguisher not only enables the keystream distinguishing attack, but also can upgrade into the key-recovery attack to reconstruct the initial states of the LFSRs. The latter actually reduces to the well-known MLD problem given the keystream long enough (or the bias large enough). By means of Fast Walsh Transform (FWT), we devise an algorithm to solve the MLD problem for any linear code with dimension L and length n within time $O(n + L \cdot 2^L)$. It is the best deterministic decoding algorithm known so far. Interestingly, an FWT-based algorithm was proposed in another context to speed up other kinds of fast correlation attacks [12].

Finally, the analysis principle is successfully applied to the core of Bluetooth encryption algorithm E0. Our key-recovery attack reconstructs the initial states of the LFSRs in time 2^{39} given 2^{39} consecutive keystream bits after $O(2^{37})$ precomputation[3]. This is the best academic key-recovery attack against the core E0 compared with all the attacks [1, 2, 14, 16–19, 24–26, 30, 49] on the core E0. Considering a maximal keystream length of 2745 bits for E0 used in Bluetooth, the attack is impractical. Nonetheless, the resynchronization flaw of E0 (see [34]) enables us to deduce non-trivial correlations of full E0 from those of the core E0; this finally leads to the fastest (and only) practical known-plaintext attack on full E0 in 2005 (see [33]).

As part of the thesis [32], this paper extends the results of [35] with a more general approach, and summarizes a design criterion for our E0-like combiner with memory to resist the proposed attacks. The rest of the paper is structured as follows. In section 2, we give the mathematical

[3] Throughout this paper, $O(\cdot)$ is used to provide a rough estimate on complexities, e.g., $O(2^{37})$ here means $c \cdot 2^{37}$ operations, where c is a small constant.

model of our E0-like combiner with memory. Then we study the correlation properties of the FSM output sequence in Section 3. The correlation properties enables to mount the distinguishing attack on our combiner in Section 4; we first build a uni-bias-based distinguisher and then a multi-bias-based distinguisher, and performance comparison between the two is also analyzed. In Section 5, we study the key-recovery attack based on our former distinguishing attacks; we show that the key-recovery attack reduces to the MLD problem. In Section 6 we investigate the MLD algorithm for a linear code. We conduct a case study on the core E0 in Section 7. Finally, we give conclusions in Section 8.

2 Mathematical Model

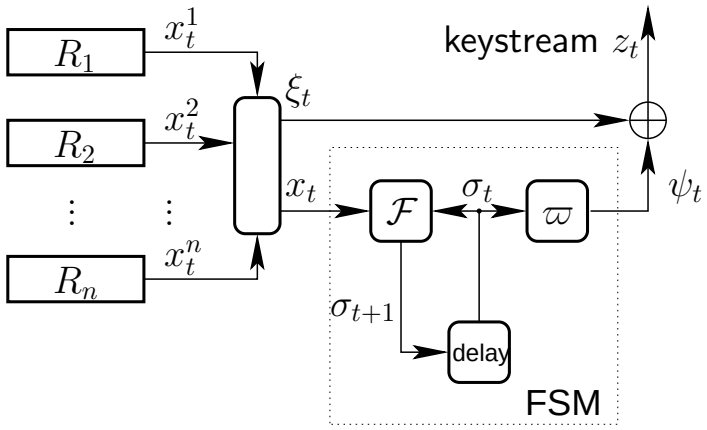

Fig. 1. The core stream cipher

Our model of the E0-like combiner with memory is depicted in Fig. 1. It belongs to the LFSR-based combiner (with or without memory). To briefly outline, the keystream generator consists of n maximum-length LFSRs denoted by $R_1, \ldots, R_n$. Let the R_i have pairwise distinct lengths L_i (for convenience, let $L_1 < L_2 < \cdots < L_n$) and primitive characteristic polynomials $p_i(x)$. Besides, the combination generator has a Finite State Machine (FSM) of k memory bits. Denote the k-bit state at time t by $\sigma_t = (\sigma_t^{k-1}, \ldots, \sigma_t^0)$. We denote λ_t hereafter the content of LFSRs at time t. Then the state of the combiner at time t is fully represented by the $(L+k)$-bit pair (λ_t, σ_t), where $L = \sum_{i=1}^{n} L_i$.

At each clock cycle t, the LFSRs output bits $x_t = (x_t^1, x_t^2, \ldots, x_t^n)$ serve as the input to the FSM. Its next state σ_{t+1} can be expressed by a nonlinear function $\mathcal{F}$ of its current state σ_t and x_t, i.e.

$$\sigma_{t+1} = \mathcal{F}(x_t, \sigma_t). \tag{1}$$

The FSM emits one bit

$$\psi_t = \varpi \cdot \sigma_t, \tag{2}$$

which is an inner product[4] of its current state σ_t and the constant $\varpi \in GF(2)^k$. Finally, the combiner generates one bit z_t of keystream, which is obtained by xoring one FSM output bit ψ_t together with the sum of the LFSRs outputs, that is,

$$\xi_t \oplus \psi_t = z_t, \tag{3}$$

where $\xi_t = \bigoplus_{i=1}^{n} x_t^i$.

Lemma 1. *Assuming that $\sigma_t \mapsto \sigma_{t+1}$ is a permutation for any x_t, if σ_0 is random and uniformly distributed, then, σ_t is random and uniformly distributed for any t. If λ_0 is random and uniformly distributed, then, λ_t is random and uniformly distributed for any t. If (λ_0, σ_0) is random and uniformly distributed, the L_1-tuple $(\sigma_0, \sigma_1, \ldots, \sigma_{L_1-1})$ is independent of x_{L_1-1}.*

Proof. Noticing that $\lambda_t \mapsto \lambda_{t+1}$ is a permutation, by induction, we know that $\lambda_0 \mapsto \lambda_t$ is a permutation for any t. Similarly, we deduce that $\sigma_0 \mapsto \sigma_t$ is a permutation for any t.

To prove the remaining part of the lemma, as $x_0, \ldots, x_{L_1-1}$ are contained in λ_0, we know that $L_1 - 1$ consecutive vectors $x_0, \ldots, x_{L_1-2}$ are i.i.d. random variables all independent of both σ_0 and x_{L_1-1} assuming that (λ_0, σ_0) is random and uniformly distributed. From this statement we apply Eq.(1) consecutively for $t = 0, \ldots, L_1-2$ and deduce that the L_1-tuple $(\sigma_0, \sigma_1, \ldots, \sigma_{L_1-1})$ is independent of x_{L_1-1} assuming that (λ_0, σ_0) is random and uniformly distributed.

□

Throughout this paper, we restrict ourselves to $\mathcal{F}$ that satisfies $\sigma_t \mapsto \sigma_{t+1}$ is a permutation for any x_t.

[4] An inner product between two ℓ-bit binary vectors $x = (x_1, \ldots, x_\ell)$ and $y = (y_1, \ldots, y_\ell)$ is defined by $x \cdot y \stackrel{\text{def}}{=} x_1 y_1 \oplus \cdots \oplus x_\ell y_\ell$.

3 Correlation Properties

Definition 2. *The bias of a random Boolean variable X is defined as*

$$\Delta(X) \stackrel{def}{=} \Pr(X=0) - \Pr(X=1) = \mathrm{E}[(-1)^X].$$

The correlation between two random Boolean variables X and Y is $\Delta(X \oplus Y)$. Assuming that (x_0, σ_0) is a uniformly distributed random vector of $(n+k)$ bits, we know that given $a, b \in GF(2)^k$, $\Delta(a \cdot \sigma_1 \oplus b \cdot \sigma_0)$ is a fixed value, which can be computed as follows. For all possible (x_0, σ_0), we use Eq.(1) to compute σ_1; thus, we can collect all possible (σ_0, σ_1) and calculate $\Delta(a \cdot \sigma_1 \oplus b \cdot \sigma_0)$ by Definition 2. The following lemma, inspired by [26], gives an easy way to compute the bias for iterative structures.

Lemma 3. *Given a set $\mathcal{E}$ and $\Theta : \mathcal{E} \times GF(2)^k \to GF(2)$ and $\Lambda : GF(2)^\epsilon \to GF(2)^k$, let X and Y be two independent random variables in $\mathcal{E}$ and $GF(2)^\epsilon$ respectively. Assuming that $\Lambda(Y)$ is uniformly distributed in $GF(2)^k$, then, for any $v \in GF(2)^\epsilon$, we have*

$$\Delta\left(\Theta\left(X, \Lambda(Y)\right) \oplus v \cdot Y\right) = \sum_{w \in GF(2)^k} \Delta(\Theta(X, \Lambda(Y)) \oplus w \cdot \Lambda(Y)) \times \Delta(w \cdot \Lambda(Y) \oplus v \cdot Y).$$

Proof. Let $Z = \Lambda(Y)$. By our assumption, Z is a random variable independent of X with uniform distribution. We have $\Delta(\Theta(X, \Lambda(Y)) \oplus w \cdot \Lambda(Y)) = \Delta(\Theta(X, Z) \oplus w \cdot Z)$ for any $w \in GF(2)^k$. We rewrite the right-hand side as follows:

$$\begin{aligned}
&\sum_w \Delta(\Theta(X,Z) \oplus w \cdot Z) \cdot \Delta(w \cdot \Lambda(Y) \oplus v \cdot Y) \\
&= \sum_w \mathrm{E}[(-1)^{\Theta(X,Z) \oplus w \cdot Z}] \cdot \mathrm{E}[(-1)^{w \cdot \Lambda(Y) \oplus v \cdot Y}] \\
&= \sum_w \left(\sum_{x,z} \Pr(x,z) \cdot (-1)^{\Theta(X,Z) \oplus w \cdot Z} \right) \cdot \left(\sum_y \Pr(y) \cdot (-1)^{w \cdot \Lambda(Y) \oplus v \cdot Y} \right) \\
&= \sum_x \sum_y \sum_z \sum_w \Pr(x,z) \cdot \Pr(y) \cdot (-1)^{\Theta(x,z) \oplus v \cdot y \oplus w \cdot (z \oplus \Lambda(y))},
\end{aligned}$$

As the inner sum over w is zero for all $z \neq \Lambda(y)$, we continue

$$\begin{aligned}
&2^k \cdot \sum_{x,y} \Pr(X = x, Z = \Lambda(y)) \cdot \Pr(Y = y) \cdot (-1)^{\Theta(x, \Lambda(y)) \oplus v \cdot y} \\
&= \sum_{x,y} \Pr(x,y) \cdot (-1)^{\Theta(x, \Lambda(y)) \oplus v \cdot y} \\
&= \mathrm{E}[(-1)^{\Theta(X, \Lambda(Y)) \oplus v \cdot Y}],
\end{aligned}$$

which is $\Delta(\Theta(X, \Lambda(Y)) \oplus v \cdot Y)$. $\square$

Now we introduce the general iterative computation method to calculate the biases.

Theorem 4. *Assuming that $\sigma_t \mapsto \sigma_{t+1}$ is a permutation for any x_t and that (λ_0, σ_0) is uniformly distributed, for any $\epsilon \leq L_1+1$ and $a, b, \alpha_1, \ldots, \alpha_\epsilon \in GF(2)^k$, we define*

$$\delta(\alpha_1, \ldots, \alpha_\epsilon) \stackrel{def}{=} \Delta(\alpha_1 \cdot \sigma_t \oplus \cdots \oplus \alpha_\epsilon \cdot \sigma_{t+\epsilon-1})$$

and the state transition matrix $U = \{U_{ab}\}$ where

$$U_{ab} \stackrel{def}{=} \Pr(\sigma_{t+1} = b | \sigma_t = a).$$

$\delta(\alpha_1, \ldots, \alpha_\epsilon)$ and U_{ab} do not depend on t. Additionally, we have

$$\delta(\alpha_1, \ldots, \alpha_\epsilon) = \frac{1}{2^k} \sum_{w \in GF(2)^k} \widehat{U}_{w,\alpha_\epsilon} \cdot \delta(\alpha_1, \ldots, \alpha_{\epsilon-2}, \alpha_{\epsilon-1} \oplus w),$$

where $\widehat{U}$ is the Walsh transform of U.

Proof. We apply Lemma 3 with $X = x_{\epsilon-2}$, $Y = (\sigma_0, \ldots, \sigma_{\epsilon-2})$, $\Lambda(Y) = \sigma_{\epsilon-2}$, $\Theta(X, \Lambda(Y)) = \alpha_\epsilon \cdot \mathcal{F}(x_{\epsilon-2}, \sigma_{\epsilon-2}) = \alpha_\epsilon \cdot \sigma_{\epsilon-1}$ and $v = (\alpha_1, \ldots, \alpha_{\epsilon-1})$. Note that the assumption of Lemma 3 holds by Lemma 1, and that the connection with $\widehat{U}$ comes from $\delta(a, b) = \Delta(a \cdot \sigma_0 \oplus b \cdot \sigma_1) = \frac{\widehat{U}_{ab}}{2^k}$. $\square$

In order to state our result on the upper bound of the correlations for the combiner's FSM output sequence of short length, we recall a few definitions from information theory (see [15]). The entropy $H(X)$ of a discrete random variable X with alphabet $\mathcal{X}$ is defined by

$$H(X) \stackrel{\text{def}}{=} -\sum_{x \in \mathcal{X}} \Pr(x) \log_2 \Pr(x).$$

The binary entropy function $h(p)$ for $0 < p < 1$ is defined by

$$h(p) \stackrel{\text{def}}{=} -p \log_2 p - (1-p) \log_2 (1-p).$$

The conditional entropy $H(Y|X)$ of Y given X is

$$H(Y|X) \stackrel{\text{def}}{=} \sum_{x \in \mathcal{X}} \Pr(x) H(Y | X = x).$$

The following results on their relationship are useful for us. For any two random variables X, Y we always have $H(X) \geq H(X|Y)$ with equality if

and only if X and Y are independent. Analogously, for any three random variables X, Y and Z, we always have $H(X|Z) - H(X|Y,Z) \geq 0$ with equality if and only if X and Y are conditionally independent given Z. Based on information theory, we have the following lemma.

Lemma 5. *With the assumptions of Theorem 4, there exists a positive ρ such that*

$$H(\psi_1|\sigma_0) = h\left(\frac{1}{2} + \frac{\rho}{2}\right),$$

and ρ only depends on the state transition matrix U.

Proof. We compute $H(\psi_1|\sigma_0)$ by definition:

$$\begin{aligned} H(\psi_1|\sigma_0) &= \sum_a H(\psi_1|\sigma_0 = a) \cdot \Pr(\sigma_0 = a) \\ &= \mathop{\mathrm{E}}_a \left[h\left(\frac{1}{2} + \frac{1}{2} \sum_{b:\varpi\cdot b=1} U_{ab} - \frac{1}{2} \sum_{b:\varpi\cdot b=0} U_{ab} \right) \right]. \end{aligned} \tag{4}$$

So there exists such a unique $\rho \geq 0$ to satisfy the equation in Lemma 5, that is,

$$h\left(\frac{1}{2} + \frac{\rho}{2}\right) = \mathop{\mathrm{E}}_a \left[h\left(\frac{1}{2} + \frac{1}{2} \sum_{b:\varpi\cdot b=1} U_{ab} - \frac{1}{2} \sum_{b:\varpi\cdot b=0} U_{ab} \right) \right], \tag{5}$$

and from Eq.(5) we know that ρ depends on U only. □

Note that Eq.(5) tells that if $|\sum_{b:\varpi\cdot b=1} U_{ab} - \sum_{b:\varpi\cdot b=0} U_{ab}|$ is a constant ρ_0 for all a, then, $\rho = \rho_0$. In particular, $\rho = 0$ if and only if $\sum_{b:\varpi\cdot b=1} U_{ab} \equiv \sum_{b:\varpi\cdot b=0} U_{ab}$, i.e. $\Pr(\psi_1 = 1|\sigma_0 = a) = \Pr(\psi_1 = 0|\sigma_0 = a) = \frac{1}{2}$, for all a. Equivalently, $\rho = 0$ if and only if $H(\psi_1|\sigma_0) = 1$, that is, $\rho = 0$ if and only if $\varpi \cdot \mathcal{F}(x_0, \sigma_0)$ and σ_0 are independent assuming (x_0, σ_0) is uniformly distributed.

From Lemma 5, we can prove the upper bound of the correlations for the combiner's FSM output sequence of short length.

Theorem 6. *For any $\epsilon \leq L_1 + 1$ and any binary $\alpha_1, \ldots, \alpha_{\epsilon-1}$, let the ϵ-bit vectors $\alpha = (\alpha_1, \ldots, \alpha_{\epsilon-1}, 1)$ and $\Psi = (\psi_0, \ldots, \psi_{\epsilon-1})$. With the assumptions of Theorem 4, we have*

$$|\Delta(\alpha \cdot \Psi)| \leq \rho,$$

where ρ is defined in Eq.(5).

Proof. First, by the property of the relation between the entropy and the conditional entropy, we deduce that

$$H(\alpha \cdot \Psi) \geq H(\alpha \cdot \Psi | \sigma_0, \ldots, \sigma_{\epsilon-2}) = H(\psi_{\epsilon-1} | \sigma_0, \ldots, \sigma_{\epsilon-2}). \quad (6)$$

According to the property of the conditional entropy, we have

$$H(\psi_{\epsilon-1} | \sigma_{\epsilon-2}) - H(\psi_{\epsilon-1} | \sigma_0, \ldots, \sigma_{\epsilon-2}) \geq 0$$

with equality if and only if $\psi_{\epsilon-1}$ and $(\sigma_{\epsilon-3}, \ldots, \sigma_0)$ are conditionally independent given $\sigma_{\epsilon-2}$, which is valid here by the precondition $\epsilon \leq L_1 + 1$ and Lemma 1. Thus, we have

$$H(\psi_{\epsilon-1} | \sigma_{\epsilon-2}) = H(\psi_{\epsilon-1} | \sigma_0, \ldots, \sigma_{\epsilon-2}) \quad (7)$$

Combining Eq.(6) and Eq.(7), we get $H(\alpha \cdot \Psi) \geq H(\psi_{\epsilon-1} | \sigma_{\epsilon-2}) = h(\frac{1}{2} + \frac{\rho}{2})$. Because $h(p)$ is symmetric in $p = \frac{1}{2}$ with the maximum at $p = \frac{1}{2}$, this is equivalent to

$$\frac{1}{2} - \frac{\rho}{2} \leq \Pr(\alpha \cdot \Psi = 0) \leq \frac{1}{2} + \frac{\rho}{2}, \quad (8)$$

Finally, we verify

$$\begin{aligned} |\Delta(\alpha \cdot \Psi)| &= |\Pr(\alpha \cdot \Psi = 0) - \Pr(\alpha \cdot \Psi \neq 0)| \\ &= |2 \cdot \Pr(\alpha \cdot \Psi = 0) - 1|. \end{aligned} \quad (9)$$

Putting Eq.(8) and Eq.(9) together we complete our proof. □

Remark 7. This theorem tells that the basic FSM design principle should satisfy $H(\psi_1 | \sigma_0) = 1$ to avoid the bias, which enables the keystream distinguishing attack and key-recovery attack as detailed in the rest of the paper.

Notice that the only purpose of the restriction on the dimension of α (i.e. $\epsilon \leq L_1 + 1$), is to ensure validity of U being the state transition matrix. In other words, if we loose this requirement by supposing U is always the state transition matrix[5], we still obtain the same upper bound ρ for $|\Delta(\alpha \cdot \Psi)|$. Though it is not known yet which tuple(s) α makes $|\Delta(\alpha \cdot \Psi)|$ the maximum from Theorem 6, one thing is certain[6]: once $H(\psi_1 | \sigma_0) = 1$, no correlation exists for sequences of bitlength up to $L_1 + 1$.

Prior to our work, correlation properties of combiners with one-bit memory, and with m-bit memory were studied in [40], and [22] respectively. As they considered correlations of a general form (i.e. correlation

[5] This is a (weak) common assumption in cryptanalysis.

[6] This result was published recently by an independent work [3] with different proof.

between any linear function of the sequence $\{\xi_t\}$ of ϵ bits and any linear function of the keystream $\{z_t\}$ of ϵ bits), ϵ is restricted to be rather small for the analysis. In our work, we restrict ourselves to a special class of correlations—correlations of the FSM output sequence (i.e. correlations of any linear function of the sequence $\{\xi_t \oplus z_t\}$ of ϵ bits). This allows to investigate those correlations for the sequence length $\epsilon \leq L_1 + 1$ with much a wider range[7].

4 The Keystream Distinguisher

4.1 The Equivalent Single LFSR

Let θ_i be the order of the characteristic polynomial $p_i(x)$ of R_i, for $i = 1, \ldots, n$. Since all $p_i(x)$ are primitive polynomials, $\theta_i = 2^{L_i} - 1$; furthermore, by Lemma 6.57 of [31, p.218], the equivalent LFSR which generates the same sequence $\{\xi_t\}$ as the sum of the n original LFSR outputs over $GF(2)$ has the characteristic polynomial $p(x) = \prod_{i=1}^{n} p_i(x)$ with order $\theta = \mathrm{lcm}(\theta_1, \theta_2, \ldots, \theta_n)$ (by Lemma 6.50, [31, p.214]) and degree $L = \sum_{i=1}^{n} L_i$.

4.2 Finding the Multiple Polynomial with Low Weight

Let L be the degree of a general polynomial $p(x)$ with order θ. We use the standard approximation[8] to estimate the minimal weight w_d of multiples of $p(x)$ with degree at most d by the following constraint: w_d is the smallest w such that

$$\frac{1}{2^L} \times \binom{d}{w-1} \geq 1. \tag{10}$$

Listed in Table 1 is the estimated[9] w_d corresponding to d with $L = 128$ by solving Inequality (10).

To find multiples with minimum weight, Canteaut and Chabaud [7] proposed an efficient algorithm for a not too large degree d (e.g. less than 2^{11}). Here, we are interested in the case with very large $d \gg 2^{11}$. So we

[7] For instance, in the core of E0 (described in Section 7.1 later), according to [24], the sequence length $\epsilon \leq 6$ by analysis of [22] for general correlations; in contrast, for the special class of correlations in our work the sequence length $\epsilon \leq 27$ (see Section 7.2).

[8] Note that this approximation of (10) is valid for typical settings in cryptography. However, it may not hold for some special cases (e.g. some of the products of two primitive polynomials with the same degree do not have any multiple polynomial of weight 3).

[9] One special case occurs for $d = \theta$ because we know the exact value of w_d.

can use the conventional birthday paradox to find $Q(x)$ with the minimal d (i.e. $w = w_d$), which takes precomputation time $PT \approx O(d^{\lceil \frac{w-1}{2} \rceil})$; alternatively, we can use the generalized birthday problem to find $Q(x)$ of same weight but higher degree with much less precomputation as tradeoff (see [53] for detail). Table 2 compares the two algorithms. Note that unless otherwise mentioned explicitly in the notations, throughout the paper, we always use $\log(\cdot)$ to represent the natural logarithm to the base of e, which is omitted from the notations.

Table 1. The estimated minimal weight w_d of multiples of $p(x)$ with degree d and order θ by (10), where $L = 128$

d	247	458	855	1749	2387	2^{18}	2^{23}	2^{27}	2^{33}	2^{44}	2^{65}	θ
w_d	≈ 31	≈ 24	≈ 20	≈ 17	≈ 16	≈ 9	≈ 7	≈ 6	≈ 5	≈ 4	≈ 3	$= 2$

Table 2. Complexity PT of finding multiple of $p(x)$ with degree d, weight w where $L = 128$

	birthday problem							
	with minimal d						tradeoff	
d	2^{18}	2^{23}	2^{27}	2^{33}	2^{44}	2^{65}	2^{32}	2^{43}
w	9	7	6	5	4	3	9	5
$\log_2 PT$	72	69	68	66	66	65	35	45

4.3 Building a Uni-bias-based Distinguisher

Let $Q(x) = \sum_{i=1}^{w} x^{q_i}$ be the normalized multiple of $p(x) = \prod_{i=1}^{n} p_i(x)$ with degree d and weight w, where $0 = q_1 < q_2 < \ldots < q_w = d$. Let α be the ϵ-bit binary vector such that $|\gamma|$ is maximal where $\gamma = \Delta(\alpha \cdot (\psi_t, \ldots, \psi_{t+\epsilon-1}))$. As $\bigoplus_{i=1}^{w} \xi_{t_0+q_i} = 0$ holds for all t_0, by Eq.(3), we deduce that

$$\bigoplus_{i=1}^{w} \alpha \cdot (z_{t_0+q_i}, \ldots, z_{t_0+q_i+\epsilon-1}) = \bigoplus_{i=1}^{w} \alpha \cdot (\psi_{t_0+q_i}, \ldots, \psi_{t_0+q_i+\epsilon-1}). \quad (11)$$

With the heuristic assumption of independence, we know from the famous Piling-up Lemma [37] that the right-hand side of Eq.(11) has a bias $|\gamma|^w$

(resp. $-|\gamma|^w$) if γ is positive (resp. negative). With standard linear cryptanalysis techniques, we can therefore distinguish the keystream $\{z_t\}$ from a truly random sequence with a number of samples within the order of magnitude of $\zeta = \gamma^{-2\cdot w}$, simply by checking the left-hand side of Eq.(11) equals zero (resp. one) most of the time with the positive (resp. negative) γ. Based on $Q(x)$ with d and w, we minimize the data complexity Ξ by choosing $\Xi = \zeta + d = \gamma^{-2\cdot w} + d$.

4.4 The Multi-bias-based Distinguisher

Preliminaries

Definition 8. *Given $f, g : GF(2)^\ell \to \mathbf{R}$, for $a \in GF(2)^\ell$, we define*

1. $(f \otimes g)(a) = \sum_{b \in GF(2)^\ell} f(b) \cdot g(a \oplus b)$
   $f^{\otimes w}(a) = (\underbrace{f \otimes \cdots \otimes f}_{w \text{ times}})(a)$
2. $\widehat{f}(a) = \sum_{b \in GF(2)^\ell} (-1)^{a \cdot b} f(b)$
3. $\|f\| = \sqrt{\sum_{a \in GF(2)^\ell} f^2(a)}$
4. $\Delta(f) = 2^{\frac{\ell}{2}} \cdot \left\| f - \frac{1}{2^\ell} \cdot \mathbf{1} \right\|$, *where* $\mathbf{1}$ *denotes a constant function equal to* 1.

Note that the first two definitions correspond to convolution and Walsh transform respectively. We recall these basic facts: for any $f, g : GF(2)^\ell \to \mathbf{R}$, we have

- $\widehat{f \otimes g}(a) = \widehat{f}(a) \cdot \widehat{g}(a)$, for all $a \in GF(2)^\ell$;
- $2^\ell \|f\|^2 = \|\widehat{f}\|^2$;
- if f is a distribution, i.e. $\sum_a f(a) = 1$ and $f(a) \geq 0$ for all $a \in GF(2)^\ell$, then the distribution of the XOR of w i.i.d. random vectors with distribution f is $f^{\otimes w}$, moreover, $\Delta^2(f) = \sum_{a \neq \mathbf{0}} \widehat{f}^{\,2}(a)$;
- If the random Boolean variable A follows the distribution f, then $\Delta(f) = \Delta(A)$, where $\Delta(A)$ is defined in Definition 2, Section 3.

An Efficient Way to Deploy Multi-Biases Simultaneously We are interested in the possibility of further improving the performance of the distinguisher by using more than one bias simultaneously. To address this problem, we introduce a linear mapping $J : GF(2)^\nu \to GF(2)^\ell$ of rank ℓ. Our goal is to find a better J to lower the date complexity. Define ℓ-bit vectors

$$A_t = J(\psi_{\ell t}, \ldots, \psi_{\ell t+\nu-1})$$
$$B_t = \bigoplus_{i=1}^{w} A_{t+q_i}$$

Note that B_t can be derived from the keystream $\{z_t\}$ directly. Except for accidentally bad choices of J, we make a heuristic assumption that all A_t's are independent. Let $\mathcal{D}$ be the probability distribution of the ν-bit vector $(\psi_{\ell t}, \ldots, \psi_{\ell t+\nu-1})$, and let $\mathcal{D}_A$ be the probability distribution of the ℓ-bit vector A_t. Note that $\mathcal{D}_A$ and $\mathcal{D}$ are linked by

$$\mathcal{D}_A(b) = \sum_{a \in GF(2)^\nu} \mathcal{D}(a) \cdot \mathbf{1}_{b=J(a)}$$

for any $b \in GF(2)^\ell$. Moreover, the Walsh transforms of $\mathcal{D}_A$ and $\mathcal{D}$ are also linked by

$$\widehat{\mathcal{D}}_A(b) = \widehat{\mathcal{D}}\left(J^\top(b)\right),$$

for all $b \in GF(2)^\ell$. Now we discuss how to design J in order to reduce the data complexity. From Baignères et al. [4], we know that we can distinguish a distribution f of ℓ-bit random vectors from a uniform distribution with $1/\Delta^2(f)$ samples. Here, the distribution of B_t is $f = \mathcal{D}_A^{\otimes w}$. So the modified distinguisher needs data complexity

$$\Xi = \frac{\ell}{\Delta^2(\mathcal{D}_A^{\otimes w})} + d \text{ (bits)}.$$

Let μ be the number of nonzero b such that the Walsh coefficient $\widehat{\mathcal{D}}_A(b)$ has the largest absolute value[10] (denoted by η). Since $\Delta^2(\mathcal{D}_A^{\otimes w}) \approx \mu\eta^{2w}$, we have $\Xi \approx \frac{\ell}{\mu}\eta^{-2w} + d$. In order to lower Ξ, it is necessary to have $\ell < \mu$. This implies that only when the patterns of the μ largest coefficients are linearly dependent, the multi-bias distinguisher is more efficient than the uni-bias distinguisher; otherwise, the former is as efficient as the latter. Note that Section 4.3 actually deals with the special type of distinguishers with $\ell = \mu = 1$.

[10] Note that from Theorem 4 we have $\eta \le \gamma \le \rho$ for $\nu \le L_1 + 1$ regardless of ℓ and J, where ρ is defined in Eq.(5).

5 The Key-recovery Attack

We use the same approach as in [16] to transform our keystream distinguisher of Section 4 into a key-recovery attack to reconstruct the shortest LFSR (i.e. R_1).

Now, let $Q(x) = \sum_{i=1}^{w} x^{q_i}$ be a multiple polynomial of $\prod_{i=2}^{n} p_i(x)$ with degree d and weight w, which can be found by techniques in Section 4.2. Let $\tilde{\mathbf{x}}^{\mathbf{1}}$ be a guess for $\mathbf{x}^{\mathbf{1}}$, the initial state of R_1 which generates the keystream $\{z_t\}$ together with the other $n-1$ fixed LFSRs. Denote $\tilde{x}_t^1$ the output bit of R_1 with the initial state $\tilde{\mathbf{x}}^{\mathbf{1}}$ at time t. We define

$$r_t = \bigoplus_{i=1}^{w} \alpha \cdot (\tilde{x}_{t+q_i}^1, \ldots, \tilde{x}_{t+q_i+\epsilon-1}^1),$$
$$s_t = \bigoplus_{i=1}^{w} \alpha \cdot (z_{t+q_i}, \ldots, z_{t+q_i+\epsilon-1}),$$

for $t = 0, \ldots, \zeta - 1$ (corresponding to the data complexity $\Xi = \zeta + d$). It can be shown that $\{r_t\}$ is also an m-sequence generated by the same LFSR. Let $\mathbf{r}$ be the initial state. Let $b_t(\tilde{\mathbf{x}}^{\mathbf{1}}) \stackrel{\text{def}}{=} s_t \oplus r_t$ for $t = 0, \ldots, \zeta - 1$. Given ζ-bit sequence of $b_t(\tilde{\mathbf{x}}^{\mathbf{1}})$'s, we count the occurrences[11] $N(\tilde{\mathbf{x}}^{\mathbf{1}})$ of ones, that is,

$$N(\tilde{\mathbf{x}}^{\mathbf{1}}) \stackrel{\text{def}}{=} \sum_{t=0}^{\zeta-1} b_t(\tilde{\mathbf{x}}^{\mathbf{1}}). \tag{12}$$

Two cases of statistical characteristics arise. We use similar analysis [52] for the case $\gamma > 0$, which can be easily adjusted for $\gamma < 0$.

Case One: $\tilde{x}^1 = x^1$. We have

$$b_t(\tilde{\mathbf{x}}^{\mathbf{1}}) = \bigoplus_{i=1}^{w} \alpha \cdot (\psi_{t+q_i}, \ldots, \psi_{t+q_i+\epsilon-1}).$$

Recall from Section 4.3, we know that $p \stackrel{\text{def}}{=} \Pr(b_t(\tilde{\mathbf{x}}^{\mathbf{1}}) = 0) = \frac{1}{2} + \frac{\gamma^w}{2}$, assuming independence of all $\alpha \cdot (\psi_{t+q_i}, \ldots, \psi_{t+q_i+\epsilon-1})$ for $i = 1, \ldots, w$. So $N(\mathbf{x}^{\mathbf{1}})$ complies with the binomial distribution $\mathcal{B}(\zeta; p)$. As convention, when ζ is large and p is close to $\frac{1}{2}$, we approximate the binomial distribution of $N(\mathbf{x}^{\mathbf{1}})$ by the normal distribution $\mathcal{N}(\zeta p, \sqrt{\frac{\zeta}{4}})$, where the standard deviation is computed as $\sqrt{\zeta \cdot p(1-p)} \approx \sqrt{\frac{\zeta}{4}}$.

[11] w is fixed in the attack, so we omit it in the notation $N(\tilde{\mathbf{x}}^{\mathbf{1}})$.

Case Two: $\tilde{x}^1 \neq x^1$**.** We have

$$\sum_{\tilde{\mathbf{x}}^{\mathbf{1}} \in GF(2)^{L_1}} N(\tilde{\mathbf{x}}^{\mathbf{1}}) = \zeta \cdot 2^{L_1 - 1}$$

for any fixed keystream $\{z_t\}$. We immediately have

$$\mathrm{E}\left[\sum_{\tilde{\mathbf{x}}^{\mathbf{1}} \neq \mathbf{x}^{\mathbf{1}}} N(\tilde{\mathbf{x}}^{\mathbf{1}})\right] = \zeta \cdot 2^{L_1 - 1} - \zeta \cdot p$$

for any fixed keystream $\{z_t\}$. We deduce that the average of $N(\tilde{\mathbf{x}}^{\mathbf{1}})$ over all $\tilde{\mathbf{x}}^{\mathbf{1}} \neq \mathbf{x}^{\mathbf{1}}$ is

$$\mathrm{E}_{\tilde{\mathbf{x}}^{\mathbf{1}} \neq \mathbf{x}^{\mathbf{1}}}[N(\tilde{\mathbf{x}}^{\mathbf{1}})] = \frac{\mathrm{E}\left[\sum_{\tilde{\mathbf{x}}^{\mathbf{1}} \neq \mathbf{x}^{\mathbf{1}}} N(\tilde{\mathbf{x}}^{\mathbf{1}})\right]}{2^{L_1} - 1} = \frac{\zeta}{2} - \frac{\zeta(p - \frac{1}{2})}{2^{L_1} - 1} \approx \frac{\zeta}{2}.$$

So $N(\tilde{\mathbf{x}}^{\mathbf{1}})$ asymptotically complies with the binomial distribution $\mathcal{B}(\zeta; \frac{1}{2})$. Similarly as the former case, we approximate the binomial distribution of $N(\tilde{\mathbf{x}}^{\mathbf{1}})$ by the normal distribution $\mathcal{N}(\frac{\zeta}{2}, \sqrt{\frac{\zeta}{4}})$, where the standard deviation is computed as $\sqrt{\zeta \cdot \frac{1}{2}(1 - \frac{1}{2})} = \sqrt{\frac{\zeta}{4}}$. Since we are interested in the probability of success to distinguish the two distinct distributions, we compute the probability of error $\mathrm{Pr}_{\mathrm{err}}$ as

$$\mathrm{Pr}_{\mathrm{err}} \stackrel{\mathrm{def}}{=} \Pr\left(N\left(\mathbf{x}^{\mathbf{1}}\right) < N\left(\tilde{\mathbf{x}}^{\mathbf{1}}\right)\right) = \Pr\left(N\left(\mathbf{x}^{\mathbf{1}}\right) - N\left(\tilde{\mathbf{x}}^{\mathbf{1}}\right) < 0\right).$$

Assuming independence of $N(\mathbf{x}^{\mathbf{1}})$ and $N(\tilde{\mathbf{x}}^{\mathbf{1}})$, we expect that $N(\mathbf{x}^{\mathbf{1}}) - N(\tilde{\mathbf{x}}^{\mathbf{1}})$ asymptotically complies with the normal distribution $\mathcal{N}(\frac{\zeta\gamma^w}{2}, \sqrt{\frac{\zeta}{2}})$. We have

$$\mathrm{Pr}_{\mathrm{err}} \approx \Phi\left(-\frac{\frac{\zeta\gamma^w}{2}}{\sqrt{\frac{\zeta}{2}}}\right) = \Phi\left(-\frac{\sqrt{2\zeta}}{2} \cdot \gamma^w\right),$$

where Φ is the standard normal distribution. Thus we estimate the rank of $N(\mathbf{x}^{\mathbf{1}})$ among all $N(\tilde{\mathbf{x}}^{\mathbf{1}})$ in ascending order by

$$\mathrm{E}\left[\mathrm{Rank}_{N(\mathbf{x}^{\mathbf{1}})}\right] = (2^{L_1} - 1) \cdot \mathrm{Pr}_{\mathrm{err}} \approx \frac{2^{L_1}}{\gamma^w \sqrt{\pi\zeta}} e^{-\frac{\zeta}{4}\gamma^{2w}}. \tag{13}$$

According to the conventional estimation [11, 27] in correlation attacks, derived by channel coding theory, the critical data complexity ζ_0, on the

order of γ^{-2w}, is $\zeta_0 = \frac{L_1}{1-h(\frac{1}{2}+\frac{1}{2}\gamma^w)} \approx \frac{2L_1 \log 2}{\gamma^{2w}}$, and h is the binary entropy function. Note that this critical data complexity ζ_0 does not guarantee that $N(\mathbf{x^1})$ is the smallest (resp. largest) of all $N(\tilde{\mathbf{x}}^\mathbf{1})$ with positive (resp. negative) γ. According to [11] simulations showed the probability of success is closer to $\frac{1}{2}$ for $\zeta = \zeta_0$. Here, we are interested with a minimum ζ such that the probability of success is closer to 1. Hence, we set $\zeta = k_0\gamma^{-2w}$ for some k_0 to be determined by solving $\mathrm{E}[\mathrm{Rank}_{N(\mathbf{x^1})}] = 1$ in Formula (13). Finally, we obtain that the minimum

$$\zeta \approx \frac{4L_1 \log 2}{\gamma^{2w}} (= 2\zeta_0) \tag{14}$$

is needed to guarantee that $N(\mathbf{x^1})$ is the smallest (resp. largest) of all $N(\tilde{\mathbf{x}}^\mathbf{1})$ with positive (resp. negative) γ. Note that our analysis is consistent with simulation results in [11], which showed that the probability of success is close to 1 for $\zeta = 2\zeta_0$. Clearly, our problem of recovering R_1 right fits into the Maximum Likelihood Decoding (MLD) problem for a general linear code, as described in Section 6. Thus, solving MLD problem allows to recover $\mathbf{r}$, after which we apply linear transform to solve $\mathbf{x^1}$.

6 A Maximum Likelihood Decoding Algorithm

We first recall the following basics of linear codes (see [36] for details). Given a matrix $G_{L\times\kappa}$ (with $L < \kappa$), for every message $r = (r_1, \ldots, r_L)$, define the codeword $x = (x_1, \ldots, x_\kappa) \stackrel{\text{def}}{=} rG$. The set of all codewords form the linear code, defined by G. The code is said to have dimension L, length κ and generator matrix G. The MLD problem for the linear code is: find the message r which minimizes the Hamming distance[12] between the associated codeword x and the received vector $s = (s_1, \ldots, s_\kappa)$, i.e. find such r that minimizes $N(r) = \sum_{t=1}^{\kappa}(s_t \oplus x_t)$, where $x_t = rG_t$ (G_t denotes the t-th column vector of G).

For example, our preceding key-recovery attack in Section 5 can be transformed into the MLD problem as follows. Define the column vector G_t of the generator matrix G by $G_t = (a_0, \ldots, a_{L_1-1})^\top$, where $a_0 + a_1 x + \cdots + a_{L_1-1}x^{L_1-1} = x^t \mod p_1(x)$. And let $L = L_1$, $\kappa = \zeta$, $r = \mathbf{r}$, $x = \{r_t\}$ and $s = \{s_t\}$.

[12] The Hamming distance between two vectors $x = (x_1, \ldots, x_\ell)$ and $y = (y_1, \ldots, y_\ell)$ of equal dimension is the number of coordinates where they differ.

6.1 The Time-domain Analysis

The trivial solution to find r is an exhaustive search in the time-domain: for every message $\tilde{r}$, we compute $N(\tilde{r})$ and keep the smallest. The final record leads to r. The time complexity is $O(\kappa \cdot 2^L)$ with memory κ bits.

6.2 The Frequency-domain Analysis

We introduce an integer-valued function,

$$\mathcal{W}(x) \stackrel{\text{def}}{=} \sum_{1 \le t \le \kappa : G_t = x^\top} (-1)^{s_t},$$

for all $x \in GF(2)^L$, where $\top$ denotes the matrix transpose. We compute the Walsh transform $\widehat{\mathcal{W}}$ of $\mathcal{W}$ as follows:

$$\begin{aligned}
\widehat{\mathcal{W}}(r) &= \sum_{x \in GF(2)^L} (-1)^{r \cdot x} \mathcal{W}(x) \\
&= \sum_{t=1}^{\kappa} (-1)^{s_t \oplus r G_t} \\
&= \sum_{t=1}^{\kappa} (-1)^{s_t \oplus x_t} \\
&= \kappa - 2N(r).
\end{aligned}$$

We thereby reach the theorem below.

Theorem 9.

$$N(r) = \frac{1}{2}\left(\kappa - \widehat{\mathcal{W}}(r)\right),$$

for all $r \in GF(2)^L$.

This generalizes the result [36, p. 414] of a special case when $\kappa = 2^L$ and $G_t^\top$ corresponds to the binary representation of t. So, to solve the MLD problem, we just compute $\mathcal{W}$, perform FWT (see [54]), and find the maximum $\widehat{\mathcal{W}}(r)$ as shown in Algorithm 1.

The time and memory complexities of FWT are $O(L \cdot 2^L)$, $O(2^L)$ respectively. Since the precomputation of $\mathcal{W}$ takes time $O(\kappa)$ with memory $O(\kappa)$, we conclude that the improved MLD algorithm runs in $O(\kappa + L \cdot 2^L)$ with memory $O(2^L)$ (additionally, using linear transformation allows to compute FWT over $GF(2)^k$ with memory $O(2^k)$ where $k = \lceil \log_2 \kappa \rceil$). Note that when $\kappa \ge 2^L$, the time complexity corresponds to $O(\kappa)$, which

Algorithm 1 The frequency transformation algorithm

Inputs:
  $G = (G_1, \ldots, G_\kappa)$: the generator matrix
  keystream $s_1 s_2 \cdots s_\kappa$
Preprocessing:
  for all L-bit r **do**
    compute $\mathcal{W}(r)$ and keep in memory
  end for
Processing:
  use FWT to compute $\widehat{\mathcal{W}}$
  find r that achieves the maximal $\widehat{\mathcal{W}}(r)$
  output r

is optimal in the sense that it stands on the same order of magnitude as the data complexity does. Table 3 compares the original exhaustive search algorithm with the improved frequency transformation algorithm. Note that the technique of FWT was used in another context [12] to speed up other kinds of fast correlation attacks. In the case of the core E0 (see Section 7), we will see how it helps to speed up the attack [16] by a factor of 2^{24}. We estimate similar correlation attacks like [11] can be speeded up by a factor of 10; undoubtedly, some other attacks can be significantly improved by our FWT-based algorithm as well.

Table 3. Comparison of maximum likelihood decoding algorithms

	time	memory
Exhaustive Search	$\kappa \cdot 2^L$	κ
Frequency Transformation	$\kappa + L \cdot 2^L$	$\min(\kappa, 2^L)$

6.3 A More Generalized MLD Algorithm

We further generalize the preceding problem by finding the L-bit vector r such that given a sequence of ℓ-bit ($\ell < L$) vectors $S_1, \ldots, S_\tau$ and $f : GF(2)^\ell \to \mathbf{R}$ together with matrices $G_1, \ldots, G_\tau$ of size L by ℓ, the sequence of ℓ-bit vectors $X_1, \ldots, X_\tau$ defined by $X_t = rG_t$ minimizes $N(r) = \sum_{t=1}^{\tau} f(S_t \oplus X_t)$. It means the linear code has length $\tau\ell$, dimension L, and the generator matrix $G = (G_1, \ldots, G_\tau)$. Note that our previous problem in Section 6.2 is merely a special case of $\ell = 1$, $\tau = \kappa$ and $f(a) = a$ for $a \in GF(2)$.

Define a real function

$$\mathcal{W}(x) = \frac{1}{2^\ell} \sum_{1 \le t \le \tau, a \in GF(2)^\ell : aG_t^\top = x} (-1)^{a \cdot S_t} \widehat{f}(a),$$

for all $x \in GF(2)^L$. We compute the Walsh transform $\widehat{\mathcal{W}}$ of $\mathcal{W}$ as follows:

$$\begin{aligned}
\widehat{\mathcal{W}}(r) &= \sum_{x \in GF(2)^L} (-1)^{r \cdot x} \mathcal{W}(x) \\
&= \frac{1}{2^\ell} \sum_{t=1}^{\tau} \sum_{a \in GF(2)^\ell} (-1)^{a \cdot (rG_t \oplus S_t)} \widehat{f}(a) \\
&= \sum_{t=1}^{\tau} f(rG_t \oplus S_t) \\
&= N(r).
\end{aligned}$$

Algorithm 2 directly follows above computation. The total running time of our algorithm is $O(\tau \ell L 2^\ell + L 2^L)$ with memory $O(2^L)$. To speed up the computation of $\mathcal{W}$, we could precompute the inner products of all pairs of ℓ-bit vectors in time $O(2^{2\ell})$ with memory $O(2^{2\ell})$. Thus, the total running time of the algorithm is $O(2^{2\ell} + \tau L 2^\ell + L 2^L)$ with memory $O(2^{2\ell} + 2^L)$.

Algorithm 2 The generalized MLD algorithm

Parameters:
  f, ℓ
Inputs:
  $G = (G_1, \ldots, G_\tau)$: the generator matrix
  vector stream $S_1, S_2, \cdots, S_\tau$
Processing:
  apply FWT to compute the table of $\widehat{f}$
  initialize the table of $\mathcal{W}$ to 0
  for all ℓ-bit a **do**
    for $t = 1, \ldots, \tau$ **do**
      increment $\mathcal{W}(aG_t^\top)$ by $\frac{1}{2^\ell}(-1)^{a \cdot S_t} \widehat{f}(a)$
    end for
  end for
  use FWT to compute $\widehat{\mathcal{W}}$
  find r that achieves the minimal $\widehat{\mathcal{W}}(r)$
  output r

In the special case that $G_{t+1} = AG_t$ for $t = 1, \ldots, \tau$, we precompute another table to map any L-bit vector x to $xA^\top$. It takes time $O(2^L)$

with memory $O(2^L)$. The total time of the algorithm is thus $O(2^{2\ell} + (L + \tau)2^\ell + L2^L)$, with memory $O(2^{2\ell} + 2^L)$. Note that above special case is applicable to the core E0 (see Section 7).

6.4 Comments

According to [5], the general decoding problem for linear codes is shown to be NP-complete (see [21] for definition) in the sense that the known deterministic algorithm that decodes an arbitrary linear code with dimension L and length κ performs an exhaustive trial on all possible codewords. Thus, prior to us, the best deterministic decoding algorithm takes time $O(2^L \times \kappa)$. In our work, we showed that the decoding time $O(L \cdot 2^L + \kappa)$ is achievable and it grows linear in κ. This makes it possible now to decode the linear code with not so large dimension but very large length in which case the naive exhaustive decoding is infeasible.

7 Case Study: the Core of Bluetooth E0

7.1 Description

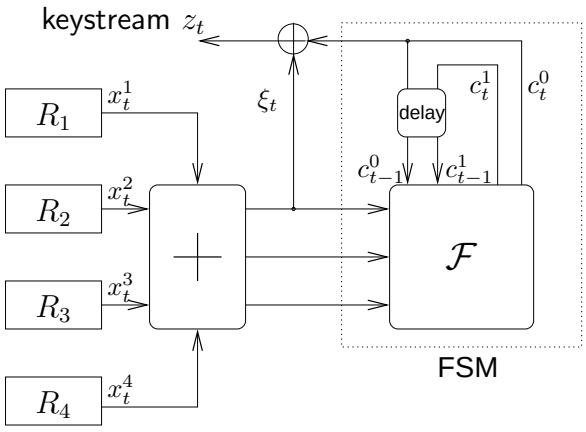

Fig. 2. Outline of the core E0

Specified in [6], the core keystream generator E0 (Fig. 2) used in Bluetooth fits in the model in Section 2: $n = 4$, $L_1 = 25$, $L_2 = 31$, $L_3 = 33$, $L_4 = 39$ (thus $L = 128$) with primitive characteristic polynomials

$$p_1(x) = x^{25} + x^{17} + x^{13} + x^5 + 1,$$
$$p_2(x) = x^{31} + x^{19} + x^{15} + x^7 + 1,$$
$$p_3(x) = x^{33} + x^{29} + x^9 + x^5 + 1,$$
$$p_4(x) = x^{39} + x^{35} + x^{11} + x^3 + 1,$$

respectively. The state σ_t of the FSM contains (c_{t-1}, c_t) of k bits, where $k = 4$ and $c_t = (c_t^1, c_t^0)$ has 2 bits. Let $w(x_t) \stackrel{\text{def}}{=} \sum_{i=1}^{4} x_t^i$ be the Hamming weight[13] of x_t. The FSM has the update function $\mathcal{F} : (w(x_t), c_{t-1}, c_t) \mapsto (c_t, c_{t+1})$. Computing c_{t+1} from σ_t can be described by

$$c_{t+1}^1 = \upsilon_{t+1}^1 \oplus c_t^1 \oplus c_{t-1}^0,$$
$$c_{t+1}^0 = \upsilon_{t+1}^0 \oplus c_t^0 \oplus c_{t-1}^1 \oplus c_{t-1}^0,$$

where the 2-bit $\upsilon_{t+1} = (\upsilon_{t+1}^1, \upsilon_{t+1}^0)$ is defined by

$$\upsilon_{t+1} = \left\lfloor \frac{w(x_t) + 2 \cdot c_t^1 + c_t^0}{2} \right\rfloor.$$

Table 4 shows the state transition of the FSM, where the four-bit state is represented in the quaternary system (e.g. the FSM changes from $\sigma_t = 13$ into $\sigma_{t+1} = 32$ by the input $w(x_t) = 2$). One can check Table 4 by above equations.

Table 4. State transition of σ_{t+1} given $w(x_t)$ and σ_t

		σ_t																
		00	01	02	03	10	11	12	13	20	21	22	23	30	31	32	33
$w(x_t)$	0	00	11	23	32	03	12	20	31	01	10	22	33	02	13	21	30
	1	00	10	23	31	03	13	20	32	01	11	22	30	02	12	21	33
	2	01	10	20	31	02	13	23	32	00	11	21	30	03	12	22	33
	3	01	13	20	30	02	10	23	33	00	12	21	31	03	11	22	32
	4	02	13	21	30	01	10	22	33	03	12	20	31	00	11	23	32

With $\varpi = 01$ in Eq.(2), at each clock cycle t, the FSM emits one bit $\psi_t = c_t^0$. The keystream output bit is $z_t = x_t^1 \oplus x_t^2 \oplus x_t^3 \oplus x_t^4 \oplus c_t^0$.

7.2 Correlations

From Section 3, we know that if (λ_0, σ_0) is uniformly distributed, then, for $\epsilon \leq 26$ and any $\alpha_1, \ldots, \alpha_\epsilon \in GF(2)^4$, $\delta(\alpha_1, \ldots, \alpha_\epsilon) = \Delta(\alpha_1 \cdot \sigma_t \oplus \cdots \oplus$

[13] Recall that the Hamming weight of a vector is the number of 1's of its coordinates. Note that the Hamming weight of a vector always equals its Hamming distance (defined in Section 6) to the all zero vector of equal dimension.

$\alpha_\epsilon \cdot \sigma_{t+\epsilon-1}$) is a constant and does not depend on t. It can be computed by Theorem 4. However, notice that the core E0 has such a special FSM that the two consecutive states σ_t and σ_{t+1} are half overlapped (i.e. 2-bit c_t is contained in both). Therefore, to compute the value of $\Delta(\alpha_1 \cdot \sigma_0 \oplus \cdots \oplus \alpha_\epsilon \cdot \sigma_{\epsilon-1})$, the sequence $\alpha_1, \ldots, \alpha_\epsilon$ is not unique. So, we resort to another notation Ω for the unique expression of the same thing instead.

For $\epsilon \leq 27$ and any $a_1, \ldots, a_\epsilon \in GF(2)^2$, let $\Omega(a_1, \ldots, a_\epsilon) \stackrel{\text{def}}{=} \Delta(a_1 \cdot c_0 \oplus \cdots \oplus a_\epsilon \cdot c_{\epsilon-1})$. Similarly to Theorem 4, we apply Lemma 3 with $X = x_{\epsilon-2}$, $Y = (c_0, \ldots, c_{\epsilon-2})$, $\Lambda(Y) = (c_{\epsilon-3}, c_{\epsilon-2})$, $\Theta(X, \Lambda(Y)) = a_\epsilon \cdot c_{\epsilon-1}$ and $v = (a_1, \ldots, a_{\epsilon-1})$ and obtain the following result. Assuming (λ_0, σ_0) is uniformly distributed, for any $\epsilon \leq 27$ and $a_1, \ldots, a_\epsilon \in GF(2)^2$, we have

$$\Omega(a_1, \ldots, a_\epsilon) = \sum_{w_0, w_1 \in GF(2)^2} \Omega(w_0, w_1, a_\epsilon) \times \Omega(a_1, \ldots, a_{\epsilon-3}, a_{\epsilon-2} \oplus w_0, a_{\epsilon-1} \oplus w_1).$$

Here is a full list of nonzero triplets:

$$\begin{array}{lll} \Omega(0,0,0) = 1, & \Omega(1,3,2) = \frac{1}{4}, & \Omega(2,3,3) = -\frac{5}{8}, \\ \Omega(1,0,2) = \frac{5}{8}, & \Omega(2,0,3) = \frac{1}{4}, & \Omega(3,3,1) = -\frac{1}{4}. \end{array}$$

With the list, we computed all ϵ-tuple biases for $\epsilon \leq 27$ and found out that the largest two biases are $\Omega(1,1,1,1,1) = -\frac{25}{256}$ and $\Omega(1,0,0,0,0,1) = \frac{25}{256}$. Both biased were mentioned in [17, 24] without formal proof. Below we give formal proof on the two biases.

Property 10. Assuming (λ_t, σ_t) is random and uniformly distributed, we have

$$\Pr(c_t^0 \oplus c_{t+1}^0 \oplus c_{t+2}^0 \oplus c_{t+3}^0 \oplus c_{t+4}^0 = 1) = \frac{1}{2} + \frac{25}{512}.$$

Proof. We show the equivalent $\Omega(1,1,1,1,1) = -\frac{25}{256}$ as follows:

$$\begin{aligned} \Omega(1,1,1,1,1) &= \Omega(3,3,1) \cdot \Omega(1,1,1 \oplus 3, 1 \oplus 3) \\ &= -\frac{1}{4}\Omega(1,1,2,2) \\ &= -\frac{1}{4} \sum_{w_0, w_1} \Omega(w_0, w_1, 2) \cdot \Omega(1, 1 \oplus w_0, 2 \oplus w_1) \\ &= -\frac{1}{4} \left(\Omega(1,0,2)\Omega(1,1 \oplus 1,2) + \Omega(1,3,2)\Omega(1,1 \oplus 1, 2 \oplus 3)\right) \\ &= -\frac{1}{4} \left(\Omega^2(1,0,2) + \Omega(1,3,2)\Omega(1,0,1)\right) \\ &= -\frac{25}{256}. \end{aligned}$$

□

Remark 11. Assuming $w(x_t) = 2$ holds for $t = t_0, t_0 + 1, t_0 + 2$, then, regardless of the value of σ_{t_0}, we always have

$$c^0_{t_0} \oplus c^0_{t_0+1} \oplus c^0_{t_0+2} \oplus c^0_{t_0+3} \oplus c^0_{t_0+4} = 1.$$

Since $\Pr(w(x_t) = 2) = \frac{6}{16}$, this seems to suggest that

$$\Pr(c^0_t \oplus c^0_{t+1} \oplus c^0_{t+2} \oplus c^0_{t+3} \oplus c^0_{t+4} = 1) \approx \frac{1}{2} + (\frac{6}{16})^3 = \frac{1}{2} + \frac{27}{512},$$

which explains the bias in Property 10. This special case was not pointed out in [17, 24] however.

Property 12. Assuming (λ_t, σ_t) is random and uniformly distributed, we have

$$\Pr(c^0_t = c^0_{t+5}) = \frac{1}{2} + \frac{25}{512}.$$

Proof. This bias is similarly proved from $\Omega(1, 0, 0, 0, 0, 1) = \frac{25}{256}$. □

Throughout the rest of the paper, we let

$$\gamma = \Omega(1, 0, 0, 0, 0, 1) = -\Omega(1, 1, 1, 1, 1) = \frac{25}{256}.$$

Besides the above two largest biases, we have the only second largest bias up to 27 bits $\Omega(1, 0, 1, 1) = -2^{-4}$. This bias was already proved in [26]. Now, we apply Theorem 6 in Section 3 to compute the theoretical upper bound of $\Omega(a)$ for any a of at most 27 tuples and compare γ with it. To show this, we first list the state transition matrix U (where dashed entries denote zeros) as follows.

$$U = \begin{bmatrix}
\frac{5}{16} & \frac{10}{16} & \frac{1}{16} & - & - & - & - & - & - & - & - & - & - & - & - & - \\
- & - & - & - & \frac{10}{16} & \frac{1}{16} & - & \frac{5}{16} & - & - & - & - & - & - & - & - \\
- & - & - & - & - & - & - & - & \frac{10}{16} & \frac{1}{16} & - & \frac{5}{16} & - & - & - & - \\
- & - & - & - & - & - & - & - & - & - & - & - & \frac{5}{16} & \frac{10}{16} & \frac{1}{16} & - \\
- & \frac{1}{16} & \frac{10}{16} & \frac{5}{16} & - & - & - & - & - & - & - & - & - & - & - & - \\
- & - & - & - & \frac{5}{16} & - & \frac{1}{16} & \frac{10}{16} & - & - & - & - & - & - & - & - \\
- & - & - & - & - & - & - & - & \frac{5}{16} & - & \frac{1}{16} & \frac{10}{16} & - & - & - & - \\
- & - & - & - & - & - & - & - & - & - & - & - & \frac{1}{16} & \frac{10}{16} & \frac{5}{16} & \\
\frac{10}{16} & \frac{5}{16} & - & \frac{1}{16} & - & - & - & - & - & - & - & - & - & - & - & - \\
- & - & - & - & \frac{1}{16} & \frac{10}{16} & \frac{5}{16} & - & - & - & - & - & - & - & - & - \\
- & - & - & - & - & - & - & \frac{1}{16} & \frac{10}{16} & \frac{5}{16} & - & - & - & - & - & - \\
- & - & - & - & - & - & - & - & - & - & - & \frac{10}{16} & \frac{5}{16} & - & \frac{1}{16} & \\
\frac{1}{16} & - & \frac{5}{16} & \frac{10}{16} & - & - & - & - & - & - & - & - & - & - & - & - \\
- & - & - & - & - & \frac{5}{16} & \frac{10}{16} & \frac{1}{16} & - & - & - & - & - & - & - & - \\
- & - & - & - & - & - & - & - & \frac{5}{16} & \frac{10}{16} & \frac{1}{16} & - & - & - & - & - \\
- & - & - & - & - & - & - & - & - & - & - & \frac{1}{16} & - & \frac{5}{16} & \frac{10}{16} &
\end{bmatrix}$$

From U, we notice that $|\sum_{b:\varpi\cdot b=1} U_{ab} - \sum_{b:\varpi\cdot b=0} U_{ab}|$ remains a constant $\rho_0 = \frac{4}{16} = 2^{-2}$ for all a. Hence $\rho = \rho_0 = 2^{-2}$. Consequently, applying Theorem 6, we know

$$|\Omega(a)| \leq 2^{-2},$$

for any a of at most 27 tuples. We check that $\gamma \approx 2^{-3.36} < 2^{-2}$.

7.3 Keystream Distinguishers

We are ready to build a distinguisher for the core E0 upon above largest correlations together with the multiple $Q(x)$ of $\prod_{i=1}^{4} p_i(x)$ with degree d and weight w, which can be precomputed by birthday paradox as mentioned in Section 4.2 or easy manual calculation as follows:

Examples of $Q(x)$ with Weight Four Recall that $\theta_i = 2^{L_i} - 1$ is the order of $p_i(x)$ for $i = 1, 2, 3, 4$. By definition, $p_i(x)|x^{\theta_i} + 1$. On the other hand, $p_i(x)p_j(x)|\mathrm{lcm}(x^{\theta_i} + 1, x^{\theta_j} + 1) = x^{\mathrm{lcm}(\theta_i,\theta_j)} + 1$ for $i \neq j$, hence we deduce the following three multiple polynomials of $p(x)$ with weight 4 with ease:

$$\begin{aligned}
Q_1(x) &= (x^{\mathrm{lcm}(\theta_1,\theta_2)} + 1)(x^{\mathrm{lcm}(\theta_3,\theta_4)} + 1), \\
Q_2(x) &= (x^{\mathrm{lcm}(\theta_1,\theta_3)} + 1)(x^{\mathrm{lcm}(\theta_2,\theta_4)} + 1), \\
Q_3(x) &= (x^{\mathrm{lcm}(\theta_1,\theta_4)} + 1)(x^{\mathrm{lcm}(\theta_2,\theta_3)} + 1),
\end{aligned}$$

where

$\mathrm{lcm}(\theta_1, \theta_2) = 2^{56} - 2^{31} - 2^{25} + 1$, $\mathrm{lcm}(\theta_1, \theta_3) = 2^{58} - 2^{33} - 2^{25} + 1$,
$\mathrm{lcm}(\theta_1, \theta_4) = 2^{64} - 2^{39} - 2^{25} + 1$, $\mathrm{lcm}(\theta_2, \theta_3) = 2^{64} - 2^{33} - 2^{31} + 1$,
$\mathrm{lcm}(\theta_2, \theta_4) = 2^{70} - 2^{39} - 2^{31} + 1$, $\mathrm{lcm}(\theta_3, \theta_4) = (2^{39} - 1) \sum_{i=0}^{10} 2^{3i}$.

The degrees of $Q_1(x), Q_2(x), Q_3(x)$ are approximately $2^{69}, 2^{70}, 2^{65}$ respectively. Note that we may also expect optimal multiples with degree on the same order of magnitude and weight 3 from Table 1.

Primary Distinguisher Table 5 summarizes the best performance of our primary (uni-bias-based) distinguisher for the core E0 based on either the use of $Q_3(x)$ with weight 4, or a search of $Q(x)$, when we choose $\alpha = (1, 1, 1, 1, 1)$ or $(1, 0, 0, 0, 0, 1)$.

Table 5. Summary of the best primary distinguisher for the core E0

type		d	w	precomputation	data	time
use $Q(x) = Q_3(x)$		2^{65}	4	-	2^{65}	
find $Q(x)$ with	minimal d	2^{33}	5	2^{66}	2^{34}	
	tradeoff	2^{43}	5	2^{45}	2^{43}	

Advanced Distinguisher From Section 4.4, we know that the multi-bias-based distinguisher improves the uni-bias-based one only when the patterns of the largest correlation coefficients are linearly dependent, which happens to be true in the core E0: recall from Property 10 and Property 12 that the 6-tuple patterns of the three largest biases satisfy the linear relation,

$$(1, 1, 1, 1, 1, 0) \oplus (0, 1, 1, 1, 1, 1) = (1, 0, 0, 0, 0, 1).$$

As a simple solution we may just pick $\nu = 6$, $\ell = 2$, $J_1 = (1, 1, 1, 1, 1, 0)$ and $J_2 = (0, 1, 1, 1, 1, 1)$ (where J_i is the i-th row of J), then we obtain $\mu = 3$. And the data complexity Ξ is reduced to a factor of $\frac{2}{3}$ for negligible d. Indeed, recall that we proved by computation that the largest Walsh coefficient for $\nu \leq 27$ are either $(0, \ldots, 0, 1, 1, 1, 1, 1, 0, \ldots, 0)$ or $(0, \ldots, 0, 1, 0, 0, 0, 0, 1, 0, \ldots, 0)$. Thus $\mu \leq (\nu - 4) + (\nu - 5) = 2\nu - 9$. This leads to a more general solution, if we pick $\nu = \ell + 4$, and the i-th row of J as

$$J_i = (\underbrace{0, \ldots, 0}_{i-1 \text{ zeros}}, 1, 1, 1, 1, 1, \underbrace{0, \ldots, 0}_{\nu-i-4 \text{ zeros}}) \text{ for } i = 1, \ldots, \ell, \qquad (15)$$

then we obtain $\mu = 2\ell - 1$. And so the improved factor $\frac{\ell}{2\ell-1}$ of data complexity Ξ tends to $\frac{1}{2}$ for negligible d when ℓ goes to infinity; however, because of the underlying assumption for the core E0, ν is restricted to no larger than 27, i.e. $\ell \leq 23$. To conclude, we show that the modified distinguisher (Algorithm 3) needs data complexity

$$\Xi \approx \frac{\ell}{2\ell - 1} \cdot \gamma^{-2w} + d, \text{ for } 1 \leq \ell \leq 23. \tag{16}$$

Table 6 shows the best improvement achieved with $\ell = 23$. We see that the minimum Ξ drops from previous 2^{34} to 2^{33}.

Algorithm 3 The advanced distinguisher for the core E0

Parameters:
$\ell \in [1, 23], \nu = \ell + 4$
$J : GF(2)^{\nu} \to GF(2)^{\ell}$ defined in (15)
$\mathcal{D}_A$: the probability distribution of the ℓ-bit vector A_t
$Q(x) = \sum_{i=1}^{w} x^{q_i}$: the multiple polynomial of $p_1(x)p_2(x)p_3(x)p_4(x)$ with degree d
Ξ: the sample size by Eq.(16)
Inputs:
keystream $z_0 z_1 \cdots z_{\Xi-1}$ of either a truly random source $\mathcal{S}_0$ or the output $\mathcal{S}_1$ generated by the core E0
initialize counters $u_0, u_1, \ldots, u_{2^{\ell}-1}$
for $t = 0, 1, \ldots, \lfloor \frac{\Xi - d - 4}{\ell} \rfloor - 1$ **do**
  compute $b = \bigoplus_{i=1}^{w} J(z_{\ell t + q_i}, \cdots, z_{\ell t + q_i + \nu - 1})$
  increment u_b
end for
if $\sum_b u_b \cdot \log\left(2^{\ell} \cdot \mathcal{D}_A^{\otimes w}(b)\right) > 0$ **then**
  accept $\mathcal{S}_1$ as the source
else
  accept $\mathcal{S}_0$ as the source
end if

Table 6. Data complexity Ξ of the advanced distinguisher for the core E0

d	L	247	458	855	1749	2387	2^{18}	2^{23}	2^{27}	2^{33}	2^{44}	2^{65}	2^{32}	2^{43}
w	49	31	24	20	17	16	9	7	6	5	4	3	9	5
$\log_2 \Xi$	328	208	161	134	114	107	60	46	40	33	44	65	60	43

7.4 The Key-recovery Attack

Here we consider the key-recovery attack of how to reconstruct the initial states of the LFSRs for the core E0. Let $Q(x) = \sum_{i=1}^{w} x^{q_i}$ be the multiple polynomial of $\prod_{i=2}^{4} p_i(x)$ with degree d and weight w. $Q(x)$ can be found with (precomputation) complexity PC by techniques in Section 4.2. Table 7 lists the corresponding triplets (w, d, PC) for small w. As detailed in Section 5, we use the MLD algorithm in Section 6.2 to recover $\mathbf{x^1}$. Table 8 shows our estimated minimal ζ corresponding to w by Eq.(14). Moreover, we conduct the same analysis as in Section 7.3 to decrease ζ by a factor of $\frac{\ell}{2\ell-1}$ for $1 \leq \ell \leq 23$; and we apply the technique introduced in Section 6.3 to obtain the time complexity $O(\Xi + \theta_1 \cdot 2^{\ell} + L_1 \cdot 2^{L_1})$, where $\Xi = \zeta + d$. The attack complexities to recover R_1 for the core E0 are listed in Table 9 for two best cases denoted by A and B, where we choose $\ell = 12$.

Table 7. Complexity PC of finding the multiple of $p_2(x)p_3(x)p_4(x)$ with degree d and weight w

	birthday problem				
	with minimal d				tradeoff
weight w	5	4	3	2	5
degree d	2^{27}	2^{36}	2^{52}	2^{100}	$2^{34.3}$
precomputation PC	2^{54}	2^{54}	2^{52}	-	$2^{36.3}$

Table 8. The estimated minimal ζ corresponding to w by Eq.(14) where $L_1 = 25$, $\gamma = 25/256$

w	5	4	3	2	1
ζ	2^{40}	2^{33}	2^{27}	2^{20}	2^{14}

Table 9. Summary of primary partial key-recovery attacks against R_1 for the core E0

	w	d	ζ	data Ξ	precomputation PT	time	memory
Attack A	5	$2^{34.3}$	2^{39}	2^{39}	$2^{36.3}$	2^{39}	2^{25}
Attack B	4	2^{36}	2^{33}	2^{36}	2^{54}	2^{36}	2^{25}

Once we recover R_1, we target R_2 next based on multiple of $p_3(x)p_4(x)$. Last, we use the technique of guess and determine in [19] to solve R_3 and R_4 with knowledge of the shortest two LFSRs. The detailed complexities of each step are shown in Table 10. A comparison of our attacks with the similar attack[14] [16] and the best attacks [14,25] (both were algebraic attacks) is shown in Table 11 for Case A and B.

Table 10. Detailed complexities of our key-recovery attack against the core E0

	w	d	ζ	data Ξ	precomputation PT	time	memory
R_1	5	$2^{34.3}$	2^{39}	2^{39}	$2^{36.3}$	2^{39}	2^{25}
R_2	3	2^{36}	2^{27}	2^{36}	2^{37}	2^{36}	2^{27}
R_3 and R_4	-	-	-	76	-	2^{33}	-
total	-	-	-	2^{39}	2^{37}	2^{39}	2^{27}

Table 11. Complexities comparison of our attacks with the similar attack [16] and the best attacks [14, 25]

		precomputation	time	data	memory
Algebraic attack	[14, 25]	2^{37}	2^{49}	$2^{23.4}$	2^{37}
Similar attack	[16]	2^{54}	2^{63}	2^{34}	2^{34}
Our attacks	A	2^{37}	2^{39}	2^{39}	2^{27}
	B	2^{54}	2^{37}	2^{36}	2^{27}

Experimental Results with $\mathbf{w = 1}$ We did the small-scale experiment to verify our analysis in Section 5 on the keystream $\{\bigoplus_{i=2}^{4}(x_t^i \oplus z_t)\}$ instead of $\{z_t\}$ to save the trouble of searching the multiple $Q(x)$ of $\prod_{i=2}^{4} p_i(x)$ with low weight (herein $w = 1$). First, we test the rank of $N(\mathbf{x^1})$ among those of all the 2^{L_1} values of $N(\tilde{\mathbf{x}}^\mathbf{1})$ (see Eq.(12) for definition) for a total of 100 randomly chosen initial states of the core E0. From Eq.(13), we have $\mathrm{E}[\mathrm{Rank}_{N(\mathbf{x^1})}] = 1$ for $\zeta = 2^{14}$. It turned out that $N(\mathbf{x^1})$ ranks uniquely the top without exception.

Second, we choose some random $\mathbf{x^1}$, then compute the corresponding average and variance of $\frac{N(\tilde{\mathbf{x}}^\mathbf{1})}{\zeta}$ over all $\tilde{\mathbf{x}}^\mathbf{1} \neq \mathbf{x^1}$ individually, it turned out

[14] The estimate of data complexity in [16] uses a different heuristic formula than ours. However we believe that their estimate and ours in Attack B are essentially the same.

that $\mathrm{Var}(\frac{N(\tilde{\mathbf{x}}^{\mathbf{1}})}{\zeta}) \approx 1.526 \times 10^{-5}$, approximately the same as the expected $\mathrm{Var}(\frac{N(\tilde{\mathbf{x}}^{\mathbf{1}})}{\zeta}) = \frac{1}{\zeta^2}\mathrm{Var}(N(\tilde{\mathbf{x}}^{\mathbf{1}})) = \frac{1}{4\zeta} = 2^{-16} \approx 1.526 \times 10^{-5}$; and we got a consistent average of 0.5. The left curve in Figure 3 corresponds to the experimental probability distribution of $\frac{N(\tilde{\mathbf{x}}^{\mathbf{1}})}{\zeta}$ for $\tilde{\mathbf{x}}^{\mathbf{1}} \neq \mathbf{x}^{\mathbf{1}}$, where the dotted line represents the central symmetric line.

Last, we accordingly tested the average and variance of $\frac{N(\mathbf{x}^{\mathbf{1}})}{\zeta}$ for 2^{25} random initial states of the core E0. And we got the average of around 0.5488 with variance 2.121×10^{-5} (in contrast to the estimation of average $\frac{281}{512} \approx 0.5488$, variance $2^{-16} \approx 1.526 \times 10^{-5}$ respectively). Its experimental probability distribution is drawn on the right curve of Figure 3. It is worth noticing that the two curves are indeed distinct.

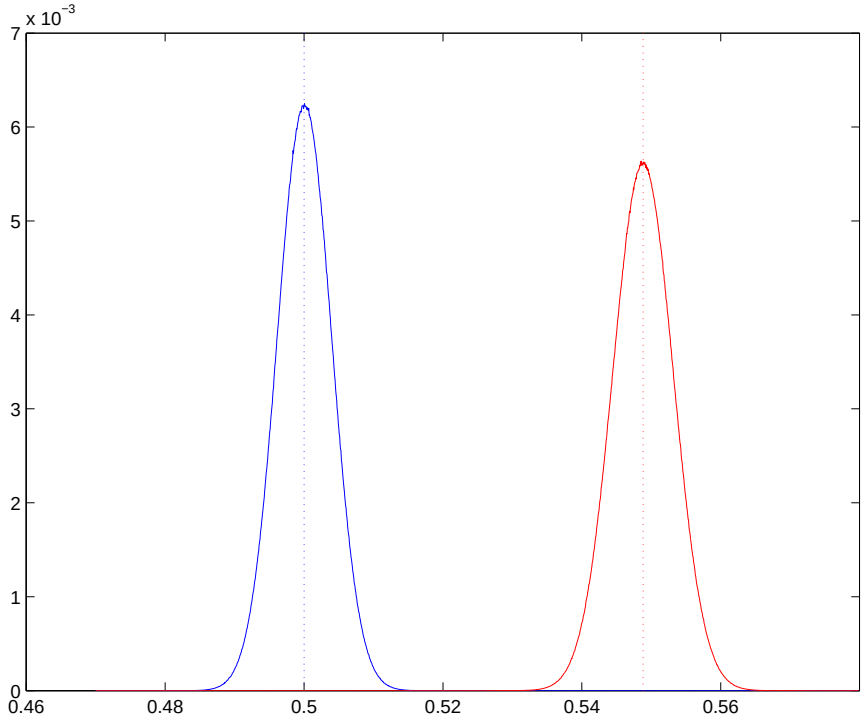

Fig. 3. The two distinct probability distributions of $\frac{N(\tilde{\mathbf{x}}^{\mathbf{1}})}{\zeta}$ for $\tilde{\mathbf{x}}^{\mathbf{1}} \neq \mathbf{x}^{\mathbf{1}}$ (left) and $\tilde{\mathbf{x}}^{\mathbf{1}} = \mathbf{x}^{\mathbf{1}}$ (right).

8 Conclusions

In this paper, we propose an E0-like combiner with memory as a keystream generator. We formulate a systematic computation method to calculate correlations of the FSM output sequences (up to certain bits) by a recursive expression. In addition, we give a upper bound of the correlations, which is useful to the designer. When correlations are found, we can build either a uni-bias-based or multi-bias-based distinguisher to distinguish the keystream produced by the combiner from a truly random sequence. We apply the concept of convolution to the analysis of the multi-bias-based distinguisher that uses all correlations. Based on the theory of [4], it is shown that the multi-bias-based distinguisher outperforms the uni-bias-based distinguisher only when the largest biases are linearly dependent. The keystream distinguisher not only enables the keystream distinguishing attack, but also can upgrade into the key-recovery attack to reconstruct the initial states of the LFSRs. The latter actually reduces to the well-known MLD problem given the keystream long enough (or the bias large enough). By means of FWT, we devise an MLD algorithm to recover the closest codeword for any linear code. It is the best deterministic decoding algorithm known so far.

The analysis principle is successfully applied to the core of Bluetooth encryption algorithm E0 completely. Our key-recovery attack reconstructs the initial states of the LFSRs in 2^{39} time given 2^{39} consecutive keystream bits after $O(2^{37})$ precomputation. This is the best academic key-recovery attack against the core E0 compared with all the attacks [1, 2, 14, 16–19, 24–26, 30, 49] on the core E0. Considering a maximal keystream length of 2745 bits for E0 used in Bluetooth, the attack is impractical. Meanwhile, our proposed MLD algorithm can be easily adapted to speed up a class of fast correlation attacks.

All in all, an ideal nonlinear combiner with memory should satisfy one necessary design principle: the FSM must generate no biased output sequence, i.e.

$$H(\psi_1|\sigma_0) = 1.$$

References

1. Frederik Armknecht. Improving fast algebraic attacks. In B. Roy and W. Meier, editors, *Fast Software Encryption2004*, volume 3017 of *Lecture Notes in Computer Science*, pages 65–82. Springer-Verlag, 2004.
2. Frederik Armknecht and Matthias Krause. Algebraic attacks on combiners with memory. In D. Boneh, editor, *Advances in Cryptology - CRYPTO2003*, volume 2729 of *Lecture Notes in Computer Science*, pages 162–175. Springer-Verlag, 2003.

3. Frederik Armknecht, Matthias Krause, and Dirk Stegemann. Design principles for combiners with memory. In S. Maitra, C. E. V. Madhavan, and R. Venkatesan, editors, *Progress in Cryptology - INDOCRYPT2005*, volume 3797 of *Lecture Notes in Computer Science*, pages 104–117. Springer-Verlag, 2005.
4. Thomas Baignères, Pascal Junod, and Serge Vaudenay. How far can we go beyond linear cryptanalysis? In P. J Lee, editor, *Advances in Cryptology - ASIACRYPT2004*, volume 3329 of *Lecture Notes in Computer Science*, pages 432–450. Springer-Verlag, 2004.
5. Elwyn R. Berlekamp, Robert J. McEliece, and Henk C. A. Van Tilborg. On the inherent intractability of certain coding problems. *IEEE Transactions on Information Theory*, IT-24(3):384–386, May 1978.
6. Bluetooth specification (version 2.0 + EDR), Nov. 2004. available on-line at `http://www.bluetooth.org`.
7. Anne Canteaut and Florent Chabaud. A new algorithm for finding minimum-weight words in a linear code: Application to McEliece's cryptosystem and to narrow-sense BCH codes of length 511. *IEEE Transactions on Information Theory*, 44(1):367–378, Jan. 1998.
8. Anne Canteaut and Eric Filiol. Ciphertext only reconstruction of stream ciphers based on combination generators. In B. Schneier, editor, *Fast Software Encryption2000*, volume 1978 of *Lecture Notes in Computer Science*, pages 165–180. Springer-Verlag, 2001.
9. Anne Canteaut and Michaël Trabbia. Improved fast correlation attacks using parity-check equations of weight 4 and 5. In B. Preneel, editor, *Advances in Cryptology - EUROCRYPT2000*, volume 1807 of *Lecture Notes in Computer Science*, pages 573–588. Springer-Verlag, 2000.
10. Vladimir Chepyzhov and Ben Smeets. On a fast correlation attack on certain stream ciphers. In D. W. Davies, editor, *Advances in Cryptology - EUROCRYPT'91*, volume 547 of *Lecture Notes in Computer Science*, pages 176–185. Springer-Verlag, 1991.
11. Vladimir V. Chepyzhov, Thomas Johansson, and Ben Smeets. A simple algorithm for fast correlation attacks on stream ciphers. In B. Schneier, editor, *Fast Software Encryption2000*, volume 1978 of *Lecture Notes in Computer Science*, pages 181–195. Springer-Verlag, 2001.
12. Philippe Chose, Antoine Joux, and Michel Mitton. Fast correlation attacks: An algorithmic point of view. In L. R. Knudsen, editor, *Advances in Cryptology - EUROCRYPT2002*, volume 2332 of *Lecture Notes in Computer Science*, pages 209–221. Springer-Verlag, 2002.
13. Andrew Clark, Jovan Dj. Golić, and Ed Dawson. A comparison of fast correlation attaks. In D. Gollmann, editor, *Fast Software Encryption'96*, volume 1039 of *Lecture Notes in Computer Science*, pages 145–157. Springer-Verlag, 1996.
14. Nicolas T. Courtois. Fast algebraic attacks on stream ciphers with linear feedback. In D. Boneh, editor, *Advances in Cryptology - CRYPTO2003*, volume 2729 of *Lecture Notes in Computer Science*, pages 176–194. Springer-Verlag, 2003.
15. Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory*. Wiley, 1991.
16. Patrik Ekdahl. *On LFSR Based Stream Ciphers: Analysis and Design*. PhD thesis, Lund Univ., Nov. 2003.
17. Patrik Ekdahl and Thomas Johansson. Some results on correlations in the Bluetooth stream cipher. In *Proceedings of the 10th Joint Conference on Communications and Coding, Austria*, 2000.

18. Scott Fluhrer. Improved key recovery of level 1 of the Bluetooth encryption system, 2002. available on-line at `http://eprint.iacr.org/2002/068`.
19. Scott Fluhrer and Stefan Lucks. Analysis of the E0 encryption system. In S. Vaudenay and A. Youssef, editors, *Selected Areas in Cryptography 2001*, volume 2259 of *Lecture Notes in Computer Science*, pages 38–48. Springer-Verlag, 2002.
20. Réjane Forré. A fast correlation attack on nonlinearly feedforward filtered shift-register sequences. In J. J. Quisquater and J. Vandewalle, editors, *Advances in Cryptology - EUROCRYPT'89*, volume 434 of *Lecture Notes in Computer Science*, pages 586–595. Springer-Verlag, 1990.
21. Michael R. Garey and David S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. Freeman and company, 2000.
22. Jovan Dj. Golić. Correlation properties of a general binary combiner with memory. *Journal of Cryptology*, 9:111–126, 1996.
23. Jovan Dj. Golić. On the security of nonlinear filter generators. In D. Gollmann, editor, *Fast Software Encryption'96*, volume 1039 of *Lecture Notes in Computer Science*, pages 173–188. Springer-Verlag, 1996.
24. Jovan Dj. Golić, Vittorio Bagini, and Guglielmo Morgari. Linear cryptanalysis of Bluetooth stream cipher. In L. R. Knudsen, editor, *Advances in Cryptology - EUROCRYPT2002*, volume 2332 of *Lecture Notes in Computer Science*, pages 238–255. Springer-Verlag, 2002.
25. Philip Hawkes and Gregory G. Rose. Rewriting variables: The complexity of fast algebraic attacks on stream ciphers. In M. Franklin, editor, *Advances in Cryptology - CRYPTO2004*, volume 3152 of *Lecture Notes in Computer Science*, pages 390–406. Springer-Verlag, 2004.
26. Miia Hermelin and Kaisa Nyberg. Correlation properties of the Bluetooth combiner. In J. Song, editor, *Information Security and Cryptology - ICISC'99*, volume 1787 of *Lecture Notes in Computer Science*, pages 17–29. Springer-Verlag, 2000.
27. Thomas Johansson and Frederik Jönsson. Fast correlation attacks based on turbo code techniques. In M. Wiener, editor, *Advances in Cryptology - CRYPTO'99*, volume 1666 of *Lecture Notes in Computer Science*, pages 181–197. Springer-Verlag, 1999.
28. Thomas Johansson and Frederik Jönsson. Improved fast correlation attacks on stream ciphers via convolutional codes. In J. Stern, editor, *Advances in Cryptology - EUROCRYPT'99*, volume 1592 of *Lecture Notes in Computer Science*, pages 347–362. Springer-Verlag, 1999.
29. Thomas Johansson and Frederik Jönsson. Fast correlation attacks through reconstruction of linear polynomials. In M. Bellare, editor, *Advances in Cryptology - CRYPTO2000*, volume 1880 of *Lecture Notes in Computer Science*, pages 300–315. Springer-Verlag, 2000.
30. Matthias Krause. BDD-based cryptanalysis of keystream generators. In L. R. Knudsen, editor, *Advances in Cryptology - EUROCRYPT2002*, volume 2332 of *Lecture Notes in Computer Science*, pages 222–237. Springer-Verlag, 2002.
31. Rudolf Lidl and Harald Niederreiter. *Introduction to Finite Fields and Their Applications*. Cambridge, 1986.
32. Yi Lu. *Applied Stream Ciphers in Mobile Communications*. PhD thesis, EPFL, 2006.
33. Yi Lu, Willi Meier, and Serge Vaudenay. The conditional correlation attack: A practical attack on Bluetooth encryption. In V. Shoup, editor, *Advances in Cryptology - CRYPTO2005*, volume 3621 of *Lecture Notes in Computer Science*, pages 97–117. Springer-Verlag, 2005.

34. Yi Lu and Serge Vaudenay. Cryptanalysis of Bluetooth keystream generator two-level E0. In P. J Lee, editor, *Advances in Cryptology - ASIACRYPT2004*, volume 3329 of *Lecture Notes in Computer Science*, pages 483–499. Springer-Verlag, 2004.
35. Yi Lu and Serge Vaudenay. Faster correlation attack on Bluetooth keystream generator E0. In M. Franklin, editor, *Advances in Cryptology - CRYPTO2004*, volume 3152 of *Lecture Notes in Computer Science*, pages 407–425. Springer-Verlag, 2004.
36. Florence Jessie MacWilliams and Neil J. A. Sloane. *The Theory of Error-correcting Codes*. North-Holland, nineth edition, 1996.
37. Mitsuru Matsui. Linear cryptanalysis method for DES cipher. In T. Helleseth, editor, *Advances in Cryptology - EUROCRYPT'93*, volume 765 of *Lecture Notes in Computer Science*, pages 386–397. Springer-Verlag, 1994.
38. Willi Meier and Othmar Staffelbach. Fast correlation attacks on stream ciphers (extended abstract). In C. Günther, editor, *Advances in Cryptology - EUROCRYPT'88*, volume 330 of *Lecture Notes in Computer Science*, pages 301–314. Springer-Verlag, 1988.
39. Willi Meier and Othmar Staffelbach. Fast correlation attacks on certain stream ciphers. *Journal of Cryptology*, 1(3):159–176, 1989.
40. Willi Meier and Othmar Staffelbach. Correlation properties of combiners with memory in stream ciphers. *Journal of Cryptology*, 5:67–86, Nov. 1992.
41. Alfred J. Menezes, Paul C. van. Oorschot, and Scott A. Vanstone. *Handbook of Applied Cryptography*. CRC, 1996.
42. Miodrag J. Mihaljević, Marc P. C. Fossorier, and Hideki Imai. A low-complexity and high-performance algorithm for the fast correlation attack. In B. Schneier, editor, *Fast Software Encryption2000*, volume 1978 of *Lecture Notes in Computer Science*, pages 196–212. Springer-Verlag, 2001.
43. Miodrag J. Mihaljević, Marc P. C. Fossorier, and Hideki Imai. Fast correlation attack algorithm with list decoding and an application. In M. Matsui, editor, *Fast Software Encryption2001*, volume 2355 of *Lecture Notes in Computer Science*, pages 196–210. Springer-Verlag, 2002.
44. Miodrag J. Mihaljević and Jovan Dj. Golić. A fast iterative algorithm for a shift register initial state reconstruction given the noisy output sequence. In J. Seberry and J. Pieprzyk, editors, *Advances in Cryptology - AUSCRYPT'90*, volume 453 of *Lecture Notes in Computer Science*, pages 165–175. Springer-Verlag, 1990.
45. Miodrag J. Mihaljević and Jovan Dj. Golić. A comparison of cryptanalytic principles based on iterative error-correction. In D. W. Davies, editor, *Advances in Cryptology - EUROCRYPT'91*, volume 547 of *Lecture Notes in Computer Science*, pages 527–531. Springer-Verlag, 1991.
46. Walter T. Penzhorn. Correlation attaks on stream ciphers: Computing low-weight parity checks based on error-correcting codes. In D. Gollmann, editor, *Fast Software Encryption'96*, volume 1039 of *Lecture Notes in Computer Science*, pages 159–172. Springer-Verlag, 1996.
47. Rainer A. Rueppel. *Analysis and Design of Stream Ciphers*. Springer-Verlag, 1986.
48. Rainer A. Rueppel. Correlation immunity and the summation generator. In H. C. Williams, editor, *Advances in Cryptology - CRYPTO'85*, volume 218 of *Lecture Notes in Computer Science*, pages 260–272. Springer-Verlag, 1986.
49. Markku Saarinen. Re: Bluetooth and E0, 2000. posted at `sci.crypt.research`.
50. Thomas Siegenthaler. Decrypting a class of stream ciphers using ciphertext only. *IEEE Transactions on Computers*, C-34(1):81–85, Jan. 1985.
51. Thomas Siegenthaler. Cryptanalysts representation of nonlinearly filtered ML-sequences. In F. Pichler, editor, *Advances in Cryptology - EUROCRYPT'85*, vol-

ume 219 of *Lecture Notes in Computer Science*, pages 103–110. Springer-Verlag, 1986.
52. Serge Vaudenay. An experiment on DES - statistical cryptanalysis. In *Proceedings of the 3rd ACM Conferences on Computer Security*, pages 139–147, 1996.
53. David Wagner. A generalized birthday problem. In M. Yung, editor, *Advances in Cryptology - CRYPTO2002*, volume 2442 of *Lecture Notes in Computer Science*, pages 288–304. Springer-Verlag, 2002.
54. Rao K. Yarlagadda and John E. Hershey. *Hadamard Matrix Analysis and Synthesis with Applications to Communications and Signal/Image Processing*. Kluwer Academic, 1997.
55. Kencheng Zeng and Minqiang Huang. On the linear syndrome method in cryptanalysis. In S. Goldwasser, editor, *Advances in Cryptology - CRYPTO'88*, volume 403 of *Lecture Notes in Computer Science*, pages 469–478. Springer-Verlag, 1990.