J Cryptol (2018) 31:394–433
https://doi.org/10.1007/s00145-017-9260-1

Journal of
CRYPTOLOGY

Practical Cryptanalysis of Bluetooth Encryption with Condition Masking

Bin Zhang

TCA Laboratory, SKLCS, Institute of Software, Chinese Academy of Sciences, Beijing, China
State Key Laboratory of Cryptology, P.O.Box 5159, Beijing 100878, China
University of Chinese Academy of Sciences, Beijing 100049, China
martin_zhangbin@hotmail.com

Chao Xu

State Key Laboratory of Cryptology, P.O.Box 5159, Beijing 100878, China

Dengguo Feng

TCA Laboratory, SKLCS, Institute of Software, Chinese Academy of Sciences, Beijing, China
State Key Laboratory of Cryptology, P.O.Box 5159, Beijing 100878, China

Communicated by Vincent Rijmen.

Received 29 June 2015 / Revised 8 June 2017
Online publication 10 July 2017

Abstract. In this paper, we study the security of a general two-level E0-like encryption model and its instance, the real-world Bluetooth encryption scheme. Both unconditional and conditional correlation properties of the two-level model are investigated in theory and a key-recovery framework based on condition masking, that studies how to choose the condition to get better tradeoffs on the time/memory/data complexity curve, is refined. A novel design criterion to resist the attack is proposed and analyzed. Inspired by these cryptanalytic principles, we describe more threatening and real time attacks on two-level E0. It is shown that only the latest four inputs going into the FSM play the most important role in determining the magnitude of the conditional correlation and the data complexity analysis of the previous practical attacks on two-level E0 are inaccuracy. A new decoding method to improve the data complexity is provided. In the known-IV scenario, if the first 24 bits of 2^{24} frames are available, the secret key can be reliably found with 2^{25} on-line computations, $2^{21.1}$ off-line computations and 4 MB memory. Then, we convert the attack into a ciphertext-only attack, which needs the first 24 bits of 2^{26} frames and all the complexities are under 2^{26}. This is the first practical ciphertext-only attack on the real Bluetooth encryption scheme so far. A countermeasure is suggested to strengthen the security of Bluetooth encryption in practical applications.

Keywords. Stream ciphers, Bluetooth, Two-level E0, Ciphertext-only, Condition masking.

© International Association for Cryptologic Research 2017

1. Introduction

The dedicated hardware-oriented stream ciphers are widely used in the resource constrained environments with limited storage, gate count, or low power supply/consumption, e.g., A5/1 in the GSM and the two-level E0 encryption scheme in Bluetooth. These LFSR-based primitives are designed in the 1990s of last century as the typical examples of irregular clocking generator and the keystream generator with memory.

Bluetooth is a wireless technology standard managed by the Bluetooth Special Interest Group (SIG), whose applications are ubiquitous nowadays, e.g., at home, in hospitals, assembly lines, aircrafts, and wearable computers. The Bluetooth standard, authorized by IEEE 802.15.1 [3], adopts the two-level E0 stream cipher to protect the privacy between different devices, such as personal computers, laptops and mobile phones, that operate over a short range and at low power. Although being a long-standing problem in stream ciphers, the security analysis of two-level E0 is still of great practical importance, as pointed out by Prof. Preneel in [27]. In the latest version of Bluetooth Specification v4.2 [3], the E0 stream cipher is still being used to protect the user information all over the world.

Correlation attack [31] is a classical method in the cryptanalysis of stream ciphers, which exploits some statistically biased relation between the produced keystream and the output of certain underlying sequence. In the 1990s of last century, the correlation properties of combiners with memory are analyzed in theory [9,25]. Based on these identified correlations, for LFSR-based stream ciphers, the initial state of the target LFSR can be recovered by (fast) correlation attacks [4,5,12,13,24]. Further, in [15, 16], the notion of correlation was extended to conditional correlation, that studied the linear correlation of the inputs conditioned on a given output pattern of some nonlinear functions. Later at Crypto 2005 [19], the conditional correlation was assigned with a dual meaning, i.e., the correlation of the output of a function conditioned on some unknown input,[1] which is uniformly distributed and was applied to analyze the security of two-level E0. Since the conditional correlation is no smaller than the unconditional ones, it is expected that better attacks could be achieved if such conditional correlations are exploited appropriately. In the special case that holds for two-level E0, the condition vector is determined linearly by some key-related material and the public nonce, and thus the adversary will get for free the various condition vectors for different target functions corresponding to different values of the nonce and expect to observe the biased sample sequence for the correct key and unbiased sequences for the wrong candidates. Given a pool of sample sequences derived from the guessed values of the condition vector and some public information, a statistical distinguisher can be mounted accordingly to restore the secret key.

The keystream generator E0 used in Bluetooth is a LFSR-based nonlinear combiner with 4-bit memory, which is a modification of the summation generator [28]. In practice, the E0 cipher is frequently re-synchronized as a two-level scheme and the keystream generated for each frame is only 2790 bits.[2] Thus, most of the published

[1] For simplicity, the bit string consisting of the unknown input bits is called the condition vector hereafter.

[2] In the Bluetooth Specification v4.2, the maximum length of keystream is changed from 2745 to 2790.

attacks [1,6,7,11,14,21,29,30] that work on one impractically long frame of keystream remain the academic interest only and have little impact on the practical usage of Bluetooth encryption. Currently, a few attacks [7,8,10,19,20,26] apply to the two-level E0. The best known-IV attack in [19] requires 2^{38} on-line computations, 2^{38} off-line computations and 2^{33} memory to restore the original encryption key, given the first 24 bits of $2^{23.8}$ frames in theory (while in experiments, it needs about 19-, 37-h and 64 GB storage, given the first 24 bits of 2^{26} frames).

Our Contributions In this paper, we first propose a generalized mathematical model that inherits the spirit of the two-level E0 encryption scheme, and study its both unconditional and conditional correlation properties. A fast recursive method with time complexity justification is formulated to compute the unconditional correlations in the general core keystream generator. Besides, the conditional correlation properties of the two-level model are derived and analyzed by the condition masking technique, which instead of considering the correlations conditioned on the whole condition vector, only a subset of the condition vector is taken into account when investigating the correlations. This generalizes the concept of linear mask by depicting the condition as the value selected according to a mask and studying how to choose the condition to achieve better tradeoffs between time/memory/data complexities.

It is expected that with a careful selection of the condition mask, better tradeoffs between the attack complexities can be reached compared to the case of simply choosing the full condition vector. Based on the new notion, a theoretical framework is established to efficiently restore the secret key in the model, which includes the former framework in [19] as a special case. The subtle difference between the new framework and the previous one in [19] is pointed out, which is demonstrated by the concrete attack on the real two-level E0 later. Based on a dedicated linear approximation of the two-level model, both bitwise and vectorial key recovery attacks are mounted and analyzed. During the process, a necessary and sufficient condition that determines when the adversary could gain in correlation by moving from low-dimension to high-dimension in the conditional correlation attack in the general model is proved. Furthermore, a novel design criterion for the general model to achieve desirable security level is proposed as a countermeasure to resist the attack, which is shown to be lightweight and very efficient in practice.

Then under the above cryptanalytic principles, we systematically study the security of the real two-level Bluetooth encryption scheme. Our main observation is that it is of high probability that only a subset of bits in the whole condition vector determines the magnitude of the bias, e.g., in the E0 combiner, only the latest four LFSR bits entering into the FSM play the most important role. Thus, the time/memory complexities of the conditional correlation attack against two-level E0 can be significantly reduced by properly choosing the condition mask.

We start with a revisiting of the unconditional correlation properties in the Bluetooth combiner. Note that the former relevant result, the Corollary 6 in [21], can only compute a special type of unconditional correlations in the core combiner, i.e., the correlations of the pure FSM output sequence. For the correlations between all the input linear functions and all the output linear functions, only a small mask length up to 6-bit is provided in [10]. Here, we present the complete recursive formula for fast computation of such correlations in the E0 combiner, which goes beyond the time/memory complexity barriers of the Fast Walsh Transform (FWT) [18,32] and has a reasonable practical complexity for a wide

range of the length of the linear mask. It is stated in the conclusion section of [10] that the complexity of their attack against E0 can be further decreased by exploiting m-bit linear correlation for $m > 6$ if such correlations are feasible to compute. We efficiently solve this problem by using our method to recursively compute and verify all the unconditional correlations up to 14-bit with a low complexity.

Second, we comprehensively investigate the conditional correlations inside the two-level E0 with the tool of condition masking. The target function inherent in E0 used to compute the conditional correlations in [19] is generalized, and a large class of correlations conditioned on both the linear mask and the condition mask is presented. Although the correlation conditioned on the full condition vector is maximum in the value, it is not generally optimum in the global time/memory/data complexities aspect. The time/memory complexities are closely associated with the condition. An adversary need not to guess the full condition vector and what he has to guess is determined by the condition mask he has chosen. In this way, the time/memory complexities can be considerably reduced.

Third, following the general principles of high-dimensional attacks, the vectorial approach is studied. The vectors used in our attack are carefully constructed and indeed work well to keep the data complexity as low as possible without a penalty in the time/memory complexities. In the process, we point out that the data complexity analysis of the attacks in [19] and [34] are inaccuracy. The exact data complexities in theory of the previous attacks are all above the 2^{26} bound due to an inaccurate formula used in [19] and [34]. We correct the data complexity and show how to reduce it below the 2^{26} bound by a combination with the list decoding[3] and multi-pass decoding techniques,[4] which results in the data complexity reduced to 2^{24}. As a result of all the above techniques, it is shown that if the first 24 bits of 2^{24} frames are available, the secret key can be reliably found with 2^{25} on-line computations, $2^{21.1}$ off-line computations and 4 MB memory in the known-IV scenario. Our attacks have been fully implemented in C language on one core of a single PC. Due to the small memory consumption and low time complexity, it is repeated thousands of times with randomly generated keys and IVs, while the attack in [19] is only executed 30 times for a fixed master key with 2^{26} frames. On average, it takes only a few seconds to restore the original encryption key. To our knowledge, this is the best and most threatening *known-IV* attack on the real Bluetooth encryption scheme so far. Besides, compared to the experimental attack in [34], the success probability of our new attack is improved as well.

Finally, we further convert the above known-IV attack into a ciphertext-only attack against the real two-level E0, based on the fact that in any stretch of written language, certain letters and combinations of letters occur with varying frequencies, i.e., the plaintexts are not random. Thus, we can always find some biases among the plaintext bits. Then, it is shown that if the first 24 bits of 2^{26} frames are available, the secret key can be reliably found with 2^{26} on-line computations and $2^{21.1}$ off-line computations in the ciphertext-only scenario, which is the *first* practical ciphertext-only attack on

[3] The list decoding method means we select a list of candidates other than a unique solution at some step of the attack, please see Sect. 9 for details.

[4] Here, we borrow the idea from [33] in the sense that there are several passes/steps in the attack to identify the correct key, please see Sect. 9 for details.

the two-level bluetooth encryption scheme so far. The practical implementation of the ciphertext-only attack is provided as well. An efficient countermeasure to improve the security of the two-level E0 encryption scheme is summarized to prolong the existence life of the Bluetooth standard in practice.

This paper is organized as follows. We first present some preliminaries used in our work in Sect. 2. Then, the generalized mathematical model of the two-level encryption scheme is provided in Sect. 3. The correlation properties of the two-level model, both unconditional and conditional, are studied in Sect. 4 with the new framework for recovering the secret key in the model. A full description of the real two-level E0 scheme is presented in Sect. 5. In Sect. 6, a brief review of the best previous attack against the two-level E0 is given. Various correlation properties in the E0 combiner, e.g., unconditional and conditional correlations based on condition masking are studied in Sect. 7. Then, both bitwise and vectorial key recovery attacks based on condition masking are developed in Sect. 8 with theoretical analysis. In Sect. 9, the practical implementation of the known-IV attack is described. In Sect. 10, we detail the first ciphertext-only attack on two-level E0, while the practical implementation of the ciphertext-only attack is provided in Sect. 11. Finally, some conclusions are provided in Sect. 12.

2. Preliminaries

In this section, some basic notations and definitions are presented. Denote the binary field by GF(2) and the m-dimensional extension field of GF(2) by GF(2^m). Similarly, denote the m-dimensional vector space over GF(2) by GF(2)^m. The set of real numbers is denoted by $\mathbf{R}$. The inner product of two n-dimensional vectors γ and ρ over GF(2^m) ($m \geq 1$) is $\gamma \cdot \rho =< \gamma, \rho >=< (\gamma_0, \ldots, \gamma_{n-1}), (\rho_0, \ldots, \rho_{n-1}) >= \bigoplus_{i=0}^{n-1} \gamma_i \rho_i$. The Hamming weight of a vector or a polynomial is denoted by $wt(\cdot)$, i.e., the number of nonzero components or coefficients.

Definition 1. The correlation (or bias) of a random Boolean variable X is $\epsilon(X) = \Pr(X = 1) - \Pr(X = 0)$.

Note that in some articles, $\epsilon(X) = \Pr(X = 0) - \Pr(X = 1)$. The only difference is the sign of the correlation. Let ξ be an arbitrary set, given the function $f : \xi \to \mathrm{GF}(2)^r$, the distribution D_f of $f(X)$ with $X \in \xi$ uniformly distributed is

$$D_f(a) = \frac{1}{|\xi|} \sum_{X \in \xi} \mathbf{1}_{f(X)=a}$$

for all $a \in \mathrm{GF}(2)^r$.

Definition 2. The Squared Euclidean Imbalance (SEI) of a distribution D_f is defined as $\Delta(D_f) = 2^r \sum_{a \in GF(2)^r} (D_f(a) - \frac{1}{2^r})^2$.

$\Delta(D_f)$ measures the distance between the target distribution D_f and the uniform distribution. Specially, for $r = 1$, we have $\Delta(D_f) = \epsilon^2(D_f)$. For brevity, we use the

$\epsilon(f)$, $\Delta(f)$ to represent $\epsilon(D_f)$, $\Delta(D_f)$, respectively, hereafter. Similarly, $\mathrm{E}[\Delta(h_{\mathcal{B}})]$ is used to measure the conditional correlations, where the expectation is taken over all the uniformly distributed $\mathcal{B}$. Next, we give the definitions of the Walsh Transform and convolution transform respectively.

Definition 3. Given a function $f : \mathrm{GF}(2)^n \to \mathbf{R}$, for $\omega \in \mathrm{GF}(2)^n$, the Walsh Transform of f at point ω is defined as $\hat{f}(\omega) = \sum_{x \in GF(2)^n} f(x)(-1)^{\langle \omega, x \rangle}$.

Definition 4. Given two functions $f, g : \mathrm{GF}(2)^n \to \mathbf{R}$, the convolution transform of f and g is defined as $(f \otimes g)(x) = \sum_{y \in GF(2)^n} f(y) \cdot g(x \oplus y)$. Further, we have the relation

$$\widehat{(f \otimes g)}(x) = \hat{f}(x) \cdot \hat{g}(x),$$

for all $x \in GF(2)^n$.

It is well known that the Walsh transform of f can be computed efficiently with an algorithm called Fast Walsh Transform (FWT) [32] in $n2^n$ time and 2^n memory. The preparation of f takes 2^n time, and thus the total time complexity is $2^n + n2^n$. The convolution transform between f and g could be computed by invoking three times the FWT algorithm, i.e., $\hat{f}$, $\hat{g}$ and $\widehat{\widehat{f \otimes g}}$.

3. Mathematical Model

Our model of the two-level E0-like encryption scheme is depicted in Fig. 1, which consists of two phases: the payload key generator in the first level and the keystream generator in the second level.

An E0-like keystream generator, as defined in [22], lies at the core of the two-level model. There are n maximum-length LFSRs in the generator, denoted by LFSR_i ($1 \le i \le n$) of length L_i-bit, together with a Finite State Machine (FSM) of k memory bits. Without loss of generality, let the LFSR_is have pairwise distinct lengths L_i satisfying $L_1 < L_2 < \cdots < L_n$ and primitive characteristic polynomials $p_i(x) \in \mathrm{GF}(2)[x]$. Denote the time instant at the first level by t and at the second level by t', respectively.

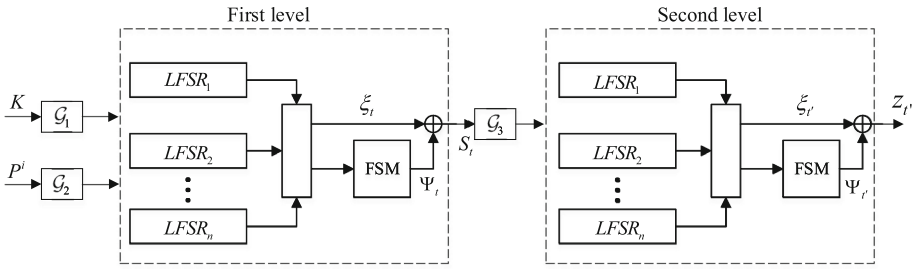

Fig. 1. Structure of the two-level model.

The content of the LFSRs at time t is denoted by ζ_t. At time t, denote the n output bits of LFSRs by $B_t = (b_t^1, \ldots, b_t^n)$, which is also the input to the FSM, and the FSM state by $\sigma_t \in \mathrm{GF}(2)^k$. Then, the next state σ_{t+1} of the FSM can be computed by the current FSM state σ_t and B_t via $\sigma_{t+1} = \mathcal{F}(B_t, \sigma_t)$, where $\mathcal{F} : \sigma_t \mapsto \sigma_{t+1}$ is a permutation for any B_t. The FSM outputs one bit $\Psi_t = \omega_c \cdot \sigma_t$, which is an inner product of its current state σ_t and a constant $\omega_c \in \mathrm{GF}(2)^k$. The core combiner generates one keystream bit z_t as the xor of the FSM output bit Ψ_t and the sum of the LFSRs outputs, i.e., $\Psi_t \oplus \xi_t = z_t$, where $\xi_t = \oplus_{i=1}^n b_t^i$.

Next, we provide a formal description of the workflow of the two-level model. At the first level, the secret key and public nonce P^i (IV)[5] are mixed by two affine transforms $\mathcal{G}_1$ and $\mathcal{G}_2$, then loaded into the n LFSRs linearly. With the preset null state in the FSM, the core generator runs a certain number of clocks and produce η_1-bit output ($\eta_1 > \sum_{i=1}^n L_i$). The last generated $L = \sum_{i=1}^n L_i$ output bits at the first level are permutated into the n LFSRs by another affine transform $\mathcal{G}_3$ and keep the content of the FSM at the end of the first level. We stress here that $\mathcal{G}_3$ inherits the feature that the last $\sum_{i=1}^n L_i$ output bits are only permutated into the LFSRs, without any linear combination among the manipulated bits, for efficiency reasons. From this combined internal state, the core generator produces η_2-bit keystream for encryption for the i-th frame during the second level.

4. Correlations Properties of the Two-Level Model

In this section, both the unconditional and conditional correlations properties based on condition masking of the two-level model are studied, which naturally lead to our new key recovery framework.

4.1. *Unconditional Linear Correlations*

We first study the unconditional correlation properties of the second level, which are exploited in the linear approximation process of the two-level model. Inspired by [10,21], we give a general way to efficiently compute the unconditional correlations at the second level.

Let $\Omega(a, \langle \omega, u \rangle)$ be the correlation $\epsilon(a \cdot \sigma_{t'+1} \oplus \omega \cdot \sigma_{t'} \oplus u \cdot B_{t'})$ of two consecutive steps in the keystream generation, where $a \in \mathrm{GF}(2)^k, u \in \mathrm{GF}(2)^n, \omega \in \mathrm{GF}(2)^k$ are linear masks and $B_{t'}$ represent the n output bits of LFSRs at time t' of the second level. For brevity, we denote the unconditional correlation for a continuous d time instants by

$$\delta(\langle a_1, u_1 \rangle, \ldots, \langle a_{d-1}, u_{d-1} \rangle, a_d) = \epsilon(a_1 \cdot \sigma_{t'+1} \oplus u_1 \cdot B_{t'+1} \oplus \cdots \oplus a_{d-1} \cdot \sigma_{t'+d-1} \oplus u_{d-1} \cdot B_{t'+d-1} \oplus a_d \cdot \sigma_{t'+d}), \quad (1)$$

where $u_1, \ldots, u_{d-1} \in \mathrm{GF}(2)^n, a_1, \ldots, a_{d-1}, a_d \in \mathrm{GF}(2)^k$. The following theorem can be used to compute the correlation for iterative structures [11].

[5] The superscript i is used to indicate the context of the i-th frame.

Theorem 5. *Given functions* $f : GF(2)^m \times GF(2)^p \to GF(2)$ *and* $g : GF(2)^q \to GF(2)^p$, *let* $X \in GF(2)^m$ *and* $Y \in GF(2)^q$ *be two independent random variables. Then, for all* $u \in GF(2)^m$, $v \in GF(2)^q$, *we have*

$$\delta(f(X, g(Y)) \oplus u \cdot X \oplus v \cdot Y) = \sum_{\omega \in GF(2)^p} \delta(f(X, g(Y)) \oplus u \cdot X \oplus \omega \cdot g(Y)) \delta(\omega \cdot g(Y) \oplus v \cdot Y).$$

Now we can present our general iterative computation method to calculate the unconditional correlation in (1).

Theorem 6. *Assume that the initial state* (ζ_0, σ_0) *is random and uniformly distributed in the model, and then we have*

$$\delta(\langle a_1, u_1\rangle, \ldots, \langle a_{d-1}, u_{d-1}\rangle, a_d) = \sum_{\omega \in GF(2)^k} \Omega(a_d, \langle \omega, u_{d-1}\rangle) \cdot \delta(\langle a_1, u_1\rangle, \ldots, \langle a_{d-2}, u_{d-2}\rangle, a_{d-1} \oplus \omega).$$

Proof. To apply Theorem 5, we set $X = B_{t'+d-1}, Y = (\langle \sigma_{t'+1}, B_{t'+1}\rangle, \ldots, \langle \sigma_{t'+d-2}, B_{t'+d-2}\rangle, \sigma_{t+d-1})$, $g(Y) = \sigma_{t'+d-1}$, $f(X, g(Y)) = a_d \cdot \sigma_{t'+d}$, $u = u_{d-1}$ and $v = (\langle a_1, u_1\rangle, \ldots, \langle a_{d-2}, u_{d-2}, a_{d-1}\rangle)$. Thus, we have

$$\begin{aligned}
&\delta(\langle a_1, u_1\rangle, \ldots, \langle a_{d-1}, u_{d-1}\rangle, a_d) \\
&= \delta(f(X, g(Y)) \oplus u_{d-1} \cdot X \oplus v \cdot Y) \\
&= \sum_{\omega \in GF(2)^k} \delta(f(X, g(Y)) \oplus u \cdot X \oplus \omega \cdot g(Y)) \cdot \delta(\omega \cdot g(Y) \oplus v \cdot Y) \\
&= \sum_{\omega \in GF(2)^k} \delta(a_d \cdot \theta_{t'+d} \oplus u_{d-1} \cdot B_{t'+d-1} \oplus \omega \cdot \theta_{t'+d-1}) \\
&\quad \cdot \delta(\omega \cdot \theta_{t'+d-1} \oplus a_1 \cdot \theta_{t'+1} \oplus \cdots \oplus a_{d-1} \oplus \theta_{t'+d-1}) \\
&= \sum_{\omega \in GF(2)^k} \Omega(a_d, \langle \omega, u_{d-1}\rangle) \cdot \delta(\langle a_1, u_1\rangle, \ldots, \langle a_{d-2}, u_{d-2}\rangle, a_{d-1} \oplus \omega),
\end{aligned}$$

which completes the proof. □

Theorem 6 is a generalization of the formulas in [21,22]. It can compute the unconditional correlations between all the input linear functions and all the output linear functions without any miss. Some illustrative examples are given in the two-level E0 case later in Sect. 7.1.

Denote the m consecutive keystream bits as $Z_{t'}^m$ and the m continuous LFSR inputs as $B_{t'}^m$, where $v \cdot Z_{t'}^m = \bigoplus_{j=0}^{m-1} v_j z_{t'+j}$, and $W \cdot B_{t'}^m = \bigoplus_{j=0}^{m-1} (\omega_j \cdot B_{t'+j})$ are two linear functions defined by a $n \times m$ matrix $W = (\omega_0, \ldots, \omega_{m-1})$ and a vector v. We can ignore the effect of t', for the correlations are time-invariant. Then, we have the following corollary on the time complexity of the above recursive method.

Corollary 7. *The recursive expression in Theorem 6 can compute all the correlation coefficients of the form $\epsilon(W \cdot B^m \oplus v \cdot Z^m)$, i.e., the unconditional correlations between the LFSR output sequence and the keystream sequence of the general combiner in $k^m \cdot (n+1)^{m-1}$ iterations.*

Proof. We first prove that all the $\epsilon(W \cdot B^m \oplus v \cdot Z^m)$ can be computed by Theorem 6. Assume $W = (\omega_0, \ldots, \omega_{m-1})$, $v = (v_0, \ldots, v_{m-1})$, where $\omega_i \in GF(2)^n$, $v_i \in GF(2)$ and $\mathbf{1}_n$ represents the vector with all the components being 1, and then we have

$$\begin{aligned}
\epsilon(W \cdot B^m \oplus v \cdot Z^m) &= \epsilon(\omega_0 \cdot B_0 \oplus \cdots \omega_{m-1} \cdot B_{m-1} \oplus v_0 z_0 \oplus \cdots v_{m-1} z_{m-1}) \\
&= \epsilon(\omega_0 \cdot B_0 \oplus \cdots \omega_{m-1} \cdot B_{m-1} \oplus v_0(\mathbf{1}_n \cdot B_0 \oplus a_0 \cdot \sigma_0) \\
&\quad \oplus \cdots \oplus v_m(\mathbf{1}_n \cdot B_{m-1} \oplus a_{m-1} \cdot \sigma_{m-1})) \\
&= \epsilon((\omega_0 \oplus v_0 \mathbf{1}_n) \cdot B_0 \oplus (v_0 a_0) \cdot \sigma_0 \oplus \cdots \oplus (\omega_{m-1} \oplus v_{m-1} \mathbf{1}_n) \\
&\quad \cdot B_{m-1} \oplus (v_{m-1} a_{m-1}) \cdot \sigma_{m-1}),
\end{aligned}$$

where $a_i = v_i \cdot \omega_c$ for $0 \le i \le m-1$. Note that if the linear mask $\omega_{m-1} \oplus v_{m-1}\mathbf{1}_n$ of B_{m-1} is not 0, since the variable B_{m-1} will be independent to all the other variables, then the total correlation will be 0. Hence, we always assume $\omega_{m-1} \oplus v_{m-1}\mathbf{1}_n = \mathbf{0}_n$. Now, we can compute the above correlation by Theorem 6. For a certain m, we have $v_0 a_0 \neq 0$ and $v_{m-1} a_{m-1} \neq 0$. Because of the symmetry of the combiner's output and next-state functions with respect to the n input variables, the correlation depends on $v_i a_i$ and $wt(B_i)$. Hence, computing all the correlations at the second level only needs about $k^m \cdot (n+1)^{m-1}$ iterations. □

Theorem 6 and Corollary 7 are used in the linear approximation of the second level in the model.

4.2. *Conditional Correlations Based on Condition Masking*

Now let us look at the conditional correlation properties of the two-level model. Several consecutive steps of the core generator can be regarded as a vectorial Boolean function, and we would like to investigate the conditional correlation properties of this derived function.

Generally, there are two sets of inputs to the FSM in the first level at time t, i.e., the n LFSR output bits $B_t = (b_t^1, \ldots, b_t^n)$ and the k memory bits $\sigma_t = (\sigma_t^{k-1}, \ldots, \sigma_t^0) \in \mathrm{GF}(2)^k$. Consider l continuous time instants and let $\gamma = (\gamma_0, \gamma_1, \ldots, \gamma_{l-1}) \in \mathrm{GF}(2)^l$ be a linear mask with $\gamma_0 = \gamma_{l-1} = 1$. Define the inputs to the FSM as

$$\mathcal{B}_t = B_t B_{t+1} \cdots B_{t+l-2} \in \mathrm{GF}(2^{n(l-1)}), \quad \sigma_{t+1} \in \mathrm{GF}(2)^k$$

and the FSM outputs $C_t = (\omega \cdot \sigma_t, \ldots, \omega \cdot \sigma_{t+l-1})$. Then, the function $h^\gamma_{\mathcal{B}_t} : \sigma_t \to \gamma \cdot C_t$ is well defined, as $\gamma_0 = \gamma_{l-1} = 1$ is necessary and sufficient to recursively compute $\gamma \cdot C_t$ with the knowledge of $\mathcal{B}_t$ and σ_t as shown in Fig. 2. The bias $\epsilon(h^\gamma_{\mathcal{B}_t})$ can be easily computed by an exhaustive search over all the possible values of σ_t. For different values of $\mathcal{B}_t$, the bias $\epsilon(h^\gamma_{\mathcal{B}_t})$ may be different, while the mean value $\mathrm{E}[\epsilon(h^\gamma_{\mathcal{B}_t})]$ is a good

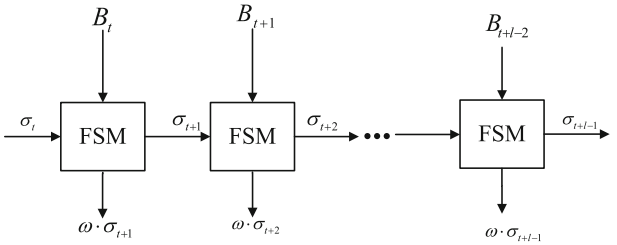

Fig. 2. The computation process of C_t.

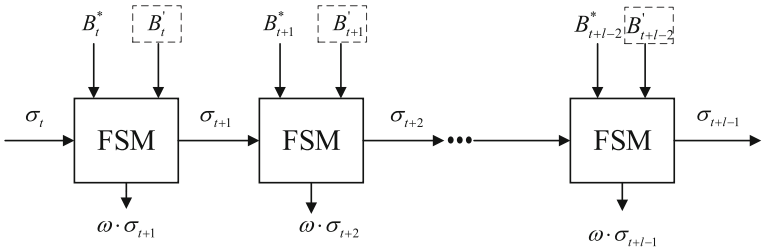

Fig. 3. The new computation process of C_t.

estimate in the attacks. In general, we may expect to see the bias with a proper value of γ. Now, we are ready for the definition of condition mask.

Definition 8. Given a function $h : GF(2)^u \times GF(2)^v \rightarrow GF(2)^r$ with $\mathcal{B} \in GF(2)^u$, $X \in GF(2)^v$, where $\mathcal{B}$ is the key-related part and the possible condition vector. Let $\mathcal{B} = (b_0, \ldots, b_{u-1}) \in GF(2)^u$ and $\lambda = (\lambda_0, \lambda_1, \ldots, \lambda_{u-1}) \in GF(2)^u$ with $supp(\lambda) = \{0 \leq i \leq u-1 | \lambda_i = 1\} = \{l_1, \ldots, l_m\}$ $(l_j < l_{j+1})$. Then, the shrunken vector of $\mathcal{B}$ defined by λ is $\mathcal{B}' = (b_{l_1}, \ldots, b_{l_m}) \in GF(2)^m$. Here, λ is called the condition mask of $\mathcal{B}$. Further, other bits in $\mathcal{B}$ form another vector and are denoted by $\mathcal{B}^* \in GF(2)^{u-m}$, which is the complement part of $\mathcal{B}'$. We define an operator $'\backslash'$ to represent the above process and have $\mathcal{B}^* = \mathcal{B} \setminus \mathcal{B}'$.

This definition indicates that the adversary may not use the full vector as the condition, but only search the correlations conditioned on a subset of $\mathcal{B}$ defined by a mask λ.

For the model, given a condition mask $\lambda = (\lambda_t, \lambda_{t+1}, \ldots, \lambda_{t+l-2}) \in \mathrm{GF}(2)^{n(l-1)}$, where $\lambda_j \in \mathrm{GF}(2)^n$ corresponds to B_j for $j = t, t+1, \ldots, t+l-2$, let the condition vector defined by λ be $\mathcal{B}_t'$ and its complement $\mathcal{B}_t^*$ which includes the other bits. The target function $h^{\gamma}_{\mathcal{B}_t}$ can now be generalized as

$$h^{\Lambda}_{\mathcal{B}_t'} : \sigma_t, \mathcal{B}_t^* \rightarrow \gamma \cdot C_t \oplus \eta \cdot \mathcal{B}_t^*, \tag{2}$$

where $\Lambda = (\gamma, \eta)$ and $|\eta| = |\mathcal{B}_t^*|$.[6] As we can see, this function induces a large class of correlations based on both the linear mask and the condition mask.

[6] $|\cdot|$ denotes the cardinality of a vector.

Figure 3 also shows that though the straightforward computation process of C_t is frustrated by the condition mask $\lambda \neq \mathbf{1}_u$, the bias can still be computed. Since $\mathcal{B}_t$ is the outputs of the LFSRs, it is the key-related material in the first level. In [19], the attacker guesses the full vector $\mathcal{B}_{t+1}$, while now he/she only needs to guess $\mathcal{B}'_t$, a part of $\mathcal{B}_t$, to mount the attack on the model.[7] This is the reason that the time/memory complexities of the attack can be significantly reduced.

Note that in the initialization phase, $\mathcal{B}_t$ at level one can be expressed by

$$\mathcal{B}_t^i = L_t(K) \oplus L'_t\left(P^i\right), \tag{3}$$

where L_t and L'_t are the known linear functions dependent on l and t. The knowledge of $\mathcal{B}_t^i$ will directly lead to the linear equations on the original encryption key. This motivates us to study the bias $\epsilon(h^{\Lambda}_{\mathcal{B}'_t})$ defined by a certain condition mask λ.

The following property shows that the more knowledge of the LFSR bits $\mathcal{B}$, the larger conditional correlation we will obtain, which exactly matches the intuition.

Property 9. *Given a function f with a partial input $\mathcal{B}$ and two condition masks λ_1, λ_2, let $\mathcal{B}_1$ be the condition vector defined by λ_1 and $\mathcal{B}_2$ be the condition vector defined by λ_2. If $supp(\lambda_2) \subseteq supp(\lambda_1)$, then we have $E[\Delta(f_{\mathcal{B}_1})] \geq E[\Delta(f_{\mathcal{B}_2})]$, where equality holds if and only if $D_{f_{\mathcal{B}_1}}$ is independent of $\mathcal{B}_1 \setminus \mathcal{B}_2$.*

Proof. By Definition 2, we have $\mathrm{E}[\Delta(f_{\mathcal{B}_2})] = 2^r \sum_{a \in GF(2)^r} \mathrm{E}_{\mathcal{B}_2}\big[(D_{f_{\mathcal{B}_2}}(a) - \frac{1}{2^r})^2\big]$, where the expectation is taken over uniformly distributed $\mathcal{B}_2$ for the fixed a. Because of $D_{f_{\mathcal{B}_2}}(a) = E_{\mathcal{B}_1 \setminus \mathcal{B}_2}[D_{f_{\mathcal{B}_1}}(a)]$ for any fixed a, we have

$$\begin{aligned}
E_{\mathcal{B}_2}[\Delta(f_{\mathcal{B}_2})] &= 2^r \sum_{a \in GF(2)^r} E_{\mathcal{B}_2}\left[\left(E_{\mathcal{B}_1 \setminus \mathcal{B}_2}[D_{f_{\mathcal{B}_1}}(a)] - \frac{1}{2^r}\right)^2\right] \\
&= 2^r \sum_{a \in GF(2)^r} E_{\mathcal{B}_2}\left[E^2_{\mathcal{B}_1 \setminus \mathcal{B}_2}\left[D_{f_{\mathcal{B}_1}}(a) - \frac{1}{2^r}\right]\right] \\
&\leq 2^r \sum_{a \in GF(2)^r} E_{\mathcal{B}_2}\left[E_{\mathcal{B}_1 \setminus \mathcal{B}_2}\left[\left(D_{f_{\mathcal{B}_1}}(a) - \frac{1}{2^r}\right)^2\right]\right] \\
&= 2^r \sum_{a \in GF(2)^r} E_{\mathcal{B}_2, \mathcal{B}_1 \setminus \mathcal{B}_2}\left[\left(D_{f_{\mathcal{B}_1}}(a) - \frac{1}{2^r}\right)^2\right] = E_{\mathcal{B}_1}[\Delta(f_{\mathcal{B}_1})].
\end{aligned}$$

The inequality is obtained according to the theory of statistics that for any fixed a, $E^2_{\mathcal{B}_1 \setminus \mathcal{B}_2}[D_{f_{\mathcal{B}_1}}(a) - \frac{1}{2^r}] \leq E_{\mathcal{B}_1 \setminus \mathcal{B}_2}[(D_{f_{\mathcal{B}_1}}(a) - \frac{1}{2^r})^2]$ where equality holds if and only if $D_{f_{\mathcal{B}_1}}$ is independent of the condition vector $\mathcal{B}_1 \setminus \mathcal{B}_2$. □

[7] In the real E0, the FSM state at time t always contains the 2 bits c_t^0 and c_{t-1}^0, while in the general model there is not such a property necessarily.

From this property, give a function $h : \mathrm{GF}(2)^u \times \mathrm{GF}(2)^v \to \mathrm{GF}(2)^r$ with $\mathcal{B} \in \mathrm{GF}(2)^u$, $X \in \mathrm{GF}(2)^v$ and a condition mask λ, we have $\mathrm{E}[\Delta(h_{\mathcal{B}})] \geq \mathrm{E}[\Delta(h_{\mathcal{B}'})] \geq \Delta(h)$. Moreover, for a fixed condition mask λ, its maximum bias $\max_{\Lambda}(\mathrm{E}[\Delta(h^{\Lambda}_{\mathcal{B}'_i})])$ among all the linear masks Λ is an essential measure of it. The larger the maximum bias, the better the condition mask is. The best choice of the condition mask can be determined according to the context of the underlying primitive.

4.3. *Key Recovery Attacks on the Model*

As mentioned before, the essential problem lies in the core is to distinguish a biased sample sequence from a pool of random-like sample sequences. Since the involved sample sequences are derived from some key-related information, this distinguisher can be used to identify the correct key. Formally, given a function $f : \mathrm{GF}(2)^m \times \mathrm{GF}(2)^{u-m} \times \mathrm{GF}(2)^v \to \mathrm{GF}(2)^r$ and a condition mask λ, let

$$f_{\mathcal{B}'}(\mathcal{B}^*, X) = f(\mathcal{B}', \mathcal{B}^*, X),$$

where $\mathcal{B} = \mathcal{B}' \cup \mathcal{B}^* \in \mathrm{GF}(2)^u$, $X \in \mathrm{GF}(2)^v$. Here, the condition vector defined by λ is $\mathcal{B}' \in \mathrm{GF}(2)^m$ and $\mathcal{B}^* = \mathcal{B} \setminus \mathcal{B}'$. If $\mathcal{B}'$ is determined by κ-bit key information, then denote by $\mathcal{B}'^{\mathcal{K}}$ the value derived when the guessing value of the key material is $\mathcal{K}$, then the formal description of the problem is as follows.

Definition 10. There are 2^κ sequences of $\mathcal{N}$ samples with the following characteristics: one biased sequence has $\mathcal{N}$ samples $(f_{\mathcal{B}'^{\mathcal{K}}_i}, \mathcal{B}'^{\mathcal{K}}_i)$ $(i = 1, \ldots, \mathcal{N})$ with the correct key $\mathcal{K}$; the other $2^\kappa - 1$ sequences consists of $\mathcal{N}$ independently and uniformly distributed random variables $(Z^K_i, \mathcal{B}'^K_i)$ $(i = 1, \ldots, \mathcal{N})$ with the wrong keys. The problem is to efficiently distinguish the biased sequence from the other sequences with the minimum number $\mathcal{N}$ of samples.

Following [2], the minimum number $\mathcal{N}$ of samples for an optimal distinguisher using the unconditional correlation to effectively distinguish a sequence of $\mathcal{N}$ output samples of f from $(2^\kappa - 1)$ truly random sequences of equal length is

$$\mathcal{N} = \frac{4\kappa \log 2}{\Delta(f)},$$

while with the smart distinguisher in [19] based on the condition vector $\mathcal{B}$, the number of sample needed is

$$\mathcal{N}_{\mathcal{B}} = \frac{4\kappa \log 2}{\mathrm{E}[\Delta(f_{\mathcal{B}})]}.$$

Since $\mathrm{E}[\Delta(f_{\mathcal{B}})] \geq \Delta(f)$, we have $\mathcal{N}_{\mathcal{B}} \leq \mathcal{N}$. In our condition masking terminology, we have the following theorem on the attack complexities.

Theorem 11. *Given a condition mask λ, Algorithm 1 solves the problem in Definition 10 with*

$$\mathcal{N}_{\mathcal{B}'} = \frac{4\kappa \log 2}{E[\Delta(f_{\mathcal{B}'})]}$$

samples and the time complexity is $O(\mathcal{N}_{\mathcal{B}'} \cdot 2^{\kappa})$, where the condition bits $\mathcal{B}'$ is defined by λ, the expectation is taken over all the uniformly distributed $\mathcal{B}'$. Further, if the $\mathcal{B}_i'^K$ and Z_i^K can be expressed by

$$\mathcal{B}_i'^K = L(K) \oplus a_i, \tag{4}$$

$$Z_i^K = L'(K) \oplus a_i' \oplus g\left(\mathcal{B}_i'^K\right), \tag{5}$$

for all κ-bit K and $i = 1, 2, \ldots, \mathcal{N}$, where g is an arbitrary function, L, L' are linear functions, and a_i, a_i' are independently and uniformly distributed constants known to the distinguisher. Under these assumptions, we can use the FWT algorithm to achieve the optimal time complexity $O(\mathcal{N}_{\mathcal{B}'} + \kappa 2^{\kappa+1})$ with pre-computation $O(\kappa 2^{\kappa})$ and $|\mathcal{B}'| = \kappa$.

Proof. The case $\lambda = \mathbf{1}_u$ was proved in [19]. When $\lambda \neq \mathbf{1}_u$, we can make a substitution $T = \lambda \diamond \mathcal{B}$ and use the same way to prove this theorem, where $\diamond$ represents the action of the condition mask on $\mathcal{B}$. □

Algorithm 1 The key recovery framework on the model based on condition masking

Parameters: $\mathcal{N}$, λ, $\mathcal{B}$ and $D_{f_{\mathcal{B}'}}$
input:
1: for $i = 1, 2, \ldots, \mathcal{N}$, $\mathcal{B}_i'^K$ for all κ-bit K
2: $Z_i^{\mathcal{K}} = f_{\mathcal{B}'}(\mathcal{B}_i^{*\mathcal{K}}, X_i)$ for the correct key $\mathcal{K}$ with uniformly and independently distributed v-bit vectors X_i and $\mathcal{B}_i^{*\mathcal{K}} = \mathcal{B}_i^{\mathcal{K}} \setminus \mathcal{B}'^{\mathcal{K}_i}$
3: uniformly and independently distributed Z_i^K for all keys $K \neq \mathcal{K}$
Goal: find $\mathcal{K}$
Processing:
4: **for** all κ-bit K **do**
5: $G(K) \leftarrow 0$
6: **for** $i = 1, \ldots, \mathcal{N}$ **do**
7: $G(K) \leftarrow G(K) + \log_2(2^r \cdot D_{f_{\mathcal{B}_i'^K}}(Z_i^K))$
8: **end for**
9: **end for**
10: output $\mathcal{K}$ that maximizes the grade $G(\mathcal{K})$

Remarks. There is a subtle difference between the case $\lambda = \mathbf{1}_u$ and $\lambda \neq \mathbf{1}_u$, i.e., our framework is different from the one in [19]. Precisely, the premises (4) and (5) when $\lambda \neq \mathbf{1}_u$ can be the same as those when $\lambda = \mathbf{1}_u$ in many cases, i.e., even if $\lambda \neq \mathbf{1}_u$, we can still have the same conclusion about the complexity reduction under the premise of $\lambda = \mathbf{1}_u$:

$$\mathcal{B}_i^K = L(K) \oplus a_i, \tag{4'}$$

$$Z_i^K = L'(K) \oplus a_i' \oplus g(\mathcal{B}_i^k). \tag{5'}$$

This fact results from the linear approximation process of the underlying primitive. For the Bluetooth two-level E0, we will demonstrate this issue in the following. We believe there are other cases that our arguments hold.

We should not ignore the impact of the cardinality of the condition vector $|\mathcal{B}'| = \kappa$ on the time/memory complexities. It is easy to see that for $\lambda \neq \mathbf{1}_u$, the cardinality κ can be reduced and the time/memory complexities can be exponentially reduced accordingly. It is expected that with a careful choice of the condition mask, we can get better tradeoffs on the time/memory/data complexity curve compared to the case $\lambda = \mathbf{1}_u$. This is why, we introduce the notion of condition masking.

Further, note that not all the bits in the condition vector $\mathcal{B}$ have the same influence on the correlation. In fact, some are more important than others, i.e., it is of high probability that only a subset of the condition bits can determine the magnitude of the correlation. Thus, it is the crucial task of the adversary to determine the most important part of the condition vector for each specified primitive.

Next, we build the linear approximations of the two-level model with condition masking. The linear approximation is based on the re-initialization property of the model, detailed in Sect. 3. As previously stated, we make the following assumption.

Assumption 1. The affine transform $\mathcal{G}_3$ is just a bit permutation of the input variables, i.e., no linear combination among the manipulated bits is introduced when loading the last $L = \sum_{i=1}^{n} L_i$ bits generated at the end of the first level into the n LFSRs at the beginning of the second level.

Throughout this paper, in order to distinguish the ξ_t, Ψ_t at the first level and $\xi_{t'}, \Psi_{t'}$ at the second level, we introduce some notations as follow. Let $R_t = \xi_t$, $V_{t'} = \xi_{t'}$ and $\alpha_t = \Psi_t$, $\beta_{t'} = \Psi_{t'}$. Denote the last generated L bits at the first level by $S^i_{[-L+1,\ldots,0]}$ in the model, where $S^i_{[-L+1,\ldots,0]} = R^i_{[-L+1,\ldots,0]} \oplus \alpha^i_{[-L+1,\ldots,0]}$. We also have

$$V^i_{[1,\ldots,L]} = \mathcal{G}_3(R^i_{[-L+1,\ldots,0]}) \oplus \mathcal{G}_3(\alpha^i_{[-L+1,\ldots,0]}).$$

For brevity, we define $(U^i_1, \ldots, U^i_L) = \mathcal{G}_3(R^i_{[-L+1,\ldots,0]})$. According to $\mathcal{G}_3$, V^i can be expressed as

$$V^i_{t'} = U^i_{t'} \oplus \bigoplus_{j=1}^{n} \alpha^i_{t_j}, \text{ for } t' = 1, \ldots, L_1,$$

where t_i are the fixed time instants of α^i before the application of $\mathcal{G}_3$ dependent on each considered primitive.

Note that we have $U^i_{t'} = H_{t'}(K) \oplus H'_{t'}(P^i)$, where $H_{t'}$, $H'_{t'}$ are public linear functions dependent on t'. At the second level, $z_{t'} = V_{t'} \oplus \beta_{t'}$ holds. Hence we have

$$z_{t'} \oplus H_{t'}(K) \oplus H'_{t'}(P^i) = \bigoplus_{j=1}^{n} \alpha^i_{t_j} \oplus \beta^i_{t'}, \text{ for } t' = 1, \ldots, L_1. \tag{6}$$

Given a linear mask γ with $|\gamma| = l$, let $Z^i_{t'} = (z^i_{t'}, \ldots, z^i_{t'+l-1})$. Since at the second level, the L-bit keystream S^i_t are loaded back into the n LFSRs according to the bit permutation $\mathcal{G}_3$, then Eq. (6) can be rewritten with the linear mask notation as

$$\mathcal{G}_3(\gamma) \cdot (Z^i_{t'} \oplus \mathcal{L}_{t'}(K) \oplus \mathcal{L}'_{t'}(P^i)) = \bigoplus_{j=1}^{n} (\gamma \cdot C^i_{t_j}) \oplus \mathcal{G}_3(\gamma) \cdot C^i_{t'}, \tag{7}$$

for $i = 1, \ldots, \mathcal{N}$, $\mathcal{L}_{t'}$, $\mathcal{L}'_{t'}$ are fixed linear functions which can be derived from $H_{t'}$, $H'_{t'}$ and $\mathcal{G}_3(\gamma)$ is the resultant linear mask after the restricted action of the bit permutation $\mathcal{G}_3$. Equation (7) corresponds to the case of $\lambda = \mathbf{1}_u$.

By Eq. (2), we can rewrite this equation as follows:

$$\begin{aligned} &\mathcal{G}_3(\gamma) \cdot (Z^i_{t'} \oplus \mathcal{L}_{t'}(K) \oplus \mathcal{L}'_{t'}(P^i)) \oplus \bigoplus_{j=1}^{n} (\eta \cdot \mathcal{B}^{*i}_{t_j}) \\ &= \bigoplus_{j=1}^{n} (\gamma \cdot C^i_{t_j} \oplus \eta \cdot \mathcal{B}^{*i}_{t_j}) \oplus \mathcal{G}_3(\gamma) \cdot C^i_{t'} \,. \end{aligned} \tag{8}$$

For brevity, given masks λ and Λ, we use the simplified notations $h^{\Lambda}_{\mathcal{B}'^i_t}$, $h^{\mathcal{G}_3(\gamma)}$ to denote $h^{\Lambda}_{\mathcal{B}'^i_t}(\mathcal{B}^{*i}_t, \sigma^i_t)$, $h^{\mathcal{G}_3(\gamma)}(\mathcal{B}^i_{t'}, \theta^i_{t'})$ hereafter. Besides, Eq. (3) implies that $\mathcal{B}^{*i}_t = \mathcal{B}^i_t \setminus \mathcal{B}'^i_t$ is the linear combination of K and P^i. Now Eq. (8) becomes

$$\mathcal{G}_3(\gamma) \cdot (Z^i_{t'} \oplus \mathcal{L}_{t'}(K) \oplus \mathcal{L}'_{t'}(P^i)) \oplus \eta \cdot (L_1(K) \oplus L_2(P^i)) = \bigoplus_{j=1}^{n} h^{\Lambda}_{\mathcal{B}'^i_{t_j}} \oplus h^{\mathcal{G}_3(\gamma)}, \tag{9}$$

where L_1, L_2 are public linear functions and $h^{\mathcal{G}_3(\gamma)}$ is the unconditioned function at the second level. Equation (9) is the hybrid bitwise linear approximation based on condition masking for the two-level model in Fig. 1, where $h^{\Lambda}_{\mathcal{B}'^i_{t_j}}$ are derived from the first level and $h^{\mathcal{G}_3(\gamma)}$ contains the unconditional correlation for the second level.

4.4. *Bitwise Key Recovery Attack on the Model*

Given the condition mask λ and the linear masks $\Lambda = (\gamma, \eta)$, we define the following sign function to estimate the effective value of $h^{\Lambda}_{\mathcal{B}'^i_t}$ (Eq. (2)):

$$g^{\Lambda}(\mathcal{B}_t'^i) = \begin{cases} 1, & \text{if } \epsilon\left(h^{\Lambda}_{\mathcal{B}_t'^i}\right) > 0 \\ 0, & \text{if } \epsilon\left(h^{\Lambda}_{\mathcal{B}_t'^i}\right) < 0 \end{cases} \tag{10}$$

for all $\mathcal{B}_t'^i \in \mathrm{GF}(2)^{wt(\lambda)}$ such that $\epsilon(h^{\Lambda}_{\mathcal{B}_t'^i}) \neq 0$. For brevity, let

$$\mathcal{B}^i_\lambda = (\mathcal{B}_{t_1}'^i, \mathcal{B}_{t_2}'^i, \ldots, \mathcal{B}_{t_n}'^i), \quad \mathcal{X}^i = (Y^i_{t_1}, Y^i_{t_2}, \ldots, Y^i_{t_n}, X^i_{t'}, \mathcal{B}^i_{t'}),$$

where $Y^i_{t_j} = (\sigma^i_{t_j}, \mathcal{B}^*_{t_j})$ is the unknown input to $h^{\Lambda}_{\mathcal{B}'^i_{t_j}}$, and $X^i_{t'}, \mathcal{B}^i_{t'}$ are the inputs to $h^{\mathcal{G}_3(\gamma)}$. By Eq. (3) and (9), the knowledge of the key K is contained in $\mathcal{B}'^i_{t_j}$, $\mathcal{L}_{t'}(K)$ and $L_1(K)$. Let $K_1 = (L_{t_1}(K), L_{t_2}(K), \ldots, L_{t_n}(K))$ be the $wt(\lambda)n$ bits contained in $\mathcal{B}^i_\lambda$ and $K_2 = \mathcal{G}_3(\gamma) \cdot \mathcal{L}_{t'}(K) \oplus \eta \cdot L_1(K)$ be the subkeys. Denote by $\widetilde{\cdot}$ the guessed value of the argument. First, choose an appropriate condition mask λ and guess the subkeys $\widetilde{K_1}, \widetilde{K_2}$. As P^i is known for each frame $i = 1, \ldots, \mathcal{N}$, we can compute the condition vector $\mathcal{B}^i_\lambda$. Second, to distinguish the correct keys from the wrong ones, we define a mapping $\mathcal{F}^{\Lambda}_{B^i_\lambda}(\mathcal{X}^i)$ as follows.

$$\mathcal{F}^{\Lambda}_{\mathcal{B}^i_\lambda}(\mathcal{X}^i) = \begin{cases} \bigoplus_{j=1}^{n}\left(h^{\Lambda}_{\mathcal{B}'^i_{t_j}} \oplus g^{\Lambda}\left(\widetilde{\mathcal{B}'^i_{t_j}}\right)\right) \oplus h^{\mathcal{G}_3(\gamma)}, & \text{if } \prod_{j=1}^{n} \epsilon\left(h^{\Lambda}_{\mathcal{B}'^i_{t_j}}\right) \neq 0 \\ \text{a truly random bit,} & \text{otherwise} \end{cases}$$

With Eq. (10) the value of $\mathcal{F}^{\Lambda}_{B^i_\lambda}(\mathcal{X}^i)$ can be computed as

$$\mathcal{F}^{\Lambda}_{\mathcal{B}^i_\lambda}(\mathcal{X}^i) = \mathcal{G}_3(\gamma) \cdot \left(Z^i_{t'} \oplus \mathcal{L}'_{t'}(P^i)\right) \oplus \eta \cdot L_2(P^i) \oplus \widetilde{K_2} \oplus \bigoplus_{j=1}^{n} g^{\Lambda}\left(\widetilde{\mathcal{B}'^i_{t_j}}\right).$$

If $\mathcal{N}$ frames are available, we can compute the value of $\mathcal{F}^{\Lambda}_{\mathcal{B}^i_\lambda}(\mathcal{X}^i)$ for each possible key by the above equation $\mathcal{N}$ times. With appropriate choice of Λ and λ, if K_1, K_2 are correctly guessed, then $\mathrm{E}[\Delta(\mathcal{F}^{\Lambda}_{B^i_\lambda}(\mathcal{X}^i))] > 0$ and we expect $\mathcal{F}^{\Lambda}_{\mathcal{B}^i_\lambda}(\mathcal{X}^i)$ equals one most of the times. Otherwise, $\mathcal{F}^{\Lambda}_{\mathcal{B}^i_\lambda}(\mathcal{X}^i)$ is estimated by the uniform distribution. Third, we get $\mathcal{N}$ outputs of the source for every possible key. Submitting these samples to the distinguisher in Algorithm 1, with the $\kappa = wt(\lambda)n + 1$, $u = n(l-1)$, $m = wt(\lambda)$, $v = (n+1)k + n(n+1)(l-1) - wt(\lambda)n$ and $r = 1$, we are expected to successfully restore the correct keys.

4.5. *Vectorial Key Recovery Attack on the Model*

Now we enhance the above attack by using multiple linear approximations simultaneously. Since the conditional correlations based on condition masking are not likely to be larger than those based on the whole condition vector, we appeal to the vectorial approach to keep the data complexity as low as possible.

Assume we use s mutually independent linear approximations. Let $\Gamma = (\Lambda_1, \ldots, \Lambda_s)$ and $\Gamma' = (\mathcal{G}_3(\gamma_1), \ldots, \mathcal{G}_3(\gamma_s))$ denote the linear mask of these s approximations, where $\Lambda_i = (\gamma_i, \eta_i)$, and $|\gamma_1| = \cdots = |\gamma_s| = l$ with $s < l$. Let

$$\mathcal{F}^{\Gamma}_{\mathcal{B}^i_\lambda}(\mathcal{X}^i) = \left(\mathcal{F}^{\Lambda_1}_{\mathcal{B}^i_\lambda}, \ldots, \mathcal{F}^{\Lambda_s}_{\mathcal{B}^i_\lambda}\right), g^{\Gamma} = \left(g^{\Lambda_1}(\mathcal{B}'^i_t), \ldots, g^{\Lambda_s}(\mathcal{B}'^i_t)\right)$$

and $h^{\Gamma}_{\mathcal{B}'^i_t} = (h^{\Lambda_1}_{\mathcal{B}'^i_t}, \ldots, h^{\Lambda_s}_{\mathcal{B}'^i_t})$, $h^{\Gamma'} = (h^{\mathcal{G}_3(\gamma_1)}, \ldots, h^{\mathcal{G}_3(\gamma_s)})$. Here, the first $g^{\Lambda_1}(\mathcal{B}'^i_t)$ in g^{Γ} is determined by Eq. (10). The other bits are determined as follows: for the j-th bit, we just let it be an uniformly distributed bit if $\epsilon(h^{\Lambda_1}_{\mathcal{B}'^i_{t_j}}) = 0$, otherwise take 0 or 1 according to the definition in Eq. (10). Since we have found the efficient condition mask λ and linear mask $\Lambda_1 = (\gamma_1, \omega_1)$ in the bitwise attack, we extend $\mathcal{F}^{\Lambda_1}_{\mathcal{B}^i_\lambda}$ to a s-dimensional vector, i.e.,

$$\mathcal{F}^{\Gamma}_{\mathcal{B}^i_\lambda}(\mathcal{X}^i) = \begin{cases} \bigoplus_{j=1}^{n}\left(h^{\Gamma}_{\mathcal{B}'^i_{t_j}} \oplus g^{\Gamma}\left(\widetilde{\mathcal{B}'^i_{t_j}}\right)\right) \oplus h^{\Gamma'}, & \text{if } \prod_{j=1}^{n}\epsilon\left(h^{\Lambda_1}_{\mathcal{B}'^i_{t_j}}\right) \neq 0 \\ \text{a uniformly distributed } s\text{-bit vector}, & \text{otherwise.} \end{cases}$$

In this way, we have constructed an approximation of two-level model in the vectorial approach. For the correct guess $\widetilde{K} = K$, we have $\mathcal{F}^{\Gamma}_{\mathcal{B}^i_\lambda}(\mathcal{X}^i) = \bigoplus_{j=1}^{n}(h^{\Gamma}_{\mathcal{B}'^i_{t_j}} \oplus g^{\Gamma}(\mathcal{B}'^i_{t_j})) \oplus h^{\Gamma'}$ and $\mathrm{E}[\Delta(\mathcal{F}^{\Gamma}_{\mathcal{B}^i_\lambda}(\mathcal{X}^i))] > 0$. For each wrong guess, the components of the s-dimensional vector $\mathcal{F}^{\Gamma}_{\mathcal{B}^i_\lambda}$ are uniformly distributed and we estimate the distribution $D_{\mathcal{F}^{\Gamma}_{\mathcal{B}^i_\lambda}(\mathcal{X}^i)}$ as a s-bit uniform distribution for all i such that $\mathrm{E}[\Delta(\mathcal{F}^{\Gamma}_{\mathcal{B}^i_\lambda}(\mathcal{X}^i))] = 0$.

With the appropriate choice of $\Gamma = (\Lambda_1, \ldots, \Lambda_s)$, we can get larger correlation values than those in the bitwise case. Thus, the data complexity $N_{\mathcal{B}'}$ is effectively reduced compared to the bitwise attack. Again, submitting 2^{κ} sequences of $\mathcal{N}_{\mathcal{B}'}$ pairs $(\mathcal{F}^{\Gamma}_{\mathcal{B}^i_\lambda}(\mathcal{X}^i), \widetilde{\mathcal{B}^i_\lambda})$ to Algorithm 1, we can eventually recover the κ-bit K.

Now we study how to choose the linear mask vector Γ. We first select a linear mask $\Lambda_1 = (\gamma_1, \eta_1)$ in the bitwise attack. Under this Λ_1, we search for other masks $\Lambda_j (j \geq 2)$ to maximize the total correlation. The following theorem provides a guideline for an adversary to construct the vector by depicting the criterion when he/she could gain in correlation by moving from $(s-1)$-dimension unit to s-dimension unit.

Theorem 12. *Let $\Gamma_s = (\Lambda_1, \ldots, \Lambda_s)$ be the linear mask in the s-dimensional attack with condition vector $\mathcal{B}$ and condition mask λ. Denote the joint probability by $P_{a_1 \cdots a_s} = P(h^{\Lambda_1}_{\mathcal{B}'} = a_1, \ldots, h^{\Lambda_s}_{\mathcal{B}'} = a_s)$, where $a_i \in GF(2)$ for $1 \leq i \leq s$. Let $P_{00\cdots00} = \frac{1}{2^s} + \xi_{00\cdots00}$, $P_{00\cdots01} = \frac{1}{2^s} + \xi_{00\cdots01}, \ldots, P_{11\cdots11} = \frac{1}{2^s} + \xi_{11\cdots11}$, where $-\frac{1}{2^s} \leq \xi_j \leq \frac{1}{2^s}$ for all $j \in GF(2)^s$ and $\sum_{j \in GF(2)^s} \xi_j = 0$, then $\Delta(h^{\Gamma_s}_{\mathcal{B}'}) \geq \Delta(h^{\Gamma_{s-1}}_{\mathcal{B}'})$, where the equality holds if and only if*

$$\xi_{00\cdots00} = \xi_{00\cdots01}, \xi_{00\cdots10} = \xi_{00\cdots11}, \ldots, \xi_{11\cdots10} = \xi_{11\cdots11}.$$

Proof. From the assumption, the $(s-1)$-dimensional joint probability can be computed as[8] $P_{00\cdots0*} = \frac{1}{2^{s-1}} + \xi_{00\cdots00} + \xi_{00\cdots01}$, $P_{00\cdots1*} = \frac{1}{2^{s-1}} + \xi_{00\cdots10} + \xi_{00\cdots11}, \cdots, P_{11\cdots1*} = \frac{1}{2^{s-1}} + \xi_{11\cdots10} + \xi_{11\cdots11}$. By the definition of SEI, we have

$$\begin{aligned}\Delta(h_{\mathcal{B}'}^{\Gamma_s}) &= 2^s(\xi_{00\cdots00}^2 + \xi_{00\cdots01}^2 + \cdots + \xi_{11\cdots11}^2),\\ \Delta(h_{\mathcal{B}'}^{\Gamma_{s-1}}) &= 2^{s-1}((\xi_{00\cdots00} + \xi_{00\cdots01})^2 + (\xi_{00\cdots10} + \xi_{00\cdots11})^2\\ &\quad + \cdots + (\xi_{11\cdots10} + \xi_{11\cdots11})^2).\end{aligned}$$

We can see that $\Delta(h_{\mathcal{B}'}^{\Gamma_s}) - \Delta(h_{\mathcal{B}'}^{\Gamma_{s-1}}) = 2^{s-1}((\xi_{00\cdots00} - \xi_{00\cdots01})^2 + (\xi_{00\cdots10} - \xi_{00\cdots11})^2 + \cdots + (\xi_{11\cdots10} - \xi_{11\cdots11})^2) \geq 0$, from which we can easily derive the conclusion. □

This theorem indicates that high-dimensional attack will always be better than or at least be the same as low-dimensional attacks. Besides, if an adversary choose the linear masks following the rules in this theorem, then he could always gain in correlation. Further, there are some other rules when choosing Γ. First, the linear masks γ_j for $j = 1, \ldots, s$ should be linearly independent with $s \leq l - 2$. Second, when the key is wrong, $\mathcal{F}_{\mathcal{B}_\lambda^i}^{\Lambda_j}$ is an uniformly distributed bit for $1 \leq j \leq s$ in the bitwise attack. If they are independent to each other, $\mathcal{F}_{\mathcal{B}_\lambda^i}^{\Gamma}$ follows a s-bit uniform distribution. Thus, when choosing the new $\Lambda_j = (\gamma_j, \omega_j)$ $(j > 1)$, we should keep the independence among the different components $\mathcal{F}_{\mathcal{B}_\lambda^i}^{\Lambda_j}$ for $j = 1, \ldots, s$.

4.6. *Security Bound of the Two-Level Model*

Now we derive the security bound of the two-level model from the above attacks. By the definition of g^Λ in Eq. (10), for a certain $\mathcal{B}_\lambda^i$, $g^\Lambda(\widetilde{\mathcal{B}_{t_j}^{\prime i}})$ is a fixed value not depending on $\mathcal{X}^i$. Consequently, g^Γ has no influence on $\Delta(\mathcal{F}_{\mathcal{B}_\lambda^i}^{\Gamma})$. Thus, we have the data complexity[9]

$$\mathcal{N}_{\mathcal{B}'} = \frac{4\kappa \log 2}{\mathrm{E}\left[\Delta\left(\mathcal{F}_{\mathcal{B}_\lambda^i}^{\Gamma}\right)\right]}. \tag{11}$$

Now let us look at the time complexity of the attack. From the expression of $\mathcal{F}_{\mathcal{B}_\lambda^i}^{\Gamma}$, it can be easily verified that this expression fulfills the Theorem 11, so our attack can also use the FWT to get the optimal time complexity. For all the subkeys $K = (K_1, K_2) \in \mathrm{GF}(2)^{wt(\lambda)n-1} \times \mathrm{GF}(2)$, where K_1 and K_2 are defined in Sect. 4.4, we define $\mathcal{H}, \mathcal{H}'$ as follows:

[8] P_{j*} represents the $s-1$-dimensional marginal distribution, where $j \in \mathrm{GF}(2)^{s-1}$.

[9] $\mathrm{E}[\Delta(h_{\mathcal{B}_t'^i}^{\Gamma})]$ dose not depend on t.

$$\mathcal{H}(K) = \sum_{i=1}^{\mathcal{N}_{\mathcal{B}'}} \mathbf{1}_{L'_{t_1}(P^i),\ldots,L'_{t_n}(P^i)=K_1 \text{ and } (x_1,\ldots,x_s)=(K_2,1,\ldots,1)},$$

$$\mathcal{H}'(K) = \begin{cases} 0, & \text{if } \prod_{j=1}^{n} \epsilon \left(h_{K_{1,j}}^{\Lambda_1}\right) = 0 \\ \log 2^s D_{\mathcal{F}_{K_{1\lambda}}^{\Gamma}}((K_2, 1, \ldots, 1) \oplus (y_1, \ldots, y_s)), & \text{otherwise} \end{cases}$$

where $x_j = \mathcal{G}_3(\gamma_j) \cdot (Z_{t'}^i \oplus \mathcal{L}'_{t'}(P^i)) \oplus \omega_j \cdot L_2(P^i)$ and $y_j = \bigoplus_{i=1}^{n} g^{\Lambda_j}(K_{1,i})$ for $j = 1, \ldots, s$.

In Algorithm 1, the grade $G(K)$ is a simple convolution between $\mathcal{H}$ and $\mathcal{H}'$ (also in [19]), thus we have $G(K) = \frac{1}{2^l}\widehat{\mathcal{H}''(K)}$ where $\mathcal{H}''(K) = \widehat{\mathcal{H}}(K) \cdot \widehat{\mathcal{H}'}(K)$. Therefore, the total time complexity is $\mathcal{N}_{\mathcal{B}'} + \kappa \cdot 2^{\kappa+1}$.

In order to give the security bound of the two-level model, we first consider the bitwise approximation, i.e., Eq. (9). Let the largest conditional correlation at the first level in the model be $\epsilon_{max,1}$ and the largest unconditional correlation at the second level be $\epsilon_{max,2}$ following the restricted consistency of the linear mask $\mathcal{G}_3(\gamma)$. The total correlation of the linear approximation can be derived as $\epsilon_t = \epsilon_{max,1}^n \cdot \epsilon_{max,2}$. Hence, the required data complexity is

$$\mathcal{N}_{\mathcal{B}'} = \frac{4\kappa \log 2}{\epsilon_{max,1}^{2n} \cdot \epsilon_{max,2}^2}, \tag{12}$$

and the total time complexity is

$$T = \frac{4\kappa \log 2}{\epsilon_{max,1}^{2n} \cdot \epsilon_{max,2}^2} + \kappa \cdot 2^{\kappa+1}. \tag{13}$$

From Eqs. (12) and (13), to strengthen the security of the two-level model, we can take some strategy to reduce the conditional correlation $\epsilon_{max,1}$ and the unconditional correlation $\epsilon_{max,2}$ to the extent that the resultant $T > 2^\kappa$ and/or $\mathcal{N}_{\mathcal{B}'} > 2^\kappa$ for $\kappa = wt(\lambda)n + 1$.

Remarks. Let us take a closer look at the linear approximation process of the above attack in Sect. 4.3. The key reason that the linear approximation at the first level and that at the second level of the model can be connected together and efficiently exploited is that the affine transform $\mathcal{G}_3$ only permutes the keystream bits without any linear combination among them. Thus, each permutated keystream bit is associated with only 1 noise variable from the FSM and when combined together at the second level, there are n noise variables from the FSM at the first level and 1 noise variable from the second level, which ultimately determine the conditional and unconditional bias. Therefore, to reduce the correlations, the most efficient strategy is to increase the number of noise variables from the FSM by some method. What we suggest to reach this aim is as follows.

If we run the core generator at the second level for a number of ticks first without outputting the keystream, i.e., drop off some amount of the keystream prefix at the beginning of the second level, then the noise variables from the first level will propagate and increase the associated number of noise variables for each LFSR variable with the executing of the n LFSRs. It is expected that with an appropriate choice of the number

of the dropping off keystream bits at the beginning of the second level, the correlations will be reduced to the desirable extent so that the corresponding time/data complexities of Algorithm 1 will exceed the security bound. □

The following theorem gives the relation between the suggested number of keystream bits dropped off at the beginning of the second level and the correlations.

Theorem 13. *Let θ be the largest unconditional correlation at the first level in the model and $\rho_i = wt(p_i(x)) - 1$ be the number of tap positions of LFSR R_i $(1 \le i \le n)$ of the core generator in the model, and then after dropping off tL_n $(t \ge 1)$ keystream bits at the beginning of the second level, we have*

$$\epsilon_{uc} \le \theta^{t \cdot \sum_{i=1}^{n} \rho_i + 1}, \tag{14}$$

where ϵ_{uc} is the unconditional correlation of the linear approximation of the model in Eq. (7) in Sect. 4.3; further, if we take the conditional correlation into account at the first level, we have

$$\epsilon_{hybrid} \le \epsilon_{max,1}^{t \cdot \sum_{i=1}^{n} \rho_i} \cdot \epsilon_{max,2}, \tag{15}$$

for the hybrid linear approximation in Eq. (9) in Sect. 4.3.

Proof. First note that for each LFSR i in the generator, the new variable introduced by the LFSR clocking in the model is the xor of ρ_i initial permuted variables at the end of the first level, thus after 1 full circle clocking of the LFSR, i.e., L_i ticks, each variable in the current internal state of LFSR i depends on ρ_i initial permuted variables. Second, after dropping off tL_n $(t \ge 1)$ keystream bits at the beginning of the second level, i.e., after t full circles of the underlying LFSR, each new variable in the current internal state of LFSR i depends on $t \cdot \rho_i$ initial permuted variables in the underlying LFSR.

Besides, from $z_t = \Psi_t \oplus \bigoplus_{i=1}^{n} b_t^i$ and $L_1 < L_2 < \cdots < L_n$, we know that after dropping off tL_n $(t \ge 1)$ keystream bits at the beginning of the second level, the quantity $\bigoplus_{i=1}^{n} b_t^i$ is associated with at least $t \cdot \sum_{i=1}^{n} \rho_i$ initial permuted variables at the beginning of the second level. Then, by the Eqs.(7) and (9), we complete the proof. □

Let $|K| = \kappa$ and $|IV| = \varsigma$, and this theorem indicates that to strengthen the security of the two-level model, it suffices to discard some keystream prefix at the beginning of the second level. To frustrate Algorithm 1, it suffices to reduce the involved correlations to the extent that either the data complexity (the number of required frames) in Eq. (12) or the time complexity in Eq. (13) exceed the security bounds, i.e., $T > 2^{\kappa}$ or $\mathcal{N}_{\mathcal{B}'} > 2^{\varsigma}$, which leads to the following criterion for the two-level model in Fig. 1.

– It is necessary to discard the first tL_n keystream bits at the beginning of the second level in the model for some $t \ge 1$.

Now we are ready to look at the real-world Bluetooth encryption scheme, one instance of the above two-level model, and study its security against the outlined conditional correlation attacks (Fig. 4).

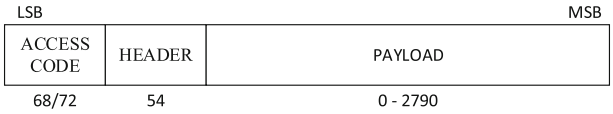

Fig. 4. General basic rate packet format.

5. Description of Bluetooth Encryption

Bluetooth is a wireless technology standard for exchanging data over short distances (using short-wavelength UHF radio waves in the ISM band from 2.4 to 2.485 GHz) from fixed and mobile devices, and building personal area networks (PANs). In order to provide usage protection and information confidentiality, the system applies security measures at both the application layer and the link layer. These measures are designed to be appropriate for a peer environment. Before introducing the encryption of Bluetooth device, we first describe the packets used by the Bluetooth devices. The general packet format of Basic Rate packets is shown in the following figure. The access code is 72 or 68 bits, and the header is 54 bits. The payload ranges from zero to a maximum of 2790 bits. User information can be protected by encryption of the packet payload; the access code and the packet header shall never be encrypted. The security mechanisms in Bluetooth have three phases: Legacy, Secure Simple Pairing, and Secure Connections, shown in the following table. The encryption of the payload is carried out with a stream cipher, called E0, that shall be re-synchronized for every payload. The description here is according to the official specification in [3]. The size of the secret key used in two-level E0 is 128 bits, and the IV consists of 74 bits, 26 of which are derived from a real time clock, while the remaining 48 address bits are depending on users. The core is a modification of the summation generator with 4-bit memory, i.e., $\sigma_t = X_t = (c_{t-1}, c_t) = (c^1_{t-1}, c^0_{t-1}, c^1_t, c^0_t)$, as shown in Fig. 5.

The core keystream generation of E0

Processing:
1: $z_t = b^1_t \oplus b^2_t \oplus b^3_t \oplus b^4_t \oplus c^0_t$
2: $s_{t+1} = (s^1_{t+1}, s^0_{t+1}) = \lfloor \frac{b^1_t + b^2_t + b^3_t + b^4_t + 2c^1_t + c^0_t}{2} \rfloor$
3: $c^0_{t+1} = s^0_{t+1} \oplus c^0_t \oplus c^1_{t-1} \oplus c^0_{t-1}, c^1_{t+1} = s^1_{t+1} \oplus c^1_t \oplus c^0_{t-1}$
4: $(c_{t-1}, c_t) \leftarrow (c_t, c_{t+1})$
5: update the LFSRs

Precisely, the keystream generator consists of four regularly clocked LFSRs whose lengths are 25, 31, 33 and 39 bits, respectively (128 bits in total). The LFSRs are indexed in the order of increasing length. All the feedback polynomials are primitive and have 5 nonzero terms each. Their outputs are combined by a Finite State Machine (FSM) with 4 bits memory. At each time t, the following steps are executed (Table 1).

It is easy to see that the four LFSRs are equivalent to a single 128-bit LFSR whose output bit R_t is obtained by xoring the outputs of the four basic LFSRs, i.e., $R_t =$

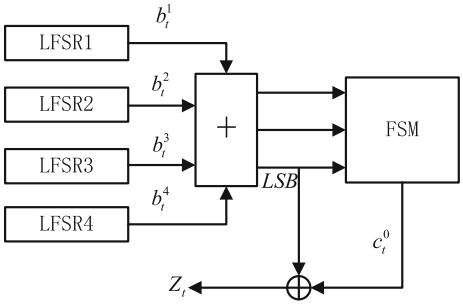

Fig. 5. The core keystream generator of E0.

Table 1. Security algorithms.

Security mechanisms	Legacy	Secure simple pairing	Secure connections
Encryption	E0	E0	AES-CCM

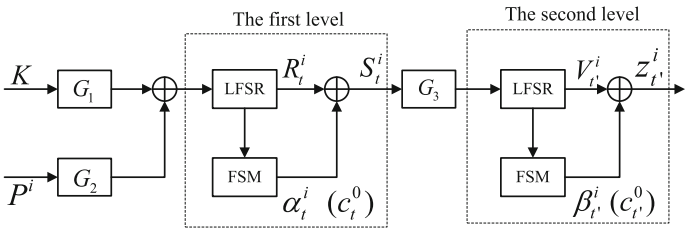

Fig. 6. The real two-level bluetooth encryption scheme.

$b_t^1 \oplus b_t^2 \oplus b_t^3 \oplus b_t^4$ and $z_t = R_t \oplus c_t^0$. Next, we introduce the real two-level E0 scheme, as shown in Fig. 6. As before, we refer the time instant t and t' to the context of E0 level one and level two and denote c_t^0, $c_{t'}^0$ by α_t, $\beta_{t'}$, respectively.

1. *(The first level)* The LFSRs are preset to zero. Given the secret key K and some IV P^i, the LFSRs are initialized linearly as $R^i_{[-199,\ldots,-72]} = (R^i_{-199}, \ldots, R^i_{-72}) = R^i_{[-199,\ldots,-72]} = G_1(K) \oplus G_2(P^i)$, where G_1 and G_2 are public affine transformations over $\mathrm{GF}(2)^{128}$.
2. The initial 4 memory bits of FSM are all set to 0. After clocking E0 200 times, we only keep the last produced 128-bit output $S^i_{[-127,\ldots,0]} = R^i_{[-127,\ldots,0]} \oplus \alpha^i_{[-127,\ldots,0]}$. Let M be the state transmission matrix of the equivalent LFSR over $\mathrm{GF}(2)^{128}$, i.e., $R^i_{[-127,\ldots,0]} = M^{72}(R^i_{[-199,\ldots,-72]})$. Note that because of the linear functions G_1, G_2 and M, the last 128 bits of R^i_t can be written as $R^i_{[-127,\ldots,0]} = (M^{72} \circ G_1)(K) \oplus (M^{72} \circ G_2)(P^i)$.
3. $S^i_{[-127,\ldots,0]}$ is used to initialized the four LFSRs by a byte-wise affine transformation $G_3 : \mathrm{GF}(2)^{128} \to \mathrm{GF}(2)^{128}$, detailed in Fig. 7, this process can be expressed by $V^i_{[1,\ldots,128]} = G_3(S^i_{[-127,\ldots,0]})$.

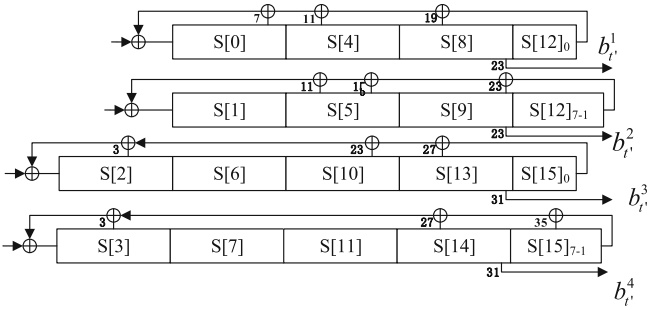

Fig. 7. Distribution of the last 128 bits in the first level.

4. *(The second level)* The FSM initial state remains the same as it was in the end of the first level. Then, E0 produces the keystream $z_{t'}^i = V_{t'}^i \oplus \beta_{t'}^i$ of the i-th frame for $t' = 1, \ldots, 2790$.

6. Previous Attacks on Two-Level E0

At Crypto 2005, Lu, Meier and Vaudenay presented a conditional correlation attack on two-level E0 in [19]. They consider several consecutive steps of the generator as a vectorial function and investigate the conditional correlation properties of this function. Based on these properties, if an adversary is given a pool of keystream frames generated with the same key and different IVs,[10] a statistical distinguisher can be constructed which could distinguish the biased sample sequence from the other sequences consisting of independently and uniformly distributed variables. The biased sample sequence is characterized by some key-related information, which can then be used to identify the correct encryption key.

Note that in the real E0, there is a delay effect in the FSM state, i.e., the current FSM state at time t always contains the previous bit c_{t-1}^0, which will make a difference in defining the target function in Eq. (2) to increase the involved time instants by 1, as shown below. Thus, the adversary could gain one time instant for free and reduce the guess space by 4 bits for free. Precisely, there are two sets of inputs to the FSM in E0 encryption scheme at time t, i.e., the four LFSR output bits $B_t = (b_t^1, b_t^2, b_t^3, b_t^4)$ and the 4 memory register bits $X_t = (c_{t-1}, c_t) \in \mathrm{GF}(2)^4$. Consider l continuous time instants and let $\gamma = (\gamma_0, \gamma_1, \ldots, \gamma_{l-1}) \in \mathrm{GF}(2)^l$ be a linear mask with $\gamma_0 = \gamma_{l-1} = 1$ and $\bar{\gamma} = (\gamma_{l-1}, \gamma_{l-2}, \ldots, \gamma_0)$ be the linear mask in reverse order. Define the inputs as

$$\mathcal{B}_{t+1} = B_{t+1} B_{t+2} \cdots B_{t+l-2} \in \mathrm{GF}(2^{4(l-2)}),\ X_{t+1} = (c_t, c_{t+1}) \in \mathrm{GF}(2)^4$$

[10] As the P^is are affine transformation of a 26-bit clock and a master device address, the maximum number of available frames for a fixed key is 2^{26}.

and the FSM outputs $C_t = (c_t^0, \ldots, c_{t+l-1}^0)$. Then, the function $h^{\gamma}_{\mathcal{B}_{t+1}} : X_{t+1} \to \gamma \cdot C_t$ is also well defined. It is shown in [19] that given $\mathcal{B}_{t+1}$, $\gamma \cdot C_t$ is heavily biased for properly chosen linear mask γ.

A statistical distinguisher can thus be constructed based on the biased distribution of $\gamma \cdot C_t$. Since $\mathcal{B}_{t+1}$ is the key-related material, the adversary can guess the involved key information and collect a set of sample sequences from the keystream, IVs and the guessed key value. It is expected that with the correct key, the corresponding sample sequence is biased, while for the wrong guesses, the underlying sequence will behave like a random source. By properly choosing the involved parameters, it is shown that the original encryption key K in Fig. 6 can be retrieved with 2^{38} on-line computations, 2^{38} off-line computations and 2^{33} memory, given the first 24 bits of $2^{23.8}$ frames in theory, while in practical experiments, the attack needs about 19-hour on-line time, 37-hour pre-computation for each key and 64GB storage, given the first 24 bits of 2^{26} frames.

In [19], it is mentioned that this attack was verified only 30 times for a fixed master key with 2^{26} frames, slightly less than the theoretical estimate $2^{26.5}$ frames. Further, for each possible key, there are 256 equivalent keys, which means that when using a distinguisher to determine the rank of each possible key, there are 256 equivalent candidates having the same grade. In the case that the correct key does not have the highest grade, much more time is needed to search over all the possible keys, e.g., if the grade of the right key ranks the 10th position, then we have to search $10 \cdot 256 \approx 2^{11.3}$ possible keys to find the real one. Thus, the successful probability of this attack cannot be guaranteed.

In the following sections, we will show that the adversary need not to guess all the bits in the condition vector $\mathcal{B}_{t+1}$, actually only a few bits determine the magnitude of the biased distribution of $\gamma \cdot C_t$, and thus we just need to select a condition mask to determine the most important bits in $\mathcal{B}_{t+1}$. In this way, the time/memory complexities of the above attack can be considerably reduced.

7. Correlations Properties of the Two-Level Bluetooth Encryption

In this section, we will carefully study the correlation properties of the two-level encryption scheme. First, a powerful complete recursive formula to compute the unconditional correlations of E0 is presented. Then, the conditional correlation properties based on condition masking are analyzed and computed.

7.1. *Unconditional Correlations in the E0 Keystream Generator*

In [21,22], a recursive formula for the computation of the unconditional correlation in the E0 combiner is presented. However, it only involves the pure FSM variables and cannot cover all the unconditional correlations reported in [10], e.g., the following correlation also has the largest correlation

$$\epsilon\left(c_t^0 \oplus c_{t+1}^0 \oplus c_{t+2}^0 \oplus c_{t+3}^0 \oplus b_{t+3}^1 \oplus b_{t+3}^2 \oplus c_{t+4}^0\right) = -\frac{25}{256},$$

but it cannot be found by the previous formula. Another example is

$$\epsilon(c_t^0 \oplus c_{t+1}^0 \oplus c_{t+2}^0 \oplus b_{t+3}^1 \oplus c_{t+4}^0) = \frac{25}{256}.$$

In general, let $\Omega_1(a, \langle \omega, u \rangle)$ be the correlation $\epsilon(a \cdot s_{t+1} \oplus \omega \cdot c_t \oplus u \cdot B_t)$ shown in Table 2, where $a \in \mathrm{GF}(2)^2$, $u \in \mathrm{GF}(2)^4$, $\omega \in \mathrm{GF}(2)^2$ and B_t denote the output bits of four LFSRs at time t. Besides, let $h : (x^1, x^0) \to (x^0, x^1 \oplus x^0)$ be a permutation over $\mathrm{GF}(2)^2$ and $\epsilon(\langle a_1, u_1 \rangle, \ldots, \langle a_{d-1}, u_{d-1} \rangle, a_d) = \epsilon(a_1 \cdot c_1 \oplus u_1 \cdot B_1 \oplus \cdots \oplus a_{d-1} \cdot c_{d-1} \oplus u_{d-1} \cdot B_{d-1} \oplus a_d \cdot c_d)$, where $u_1, \ldots, u_{d-1} \in \mathrm{GF}(2)^4$. We can derive the above correlation coefficient as follows.[11]

$$\begin{aligned}
\epsilon(\langle 1,0\rangle, \langle 1,0\rangle, \langle 1,0\rangle, \langle 0,1\rangle, 1) &= \sum_{\omega} \Omega_1(1, \langle \omega, 1\rangle) \cdot \epsilon(\langle 1,0\rangle, \langle 1,0\rangle, \langle 2,0\rangle, 1 \oplus \omega) \\
&= -\frac{1}{4} \cdot \epsilon\left(\langle 1,0\rangle, \langle 1,0\rangle, \langle 2,0\rangle, 2\right) \\
&= -\frac{1}{4} \sum_{\omega} \Omega_1(2, \langle \omega, 0\rangle) \cdot \epsilon(\langle 1,0\rangle, \langle 0,0\rangle, \omega) \\
&= \frac{1}{16} \cdot \epsilon(\langle 1,0\rangle, \langle 0,0\rangle, 1) + \frac{5}{32} \cdot \epsilon(\langle 1,0\rangle, \langle 0,0\rangle, 2) \\
&= \frac{1}{16} \sum_{\omega_1} \Omega_1(1, \langle \omega_1, 0\rangle) \cdot \epsilon(\langle 2,0\rangle, 1 \oplus \omega_1) \\
&\quad + \frac{5}{32} \sum_{\omega_2} \Omega_1(2, \langle \omega_2, 0\rangle) \cdot \epsilon(\langle 0,0\rangle, 2 \oplus \omega_2) \\
&= \frac{5}{32} \left(\frac{1}{4} \epsilon(\langle 0,0\rangle, 3) + \frac{5}{8} \epsilon(\langle 0,0\rangle, 0) \right) = \frac{25}{256}.
\end{aligned}$$

Our complete formula for the computation of unconditional correlations of E0 is yielded in the following theorem.

Theorem 14. *If the initial states of FSM and LFSR are both uniformly distributed, then we have*

$$\begin{aligned}
\epsilon(\langle a_1, u_1\rangle, \ldots, \langle a_{d-1}, u_{d-1}\rangle, a_d) = &- \sum_{\omega \in GF(2)^2} \Omega_1(a_d, \langle \omega, u_{d-1}\rangle) \\
&\cdot \epsilon(\langle a_1, u_1\rangle, \ldots, \langle a_{d-2} \oplus h(a_d), u_{d-2}\rangle, a_{d-1} \oplus a_d \oplus \omega).
\end{aligned}$$

Proof. First note that according to the description of the real two-level E0, we can regard the initial states of the LFSRs and the FSM as uniformly distributed random variables, and thus the premise is met. To simplify the analysis, let $Z \in \mathrm{GF}(2)^2$ be a random variable independent of B_{d-1} with uniform distribution. By the keystream generation of

[11] Note that there is a little difference from Sect. 4. Here, we divided the 4 memory bits $\sigma_t = X_t$ into (c_{t-1}, c_t).

Table 2. Biases of all the linear combinations of $\Omega_1(a, \langle u, \omega\rangle)$.

$\Omega_1(a, \langle\omega, u\rangle)$	ω																				
	0					1					2					3					
weight of u	0	1	2	3	4	0	1	2	3	4	0	1	2	3	4	0	1	2	3	4
a																				
0	-1	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
1	–	–	–	–	–	–	–	–	–	–	$\frac{1}{4}$	–	$-\frac{1}{4}$	–	$\frac{1}{4}$	–	$-\frac{1}{4}$	–	$\frac{1}{4}$	–
2	–	$-\frac{1}{4}$	–	–	–	$-\frac{1}{4}$	–	–	–	$\frac{1}{4}$	$-\frac{5}{8}$	–	$\frac{1}{8}$	–	$-\frac{1}{8}$	–	$\frac{1}{8}$	–	$-\frac{1}{8}$	–
3	$\frac{5}{8}$	–	$-\frac{1}{8}$	–	$\frac{1}{8}$	–	$-\frac{1}{8}$	–	$\frac{1}{8}$	–	–	$-\frac{1}{4}$	–	–	–	$-\frac{1}{4}$	–	–	–	$\frac{1}{4}$

E0 in Sect. 5, define $f : \mathrm{GF}(2)^4 \times \mathrm{GF}(2)^2 \to \mathrm{GF}(2)$ and $g : \mathrm{GF}(2)^{6(d-2)+2} \to \mathrm{GF}(2)^2$ as follows.

$$f(X, g(Y)) = a_d \cdot s_d \oplus u_{d-1} \cdot B_{d-1},$$

where $g(Y) = c_{d-1}$, $X = B_{d-1}$ and

$$Y = (\langle c_1, B_1\rangle, \ldots, \langle c_{d-3}, B_{d-3}\rangle, \langle c_{d-2}, B_{d-2}\rangle, c_{d-1}),$$
$$v = (\langle a_1, u_1\rangle, \ldots, \langle a_{d-3}, u_{d-3}\rangle, \langle a_{d-2} \oplus h(a_d), u_{d-2}\rangle, a_{d-1} \oplus a_d).$$

With this simplified expression, we have:

$$\begin{aligned}
&\sum_{\omega\in GF(2)^2} \Omega_1(a_d, \langle\omega, u_{d-1}\rangle) \cdot \epsilon(\langle a_1, u_1\rangle, \ldots, \langle a_{d-2} \oplus h(a_d), u_{d-2}\rangle, a_{d-1} \oplus a_d \oplus \omega) \\
&= \sum_{\omega\in GF(2)^2} \epsilon(f(X, Z) \oplus \omega \cdot Z) \cdot \epsilon(\omega \cdot g(Y) \oplus v \cdot Y) \\
&= \sum_{\omega,x,z} P(X = x, Z = z) \cdot (-1)^{f(x,z)\oplus\omega\cdot z} \sum_y P(Y = y) \cdot (-1)^{\omega\cdot g(y)\oplus v\cdot y} \\
&= \sum_{x,z,y} P(X = x, Z = z) P(Y = y) (-1)^{f(x,z)\oplus v\cdot y} \sum_\omega (-1)^{\omega\cdot(z\oplus g(y))} \\
&= \sum_{x,y} P(X = x, Y = y) (-1)^{a_d\cdot s_d \oplus u_{d-1}\cdot B_{d-1}\oplus v\cdot y} \\
&= \sum_{x,y} P(X = x, Y = y) (-1)^{a_1\cdot c_1 \oplus u_1\cdot B_1 \oplus \cdots \oplus u_{d-1}\cdot B_{d-1}\oplus a_d\cdot c_d} \\
&= -\epsilon(\langle a_1, u_1\rangle, \ldots, \langle a_{d-1}, u_{d-1}\rangle, a_d). \qquad (16)
\end{aligned}$$

Equation (16) is derived according to $a_d \cdot c_d = a_d \cdot s_d \oplus a_d \cdot c_{d-1} \oplus h(a_d) \cdot c_{d-2}$. □

Theorem 14 can compute all the unconditional correlations of the E0 combiner without any miss, e.g., it covers all the results reported in [10]. Then, we can compute $\epsilon(v \cdot Z_{t'}^m \oplus$

Table 3. The two largest correlations and the number of different linear masks when $m = 7, \ldots, 14$.

m	Value	Number	m	Value	Number
7	0.03814697	16	11	0.01680851	32
	0.02441406	640		0.01337528	32
8	0.03839111	16	12	0.0148296356201	6
	0.03814697	16		0.0101804733276	10
9	0.03662109	16	13	0.0096201896667	10
	0.01642704	16		0.0095695257187	6
10	0.01451969	16	14	0.0095968134701	12
	0.01413822	16		0.0095404870808	16

$W \cdot B_{t'}^m$) for m up to 14 with a low complexity, which is impossible for the exhaustive search method by FWT due to the large time and memory complexities, see [17] for more on Walsh Transforms. With Table 2 and the initial conditions $\epsilon((0,0),0) = -1$, and $\epsilon((a,b),c) = 0$ for $(a,b,c) \neq (0,0,0)$, we can recursively deduce the unconditional correlations. Table 3 gives some searching results of the unconditional correlations. From the table, we found that with the increasing of m, the unconditional correlations become more and more smaller. Even though these correlations cannot be used here to improve the attack of E0, they can be applied to the attack in [10] and maybe have some better results.

Corollary 15. *The recursive expression in Theorem 14 can compute all the correlation coefficients* $\epsilon(W \cdot B^m \oplus v \cdot Z^m)$ *of the second level of E0 in* $2^{m-2} \cdot 5^{m-2}$ *iterations.*

Proof. In Theorem 14, c_t^1 is usually not considered, so we only consider $a_t = 0, 1$ for $1 \leq t \leq m$. For a certain m, we have $a_1^0 = 1$ and $a_m^0 = 1$. From the initial values, if $u_1 \neq 0$, then the total correlation will be zero, so we always set $u_1 = 0$. Now we just search over all the $m-2$ undetermined coefficients $a_i^0 \in GF(2)$ and $u_i \in GF(2)^4$ for $i = 2, \ldots, m-1$ which only needs $2^{m-2} \cdot 5^{m-2}$ iterations. □

The unconditional linear correlations are used in the linear approximation of the second level in Fig. 6. Since there is no linear relation between the input to the FSM in the second level and the original encryption key, the conditional correlations cannot be used in the approximation of the second level.

7.2. *New Conditional Correlations*

As condition mask indicates that the adversary may not use the full vector as the condition, only search the correlations conditioned on a subset of $\mathcal{B}$ defined by a mask λ. In the cryptanalysis of E0, $\mathcal{B}_{t+1}$ is the key-related input. Given a condition mask $\lambda = (\lambda_{t+1}, \ldots, \lambda_{t+l-2}) \in \mathrm{GF}(2)^{4(l-2)}$, where $\lambda_j \in \mathrm{GF}(2)^4$ corresponds to B_j for $j = t+1, \ldots, t+l-2$. According to Eq. (2) and the difference between the model and the real E0 mentioned in Sect. 6, we can construct the target function of E0 as

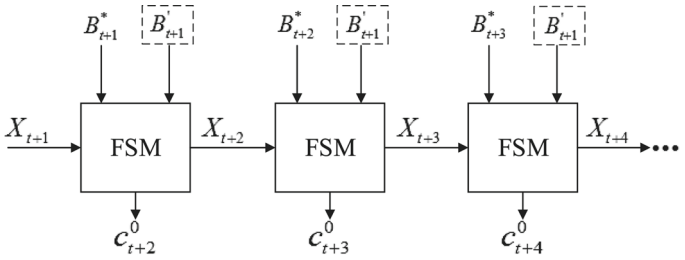

Fig. 8. The new computation process of C_t in the real bluetooth encryption.

Table 4. The bias with $\Lambda = (\gamma, \eta) = (0x1f, \mathbf{0}_{|\eta|})$ and $\lambda = 0x00f$.

$\epsilon\left(h^{\Lambda}_{\mathcal{B}'_{t+1}}\right)$	$wt(B_{t+3})$	Cardinality of B_{t+3}
0.390625	2	6
−0.390625	0, 4	2
0.0625	3	4
−0.0625	1	4

$$h^{\Lambda}_{\mathcal{B}'_{t+1}} : X_{t+1}, \mathcal{B}^*_{t+1} \rightarrow \gamma \cdot C_t \oplus \omega \cdot \mathcal{B}^*_{t+1}, \tag{17}$$

as shown in Fig. 8. Figure 8 also shows that though the computation process of C_t is frustrated by the condition mask $\lambda \neq \mathbf{1}_u$, the bias can still be computed. Here, comes an example to illustrate how to compute the bias in the condition masking setting. Assume $l = 4$ and $\lambda = \texttt{0x0f}$,[12] we have $\mathcal{B}_{t+1} = B_{t+1}B_{t+2}$, $\mathcal{B}'_{t+1} = B_{t+2}$ and $\mathcal{B}^*_{t+1} = B_{t+1}$. We can guess B_{t+2} and compute $h^{\Lambda}_{B_{t+2}}$ for all the possible choices of B_{t+1}, X_{t+1} to get $\epsilon(h^{\Lambda}_{B_{t+2}})$.

According to the specification of E0, we can construct the linear functions L_t and L'_t of Eq. (3). With this knowledge, the linear equations on the original encryption key can be acquired. For $4 \leq l \leq 6$, we have exhaustively searched the correlations based on condition masking for all the possible condition masks on a PC. All the significant biases obtained are also verified in computer simulations working on sufficiently long output sequences. The time complexity of guessing is determined by $wt(\lambda)$. To get better time/memory complexities, we restrain ourselves to the λs satisfying $1 \leq wt(\lambda) \leq 7$.

In the experiments, we have found many important masks, shown in Tables 4 and 5. Table 4 is computed with $\lambda = \texttt{0x00f}$, $\Lambda = (\gamma, \eta) = (\texttt{0x1f}, \mathbf{0}_{|\eta|})$. We get $\mathrm{E}[\Delta(h_{\mathcal{B}'_{t+1}})] \approx 2^{-3.7}$, where $\mathcal{B}'_{t+1} = B_{t+3}$. In Table 5, we choose $\lambda = \texttt{0x007f}$ and $\Lambda = (\gamma, \eta) = (\texttt{0x21}, \mathbf{0}_{|\eta|})$. From it, we get $\mathrm{E}[\Delta(h_{\mathcal{B}'_{t+1}})] \approx 2^{-3.5}$, where $\mathcal{B}'_{t+1} = B_{t+3}B_{t+4}$.

Moreover, as mentioned in Sect. 4.2, for a fixed condition mask λ, its maximum bias among all the linear masks Λ is an essential measure of it. The larger the maximum bias, the better the condition mask is. The following property indicates how to choose

[12] For brevity, we use the hexadecimal number to represent a vector.

Table 5. The bias with $\Lambda = (\gamma, \eta) = (0x21, \mathbf{0}_{|\eta|})$ and $\lambda = 0x007f$.

$\epsilon\left(h^{\Lambda}_{\mathcal{B}'_{t+1}}\right)$	$wt(B_{t+3})wt(B_{t+4})$	Cardinality of $B_{t+3}B_{t+4}$
0.46875	10, 20, 14, 24	12
−0.46875	12, 22	36
0.15625	00, 30, 04, 34	4
−0.15625	02, 32	12
0.046875	01, 33	8
−0.046875	31, 03	8
0.015625	11, 33	24
−0.015625	21, 13	24

the condition mask to make the bias as large as possible. We have verified this property by searching over all the biases of $h^{\Lambda}_{\mathcal{B}'}$ for each combination of λ, γ and ω.

Property 16. *Let* $\mathcal{B}_{t+1} = B_{t+1} \cdots B_{t+l-2} \in GF(2)^{4(l-2)}$, *and* $\lambda = (\lambda_{t+1}, \ldots, \lambda_{t+l-2})$, $\lambda' = (\lambda'_{t+1}, \ldots, \lambda'_{t+l-2})$ *are two condition masks with* $4 \leq l \leq 6$ *and* $wt(\lambda) = wt(\lambda') \geq 4$, *where* $\lambda_i, \lambda'_i \in GF(2)^4$ *correspond to* B_i. *If* $wt(\lambda_{t+l-2}) = 4$ *and* $wt(\lambda'_{t+l-2}) < 4$, *then*[13] $max_{\Lambda}(E[\Delta(h^{\Lambda}_{\mathcal{B}'_{t+1}})]) > max_{\Lambda'}(E[\Delta(h^{\Lambda'}_{\mathcal{B}'_{t+1}})])$, *except when* $l = 4$, $wt(\lambda_{t+1}) = 1$, $wt(\lambda_{t+2}) = 4$ *and* $wt(\lambda'_{t+1}) = 2$, $wt(\lambda'_{t+2}) = 3$, *in which case the maximum values are equal.*

From Property 16, $wt(B_{t+l-2})$ in $\mathcal{B}_t$ plays the most important role in the correlation values based on condition masking. This weight determines the magnitude of the corresponding bias in the condition masking case. For example, given $l = 5, \Lambda = (\gamma, \eta) = (\mathtt{0x1f}, \mathbf{0}_{|\eta|}), \lambda_1 = \mathtt{0x303}, \lambda_2 = \mathtt{0x00f}$, we can find $\mathrm{E}[\Delta(h^{\Lambda}_{\mathcal{B}_1})] = 0.020325, \mathrm{E}[\Delta(h^{\Lambda}_{\mathcal{B}_2})] = 0.078247$, where $\mathcal{B}_1$ and $\mathcal{B}_2$ are the two condition vectors determined by λ_1 and λ_2, respectively. The experimental results are depicted in the Figs. 9 and 10.[14] to show the different levels of the correlation magnitude. The fact depicted in the figures that the conditional correlation values are distributed at clearly different levels tells us that when selecting the condition masks, we should set the value of the highest four bits of λ to be $\mathtt{0xf}$. For example, given $l = 5, \eta = \mathbf{0}, \gamma = \mathtt{0x1f}, \lambda_1 = \mathtt{0x303}, \lambda_2 = \mathtt{0x00f}$ and $\lambda_3 = \mathtt{0x113}$, we can find $\mathrm{E}[\Delta(h^{\gamma,0}_{\lambda_1 \diamond \mathcal{B}_{t+1}})] = 0.020325, \mathrm{E}[\Delta(h^{\gamma,0}_{\lambda_2 \diamond \mathcal{B}_{t+1}})] = 0.078247$ and $\mathrm{E}[\Delta(h^{\gamma,0}_{\lambda_3 \diamond \mathcal{B}_{t+1}})] = 0.010162$. Not only one example, but all the experimental results confirm this claim so far.

[13] $max_{\Lambda}(\cdot)$ is the maximum function for all Λ.

[14] λ' in the figures is represented in the inverse nibble order of λ, e.g., $\lambda = 0x007f$, then $\lambda' = 0xf700$.

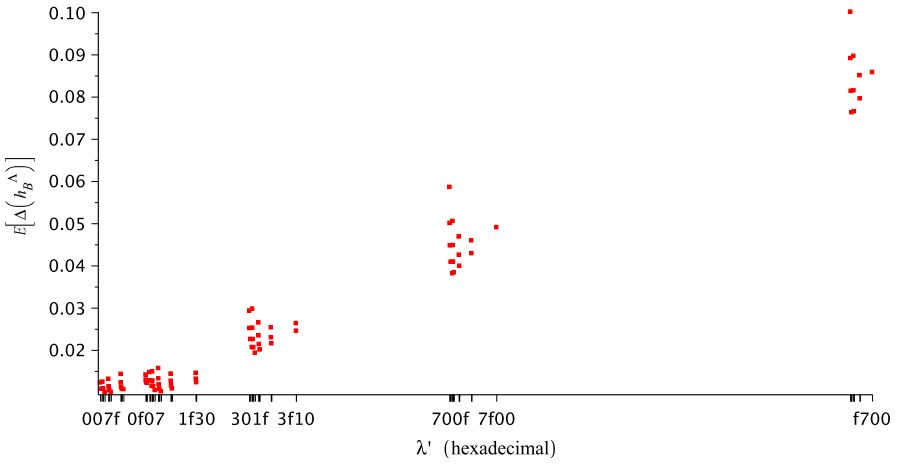

Fig. 9. $\max_{\omega}(h^{\Lambda}_{\mathcal{B}})$ for different condition masks λ with $wt(\lambda) = 7$ and $l = 6$.

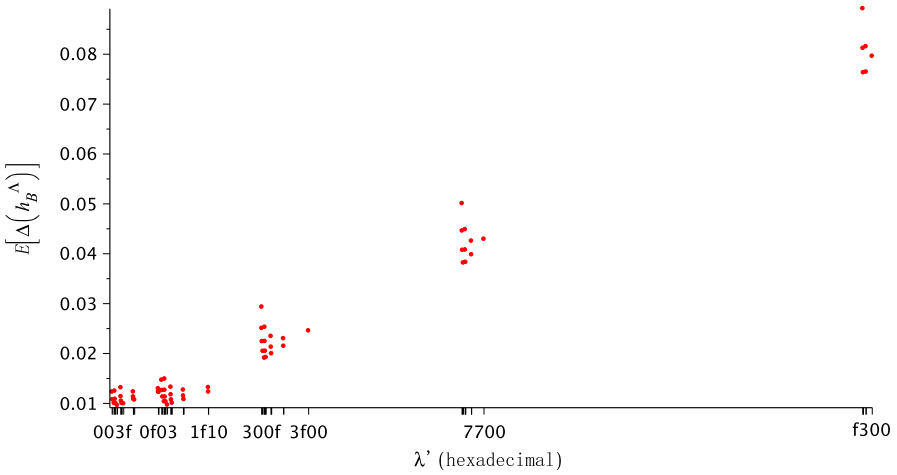

Fig. 10. $\max_{\omega}(h^{\Lambda}_{\mathcal{B}})$ for different condition masks λ with $wt(\lambda) = 6$ and $l = 6$.

8. Our Attacks with Condition Masking

According to the specification in [3], the last generated 128 bits $S^i_{[-127,\ldots,0]}$ in the first level are arranged in octets denoted by $S[0], \ldots, S[15]$, e.g., $S[0] = (S^i_{-127} S^i_{-126} \cdots S^i_{-120})$. According to Sect. 4 and Fig. 7, $V^i_{[1,\ldots,24]}$ can be expressed as

$$V^i_{t'} = U^i_{t'} \oplus \alpha^i_{t_1} \oplus \alpha^i_{t_2} \oplus \alpha^i_{t_3} \oplus \alpha^i_{t_4}, \text{ for } t' = 1, \ldots, 24.$$

Since at level two (in Fig. 6), the 128-bit keystream S^i_t are loaded in the reverse order of that at level one, then Eq. (7) can be rewritten as

$$\bar{\gamma} \cdot (Z^i_{t'} \oplus \mathcal{L}_{t'}(K) \oplus \mathcal{L}'_{t'}(P^i)) = \bigoplus_{j=1}^{4} (\gamma \cdot C^i_{t_j}) \oplus \bar{\gamma} \cdot C^i_{t'}, \tag{18}$$

for $i = 1, \ldots, \mathcal{N}$. Here, we have $t' \in \bigcup_{d=0}^{2}\{8d+1, \ldots, 8d+9-l\}$.[15]

When $t' \in \bigcup_{d=0}^{2}\{8d+1, \ldots, 8d+9-l\}$, by Eq. (2) and (17), we can rewrite this equation as follows:

$$\bar{\gamma} \cdot (Z^i_{t'} \oplus \mathcal{L}_{t'}(K) \oplus \mathcal{L}'_{t'}(P^i)) \oplus \omega \cdot (L_1(K) \oplus L_2(P^i)) = \bigoplus_{j=1}^{4} h^{\Lambda}_{\mathcal{B}'^i_{t_j+1}} \oplus h^{\bar{\gamma}}, \tag{19}$$

where L_1, L_2 are public linear functions in E0. Thus, we acquired the linear approximation of two-level E0 based on condition masking.

8.1. *Key Recovery Attack with Bitwise Linear Approximation*

From Sect. 7, the largest unconditional bias of h^{γ} is $\frac{25}{256}$ with $\gamma = (1,1,1,1,1)$ or $(1,0,0,0,0,1)$. To maximize the bias of Eq. (19), we choose these two γs in the second level approximation, and then $|\gamma| = l = 5$ or 6. Due to the high time/memory complexities, the attack in [19] only considered $l < 6$. In our attack, the time/memory complexities are not dependent on $|\gamma|$, they are determined on $wt(\lambda)$, and thus $l = 6$ can also be used in the condition masking setting.

Given the condition mask λ and the linear masks $\Lambda = (\gamma, \eta)$, in the case of E0 we have

$$\mathcal{B}^i_{\lambda} = \left(\mathcal{B}'^i_{t_1+1}, \mathcal{B}'^i_{t_2+1}, \mathcal{B}'^i_{t_3+1}, \mathcal{B}'^i_{t_4+1}\right),$$
$$\mathcal{X}^i = \left(Y^i_{t_1+1}, Y^i_{t_2+1}, Y^i_{t_3+1}, Y^i_{t_4+1}, X^i_{t'}, \mathcal{B}^i_{t'+1}\right),$$

where $Y^i_{t_j+1} = (X^i_{t_j+1}, \mathcal{B}^*_{t_j+1})$ is the unknown input to $h^{\Lambda}_{\mathcal{B}'^i_{t_j+1}}$, and $X^i_{t'}, \mathcal{B}^i_{t'+1}$ are the inputs to $h^{\bar{\gamma}}$. By the same notations in Sect. 4, we set $K_1 = (L_{t_1}(K), L_{t_2}(K), L_{t_3}(K), L_{t_4}(K))$ be the $4wt(\lambda)$ bits contained in $\mathcal{B}^i_{\lambda}$ and $K_2 = \bar{\gamma} \cdot \mathcal{L}_{t'}(K) \oplus \omega \cdot L_1(K)$ be the subkey. In the case of E0, we have the parameters $n = 4, k = 4$. With the distinguisher $\mathcal{F}^{\Lambda}_{B^i_{\lambda}}(\mathcal{X}^i)$ in Sect. 4 and the keystream sequences of E0, we can restore the correct keys.

8.2. *Key Recovery Attack with the Vectorial Approach*

Now we look at the vectorial approach. Here, we apply the vectorial key recovery attack in Sect. 4 with $n = 4, k = 4$ to the two-level E0. Assume we use s mutually independent linear approximations. Keep the same notations as before. Let $\Gamma = (\Lambda_1, \ldots, \Lambda_s)$ and $\Gamma' = (\bar{\gamma}_1, \ldots, \bar{\gamma}_s)$ denote the linear mask of these s approximations. In particular, Λ_1 is just the linear mask used in the above bitwise attack.

[15] From Eq. (18), the time instant t_j in $C^i_{t_j}$ are continuous, so the approximation is only set up in this requirement.

By the way of Sect. 4 and notice the difference between the general model and the real E0 pointed out in Sect. 6, we can construct a linear approximation of the real two-level E0 in the vectorial approach accordingly. With the appropriate choice of $\Gamma = (\Lambda_1, \ldots, \Lambda_s)$ following the principles in Sect. 4.5, we apply the vectorial key recovery attack in Sect. 4 to the case of E0. We can thus recover the κ secret bits of two-level E0.

8.3. *The Data Complexity*

The bi-biases analysis, i.e., using two bitwise linear approximations to construct a two-dimensional vector, is used in [19] to reduce the data complexity from $2^{26.5}$ to $2^{23.8}$. This method is very similar to the multidimensional linear cryptanalysis. But there are two errors in their analysis. First, the linear masks $\gamma_1 = (1, 1, 0, 1)$, $\gamma_2 = (1, 0, 1, 1)$, are chosen there. But note that the unconditional correlation $\epsilon(h^{\bar{\gamma}_2}) = 0$, so the second dimension of the vector distinguisher is always uniformly distributed. Thus, according to the multidimensional linear cryptanalysis, this vectorial distinguisher has the same SEI with the bitwise distinguisher, i.e., the bitwise linear approximation in the first dimension. This mainly comes from the fact that there are 4-bit memory in the E0 combiner, and thus any 2-bit linear combination has the zero correlation coefficient. Thus, this method cannot improve the data complexity. Second, the formula (31) in [19], i.e., $\mathbf{F}^{\Gamma}_{\mathbf{B}^i} = \Delta(h^{\bar{\Gamma}}) \cdot \mathrm{E}^4(\Delta(h^{\Gamma}_{\mathbf{B}_{t+1}}))$, is not always true. For the bitwise linear approximation, this formula is the same as pilling-up lemma, but for the vectorial method it is not always the case. Hence, the data complexity in [19] should be $2^{26.5}$ rather than $2^{23.8}$. The same problem is also in [34].

The correct way to compute the distribution of $\mathbf{F}^{\Gamma}_{\mathbf{B}^i}$ is to use the convolutional operation to combine each sub-distribution efficiently. This process can be expressed as follows:

$$\Delta(\mathbf{F}^{\Gamma}_{\mathbf{B}^i}) = \Delta\left(h^{\Gamma}_{\mathbf{B}^1_{t+1}} \otimes h^{\Gamma}_{\mathbf{B}^2_{t+1}} \otimes h^{\Gamma}_{\mathbf{B}^3_{t+1}} \otimes h^{\Gamma}_{\mathbf{B}^4_{t+1}} \otimes h^{\Gamma'}\right).$$

Note that the FWT could be used here to accelerate the final computation through the relation between the convolution and the Walsh Transform, detailed in Sect. 2. We have used this method to re-compute all the data complexities in [19] and [34] and found that the advanced algorithm in [19] cannot improve the data complexity, that is the actual data complexity should be $2^{26.5}$ rather than $2^{23.8}$. Besides, the data complexities in [34] are also not so accuracy, though the experiments confirmed the data complexity in practice. But the success probability of the attack is not very high. The main reason is that the same formula as that in [19] is used. In the following sections, we will describe a new method to improve the data complexity based on the condition masking method.

8.4. *Theoretical Analysis*

To get the optimal performance of our attack, we should carefully choose the parameters Γ and λ in the linear approximations. As explained before, each component should not be uniformly distributed. We have searched all the linear masks in [19] and found that there is no other linear approximations that have non-uniform distribution. Thus, the method in [19] cannot be transformed into the vectorial approach. On the other hand,

Table 6. Example: $\lambda = 0x000f$.

λ	γ	η	$E[\Delta(h^{\Lambda}_{\mathcal{B}'_{t+1}})]$
	(1, 0, 0, 0, 0, 1)	0	$2^{-3.7}$
$0x000f$	(1, 0, 0, 0, 1, 1)	0	$2^{-3.7}$

there is more flexibility of the condition masking method in [34] in the sense that much more linear masks are available to construct the vectorial distinguisher. By choosing different condition masks, we can use the multi-pass method and list decoding method to reduce the data complexities efficiently.

The experiments have shown that there are many large correlations based on condition masking that can be used in our attack. For example, for a condition mask $\lambda = \texttt{0x000f}$, we can choose the 2 linear masks listed in Table 6, the experimental results show $\Delta(h^{\Gamma}_{\mathcal{B}'_{t+1}}) \approx 2^{-2.7}$, where $\Gamma = ((\texttt{0x21}, \mathbf{0}), (\texttt{0x23}, \mathbf{0}))$. And $\Delta(h^{\Gamma'}) \approx 2^{-6.7}$, thus from Eq. (11) we know that the data complexity is $\mathcal{N}_{\mathcal{B}'} \approx 2^{27}$. In this example, we can recover the involved $\kappa = 17$-bit subkey. But the data complexity is higher than the real-world Bluetooth bound 2^{26}. We will use the list decoding method to reduce the data complexity to 2^{25}, which is detailed in the next Section. Here, we give the theoretical analysis why we use the data complexity 2^{25} to generate a list of 256 possible candidate keys. According to the LLR method in [2], each possible $K \in \{0,1\}^{17}$ has the grade $G_K = \mathrm{LLR}(\mathcal{F}^{\Gamma}_{K_\lambda})$. We assume that for one unknown value $K = K_0$, each sample $\mathcal{F}^{\Gamma}_{K_\lambda}$ follows the distribution D_0, whereas when $K \neq K_0$, all the $\mathcal{F}^{\Gamma}_{K_\lambda}$ follow the distribution D_1. For any $K \neq K_0$, we obtain that $G_{K_0} - G_K$ is approximately normally distributed with the expected value $\mathcal{N}_{\mathcal{B}'}\Delta(\mathsf{D}_0)$ and the standard deviation $\sqrt{2\mathcal{N}_{\mathcal{B}'}\Delta(\mathsf{D}_0)}$. Hence, the probability that a wrong key K has a better grade than the right key K_0, i.e., $G_{K_0} < G_K$ is about $\Phi(-\sqrt{\mathcal{N}_{\mathcal{B}'}\Delta(\mathsf{D}_0)/2})$, where $\Phi(t) = \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{t} e^{-\frac{1}{2}u^2}du$ is the distribution function of the standard normal distribution. Thus, the expected number of the K having larger grades than the correct key K_0 is $(2^{17}-1)\cdot\Phi(-\sqrt{\mathcal{N}_{\mathcal{B}'}\Delta(\mathsf{D}_0)/2})$. The probability that this value is smaller than 256 is $\Phi(-\sqrt{n_{\mathcal{B}'}\Delta(D_0)/2}) \leq \frac{256}{2^{17}-1}$. By the property of the standard normal distribution, we can acquire $\Phi(\sqrt{\mathcal{N}_{\mathcal{B}'}\Delta(D_0)/2}) \approx 0.998$. Since $\Delta(D_0) \approx 2^{21.44}$, we can get the new value of $\mathcal{N}_{\mathcal{B}'}$ is $2^{25.49}$. That is why, we use the data complexity $\mathcal{N}_{\mathcal{B}'} = 2^{25}$ in the experiments. Furthermore, we can use a new condition mask to repeat the same process to minimize the size of key candidates list, presented in Sect. 9. In the experiments, we find that we can always recover the correct key correctly.

Let us analyze the time complexity of the two examples discussed above. The precomputation of $\widehat{H'}$ is $17 \cdot 2^{17}$, and we need time $2 \cdot 17 \cdot 2^{17} \approx 2^{21.1}$ to compute $\widehat{\mathcal{H}}$, $\widehat{\mathcal{H}''}$, and time $n_{\mathcal{B}'} = 2^{27}$ to compute $\mathcal{H}$, so the total time is $2^{27} + 2^{21.1}$.

9. Practical Implementation of the Known-IV Attack

Our attacks have been fully implemented on one core of a single PC, running with Windows 7, Intel Core 2 Q9400 2.66GHz and 4GB RAM. In general, the exper-

imental results match the theoretical analysis quite well. We present the details as follows.

We choose the condition mask $\lambda = \texttt{0x000f}$ and $\gamma_1 = \texttt{0x21}, \eta_1 = \mathbf{0}, \gamma_2 = \texttt{0x23}, \eta_2 = \mathbf{0}, t' = 1, \mathcal{N}_{\mathcal{B}'} = 2^{25}$ (reduced by the list decoding and multi-pass method) in the experiments. We call this process the first step. Here, the list decoding method means that we select a list of key candidates other than a unique possible key. After the first step, we get a candidate list of size 256. In the second step, we choose another condition mask, $\lambda = \texttt{0x00f}$ and $\gamma = \texttt{0x1f}, \omega = \mathbf{0}$, and then use these parameters to mount a new key recovery attack to reduce the size of key list further. After that, we can always acquire the correct key.

Precisely in this configuration, we have the condition bits $\mathcal{B}'^i_{t+1} = B^i_{t+4}$. In the first step, we first collect $\mathcal{N}_{\mathcal{B}'}$ frames for a random key and store them in a binary file. It takes about 8 minutes and 160MB to fulfill this task. With these samples, we run Algorithm 1 to recover the possible keys stored in a list. The pre-computation of $\mathcal{H}'$ and $\widehat{\mathcal{H}'}$ needs about one second, and the results are stored in a 4MB table in RAM, not on the hard disk. Computing $\mathcal{H}, \widehat{\mathcal{H}}, \mathcal{H}'', \widehat{\mathcal{H}''}$ in total takes about 2 seconds. Compared with the 37 hours and 64GB table in [19], our attack can be easily carried out in real time on a single PC.

Our new attack is repeated 100 times with different randomly generated keys and IVs. In the first step of our experiments, the right key ranks first for 72 times and about 99% of the right keys are in the first 256 key candidates list. Thus, after the first step, we almost have got the right key. Then, we use the new condition masking to recover the right key in this 256 candidates and in 86 times, we get the unique one right key. The remaining experiments can reduce the size of the possible key list further. Note that in [19], the experiments are only carried out in the basic bitwise level with 2^{26} frames and repeated 30 times for a fixed key.

One run of our attack is as follows. We first use two-level E0 to generate 2^{25} frames for the key $\texttt{0x8387cb74a2b0cf437ba6995f74de39e0}$. In the experiments, we use the Mersenne twister to generate the key and the 2^{25} 74-bit IVs, and set the first IV ADR $= \texttt{0x1ad5e266c6fa}$, CLK $= \texttt{0x6260c9}$ as the benchmark, the others are xor values with the benchmark IV. We store the first 24 bits of each frame in a file and compute the $\mathcal{F}^{\Lambda}_{\mathcal{B}^i_{\lambda}}(\mathcal{X}^i), i = 1, \ldots, 2^{25}$. Note that if we choose another IV as the new benchmark IV, the values of new $\mathcal{F}^{\Lambda}_{\mathcal{B}^i_{1\lambda}}(\mathcal{X}^i), i = 1, \ldots, 2^{25}$ are just a permutation of the former ones. Second, we compute $(B^i_{t_1+4}, B^i_{t_2+4}, B^i_{t_3+4}, B^i_{t_4+4})(i \geq 2)$ by $(B^1_{t_1+4}, B^1_{t_2+4}, B^1_{t_3+4}, B^1_{t_4+4}) \oplus (\Delta_{i,1}, \Delta_{i,2}, \Delta_{i,3}, \Delta_{i,4})$, where $\Delta_{i,j}$ denotes the difference value of the i-th frame. Third, for each possible key, we use the FWT to compute the grade $G(K)$. In this instance, the grade of the right subkey $(K_1 = (\texttt{0x5}, \texttt{0xf}, \texttt{0x7}, \texttt{0x3}), K_2 = \texttt{0x1})$ is 8.118578, which ranks the first. In total, the running time is $\mathcal{N}_{\mathcal{B}'} + k \cdot 2^{k+1} \approx 2^{25}$. In order to recover more key bits, we can increase the time instant and using the same method to recover all the key bits. Table 7 gives a comparison of our attacks with the best previous attacks on two-level E0.

Table 7. Comparison of our attack in Sect. 8 with the previous attacks on two-level E0.

Attack	Pre-com	Time	Frames	Memory
[7]	–	2^{73}	–	2^{51}
[8]	2^{80}	2^{65}	2	2^{80}
[10]	2^{80}	2^{70}	45	2^{80}
[20]	–	2^{40}	2^{35}	2^{35}
[19]	2^{38}	2^{38}	$2^{23.8} \rightarrow 2^{26}$	2^{33}
Ours	$2^{21.1}$	2^{25}	2^{25}	2^{17}

10. The Ciphertext-Only Attack

In this section, we convert the attack in Sect. 8 against the real two-level E0 into a ciphertext-only attack, which is much more practical than the above known-IV (known-plaintext) attacks. In the real-world ciphertext-only scenario, the adversary only has access to a set of ciphertext bits $cp_{t'}$ intercepted from the air other than the keystream bit $z_{t'}$.

Note that $z_{t'} = cp_{t'} \oplus m_{t'}$, where $m_{t'}$ is the t'-th real plaintext bit. Let $CP_{t'} = (cp_{t'}, \ldots, cp_{t'+l-1})$ and $M_{t'} = (m_{t'}, \ldots, m_{t'+l-1})$, then the core linear approximation Eq. (19) in Sect. 8 becomes

$$\begin{aligned}\bar{\gamma} \cdot (CP^i_{t'} \oplus \mathcal{L}_{t'}(K) \oplus \mathcal{L}'_{t'}(P^i)) \oplus \omega \cdot (L_1(K) \oplus L_2(P^i)) \\ = \bigoplus_{j=1}^{4} h^{\Lambda}_{\mathcal{B}^i_{t_j+1}} \oplus h^{\bar{\gamma}} \oplus \bar{\gamma} \cdot M^i_{t'} \,.\end{aligned}$$

Further note that the adversary always has some knowledge of the statistical distribution of the plaintext characters. Here, for brevity, we assume that the plaintexts consist of natural English sentences represented by ASCII codes. The ASCII codes and the statistical property of these symbols are listed in Table 8.

In this case, the statistical distribution of the plaintext is usually heavily biased, i.e., $\epsilon(\bar{\gamma} \cdot M^i_{t'}) \neq 0$. Let us denote the corresponding bias by ϵ_M. Besides, assume that the bias of Eq. (19) is ϵ_Z, and then according to the Piling-up Lemma, we can compute the total bias of the above linear approximation in the ciphertext-only attack as $2\epsilon_M\epsilon_Z$, and the other parts of the attack are the same as the previous known-IV attack.

We use the bitwise linear approximation to mount the ciphertext-only attack. The reason that the vectorial approach cannot be applied here is as follows. Assume we use the same parameter configuration as that in the known-plaintext attack, i.e., $\lambda =$ `0x000f`, $\Gamma = ((\texttt{0x21}, \mathbf{0}), (\texttt{0x23}, \mathbf{0}))$. The target distribution in the ciphertext-only attack now becomes $\mathcal{F}^{\Gamma}_{\mathcal{B}_\lambda} \oplus (\gamma_1 \cdot M, \gamma_2 \cdot M)$. By Table 8, we can compute the SEI of the plaintext as $\Delta((\gamma_1 \cdot M, \gamma_2 \cdot M)) = 1.04$, which is much larger than the bitwise bias of plaintext. But when computing the SEI of the distribution of $\mathcal{F}^{\Gamma}_{\mathcal{B}_\lambda} \oplus (\gamma_1 \cdot M, \gamma_2 \cdot M)$ using the convolution method, the SEI decreases very fast and becomes $\Delta(\mathcal{F}^{\Gamma}_{\mathcal{B}_\lambda} \oplus (\gamma_1 \cdot M, \gamma_2 \cdot M)) \approx 2^{-27.19}$ and accordingly the data complexity becomes $\mathcal{N}_{\mathcal{B}'} \approx 2^{32.75}$, which is

Table 8. Relative frequencies of letters in english text.

Letter	ASCII	Frequency (%)	Letter	ASCII	Frequency (%)
a	01100001	8.167	n	01101110	6.749
b	01100010	1.492	o	01101111	7.507
c	01100011	2.782	p	01110000	1.929
d	01100100	4.253	q	01110001	0.095
e	01100101	12.702	r	01110010	5.987
f	01100110	2.228	s	01110011	6.327
g	01100111	2.015	t	01110100	9.056
h	01101000	6.094	u	01110101	2.758
i	01101001	6.966	v	01110110	0.978
j	01101010	0.153	w	01110111	2.360
k	01101011	0.772	x	01111000	0.150
l	01101100	4.025	y	01111001	1.974
m	01101101	2.406	z	01111010	0.074

well above the upper bound of 2^{26}. The decreasement of the SEI is mainly caused by the computation of the convolution between the two distributions. On the other hand, for the bitwise linear approximation case, the distribution of the xor of two underlying distributions can be computed by the piling-up lemma, which is detailed in the next Section.

11. Practical Implementation of the Ciphertext-Only Attack

In the experiments of the ciphertext-only attack, we choose the condition mask $\lambda =$ `0x00f` and $\gamma =$ `0x1f`, $\eta = \mathbf{0}$. In this configuration, we have the conditional correlation $\Delta(h^{\Lambda}_{\mathcal{B}'_{t+1}}) \approx 2^{-3.67}$ and the unconditional correlation $\Delta(h^{\gamma}) = 2^{-6.71}$. Assume the plaintext are represented by ASCII codes and we can use the Table 8 to compute the SEI of the plaintext, i.e., $\Delta(\bar{\gamma} \cdot M_t) \approx 2^{-1.82}$. Therefore, according to Eq. (11), we can calculate the data complexity of the ciphertext-only attack as $\mathcal{N}_{\mathcal{B}'} \approx 2^{28.79}$, which is larger than the upper bound of 2^{26} in the real Bluetooth system for a fixed key. To compensate this issue, we use the same list decoding and multi-pass method as those in the known-IV scenario to assure a high success probability.

In our experiments, we set the $\mathcal{N}_{\mathcal{B}'} \approx 2^{26}$ (slightly less than the theoretical estimate $2^{28.79}$). The pre-computation of $\widehat{\mathcal{H}'}$ is $17 \cdot 2^{17}$, and we need time $2 \cdot 17 \cdot 2^{17} \approx 2^{21.1}$ to compute $\widehat{\mathcal{H}}$, $\widehat{\mathcal{H}''}$, and time $n_{\mathcal{B}'} = 2^{26}$ to compute $\mathcal{H}$, so the total time is $2^{26} + 2^{21.1}$. Thus, our ciphertext-only attack can be easily carried out in real time on one core of a single PC and the cost of our attack is nearly the same as that of the known-plaintext attack.

In order to acquire enough number of the plaintexts that conform to the relative frequencies of the natural English, we collected the plaintexts from many famous novels. Then, we encrypted each letter by the two-level E0 scheme as a Bluetooth frame, and thus we can get 2^{26} frames. As described in Sect. 5, we have known the format of the Bluetooth frame. Therefore, in the practical application we can use a Bluetooth sniffer to

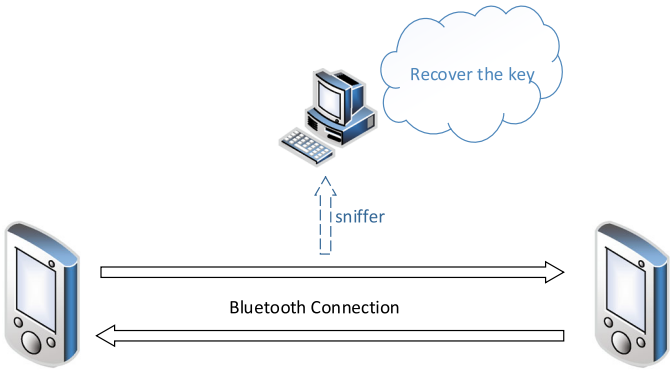

Fig. 11. The practical attack scenario of bluetooth encryption.

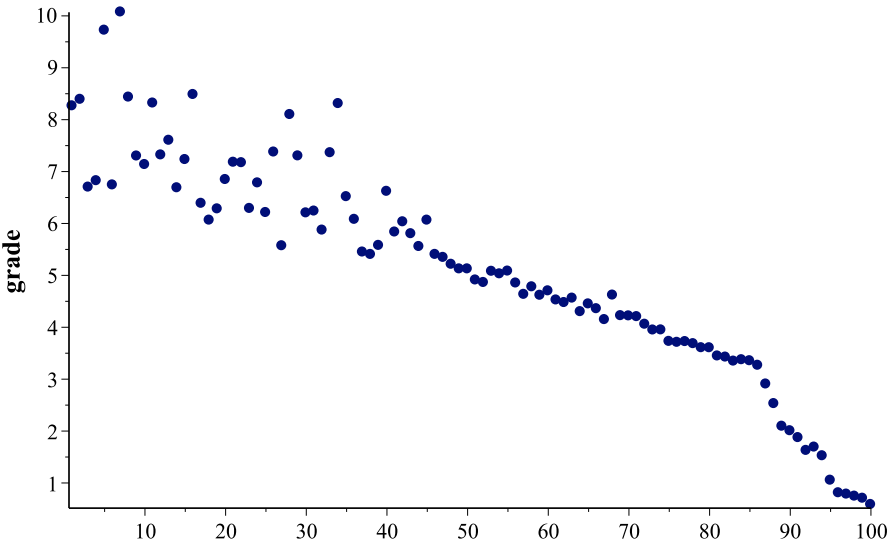

Fig. 12. The distribution of grades.

grab the Bluetooth packets from the Bluetooth devices, described in Fig. 11. We repeated 100 times of our ciphertext-only attack with different randomly generated keys and IVs. For each pair of key and IV, we generate 2^{26} Bluetooth frames which are stored in a binary file. Then, we use the algorithm which is similar to Algorithm 1 to recover the key. In the experiments, we take the first 256 candidates in the list as the possible keys for each run. We found that in about 69 runs, the correct key ranks among the first 256 candidates. Figure 12 shows the distribution of grades in the 100 times experiments. We can see that more than 70% grades are larger then 4. We can use some new condition masks, e.g., $\lambda = \texttt{0x01f}, \texttt{0x000f}$, to do the same experiments as above. This method will increase the success probability and decrease the size of key candidates. After recovering the partial key bits, the other bits of key can be acquired in the same way. The theoretical

Table 9. Complexities of our ciphertext-only attack.

Mask	γ	η	Pre-com	Time	Frames	Memory
$0x00f$	$0x1f$	0	2^{26}	2^{26}	2^{26}	2^{17}

analysis is the same as in the known-IV attack. One run of our attack is as follows. We first generate 2^{26} frames by the key `0x1c774e7b1626ed02f2b9b6b49afb82a1` and encrypt the plaintexts to generate 2^{26} ciphertexts, which need about 320Mb to be stored. The flow of the ciphertext-only attack is the same as the known-plaintext attack. In this mentioned instance, the grade of the right $K_1 = (\texttt{0x3}, \texttt{0x0}, \texttt{0xc}, \texttt{0x2})$, $K_2 = \texttt{0x1}$ is 9.020190, which ranks the first. The complexities of our attack are listed in Table 9.

Countermeasure. Following the design criterion in Sect. 4.6, we recommend to discard the first $2 \cdot 39 = 78$ keystream bits at the beginning of the second level to resist against our attack. In this case, the unconditional correlations can be reduced to below 2^{-218}, which will frustrate our attack completely.

12. Conclusions

In this paper, we have studied the security of a general two-level E0-like encryption model and the real-world Bluetooth encryption scheme. A fast recursive method with time complexity analysis is formulated to compute the unconditional correlations in the general core keystream generator. Besides, the conditional correlation properties of the two-level model are derived and analyzed by the condition masking technique. A key recovery framework is established to extract the secret key in the model, which has more generality compared to the previous one. Both bitwise and vectorial attacks have been mounted on the model with theoretical analysis. A novel design criterion is suggested to resist our attack. As the case study, we described more threatening and real time attacks on two-level E0. Our attacks have been fully implemented in C language on one core of a single PC and are repeated hundreds of times with randomly generated keys and IVs. On average, it takes only a few seconds to restore the original encryption key. This clearly demonstrates the superiority of our method. Finally, we converted the attack into a ciphertext-only attack with only small increments in the complexities. This is the first practical ciphertext-only attack against the Bluetooth encryption in the real-world so far. We suggest to discard the first 78 keystream bits at the beginning of the second level to strengthen the security of Bluetooth encryption.

Acknowledgements

The authors would like to thank the anonymous reviewers for very helpful comments. This work is supported by the program of the National Natural Science Foundation of China (Grant No. 61572482) and National Grand Fundamental Research 973 Programs of China (Grant No. 2013CB338002).

References

[1] F. Armknecht, M. Krause, Algebraic attacks on combiners with memory, in D. Boneh (ed.) *Advances in Cryptology—CRYPTO 2003*. Lecture Notes in Computer Science, vol. 2729 (Springer, Berlin, 2003), pp. 162–175

[2] T. Baignères, P. Junod, S. Vaudenay, How far can we go beyond linear cryptanalysis?, in P. Lee (ed.) *Advances in Cryptology—ASIACRYPT 2004*. Lecture Notes in Computer Science, vol. 3329 (Springer, Berlin, 2004), pp. 113–128

[3] Bluetooth SIG. Specification of the Bluetooth system. volume 4.2 (2016). https://www.bluetooth.com/specifications/adopted-specifications

[4] A. Canteaut, M. Trabbia, Improved fast correlation attacks using parity-check equations of weight 4 and 5, in B. Preneel (ed.) *Advances in Cryptology—EUROCRYPT 2000*. Lecture Notes in Computer Science, vol. 1807 (Springer, Berlin, 2000), pp. 573–588

[5] P. Chose, A. Joux, M. Mitton, Fast correlation attacks: an algorithmic point of view, in L. Knudsen (ed.) *Advances in Cryptology—EUROCRYPT 2002*. Lecture Notes in Computer Science, vol. 2332 (Springer, Berlin, 2002), pp. 209–221

[6] N. Courtois, Fast algebraic attacks on stream ciphers with linear feedback, in D. Boneh (ed.) *Advances in Cryptology—CRYPTO 2003*. Lecture Notes in Computer Science, vol. 2729 (Springer, Berlin, 2003), pp. 176–194

[7] S. Fluhrer, S. Lucks, Analysis of the E0 encryption system, in S. Vaudenay, A. Youssef (eds.) *Selected Areas in Cryptography—SAC 2001*. Lecture Notes in Computer Science, vol. 2259 (Springer, Berlin, 2001), pp. 38–48

[8] S.R. Fluhrer. Improved key recovery of level 1 of the Bluetooth encryption system (2002). http://eprint.iacr.org/2002/068

[9] J. Golić, Correlation properties of a general binary combiner with memory. *J. Cryptol.* **9**, 111–126 (1996)

[10] J. Golić, V. Bagini, G. Morgari, Linear cryptanalysis of Bluetooth stream cipher, in L. Knudsen (ed.) *Advances in Cryptology—EUROCRYPT 2002*. Lecture Notes in Computer Science, vol. 2332 (Springer, Berlin, 2002), pp. 238–255

[11] M. Hermelin, K. Nyberg, Correlation properties of the Bluetooth combiner, in J.S. Song (ed.) *Information Security and Cryptology—ICISC'99*. Lecture Notes in Computer Science, vol. 1787 (Springer, Berlin, 2000), pp. 17–29

[12] T. Johansson, F. Jönsson, Improved fast correlation attacks on stream ciphers via convolutional codes, in J. Stern (ed.) *Advances in Cryptology—EUROCRYPT'99*. Lecture Notes in Computer Science, vol. 1592 (Springer, Berlin, 1999), pp. 347–362

[13] T. Johansson, F. Jönsson, Fast correlation attacks through reconstruction of linear polynomials, in M. Bellare (ed.) *Advances in Cryptology—CRYPTO 2000*. Lecture Notes in Computer Science, vol. 1880 (Springer, Berlin, 2000), pp. 300–315

[14] M. Krause, BDD-based cryptanalysis of keystream generators, in L. Knudsen (ed.) *Advances in Cryptology—EUROCRYPT 2002*. Lecture Notes in Computer Science, vol. 2332 (Springer, Berlin, 2002), pp. 222–237

[15] S. Lee, S. Chee, S. Park, S. Park, Conditional correlation attack on nonlinear filter generators, in K. Kim, T. Matsumoto (eds.) *Advances in Cryptology—ASIACRYPT'96*. Lecture Notes in Computer Science, vol. 1163 (Springer, Berlin, 1996), pp. 360–367

[16] B. Löhlein, Attacks based on conditional correlations against the nonlinear filter generator. https://eprint.iacr.org/2003/020.pdf

[17] Y. Lu. Sampling with Walsh Transforms (2015). http://arxiv.org/abs/1502.06221

[18] Y. Lu, Y. Desmedt, Walsh transforms and cryptographic applications in bias computing. *Cryptogr. Commun.* **8**(3), 435–453 (2016)

[19] Y. Lu, W. Meier, S. Vaudenay, The conditional correlation attack: a practical attack on Bluetooth encryption, in V. Shoup (ed.) *Advances in Cryptology—CRYPTO 2005*. Lecture Notes in Computer Science, vol. 3621 (Springer, Berlin, 2005), pp. 97–117

[20] Y. Lu, S. Vaudenay, Cryptanalysis of Bluetooth keystream generator two-level E0, in P. Lee (ed.) *Advances in Cryptology—ASIACRYPT 2004*. Lecture Notes in Computer Science, vol. 3329 (Springer, Berlin, 2004), pp. 147–158

[21] Y. Lu, S. Vaudenay, Faster correlation attack on Bluetooth keystream generator E0, in M. Franklin (ed.) *Advances in Cryptology—CRYPTO 2004*. Lecture Notes in Computer Science, vol. 3152 (Springer, Berlin, 2004), pp. 35–49
[22] Y. Lu, S. Vaudenay, Cryptanalysis of an E0-like combiner with memory. *J. Cryptol.* **21**, 430–457 (2008)
[23] M. Matsui, Linear cryptanalysis method for DES cipher, in T. Helleseth (ed.) *Advances in Cryptology—EUROCRYPT'93*. Lecture Notes in Computer Science, vol. 765 (Springer, Berlin, 1994), pp. 386–397
[24] W. Meier, O. Staffelbach, Fast correlation attacks on certain stream ciphers. *J. Cryptol.* **1**, 159–176 (1989)
[25] W. Meier, O. Staffelbach, Correlation properties of combiners with memory in stream ciphers. *J. Cryptol.* **5**, 67–86 (1992)
[26] N. Petrakos, G.W. Dinolt, J.B. Michael, P. Stanica, Cube-type algebraic attacks on wireless encryption protocols. *Publ. IEEE Comput. Soc. Comput.* **42**(10), 103 –105 (2009)
[27] B. Preneel, Stream ciphers: past, present and future. https://securewww.esat.kuleuven.be/cosic/publications/talk-197.pdf
[28] R.A. Rueppel, Correlation immunity and the summation generator, in H.C. Williams (ed.) *Advances in Cryptology—CRYPTO'85*. Lecture Notes in Computer Science, vol. 218 (Springer, Berlin, 1986), pp. 260–272
[29] M. Saarinen. Re: bluetooth and E0 (2000). `sci.crypt.research`, 02/09/00
[30] Y. Shaked, A. Wool, Cryptanalysis of the Bluetooth E0 cipher using OBDD's, in S. Katsikas, J. López, M. Backes, S. Gritzalis, B. Preneel (eds.) *Information Security*. Lecture Notes in Computer Science, vol. 4176 (Springer, Berlin, 2006), pp. 187–202
[31] T. Siegenthaler, Decrypting a class of stream ciphers using ciphertext only. *IEEE Trans. Comput.* **C-34**, 81–85 (1985)
[32] R.K. Yarlagadda, J.E. Hershey, *Hadamard matrix analysis and synthesis with applications to communications and signal/Image processing* (Kluwer Academic Publishers, 1997)
[33] B. Zhang, D. Feng, Multi-pass fast correlation attack on stream ciphers, in E. Biham, A.M. Youssef (eds.) *Selected Areas in Cryptography—SAC 2006*. Lecture Notes in Computer Science, vol. 4356 (Springer, Berlin, 2007), pp. 234–248
[34] B. Zhang, C. Xu, D. Feng, Real time cryptanalysis of Bluetooth encryption with condition masking, in R. Canetti, J.A. Garay (eds.) *Advances in Cryptology—CRYPTO 2013*. Lecture Notes in Computer Science, vol. 8042 (Springer, Berlin, 2013), pp. 165–182