Real Time Cryptanalysis of Bluetooth Encryption with Condition Masking*
(Extended Abstract)

Bin Zhang[1], Chao Xu[2], and Dengguo Feng[2]

[1] State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences, Beijing, 100093, P.R. China
[2] Institute of Software, Chinese Academy of Sciences, Beijing 100190, P.R. China
{zhangbin,xuchao}@is.iscas.ac.cn

Abstract. The Bluetooth standard authorized by IEEE 802.15.1 adopts the two-level E0 stream cipher to protect short range privacy in wireless networks. The best published attack on it at Crypto 2005 requires 2^{38} on-line computations, 2^{38} off-line computations and 2^{33} memory (which amount to about 19-hour, 37-hour and 64GB storage in practice) to restore the original encryption key, given the first 24 bits of $2^{23.8}$ frames. In this paper, we describe more threatening and real time attacks against two-level E0 based on condition masking, a new cryptanalytic technique that characterizes the conditional correlation attacks on stream ciphers. The idea is to carefully choose the condition to get better tradeoffs on the time/memory/data complexity curve. It is shown that if the first 24 bits of $2^{22.7}$ frames is available, the secret key can be reliably found with 2^{27} on-line computations, $2^{21.1}$ off-line computations and 4MB memory. Our attacks have been fully implemented on one core of a single PC. It takes only a few seconds to restore the original encryption key. This is the best known-IV attack on the real Bluetooth encryption scheme so far.

Keywords: Stream ciphers, Correlation, Condition masking, Bluetooth two-level E0.

1 Introduction

Bluetooth and WiFi wireless networks are ubiquitous nowadays. The Bluetooth standard [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

* This work was supported by the National Grand Fundamental Research 973 Program of China(Grant No. 2013CB338002), the Strategic Priority Research Program of the Chinese Academy of Sciences (Grant No. XDA06010701), IIE's Research Project on Cryptography (Grant No. Y3Z0016102) and the programs of the National Natural Science Foundation of China (Grant No. 60833008, 60603018, 61173134, 91118006, 61272476)

R. Canetti and J.A. Garay (Eds.): CRYPTO 2013, Part I, LNCS 8042, pp. 165–182, 2013.

problem in stream ciphers, the security analysis of two-level E0 is still of great practical importance, as pointed out by Prof. Preneel in [25].

Correlation attack [28] 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 90's, the correlation properties of combiners with memory is analyzed [9,23] in theory. Based on these correlations, for LFSR-based stream ciphers, the initial state of the target LFSR can be recovered by (fast) correlation attacks [4,5,12,13,22]. 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 function. Later at Crypto 2005 [17], the conditional correlation is assigned with a dual meaning, i.e., the correlation of the output of a function conditioned on some unknown input, called condition vector, which is uniformly distributed and is applied to analyze the security of two-level E0. Usually, the condition vector is some key related material and if a good conditional correlation exists, it is expected that the adversary will observe the biased sample sequence for the correct key and unbiased sequences for the wrong candidates. Thus, a distinguisher can be mounted to restore the secret key given a pool of sample sequences derived from the guessed values of the condition vector and some public information.

In practice, the E0 cipher is frequently re-synchronized as a two-level scheme and the keystream generated for each frame is only 2745 bits. Thus, most of the published attacks [1,6,11,14,19,26,27] 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 [7,8,10,17,18,24] apply to the two-level E0. The cube attack in [24] works under the unrealistic assumption that the output of LFSRs at any clock cycle is available and it is a chosen-IV attack. The best known-IV attack in [17] 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-hour, 37-hour and 64GB storage, given the first 24 bits of 2^{26} frames). Note that this attack depends dominantly on the external data transfer rate between the hard disk and main memory and the pre-computation, which has to be done once for *each* key, is too time-consuming.

In this paper, we propose a new cryptanalytic technique, called condition masking, to characterize the conditional correlation attacks on stream ciphers. The attack in [17] considered the correlations conditioned on the whole condition vector, whereas we investigate the correlations only based on a subset of the condition vector. 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. Our main observation is that it is of high probability that only a subset of bits in the whole condition vector determine the magnitude of the bias, e.g., in the E0 combiner, only the latest four input bits to the FSM play the most important role. The theoretical framework in [17] is refined based on this

notion and it is shown that the time/memory complexities of the attack against two-level E0 can be significantly reduced by properly choosing the condition mask.

Precisely, we first present the complete[1] formula for fast computation of unconditional correlations in the E0 combiner, and thus efficiently solve the 11-year old open problem of Golić in [10]. Second, we precisely study the conditional correlations in two-level E0 with the condition masking. The target function inherent in E0 used to compute the conditional correlation in [17] 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, combined with the vectorial approach[2], the data complexity of our attack can be reduced or at least kept at the same magnitude level of that in [17] as well. A necessary and sufficient condition that determines when the adversary could gain in the correlation by moving from bit to small vector (or from low-dimension to high-dimension) in the conditional correlation attack is proved in theory. Based on it, the vectors used in our attack are constructed and indeed work well to keep the data complexity as small as possible without a penalty in the time or memory complexities. As a result of all the above techniques, it is shown that if the first 24 bits of $2^{22.7}$ frames is available, the secret key can be reliably found with 2^{27} on-line computations, $2^{21.1}$ off-line computations and 4MB memory. Other choices of tradeoff parameters are also possible. 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 [17] is only executed 30 times for a fixed 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.

This paper is organized as follows. A full description of the two-level E0 scheme is presented in Section 2. Various correlation properties about the E0 combiner, e.g., unconditional and conditional correlations based on condition masking are studied in Section 3. Inspired by these findings, both bitwise and vector-wise key recovery attacks based on condition masking are developed in Section 4. In Section 5, the practical implementation of our attack is described with the experimental results. Finally, some conclusions are provided and future work are pointed out in Section 6.

[1] Here 'complete' means that the formula can cover all the correlated input and output linear masks.

[2] Using multiple linear approximations at the same time.

2 Description of Bluetooth Two-Level E0

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 is 74 bits. The core is a modification of the summation generator with 4-bit memory. Precisely, the keystream generator consists of four regularly-clocked LFSRs whose lengths are 25, 31, 33 and 39 bits, respectively (128 bits in total). Their outputs are combined by a Finite State Machine (FSM) with 4 bits memory. At each time t, the following steps are executed.

The keystream generation of E0

Parameters:
1: $B_t = (b_t^1, b_t^2, b_t^3, b_t^4) \in GF(2)^4$ denote the output bits of four LFSRs
2: $X_t \in GF(2)^4$ denotes the 4 memory bits $(c_{t-1}, c_t) = (c_{t-1}^1, c_{t-1}^0, c_t^1, c_t^0)$
3: z_t is the keystream bit
Input: X_t, B_t
5: $z_t = b_t^1 \oplus b_t^2 \oplus b_t^3 \oplus b_t^4 \oplus c_t^0$
6: $s_{t+1} = (s_{t+1}^1, s_{t+1}^0) = \lfloor \frac{b_t^1 + b_t^2 + b_t^3 + b_t^4 + 2c_t^1 + c_t^0}{2} \rfloor$
7: $c_{t+1}^0 = s_{t+1}^0 \oplus c_t^0 \oplus c_{t-1}^1 \oplus c_{t-1}^0$, $c_{t+1}^1 = s_{t+1}^1 \oplus c_t^1 \oplus c_{t-1}^0$
8: $(c_{t-1}, c_t) \leftarrow (c_t, c_{t+1})$
9: update the LFSRs

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., $R_t = b_t^1 \oplus b_t^2 \oplus b_t^3 \oplus b_t^4$ and $z_t = R_t \oplus c_t^0$.

Next, we introduce the two-level E0 scheme, as shown in Fig. 1. 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 $\alpha_t, \beta_{t'}$ respectively.

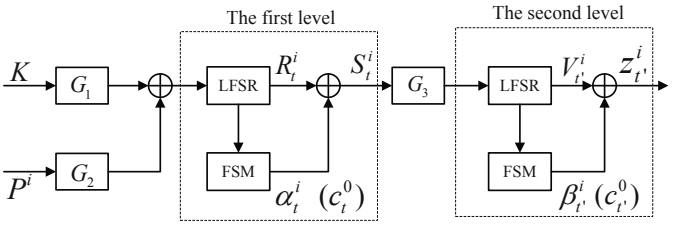

Fig. 1. Two-level E0 encryption scheme

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 $R^i_{[-199,\cdots,-72]} = (R^i_{-199}, \cdots, R^i_{-72}) = G_1(K) \oplus G_2(P^i)$, where G_1 and G_2 are public affine transformations over $GF(2)^{128}$.[3]

[3] Hereafter we always use the superscript i to indicate the context of the i-th frame.

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 $S^i_{[-127,\cdots,0]} = R^i_{[-127,\cdots,0]} \oplus \alpha^i_{[-127,\cdots,0]}$. Let M be the state transmission matrix of the equivalent LFSR over $GF(2)^{128}$, i.e., $R^i_{[-127,\cdots,0]} = M^{72}(R^i_{[-199,\cdots,-72]})$. 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 $R^i_{[-127,\cdots,0]} = (M^{72} \circ G_1)(K) \oplus (M^{72} \circ G_2)(P^i)$.
3. $S^i_{[-127,\cdots,0]}$ is used to initialized the four LFSRs by a byte-wise affine transformation $G_3 : GF(2)^{128} \to GF(2)^{128}$, detailed in Section 4.1 and Appendix B, this process can be expressed by $V^i_{[1,\cdots,128]} = G_3(S^i_{[-127,\cdots,0]})$.
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 $z^i_{t'} = V^i_{t'} \oplus \beta^i_{t'}$ of the i-th frame for $t' = 1, \cdots, 2745$.

3 Correlations in the Bluetooth Combiner

In this section, we will carefully study both the unconditional and conditional correlation properties of the E0 combiner.

3.1 Unconditional Linear Correlations

We first give the definition of correlation used in this paper.

Definition 1. *The correlation (or bias) of a random Boolean variable X is* $\epsilon(X) = Pr(X = 1) - Pr(X = 0)$.[4]

Let $\Omega(a, (\omega, u))$ denote the correlation $\epsilon(a \cdot s_{t+1} \oplus \omega \cdot c_t \oplus u \cdot B_t)$, where $a \in GF(2)^2$, $u \in GF(2)^4$, $\omega \in GF(2)^2$ and B_t denote the output bits of four LFSRs at time t. From Section 2, note that s_{t+1} is symmetric with respect to each b^i_t and depends only on $wt(B_t)$.[5] Our complete formula for the computation of unconditional correlations is as follows.

Theorem 2. *Let $h : (x^1, x^0) \to (x^0, x^1 \oplus x^0)$ be a permutation over $GF(2)^2$ and $\delta((a_1, u_1), \cdots, (a_{d-1}, u_{d-1}), a_d) = \epsilon(a_1 \cdot c_1 \oplus u_1 \cdot B_1 \oplus \cdots \oplus a_{d-1} \cdot c_{d-1} \oplus u_{d-1} \cdot B_{d-1} \oplus a_d \cdot c_d)$, where $a_1, \cdots, a_d \in GF(2)^2$ and $u_1, \cdots, u_{d-1} \in GF(2)^4$. If the initial state of the FSM is uniformly distributed, then we have*

$$\delta((a_1, u_1), \cdots, (a_{d-1}, u_{d-1}), a_d) = -\sum_{\omega \in GF(2)^2} \Omega(a_d, (\omega, u_{d-1})) \cdot \delta((a_1, u_1),$$
$$\cdots, (a_{d-3}, u_{d-3}), (a_{d-2} \oplus h(a_d), u_{d-2}), a_{d-1} \oplus a_d \oplus \omega).$$

Theorem 2 is a generalization of the formula in [19,20]. It can compute all the unconditional correlations of the E0 combiner without any miss, e.g., it covers all the results reported in [10].

[4] Note that in some articles, $\epsilon(X) = Pr(X = 0) - Pr(X = 1)$. The only difference is the sign of the correlation.

[5] $wt(\cdot)$ denotes the Hamming weight of a vector.

3.2 Conditional Correlations Based on Condition Masking

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 $\gamma = (\gamma_0, \gamma_1, \cdots, \gamma_{l-1}) \in GF(2)^l$ be a linear mask with $\gamma_0 = \gamma_{l-1} = 1$. $\bar{\gamma} = (\gamma_{l-1}, \gamma_{l-2}, \cdots, \gamma_0)$ is the linear mask in reverse order. Define the inputs $\mathcal{B}_{t+1} = B_{t+1}B_{t+2}\cdots B_{t+l-2} \in \mathrm{GF}(2^{4(l-2)}), X_{t+1} = (c_t, c_{t+1}) \in \mathrm{GF}(2)^4$ and the FSM outputs $C_t = (c_t^0, \cdots, c_{t+l-1}^0)$. Then the function $h^{\gamma}_{\mathcal{B}_{t+1}} : X_{t+1} \to \gamma \cdot C_t$ conditioned on $\mathcal{B}_{t+1}$ is well defined. It is shown in [17] that given $\mathcal{B}_{t+1}$, $\gamma \cdot C_t$ is heavily biased for properly chosen linear mask γ.

Consider the function $h^{\gamma}_{\mathcal{B}_{t+1}} : X_{t+1} \to \gamma \cdot C_t$. With the knowledge of $\mathcal{B}_{t+1}$ and X_{t+1}, we can recursively compute C_t. The bias $\epsilon(h^{\gamma}_{\mathcal{B}_{t+1}})$ can be easily computed by an exhaustive search over all the possible values of X_{t+1}. For different values of $\mathcal{B}_{t+1}$, the bias $\epsilon(h^{\gamma}_{\mathcal{B}_{t+1}})$ may be different, while the mean value $E[\epsilon(h^{\gamma}_{\mathcal{B}_{t+1}})]$ is a good estimate in the attacks. The following definitions are essential in our attacks.

Definition 3. *Let ξ be an arbitrary set, given the function $f : \xi \to GF(2)^r$, 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 $a \in GF(2)^r$. As in [2], the Squared Euclidean Imbalance (SEI) of a distribution D_f is defined as $\Delta(D_f) = 2^r \sum_{a\in GF(2)^r}(D_f(a) - \frac{1}{2^r})^2$. SEI measures the distance between the target distribution 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, $E[\Delta(h_{\mathcal{B}})]$ is used to measure the conditional correlations. Now we are ready for the definition of condition masking.

Definition 4. *Given a function $h : GF(2)^u \times GF(2)^v \to GF(2)^r$ with inputs $\mathcal{B} \in GF(2)^u$, $X \in GF(2)^v$, where $\mathcal{B}$ is the key related part and the possible condition vector. Let $\mathcal{B} = (b_0, \cdots, b_{u-1}) \in GF(2)^u$ and $\lambda = (\lambda_0, \lambda_1, \ldots, \lambda_{u-1}) \in GF(2)^u$ with $supp(\lambda) = \{0 \le i \le u-1 | \lambda_i = 1\} = \{l_1, \cdots, l_m\}$ $(l_j < l_{j+1})$. Then the shrunken vector of $\mathcal{B}$ defined by λ is $\mathcal{B}' = (b_{l_1}, \cdots, b_{l_m}) \in GF(2)^m$. Here λ is called the condition mask of $\mathcal{B}$. Further, other bits in $\mathcal{B}$ form another vector and is 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 maybe 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. In the cryptanalysis of E0, $\mathcal{B}_{t+1}$ is the key related input. Given a condition mask $\lambda = (\lambda_{t+1}, \cdots, \lambda_{t+l-2}) \in \mathrm{GF}(2)^{4(l-2)}$, where $\lambda_j \in \mathrm{GF}(2)^4$ corresponds to B_j for $j = t+1, \cdots, t+l-2$, denote the condition vector defined by λ by $\mathcal{B}'_{t+1}$ and its complement by $\mathcal{B}^*_{t+1}$ which includes the other bits. The function $h^{\gamma}_{\mathcal{B}_{t+1}}$ can now be generalized as

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

where $\Lambda = (\gamma, \omega)$ and $|\omega| = |\mathcal{B}^*_{t+1}|$.[6] As we can see, this function induces a large class of correlations based on both the linear mask and the condition mask.

Although the computation process of C_t is frustrated by the condition mask $\lambda \neq \mathbf{1}_u$, the bias can still be computed. For example, given $l = 4$ and $\lambda = 0x0f$,[7] 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 $\epsilon(h^{\Lambda}_{B_{t+2}})$. Since $\mathcal{B}_{t+1}$ is the outputs of the LFSRs, it is the key related material. In [17], the attacker guesses the full vector $\mathcal{B}_{t+1}$, while now he/she only needs to guess $\mathcal{B}'_{t+1}$, a part of $\mathcal{B}_{t+1}$, to mount the attack. 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 as $\mathcal{B}^i_t = L_t(K) \oplus L'_t(P^i)$, where L_t and L'_t are the public linear functions. The knowledge of $\mathcal{B}^i_t$ will directly lead to the linear equations on the original key. This motivates us to study the bias $\epsilon(h^{\Lambda}_{\mathcal{B}'_{t+1}})$ defined by a certain condition mask λ. 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, one is listed in the following Table 1. Table 1 is computed with $\lambda = 0x00f, \Lambda = (\gamma, \omega) = (0x1f, \mathbf{0}_{|\omega|})$. We get $E[\Delta(h_{\mathcal{B}'_{t+1}})] \approx 2^{-3.7}$, where $\mathcal{B}'_{t+1} = B_{t+3}$. The following property, shows that the more knowledge of the LFSR bits $\mathcal{B}$, the larger conditional correlation we will obtain.

Proposition 5. *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(λ_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 $\mathcal{B}_1 \setminus \mathcal{B}_2$.*

From this proposition, give a function $h : GF(2)^u \times GF(2)^v \to GF(2)^r$ with $\mathcal{B} \in GF(2)^u$, $X \in GF(2)^v$ and a condition mask λ, we have $E[\Delta(h_{\mathcal{B}})] \geq E[\Delta(h_{\mathcal{B}'})] \geq \Delta(h)$. Moreover, for a fixed condition mask λ, its maximum bias among all the

Table 1. The bias with $\Lambda = (\gamma, \omega) = (0x1f, \mathbf{0}_{|\omega|})$ and $\lambda = 0x00f$

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

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

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

linear masks Λ is an essential measure of it. The larger the maximum bias, the better the condition mask is. The following proposition indicates how to choose the condition mask to make the bias large. We have verified this property by searching over all the biases of $h^{\Lambda}_{\mathcal{B}'}$ for each combination of λ, γ and ω.

Proposition 6. *For $4 \le l \le 6$, let $\mathcal{B}_{t+1} = B_{t+1} \cdots B_{t+l-2} \in GF(2)^{4(l-2)}$, and $\lambda = (\lambda_{t+1}, \cdots, \lambda_{t+l-2})$, $\lambda' = (\lambda'_{t+1}, \cdots, \lambda'_{t+l-2})$ are two condition masks with $wt(\lambda) = wt(\lambda') \ge 4$, 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 $max_{\Lambda}(E[\Delta(h^{\Lambda}_{\mathcal{B}'_{t+1}})]) > max_{\Lambda'}(E[\Delta(h^{\Lambda'}_{\mathcal{B}'_{t+1}})])$,[8] expect that 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 Proposition 6, $wt(B_{t+l-2})$ in $\mathcal{B}_t$ plays the most important role in the correlation values based on condition masking, which determines the magnitude of the corresponding bias. This fact tells us that when selecting the condition masks, we should set the highest four bits of λ to $0xf$.

4 Our Attacks with Condition Masking

In this section, our attack with the condition masking method is presented in a step-by-step manner.

4.1 Preliminaries

A statistical distinguisher can be constructed based on the biased distribution of $\gamma \cdot C_t$ in [17]. 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. By properly choosing the involved parameters, 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.

As mentioned before, the essential problem lies in the core of the attack 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 $f : GF(2)^m \times GF(2)^{u-m} \times GF(2)^v \to GF(2)^r$ and a condition mask λ, let $f_{\mathcal{B}'}(\mathcal{B}^*, X) = f(\mathcal{B}', \mathcal{B}^*, X)$ with $\mathcal{B} = \mathcal{B}' \cup \mathcal{B}^* \in GF(2)^u$, $X \in GF(2)^v$. Here the condition vector defined by λ is $\mathcal{B}' \in GF(2)^m$ and $\mathcal{B}^* = \mathcal{B} \setminus \mathcal{B}'$. If $\mathcal{B}'$ is determined by k-bit key information, then denote by $\mathcal{B}'^{\mathcal{K}}$ the value derived when the guessing value of the key material is $\mathcal{K}$, now the problem is as follows.

Definition 7. *There are 2^k sequences of n samples with the following characteristics: one biased sequence has n samples $(f_{\mathcal{B}'^{\mathcal{K}}_i}, \mathcal{B}^{\mathcal{K}}_i)$ $(i = 1, \ldots, n)$ with the*

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

correct key $\mathcal{K}$; the other $2^k - 1$ sequences consists of n independently and uniformly distributed random variables $(Z_i^K, \mathcal{B}_i'^K)$ $(i = 1, \ldots, n)$ with the wrong keys $K \neq \mathcal{K}$. The problem is to efficiently distinguish the biased sequence from the other sequences with the minimum number n of samples.

Following [2], the minimum number n of samples for an optimal distinguisher using the unconditional correlation to effectively distinguish a sequence of n output samples of f from $(2^k - 1)$ truly random sequences of equal length is $n = \frac{4k \log 2}{\Delta(f)}$, while with the smart distinguisher in [17] based on the condition vector $\mathcal{B}$, the number of samples needed is $n_{\mathcal{B}} = \frac{4k \log 2}{E[\Delta(f_{\mathcal{B}})]}$. Since $E[\Delta(f_{\mathcal{B}})] \geq \Delta(f)$, we have $n_{\mathcal{B}} \leq n$. In our condition masking terminology, detailed in Appendix A Theorem 10, the data complexity becomes $n_{\mathcal{B}'} = \frac{4k \log 2}{E[\Delta(f_{\mathcal{B}'})]}$, and the online time complexity are $O(n_{\mathcal{B}'} + k2^{k+1})$ with pre-computation $O(k2^k)$. Besides, $|\mathcal{B}'| = k$.

We should not ignore the impact of the cardinality of the condition vector $|\mathcal{B}'| = k$ on the time/memory complexities. It is easy to see that for $\lambda \neq \mathbf{1}_u$, the cardinality k can be reduced and accordingly the time/memory complexities can be exponentially reduced. 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 to represent this phenomenon. 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. For example, Proposition 6 shows that in the E0 FSM, only the latest four bits of $\mathcal{B}_{t+1}$ play the most important role. This is the key observation of our attack.

Next, we build the linear approximations with condition masking. The linear approximation is based on the re-initialization flaw of two-level E0 [18] detailed in the Appendix B. Precisely, we have $\bar{\gamma} \cdot (Z_{t'}^i \oplus \mathcal{L}_{t'}(K) \oplus \mathcal{L}'_{t'}(P^i)) = \bigoplus_{j=1}^{4} (\gamma \cdot C_{t_j}^i) \oplus \bar{\gamma} \cdot C_{t'}^i$, for $i = 1, \cdots, n$ and $\mathcal{L}_{t'}, \mathcal{L}'_{t'}$ are public linear functions. Here we have $t' \in \bigcup_{d=0}^{2} \{8d + 1, \cdots, 8d + 9 - l\}$. By Eq.(1), we can rewrite this equation as follow:

$$\bar{\gamma} \cdot (Z_{t'}^i \oplus \mathcal{L}_{t'}(K) \oplus \mathcal{L}'_{t'}(P^i)) \oplus \bigoplus_{j=1}^{4} (\omega \cdot \mathcal{B}_{t_j+1}^{*i}) = \bigoplus_{j=1}^{4} (\gamma \cdot C_{t_j}^i \oplus \omega \cdot \mathcal{B}_{t_j+1}^{*i}) \oplus \bar{\gamma} \cdot C_{t'}^i. \quad (2)$$

For brevity, given masks λ and Λ, we use the simplified notations $h^{\Lambda}_{\mathcal{B}'^i_{t+1}}, h^{\bar{\gamma}}$ to denote $h^{\Lambda}_{\mathcal{B}'^i_{t+1}}(\mathcal{B}_{t+1}^{*i}, X_{t+1}^i)$, $h^{\bar{\gamma}}(\mathcal{B}_{t'+1}^i, X_{t'+1}^i)$ hereafter. Since $\mathcal{B}_{t+1}^{*i} = \mathcal{B}_{t+1}^i \setminus \mathcal{B}'^i_{t+1}$ is the linear combination of K and P^i. Now Eq.(2) becomes

$$\bar{\gamma} \cdot (Z_{t'}^i \oplus \mathcal{L}_{t'}(K) \oplus \mathcal{L}'_{t'}(P^i)) \oplus \omega \cdot (L_1(K) \oplus L_2(P^i)) = \bigoplus_{j=1}^{4} h^{\Lambda}_{\mathcal{B}'^i_{t_j+1}} \oplus h^{\bar{\gamma}}, \quad (3)$$

where L_1, L_2 are public linear functions. Eq.(3) is the hybrid bitwise linear approximation based on condition masking for two-level E0, where $h^{\Lambda}_{\mathcal{B}'^i_{t_j+1}}$ are

derived from the first level and $h^{\bar{\gamma}}$ contains the unconditional correlation for the second level.

4.2 Key Recovery Attack with Bitwise Linear Approximation

From Section 3, 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.(3), we choose these two γs in the second level approximation, then $|\gamma| = l = 5$ or 6. Due to the high time/memory complexities, the attack in [17] only considered $l < 6$. While in our attack, the time/memory complexities are not dependent on $|\gamma|$, they are determined by $wt(\lambda)$, thus $l = 6$ can also be used in the condition masking setting.

Given the condition mask λ and the linear masks $\Lambda = (\gamma, \omega)$, we define the following sign function to estimate the effective value of $h^{\Lambda}_{\mathcal{B}'^i_{t+1}}$ (Eq.(1)):

$$g^{\Lambda}(\mathcal{B}'^i_{t+1}) = \begin{cases} 1, & \text{if } \epsilon(h^{\Lambda}_{\mathcal{B}'^i_{t+1}}) > 0 \\ 0, & \text{if } \epsilon(h^{\Lambda}_{\mathcal{B}'^i_{t+1}}) < 0 \end{cases} \tag{4}$$

for all $\mathcal{B}'^i_{t+1} \in GF(2)^{wt(\lambda)}$ such that $\epsilon(h^{\Lambda}_{\mathcal{B}'^i_{t+1}}) \neq 0$. For brevity, let

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

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 $X^i_{t'+1}, \mathcal{B}^i_{t'+1}$ are the inputs to $h^{\bar{\gamma}}$. By Eq.(3), the knowledge of the key K is contained in $\mathcal{B}^i_{\lambda}, \mathcal{L}_{t'}(K)$ and $L_1(K)$. Let the $4wt(\lambda)$ bits $K_1 = (L_{t_1}(K), L_{t_2}(K), L_{t_3}(K), L_{t_4}(K))$ contained in $\mathcal{B}^i_{\lambda}$ and $K_2 = \bar{\gamma} \cdot \mathcal{L}_{t'}(K) \oplus \omega \cdot L_1(K)$ be the subkeys. Denote by $\widetilde{\cdot}$ the guessed value of the argument. The attack is detailed as follow.

First, choose an appropriate condition mask λ and guess the subkeys $\widetilde{K_1}$ and $\widetilde{K_2}$. As P^i is known for each frame $i = 1, \cdots, n$, 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.

$$\mathcal{F}^{\Lambda}_{\mathcal{B}^i_{\lambda}}(\mathcal{X}^i) = \begin{cases} \bigoplus_{j=1}^{4}(h^{\Lambda}_{\mathcal{B}'^i_{t+1}} \oplus g^{\Lambda}(\widetilde{\mathcal{B}'^i_{t_j+1}})) \oplus h^{\bar{\gamma}}, & \text{if } \prod_{j=1}^{4} \epsilon(h^{\Lambda}_{\mathcal{B}'^i_{t_j+1}}) \neq 0 \\ \text{a truly random bit,} & \text{otherwise} \end{cases}$$

With Eq.(4) the value of $\mathcal{F}^{\Lambda}_{B^i_{\lambda}}(\mathcal{X}^i)$ can be computed as

$$\mathcal{F}^{\Lambda}_{\mathcal{B}^i_{\lambda}}(\mathcal{X}^i) = \bar{\gamma} \cdot (Z^i_{t'} \oplus \mathcal{L}'_{t'}(P^i)) \oplus \omega \cdot L_2(P^i) \oplus \widetilde{K_2} \oplus \bigoplus_{j=1}^{4} g^{\Lambda}(\widetilde{\mathcal{B}'^i_{t_j+1}}).$$

If 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 n times. With appropriate choice of Λ and λ, if K_1, K_2 are correctly guessed, then $E[\Delta(\mathcal{F}^{\Lambda}_{B^i_{\lambda}}(\mathcal{X}^i))] > 0$ and we expect $\mathcal{F}^{\Lambda}_{\mathcal{B}^i_{\lambda}}(\mathcal{X}^i)$ equals one most of the time. Otherwise, $\mathcal{F}^{\Lambda}_{\mathcal{B}^i_{\lambda}}(\mathcal{X}^i)$ is estimated by the uniform

distribution, proved in [17]. Third, we get n outputs of the source for every possible key. Submitting these samples to the distinguisher in Algorithm 1 in Appendix A, with the $k = 4wt(\lambda) + 1, u = 16(l-2), m = wt(\lambda), v = 20 + 20(l-2) - 4wt(\lambda)$ and $r = 1$, we are expected to successfully restore the correct keys.

4.3 Key Recovery Attack with the Vectorial Approach

Now we enhance the above attack by using multiple linear approximations simultaneously. Since the 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 and let $\Gamma = (\Lambda_1, \cdots, \Lambda_s)$ and $\Gamma' = (\bar{\gamma_1}, \cdots, \bar{\gamma_s})$ denote the linear mask of these s approximations, where $\Lambda_i = (\gamma_i, \omega_i)$, and $|\gamma_1| = \cdots = |\gamma_s| = l$ with $s < l$. Especially, Λ_1 is just the linear mask used in the above bitwise attack. For brevity, let $g^\Gamma = (g^{\Lambda_1}(\mathcal{B}'^i_{t+1}), \cdots, g^{\Lambda_s}(\mathcal{B}'^i_{t+1}))$, $h^\Gamma_{\mathcal{B}'^i_{t+1}} = (h^{\Lambda_1}_{\mathcal{B}'^i_{t+1}}, \cdots, h^{\Lambda_s}_{\mathcal{B}'^i_{t+1}})$, $\mathcal{F}^\Gamma_{\mathcal{B}^i_\lambda}(\mathcal{X}^i) = (\mathcal{F}^{\Lambda_1}_{\mathcal{B}^i_\lambda}, \cdots, \mathcal{F}^{\Lambda_s}_{\mathcal{B}^i_\lambda})$ and $h^{\Gamma'} = (h^{\bar{\gamma_1}}, \cdots, h^{\bar{\gamma_s}})$. Here the first $g^{\Lambda_1}(\mathcal{B}'^i_{t+1})$ in g^Γ is determined by Eq.(4). The other bits are determined as follow: e.g., for the j-th bit, we just let it be an uniformly distributed bit if $\epsilon(h^{\Lambda_1}_{\mathcal{B}'^i_{t_j+1}}) = 0$, otherwise take 0 or 1 according to the definition in Eq.(4). 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.,

$$\mathcal{F}^\Gamma_{\mathcal{B}^i_\lambda}(\mathcal{X}^i) = \begin{cases} \bigoplus_{j=1}^4 (h^\Gamma_{\mathcal{B}'^i_{t_j+1}} \oplus g^\Gamma(\widetilde{\mathcal{B}'^i_{t_j+1}})) \oplus h^{\Gamma'}, & \text{if } \prod_{j=1}^4 \epsilon(h^{\Lambda_1}_{\mathcal{B}'^i_{t_j+1}}) \neq 0 \\ \text{a uniformly distributed } s\text{-bit vector}, & \text{otherwise.} \end{cases}$$

In this way, we have constructed an approximation of two-level E0 in the vectorial approach. For the correct guess $\widetilde{K} = K$, we have $\mathcal{F}^\Gamma_{\mathcal{B}^i_\lambda}(\mathcal{X}^i) = \bigoplus_{j=1}^4 (h^\Gamma_{\mathcal{B}'^i_{t_j+1}} \oplus g^\Gamma(\mathcal{B}'^i_{t_j+1})) \oplus h^{\Gamma'}$ and $E[\Delta(\mathcal{F}^\Gamma_{\mathcal{B}^i_\lambda}(\mathcal{X}^i))] > 0$. 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 $E[\Delta(\mathcal{F}^\Gamma_{\mathcal{B}^i_\lambda}(\mathcal{X}^i))] = 0$. With the appropriate choice of $\Gamma = (\Lambda_1, \cdots, \Lambda_s)$, 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^k sequences of $n_{\mathcal{B}'}$ pairs $(\mathcal{F}^\Gamma_{\mathcal{B}^i_\lambda}(\mathcal{X}^i), \widetilde{\mathcal{B}^i_\lambda})$ to Algorithm 1 in Appendix A, we can eventually recover the k-bit K.

Now we study how to choose the linear mask vector Γ. We first select a linear mask $\Lambda_1 = (\gamma_1, \omega_1)$ in the bitwise attack. Under this Λ_1, we search for other masks Λ_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 8. *Let $\Gamma_s = (\Lambda_1, \cdots, \Lambda_s)$ be the linear mask in the s-dimensional attack with condition vector $\mathcal{B}$ and condition mask λ. Denote the joint probability by $P_{a_1 \cdots a_s} = P(h^{\Lambda_1}_{\mathcal{B}'} = a_1, \cdots, h^{\Lambda_s}_{\mathcal{B}'} = a_s)$, where $a_i \in GF(2)$ for $1 \le i \le s$. Let $P_{00\cdots00} = \frac{1}{2^s} + \xi_{00\cdots00}$, $P_{00\cdots01} = \frac{1}{2^s} + \xi_{00\cdots01}$, $\cdots$, $P_{11\cdots11} = \frac{1}{2^s} + \xi_{11\cdots11}$, where $-\frac{1}{2^s} \le \xi_j \le \frac{1}{2^s}$ for all $j \in GF(2)^s$ and $\sum_{j \in GF(2)^s} \xi_j = 0$, then $\Delta(h^{\Gamma_s}_{\mathcal{B}'}) \ge \Delta(h^{\Gamma_{s-1}}_{\mathcal{B}'})$, where the equality holds if and only if $\xi_{00\cdots00} = \xi_{00\cdots01}, \xi_{00\cdots10} = \xi_{00\cdots11}, \cdots, \xi_{11\cdots10} = \xi_{11\cdots11}$.*

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 $j = 1, \cdots, s$ should be linearly independent with $s \le l - 2$. Second, when the key is wrong, $\mathcal{F}^{\Lambda_j}_{\mathcal{B}^i_\lambda}$ is an uniformly distributed bit for $1 \le j \le s$ in the bitwise attack. If they are independent to each other, $\mathcal{F}^{\Gamma}_{\mathcal{B}^i_\lambda}$ 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 $\mathcal{F}^{\Lambda_j}_{\mathcal{B}^i_\lambda}$ for $j = 1, \cdots, s$. Third, for a fixed Λ_1, when we choose some new $\Lambda = (\gamma, \omega)$ to constitute the vector, we should choose such γ that $\bar{\gamma}$ makes the unconditional correlation $\epsilon(h^{\bar{\gamma}}) = 0$ in the second level approximation, as such γ does not increase the time complexity after the extension to high-dimensional attack, which is shown in the following theorem.

Theorem 9. *Let $\Lambda_1 = (\lambda_1, \omega_1)$ be a linear mask adopted in the bitwise attack, if the jth-dimensional linear mask $\gamma_j (j \ge 2)$ makes the unconditional correlation $\epsilon(h^{\bar{\gamma}_j}) = 0$ in the approximation of the second level E0, then γ_j does not increase the time complexity when extending the $j-1$-dimensional vector to the j-dimensional vector.*

4.4 Theoretical Analysis

Now we present the theoretical justifications of our attack. We first introduce the definition of Walsh Transform and the convolution transform.

Given $f : GF(2)^k \to \mathbf{R}$, the Walsh transform $\hat{f}$ is $\hat{f}(\omega) = \sum_{x \in GF(2)^k} f(x)(-1)^{\omega \cdot x}$, and its inverse transform is $f(x) = 2^{-k} \sum_{\omega \in GF(2)^k} \hat{f}(\omega)(-1)^{\omega \cdot x}$. The convolution function of f and g is $(f \otimes g)(a) = \sum_{b \in GF(2)^k} f(b) \cdot g(a \oplus b)$ for $a \in GF(2)^k$. Further, the convolution and Walsh Transform are transformable, i.e., $\widehat{f \otimes g}(a) = \hat{f}(a) \cdot \hat{g}(a)$, for all $a \in GF(2)^k$.

To compute the convolution function $(f \otimes g)(a)$, we just perform the FWT of f and g, multiply them together and then use the inverse Walsh transform. The time and memory complexities of FWT are $O(k2^k)$ and $O(2^k)$, respectively.

By the definition of g^{Λ}, for a certain $\mathcal{B}^i_\lambda$, $g^{\Lambda}(\widetilde{\mathcal{B}'^i_{t_j+1}})$ is a fixed value not depending on $\mathcal{X}^i$. Consequently, g^{Γ} has no influence on $\Delta(\mathcal{F}^{\Gamma}_{\mathcal{B}^i_\lambda})$. We apply the Piling-up Lemma [21] and have the data complexity[9]

$$n_{\mathcal{B}'} = \frac{4k\log 2}{E[\Delta(\mathcal{F}^{\Gamma}_{\mathcal{B}^i_\lambda})]} = \frac{4k\log 2}{\Delta(h^{\Gamma'})\prod_{j=1}^{4} E[\Delta(h^{\Gamma}_{\mathcal{B}'^i_{t_j+1}})]} = \frac{4k\log 2}{\Delta(h^{\Gamma'})E^4[\Delta(h^{\Gamma}_{\mathcal{B}'^i_{t+1}})]}. \quad (5)$$

Now let us discuss the time complexity of our attack. From the expression of $\mathcal{F}^{\Gamma}_{\mathcal{B}^i_\lambda}$, it can be easily verified that this expression fulfills Theorem 10 in Appendix A, so our attack can also use the FWT to get the optimal time complexity. For all the subkeys $K = (K_1, K_2) \in GF(2)^{k-1} \times GF(2)$, where K_1 and K_2 are defined in Section 4.2, we define $\mathcal{H}, \mathcal{H}'$ as follows:

$$\mathcal{H}(K) = \sum_{i=1}^{n_{\mathcal{B}'}} \mathbf{1}_{L'_{t_1}(P^i),\cdots,L'_{t_4}(P^i)=K_1 \text{ and } (\theta_1,\cdots,\theta_s)=(K_2,1,\cdots,1)},$$

$$\mathcal{H}'(K) = \begin{cases} 0, & \text{if } \prod_{j=1}^{4}\epsilon(h^{\Lambda_1}_{K_{1,j}}) = 0 \\ \log 2^k D_{\mathcal{F}^{\Gamma}_{K_\lambda}}((K_2,1,\cdots,1)\oplus(\eta_1,\cdots,\eta_s)), & \text{otherwise} \end{cases}$$

where $\theta_j = \bar{\gamma}_j \cdot (Z^i_{t'} \oplus \mathcal{L}'_{t'}(P^i)) \oplus \omega_j \cdot L_2(P^i)$ and $\eta_j = \bigoplus_{i=1}^{4} g^{\Lambda_j}(K_{1,i})$ for $j = 1,\cdots,s$. In Algorithm 1 in Appendix A, the grade $G(K)$ is a simple convolution between $\mathcal{H}$ and $\mathcal{H}'$ (also in [17]), thus we have $G(K) = \frac{1}{2^k}\widehat{\mathcal{H}''(K)}$ where $\mathcal{H}''(K) = \widehat{\mathcal{H}}(K)\cdot\widehat{\mathcal{H}'}(K)$. Note that $\widehat{\mathcal{H}'}$ can be pre-computed in time $O(k\cdot 2^k)$ and $O(2^k)$ memory. The preparation of $\mathcal{H}$ needs $O(n_{\mathcal{B}'})$ online computation. $\widehat{\mathcal{H}}$ and $\widehat{\mathcal{H}''}$ need twice of FWT with time complexity $O(k\cdot 2^{k+1})$ and $O(2^{k+1})$ memory. Therefore, the total time complexity is $O(n_{\mathcal{B}'} + k\cdot 2^{k+1})$.

To get the optimal performance of our attack, we should carefully choose the parameters Γ and λ in the linear approximations. The experiments show that there are many large correlations based on condition masking that can be used in our attack. For example, for a condition mask $\lambda = 0x00f$, we choose 3 linear masks in the following Table 2, the experimental results show $\Delta(h^{\Gamma}_{\mathcal{B}'_{t+1}}) \approx 2^{-2.6}$, where $\Gamma = ((0x1f,\mathbf{0}),(0x1d,\mathbf{0}),(0x15,0x1))$. And $\Delta(h^{\Gamma'}) \approx 2^{-6.7}$, so we conclude from Eq.(5) that the data complexity is $n_{\mathcal{B}'} \approx 2^{22.7}$. In this example, we can recover the $k = 17$-bit subkey. Let us look at the time complexity in this

Table 2. Example: $\lambda = 0x00f$

λ	γ	ω	$E[\Delta(h^{\Lambda}_{\mathcal{B}'_{t+1}})]$
$0x00f$	$(1,1,1,1,1)$	0	$2^{-3.7}$
	$(1,1,1,0,1)$	0	$2^{-3.7}$
	$(1,0,1,0,1)$	$0x1$	$2^{-7.6}$

[9] $E[\Delta(h^{\Gamma}_{\mathcal{B}'^i_{t+1}})]$ dose not depend on t.

case. The pre-computation 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 $n_{\mathcal{B}'} = 2^{22.7}$ to compute $\mathcal{H}$, so the total time is $2^{22.7} + 2^{21.1}$.

5 Practical Implementation

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

We choose the condition mask $\lambda = 0x00f$ and $\gamma_1 = 0x1f, \omega_1 = \mathbf{0}, \gamma_2 = 0x1d, \omega_2 = \mathbf{0}, t' = 1, n_{\mathcal{B}'} = 2^{24}$ (slightly more than the theoretical estimate $2^{23.1}$) in the experiments. The condition bits $\mathcal{B}'^i_{t+1} = B^i_{t+3}$. We first collect $n_{\mathcal{B}'}$ frames for a random key and store them in a binary file. It takes about 4 minutes and 80MB to fulfill this task. With these samples, we run Algorithm 1 in Appendix A to recover the key. 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 [17], our attack can be easily carried out in real time on a single PC.

Our attack is repeated 6000 times with different randomly generated keys and IVs. In our experiments, the right key does not always rank first. The reason is that when our guess is wrong, the distribution of $g^{\Lambda}(B^i_{t+3})$ does not behave exactly as the uniform distribution from the Table 1. We take the first 256 candidates in the list as the possible keys for each run (corresponding to the 256 key candidates equivalent to each other in the experiment of [17]). There is only one correct key in their equivalent key candidates, thus they also need to test these 256 equivalent key candidates to recover the right key. The success probability of our attack is about 38.6% in this case, which can be raised very high by running it several times or by taking more candidates in the rank list. Note that in [17], the experiments are only carried out in the basic bitwise level with 2^{26} frames and repeated 30 times for a fixed key. If the key is changed, the precomputation of the attack in [17] has to be done again. This fact greatly weakens the practical effect of their attack.

During the experiments, we also found many other different condition masks that can improve the attack in [17], some of which are listed in Table 3. The detailed description of one run of our attack can be found in the full version of the paper.

Table 3. The complexities of our attack with different condition masks

mask	$(\gamma_1, \cdots, \gamma_s)$	$(\omega_1, \cdots, \omega_s)$	*Precom*	*Time*	*Frames*	*Memory*
$0x00f$	$(1f, 1d)$	$(0, 0)$	$2^{21.1}$	2^{27}	$2^{23.1}$	2^{17}
$0x00f$	$(1f, 1d, 15)$	$(0, 0, 1)$	$2^{21.1}$	2^{27}	$2^{22.7}$	2^{17}
$0x101f$	$(21, 23, 31, 35)$	$(0, 0, 0, 0)$	$2^{29.6}$	$2^{30.6}$	$2^{21.4}$	2^{25}
$0x007f$	$(21, 23, 33, 37)$	$(0, 0, 0, 0)$	$2^{33.9}$	$2^{34.9}$	$2^{20.2}$	2^{29}

6 Conclusions

In this paper, we have introduced a new cryptanalytic technique, called condition masking, to characterize the conditional correlation attacks on stream ciphers. Based on this new concept, we have investigated the conditional correlations of the two-level E0 scheme and found many useful conditional correlations for the first time. Combined these correlations with the vectorial approach, we studied the practical security of two-level E0 and developed the best and most threatening *known-IV* attack on the real Bluetooth encryption scheme so far. Our attacks have been fully implemented in C code on one core of a single PC and are repeated thousands 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 new method. We believe our new method is generic and applicable to other stream and block ciphers as well. It is our future work to study the practical ciphertext-only attack on the real Bluetooth encryption scheme using the condition masking method. Table 4 gives a comparison of our attacks with the best previous attacks on two-level E0.

Table 4. Comparison of our attacks with the previous attacks on two-level E0

Attack	*Precom*	*Time*	*Frames*	*Memory*
[7]	-	2^{73}	-	2^{51}
[8]	2^{80}	2^{65}	2	2^{80}
[10]	2^{80}	2^{70}	45	2^{80}
[18]	-	2^{40}	2^{35}	2^{35}
[17]	2^{38}	2^{38}	$2^{23.8}$	2^{33}
Ours	$2^{21.1}$	2^{27}	$2^{22.7}$	2^{17}
Ours	$2^{29.6}$	$2^{30.6}$	$2^{21.4}$	2^{25}
Ours	$2^{33.9}$	$2^{34.9}$	$2^{20.2}$	2^{29}

References

1. Armknecht, F., Krause, M.: Algebraic attacks on combiners with memory. In: Boneh, D. (ed.) CRYPTO 2003. LNCS, vol. 2729, pp. 162–175. Springer, Heidelberg (2003)
2. Baignères, T., Junod, P., Vaudenay, S.: How far can we go beyond linear cryptanalysis? In: Lee, P.J. (ed.) ASIACRYPT 2004. LNCS, vol. 3329, pp. 432–450. Springer, Heidelberg (2004)
3. SIG Bluetooth. Specification of the bluetooth system. volume 4.0 (2010)
4. Canteaut, A., Trabbia, M.: Improved fast correlation attacks using parity-check equations of weight 4 and 5. In: Preneel, B. (ed.) EUROCRYPT 2000. LNCS, vol. 1807, pp. 573–588. Springer, Heidelberg (2000)
5. Chose, P., Joux, A., Mitton, M.: Fast correlation attacks: An algorithmic point of view. In: Knudsen, L.R. (ed.) EUROCRYPT 2002. LNCS, vol. 2332, pp. 209–221. Springer, Heidelberg (2002)

6. Courtois, N.T.: Fast algebraic attacks on stream ciphers with linear feedback. In: Boneh, D. (ed.) CRYPTO 2003. LNCS, vol. 2729, pp. 176–194. Springer, Heidelberg (2003)
7. Fluhrer, S.R., Lucks, S.: Analysis of the E_0 encryption system. In: Vaudenay, S., Youssef, A.M. (eds.) SAC 2001. LNCS, vol. 2259, pp. 38–48. Springer, Heidelberg (2001)
8. Fluhrer, S.R., Cisco Systems Inc.: Improved key recovery of level 1 of the bluetooth encryption system. Cambridge University Press (2002), http://eprint.iacr.org/2002/068
9. Golić, J.: Correlation properties of a general binary combiner with memory. Journal of Cryptology 9, 111–126 (1996)
10. Golić, J.D., Bagini, V., Morgari, G.: Linear cryptanalysis of bluetooth stream cipher. In: Knudsen, L.R. (ed.) EUROCRYPT 2002. LNCS, vol. 2332, pp. 238–255. Springer, Heidelberg (2002)
11. Hermelin, M., Nyberg, K.: Correlation properties of the bluetooth combiner. In: Song, J.S. (ed.) ICISC 1999. LNCS, vol. 1787, pp. 17–29. Springer, Heidelberg (2000)
12. Johansson, T., Jönsson, F.: Improved fast correlation attacks on stream ciphers via convolutional codes. In: Stern, J. (ed.) EUROCRYPT 1999. LNCS, vol. 1592, pp. 347–362. Springer, Heidelberg (1999)
13. Johansson, T., Jönsson, F.: Fast correlation attacks through reconstruction of linear polynomials. In: Bellare, M. (ed.) CRYPTO 2000. LNCS, vol. 1880, pp. 300–315. Springer, Heidelberg (2000)
14. Krause, M.: BDD-based cryptanalysis of keystream generators. In: Knudsen, L.R. (ed.) EUROCRYPT 2002. LNCS, vol. 2332, pp. 222–237. Springer, Heidelberg (2002)
15. Lee, S., Chee, S., Park, S., Park, S.: Conditional correlation attack on nonlinear filter generators. In: Kim, K.-C., Matsumoto, T. (eds.) ASIACRYPT 1996. LNCS, vol. 1163, pp. 360–367. Springer, Heidelberg (1996)
16. Löhlein, B.: Attacks based on conditional correlations against the nonlinear filter generator, http://eprint.iacr.org/2003/020
17. Lu, Y., Meier, W., Vaudenay, S.: The conditional correlation attack: A practical attack on bluetooth encryption. In: Shoup, V. (ed.) CRYPTO 2005. LNCS, vol. 3621, pp. 97–117. Springer, Heidelberg (2005)
18. Lu, Y., Vaudenay, S.: Cryptanalysis of bluetooth keystream generator two-level E0. In: Lee, P.J. (ed.) ASIACRYPT 2004. LNCS, vol. 3329, pp. 483–499. Springer, Heidelberg (2004)
19. Lu, Y., Vaudenay, S.: Faster correlation attack on bluetooth keystream generator E0. In: Franklin, M. (ed.) CRYPTO 2004. LNCS, vol. 3152, pp. 407–425. Springer, Heidelberg (2004)
20. Lu, Y., Vaudenay, S.: Cryptanalysis of an e0-like combiner with memory. Journal of Cryptology 21, 430–457 (2008)
21. Matsui, M.: Linear cryptanalysis method for DES cipher. In: Helleseth, T. (ed.) EUROCRYPT 1993. LNCS, vol. 765, pp. 386–397. Springer, Heidelberg (1994)
22. Meier, W., Staffelbach, O.: Fast correlation attacks on certain stream ciphers. Journal of Cryptology 1, 159–176 (1989)
23. Meier, W., Staffelbach, O.: Correlation properties of combiners with memory in stream ciphers. Journal of Cryptology 5, 67–86 (1992)
24. Petrakos, N., Dinolt, G.W., Michael, J.B., Stanica, P.: Cube-type algebraic attacks on wireless encryption protocols. Computer 42(10), 103–105 (2009)

25. Preneel, B.: Stream ciphers: Past, present and future (2010)
26. Saarinen, M.: Re: Bluetooth and E0. Posting to Sci. Crypt. Research 2(09) (2000)
27. Shaked, Y., Wool, A.: Cryptanalysis of the bluetooth E_0 cipher using oBDD's. In: Katsikas, S.K., López, J., Backes, M., Gritzalis, S., Preneel, B. (eds.) ISC 2006. LNCS, vol. 4176, pp. 187–202. Springer, Heidelberg (2006)
28. Siegenthaler, T.: Decrypting a class of stream ciphers using ciphertext only. IEEE Transactions on Computers C-34, 81–85 (1985)

A The Key Recovery Distinguisher Based on Condition Masking

Algorithm 1. The key recovery method based on condition masking

Parameters: n, λ, $\mathcal{B}$ and $D_{f_{\mathcal{B}'}}$
input:
1: for $i = 1, 2, \ldots, n$, $\mathcal{B}_i'^K$ for all k-bit K
2: $Z_i^{\mathcal{K}} = f_{\mathcal{B}'}(\mathcal{B}_i^{*\mathcal{K}}, X_i)$ for the right 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 the wrong keys K such that $K \neq \mathcal{K}$
Goal: find $\mathcal{K}$
Processing:
4: **for** all k-bit K **do**
5: $G(K) \leftarrow 0$
6: **for** $i = 1, \cdots, n$ **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 $G(\mathcal{K})$

Theorem 10. *Given a condition mask λ, the above Algorithm 1 solves the problem in Definition 7 with $n_{\mathcal{B}'} = \frac{4k \log 2}{E[\Delta(f_{\mathcal{B}'})]}$ samples and the time complexity is $O(n_{\mathcal{B}'} \cdot 2^k)$, 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*

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

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

for all k-bit K and $i = 1, 2, \cdots, n$, 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 $O(n_{\mathcal{B}'} + k2^{k+1})$ with pre-computation $O(k2^k)$. Besides, $|\mathcal{B}'| = k$.

B The Linear Approximation of Two-level E0

Following the specification in [3], the last generated 128 bits $S^i_{[-127,\cdots,0]}$ in the first level are arranged in octets denoted by $S[0],\cdots,S[15]$, e.g., $S[0] = (S^i_{-127}S^i_{-126}\cdots S^i_{-120})$, where $S^i_{[-127,\cdots,0]} = R^i_{[-127,\cdots,0]} \oplus \alpha^i_{[-127,\cdots,0]}$. From Section 2, we have $V^i_{[1,\cdots,128]} = G_3(R^i_{[-127,\cdots,0]}) \oplus G_3(\alpha^i_{[-127,\cdots,0]})$, where G_3 is depicted in Fig.2. For brevity, we define $(U^i_1,\cdots,U^i_{128}) = G_3(R^i_{[-127,\cdots,0]})$. According to Fig.2, $V^i_{[1,\cdots,24]}$ can be expressed as $V^i_{t'} = U^i_{t'} \oplus \alpha^i_{t_1} \oplus \alpha^i_{t_2} \oplus \alpha^i_{t_3} \oplus \alpha^i_{t_4}$, for $t' = 1,\cdots,24$, where t_1, t_2, t_3, t_4 are the fixed time instants of α^i before the application of G_3.

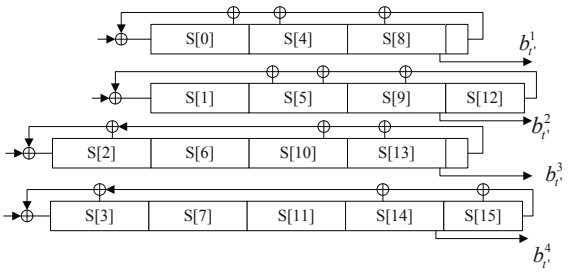

Fig. 2. Distribution of the last 128 bits in the first level

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

$$z_{t'} \oplus H_{t'}(K) \oplus H'_{t'}(P^i) = \alpha^i_{t_1} \oplus \alpha^i_{t_2} \oplus \alpha^i_{t_3} \oplus \alpha^i_{t_4} \oplus \beta^i_{t'}, \text{ for } t' = 1,\cdots,24. \quad (6)$$

Given a linear mask γ with $|\gamma| = l$, let $Z^i_{t'} = (z^i_{t'},\cdots,z^i_{t'+l-1})$. Since at level two (in Fig.2), the 128-bit keystream S^i_t are loaded in the reverse order of that at level one, then Eq.(6) can be rewritten with the linear mask notation as

$$\bar{\gamma} \cdot (Z^i_{t'} \oplus \mathcal{L}_{t'}(K) \oplus \mathcal{L}'_{t'}(P^i)) = \bigoplus_{j=1}^{4} (\gamma \cdot C^i_{t_j}) \oplus \bar{\gamma} \cdot C^i_{t'}, \quad (7)$$

for $i = 1,\cdots,n$ and $\mathcal{L}_{t'}, \mathcal{L}'_{t'}$ are fixed linear functions which can be derived from $H_{t'}, H'_{t'}$. Here we have $t' \in \bigcup_{d=0}^{2}\{8d+1,\cdots,8d+9-l\}$.[10] Eq.(7) corresponds to the case of $\lambda = \mathbf{1}_u$.

[10] From Eq.(7), the time instant t_j in $C^i_{t_j}$ are continuous, so the approximation is only set up in this requirement.