Linear Cryptanalysis of Bluetooth Stream Cipher

Jovan Dj. Golić, Vittorio Bagini, and Guglielmo Morgari

Rome CryptoDesign Center, Gemplus
Via Pio Emanuelli 1, 00143 Rome, Italy
{jovan.golic, vittorio.bagini, guglielmo.morgari}@gemplus.com

Abstract. A general linear iterative cryptanalysis method for solving binary systems of approximate linear equations which is also applicable to keystream generators producing short keystream sequences is proposed. A linear cryptanalysis method for reconstructing the secret key in a general type of initialization schemes is also developed. A large class of linear correlations in the Bluetooth combiner, unconditioned or conditioned on the output or on both the output and one input, are found and characterized. As a result, an attack on the Bluetooth stream cipher that can reconstruct the 128-bit secret key with complexity about 2^{70} from about 45 initializations is proposed. In the precomputation stage, a database of about 2^{80} 103-bit words has to be sorted out.

Key words Linear cryptanalysis, linear correlations, iterative probabilistic decoding, reinitialization.

1 Introduction

Bluetooth$^{\text{TM}}$ is a standard for wireless short-range connectivity specified by the Bluetooth$^{\text{TM}}$ Special Interest Group in [1]. The specification defines a stream cipher algorithm E_0 to be used for point-to-point encryption within the Bluetooth network. The algorithm consists of a keystream generator, derived from the well-known summation generator, and an initialization scheme which is based on the keystream generator. The size of the secret key used for encryption is 128 bits, and the initialization vector (IV) consists of 74 bits, 26 of which are derived from a real-time clock, while the remaining 48 are the address bits depending on users. The internal state of the keystream generator is 132 bits long, and the keystream sequences produced are very short, that is, at most 2745 bits for each initialization vector. The description of the Bluetooth security protocol given in [1] is not quite clear and, according to some interpretations, a number of security weaknesses of the protocol are presented in [12].

The keystream generator is a binary combiner composed of four linear feedback shift registers (LFSR's) of total length 128 that are combined by a nonlinear function with 4 bits of memory which is a modified combining function of the summation generator. This modification turns out to be important as it reduces some correlation weaknesses of the summation generator identified in [17] and [10]. Some further interesting improvements to this end which require minor modifications of the combining function are proposed in [14]. However, according to [1] and [14], the short keystream sequences should prevent the correlation attacks based on the correlation properties of the Bluetooth combiner.

Due to a large size of the internal state, the complexity of general time-memory or time-memory-data tradeoff attacks (e.g., see [9]) for realistic amounts of known keystream data seems to be higher than the complexities reported below. Besides, as such attacks aim at recovering an internal or the initial state of the keystream generator, they are not directly applicable to Bluetooth if the objective is to recover the secret key because of the initialization scheme used. The basic divide-and-conquer attack on the Bluetooth combiner directly follows from the similar attack [3] on the summation generator (also see [12]). In such an attack, 89 bits of the initial states of the three shortest LFSR's along with 4 initial memory bits are guessed. This allows to recover the output sequence of the longest LFSR from the keystream sequence. Altogether, about 132 keystream bits are needed to identify the correct guess. The same attack applies to the initialization scheme, so that the secret key can be reconstructed in about 2^{93} steps from just one IV, where the step complexity is the same as in the exhaustive search method. If one guesses 56 bits of the two shortest LFSR's and applies a sort of the branching method [9] for producing a system of linear equations, then the initial states of the other two LFSR's can be recovered in about 2^{84} steps, and some optimizations are possible [4]. The secret key can be obtained in a similar way.

The main objective of this paper is to identify a large class of linear correlations in the Bluetooth combiner which, in spite of the short keystream sequences, enable one to reconstruct not only the LFSR initial states, but also the secret key from a relatively small number of IV's. More precisely, we consider the unconditioned linear correlations, the linear correlations conditioned on the output, and the linear correlations conditioned on both the output and one guessed input. The resulting system of linear equations holding with probabilities different from one half can then be solved by a general linear iterative cryptanalysis method similar to iterative probabilistic decoding algorithms used in fast correlation attacks. The secret key can be recovered by a related linear cryptanalysis method from a number of IV's. The total complexity is about 2^{70} steps, with the step complexity comparable to one of the exhaustive search method, the required number of IV's is about 45, and the precomputation stage consists in sorting out a database of about 2^{80} 103-bit words.

Description of the Bluetooth stream cipher is provided in Section 2. The linear correlations are explained and characterized in Section 3, the general method for solving binary systems of approximate linear equations and its application to the Bluetooth keystream generator are presented in Section 4, and a linear cryptanalysis method for initialization schemes is proposed in Section 5. Optimal choices of parameters for concrete attacks are discussed in Section 6 and conclusions are given in Section 7. Analogous linear correlations computed for the modified Bluetooth combiner [14] are displayed in the Appendix.

2 Description of Bluetooth Stream Cipher

The description is based on [1], but only the details relevant for our linear cryptanalysis method will be presented. The main component of the Bluetooth stream cipher algorithm is the keystream generator (Bluetooth combiner) which is derived from the well-known summation generator with four input LFSR's. The LFSR lengths are 25, 31, 33, and 39 (128 in total) and all the feedback polynomials are primitive and have 5 nonzero terms each. All the LFSR's are regularly clocked and their binary outputs are combined by a nonlinear function with 4 bits of memory. Let $\mathbf{x}^i = (x_t^i)_{t=0}^{\infty}$ denote the output sequence of LFSR$_i$, $1 \leq i \leq 4$, where the LFSR's are indexed in order of increasing length. The internal memory of the combiner at time t consists of 4 memory bits $C_t = (c_t, c_{t-1})$, where 2 carry bits $c_t = (c_t^0, c_t^1)$ are defined in terms of 2 auxiliary carry bits $s_t = (s_t^0, s_t^1)$. Let $\mathbf{z} = (z_t)_{t=0}^{\infty}$ denote the output sequence of the combiner. Then the output sequence of the combiner is defined recursively by

$$z_t = x_t^1 \oplus x_t^2 \oplus x_t^3 \oplus x_t^4 \oplus c_t^0 \tag{1}$$

$$c_{t+1}^0 = s_{t+1}^0 \oplus c_t^0 \oplus c_{t-1}^0 \oplus c_{t-1}^1, \qquad c_{t+1}^1 = s_{t+1}^1 \oplus c_t^1 \oplus c_{t-1}^0 \tag{2}$$

$$(s_{t+1}^0, s_{t+1}^1) = \left\lfloor \frac{x_t^1 + x_t^2 + x_t^3 + x_t^4 + 2c_t^1 + c_t^0}{2} \right\rfloor \tag{3}$$

with integer summation in the last equation, where the initial 4 memory bits $(c_0^0, c_0^1, c_{-1}^0, c_{-1}^1)$ have to be specified. Note that in the summation generator the memory consists of only 2 bits of the carry s_t, i.e., $c_t = s_t$.

Due to frequent resynchronizations, the maximal keystream sequence length produced from a given initial state of the keystream generator is only 2745 bits. The initial state consists of 128 bits defining the initial LFSR states and 4 initial memory bits. They are produced by an initialization scheme from (at most) 128 secret key bits and the known 74-bit IV consisting of 48 address bits depending on users and of variable 26 bits derived from a real-time master clock. The secret key itself is derived from some secret and some known random information by another algorithm, which is irrelevant for our cryptanalysis.

The initialization scheme is the Bluetooth combiner initialized with some secret key bits and some IV bits, while the initial 4 memory bits are all set to 0. The remaining secret key bits and IV bits are added modulo 2, one at a time, to the feedback bits of individual LFSR's, for a number of times depending on the LFSR. The details are not important, except for the fact that the LFSR sequences in the initialization scheme linearly depend on the secret key and IV. The combiner is clocked 200 times and the last produced 128 output bits are permuted in a specified way to define the LFSR initial states, while the last 4 memory bits are used as the initial 4 memory bits for keystream generation.

3 Linear Correlations in Bluetooth Combiner

The basis of the linear cryptanalysis method to be developed are linear relations among the input bits to the Bluetooth combiner that hold with probabilities different from one half, in the probabilistic model in which the input sequences are modeled as purely random, i.e., as mutually independent sequences of independent and balanced (uniformly distributed) binary random variables. Such linear relations are called linear correlations since they are directly or indirectly dependent on the known output sequence. The first point to analyze is the asymptotic distribution of the 4 memory bits in this probabilistic model, if the initial 4 memory bits are either fixed or purely random. In the summation generator, due to the fact that the nonlinear function (3) is not balanced, it follows that the 2 carry bits are not balanced asymptotically, and this is the main source of a number of correlation weaknesses derived and exploited in [17] and [10]. However, in the case of Bluetooth, due to the introduced linear functions (2), C_{t+1} is a balanced function of C_t and $X_t = (x_t^1, x_t^2, x_t^3, x_t^4)$ and hence C_t is balanced for every t if it is balanced for $t = 0$. Moreover, this also holds asymptotically, when t increases, if the initial memory bits, C_0, are fixed, because the underlying Markov chain is ergodic, and the convergence to the stationary distribution is very fast.

Consider a block of m consecutive output bits, $Z_t^m = (z_t, z_{t-1}, \cdots, z_{t-m+1})$ as a function of the corresponding block of m consecutive inputs $X_t^m = (X_t, X_{t-1}, \cdots, X_{t-m+1})$ and the preceding memory bits C_{t-m+1}. Assume that X_t^m and C_{t-m+1} are balanced and mutually independent. Then, according to [7], if $m \geq 5$, then there must exist linear correlations between the output and input bits, but they may also exist if $m \leq 4$. As the correlations are time invariant, we introduce the notation $Z^m = F^m(X^m, C)$, where $Z^m = (z_j)_{j=0}^{m-1}$ and $X^m = (X_j)_{j=0}^{m-1}$. By virtue of the linear output function (1), it follows that $F^m(X^m, C)$ is a balanced function that is also balanced for any fixed C. The input block X^m of $4m$ bits can be rearranged into $X^m = (X_i^m)_{i=1}^4$, where $X_i^m = (x_j^i)_{j=0}^{m-1}$ is the i-th input block of m bits, corresponding to the output of LFSR_i. Then (1) implies that $F^m(X^m, C)$ is balanced for any fixed X_i^m and, also, for any fixed X_i^m and C combined.

Let f and g be two Boolean functions of an n-bit input vector which is assumed to be uniformly distributed. Then the correlation coefficient between f and g conditioned on a subset $\mathcal{X} \subseteq \{0,1\}^n$ is defined as

$$\begin{aligned} c(f, g \mid \mathcal{X}) &= \Pr(f(X) = g(X) \mid X \in \mathcal{X}) - \Pr(f(X) \neq g(X) \mid X \in \mathcal{X}) \\ &= \frac{1}{|\mathcal{X}|} \sum_{X \in \mathcal{X}} (-1)^{f(X) \oplus g(X)} = \frac{1}{|\mathcal{X}|} \sum_{X \in \mathcal{X}} (-1)^{f(X)} (-1)^{g(X)}. \end{aligned} \tag{4}$$

The correlation coefficients conditioned on $\mathcal{X}$ between f and all linear functions l are thus determined by the Walsh transform of a real-valued function defined as $(-1)^{f(X)}$ for $X \in \mathcal{X}$ and as 0 otherwise. They can be computed by the fast Walsh transform algorithm of complexity $O(n2^n)$.

All the correlations of interest to be described below correspond to (4) and were feasible to compute exhaustively for $m \leq 6$. All significant correlation coefficients were also tested by computer simulations on sufficiently long output sequences. It turns out that for $m \leq 3$ the correlation coefficients are equal to zero in all the cases. For $4 \leq m \leq 6$, it turns out that relatively large absolute values of the correlation coefficients along with the associated input linear functions can be characterized in terms of the underlying conditions. In addition, it also turns out that the Boolean functions specifying the signs of the correlation coefficients have relatively simple characterizations. The Boolean sign function is here defined as $\mathrm{sign}(c) = 0$ for $c > 0$ and $\mathrm{sign}(c) = 1$ for $c < 0$.

3.1 Unconditioned Linear Correlations

The first type of correlations to be considered are the correlations between linear functions of input bits and linear functions of output bits, as introduced in [7]. Namely, let $\mathbf{W} \cdot X^m = \bigoplus_{i=1}^{4} \bigoplus_{j=0}^{m-1} w_{ij} x_j^i$ and $\mathbf{v} \cdot Z^m = \bigoplus_{j=0}^{m-1} v_j z_j$ denote two such linear functions defined by a matrix $\mathbf{W}$ and a vector $\mathbf{v}$, respectively. We want to find all $\mathbf{W}$ and $\mathbf{v}$ such that the correlation coefficient $c(\mathbf{W} \cdot X^m, \mathbf{v} \cdot Z^m)$ is relatively large in absolute value. Define the (column) weights of $\mathbf{W}$ as $w_j = \sum_{i=1}^{4} w_{ij}$, $0 \leq j \leq m-1$. Then the main property, observed in [14], which follows from the symmetry of the combiner output and next-state functions with respect to 4 input variables, is that the correlation coefficient depends on $\mathbf{v}$ and only on the weights of $\mathbf{W}$, i.e., on the weight vector $\mathbf{w} = (w_j)_{j=0}^{m-1}$.

4-bit case There are 96 pairs of input/output linear functions that are mutually correlated, with nonzero correlation coefficients $\pm 1/16$. The output and input linear functions respectively have the weight patterns $(1, v_1, v_2, 1)$ and $(4, w_1, w_2, 4)$ such that $(w_1)_2 \neq v_1$, where $(w_1)_2 \stackrel{\mathrm{def}}{=} w_1 \bmod 2$. Each of 2 output linear functions with $v_2 = 0$ is correlated to $16 = 8 \times 2$ input linear functions with $w_2 \in \{0, 4\}$. Each of 2 output linear functions with $v_2 = 1$ is correlated to $32 = 8 \times 4$ input linear functions with $w_2 = 3$. One of these 96 pairs was theoretically found out in [14]. For all the pairs,

$$\mathrm{sign}(c) \;=\; v_1 \oplus (\lfloor w_1/2 \rfloor)_2 \oplus \lfloor w_2/4 \rfloor. \tag{5}$$

5-bit case There are 8 nonzero correlation coefficients $\{\pm 25/256, \pm 5/256, \pm 1/64, \pm 1/256\}$. The largest absolute value is attained by the following 16 pairs of input/output linear functions: each of 2 output linear functions with pattern $(1, v_1, 1, 1, 1)$ is correlated to 8 input linear functions with weight pattern $(4, w_1, 4, 4, 4)$ such that $(w_1)_2 \neq v_1$. For such pairs, the sign function is given by the first two terms on the right-hand-side of (5).

6-bit case There are 12 nonzero correlation coefficients $\{\pm 25/256, \pm 25/1024, \pm 5/256, \pm 5/1024, \pm 1/256, \pm 1/1024\}$. The largest absolute value is attained by the following 16 pairs of input/output linear functions: each of 2 output linear functions with pattern $(1, v_1, 0, 0, 0, 1)$ is correlated to 8 input linear functions with weight pattern $(4, w_1, 0, 0, 0, 4)$ such that $(w_1)_2 = v_1$. For such pairs, the sign function is given by the second term on the right-hand-side of (5).

3.2 Linear Correlations Conditioned on Output

The second type of correlations to be considered are the correlations between linear functions of input bits and the all-zero function when conditioned on the output bits. Namely, for every output Z^m, we would like to find all $\mathbf{W}$ such that the conditioned correlation coefficient $c(\mathbf{W} \cdot X^m, 0 \mid F^m(X^m, C) = Z^m)$ is relatively large in absolute value. One can also prove that for any given Z^m, the conditioned correlation coefficient depends only on the weight vector $\mathbf{w}$. The conditioned correlation coefficients are generally larger than the unconditioned ones, because they fully exploit the information contained in the known output sequence. Recall that $Z^m = (z_0, z_1, \cdots, z_{m-1})$.

4-bit case For each output value Z^4, there are 96 input linear functions with nonzero correlation coefficients, equal to $\pm 1/16$, with weight pattern $(4, w_1, w_2, 4)$ where w_1 is arbitrary and $w_2 \in \{0, 3, 4\}$. For such functions,

$$\text{sign}(c) \;=\; (1 \oplus z_1)(1 \oplus (w_1)_2) \oplus (\lfloor w_1/2 \rfloor)_2 \oplus \lfloor w_2/4 \rfloor \oplus z_0 \oplus z_2(w_2)_2 \oplus z_3. \quad (6)$$

5-bit case For each output value Z^5, there are 12 nonzero correlation coefficients with 6 different absolute values. The largest absolute value 29/256 is attained by 8 input linear functions with weight pattern $(4, w_1, 4, 4, 4)$ such that $(w_1)_2 = z_1$. The second largest absolute value 21/256 is attained by 8 input linear functions with weight pattern $(4, w_1, 4, 4, 4)$ such that $(w_1)_2 \neq z_1$. For all 16 functions,

$$\text{sign}(c) \;=\; (1 \oplus z_1)(1 \oplus (w_1)_2) \oplus (\lfloor w_1/2 \rfloor)_2 \oplus z_0 \oplus z_2 \oplus z_3 \oplus z_4. \quad (7)$$

6-bit case For each output value Z^6, there are 100 nonzero correlation coefficients with 50 different absolute values. Except for the value 83/1024, corresponding to another type of input linear functions, the largest 7 absolute values are attained by exactly 16 input linear functions with weight pattern $(4, w_1, 0, 0, 0, 4)$ and depend on $(w_1)_2 \oplus z_1 \oplus z_4$ and (z_2, z_4) in a way shown in the following table. For such functions,

$$\text{sign}(c) \;=\; (\lfloor w_1/2 \rfloor)_2 \oplus z_0 \oplus z_1(w_1)_2 \oplus z_5. \quad (8)$$

$(w_1)_2$	$\neg(z_1 \oplus z_4)$				$z_1 \oplus z_4$			
(z_2, z_4)	(0,0)	(1,1)	(1,0)	(0,1)	(0,0)	(1,0)	(1,1)	(0,1)
$\lvert c \rvert$	$\frac{139}{1024}$	$\frac{129}{1024}$	$\frac{119}{1024}$	$\frac{113}{1024}$	$\frac{79}{1024}$	$\frac{79}{1024}$	$\frac{73}{1024}$	$\frac{69}{1024}$

3.3 Linear Correlations Conditioned on Output and One Input

The third type of correlations to be considered are the correlations between linear functions of 3 inputs and the all-zero function when conditioned on the output and one assumed input. More precisely, with the notation $X^m_{2-4} = (X^m_i)^4_{i=2}$, let $\mathbf{W} \cdot X^m_{2-4} = \bigoplus_{i=2}^{4} \bigoplus_{j=0}^{m-1} w_{ij} x^i_j$ denote such a linear function of 3 inputs. For every assumed input X_1^m and every possible output Z^m, we would like to find all $\mathbf{W}$ such that the conditioned correlation coefficient $c(\mathbf{W}{\cdot}X_{2-4}^m, 0|X_1^m, F^m(X^m, C) = Z^m)$ is relatively large in absolute value. One can similarly prove that for any assumed X_1^m and any given Z^m, the conditioned correlation coefficient depends only on the weight vector $\mathbf{w} = (w_j)_{j=0}^{m-1}$, where now $w_j = \sum_{i=2}^4 w_{ij}$, $0 \le j \le m-1$. These correlation coefficients are generally larger than the ones conditioned only on the output, because of the information provided by one known input. Recall that $Z^m = (z_0, z_1, \cdots, z_{m-1})$ and $X_1^m = (x_0^1, x_1^1, \cdots, x_{m-1}^1)$.

4-bit case For each input value X_1^4, there are 4 nonzero correlation coefficients $\pm 1/4$ and $\pm 1/8$. The absolute value $1/4$ is attained by an average of 2 (out of 8) input linear functions with weight pattern $(3, w_1, 3, 3)$ such that $z_2 \neq x_2^1$ and $(w_1)_2 \neq z_1 \oplus x_1^1$. For such functions,

$$\mathrm{sign}(c) \;=\; 1 \oplus \lfloor w_1/2 \rfloor \oplus z_0 \oplus z_3 \oplus x_0 \oplus x_1 \oplus x_2 \oplus x_3 \oplus z_1 x_1. \tag{9}$$

The absolute value $1/8$ is attained for every output value Z^4 by 16 (out of 32) input linear functions with weight pattern $(3, w_1, w_2, 3)$ such that $(w_1)_2 \neq z_1 \oplus x_1^1$ and $(w_2)_2 = 0$. For such functions,

$$\mathrm{sign}(c) = \lfloor w_1/2 \rfloor \oplus z_0 \oplus z_2 \lfloor w_2/2 \rfloor \oplus z_3 \oplus x_0 \oplus x_1 \oplus x_2 \lfloor w_2/2 \rfloor \oplus x_3 \oplus z_1 x_1. \tag{10}$$

So, per each X_1^4 and Z^4, $|c| = 1/4$ is on average attained by 2 out of 8 input linear functions and $|c| = 1/8$ is attained by 16 out of 32 input linear functions.

5-bit case For each input value X_1^5, there are 12 nonzero correlation coefficients with 6 different absolute values. The largest 3 absolute values are attained for every output value Z^5 by 4 (out of 8) input linear functions with weight pattern $(3, w_1, 3, 3, 3)$ such that $(w_1)_2 \neq z_1 \oplus x_1^1$. The dependence of $|c|$ on (X_1^5, Z^5) along with the average number α of the corresponding input linear functions are shown in the following table. For the remaining 4 input linear functions such that $(w_1)_2 = z_1 \oplus x_1^1$, $|c| = 1/32$ for every (X_1^5, Z^5). For all 8 functions,

$$\begin{aligned}\mathrm{sign}(c) \;=\; & 1 \oplus \lfloor w_1/2 \rfloor \oplus (w_1)_2 \oplus z_0 \oplus z_1 \oplus z_1 (w_1)_2 \oplus z_2 \oplus z_3 \oplus z_4 \\ & \oplus x_0 \oplus x_1 (w_1)_2 \oplus x_2 \oplus x_3 \oplus x_4 \oplus z_1 x_1. \end{aligned} \tag{11}$$

$\lvert c\rvert$	α	$(z_2 \oplus x_2^1, z_3 \oplus x_3^1)$
9/32	1	(0,0)
3/16	2	(0,1) or (1,0)
1/8	1	(1,1)

6-bit case For each input value X_1^6, there are 50 nonzero correlation coefficients with 25 different absolute values. The most significant absolute values are attained by 4 (out of 8) input linear functions with weight pattern $(3, w_1, 0, 0, 0, 3)$ such that $(w_1)_2 = z_1 \oplus x_1^1$, but some large absolute values are also achieved by the remaining 4 input linear functions such that $(w_1)_2 \neq z_1 \oplus x_1^1$.

For all 8 functions,

$$\begin{aligned}\mathrm{sign}(c) &= (w_1)_2 \oplus \lfloor w_1/2 \rfloor \oplus z_0 \oplus z_1 \oplus z_1(w_1)_2 \oplus z_5 \oplus x_0 \oplus x_1(w_1)_2 \oplus x_2(w_1)_2 \\ &\oplus x_4(w_1)_2 \oplus x_5 \oplus z_1x_1 \oplus z_1x_2 \oplus z_1x_4 \oplus z_4x_2(w_1)_2 \oplus z_4x_4(w_1)_2 \oplus x_1x_2 \oplus x_1x_4 \\ &\oplus x_2x_4(w_1)_2 \oplus z_1z_4x_2 \oplus z_1z_4x_4 \oplus z_1x_2x_4 \oplus z_4x_1x_2 \oplus z_4x_1x_4 \oplus x_1x_2x_4. \quad (12)\end{aligned}$$

The dependence of $|c|$ on (X_1^6, Z^6) along with the average number α of the corresponding input linear functions are shown in the following table for $|c| > 1/8$ and an average number 4.125 of such functions. For an average number 3.875 of remaining input linear functions, $|c| < 1/8$. The displayed 7 values along with 21/128 and 19/128, corresponding to other input linear functions, are the largest possible.

$\lvert c\rvert$	α	CONDITIONS	FUNCTIONS
9/32	1	$z_2 = x_2^1, z_4 = x_4^1$	$(w_1)_2 = z_1 \oplus x_1^1$
13/64	1/2	$z_2 = x_2^1, z_4 \neq x_4^1 = 1$	$(w_1)_2 = z_1 \oplus x_1^1$
3/16	9/8	$z_2 \neq x_2^1$, ($z_4 = x_4^1$ or $z_4 \neq x_4^1 = x_2^1 = 1, z_3 = x_3^1$)	$(w_1)_2 = z_1 \oplus x_1^1$
11/64	5/8	$z_4 \neq x_4^1$, ($z_2 = x_2^1, x_4^1 = 0$ or $z_2 \neq x_2^1 = x_4^1 = 1, z_3 \neq x_3^1$)	$(w_1)_2 = z_1 \oplus x_1^1$
5/32	1/8	$z_2 \neq x_2^1, z_4 \neq x_4^1, x_2^1 = x_4^1 = 0, z_3 = x_3^1$	$(w_1)_2 = z_1 \oplus x_1^1$
9/64	5/8	$z_2 \neq x_2^1, z_4 \neq x_4^1, x_2^1 = x_4^1 = 0, z_3 \neq x_3^1$	$(w_1)_2 = z_1 \oplus x_1^1$
		$z_4 \neq x_4^1, z_2 = x_2^1, z_3 = x_3^1$	$(w_1)_2 \neq z_1 \oplus x_1^1$
17/128	1/8	$z_4 \neq x_4^1, z_2 \neq x_2^1 = x_4^1 = 1, z_3 = x_3^1$	$(w_1)_2 \neq z_1 \oplus x_1^1$

4 Linear Iterative Cryptanalysis of Keystream Generator

The objective of the linear cryptanalysis of the Bluetooth keystream generator is to reconstruct 128 bits of the LFSR initial states from a given segment of the keystream sequence of length at most 2745 bits, by using the linear correlations described in Section 3. Accordingly, the starting point of the cryptanalysis is a set of linear equations in the initial state bits which hold with probabilities different from one half. The aim is to find a solution to this system that is consistent with the given probabilities.

4.1 Solving Approximate Linear Systems

Let $\mathbf{x} = (x_j)_{j=1}^k$ be a vector of k binary variables and let $\mathbf{y} = (y_i)_{i=1}^n$ be a vector of n, $n \geq k$, binary variables that are defined as linear functions of $\mathbf{x}$, that is, $y_i = l_i(\mathbf{x})$, $1 \leq i \leq n$. In matrix notation, $\mathbf{y} = \mathbf{G}^{\mathrm{T}}\mathbf{x}$, where $\mathbf{G}$ is an $k \times n$ matrix whose columns correspond to linear functions l_i, and vectors are represented as one-column matrices. It is assumed that $\mathbf{G}$ has full rank k (linearly independent rows), which means that the linear transform defined by $\mathbf{G}^{\mathrm{T}}$ is injective, so that $\mathbf{y}$ uniquely determines $\mathbf{x}$.

It is further assumed that $\mathbf{y}$ is known only probabilistically, in terms of the marginal probabilities $\Pr(y_i = 0) = p_i$, $1 \leq i \leq n$. More precisely, a probabilistic model is assumed in which the variables y_i, $1 \le i \le n$, are mutually independent. Since in this case they can take arbitrary values, define an event $\mathcal{L}$ that $\mathbf{y}$ belongs to the range of the linear transform determined by $\mathbf{G}^{\mathrm{T}}$, i.e., that y_i are linearly dependent according to $\mathbf{G}^{\mathrm{T}}$, i.e., that the linear system $\mathbf{y} = \mathbf{G}^{\mathrm{T}}\mathbf{x}$ has a (unique) solution in $\mathbf{x}$. Now, in this model the most likely solution to the linear system is the one that maximizes the conditioned block probability

$$\begin{aligned}\Pr(\mathbf{x}|\mathcal{L}) &= \Pr(\mathbf{y} = \mathbf{G}^{\mathrm{T}}\mathbf{x}|\mathcal{L}) \\ &= \frac{\Pr(\mathbf{y} = \mathbf{G}^{\mathrm{T}}\mathbf{x})}{\Pr(\mathcal{L})} = \frac{1}{\Pr(\mathcal{L})} \prod_{i=1}^{n} p_i^{1-l_i(\mathbf{x})}(1-p_i)^{l_i(\mathbf{x})}. \end{aligned} \tag{13}$$

It follows that $\Pr(\mathcal{L}) = \sum_{\mathbf{x}\in\{0,1\}^k} \prod_{i=1}^{n} p_i^{1-l_i(\mathbf{x})}(1-p_i)^{l_i(\mathbf{x})}$. Of course, 2^k steps are required to find the solution.

The problem is in fact directly related to a decoding problem for the binary linear (n, k) block code C with a generator matrix $\mathbf{G}$ on a time-varying memoryless binary symmetric channel (BSC) with error probabilities $1-p_i$, $1 \le i \le n$. Namely, in a probabilistic model in which the codewords are equiprobable, if the all-zero word is observed at the output of this BSC, then the posterior probability of an information word $\mathbf{x}$ is the same as the conditioned probability (13). However, our model is more appropriate as it directly deals with the problem considered and as such does not involve any communication channel (e.g., symmetry is not needed).

Another approach is to find a solution that maximizes each of the conditioned bit probabilities $\Pr(x_j|\mathcal{L})$, $1 \le j \le k$. This requires only k steps, but such probabilities have to be computed, and that requires 2^{n-k} steps if the well-known Hartmann-Rudolph algorithm [13] is applied. For linear codes, this algorithm minimizes the decoding error probability for individual symbols rather than blocks of symbols. It can also be used for computing the conditioned bit probabilities $\hat{p}_i = \Pr(y_i = 0|\mathcal{L})$, $1 \le i \le n$, which are important for iterative algorithms. In our problem, k is large and the probabilities p_i are rather close to one half, so that the decision error probabilities can be small only if $n-k$ is also large. Therefore, the Hartmann-Rudolph algorithm is computationally infeasible and numerical approximations are required.

Let $\mathbf{H}$ denote a parity-check matrix of the code C, i.e., a generator matrix of its dual code C^{d}. $\mathbf{H}$ is an $(n-k) \times n$ matrix of full rank $n-k$ such that $\mathbf{H}\mathbf{G}^{\mathrm{T}} = \mathbf{0}$. Recall that C^{d} is a binary linear $(n, n-k)$ code consisting of all the binary vectors $\mathbf{v} = (v_i)_{i=1}^{n}$ that are orthogonal to each codeword $\mathbf{y}$ from C ($\mathbf{v}\cdot\mathbf{y} = v_1y_1 \oplus \cdots \oplus v_ny_n = 0$). The dual codewords represent the linear relations among the codeword bits and are hence called the parity checks. Instead of taking into account all 2^{n-k} parity checks as in the Hartmann-Rudolph algorithm, one can only consider numerically more important parity checks having a relatively low weight, which is defined as the number of nonzero terms minus one.

Let V_i be a set of parity checks $\mathbf{v}$ involving the i-th codeword bit y_i, i.e., such that $v_i = 1$. Let $c_i = 2p_i - 1$ and $\hat{c}_i = 2\hat{p}_i - 1$ denote the corresponding unconditioned and conditioned correlation coefficients of y_i, respectively. Then, according to [11], we get an approximate expression

$$\hat{c}_i \; = \; /c_i + \sum_{\mathbf{v} \in V_i} \prod_{j=1:\; v_j=1, j \neq i}^{n} c_j/ \tag{14}$$

where the clipping function $/(\cdot)/$ ensures that $|\hat{c}_i| \leq 1$. Interestingly, this expression can also be obtained as the limit form, when all c_i tend to zero, of the well-known expression (e.g., see [18])

$$\frac{1 - \hat{c}_i}{1 + \hat{c}_i} \; = \; \frac{1 - c_i}{1 + c_i} \prod_{\mathbf{v} \in V_i} \frac{1 - \prod_{j=1:\; v_j=1, j \neq i}^{n} c_j}{1 + \prod_{j=1:\; v_j=1, j \neq i}^{n} c_j} \tag{15}$$

which is used if the parity checks from each V_i are orthogonal, that is, if the i-th bit is the only bit that they share in common. Expression (14) appears to be more appropriate as the orthogonality is not required. In both expressions, the product term $\prod_{j=1:\; v_j=1, j \neq i}^{n} c_j$ represents the correlation coefficient of the binary sum of all the bits y_j other than y_i from the parity check $\mathbf{v}$ involving y_i. The absolute value of this correlation coefficient is a measure of information about y_i contained in the considered parity check $\mathbf{v}$. Accordingly, low-weight parity checks are more informative than the others.

The most effective way is to use (14) iteratively, in each iteration improving the conditioned correlation coefficients $\hat{c}_i$. The iterations are useful because (14) is only an approximate expression and because hard decisions based on conditioned bit probabilities generally do not result in codewords. Instead of directly recycling (14), by substituting $\hat{c}_i$ from the current iteration for c_i in the next iteration (e.g., see [16]), one can also use a more sophisticated and more effective belief propagation recycling [6] (e.g., see [15], [5], and [11]). According to both experimental and theoretical [11] arguments, the correlation coefficients will converge in a relatively small number of iterations to values ± 1 for most coordinates, and, in the case of success, the final hard decisions on individual bits will result in a binary word at a small Hamming distance from a codeword. A simple, information set decoding technique will then yield this codeword along with the corresponding information word $\mathbf{x}$, which is the desired solution to the approximate linear system under consideration.

It is important to point out the conditions for success for both the approaches described above. In accordance with the capacity argument, the decision error probability of the block-based approach using (13) can be made arbitrarily close to zero if

$$\sum_{i=1}^{n} c_i^2 \; \geq \; k. \tag{16}$$

This means that we can reliably distinguish a correct solution from the remaining $2^k - 1$ incorrect solutions if (16) is satisfied. On the other hand, provided that $|c_i| \; = \; c$, $1 \; \leq \; i \; \leq \; n$, it is theoretically argued in [11] that the average bit-decision error probability of the iterative approach using (14) will be close to zero if $\sum_w M_w c^{w-1} > 1$, where M_w is the average number per bit of the parity checks of weight w that are used in (14). In the limit, when c tends to zero, this condition coincides with the similar condition from [18] corresponding to (15) which is both theoretically derived and experimentally verified (see also [2]). Anyway, both conditions are also supported by numerous experimental results on fast correlation attacks on regularly clocked LFSR's (e.g., see [10]).

We will work with a stronger condition, resulting in higher complexity estimates,

$$\sum_w M_w \, c^w \;\geq\; 1 \tag{17}$$

where the exponent $w-1$ is conservatively replaced by w. The new condition can be given another interpretation, directly in terms of (14) or (15). Namely, if we assume that M_w is exactly the number of parity checks of weight w that are used for the i-th bit, then (14) reduces to

$$\hat{c}_i \;=\; /c_i + \sum_w (m_w^+ - m_w^-) c^w / \tag{18}$$

where m_w^+ and m_w^- denote the numbers of parity checks of weight w with the positive and negative sign of the product, respectively. Now, if $y_i = 0$ or $y_i = 1$, then the expected value of $m_w^+ - m_w^-$ is equal to $+M_w c^w$ or $-M_w c^w$, respectively. So, the contribution of the parity checks of weight w can be regarded (statistically) significant for the iterative process to converge to the most likely (correct) values of y_i for each $1 \leq i \leq n$ if $M_w c^w \geq 1$. Accordingly, by combining the contributions of parity checks of different weights we get the condition (17). This condition demonstrates the significant advantage of iterative over one-step algorithms which when applied to individual bits, in light of (16), will be successful if $\sum_w M_w c^{2w} \geq 1$.

4.2 Application to Bluetooth

The main approach to be pursued here is one in which the initial state of the shortest LFSR, LFSR_1, is guessed, so that linear correlations conditioned on both the output and one input can be utilized. This is needed in order to minimize the precomputation complexity to be given below. Let $n = \alpha n_0$ be the number of linear equations chosen out of those resulting from the 4-bit, 5-bit, and 6-bit linear correlations described in Section 3. Here, α is the average number of chosen equations per output bit and $n_0 = 2740$ is the maximum number of output bits that can be used if 6-bit linear correlations are exploited. For each output bit, the average number of α equations are chosen from a possibly larger set of $\beta = \gamma\alpha$ equations that is independent of the observed 6-bit output segment and of the known 6-bit input segment resulting from the guessed initial state of LFSR_1. By expressing each LFSR bit involved as a linear function of the initial state bits, each obtained linear equation becomes a linear equation in the unknown 103=128-25 initial state bits of all the LFSR's but the shortest.

The correlation coefficients associated with these linear equations depend on the known output segment, of maximal length 2745 bits, and on the guessed input segment of the same length. According to Section 3, the linear equations can be grouped in several types corresponding to 4-bit, 5-bit, and 6-bit linear correlations. For each such group, there are several absolute values of the correlation coefficients, each appearing with a given probability. Altogether, let the absolute value μ_j appear with probability ν_j, where $\nu_j = \alpha_j/\alpha$ and α_j is the average number of linear equations with the absolute value of the correlation coefficient equal to μ_j. For each output bit, the equations chosen and the absolute values and the signs of the associated correlation coefficients depend on the known output and guessed input. Two average values of the correlation coefficient magnitudes are important for measuring the success of the linear cryptanalysis to be applied. One, which is related to the iterative bit-based approach and (14) and (17), is the weighted geometric mean $\mu = \prod_j \mu_j^{\nu_j}$. Note that $\mu \leq \bar{\mu} = \sum_j \nu_j \mu_j$. The other, which is related to the block-based approach and (16), is the expected value of the squares $\bar{\mu^2} = \sum_j \nu_j \mu_j^2$.

For the iterative approach, it is necessary to find the linear dependencies among the obtained linear equations (codeword bits) that involve only a relatively small number of linear equations, that is, to determine the corresponding low-weight parity checks. The weights to be used should result in numbers of parity checks that should be sufficient for success according to the condition (17). The number of parity checks of a given weight, w, per codeword bit is a characteristic of the produced linear system (linear code), which depends on the observed output and one guessed input, and can be modeled by assuming that the system is randomly generated as the expected value

$$M_w \;=\; 2^{-103}\binom{n-1}{w} \;\approx\; 2^{-103}\frac{n^w}{w!} \tag{19}$$

where the approximation error is negligible if $w \ll n$.

Consequently, if we utilize all the parity checks of weight at most w, the success condition (17), with the geometric mean of the correlation coefficient magnitudes, becomes

$$\sum_{j=2}^{w} \frac{n^j}{j!}\,\mu^j \;\geq\; 2^{103} \tag{20}$$

(parity checks of weight 1 are impossible for the problem considered). As the term with the maximal weight, w, is dominant, we finally get the condition

$$w\,(\log_2 n_0 + \log_2 \alpha - \log_2 \frac{1}{\mu}) \;\geq\; 103 + \log_2 w! \tag{21}$$

which can be solved numerically to give the minimal required weight w. *This condition is conservative because we neglected the contribution of terms with weight lower than w and because M_w is expected to be larger than* (19) *due to the specific structure of the obtained linear equations for the Bluetooth keystream generator* *(i.e., each parity check gives rise to more parity checks through appropriate phase shifts).*

As the iterative algorithm has to be run for each of 2^{25} guesses about the LFSR_1 initial state, its complexity can be expressed as

$$C = 2^{25} \cdot n \cdot \frac{1}{\mu^w} = 2^{25+\log_2 n_0+\log_2 \alpha+w\log_2 \frac{1}{\mu}} \tag{22}$$

where a computational step consists of all the computations per bit for a number of iterations, which on average is not greater than about 10. In each iteration, the computations are predominantly determined by the number of real multiplications needed to compute (14) for every bit and for parity checks of weight w. This number is given as $3n/\mu^w$, in view of a simple fact that only $3(w-1)$ real multiplications are needed to compute all $w+1$ products of w elements out of a set of $w+1$ elements. Accordingly, a computational step approximately consists of at most 30 real multiplications, where an 8-bit precision will suffice. As a real product of two 8-bit words can be performed by an average of 3 real additions, each requiring about $8 \cdot 3 = 24$ binary operations, the step consists of about 2160 binary operations. This is comparable with one step of the exhaustive search method which consists of about $128 \cdot 15 = 1920$ binary operations.

The next point to be explained is how to generate all the parity checks of weight at most w for a possibly larger set of γn linear equations. This can be done in precomputation time, by computing and sorting all the linear combinations of $\lceil (w+1)/2 \rceil$ linear equations, altogether about $(\gamma n)^{\lceil (w+1)/2 \rceil}/\lceil (w+1)/2 \rceil!$ of them. The matches obtained by sorting directly give all the linear combinations of at most $2\lceil (w+1)/2 \rceil$ linear equations that evaluate to zero identically (e.g., see [8]). More precisely, we have to sort out only

$$\begin{aligned} D &\approx \frac{\gamma^{\lceil (w+1)/2 \rceil}}{\lceil (w+1)/2 \rceil!} \left(2^{103/w} (w!)^{1/w} \frac{1}{\mu} \right)^{\lceil (w+1)/2 \rceil} \\ &\approx \frac{(w!)^{\lceil (w+1)/2 \rceil / w}}{\lceil (w+1)/2 \rceil!} 2^{103\lceil (w+1)/2 \rceil / w + \lceil (w+1)/2 \rceil (\log_2 \gamma + \log_2 \frac{1}{\mu})} \end{aligned} \tag{23}$$

randomly chosen linear combinations, represented as 103-bit words. The total obtained number of matches per bit, i.e., the total number of parity checks per each equation is then γ^w/μ^w. They are all stored as the final result of precomputation.

Now, given an output segment and each guessed input, we have to filter $1/\mu^w$ parity checks out of a set of γ^w/μ^w collected parity checks, for each of n linear equations. If $\gamma > 1$, then γ^w/μ^w parity checks can be sorted out, with respect to n_0 bit positions and β indexes of linear equations per each bit position, so that the filtering takes only about $1/\mu^w$ steps. The complexity of filtering is then given by (22), but the corresponding step complexity is negligible in comparison with one of the iterative algorithm.

After the iterative algorithm has converged to probabilities close to 0 or 1, if the guess about the LFSR_1 initial state was correct, then the 103 bits of the remaining LFSR initial states, along with the initial 4 memory bits, can be reconstructed by information set decoding (e.g., by looking for error-free sets of 103 linearly independent equations), with complexity much smaller than (22).

5 Linear Cryptanalysis of Initialization Scheme

The objective of the linear cryptanalysis of the Bluetooth initialization scheme is to reconstruct 128 bits of the secret key from a given number of 128-bit (or 132-bit) outputs of the Bluetooth initialization scheme obtained from the same secret key and different IV's. Such outputs can be obtained by the linear iterative cryptanalysis method described in Section 4. As the initialization scheme is essentially the same as the keystream generator, for each IV we will again use the linear correlations described in Section 3 to produce another approximate system of linear equations. Other approaches, possibly requiring a smaller number of IV's, may also exist (e.g., see [4]).

The main point facilitating the linear cryptanalysis is that the secret key and IV are linearly combined together to form the initial state of the Bluetooth keystream generator used for initialization. Therefore, each equation linear in LFSR bits can be expressed as the binary sum of an equation linear in secret key bits and an equation linear in IV bits which itself can be evaluated as IV is known. If the same linear equation in LFSR bits is used with different, say q, IV's, one thus effectively obtains q independent observations of the same linear function, say y, of secret key bits. If the correlation coefficient associated with the i-th equation is c_i and if s_i is the value of the linear function of the corresponding IV_i, then the correlation coefficient associated with the i-th observation is $(-1)^{s_i}c_i$, $1 \leq i \leq q$. In view of (15), the combined correlation coefficient, $\hat{c}$, of y is then determined by

$$\frac{1-\hat{c}}{1+\hat{c}} \;=\; \prod_{i=1}^{q}\left(\frac{1-c_i}{1+c_i}\right)^{(-1)^{s_i}} \tag{24}$$

or, approximately, for small c_i, by

$$\hat{c} \;=\; /\sum_{i=1}^{q}(-1)^{s_i}c_i/. \tag{25}$$

Assume that $|c_i| = c$, $1 \leq i \leq q$. Then, if $y = 0$ or $y = 1$, the expected value of $\hat{c}$ is equal to $+qc^2$ or $-qc^2$, respectively. So, the combined correlation coefficient will be close to ± 1 if $q \geq 1/c^2$. In general, in view of (16), it will be close to ± 1 if

$$\sum_{i=1}^{q} c_i^2 \;\geq\; 1. \tag{26}$$

This condition determines the minimal q required for reconstructing the correct value of y with a small probability of decision error.

To minimize the required number of IV's, we will again use linear correlations conditioned on the output and one input, which has to be guessed. Assuming that the sizes of the secret subkeys controlling indivudual LFSR's are the same as their respective lenghts, we have to guess 25 secret key bits controlling LFSR_1. Let an average number of α out of a set of $\beta = \gamma\alpha$ linear equations be chosen for each of $n_0 = 123$ available output bits, provided that 6-bit linear correlations are exploited. For q IV's, each of the resulting βn_0 linear functions of the remaining 103 secret key bits is then treated in the way explained above. Note that each of the functions will on average appear q/γ instead of q times. Then the condition (26) reduces to

$$q \;\geq\; \gamma \, \frac{1}{\bar{\mu^2}} \tag{27}$$

where $\bar{\mu^2} = \sum_j \nu_j \mu_j^2$ is the mean square value of the used correlation coefficients μ_j appearing with probabilities ν_j. The linear correlations to be used should be chosen so as to minimize q. The resulting βn_0 linear equations in 103 secret key bits which hold with probabilities close to 1 can then be solved by information set decoding if the guess about the 25 secret key bits is correct. As the complexity of reconstructing the secret key from given q outputs of the initialization scheme is much smaller than the complexity of reconstructing these outputs, the total complexity is determined by the latter, and is q times larger than (22).

6 Optimal Complexities

There are many possible choices of the linear correlations described in Section 3 to be used in the linear iterative cryptanalysis of the keystream generator in order to reconstruct the LFSR initial states. The objective is to minimize the computation complexity C given by (22) and the precomputation complexity D given by (23). However, this is not possible to achieve simultaneously, as there is a tradeoff between the two criteria.

In general, C is minimal if one uses the linear correlations conditioned on the output, described in Section 3.2, but D is then relatively large. In this case, we do not have to guess one input and the complexity analysis is the same as in Section 4.2 except that the number of LFSR initial state bits to be reconstructed is now 128 instead of 103. For example, if we choose to use the largest 4 6-bit correlation coefficients and the corresponding 8 input linear functions, from the condition (21) get $w = 15$ and hence $C \approx 2^{60}$ and $D \approx 2^{98.5}$. By guessing one input we generally decrease D and increase C. Two illustrative examples are explained in more detail below.

First, choose the 2 largest 5-bit and the 3 largest 6-bit conditioned correlation coefficients to work with. In this case, we get $\alpha = (1+2)+(1+1/2+9/8) = 5.625$, $\beta = 8+8 = 16$, $\gamma = 16/5.625 \approx 2.8444$, and

$$\mu \;=\; \left(\left(\frac{9}{32}\right)^2 \left(\frac{13}{64}\right)^{1/2} \left(\frac{3}{16}\right)^{25/8}\right)^{1/5.625} \;\approx\; 0.2181. \tag{28}$$

Then (21) yields $w = 11$ and hence we get $C \approx 2^{63.07}$ and $D \approx 2^{82.68}$.

Second, choose the largest 2, 3, and 7 conditioned correlation coefficients from 4-bit, 5-bit, and 6-bit linear correlations to work with, respectively. In this case, we get $\alpha = (2+16)+(1+2+1)+(1+1/2+9/8+5/8+1/8+5/8+1/8) = 26.125$, $\beta = (8+32)+8+8 = 56$, $\gamma = 56/26.125 \approx 2.1435$, and

$$\mu = \left(\left(\frac{9}{32}\right)^2\left(\frac{1}{4}\right)^2\left(\frac{13}{64}\right)^{1/2}\left(\frac{3}{16}\right)^{25/8}\left(\frac{11}{64}\right)^{5/8}\left(\frac{5}{32}\right)^{1/8}\left(\frac{9}{64}\right)^{5/8}\left(\frac{17}{128}\right)^{1/8}\left(\frac{1}{8}\right)^{17}\right)^{1/26.125}$$
$$\approx 0.1504. \tag{29}$$

Then (21) yields $w = 9$ and hence we get $C \approx 2^{65.73}$ and $D \approx 2^{79.74}$.

There are also different possible choices of the linear correlations to be used in the linear cryptanalysis of the initialization scheme in order to reconstruct the secret key. The objective is to minimize the required number q of IV's given by (27). To this end, the 25 secret key bits controlling the shortest LFSR are guessed. It is slightly better to work with 6-bit than 5-bit linear correlations conditioned on the output and one input. If we use the 7 largest 6-bit conditioned correlation coefficients, we get $\gamma = 8/4.125 \approx 1.9394$ and

$$\bar{\mu^2} = \frac{1}{4.125}\left(\left(\frac{9}{32}\right)^2 + \frac{1}{2}\left(\frac{13}{64}\right)^2 + \frac{9}{8}\left(\frac{3}{16}\right)^2 + \frac{5}{8}\left(\frac{11}{64}\right)^2 + \frac{1}{8}\left(\frac{5}{32}\right)^2 + \frac{5}{8}\left(\frac{9}{64}\right)^2 + \frac{1}{8}\left(\frac{17}{128}\right)^2\right)$$
$$\approx 0.04251. \tag{30}$$

Then (27) yields $q \approx 45.2 \approx 2^{5.51}$, so that the total complexity of the secret key reconstruction increases to $C \approx 2^{68.58}$ and $C \approx 2^{71.24}$ for the two cases described above, respectively.

7 Conclusions

The developed linear cryptanalysis method shows that correlation attacks may also be applicable to stream ciphers producing very short keystream sequences which are reinitialized frequently by using a cryptographically strong initialization scheme. The complexity analysis concentrates on mathematical rather than practical implementation arguments. The obtained attack complexities for the Bluetooth stream cipher are overestimated as they are based on a conservative assumption about the underlying parity-check weight distribution. It is in principle possible that the complexity can be further decreased by exploiting m-bit linear correlations for $m > 6$ if they are feasible to compute. It may also be possible that the actual precomputation complexity is lower than predicted.

Consequently, at least from the theoretical standpoint, there is a need to redesign the Bluetooth stream cipher, maybe by using the improvement suggested in [14]. This modified Bluetooth stream cipher appears to be more resistant to the linear cryptanalysis, but might not be the optimal choice (see the Appendix).

Appendix

A Linear Correlations in Modified Bluetooth Combiner

We exhaustively computed all the m-bit linear correlations for $m \leq 6$ in the modified Bluetooth combiner proposed in [14]. The distributions of the largest correlation coefficients are determined and displayed here.

The only modification relates to the linear update functions for the 4 memory bits. Namely, instead of (2), we now have

$$c_{t+1}^0 \;=\; s_{t+1}^0 \oplus c_t^0 \oplus c_{t-1}^0, \qquad c_{t+1}^1 \;=\; s_{t+1}^1 \oplus c_t^1 \oplus c_{t-1}^1. \tag{31}$$

The stationary distribution of the 4 memory bits remains to be uniform.

First of all, there are no nonzero correlation coefficients for $m \leq 4$. The largest absolute values of the correlation coefficients, $|c|$, and the (average) numbers, α, of linear functions attaining them are shown in the following tables. In the last table we also show the total numbers, β, of linear functions out of which the desired α are chosen. In this respect, note that for each m, each smaller set is contained in the next larger. For $m = 5$, there are no other nonzero correlation coefficients. For $m = 6$, there are also 46080 and 57600 pairs of input/output linear functions with $|c| = 1/512$ and $|c| = 1/1024$, respectively, whereas $\alpha = 65024$ for $1/1024 \leq |c| \leq 13/1024$, conditioned on output, and $\alpha = 1260$ for $1/64 \leq |c| \leq 3/64$, conditioned on output and one input, and there are no other nonzero correlation coefficients. A general conclusion is that the absolute values of the correlation coefficients are smaller than in the Bluetooth combiner, but their numbers are considerably larger.

Unconditioned Linear Correlations

m	5			6			
$\|c\|$	$\frac{5}{128}$	$\frac{1}{64}$	$\frac{1}{128}$	$\frac{25}{1024}$	$\frac{5}{512}$	$\frac{5}{1024}$	$\frac{1}{256}$
α	256	1536	3840	256	3072	7680	9216

Linear Correlations Conditioned on Output

m	5			6			
$\|c\|$	$\frac{7}{128}$	$\frac{3}{128}$	$\frac{1}{128}$	$\frac{33}{1024}$	$\frac{25}{1024}$	$\frac{21}{1024}$	$\frac{15}{1024}$
α	128	768	3200	64	128	256	64

Linear Correlations Conditioned on Output and One Input

m	5			6		
$\|c\|$	$\frac{3}{16}$	$\frac{1}{8}$	$\frac{1}{16}$	$\frac{9}{64}$	$\frac{3}{32}$	$\frac{1}{16}$
α	8	16	120	4	16	16
β	64	128	512	64	128	128

References

1. Bluetooth$^{\text{TM}}$, *Bluetooth Specification*, Version 1.1, Feb. 2001.
2. V. Chepyzhov and B. Smeets, "On a fast correlation attack on stream ciphers," Advances in Cryptology - EUROCRYPT '91, *Lecture Notes in Computer Science*, vol. 547, pp. 176-185, 1991.
3. E. Dawson and A. Clark, "Divide and conquer attacks on certain classes of stream ciphers," *Cryptologia*, vol. 18, pp. 25-40, 1994.
4. S. Fluhrer and S. Lucks, "Analysis of the E_0 encryption system," Selected Areas in Cryptography - SAC 2001, *Lecture Notes in Computer Science*, vol. 2259, pp. 38-48, 2001.
5. M. P. C. Fossorier, M. J. Mihaljević, and H. Imai, "Reduced complexity iterative decoding of low-density parity check codes based on belief propagation," *IEEE Trans. Commun.*, vol. 47, pp. 673-680, May 1999.
6. R. G. Gallager, "Low-density parity-check codes," *IRE Trans. Inform. Theory*, vol. 8, pp. 21-28, Jan. 1962.
7. J. Dj. Golić, "Correlation properties of a general binary combiner with memory," *Journal of Cryptology*, vol. 9, pp. 111-126, 1996.
8. J. Dj. Golić, "Computation of low-weight parity-check polynomials," *Electronics Letters*, vol. 32, pp. 1981-1982, Oct. 1996.
9. J. Dj. Golić, "Cryptanalysis of alleged A5 stream cipher," Advances in Cryptology - EUROCRYPT '97, *Lecture Notes in Computer Science*, vol. 1233, pp. 239-255, 1997.
10. J. Dj. Golić, M. Salmasizadeh, and E. Dawson, "Fast correlation attacks on the summation generator," *Journal of Cryptology*, vol. 13, pp. 245-262, 2000.
11. J. Dj. Golić, "Iterative optimum symbol-by-symbol decoding and fast correlation attacks," *IEEE Trans. Inform. Theory*, vol. 47, pp. 3040-3049, 2001.
12. M. Jakobsson and S. Wetzel, "Security weaknesses in Bluetooth," Topics in Cryptology - CT-RSA 2001, *Lecture Notes in Computer Science*, vol. 2020, pp. 176-191, 2001.
13. C. R. P. Hartmann and L. D. Rudolph, "An optimum symbol-by-symbol decoding rule for linear codes," *IEEE Trans. Inform. Theory*, vol. 22, pp. 514-517, Sept. 1976.
14. M. Hermelin and K. Nyberg, "Correlation properties of the Bluetooth combiner," Information Security and Cryptology - ICISC '99, *Lecture Notes in Computer Science*, vol. 1787, pp. 17-29, 1999.
15. D. J. C. MacKay, "Good error-correcting codes based on very sparse matrices," *IEEE Trans. Inform. Theory*, vol. 45, pp. 399-431, Mar. 1999.
16. W. Meier and O. Staffelbach, "Fast correlation attacks on certain stream ciphers," *Journal of Cryptology*, vol. 1, pp. 159-176, 1989.
17. W. Meier and O. Staffelbach, "Correlation properties of combiners with memory in stream ciphers," *Journal of Cryptology*, vol. 5, pp. 67-86, 1992.
18. M. J. Mihaljević and J. Dj. Golić, "A method for convergence analysis of iterative probabilistic decoding," *IEEE Trans. Inform. Theory*, vol. 46, pp. 2206-2211, Sept. 2000.