Applied Stream Ciphers in Mobile Communications

THÈSE N° 3491 (2006)

PRÉSENTÉE À LA FACULTÉ INFORMATIQUE ET COMMUNICATIONS

Institut de systèmes de communication

SECTION DES SYSTÈMES DE COMMUNICATION

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

POUR L'OBTENTION DU GRADE DE DOCTEUR ÈS SCIENCES

PAR

Yi LU

B.Eng. in Computer Science & Technology, Beijing Polytechnic University
Chine, et de nationalité chinoise

devant le jury

Prof. Serge Vaudenay, directeur de thèse

Dr. Anne Canteaut, rapporteur

Prof. Willi Meier, rapporteur

Prof. Amin Shokrollahi, rapporteur

Lausanne, EPFL
2006

誰言寸草心，
報得三春暉。

Abstract

This dissertation is concerned with cryptanalysis of E0, the stream cipher used in the short-range wireless radio standard Bluetooth, and of its generalization by means of correlation attacks. It consists of three parts.

In the first part, we propose an E0-like combiner with memory as the core stream cipher. First, we formulate a systematic and simple method to compute the correlations. An upper bound of the correlations is given. Second, we show how to build either a uni-bias-based or multi-bias-based distinguisher to distinguish the keystream produced by the combiner from a truly random sequence, once correlations are found. The data complexity of either distinguisher is analyzed for performance comparison. The keystream distinguisher is then upgraded for use in the key-recovery attack. The latter reduces to the well-known maximum likelihood decoding problem given the keystream long enough.

In the second part, the core stream cipher is transformed into the dedicated stream cipher by attaching the one-level or two-level initialization scheme. We show that the correlation attack on the core stream cipher leads to the correlation attack on the dedicated stream cipher with the one-level initialization scheme (with equal bias), but not necessarily so with the two-level initialization scheme.

In the last part, we generalize the existing concept of conditional correlations and study conditional correlation attacks against stream ciphers and other cryptosystems. A general framework is developed for smart distinguishers, which exploit those generalized conditional correlations. Based on the theory of the traditional distinguisher, we derive the number of samples necessary for a smart distinguisher to succeed. It allows to prove that the smart distinguisher improves on the traditional basic distinguisher.

As an application of all our analysis, it leads to the fastest (and only) practical known-plaintext attack on Bluetooth encryption so far. Our attack recovers the encryption key using the first 24 bits of $2^{23.8}$ frames and with 2^{38} computations.

Keywords: cryptanalysis, stream cipher, E0, correlation

Résumé

Cette thèse traite des cryptanalyses, utilisant des attaques par corrélation, de E0 (le chiffrement à flots utilisé dans le standard Bluetooth de communication sans-fil) et d'une généralisation de E0. Elle est composée de trois parties.

Dans la première partie, nous proposons une construction centrée sur un chiffrement à flots utilisant une fonction de combinaison à mémoire similaire à celle de E0. D'abord nous présentons une méthode simple et systématique pour calculer les corrélations. Nous donnons ainsi une borne supérieure aux corrélations. Ensuite, nous montrons comment, une fois des corrélations trouvées, construire deux distingueurs utilisant soit un seul, soit plusieurs de ces biais afin de distinguer une suite chiffrante d'une suite purement aléatoire. Afin de comparer les performances de ces deux distingueurs, nous analysons leurs quantités de données nécessaires. Ce distingueur de suite chiffrante est ensuite modifié pour être utilisé dans une attaque permettant de retrouver la clef. Cette attaque se ramène au problème bien connu de décodage à maximum de vraisemblance, étant donné une suite chiffrante assez longue.

Dans la deuxième partie, nous considérons le chiffrement à flot complet en ajoutant au chiffrement central précédent l'un des schémas d'initialisation à un ou à deux niveaux. Nous montrons que l'attaque par corrélation sur le chiffrement à flot central mène à une attaque sur le chiffrement complet utilisant un schéma d'initialisation à un niveau (avec un biais identique) mais pas nécessairement avec le schéma d'initialisation à deux niveaux.

Dans la dernière partie, nous généralisons le concept connu de corrélations conditionnelles et étudions des attaques par corrélation conditionnelles sur des chiffrements à flots et d'autres cryptosystèmes. Un cadre général est développé pour des distingueurs améliorés, tirant profit de ces corrélations conditionnelles généralisées. En se reposant sur la théorie des distingueurs traditionnels, nous déduisons le nombre d'échantillons nécessaires au succès d'un distingueur intelligent. Cela permet de prouver que les distingueurs améliorés font mieux que les

distingueurs basics traditionnels.

En appliquant toutes nos analyses, nous obtenons la plus rapide (mais aussi la seule réalisable à ce jour) attaque à clair connu applicable en pratique au chiffrement Bluetooth. Notre attaque permet de retrouver la clef de chiffrement en utilisant les 24 premiers bits de $2^{23.8}$ paquets et en 2^{38} calculs.

Mots-clés: cryptanalyse, chiffrement à flots, E0, corrélation

Acknowledgement

First of all, I would like to thank Prof. Serge Vaudenay for offering me a precious chance to start Ph.D. in the lab LASEC and for his hand-by-hand teaching, his special patience and constant encouragement in my work and above all for his model of doing scientific research with strict attitude, passion and perseverance.

It is a great honor for me to have Dr. Anne Canteaut, Prof. Willi Meier, Prof. Amin Shokrollahi and the doctoral program director Prof. Emre Telatar in the jury. I sincerely thank them for their investing time and energy reading my dissertation and proposing valuable advices. My research experience and ability expands a lot from wonderful collaboration with Prof. Willi Meier. Besides two coauthors of my papers—Serge Vaudenay and Willi Meier, it is also a golden opportunity for me to learn lots of valuable knowledge and receive patient teaching from those researchers (in time order): Andrea Ridolfi, Orr Dunkelman, Anne Canteaut, Antoine Joux.

I also deeply thank all LASEC members for their long-term generous support throughout my Ph.D. studies: Pascal Junod, Gildas Avoine, Philippe Oechslin, Brice Canvel, Jean Monnerat, Thomas Baignères, Matthieu Finiasz, Martin Vuagnoux, Sylvain Pasini, and our associate members Simon Künzli, Claude Barral, Julien Brouchier. Of course, life abroad would be unutterably tough otherwise without presence of LASEC secretary Martine Corval, as well as France Faille from the neighboring lab LSR to have smalltalks and offer advices of all kinds and settle my daily-life troubles. I would not ever forget my friends Sibi Raj Bhaskaran Pillai, Nadja Subotic, Ivana Arsic, Hien-Dat Tran, Luiz Angelo Barchet-Estefanel, Stevan Ignjatovic, Shai Tirosh, Arnas Kupsys, Pawel Wojciechowski, David Cavin, Sonja Buchegger, Rini Nur Hasanah, Hassina Bounif, Tina Amper, Marc Kelliny, Carine Grannavel, Chunmei Zhang, Ronel Tolo, Joseph Awolebo, Wanjun Mi, Hongze Lu, Yih-teen Lee, Shijun Yu, Ruohua Zhou, Beilu Shao, Hai Zhan, Jie Wu, Ge Zhuang, Jian Yang, Qing Li, Ying Liu, Li Chen, Jiyong Zhang, Haoming Wang, Jun Luo, Changyan Di, Hong Shu, Baohong Liu, Yanzhou Zhou, Kewei

Zhu, Ye Zhang, Yu Lei, Huan Du, Yiqing Guan, Fengxiang Jin, Zhaosong Qu, Qing Zhu, Rui Guo, Zhan Liang.

Last but not least, without a root—my family, I am no longer myself.

Contents

Chapter 1

Thesis Outline

In order to protect confidentiality, the encryption scheme is used to transform the plaintext message into the ciphertext by the key. Depending on whether encryption and decryption use the same key or not, an encryption scheme can be either symmetric or asymmetric. Block ciphers and stream ciphers are two families of symmetric encryption schemes. The former tends to encrypt groups of characters of the plaintext message simultaneously, while the latter encrypts individual character of the plaintext message one at a time. Stream ciphers are inherently suitable for the time-critical applications or processing-constrained devices to meet requirements of performance extremes (e.g. speed, area, power supply and power consumption). For this reason, they are especially suitable for wireless encryptions. One notable example is the stream cipher E0 used in the short-range wireless radio standard Bluetooth. As the conclusion of my Ph.D. studies at EPFL, this dissertation is concerned with cryptanalysis of E0 and its generalization. Here is the contents in brief for each upcoming chapter:

In Chapter 2, we introduce the background of stream ciphers. Starting from the classic stream ciphers, we introduce their classification with focus on one of the oldest and most popular classes, namely, the LFSR-based stream ciphers where LFSR refers to Linear Feedback Shift Register. Then, we discuss generic attacks on classic stream ciphers: time-memory tradeoff, guess and determine, algebraic attack and correlation attack. Accordingly, we review on the basic design principles for the LFSR-based stream ciphers. Next, we take a closer look at enumerative examples of those stream ciphers in the real world. For clarity, they are called dedicated stream ciphers to be distinguished from the classic (textbook) stream ciphers. Finally, we discuss existing attacks on the individual dedicated stream cipher and the current immature generic attacks against the

dedicated stream ciphers.

In Chapter 3, we propose an E0-like combiner with memory as the core stream cipher in the dedicated stream ciphers. First, we formulate a systematic and simple method to compute correlations. An upper bound of the correlations is given. Second, we show how to build either a uni-bias-based or multi-bias-based distinguisher to distinguish the keystream produced by the combiner from a truly random sequence, once correlations are found. The data complexity of either distinguisher is carefully analyzed for performance comparison. The keystream distinguisher is then upgraded for use in the key-recovery attack. The latter actually reduces to the well-known Maximum Likelihood Decoding (MLD) problem given the keystream long enough. We devise a general algorithm to solve the MLD problem for any linear code. The analysis is demonstrated to attack the core of E0, which results in the best known key-recovery attack.

By attaching the one-level or two-level initialization scheme, the core stream cipher is transformed into the dedicated stream cipher in Chapter 4. For the cryptanalysis of the dedicated stream ciphers, we concentrate on the correlation attacks. Our results show that the correlation attack on the core stream cipher leads directly to the correlation attack on the dedicated stream cipher with the one-level initialization scheme (with equal bias), but not necessarily so with the two-level initialization scheme. In the continued case study, we apply the analysis to the attack on E0 with one-level and two-level initialization scheme respectively (which we call one-level and two-level E0 in short respectively). The correlation attack on two-level E0 is feasible due to a resynchronization flaw, which we detect for the first time and allows to deduce the correlations of two-level E0 from those of the core of E0.

In Chapter 5, we generalize the existing concept of conditional correlations in literature and study conditional correlation attacks against stream ciphers and other cryptosystems, in case the computation of the output allows for side information related to correlations conditioned on the input. A general framework is developed for smart distinguishers, which exploit those generalized conditional correlations. Based on the theory of the traditional distinguisher, we derive the number of samples necessary for a smart distinguisher to succeed. It allows to prove that the generalized conditional correlation is no smaller than the unconditional correlation. In other words, the smart distinguisher improves on the traditional basic distinguisher. Finally, as an application of our generalized conditional correlations, a conditional correlation attack on the two-level E0 is developed and optimized. This is the fastest (and only) practical known-plaintext attack on Bluetooth encryption so far. Our best attack fully recovers the original

encryption key using the first 24 bits of $2^{23.8}$ frames and with 2^{38} computations.

In Chapter 6, we give conclusions and discuss open questions in this area.

Chapter 3, Chapter 4 and Chapter 5 are the original contribution of this thesis. They come from the extention of my published papers respectively:

- ⋆ Yi Lu, Serge Vaudenay, *Faster Correlation Attack on Bluetooth Keystream Generator E0*, Advances in Cryptology - CRYPTO 2004, Lecture Notes in Computer Science, vol.3152, M. Franklin Ed., Springer-Verlag, pp. 407-425, 2004

- ⋆ Yi Lu, Serge Vaudenay, *Cryptanalysis of Bluetooth Keystream Generator Two-level E0*, Advances in Cryptology - ASIACRYPT 2004, Lecture Notes in Computer Science, vol.3329, P. J. Lee Ed., Springer-Verlag, pp. 483-499, 2004

- ⋆ Yi Lu, Willi Meier, Serge Vaudenay, *The Conditional Correlation Attack: A Practical Attack on Bluetooth Encryption*, Advances in Cryptology - CRYPTO 2005, Lecture Notes in Computer Science, vol.3621, V. Shoup Ed., Springer-Verlag, pp. 97-117, 2005

Chapter 2

Introduction

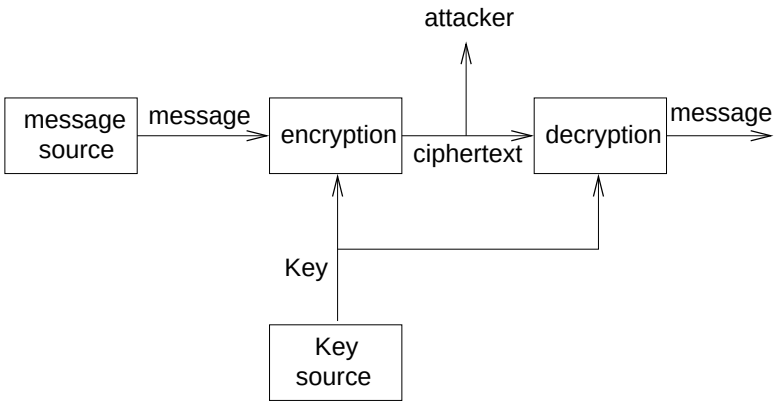

Figure 2.1: Symmetric-key secrecy system

Due to Shannon [106], a secrecy system[1] between a transmitter and a receiver can be best illustrated by Fig. 2.1. At the transmitting end, the ciphertext is produced by encrypting the message using the key. Upon reception of the ciphertext, it is decrypted using the same key to obtain the original message. As the channel between the transmitter and the receiver is insecure, the ciphertext is subject to falling on the hand of the attacker. The attacker's task is to reconstruct the message from the ciphertext. In order to describe the ideal cryptosystem where knowledge of the ciphertext leaks no information about the message itself to the

[1] This is actually a symmetric-key cryptosystem, as the encryption and decryption use the same key. Its counterpart—public-key cryptosystem was introduced in 1979 [36].

attacker, Shannon first introduced the notion of perfect secrecy. Mathematically speaking, let X, Y be the plaintext and ciphertext respectively. Perfect secrecy means for all X, Y we always have

$$\Pr(Y|X) = \Pr(Y).$$

A good example of the cipher that achieves perfect secrecy is the Vernam cipher [112] (a.k.a. one-time pad). In the Vernam cipher, the key length is equal to the message length and the key is chosen randomly with uniform distribution. The encryption is just bitwise XOR of the message and the key. And the XOR of the ciphertext and the key yields the original message.

Though one-time pad achieves perfect secrecy, it is impractical since the key length needs to be equally long as the message and has to be refreshed for every message. Hence the development of cryptography. Generally speaking, there exist two encryption schemes: block ciphers and stream ciphers. Stream ciphers encrypt individual character (usually bit) of a plaintext message one at a time[2]. By contrast, block ciphers tend to simultaneously encrypt groups (usually 64 or 128 bits) of characters of the plaintext message.

Main characteristics of stream ciphers can be summarized as the following:

- speed: faster in hardware,
- hardware implementation cost: low,
- error propagation: limited or no error propagation,
- synchronization requirement: to allow for proper decryption, the sender and receiver must be synchronized (i.e. using the same key and operating at the same position within the key). As detailed in Section 2.1.1, stream ciphers are commonly classified as synchronous stream ciphers and self-synchronizing stream ciphers according to their capability of re-establishing proper decryption automatically after loss of synchronization.

Moreover, though fruitful in theoretical development, the unfavorable situation to stream ciphers is that in the open literature, relatively few fully-specified algorithms are available, let alone to be standardized.

[2]Note that the aforementioned Vernam Cipher belongs to stream cipher.

2.1 Classic Stream Ciphers

2.1.1 Classification

Let K, x_i, y_i, σ_i denote the key, the plaintext digit, the ciphertext digit, and the internal state at time i respectively. A general stream cipher [98] is described by the equations

$$\begin{aligned} \sigma_{i+1} &= F(\sigma_i, x_i, K), \\ y_i &= f(\sigma_i, x_i, K), \end{aligned}$$

where F is the next-state function and f is the output function. Typically, f is defined as

$$y_i = x_i \oplus g(\sigma_i, K).$$

The sequence $\{z_i = g(\sigma_i, K)\}$ is referred to as the keystream. And the algorithm to produce $\{z_t\}$ is called the keystream generator (a.k.a. running-key generator or pseudorandom sequence generator). To decrypt, we just compute

$$x_i = y_i \oplus g(\sigma_i, K).$$

Depending on the definition of F, stream ciphers are popularly classified as synchronous or self-synchronizing.

Synchronous Stream Ciphers

A synchronous stream cipher is one in which the keystream is produced independently of the plaintext (and of the ciphertext), i.e.

$$\sigma_{i+1} = F_1(\sigma_i, K).$$

For example, the OFB mode of a block cipher is a synchronous stream cipher. Two basic properties of synchronous stream ciphers are obvious following the definition: extra synchronization mechanism is mandatory in case of loss of synchronization; it has no error propagation.

Self-synchronizing Stream Ciphers

A self-synchronizing (a.k.a. asynchronous) stream cipher is one in which the keystream is produced as a function of the key and a fixed number of previous ciphertext digits. The typical mode is the cipher feedback mode

$$\sigma_i = F_2(K, y_{i-N}, y_{i-N+1}, \ldots, y_{i-1}),$$

where N is a constant. For example, a block cipher in one-bit cipher feedback mode is an asynchronous stream cipher. Accordingly, two basic properties of asynchronous stream ciphers include: as the name implies, self-synchronization is enabled in case of loss of synchronization; it suffers limited error propagation only.

2.1.2 LFSR-based Stream Ciphers

Linear Feedback Shift Register (LFSR) is perhaps the most popular building block of stream ciphers among other constructions[3] (e.g. cellular automata [83, 115, 116], t-functions [34, 60, 67–69]). In this section, we will give a brief review on construction methods of LFSR-based stream ciphers, and detailed descriptions of the example keystream generators can be found in [98].

Nonlinear Combination Generator

The nonlinear combination generator (or combiner in short) consists of several (regularly-clocked) LFSRs. The keystream is generated as a nonlinear function of the outputs of the component LFSRs, which is called the combining function. Additionally, the combiner may have some memory (i.e. an extra FSM controls the computation of the next state from the current state), and the combining function involves the memory bits. The summation generator [97] belongs to the combiner with memory as well as our core stream cipher defined in Section 3.1, Chapter 3.

Nonlinear Filter Generator

The nonlinear filter generator consists of a single (regularly-clocked) LFSR. The keystream is generated as a nonlinear function of the contents from a fixed subset of the stages of the LFSR, which is called the filtering function.

Clock-controlled Generator

Unlike the nonlinear combiner or filter generator, in this class of the generators, a clocking mechanism defines how each component LFSR is clocked. The

[3]Note that mode of operation of block ciphers can be considered as a keystream generator, such as the output feedback mode (OFB), the cipher feedback mode (CFB) and the counter mode (CTR). And cryptanalysis of such stream ciphers generally involves attacking the underlying block cipher.

keystream is usually generated as the XOR of the LFSR outputs or simply the output of one specified LFSR. Examples include the alternating step generator, the shrinking generator (see [55] for a survey).

2.1.3 Attacks

According to the purpose of the attack, it can be divided into three categories.

- Distinguishing attack: to distinguish the output of the keystream generator from a truly random sequence.
- Predicting attack: to predict the output of the keystream generator with or without a keystream of limited length.
- Key-recovery attack: to obtain the key.

The predicting attack implies a distinguishing attack. The key-recovery attack implies the predicting attack (and hence the distinguishing attack). Obviously, the most powerful attack of all is the key-recovery attack.

Meanwhile, according to the assumptions of the cryptanalyst, the attack can be either a known-plaintext attack or a ciphertext-only attack. The former implies the keystream is known. The latter essentially involves investigation of the redundancy in the plaintext and thus is application-dependent, which is less common in cryptanalysis. Now, we discuss generic attacks on stream ciphers.

Time-memory Tradeoff

The attack is not only effective to stream ciphers, but also applicable to block ciphers. In the landmark paper [58] in 1980, it was shown for the first time that a tradeoff can be achieved between time complexity and memory complexity of attacking a general cryptosystem. The idea was lately adjusted in the case of stream ciphers for a tradeoff [8,16] between time, memory and data complexities. Most recently, the tradeoff attack was further improved in [61, 94].

Guess and Determine

The basic idea is to guess a few part of the key, and use the knowledge of the keystream generator to solve the rest of the key that generates the target keystream. It is especially suitable to attack LFSR-based stream ciphers, where only the states of a few shortest LFSRs are guessed.

Algebraic Attack

The algebraic attack [3, 4, 26–29, 43, 57, 66, 79] is comparatively new in the research literature but has received lots of attention; it is also applicable to block ciphers (e.g. see [30]). In short, when there is a multivariate relation involving only the key and the keystream output, the key can be found by using either the linearization method or XL method to solve the (overdefined) system of multivariate equations. The LFSR-based stream ciphers are potentially vulnerable against this attack and it has been successfully demonstrated that the algebraic attack against a series of stream ciphers is very practical and efficient [3,4,27–29,57,90]. One major drawback of this method, however, is the difficulty in complexity estimate for both time and data complexity, which arises from the tough underlying problem of solving the equations.

Correlation Attack

Vast body of intensive research literature covers this kind of attacks for two decades. Initially targeting at the nonlinear combiners, Siegenthaler first introduced the correlation attacks [108] in the middle of the 1980's. The basic idea is to "divide and conquer" when the keystream output is correlated to the individual LFSR output sequence due to the poor choice of the combining function. That is, instead of the naive exhaustive search on all possible combination of the initial states of the component LFSRs, we only perform an exhaustive search on each individual LFSR independently and test the correlation between each LFSR output sequence and the keystream. The optimum (deterministic) maximum likelihood decoding strategy yields the answer for the initial state of the LFSR. Fig. 2.2 illustrate this idea. Note that by viewing the nonlinear filter generator as the nonlinear combiner with memory, the idea [108] of Siegenthaler's correlation attacks on nonlinear combiners can be applied to attack nonlinear filter generators (e.g. [49, 52, 96, 109]).

Apparently, the time complexity of the basic correlation attack [108] grows exponential in the length of the LFSR, which is impractical for a long LFSR. As a matter of fact, in coding theory, the maximum likelihood decoding problem for linear codes, according to [12], was shown to be NP-complete (see [50] for definition). The focus of cryptographers has been on the general problem where the individual LFSR may be arbitrarily long. In order to speed up the attack for the general setting, Meier and Staffelbach [80, 81] used the probabilistic iterative decoding strategy to refine the basic correlation attack into a so-called "fast

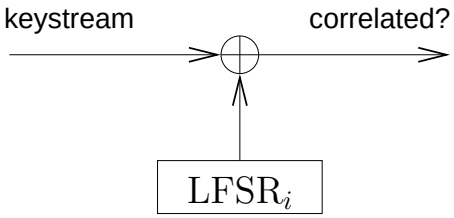

Figure 2.2: Siegenthaler's correlation attack on the nonlinear combiner

correlation attack" to reconstruct each individual LFSR. A critical factor for the efficiency of the fast correlation attack is the novel use of the multiple polynomial of the LFSR's feedback polynomial with low weight (and low degree).

This fast correlation attack of [80, 81] was improved by a series of variant fast correlation attacks (e.g. [22, 25, 88, 89, 95, 118]). Recently, various (still probabilistic) decoding techniques have proved very successful to further improve the performance of the fast correlation attack (e.g. [19,20,23,24,62–64,86,87]).

2.1.4 Basic Design Principles

Generally speaking, the development of the design criteria lags far behind that of attacks. Below we provide a collection of necessary, but by no means sufficient design principles known to date for the LFSR-based keystream generator.

- Pseudorandomness: the output of the keystream generator should not be distinguishable from a truly random sequence; otherwise, the attacker can not only mount a ciphertext-only attack (to decrypt), but also play a more dangerous game to impersonate the encryptor with high probability of success. One earlier attempt to establish some necessary conditions for a pseudorandom sequence to look random is Golomb's randomness postulates [85]. Other statistical tests were also proposed, like Maurer's universal statistical test [85] and FIPS 140-1 statistical tests [45] for randomness. Note that pseudorandomness is the common criterion that all keystream generators should comply with.

- Linear Complexity: the length of the shortest possible equivalent LFSR to generate a given binary sequence of finite length. The notable Berlekamp-Massey algorithm [76] is a very efficient algorithm to determine the linear

complexity of a finite binary sequence of bitlength n within $O(n^2)$ bit operations.

- Nonlinearity: it was first studied in [82] as a security measure of cryptographic Boolean functions. A function with low nonlinearity is prone to the linear attack [77] (or the best affine approximation attack [37]). Note that nonlinearity is also an important parameter for combination generators as shown in [20].

- Correlation Immunity: with the advent of Siegenthaler's proposed correlation attacks [108], Siegenthaler proposed the concept of correlation immunity for a cryptographic Boolean function in [107] to describe the existence of the correlation between the minimum number of input variables and output. High correlation immunity implies that many input variables must be considered jointly in the divide-and-conquer scenario, and thus is expected to increase the complexity of correlation attacks.

- Others: to resist newly-emerging algebraic attacks, the notion of algebraic immunity was first informally put forward in [29], 2003, formalized in [79] and well studied in [33] more recently. Meanwhile, the algebraic degree of a Boolean function should be high.

The problem of how to construct a good Boolean function to achieve the best possible tradeoff among above criteria is difficult, which stimulates lots of research work (e.g. [21, 44, 100, 101, 103, 104, 120]) and has still a long way to go.

2.2 Dedicated Stream Ciphers: Real-world Applications

It is trivial to see that block ciphers are designed directly for practical use in the real-world. However, this is not the case with regards to its (synchronous) stream cipher counterpart. The reason is that in order to avoid keystream reuse, an extra initialization scheme is mandatory to specify how to reinitialize the (core) keystream generator for each encryption. Hence the name of dedicated stream ciphers. For our purposes, we refer those stream ciphers without an initialization scheme as classic stream ciphers, and those with an initialization scheme (i.e. to be used in the real-world) as dedicated stream ciphers respectively.

Stream ciphers are inherently suitable for the time-critical applications or processing-constrained devices to meet requirements of performance extremes (e.g. speed, area, power supply and power consumption). Its major application, though by no means restricted to[4], is voice/video encryption in communications, e.g., VoIP (Voice over IP), digital video broadcasting system like pay-TV. And it is especially suitable for use in the wireless (mobile) communications as detailed next in Section 2.2.1) which demand high speed, minimal area, limited power supply and low power consumption.

Despite of so many wide applications, the history of stream cipher standardisation[5] is fairly short [91] in comparison to its long history of development of more than half a century. In fact, the only available standards covering stream ciphers are those produced for specific applications.

Thanks to the growing attention paid by academia and industry, recently, remarkable standardisation efforts to evaluate cryptographic primitives, including encryption, have initiated worldwide. One example is NESSIE [93] in Europe, which provides recommendations for standardisation bodies like ISO. The other is a Japanese e-government initiative CRYPTREC. Another new project eSTREAM [42], which belongs to ECRYPT [38] (European network of excellence in cryptology), aims to identify new stream ciphers suitable for widespread adoption.

2.2.1 Examples

RC4 in WEP

RC4 was designed by Ron Rivest for RSA company in 1987 and remained proprietary. However, alleged descriptions leaked in 1994 and was widespread since then (e.g. see [102]). RC4 belongs to the keystream generator without use of LFSRs and is designed for fast software implementation (7.3 cycles/byte on PIII according to [15]).

It is standardized for use in IEEE 802.11 wireless networks to protect the data at the link-layer during wireless transmission. The security protocol is called the Wired Equivalent Privacy (WEP). But WEP is flawed, and it leads to a strengthened protocol Wi-Fi Protected Access (WPA) based on RC4 to replace it. Another popular protocol using RC4 is Secure Sockets Layer (SSL) to protect internet traffic. For a survey of attacks on RC4 and WEP see [15,91] for example.

[4] e.g. HDD encryption to protect the hard disk

[5] see [35] for a detailed discussion of stream ciphers and encryption standardisation

A5/1, A5/2, A5/3 in GSM

GSM (Global System for Mobile communications) standard [92] for cell phone communication, which accounts for over one billion subscribers nowadays, uses A5/X series (namely A5/1, A5/2, A5/3) as encryption algorithms to protect over-the-air privacy. A5/1 and A5/2 were designed in the late 80's and kept secret until disclosure of the source code in 90's due to the reverse engineering work.

Both A5/1 and A5/2 are clock-controlled generators based on LFSRs. The weaker A5/2 was immediately broken, which was designed to offer a limited level of security only for reasons of political and export control. And neither does the stronger A5/1 last long. In 2002, an upgraded stream cipher A5/3 was announced and made public [1] for the wide-range evaluation. It is based on a mode of operation of the block cipher KASUMI. So far, no security weakness was reported on A5/3 in spite of research efforts on KASUMI itself. However, due to the flaw in the protocol, problem still exists [11] with GSM encryption even if A5/3 is used. For a detailed account of attacks on GSM ciphers, see [15, 91]. Note that like A5/3, other famous KASUMI-based stream ciphers in the mobile world include the confidentiality algorithm f8 in UMTS/3GPP (Universal Mobile Telecommunications System/3rd Generation Partnership Project) and the encryption algorithm GEA3 [1] for GPRS (General Packet Radio Services).

E0 in Bluetooth

Bluetooth [17] is the new emerging short-range wireless radio standard with low power consumption. It has wide applications such as Bluetooth peripherals (e.g. mice, keyboards, printers), wireless networking, file transfer between cell phones, computers and PDAs etc.

E0 is the stream cipher in Bluetooth. It is an LFSR-based combiner with memory and actually a variant of the summation generator.

Throughout this thesis, E0 serves exclusively as the subject of our case study. Attacks on E0, prior to our work, can be found in [3, 4, 28, 39, 40, 46, 47, 53, 57, 59, 70, 99]. Other interesting security problems concerning either the protocol or implementations can be found in [105, 114].

2.2.2 Attacks

The related-key attack is a major cryptanalysis method to evaluate security of dedicated stream ciphers. As a matter of fact, the term "related-key attack" was first devised by Eli Biham as a new type of attacks on block ciphers [13, 14]. According to Biham, based on some weak key scheduling algorithm, it is possible to use the key relations to mount chosen-key attacks on the block cipher, where only the relations between pairs of related keys are chosen by the attacker and the keys are unknown to him.

In the world of dedicated stream ciphers, the related-key attacks[6] becomes meaningful. This is because the user key has to initialize the keystream generator with different public parameter(s) for each encryption. Hence, the relations between those initial states can be derived from the known initialization scheme and are known to the attacker. He can therefore mount the related-key attack, assuming that he gathers many keystreams produced by the same user key. This scenario, however, does not make sense in cryptanalysis of the classic stream ciphers.

In general, cryptanalysis of classic stream cipher and that of dedicated stream cipher have different settings for the key-recovery attack. The former aims at recovering the key (or the initial state of the keystream generator) given one keystream (without length limit), while the latter aims at recovering the user key given keystreams of limited length[7] all produced by the same user key and different public parameters.

The related-key attacks are very powerful against the dedicated stream ciphers. For instance, the practical (and best) attack[8] [10] on A5/1, the practical attack on RC4 in WEP [48, 75, 110] as well as the practical (and best) attack on E0 in our thesis work all belong to the related-key attacks, which involve a careful elaborate study on the initialization scheme and the underlying core stream cipher.

Obviously, the security of the dedicated stream cipher depends on both the core (underlying) stream cipher and the initialization scheme. Nevertheless, in literature, less results are known on the general design principle of the initialization schemes, partly due to short of available examples of dedicated stream ciphers. In the early 90's, [32] first studied attacks on the linear resynchronization scheme for nonlinear filter generators. Recently, [54] extended [32] to the

[6] sometimes called rekeying attacks or resynchronization attacks

[7] typically very short keystreams due to the frequent resynchronization in practice

[8] it is based on the earlier work of [41, 78]

case where the output function may be unknown but still restricted to the memoryless keystream generator; [6] generalized [32] for combiners with memory and discussed the possibility of combining algebraic attacks and linear cryptanalysis. The work [119] in 2004 contributed to the design of dedicated stream ciphers, which for the first time proposed a strict separation of the core keystream generator and the initialization scheme and introduced the notion of inner state size efficiency as a security measure. As pointed out in [119], lots of open questions remain to be solved concerning the security of initialization schemes.

Chapter 3

Cryptanalysis of the Core Stream Cipher

3.1 Mathematical Model

Most dedicated stream ciphers are based on Linear Feedback Shift Registers[1] (LFSRs), e.g. A5/1, E0. They use such mechanism as the irregular clocking, the combination or filtering nonlinear function to destroy the fatally weak property of LFSRs: linearity. In this thesis, we focus on one of the most popular class of stream ciphers—the LFSR-based combiner (with or without memory) as the core stream cipher. The model is depicted in Fig. 3.1.

To briefly outline, the keystream generator is consisted of n maximum-length LFSRs denoted by $R_1, \ldots, R_n$. Let the R_i have pairwise distinct lengths L_i (for convenience, let $L_1 < L_2 < \cdots < L_n$) and primitive feedback polynomials $p_i(x)$. Besides, the combination generator has a Finite State Machine (FSM) of k memory bits. Denote the k-bit state at time t by $\sigma_t = (\sigma_t^{k-1}, \ldots, \sigma_t^0)$. We denote λ_t hereafter the content of LFSRs at time t. Then the state of the combiner at time t is fully represented by the $(L + k)$-bit pair (λ_t, σ_t), where $L = \sum_{i=1}^{n} L_i$.

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

$$\sigma_{t+1} = \mathcal{F}(x_t, \sigma_t). \tag{3.1}$$

[1] see Appendix A for a brief review on LFSR

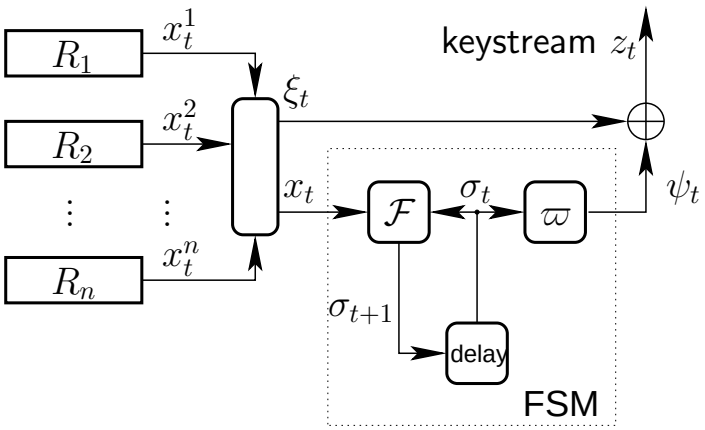

Figure 3.1: The core stream cipher

The FSM emits one bit

$$\psi_t = \varpi \cdot \sigma_t, \tag{3.2}$$

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

$$\xi_t \oplus \psi_t = z_t, \tag{3.3}$$

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

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

Proof. Noticing that λ_{t+1} (resp. σ_{t+1}) is a permutation of λ_t (resp. σ_t), by induction, we know that λ_t (resp. σ_t) is a permutation of λ_0 (resp. σ_0) for any t.

To prove the remaining part of the theorem, as $x_0, \ldots, x_{L_1-1}$ are contained in λ_0, we know that $L_1 - 1$ consecutive vectors $x_0, \ldots, x_{L_1-2}$ are i.i.d. random

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

variables all independent of both σ_0 and x_{L_1-1} assuming that (λ_0, σ_0) is random and uniformly distributed. From this statement we apply Eq.(3.1) consecutively for $t = 0, \ldots, L_1 - 2$ and deduce that the L_1-tuple $(\sigma_0, \sigma_1, \ldots, \sigma_{L_1-1})$ is independent of x_{L_1-1} assuming that (λ_0, σ_0) is random and uniformly distributed. □

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

3.2 Correlation Properties

The following definition is derived from the *normalized correlation* [84].

Definition 1 *The bias of a random Boolean variable X is defined as*

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

The normalized correlation between two random Boolean variables X and Y is just the bias of $X \oplus Y$. Assuming that (x_0, σ_0) is a uniformly distributed random vector of $(n + k)$ bits, we know that $\Delta(a \cdot \sigma_1 \oplus b \cdot \sigma_0)$ is a constant for any $a, b \in GF(2)^k$. The following important theorem, inspired by [59], gives an easy way to compute the bias for iterative structures.

Theorem 1 *Given a set $\mathcal{E}$ and $\Theta : \mathcal{E} \times GF(2)^k \to GF(2)$ and $\Lambda : GF(2)^\epsilon \to GF(2)^k$, let X and Y be two independent random variables in $\mathcal{E}$ and $GF(2)^\epsilon$ respectively. Assuming that $\Lambda(Y)$ is uniformly distributed in $GF(2)^k$, then, for any $v \in GF(2)^\epsilon$, we have*

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

Proof. Let $Z \in GF(2)^k$ be a random variable independent of X with uniform distribution. Starting from the right-hand side, we have

$$\begin{aligned} & \sum_w \Delta(\Theta(X, Z) \oplus w \cdot Z) \cdot \Delta(w \cdot \Lambda(Y) \oplus v \cdot Y) \\ = & \sum_w \mathrm{E}[(-1)^{\Theta(X,Z) \oplus w \cdot Z}] \cdot \mathrm{E}[(-1)^{w \cdot \Lambda(Y) \oplus v \cdot Y}] \\ = & \sum_x \sum_y \sum_z \sum_w \Pr(x, z) \cdot \Pr(y) \cdot (-1)^{\Theta(x,z) \oplus v \cdot y \oplus w \cdot (z \oplus \Lambda(y))}, \end{aligned} \quad (3.4)$$

As the inner sum over w in Eq.(3.4) is zero for all $z \neq \Lambda(y)$, we continue Eq.(3.4) as

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

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

Assuming (λ_0, σ_0) is uniformly distributed, for $\epsilon \leq L_1+1$ and any $\alpha_1, \ldots, \alpha_\epsilon \in GF(2)^k$, we know that $\Delta(\alpha_1 \cdot \sigma_0 \oplus \cdots \oplus \alpha_\epsilon \cdot \sigma_{\epsilon-1})$ is a constant. Denote it by $\delta(\alpha_1, \ldots, \alpha_\epsilon)$ hereafter. Let the state transition matrix $U = \{U_{ab}\}$ be defined by $U_{ab} \stackrel{\mathsf{def}}{=} \Pr(\sigma_{t+1} = b | \sigma_t = a)$, and we note that its Walsh transform $\widehat{U}_{ab} \stackrel{\mathsf{def}}{=} \sum_{a',b'} (-1)^{a \cdot a' \oplus b \cdot b'} U_{a'b'}$ satisfies

$$\widehat{U}_{ab} = 2^k \Delta(a \cdot \sigma_0 \oplus b \cdot \sigma_1) = 2^k \delta(a, b).$$

Now we introduce the general iterative computation method for $\delta(\alpha_1, \ldots, \alpha_\epsilon)$.

Corollary 1 *Assuming (λ_0, σ_0) is uniformly distributed, for any $\epsilon \leq L_1 + 1$ and $\alpha_1, \ldots, \alpha_\epsilon \in GF(2)^k$, we have*

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

Proof. We apply Theorem 1 with $X = x_{\epsilon-2}$, $Y = (\sigma_0, \ldots, \sigma_{\epsilon-2})$, $\Lambda(Y) = \sigma_{\epsilon-2}$, $\Theta(X, \Lambda(Y)) = \alpha_\epsilon \cdot \sigma_{\epsilon-1}$ and $v = (\alpha_1, \ldots, \alpha_{\epsilon-1})$. Note that the assumption of Theorem 1 holds by Property 1. □

Recall from [31] that the entropy $H(X)$ of a discrete random variable X with alphabet $\mathcal{X}$ is defined by

$$H(X) \stackrel{\mathsf{def}}{=} -\sum_{x \in \mathcal{X}} \Pr(x) \log_2 \Pr(x). \tag{3.5}$$

The binary entropy function $h(p)$ is defined by

$$h(p) \stackrel{\mathsf{def}}{=} -p \log_2 p - (1-p) \log_2 (1-p) \tag{3.6}$$

for $0 < p < 1$. The conditional entropy $H(Y|X)$ of Y given X is

$$H(Y|X) \stackrel{\text{def}}{=} \sum_{x \in \mathcal{X}} \Pr(x) H(Y|X = x). \tag{3.7}$$

For any two random variables X, Y we have

$$H(X) \geq H(X|Y) \tag{3.8}$$

with equality if and only if X and Y are independent. Analogously, for any three random variables X, Y, Z we have

$$H(X|Z) - H(X|Y, Z) \geq 0 \tag{3.9}$$

with equality if and only if X and Y are conditionally independent given Z.

Property 2 *Assuming (λ_t, σ_t) is uniformly distributed, we have a constant*

$$H(\psi_{t+1}|\sigma_t) = h\left(\frac{1}{2} + \frac{\rho}{2}\right), \tag{3.10}$$

for some positive constant ρ.

The reason goes as follows. We compute $H(\psi_{t+1}|\sigma_t)$ by definition:

$$\begin{aligned} H(\psi_{t+1}|\sigma_t) &= \sum_a H(\psi_{t+1}|\sigma_t = a) \cdot \Pr(\sigma_t = a) \\ &= \mathop{\mathrm{E}}_a \left[h\left(\frac{1}{2} + \frac{1}{2} \sum_{b:\varpi \cdot b = 1} U_{ab} - \frac{1}{2} \sum_{b:\varpi \cdot b = 0} U_{ab} \right) \right], \end{aligned} \tag{3.11}$$

which is a constant. So there exists such a unique $\rho \geq 0$ to satisfy Eq.(3.10), that is,

$$h\left(\frac{1}{2} + \frac{\rho}{2}\right) = \mathop{\mathrm{E}}_a \left[h\left(\frac{1}{2} + \frac{1}{2} \sum_{b:\varpi \cdot b = 1} U_{ab} - \frac{1}{2} \sum_{b:\varpi \cdot b = 0} U_{ab} \right) \right]. \tag{3.12}$$

Note that Eq.(3.12) tells that if $|\sum_{b:\varpi \cdot b=1} U_{ab} - \sum_{b:\varpi \cdot b=0} U_{ab}|$ is a constant ρ_0 for all a, then, $\rho = \rho_0$; in particular, $\rho = 0$ if and only if $\sum_{b:\varpi \cdot b=1} U_{ab} \equiv \sum_{b:\varpi \cdot b=0} U_{ab}$ for all a.

From Property 2, we can prove the upper bound of the correlations for the combiner's FSM output sequence.

Theorem 2 *For any $\epsilon \leq L_1+1$ and any binary $\alpha_1, \ldots, \alpha_{\epsilon-1}$, let the ϵ-bit vectors $\alpha = (\alpha_1, \ldots, \alpha_{\epsilon-1}, 1)$ and $\Psi_t = (\psi_t, \ldots, \psi_{t+\epsilon-1})$. Assuming (λ_t, σ_t) is uniformly distributed, we have*

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

where ρ is defined in Eq.(3.12).

Proof. First, by Eq.(3.8) we deduce that

$$\begin{aligned} H(\alpha \cdot \Psi_t) &\geq H(\alpha \cdot \Psi_t | \sigma_t, \ldots, \sigma_{t+\epsilon-2}) \\ &= H(\psi_{t+\epsilon-1} | \sigma_t, \ldots, \sigma_{t+\epsilon-2}). \end{aligned} \tag{3.13}$$

And according to Eq.(3.9), we have

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

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

$$H(\psi_{t+\epsilon-1} | \sigma_{t+\epsilon-2}) = H(\psi_{t+\epsilon-1} | \sigma_t, \ldots, \sigma_{t+\epsilon-2}) \tag{3.14}$$

Combining Eq.(3.13) and Eq.(3.14), we get

$$H(\alpha \cdot \Psi_t) \geq H(\psi_{t+\epsilon-1} | \sigma_{t+\epsilon-2}) = h\left(\frac{1}{2} + \frac{\rho}{2}\right).$$

Because $h(p)$ is symmetric in $p = \frac{1}{2}$ with the maximum at $p = \frac{1}{2}$, this is equivalent to

$$\frac{1}{2} - \frac{\rho}{2} \leq \Pr(\alpha \cdot \Psi_t = 0) \leq \frac{1}{2} + \frac{\rho}{2}, \tag{3.15}$$

Finally, we verify

$$\begin{aligned} |\Delta(\alpha \cdot \Psi_t)| &= |\Pr(\alpha \cdot \Psi_t = 0) - \Pr(\alpha \cdot \Psi_t \neq 0)| \\ &= |2 \cdot \Pr(\alpha \cdot \Psi_t = 0) - 1|. \end{aligned} \tag{3.16}$$

Putting Eq.(3.15) and Eq.(3.16) together we complete our proof. □

Remark 1 *This theorem tells that the basic FSM design principle should satisfy $H(\psi_{t+1} | \sigma_t) = 1$ to avoid the bias, which enables the keystream distinguishing attack and key-recovery attack as detailed in the rest of the chapter.*

Notice that the only purpose of the restriction on the dimension of α (i.e. $\epsilon \leq L_1 + 1$), is to ensure validity of U being the state transition matrix. In other words, if we loose this requirement by supposing U is always the state transition matrix[3], we still obtain the same upper bound ρ for $|\Delta(\alpha \cdot \Psi_t)|$. Though it is not known yet which tuple(s) α makes $|\Delta(\alpha \cdot \Psi_t)|$ the maximum from Theorem 2, one thing is certain[4]: once $\rho = 0$ (i.e. $\sum_{b:\varpi \cdot b=1} U_{ab} \equiv \sum_{b:\varpi \cdot b=0} U_{ab}$ for all a), no correlation exists for sequences of bitlength up to $L_1 + 1$.

Prior to our work, correlation properties of combiners with one-bit memory, and with m-bit memory were studied in [84], and [51] respectively. As they considered correlations of a general form (i.e. correlation between any linear function of the sequence $\{\xi_t\}$ of ϵ bits and any linear function of the keystream $\{z_t\}$ of ϵ bits), ϵ is restricted to be rather small for the analysis. In our work, we restrict ourselves to a special class of correlations—correlations of the FSM output sequence (i.e. correlations of any linear function of the sequence $\{\xi_t \oplus z_t\}$ of ϵ bits). This allows to investigate those correlations for the sequence length $\epsilon \leq L_1 + 1$ with much a wider range[5].

3.3 The Keystream Distinguisher

3.3.1 The Equivalent Single LFSR

Let θ_i be the order of the feedback polynomial $p_i(x)$ of R_i, for $i = 1, \ldots, n$. Since all $p_i(x)$ are primitive polynomials, $\theta_i = 2^{L_i} - 1$; furthermore, by Lemma 6.57 of [72, p.218], the equivalent LFSR to generate the same sequence of the sum of the n original LFSR outputs over $GF(2)$ has the feedback polynomial $p(x) = \prod_{i=1}^{n} p_i(x)$ with order $\theta = \mathrm{lcm}(\theta_1, \theta_2, \ldots, \theta_n)$ (by Lemma 6.50, [72, p.214]) and degree $L = \sum_{i=1}^{n} L_i$.

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

[4] This result was published most recently by an independent work [5] with different proof, which did not study the upper bound however.

[5] For instance, in the core of E0 (described in Section 3.6.1 later), according to [53], the sequence length $\epsilon \leq 6$ by analysis of [51] for general correlations; in contrast, for the special class of correlations in our work the sequence length $\epsilon \leq 27$ (see Section 3.6.2).

3.3.2 Finding the Multiple Polynomial with Low Weight

Let L be the degree of a general polynomial $p(x)$ with order θ. We use the standard approximation[6] to estimate the minimal weight w_d of multiples of $p(x)$ with degree at most d by the following constraint: w_d is the smallest w such that

$$\frac{1}{2^L} \times \left(\begin{array}{c} d \\ w-1 \end{array} \right) \geq 1. \tag{3.17}$$

Listed in Table 3.1 is the estimated[7] w_d corresponding to d with $L = 128$ by solving Inequality (3.17).

To find multiples with minimum weight, [18] proposed an efficient algorithm for a not too large degree d (e.g. less than 2^{11}). Here, we are interested in the case with very large $d \gg 2^{11}$. So we can use the conventional birthday paradox to find $Q(x)$ with the minimal d (i.e. $w = w_d$), which takes precomputation time $PT \approx O(d^{\lceil \frac{w-1}{2} \rceil})$; or we apply the generalized birthday problem [113] to find $Q(x)$ of same weight but higher degree with much less precomputation as tradeoff. Table 3.2 compares the two algorithms. Note that unless otherwise mentioned explicitly in the notations, throughout the thesis, we always use $\log(\cdot)$ to represent the natural logarithm to the base of e, which is omitted from the notations.

Table 3.1: The estimated minimal weight w_d of multiples of $p(x)$ with degree d and order θ by (3.17), where $L = 128$

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

3.3.3 Building a Uni-bias-based Distinguisher

Let $Q(x) = \sum_{i=1}^{w} x^{q_i}$ be the normalized multiple of $p(x) = \prod_{i=1}^{n} p_i(x)$ with degree d and weight w, where $0 = q_1 < q_2 < \ldots < q_w = d$. Let α be the ϵ-bit

[6] Note that this approximation of (3.17) is valid for typical settings in cryptography. However, it may not hold for some special cases (e.g. some of the products of two primitive polynomials with the same degree do not have any multiple polynomial of weight 3).

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

Table 3.2: Complexity PT of finding multiple of $p(x)$ with degree d, weight w where $L = 128$

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

binary vector such that $|\gamma|$ is maximal where $\gamma = \Delta(\alpha \cdot (\psi_t, \ldots, \psi_{t+\epsilon-1}))$. As $\bigoplus_{i=1}^{w} \bigoplus_{j=1}^{n} x_{t_0+q_i}^{j} = 0$ holds for all t_0, by Eq.(3.3), we deduce that

$$\bigoplus_{i=1}^{w} \alpha \cdot (z_{t_0+q_i}, \ldots, z_{t_0+q_i+\epsilon-1}) = \bigoplus_{i=1}^{w} \alpha \cdot (\psi_{t_0+q_i}, \ldots, \psi_{t_0+q_i+\epsilon-1}). \tag{3.18}$$

With the heuristic assumption of independence, we know from the famous Piling-up Lemma [77] that the right-hand side of Eq.(3.18) has a bias $|\gamma|^w$ (resp. $-|\gamma|^w$) if γ is positive (resp. negative). With standard linear cryptanalysis techniques, we can therefore distinguish the keystream $\{z_t\}$ from a truly random sequence with a number of samples within the order of magnitude of $\zeta = \gamma^{-2 \cdot w}$, simply by checking the left-hand side of Eq.(3.18) equals zero (resp. one) most of the time with the positive (resp. negative) γ. Based on $Q(x)$ with d and w, we minimize the data complexity Ξ by choosing $\Xi = \zeta + d = \gamma^{-2 \cdot w} + d$.

3.3.4 The Multi-bias-based Distinguisher

Preliminaries

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

1. $(f \otimes g)(a) = \sum_{b \in GF(2)^\ell} f(b) \cdot g(a \oplus b)$
   $f^{\otimes w}(a) = (\underbrace{f \otimes \cdots \otimes f}_{w \text{ times}})(a)$
2. $\widehat{f}(a) = \sum_{b \in GF(2)^\ell} (-1)^{a \cdot b} f(b)$

3. $\|f\| = \sqrt{\sum_{a \in GF(2)^\ell} f^2(a)}$

4. $\Delta(f) = 2^{\frac{\ell}{2}} \cdot \left\| f - \frac{1}{2^\ell} \cdot \mathbf{1} \right\|$, *where* $\mathbf{1}$ *denotes a constant function equal to* 1.

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

- $\widehat{f \otimes g}(a) = \widehat{f}(a) \cdot \widehat{g}(a)$, for all $a \in GF(2)^\ell$;
- $2^\ell \|f\|^2 = \|\widehat{f}\|^2$;
- if f is a distribution, i.e. $\sum_a f(a) = 1$ and $f(a) \geq 0$ for all $a \in GF(2)^\ell$, then the distribution of the XOR of w i.i.d. random vectors with distribution f is $f^{\otimes w}$, moreover, $\Delta^2(f) = \sum_{a \neq \mathbf{0}} \widehat{f}^{\,2}(a)$;
- If the random Boolean variable A follows the distribution f, then $\Delta(f) = \Delta(A)$, where $\Delta(A)$ is defined in Definition 1, Section 3.2.

An Efficient Way to Deploy Multi-Biases Simultaneously

Given a linear mapping $J : GF(2)^\nu \to GF(2)^\ell$ of rank ℓ, we define ℓ-bit vectors

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

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

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

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

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

for all $b \in GF(2)^\ell$. Now we discuss how to design J in order to reduce the data complexity. From Baignères et al. [9], we know that we can distinguish a distribution f of ℓ-bit random vectors from a uniform distribution with $1/\Delta^2(f)$ samples. Here, the distribution of B_t is $f = \mathcal{D}_A^{\otimes w}$. So the modified distinguisher needs data complexity

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

Let μ be the number of the largest Walsh coefficients $\widehat{\mathcal{D}}_A(b)$ over all nonzero b with the absolute value[8] η. Since $\Delta^2(\mathcal{D}_A^{\otimes w}) \approx \mu\eta^{2w}$, we obtain

$$\Xi \approx \frac{\ell}{\mu}\eta^{-2w} + d.$$

In order to lower Ξ, it is necessary to have $\ell < \mu$. This implies that only when the μ largest coefficients are linearly dependent, the multi-bias distinguisher is more efficient than the uni-bias distinguisher; otherwise, the former is as efficient as the latter. Note that Section 3.3.3 actually deals with the special type of distinguishers with $\ell = \mu = 1$.

3.4 The Key-recovery Attack

We approach similarly as in [39] to transform our keystream distinguisher of Section 3.3 into a key-recovery attack to reconstruct the shortest LFSR (i.e. R_1).

Now, let $Q(x) = \sum_{i=1}^{w} x^{q_i}$ be a multiple polynomial of $\prod_{i=2}^{n} p_i(x)$ with degree d and weight w, which can be found by techniques in Section 3.3.2. Let $\tilde{\mathbf{x}}^{\mathbf{1}}$ be a guess for $\mathbf{x}^{\mathbf{1}}$, the initial state of R_1 which generates the keystream $\{z_t\}$ together with the other $n-1$ fixed LFSRs. Denote $\tilde{x}_t^1$ the output bit of R_1 with the initial

[8] Note that from Corollary 1 we have $\eta \le \gamma \le \rho$ for $\nu \le L_1 + 1$ regardless of ℓ and J, where ρ is defined in Eq.(3.12).

state $\tilde{\mathbf{x}}^{\mathbf{1}}$ at time t. We define

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

for $t = 0, \ldots, \zeta - 1$ (corresponding to the data complexity $\Xi = \zeta + d$). It can be shown that $\{r_t\}$ is also an m-sequence generated by the same LFSR. Let $\mathbf{r}$ be the initial state. We define

$$b_t(\tilde{\mathbf{x}}^{\mathbf{1}}) \stackrel{\text{def}}{=} s_t \oplus r_t$$

for $t = 0, \ldots, \zeta - 1$. Given ζ-bit sequence of $b_t(\tilde{\mathbf{x}}^{\mathbf{1}})$'s, we count the occurrences[9] $N(\tilde{\mathbf{x}}^{\mathbf{1}})$ of ones, that is,

$$N(\tilde{\mathbf{x}}^{\mathbf{1}}) \stackrel{\text{def}}{=} \sum_{t=0}^{\zeta-1} b_t(\tilde{\mathbf{x}}^{\mathbf{1}}). \tag{3.19}$$

Two cases of statistical characteristics arise. We use similar analysis [111] for the case $\gamma > 0$, which can be easily adjusted for $\gamma < 0$.

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

We have

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

Recall from Section 3.3.3, we know that

$$p \stackrel{\text{def}}{=} \Pr(b_t(\tilde{\mathbf{x}}^{\mathbf{1}}) = 0) = \frac{1}{2} + \frac{\gamma^w}{2},$$

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

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

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

We have

Property 3

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

for any fixed keystream $\{z_t\}$*.*

From this property, We immediately reach another:

Property 4

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

for any fixed keystream $\{z_t\}$*.*

Remark 2 *We thus deduce that the average of* $N(\tilde{\mathbf{x}}^{\mathbf{1}})$ *over all* $\tilde{\mathbf{x}}^{\mathbf{1}} \neq \mathbf{x}^{\mathbf{1}}$ *is*

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

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

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

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

$$\mathrm{Pr}_{\mathrm{err}} \approx \Phi\left(-\frac{\frac{\zeta\gamma^w}{2}}{\sqrt{\frac{\zeta}{2}}}\right) = \Phi\left(-\frac{\sqrt{2\zeta}}{2} \cdot \gamma^w\right), \tag{3.20}$$

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

$$\mathrm{E}\left[\mathrm{Rank}_{N(\mathbf{x^1})}\right] = (2^{L_1}-1)\cdot \mathrm{Pr}_{\mathrm{err}} \approx 2^{L_1}\cdot \Phi\left(-\tfrac{\sqrt{2\zeta}}{2}\cdot\gamma^w\right). \tag{3.21}$$

As convention, we adopt the approximation $\Phi(x) \approx -\frac{1}{x\sqrt{2\pi}}e^{-\frac{x^2}{2}}$ for $x \ll -1$. Then, Eq.(3.21) reduces to

$$\mathrm{E}\left[\mathrm{Rank}_{N(\mathbf{x^1})}\right] \approx \frac{2^{L_1}}{\gamma^w\sqrt{\pi\zeta}}e^{-\frac{\zeta}{4}\gamma^{2w}}. \tag{3.22}$$

According to the conventional estimation [23, 62] in correlation attacks, the critical data complexity ζ_0 is on the order of γ^{-2w} and is defined by

$$\zeta_0 = \frac{L_1}{1-h(\frac{1}{2}+\frac{1}{2}\gamma^w)} \approx \frac{2L_1\log 2}{\gamma^{2w}},$$

and h is defined in Eq.(3.6). Hence, we set $\zeta = k_0\gamma^{-2w}$ for some k_0 to be determined. We solve $\mathrm{E}[\mathrm{Rank}_{N(\mathbf{x^1})}] = 1$ in Formula (3.22) and get

$$\log k_0 + 0.5k_0 = 2L_1\log 2 - \log\pi \approx 2L_1\log 2.$$

Since $\log k_0$ is much smaller than $0.5k_0$ when $L_1 \gg 1$, we estimate k_0 should not exceed $4L_1\log 2$. Therefore, it means that we need the minimum

$$\zeta \approx \frac{4L_1\log 2}{\gamma^{2w}} (= 2\zeta_0) \tag{3.23}$$

to guarantee that $N(\mathbf{x^1})$ is the smallest (resp. largest) of all $N(\tilde{\mathbf{x}}^{\mathbf{1}})$ with positive (resp. negative) γ. Note that this estimation $\zeta \approx 2\zeta_0$ is comparable to ζ_0. According to [23] simulations showed the probability of success is close to 1 (resp. $\frac{1}{2}$) for $\zeta = 2\zeta_0$ (resp. $\zeta = \zeta_0$) which is consistent with our analysis. Clearly, our problem of recovering R_1 right fits into the Maximum Likelihood Decoding (MLD) problem for a general linear code, as described immediately next in Section 3.5. Thus, solving MLD problem allows to recover $\mathbf{r}$, after which we apply linear transform to solve $\mathbf{x^1}$.

3.5 A Maximum Likelihood Decoding Algorithm

We first recall the following basics of linear codes (see [74] for details). Given a matrix $G_{L\times\kappa}$ (with $L < \kappa$), for every message $r = (r_1, \ldots, r_L)$, define the

codeword $x = (x_1, \ldots, x_\kappa) \stackrel{\text{def}}{=} rG$. The set of all codewords form the linear code, defined by G. The code is said to have dimension L, length κ and generator matrix G. The MLD problem for the linear code is: find the message r to minimize the Hamming distance[10] of its codeword x and the received vector $s = (s_1, \ldots, s_\kappa)$, i.e. find such r that minimizes $N(r) = \sum_{t=1}^{\kappa}(s_t \oplus x_t)$, where $x_t = rG_t$ (G_t denotes the t-th column vector of G).

For example, our preceding key-recovery attack in Section 3.4 can be transformed into the MLD problem as follows. Define the column vector G_t of the generator matrix G as

$$G_t = (a_0, \ldots, a_{L_1-1})^\top,$$

where $a_0 + a_1 x + \cdots + a_{L_1-1}x^{L_1-1} = x^t \mod p_1(x)$. And let $L = L_1$, $\kappa = \zeta$, $r = \mathbf{r}$, $x = \{r_t\}$ and $s = \{s_t\}$.

3.5.1 The Time-domain Analysis

Obviously, the trivial solution to find r is an exhaustive search in the time-domain as shown in Algorithm 1: for every message $\tilde{r}$, we compute $N(\tilde{r})$ and keep the smallest. The final record leads to r. The time complexity is $O(\kappa \cdot 2^L)$ with memory κ bits.

Algorithm 1 The exhaustive search algorithm

Inputs:
  $G = (G_1, \ldots, G_\kappa)$: the generator matrix
  keystream $s_1 s_2 \cdots s_\kappa$
Processing:
  record $\leftarrow 0$
  for all L-bit $\tilde{r}$ **do**
    compute $N(\tilde{r})$
    if $N(\tilde{r}) >$ record **then**
      $r \leftarrow \tilde{r}$
      record $\leftarrow N(\tilde{r})$
    end if
  end for
  output r

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

3.5.2 The Frequency-domain Analysis

We introduce an integer-valued function,

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

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

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

We thereby reach the theorem below.

Theorem 3

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

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

This generalizes the result [74, p. 414] of a special case when $\kappa = 2^L$ and $G_t^\top$ corresponds to the binary representation of t. So, to solve the MLD problem, we just compute $\mathcal{W}$, perform FWT (see [117]), and find the maximum $\widehat{\mathcal{W}}(r)$ as shown in Algorithm 2.

The time and memory complexities of FWT are $O(L \cdot 2^L)$, $O(2^L)$ respectively. Since the precomputation of $\mathcal{W}$ takes time $O(\kappa)$ with memory $O(\kappa)$, we conclude that our improved MLD algorithm runs in $O(\kappa + L \cdot 2^L)$ with memory $O(2^L)$ (additionally, using linear transformation allows to compute FWT over $GF(2)^k$ with memory $O(2^k)$ where $k = \lceil \log_2 \kappa \rceil$). Note that when $\kappa \ge 2^L$, the time complexity corresponds to $O(\kappa)$, which is optimal in the sense that it stands on the same order of magnitude as the data complexity does. Table 3.3 compares the original exhaustive search algorithm with the improved frequency transformation algorithm. Note that the technique of FWT was used in another context [24]

Algorithm 2 The frequency transformation algorithm

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

to speed up other kinds of fast correlation attacks. In the case of E0 (see Section 3.6), we will see how it helps to speed up the attack [39] by a factor of 2^{24}. We estimate similar correlation attacks like [23] can be speeded up by a factor of 10; undoubtedly, some other attacks can be significantly improved by our algorithm as well.

Table 3.3: Comparison of maximum likelihood decoding algorithms

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

3.5.3 A More Generalized MLD Algorithm

We further generalize the preceding problem by finding the L-bit vector r such that given a sequence of ℓ-bit ($\ell < L$) vectors $S_1, \ldots, S_\tau$ and $f : GF(2)^\ell \to \mathbf{R}$ together with matrices $G_1, \ldots, G_\tau$ of size L by ℓ, the sequence of ℓ-bit vectors $X_1, \ldots, X_\tau$ defined by $X_t = rG_t$ minimizes $N(r) = \sum_{t=1}^{\tau} f(S_t \oplus X_t)$. It means the linear code has length $\tau\ell$, dimension L, and the generator matrix $G = (G_1, \ldots, G_\tau)$. Note that our previous problem in Section 3.5.2 is merely a special case of $\ell = 1$, $\tau = \kappa$ and $f(a) = a$ for $a \in GF(2)$.

Define a real function

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

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

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

Algorithm 3 directly follows above computation. The total running time of our algorithm is $O(\tau \ell L 2^\ell + L 2^L)$ with memory $O(2^L)$. To speed up the computation of $\mathcal{W}$, we could precompute the inner products of all pairs of ℓ-bit vectors in time $O(2^{2\ell})$ with memory $O(2^{2\ell})$. Thus, the total running time of the algorithm is $O(2^{2\ell} + \tau L 2^\ell + L 2^L)$ with memory $O(2^{2\ell} + 2^L)$.

In the special case that $G_{t+1} = AG_t$ for $t = 1, \ldots, \tau$, we precompute another table to map any L-bit vector x to $xA^\top$. It takes time $O(2^L)$ with memory $O(2^L)$. The total time of the algorithm is thus $O\left(2^{2\ell} + (L + \tau)\, 2^\ell + L 2^L\right)$, with memory $O(2^{2\ell} + 2^L)$. Note that above special case is applicable to E0 (see Section 3.6).

3.5.4 An Optimum MLD Algorithm?

According to [12], the general decoding problem for linear codes is shown to be NP-complete (see [50] for definition) in the sense that the known deterministic algorithm that decodes an arbitrary linear code with dimension L and length κ performs an exhaustive trial on all possible codewords. This takes time $O(2^L)$ if we consider κ as a small negligible constant. In comparison, our result solves the problem when κ is not considered as a constant. And we showed that the decoding time is linear in κ.

Algorithm 3 The generalized MLD algorithm

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

3.6 Case Study: Bluetooth One-level E0

3.6.1 Description

Specified in [17], the core keystream generator E0 (Fig. 3.2) used in Bluetooth (a.k.a. one-level E0) fits in the model described in Section 3.1: $n = 4$, $L_1 = 25$, $L_2 = 31$, $L_3 = 33$, $L_4 = 39$ (and thus $L = 128$) with primitive feedback polynomials

$$\begin{array}{rcccccccccc} p_1(x) & = & x^{25} & + & x^{20} & + & x^{12} & + & x^8 & + & 1, \\ p_2(x) & = & x^{31} & + & x^{24} & + & x^{16} & + & x^{12} & + & 1, \\ p_3(x) & = & x^{33} & + & x^{28} & + & x^{24} & + & x^4 & + & 1, \\ p_4(x) & = & x^{39} & + & x^{36} & + & x^{28} & + & x^4 & + & 1, \end{array}$$

respectively. The state σ_t of the FSM contains k-bit (c_{t-1}, c_t), where $k = 4$ and $c_t = (c_t^1, c_t^0)$ has 2 bits. Let $w(x_t) \stackrel{\text{def}}{=} \sum_{i=1}^4 x_t^i$ be the Hamming weight[11] of x_t.

[11] Recall that the Hamming weight of a vector is the number of 1's of its coordinates. Note that the Hamming weight of a vector always equals its Hamming distance (defined in Section 3.5) to the all zero vector of equal dimension.

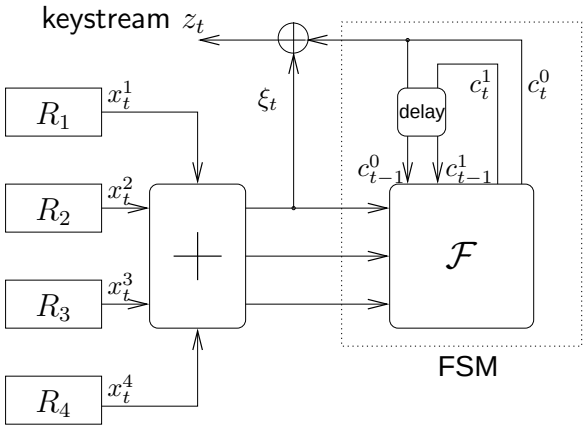

Figure 3.2: Outline of one-level E0

The FSM has the update function $\mathcal{F} : (w(x_t), c_{t-1}, c_t) \mapsto (c_t, c_{t+1})$. Computing c_{t+1} from σ_t can be described by

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

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

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

Table 3.4 shows the state transition of the FSM, where the four-bit state is represented in the quaternary system (e.g. the FSM changes from $\sigma_t = 13$ into $\sigma_{t+1} = 32$ by the input $w(x_t) = 2$). One can check Table 3.4 by above equations.

With $\varpi = 01$ in Eq.(3.2), at each clock cycle t, the FSM emits one bit $\psi_t = c_t^0$. The keystream output bit is $z_t = x_t^1 \oplus x_t^2 \oplus x_t^3 \oplus x_t^4 \oplus c_t^0$.

3.6.2 Correlations

From Section 3.2, we know that if (λ_0, σ_0) is uniformly distributed, then, for $\epsilon \leq 26$ and any $\alpha_1, \ldots, \alpha_\epsilon \in GF(2)^4$, $\delta(\alpha_1, \ldots, \alpha_\epsilon) = \Delta(\alpha_1 \cdot \sigma_0 \oplus \cdots \oplus \alpha_\epsilon \cdot \sigma_{\epsilon-1})$ is a constant. It can be computed by Corollary 1. However, notice that the core E0 has such a special FSM that the two consecutive states σ_t and σ_{t+1} are half overlapped (i.e. 2-bit c_t is contained in both). Therefore, to compute the value

Table 3.4: State transition of σ_{t+1} given $w(x_t)$ and σ_t

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

of $\Delta(\alpha_1 \cdot \sigma_0 \oplus \cdots \oplus \alpha_\epsilon \cdot \sigma_{\epsilon-1})$, the sequence $\alpha_1, \ldots, \alpha_\epsilon$ is not unique. So, we resort to another notation Ω for the unique expression of the same thing instead.

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

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

Here is a full list of nonzero triplets:

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

With it, we computed all ϵ-tuple biases for $\epsilon \leq 27$. All the largest biases, both of which were mentioned in [40, 53] (without formal proof), are summarized as below.

Property 5 *Assuming (λ_t, σ_t) is random and uniformly distributed, we have*

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

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

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

□

Remark 3 *Assuming $w(x_t) = 2$ holds for $t = t_0, t_0+1, t_0+2$, then, regardless of the value of σ_{t_0}, we always have*

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

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

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

which explains the bias in Property 5. This special case was not pointed out in [40, 53] however.

Property 6 *Assuming (λ_t, σ_t) is random and uniformly distributed, we have*

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

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

Throughout the rest of the chapter, we let

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

Besides the above two largest biases, we have the only second largest bias up to 27 bits $\Omega(1,0,1,1) = -2^{-4}$. This bias was already proved in [59]. Now, we apply Theorem 2 in Section 3.2 to compute the theoretical upper bound of $\Omega(a)$ for any a of at most 27 tuples and compare γ with it. To show this, we first list the state transition matrix U (where dashed entries denote zeros) as follows.

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

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

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

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

3.6.3 Keystream Distinguishers

We are ready to build a distinguisher for the core E0 upon above largest correlations together with the multiple $Q(x)$ of $\prod_{i=1}^{4} p_i(x)$ with degree d and weight w, which can be precomputed by birthday paradox as mentioned in Section 3.3.2 or easy manual calculation as follows:

Examples of $Q(x)$ with Weight Four

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

weight 4 with ease:

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

where

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

The degrees of $Q_1(x), Q_2(x), Q_3(x)$ are approximately $2^{69}, 2^{70}, 2^{65}$ respectively. Note that we may also expect optimal multiples with degree on the same order of magnitude and weight 3 from Table 3.1.

Primary Distinguisher

Table 3.5 summarizes the best performance of our primary (uni-bias-based) distinguisher for the core E0 based on either the use of $Q_3(x)$ with weight 4, or a search of $Q(x)$, when we choose $\alpha = (1,1,1,1,1)$ or $(1,0,0,0,0,1)$.

Table 3.5: Summary of the best primary distinguisher for the core E0

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

Advanced Distinguisher

From Section 3.3.4, we know that the multi-bias-based distinguisher improves on the uni-bias-based one if and only if the largest correlation coefficients are linearly dependent, which happens to be true in the core E0: recall from Property 5, Property 6 that the 6-tuple vectors of the three largest biases satisfy the linear relation,

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

As a simple solution we may just pick $\nu = 6$, $\ell = 2$, $\alpha_1 = (1,1,1,1,1,0)$ and $\alpha_2 = (0,1,1,1,1,1)$ (where α_i is the i-th row of J), then we obtain $\mu = 3$. And the data complexity Ξ is reduced to a factor of $\frac{2}{3}$ for negligible d. Indeed, recall that we proved by computation that the largest Walsh coefficient for $\nu \leq 27$ are either $(0,\ldots,0,1,1,1,1,1,0,\ldots,0)$ or $(0,\ldots,0,1,0,0,0,0,1,0,\ldots,0)$. Thus $\mu \leq (\nu-4)+(\nu-5) = 2\nu-9$. This leads to a more general solution, if we pick $\nu = \ell + 4$, and the i-th row of J as

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

then we obtain $\mu = 2\ell - 1$. And so the improved factor $\frac{\ell}{2\ell-1}$ of data complexity Ξ tends to $\frac{1}{2}$ for negligible d when ℓ goes to infinity; however, because of the underlying assumption for the core E0, ν is restricted to no larger than 27, i.e. $\ell \leq 23$. To conclude, we show that the modified distinguisher (Algorithm 4) needs data complexity

$$\Xi \approx \frac{\ell}{2\ell-1} \cdot \gamma^{-2w} + d, \text{ for } 1 \leq \ell \leq 23. \tag{3.24}$$

Table 3.6 shows the best improvement achieved with $\ell = 23$. We see that the minimum Ξ drops from previous 2^{34} to 2^{33}.

Table 3.6: Data complexity Ξ of the advanced distinguisher for the core E0

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

3.6.4 The Key-recovery Attack

Let $Q(x) = \sum_{i=1}^{w} x^{q_i}$ be the multiple polynomial of $\prod_{i=2}^{4} p_i(x)$ with degree d and weight w. $Q(x)$ can be found with (precomputation) complexity PC by techniques in Section 3.3.2. Table 3.7 lists the corresponding triplets (w,d,PC) for small w. Detailed in Section 3.4, we use the MLD algorithm in Section 3.5.2 to recover $\mathbf{x^1}$. Table 3.8 shows our estimated minimal ζ corresponding to w by Eq.(3.23). Moreover, we conduct the same analysis as in Section 3.6.3 to

Algorithm 4 The advanced distinguisher for the core E0

Parameters:

$\ell \in [1, 23], \nu = \ell + 4$

$J : GF(2)^{\nu} \rightarrow GF(2)^{\ell}$

$\mathcal{D}_A$: the probability distribution of the ℓ-bit vector A_t

$Q(x) = \sum_{i=1}^{w} x^{q_i}$: the multiple polynomial of $p_1(x)p_2(x)p_3(x)p_4(x)$ with degree d

Ξ: the sample size by Eq.(3.24)

Inputs:

keystream $z_0 z_1 \cdots z_{\Xi-1}$ of either a truly random source $\mathcal{S}_0$ or the output $\mathcal{S}_1$ generated by the core E0

initialize counters $u_0, u_1, \ldots, u_{2^{\ell}-1}$

for $t = 0, 1, \ldots, \lfloor \frac{\Xi-d-4}{\ell} \rfloor - 1$ **do**

compute $b = \bigoplus_{i=1}^{w} J(z_{\ell t+q_i}, \cdots, z_{\ell t+q_i+\nu-1})$

increment u_b

end for

if $\sum_b u_b \cdot \log\left(2^{\ell} \cdot \mathcal{D}_A^{\otimes w}(b)\right) > 0$ **then**

accept $\mathcal{S}_1$ as the source

else

accept $\mathcal{S}_0$ as the source

end if

decrease ζ by a factor of $\frac{\ell}{2\ell-1}$ for $1 \leq \ell \leq 23$; and we apply the technique introduced in Section 3.5.3 to obtain the time complexity $O(\Xi+\theta_1 \cdot 2^{\ell}+L_1 \cdot 2^{L_1})$, where $\Xi = \zeta + d$. The attack complexities to recover R_1 for the core E0 are listed in Table 3.9 for two best cases denoted by A and B, where we choose $\ell = 12$.

Table 3.7: Complexity PC of finding the multiple of $p_2(x)p_3(x)p_4(x)$ with degree d and weight w

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

Table 3.8: The estimated minimal ζ corresponding to w by Eq.(3.23) where $L_1 = 25$, $\gamma = 25/256$

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

Table 3.9: Summary of primary partial key-recovery attacks against R_1 for the core E0

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

Once we recover R_1, we target R_2 next based on multiple of $p_3(x)p_4(x)$. Last, we use the technique of guess and determine in [47] to solve R_3 and R_4 with knowledge of the shortest two LFSRs. The detailed complexities of each step are shown in Table 3.10. A comparison of our attacks with the similar attack[12] [39] and the best algebraic attack [28, 57] is shown in Table 3.11 for

[12]The estimate of data complexity in [39] uses a different heuristic formula than ours. However we believe that their estimate and ours in Attack B are essentially the same.

case A and B.

Table 3.10: Detailed complexities of our key-recovery attack against the core E0

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

Table 3.11: Complexities comparison of our attacks with the similar attack and the algebraic attack

		precomputation	time	data	memory
Algebraic attack	[28, 57]	2^{37}	2^{49}	$2^{23.4}$	2^{37}
Similar attack	[39]	2^{54}	2^{63}	2^{34}	2^{34}
Our attacks	A	2^{37}	2^{39}	2^{39}	2^{27}
	B	2^{54}	2^{37}	2^{36}	2^{27}

Experimental Results with $\mathbf{w = 1}$

We did the small-scale experiment to verify our analysis in Section 3.4 on the keystream $\{\bigoplus_{i=2}^{4}(x_t^i \oplus z_t)\}$ instead of $\{z_t\}$ to save the trouble of searching the multiple $Q(x)$ of $\prod_{i=2}^{4} p_i(x)$ with low weight (herein $w = 1$). First, we test the rank of $N(\mathbf{x^1})$ among those of all the 2^{L_1} values of $N(\tilde{\mathbf{x}}^\mathbf{1})$ (see Eq.(3.19) for definition) for a total of 100 randomly chosen initial states of the core E0. From Eq.(3.21), we have $\mathrm{E}[\mathrm{Rank}_{N(\mathbf{x^1})}] = 1$ for $\zeta = 2^{14}$. It turned out that $N(\mathbf{x^1})$ ranks uniquely the top without exception.

Second, we choose some random $\mathbf{x^1}$, then compute the corresponding average and variance of $\frac{N(\tilde{\mathbf{x}}^\mathbf{1})}{\zeta}$ over all $\tilde{\mathbf{x}}^\mathbf{1} \neq \mathbf{x^1}$ individually, it turned out that $\mathrm{Var}(\frac{N(\tilde{\mathbf{x}}^\mathbf{1})}{\zeta}) \approx 1.526 \times 10^{-5}$, approximately the same as the expected $\mathrm{Var}(\frac{N(\tilde{\mathbf{x}}^\mathbf{1})}{\zeta}) = \frac{1}{\zeta^2}\mathrm{Var}(N(\tilde{\mathbf{x}}^\mathbf{1})) = \frac{1}{4\zeta} = 2^{-16} \approx 1.526 \times 10^{-5}$; and we got a consistent average of 0.5. The left curve in Figure 3.3 corresponds to the experimental probability

distribution of $\frac{N(\tilde{\mathbf{x}}^{\mathbf{1}})}{\zeta}$ for $\tilde{\mathbf{x}}^{\mathbf{1}} \neq \mathbf{x}^{\mathbf{1}}$, where the dotted line represents the central symmetric line.

Last, we accordingly tested the average and variance of $\frac{N(\mathbf{x}^{\mathbf{1}})}{\zeta}$ for 2^{25} random initial states of the core E0. And we got the average of around 0.5488 with variance 2.121×10^{-5} (in contrast to the estimation of average $\frac{281}{512} \approx 0.5488$, variance $2^{-16} \approx 1.526 \times 10^{-5}$ respectively). Its experimental probability distribution is drawn on the right curve of Figure 3.3. It is worth noticing that the two curves are indeed distinct.

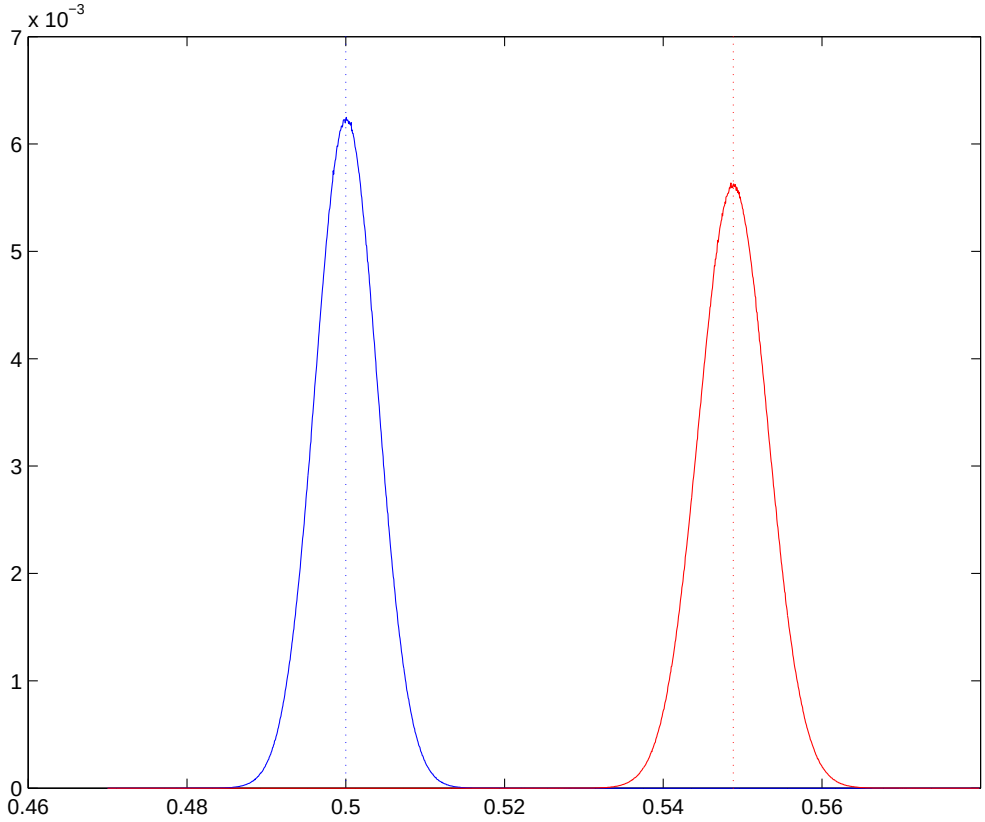

Figure 3.3: The two distinct probability distributions of $\frac{N(\tilde{\mathbf{x}}^{\mathbf{1}})}{\zeta}$ for $\tilde{\mathbf{x}}^{\mathbf{1}} \neq \mathbf{x}^{\mathbf{1}}$ (left) and $\tilde{\mathbf{x}}^{\mathbf{1}} = \mathbf{x}^{\mathbf{1}}$ (right).

3.7 Summary

In this chapter, we propose an E0-like combiner with memory as the core keystream generator. Next, we formulate a systematic computation method of correlations by a recursive expression, which makes it easier to calculate correlations of the FSM output sequences (up to certain bits). In addition, we give a useful upper bound for the correlations for the designer to take into account. When correlations are found, we can build either a uni-bias-based or multi-bias-based distinguisher to distinguish the keystream produced by the combiner from a truly random sequence. We apply the concept of convolution to the analysis of the multi-bias-based distinguisher that uses all correlations. Based on the theory of [9], it is shown that the multi-bias-based distinguisher outperforms the uni-bias-based distinguisher only when the largest biases are linearly dependent. The keystream distinguisher is very powerful not only because it enables the keystream distinguishing attack, but also because it can upgrade into the key-recovery attack. The latter actually reduces to the well-known MLD problem given the keystream long enough (or the bias large enough). By means of FWT, we devise a novel MLD algorithm to recover the closest codeword for any linear code. It is an optimal deterministic algorithm in the sense that the basic MLD problem was shown to be NP-complete in [12].

The analysis principle is successfully applied to the core E0 completely. Our key-recovery attack works in 2^{39} time given 2^{39} consecutive keystream bits after $O(2^{37})$ precomputation. This was the best academic key-recovery attack against the core E0 when published in 2004. Considering a maximal keystream length of 2745 bits for the practical E0 used in Bluetooth, the results still remain the academic interest. Moreover, our proposed MLD algorithm can be easily adapted to speed up a class of fast correlation attacks.

All in all, an ideal core keystream generator should satisfy the basic design principle: the FSM must generate no biased output sequence, i.e.

$$H(\psi_{t+1}|\sigma_t) = 1.$$

Chapter 4

The Resynchronization Scheme

4.1 Introduction

In the classic stream ciphers, the secret key must be renewed to generate fresh keystream for each encryption in order to avoid keystream reuse. In the case of the dedicated stream ciphers, it is impractical for each user to renew his secret key for each encryption, hence the need of the reinitialization mechanism by some public parameter(s) (a.k.a. nonce). A typical reinitialization scheme (a.k.a. resynchronization scheme) is depicted in Figure 4.1, where the user's private key $\mathcal{K}$ and the nonce $\mathcal{P}^i$ are used to initialize the core keystream generator by a function INIT.

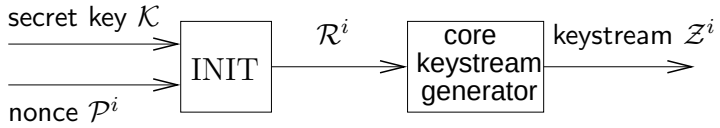

Figure 4.1: Practical keystream generator initialization scheme

Let

$$\mathcal{R}^i = \mathrm{INIT}(\mathcal{K}, \mathcal{P}^i) \tag{4.1}$$

be the initial state of the core keystream generator, where the superscript i is used throughout this chapter to indicate the context of the i-th frame. The keystream $\mathcal{Z}^i$ is expressed by

$$\mathcal{Z}^i = \mathsf{KeystreamGen}(\mathcal{R}^i). \tag{4.2}$$

When INIT satisfies

$$\mathrm{INIT}(\mathcal{K}, \mathcal{P}^i) = G_1(\mathcal{K}) \oplus G_2(\mathcal{P}^i) \tag{4.3}$$

with $G_1 : GF(2)^n \rightarrow GF(2)^L$ being affine, we call it one-level initialization scheme. When INIT is nonlinear, for example, to ease design, INIT can be defined to be

$$\mathrm{INIT}(\mathcal{K}, \mathcal{P}^i) = G_3\left(\mathsf{KeystreamGen}(G_1(\mathcal{K}) \oplus G_2(\mathcal{P}^i))\right) \tag{4.4}$$

for affine G_1, G_2 and G_3. Since the final output is obtained by passing through the keystream generator twice, we call it two-level initialization scheme.

In this chapter, we investigate the possibility of upgrading the correlation attack on the underlying core stream cipher into the correlation attack against the dedicated stream cipher. More formally, given m frames of keystreams $\mathcal{Z}^1, \ldots, \mathcal{Z}^m$ with the corresponding m nonces $\mathcal{P}^1, \ldots, \mathcal{P}^m$, let $\mathcal{R}^i = \mathrm{INIT}(\mathcal{K}, \mathcal{P}^i)$ be the unknown initial state of the core keystream generator involving the same secret key $\mathcal{K}$ of L bits. Assume that $\mathcal{R}^i_t \oplus \mathcal{Z}^i_t$ has biased pattern α of ϵ bits and bias γ, i.e. $\alpha \cdot (\mathcal{R}^i_{[t,t+\epsilon-1]} \oplus \mathcal{Z}^i_{[t,t+\epsilon-1]})$ is i.i.d. bit with bias[1] γ for all $i \in [1, m]$ and $t \in [1, n]$, where the unified notation $\mathcal{A}^i_{[a,b]}$ with the formatted subscript is used throughout the chapter to denote the vector $(\mathcal{A}^i_a, \ldots, \mathcal{A}^i_b)$ of $(b - a + 1)$ bits. Use the minimum m to recover $\mathcal{K}$.

4.2 Security Analysis

We begin with a simple problem of finding the closest sequences with fixed differences.

4.2.1 One Decoding Problem

Given L-bit sequences $\phi^1, \ldots, \phi^m$, find the L-bit sequence $r^1 = r^1_1 \cdots r^1_L$ that maximizes $N(r^1) = \sum_{i=1}^{m} \sum_{t=1}^{L} (r^1_t \oplus \phi^i_t)$.

Note that if we let $r^i = r^1 \oplus \pi^i$ and $s^i = \phi^i \oplus \pi^i$, where L-bit sequences $\pi^1, \ldots, \pi^m$ are given and $\pi^1 = \mathbf{0}$, it is clear to see that the problem of finding the sequences r^i's with fixed pairwise differences (specified by π^i's) which have the minimum Hamming distance to s^i's reduces to the above problem.

[1] see Definition 1 in Section 3.2.

Similar to the well-known approach (e.g. [53, p251]), the solution based on the idea of minority vote goes fairly easy. We have

$$r_t^1 = \text{minority}\{\phi_t^i : i = 1, \ldots, m\}$$

for all $t = 1, \ldots, L$. Note that in case of a tie for r_t^1, we have two answers for this t-th bit regardless of all the other bits. We finally obtain all the answers that achieve the same maximal $N(r^1)$. The time and memory complexities of the above algorithm both equal the data complexity $O(mL)$.

4.2.2 Applications: Attack the Resynchronization Scheme

Variant One: Symbol-based Decoding

We study the one-level resynchronization scheme Eq.(4.3) with $L \simeq n$ or $L \leq n$. For convenience, we let

$$r_t^i = \alpha \cdot \mathcal{R}^i_{[t,t+\epsilon-1]}, \tag{4.5}$$

$$s_t^i = \alpha \cdot \mathcal{Z}^i_{[t,t+\epsilon-1]}, \tag{4.6}$$

for $i \in [1, m]$ and $t \in [1, n]$. From our assumption, $(r_t^i \oplus s_t^i)$ is i.i.d. bit with bias γ for $i \in [1, m]$ and $t \in [1, n]$. Let

$$\pi^i = r^i \oplus r^1 = (G_2' \circ G_2)(\mathcal{P}^i) \oplus (G_2' \circ G_2)(\mathcal{P}^1),$$

where $G_2' : \mathcal{R}^i_{[1,n+\epsilon-1]} \mapsto r^i_{[1,n]}$ depends on α. Note that π^i's are known as $\mathcal{P}^i$'s are known. Supposing $\gamma < 0$, then, we see from [9] that regardless of L, the minimum

$$m = \frac{4 \log 2}{\gamma^2} \tag{4.7}$$

suffices to guarantee that $N(r^1)$ is the largest over all possible n-bit vectors. Thus, we apply the decoding idea in Section 4.2.1 to recover r^1 first. Once r^1 is recovered, we solve the linear Eq.(4.3) to obtain the full $\mathcal{K}$ if $L \leq n$; in case $n < L$, we get n bits of L-bit $\mathcal{K}$ and the remaining $(L-n)$ bits can be exhaustively tried within negligible time since $n \simeq L$. This makes the time/memory/data complexities all equal to $O(m \cdot L)$. The case $\gamma > 0$ can be similarly solved with same data and time complexities.

Remark 4 *With the one-level initialization scheme, the correlation attack of the underlying core stream cipher leads to that of the full dedicated stream cipher with same correlation.*

Note that for the special case of memoryless combiners, our resynchronization attack is essentially the same as the independent simultaneous work [6]; however, for the general combiners with memory, the related work [6] proposed to guess the memory state for each frame and differed from our approach.

Variant Two: Block-based Decoding

For a general INIT (e.g. $n \ll L$ in the previous variant), we want to recover as many bits of $\mathcal{K}$ as possible. We discuss how to achieve it in the following cases. Assume that for all i, we have a function[2]

$$\Upsilon^i_{[1,n']} = \mathrm{g}(K', \mathcal{Z}^i, \mathcal{P}^i, B^i),$$

for fixed n', where K' is the ℓ-bit subkey of $\mathcal{K}$ ($\ell \leq L$) and B^i is unknown (i.e. B^i is a function involving $\mathcal{K}$ and $\mathcal{Z}^i$ or $\mathcal{P}^i$). Suppose Υ^i_t is i.i.d. bit with bias γ' for all $i \in [1, m]$ and $t \in [1, n']$. We shall show that Section 4.2.1 is still applicable to recover K'. For each candidate $\widetilde{K'}$, define the grade

$$\mathrm{Grade}(\widetilde{K'}) \stackrel{\mathsf{def}}{=} \sum_{i=1}^{m} \sum_{B} (f \circ \mathrm{g})(\widetilde{K'}, \mathcal{Z}^i, \mathcal{P}^i, B), \tag{4.8}$$

for some $f : GF(2)^{n'} \rightarrow \mathbf{R}$ to be determined later. Let $\mathcal{D}$ be the distribution of the n'-bit vector, each bit of which is i.i.d. with bias γ'. With a random $\mathcal{P}^i$ and the correct $\widetilde{K'} = K'$, $B = B^i$, we know that the vector $\mathrm{g}(\widetilde{K'}, \mathcal{Z}^i, \mathcal{P}^i, B)$ complies with the distribution $\mathcal{D}$; otherwise, we approximate it to be random and uniformly distributed. Using the analysis similar to [65] which is inspired by [9, 111], when we choose $f(x) = \mathcal{D}(x) - \frac{1}{2^{n'}}$ for all n'-bit x, $\mathrm{Grade}(K')$ is expected to be the largest of all $\mathrm{Grade}(\widetilde{K'})$ with minimum

$$m \approx \frac{4(\#B)\ell \log 2}{k{\gamma'}^2} \leq \frac{4(\#B)\ell \log 2}{n'{\gamma'}^2}, \tag{4.9}$$

where k denotes total number of n'-bit x's such that $|\widehat{\mathcal{D}}(x)| = |\gamma'|$. Note that $k \geq n'$. By computing the grade exhaustively in Eq.(4.8) for all candidates, we

[2]Note that such a function can be easily deduced from the correlations of the core stream cipher and G_3 as detailed in Section 4.3.3 for the two-level initialization scheme satisfying Eq.(4.4); however, the problem of how to find such a function for a general INIT is beyond the scope of our work.

get ℓ-bit K'. Algorithm 5 illustrates this simple idea of exhaustive trial. The time complexity is $O(m \cdot \#B \cdot 2^{\ell})$. From Eq.(4.9), we see that the data complexity grows proportional to $\#B$ while the time complexity grows proportional to $(\#B)^2$.

Finally, in Table 4.1 we compare the two decoding variants in identical settings, i.e. $L = \ell$, $\gamma = \gamma'$ and $k = n' = n \simeq L$. From the table we see the main difference between the two variants is that the decoding time is linear in the key length in the first variant and exponential in the second variant, which roots from the two different decoding approaches.

Note that the related simultaneous work [6] gave some discussion on attacking the memoryless combiner with two-level initialization scheme but without detailed complexity analysis. And [6] used the similar approach to our previous symbol-based decoding variant.

Table 4.1: Comparison of two decoding variants

variants	time	memory	frames
Variant One	$4L \log 2/\gamma^2$	$4 \log 2/\gamma^2$	
Variant Two	$4(\#B)^2 2^L \log 2/\gamma^2$	$4(\#B) \log 2/\gamma^2$	

4.3 Case Study: Bluetooth Encryption

According to the Bluetooth standard [17], a two-level initialization scheme is used with the core keystream generator to produce the keystream for encryption.

4.3.1 Review on Bluetooth Reinitialization Scheme: Two-level E0

Recall that as detailed in Section 3.3.1 and Section 3.6.1, the core keystream generator E0 (both dashed boxes in Fig. 4.2) can be viewed equivalently as a nonlinear filter generator, which contains a single L-bit LFSR ($L = 128$) with the feedback polynomial equal to the product of those of the four component LFSRs and a 4-bit FSM. The keystream bit of the generator is obtained by xoring

Algorithm 5 The block-based decoding for Variant Two

Parameters:
f, g
m by Eq.(4.9)
Inputs:
nonces $\mathcal{P}^1, \ldots, \mathcal{P}^m$
keystreams $\mathcal{Z}^1, \ldots, \mathcal{Z}^m$
Processing:
for all ℓ-bit $\widetilde{K'}$ **do**
$\mathrm{Grade}(\widetilde{K'}) \leftarrow 0$
for $i = 1$ to m **do**
for all B **do**
$\mathrm{Grade}(\widetilde{K'}) \leftarrow \mathrm{Grade}(\widetilde{K'}) + (f \circ \mathrm{g})(\widetilde{K'}, \mathcal{Z}^i, \mathcal{P}^i, B)$
end for
end for
end for
find K' which has the largest $\mathrm{Grade}(K')$
output K'

the output bit of the LFSR with that of the FSM, which takes the current state of the LFSR as input and emits one bit out of its 4-bit memory[3].

Bluetooth two-level E0 (Fig. 4.2) uses two inputs: the effective encryption key $\mathcal{K}$ of length[4] $|\mathcal{K}| \in \{8i : i = 1, \ldots, 16\}$ no larger than L, and the nonce[5] $\mathcal{P}^i$. The initial state $R^i_{[1-L,0]}$ of the equivalent LFSR at the first level E0 for the i-th frame is set by

$$R^i_{[1-L,0]} = G_1(\mathcal{K}) \oplus G_2(\mathcal{P}^i), \tag{4.10}$$

for $i = 1, \ldots, m$, where G_1 and G_2 are affine transformations over $GF(2)^L$. E0 runs at level one and produces L-bit output

$$S^i_{[1-L,0]} = R^i_{[1-L,0]} \oplus \beta^i_{[1-L,0]},$$

where the detail of initializing FSM (by clocking four LFSRs) is omitted here for

[3] At each clock cycle, the FSM updates the state from its current state and LFSRs output, see Section 3.6.1 for details.

[4] For regulation reasons like the negotiation between the Bluetooth devices, the original L-bit encryption key is artificially shrunk to fewer bits.

[5] $\mathcal{P}^i$ includes a 26-bit counter and some user-dependent constant.

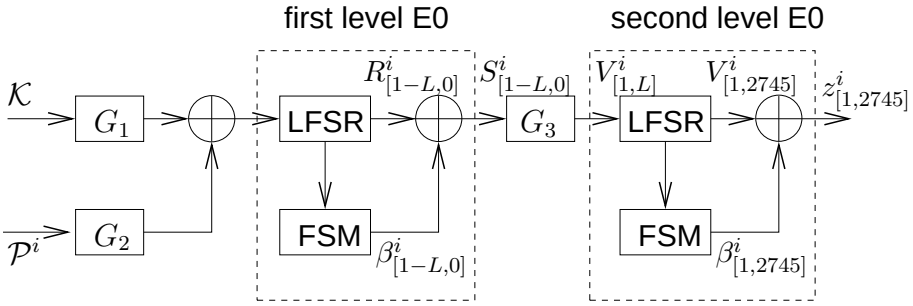

Figure 4.2: Initialization scheme of Bluetooth two-level E0

simplicity. Then the equivalent single LFSR is reinitialized for E0 level two by

$$V^i_{[1,L]} = G_3(S^i_{[1-L,0]}),$$

where $G_3 : GF(2)^L \to GF(2)^L$ is another affine transformation; however, the state of the FSM is retained. The initialization scheme is complete up to now. Note that this Bluetooth initialization scheme belongs to our two-level initialization scheme defined earlier in Eq.(4.4).

E0 runs at level two to produce the final keystream $z^i_t = V^i_t \oplus \beta^i_t$ for $t = 1, 2, \ldots, 2745$ for encryption of the i-th frame.

An Important Note on $\mathbf{G_3}$

We observe that G_3 is implemented in such a simple way[6] that the last L-bit output sequence of the first level E0 is byte-wise reloaded into the four component LFSRs in parallel at the second level E0 with only a few exceptions, which turns out to be a flaw as shown later in Section 4.3.3. Table 4.2 lists in time order the first 24 output bits of $\mathrm{LFSR}_1, \ldots, \mathrm{LFSR}_4$ individually at the beginning of E0 level two, in terms of the L-bit input $v_0, \ldots, v_{L-1}$. Note that the loading of LFSRs at E0 level two for reinitialization is in reverse order of the keystream output at E0 level one according to Table 4.2.

[6]It is believed to help to increase the rate of keystream generation.

Table 4.2: The first 24 output bits of LFSRs at E0 level two

LFSR	output bits
LFSR_1	$v_{71} \cdots v_{64}, v_{39} \cdots v_{32}, v_7 \cdots v_0$
LFSR_2	$v_{79} \cdots v_{72}, v_{47} \cdots v_{40}, v_{15} \cdots v_8$
LFSR_3	$v_{111} \cdots v_{104}, v_{87} \cdots v_{80}, v_{55} \cdots v_{48}$
LFSR_4	$v_{119} \cdots v_{112}, v_{95} \cdots v_{88}, v_{63} \cdots v_{56}$

4.3.2 Attack on One-level E0

Here, we present the attack on one-level E0 with the maximal key length $|\mathcal{K}| = L$, i.e. we can observe frames of L-bit keystream output $S^i_{[1-L,0]}$ at E0 level one[7].

Let ϵ-bit $\alpha = (1, 1, 1, 1, 1)$, where $\epsilon = 5$. And let

$$\begin{aligned} \mathcal{R}^i_{[1,L]} &\stackrel{\mathsf{def}}{=} R^i_{[1-L,0]}, \\ \mathcal{Z}^i_{[1,L]} &\stackrel{\mathsf{def}}{=} S^i_{[1-L,0]}. \end{aligned}$$

Recall that from Chapter 3, $\alpha \cdot (\mathcal{R}^i_{[t,t+4]} \oplus \mathcal{Z}^i_{[t,t+4]}) = \alpha \cdot \beta^i_{[t,t+4]}$ is assumed to be the i.i.d. bit with the largest known bias $\gamma = \Omega(\alpha) = -\lambda$ for all $i \in [1, m]$ and $t \in [1, n]$, where $n = L - 4$ and $\lambda = \frac{25}{256}$. As $n \simeq L$, we can apply the first variant decoding problem in Section 4.2.2 to recover the full $\mathcal{K}$. According to Eq.(4.7), we need $m \approx 2^{8.2} < 2^9$ frames of keystreams $\mathcal{Z}^i$'s and nonces $\mathcal{P}^i$'s for $i = 1, \ldots, m$. This implies the time/data/memory complexities $O(2^{16})$. No precomputation is needed.

To verify this, we ran experiments on 2^9 frames of the randomly-chosen 132-bit E0 initial state 2^{25} times. It turned out that we had 1.5 errors and 0.4 tie in average, which means we can easily correct all errors by an extra checking step in the end in negligible time. Table 4.3 compares our result with the four known[8] attacks [46,47,70,99] working on frames of L-bit consecutive keystreams. Note that attacks here differ from those attacks in previous chapter in that the latter concentrated on the keystream generator with a single (long) frame, while

[7] However, according to [17], Bluetooth SIG takes the correlation properties into account and adopts the two-level scheme for keystream generation in practice on purpose.

[8] In the similar approached paper [53], m is chosen as 45 for E0 level one without the complexity estimate, because the authors focused on the two-level E0 and traded m with the time complexity of E0 level one, whose time complexity is negligible with that of the E0 level two.

the former take the resynchronization scheme into account and use many short frames.

Table 4.3: Comparison of our attack with existing attacks against E0 level one given frames of L bits with $|\mathcal{K}| = L$

attacks	precomputation	time	frames	data	memory
Guess & Determine [99]	-	2^{93}	1	2^{7}	-
BDD [70]	-	2^{77}	1	2^{7}	-
Algebraic Attack [47]	-	2^{51}	2	2^{8}	2^{51}
Algebraic Attack [46]	-	$2^{23.4}$	3	$2^{8.6}$	$2^{23.4}$
Our Correlation Attack	-	2^{16}	2^{9}	2^{16}	2^{16}

4.3.3 Attack on Two-level E0

We study the correlation attack on two-level E0, i.e. we observe frames of keystreams $z^i_{[1,2745]}$ at E0 level two but the keystreams $S^i_{[1-L,0]}$ at E0 level one are unavailable. Let

$$U^i_{[1,L]} \stackrel{\text{def}}{=} G_3(R^i_{[1-L,0]}). \tag{4.11}$$

Following the description of G_3 in Table 4.2 in Section 4.3.1, we can verify that

$$\begin{aligned}
V^i_t &= U^i_t \oplus \beta^i_{-56-t} \oplus \beta^i_{-48-t} \oplus \beta^i_{-16-t} \oplus \beta^i_{-8-t}, && \text{for } t \in [1,8],\\
V^i_t &= U^i_t \oplus \beta^i_{-80-t} \oplus \beta^i_{-72-t} \oplus \beta^i_{-32-t} \oplus \beta^i_{-24-t}, && \text{for } t \in [9,16],\\
V^i_t &= U^i_t \oplus \beta^i_{-104-t} \oplus \beta^i_{-96-t} \oplus \beta^i_{-56-t} \oplus \beta^i_{-48-t}, && \text{for } t \in [17,24].
\end{aligned}$$

From above, we summarize the characteristics of V^i_t by

$$V^i_t = U^i_t \oplus \beta^i_{t_1} \oplus \beta^i_{t_2} \oplus \beta^i_{t_3} \oplus \beta^i_{t_4}, \tag{4.12}$$

for $t \in [1,24]$, where $t_1, \ldots, t_4$ are functions in terms of t and always satisfy the relations $t_1 < t_2 < t_3 < t_4$, $t_2 - t_1 = t_4 - t_3 = 8$ and $t_3 - t_2 \geq 32$ for $t \in [1,24]$. We express the keystream bit z^i_t of the second level E0 by

$$z^i_t = U^i_t \oplus \beta^i_{t_1} \oplus \beta^i_{t_2} \oplus \beta^i_{t_3} \oplus \beta^i_{t_4} \oplus \beta^i_t, \tag{4.13}$$

for $t \in [1, 24]$. Thus, for any α with $\epsilon \in [1, 8]$, let $\bar{\alpha}$ represent the vector in reverse order of α, and we have

$$\bar{\alpha} \cdot (z^i_{[t,t+\epsilon-1]} \oplus U^i_{[t,t+\epsilon-1]}) = \bigoplus_{j=1}^{4} \alpha \cdot \beta^i_{[t_j,t_j+\epsilon-1]} \oplus \bar{\alpha} \cdot \beta^i_{[t,t+\epsilon-1]} \tag{4.14}$$

holds for all $i \in [1, m]$ and $t \in \bigcup_{k=0}^{2}\{8k + 1, \ldots, 8k + 9 - \epsilon\}$, which turns out to be a potentially critical resynchronization flaw as detailed next. Note that the usage of the bar operator reflects the fact that the loading of LFSRs at E0 level two for reinitialization is in reverse order of the keystream output at E0 level one as mentioned earlier.

Assuming independence of β^i_t's, the left-hand side of Eq.(4.14) has bias $\Omega(\bar{\alpha}) \cdot \Omega(\alpha)^4$. From Section 3.6.2, the optimal $\alpha = (1, 1, 1, 1, 1)$ (with $\epsilon = 5$) leads to the largest bias $-\lambda^5$. Let

$$\begin{aligned} \mathcal{R}^i_{[1,24]} &\stackrel{\text{def}}{=} U^i_{[1,24]}, \\ \mathcal{Z}^i_{[1,24]} &\stackrel{\text{def}}{=} z^i_{[1,24]}. \end{aligned}$$

Note that $U^i_{[1,L]} = (G_3 \circ G_1)(\mathcal{K}) \oplus (G_3 \circ G_2)(\mathcal{P}^i)$ from Eq.(4.10,4.11), and thus it fits in our one-level initialization scheme defined earlier in Eq.(4.3). Given the 5-bit α, derive $r^i_{[4k+1,4k+4]}$ and $s^i_{[4k+1,4k+4]}$ from $\mathcal{R}^i_{[8k+1,8k+8]}$ and $\mathcal{Z}^i_{[8k+1,8k+8]}$ respectively for $k = 0, 1, 2$ according to Eq.(4.5,4.6). And we apply the first variant decoding problem in Section 4.2.2 to recover $n = 3(9 - 5) = 12$ bits of $\mathcal{K}$ with $\gamma = -\lambda^5$ and $m \approx 2^{35}$ frames of data.

As we are interested in gaining more bits of $\mathcal{K}$, we present in the following how to apply the second variant decoding problem in Section 4.2.2 to retrieve the full $\mathcal{K}$. For any subset $\mathcal{E}$ of the universe $\{1, 2, 3, 4\}$, we rewrite Eq.(4.14) as

$$\begin{aligned} & \bar{\alpha} \cdot (z^i_{[t,t+\epsilon-1]} \oplus U^i_{[t,t+\epsilon-1]}) \oplus \bigoplus_{j\in\mathcal{E}} \alpha \cdot \beta^i_{[t_j,t_j+\epsilon-1]} \\ = & \bigoplus_{j\in\bar{\mathcal{E}}} \alpha \cdot \beta^i_{[t_j,t_j+\epsilon-1]} \oplus \bar{\alpha} \cdot \beta^i_{[t,t+\epsilon-1]}. \end{aligned} \tag{4.15}$$

Recall from Chapter 3 we know that $\beta_{[t+1,t+\epsilon]}$ can be determined from $x_{[t+2,t+\epsilon-1]}$ and σ_{t+2}. Therefore, the left-hand side (denote by τ^i_t) of Eq.(4.15) is a function of the subkey K' of $\ell = 1 + 4|\mathcal{E}|(\epsilon - 2)$ bits, z^i, $\mathcal{P}^i$ and the unknown $B^i = \bigcup_{k\in\mathcal{E}} \sigma^i_{t_k+2}$ of $4|\mathcal{E}|$ bits ($1 \le |\mathcal{E}| \le 4$). τ^i_t has bias $-\Omega(\bar{\alpha}) \cdot \Omega(\alpha)^{4-|\mathcal{E}|}$. Again, the

optimal choice of α is $(1,1,1,1,1)$ which yields the largest bias $-\lambda^{5-|\mathcal{E}|}$. Clearly, we can view the left-hand side of Eq.(4.15) as g (with $n'=1$, $\gamma'=-\lambda^{5-|\mathcal{E}|}$) to fit in the second decoding variant problem in Section 4.2.2.

Additionally, we can further extend g as follows. For any $n' \in [1,4]$, we know that computing the n'-bit $\tau^i_{[t,t+n'-1]}$ involves the subkey of $\ell = 4|\mathcal{E}|(n'+2)+1$ bits and the same B^i together with $z^i, \mathcal{P}^i$. Treating this as our new g, we have more possibilities for the attack. Note that we have $k = 2n'-1$ according to Section 3.6.2. In Table 4.4, we compare all possible parameter choices and conclude that the best choice is $|\mathcal{E}|=1$, $\ell=13$ and $m=2^{36}$ frames with the time complexity $O(2^{53})$ or $|\mathcal{E}|=1$, $\ell=17$ and $m=2^{35}$ frames with the time complexity $O(2^{56})$.

Table 4.4: Parameter choices for the attack on two-level E0

$\lvert\mathcal{E}\rvert$	n'	k	$\lvert\gamma'\rvert$	$\#B$	ℓ	frames	time
1	1	1	λ^4	2^4	13	$2^{36.0}$	$2^{53.0}$
1	2	3	λ^4	2^4	17	$2^{34.8}$	$2^{55.8}$
1	3	5	λ^4	2^4	21	$2^{34.4}$	$2^{59.4}$
1	4	7	λ^4	2^4	25	$2^{34.2}$	$2^{63.2}$
2	1	1	λ^3	2^8	25	$2^{34.2}$	$2^{67.2}$
2	2	3	λ^3	2^8	33	$2^{33.1}$	$2^{74.1}$
2	3	5	λ^3	2^8	41	$2^{32.6}$	$2^{81.6}$
2	4	7	λ^3	2^8	49	$2^{28.4}$	$2^{81.4}$
3	1	1	λ^2	2^{12}	37	$2^{32.1}$	$2^{81.1}$
3	2	3	λ^2	2^{12}	49	$2^{30.9}$	$2^{91.9}$
3	3	5	λ^2	2^{12}	61	$2^{30.5}$	$2^{103.5}$
3	4	7	λ^2	2^{12}	73	$2^{26.3}$	$2^{107.3}$
4	1	1	λ	2^{16}	49	$2^{29.8}$	$2^{94.8}$
4	2	3	λ	2^{16}	65	$2^{28.6}$	$2^{109.6}$
4	3	5	λ	2^{16}	81	$2^{28.2}$	$2^{125.2}$
4	4	7	λ	2^{16}	97	$2^{20.0}$	$2^{125.0}$

For a full attack, we iterate the above attack with different t's with knowledge of the reconstructed subkey. The attack time complexity is bounded by the first attack. The final results are listed in Table 4.5 to compare with existing attacks [46, 47, 53] on two-level E0 for $|\mathcal{K}| = L$.

Table 4.5: Comparison of our attack with the best attacks [46, 47, 53] against two-level E0 for $|\mathcal{K}| = L$

attacks	precomputation	time	frames	data	memory
[47]	-	2^{73}	-	2^{43}	2^{51}
[46]	2^{80}	2^{65}	2	$2^{12.4}$	2^{80}
[53]	2^{80}	2^{70}	45	2^{17}	2^{80}
Our Attacks	-	2^{53}	2^{36}	$2^{40.6}$	2^{36}
	-	2^{56}	2^{35}	$2^{39.6}$	2^{35}

4.4 Summary

In this chapter, by attaching the one-level or two-level initialization scheme, we transformed the core stream cipher into the dedicated stream cipher. The possibility of upgrading the correlation attack against the underlying core stream cipher into the correlation attack against the dedicated stream cipher is further investigated for the two schemes. We showed that the correlation attack on the core stream cipher leads directly to the correlation attack on the dedicated stream cipher with the one-level initialization scheme (with equal bias), but not necessarily so with the two-level initialization scheme.

The analysis is applied to attacking the one-level and two-level E0 respectively, whose core stream cipher was systematically investigated with regards to the correlation properties in the previous chapter. For the attack on the one-level E0 which is not in use by Bluetooth specification, it proves to be very efficient within $O(2^{16})$ given 2^9 frames of 128-bit consecutive keystreams. Meanwhile, for the attack on two-level E0, based on a resynchronization flaw, we are able to deduce its non-trivial correlations from those of the core E0. This enables the correlation attack within 2^{53} (resp. 2^{56}) simple operations given the first 24 bits of 2^{36} (2^{35}) frames in order to recover the 128-bit key. Compared with existing academic attacks, this was the best attack when published in 2004; nonetheless, the attack is still impractical due to the fact that the required frame number is beyond the limit of maximum available frames.

Chapter 5

Conditional Correlation Attack

5.1 Background

The concept of (ordinary) correlations was first extended to the *conditional correlation* to describe the *linear* correlation of the inputs conditioned on a *given* (short) output pattern of a nonlinear function (with small input size) in [2,71,73]. Based on conditional correlations, the conditional correlation attack received successful studies towards the nonlinear filter generator in [2,71,73]. In this chapter, we assign a different meaning to conditional correlations, i.e. the correlation of the output of an arbitrary function (with favorable small input size) conditioned on the *unknown* (partial) input which is uniformly distributed. This might be viewed as the generalized opposite of [2, 71, 73]. As a useful application of our conditional correlations, imagine the attacker not only observes the keystream, but also has access to an intermediate computation process controlled partly by the key, which outputs a hopefully biased sequence for the right key and (presumably) unbiased sequences for wrong keys. If such side information is available, the *conditional correlation attack* may become feasible, which exploits correlations of the intermediate computation output conditioned on (part of) the inputs. In general, as informally conjectured in [73], conditional correlations are different and often larger than ordinary (unconditional) correlations, which effects reduced data complexity (as well as time complexity) of conditional correlation attacks over ordinary correlation attacks.

This initiates our work to extract a precise and general statistical model for dedicated key-recovery distinguishers based on the generalized conditional correlations. This framework deals with a specific kind of smart distinguishers

that exploit correlations conditioned on the (partial) input, which is not restricted to keystream generators and is also applicable to other scenarios (e.g. side channel attacks like fault attacks demonstrated in [7]).

5.2 Preliminaries

Given the function $f : \mathcal{E} \to GF(2)^\ell$, define the distribution D_f of $f(X)$ with X uniformly distributed, i.e.,

$$D_f(a) \stackrel{\text{def}}{=} \frac{1}{|\mathcal{E}|} \sum_{X \in \mathcal{E}} \mathbf{1}_{f(X)=a}$$

for all $a \in GF(2)^\ell$. Following [9], recall that the Squared Euclidean Imbalance (SEI) of the distribution D_f is defined by

$$\mathsf{SEI}(D_f) = \Delta^2(D_f) = 2^\ell \sum_{a \in GF(2)^\ell} \left(D_f(a) - \frac{1}{2^\ell} \right)^2, \tag{5.1}$$

where $\Delta(\cdot)$ was defined in Section 3.3.4 and with $\ell = 1$ it is referred to as the well-known term *correlation*. For brevity, we adopt the simplified notations $\Delta(f), \mathsf{SEI}(f)$ to denote $\Delta(D_f), \mathsf{SEI}(D_f)$ respectively hereafter. From the theory of hypothesis testing and Neyman-Pearson likelihood ratio (see [9]), $\mathsf{SEI}(f)$ tells us that the minimum number n of samples for an optimal distinguisher to effectively distinguish a sequence of n output samples of f from $(2^L - 1)$ truly random sequences of equal length is

$$n = \frac{4L \log 2}{\mathsf{SEI}(f)}. \tag{5.2}$$

Note that the result in Eq.(5.2) with $\ell = 1$ has long been known up to a constant factor $\frac{1}{2}$ in the theory of channel coding. In fact, correlation attacks has been very successful for almost two decades to apply the distinguisher that analyzes the biased sample of a single bit (i.e. the case $\ell = 1$) in order to reconstruct the L-bit key (or subkey), where only the right key can produce a biased sequence while all the wrong keys produce unbiased sequences. In Chapter 3, on the sound theoretical basis [9] of the generalized distinguisher, we showed that this generalized distinguisher helps to improve the correlation attack when considering multi-biases simultaneously.

5.3 Our Problem

Given a function $f : GF(2)^u \times GF(2)^v \to GF(2)^r$, let $f_{\mathcal{B}}(X) = f(\mathcal{B}, X)$ for $\mathcal{B} \in GF(2)^u$ and $X \in GF(2)^v$, where the notation $f_{\mathcal{B}}(\cdot)$ is used to replace $f(\cdot)$ whenever $\mathcal{B}$ is given. Consider such a game between a player and an oracle. Each time the oracle secretly generates $\mathcal{B}, X$ independently and uniformly to compute $f_{\mathcal{B}}(X)$; the player, in turn, sends a guess on the current value of the partial input $\mathcal{B}$. Only when he guesses correctly, the oracle would output the value of $f_{\mathcal{B}}(X)$, otherwise, it would output a random and uniformly distributed $Z \in GF(2)^r$. Suppose the player somehow manages to collect 2^L sequences of n interaction samples with the following characteristics: one sequence has n samples $(f_{\mathcal{B}_i^{\mathcal{K}}}(X_i), \mathcal{B}_i^{\mathcal{K}})$ $(i = 1, \ldots, n)$ where $\mathcal{B}_i^{\mathcal{K}}$'s and X_i's are independently and uniformly distributed; the remaining $(2^L - 1)$ sequences all consist of n independently and uniformly distributed random variables $(Z_i^K, \mathcal{B}_i^K)$ $(i = 1, \ldots, n)$ for $K \neq \mathcal{K}$. One interesting question to the player is how to distinguish the biased sequence from the other sequences by using the minimum number n of samples.

Note that the above problem is of special interest in key-recovery attacks, including the related-key attacks, where $\mathcal{B}_i^{\mathcal{K}}$'s are the key-related material (i.e. computable with the key and other random public parameters) and the oracle can be viewed as an intermediate computation process accessible to the attacker with only a limited number of queries. Thus, when the attacker knows the right key $\mathcal{K}$ he can collect n (hopefully biased) samples of f; on the other hand, if he uses the wrong key, he will only collect an unbiased sequence.

5.4 Smart Distinguisher with Side Information

From Section 5.2, we know that the minimum number n of samples for the basic distinguisher which does not use the partial input $\mathcal{B}_i$'s is $n = 4L \log 2 / \mathsf{SEI}(f)$. When the samples are incorporated with the $\mathcal{B}_i$'s, we can prove the following stronger result.

Theorem 4 *The smart distinguisher (in Algorithm 6) solves our above problem with*

$$n = \frac{4L \log 2}{\mathrm{E}[\mathit{SEI}(f_{\mathcal{B}})]} \tag{5.3}$$

and the time complexity $O(n \cdot 2^L)$, where the expectation is taken over all the uniformly distributed $\mathcal{B}$.

Remark 5 *Our smart distinguisher (Algorithm 6) turns out to be a derivative of the basic distinguisher in [9] and the result Eq.(5.3) for the simple case $r = 1$ was already pointed out (without proof) in [53] with a mere difference of a negligible constant term $2 \log 2 \approx 2^{0.47}$. Also note that the quantity $\mathrm{E}[\mathsf{SEI}(f_{\mathcal{B}})]$ in Eq.(5.3) measures the correlation of the output of an arbitrary function conditioned on the (partial) input which is uniformly distributed and* unknown[1]*. In contrast, prior to our work, the conditional correlation, that refers to the linear correlation of the inputs conditioned on a given (short) output pattern of a nonlinear function, was well studied in [2, 71, 73] based on a different statistical distance other than SEI. Highly motivated by the security of the nonlinear filter generator, their research focused on the case where the nonlinear function is the augmented nonlinear filter function (with small input size) and the inputs are the involved LFSR taps. Obviously, the notion of our conditional correlation can be seen as the generalized opposite of [2, 71, 73], that addresses the issue of how to make the most use of all the data for the success.*

Proof. Let us introduce a new distribution D over $GF(2)^{r+u}$ from $D_{f_{\mathcal{B}}}$ defined by

$$D(\mathcal{B}, Z) = \frac{1}{2^u} D_{f_{\mathcal{B}}}(Z), \tag{5.4}$$

for all $\mathcal{B} \in GF(2)^u, Z \in GF(2)^r$. We can see that our original problem is transformed into that of the basic distinguisher to distinguish D from a uniform distribution. According to Section 5.2, we need minimum $n = 4L \log 2/\mathsf{SEI}(D)$. So we compute $\mathsf{SEI}(D)$ by Eq.(5.1,5.4):

$$\begin{aligned}
\mathsf{SEI}(D) &= 2^{r+u} \sum_{\mathcal{B} \in GF(2)^u} \sum_{Z \in GF(2)^r} \left(D(\mathcal{B}, Z) - \frac{1}{2^{r+u}} \right)^2 \\
&= 2^{r+u} \sum_{\mathcal{B} \in GF(2)^u} \sum_{Z \in GF(2)^r} \left(\frac{1}{2^u} D_{f_{\mathcal{B}}}(Z) - \frac{1}{2^{r+u}} \right)^2 \\
&= 2^{-u} \sum_{\mathcal{B} \in GF(2)^u} 2^r \sum_{Z \in GF(2)^r} \left(D_{f_{\mathcal{B}}}(Z) - \frac{1}{2^r} \right)^2 \\
&= \mathrm{E}[\mathsf{SEI}(f_{\mathcal{B}})]. \qquad (5.5)
\end{aligned}$$

Meanwhile, the best distinguisher tries to maximize the probability $\prod_{i=1}^{n} D(\mathcal{B}_i, Z_i)$, i.e. the conditional probability $\prod_{i=1}^{n} D_{f_{\mathcal{B}_i}}(Z_i)$. As the conventional approach, we

[1] According to the rule of our game, it is unknown to the distinguisher whether the sample $\mathcal{B}$ is the correct value used for the oracle to compute $f_{\mathcal{B}}(X)$ or not.

know that this is equivalent to maximize $G = \sum_{i=1}^{n} \log_2(2^r \cdot D_{f_{\mathcal{B}_i}}(Z_i))$ as shown in Algorithm 6. The time complexity of the distinguisher[2] is obviously $O(n \cdot 2^L)$. □

Algorithm 6 The smart distinguisher with side information

Parameters:

n set by Eq.(5.3)

$D_{f_{\mathcal{B}}}$ for all $\mathcal{B} \in GF(2)^u$

Inputs:

uniformly and independently distributed u-bit $\mathcal{B}_1^K, \ldots, \mathcal{B}_n^K$ for all L-bit K

$Z_1^{\mathcal{K}}, \ldots, Z_n^{\mathcal{K}} = f_{\mathcal{B}_1^{\mathcal{K}}}(X_1), \ldots, f_{\mathcal{B}_n^{\mathcal{K}}}(X_n)$ for one fixed L-bit $\mathcal{K}$ with uniformly and independently distributed v-bit vectors $X_1, \ldots, X_n$

uniformly and independently distributed sequences $Z_1^K, Z_2^K, \ldots, Z_n^K$ for all L-bit K such that $K \neq \mathcal{K}$

Goal: find $\mathcal{K}$

Processing:

for all L-bit K **do**

  $G(K) \leftarrow 0$

  for $i = 1, \ldots, n$ **do**

    $G(K) \leftarrow G(K) + \log_2\left(2^r \cdot D_{f_{\mathcal{B}_i^K}}(Z_i^K)\right)$

  end for

end for

output $\mathcal{K}$ that maximizes $G(\mathcal{K})$

5.5 Optimal Smart Distinguisher

Theorem 5 *The distinguisher (in Algorithm 6) can be optimized to achieve the time complexity $O(n + L \cdot 2^{L+1})$ with precomputation $O(L \cdot 2^L)$, when $\mathcal{B}_i^K$'s and Z_i^K's can be expressed by:*

$$\mathcal{B}_i^K = \mathcal{L}(K) \oplus c_i\,, \tag{5.6}$$

$$Z_i^K = \mathcal{L}'(K) \oplus c_i' \oplus g(\mathcal{B}_i^K)\,, \tag{5.7}$$

[2] In this thesis, we only discuss the deterministic distinguisher. For the probabilistic distinguisher, many efficient and general decoding techniques (e.g. the probabilistic iterative decoding), which are successful in correlation attacks, were carefully presented in the related work [73] and such techniques also apply to our distinguisher.

for all L-bit K and $i = 1, 2, \ldots, n$, where g is an arbitrary function, $\mathcal{L}, \mathcal{L}'$ are $GF(2)$-linear functions, and c_i's, c_i''s are independently and uniformly distributed which are known to the distinguisher.

Proof. Let us first introduce two functions $\mathcal{H}, \mathcal{H}'$:

$$\mathcal{H}(K) = \sum_{i=1}^{n} \mathbf{1}_{\mathcal{L}(K)=c_i \text{ and } \mathcal{L}'(K)=c_i'} \tag{5.8}$$

$$\mathcal{H}'(K) = \log_2 \left(2^r \cdot D_{f_{\mathcal{L}(K)}} \left(\mathcal{L}'(K) \oplus g\left(\mathcal{L}(K)\right)\right)\right) \tag{5.9}$$

for $K \in GF(2)^L$. We can see that $G(K)$ computed in Line 7 to 10, Algorithm 6 is nothing but a simple convolution (denoted by $\otimes$) between $\mathcal{H}$ and $\mathcal{H}'$:

$$G(K) = (\mathcal{H} \otimes \mathcal{H}')(K) \stackrel{\text{def}}{=} \sum_{K' \in GF(2)^L} \mathcal{H}(K')\mathcal{H}'(K \oplus K'), \tag{5.10}$$

for all $K \in GF(2)^L$. It is known that convolution and Walsh transform (denoted by the hat symbol) are transformable, so we have

$$G(K) = \frac{1}{2^L} \widehat{\widehat{\mathcal{H} \otimes \mathcal{H}'}}(K) = \frac{1}{2^L} \widehat{\mathcal{H}''}(K), \tag{5.11}$$

where $\mathcal{H}''(K) = \widehat{\mathcal{H}}(K) \cdot \widehat{\mathcal{H}'}(K)$. This means that after computing $\mathcal{H}$ and $\mathcal{H}'$, the time complexity of our smart distinguisher would be dominated by three times of FWT, i.e. computing $\widehat{\mathcal{H}}, \widehat{\mathcal{H}'}, \widehat{\mathcal{H}''}$ in $O(3L \cdot 2^L)$. Moreover, since only c_i's, c_i''s may vary from one run of the attack to another, which are independent of $\mathcal{H}'$, we can also precompute $\widehat{\mathcal{H}'}$ and store it in the table; finally, the real-time processing only takes time $O(n + L \cdot 2^{L+1})$. □

In Section 5.7, Theorem 5 is directly applied to attacking Bluetooth two-level E0 for a truly practical attack.

5.6 Conditional Correlation vs. Unconditional Correlation

Theorem 6 *We have*

$$\mathrm{E}[\mathit{SEI}(f_{\mathcal{B}})] \geq \mathit{SEI}(f),$$

where equality holds if and only if (iff) $D_{f_{\mathcal{B}}}$ is statistically independent of $\mathcal{B}$.

Remark 6 *As* $\mathrm{E}[SEI(f_{\mathcal{B}})], SEI(f)$ *measures the conditional correlation and the unconditional correlation respectively, this property convinces us that the former is no smaller than the latter. This relationship between the conditional correlation and the unconditional correlation was informally conjectured in [73]. We conclude from Eq.(5.3) that the smart distinguisher having partial (or side) information (i.e.* $\mathcal{B}$ *herein) about the biased source generator (i.e.* $f_{\mathcal{B}}$ *herein) always works better than the basic distinguisher governing no knowledge of that side information, as long as the generator is statistically dependent on the side information. Our result verifies the intuition that the more the distinguisher knows about the generation of the biased source, the better it works. In particular, Theorem 6 implies that even if the fact that* $SEI(f) = 0$ *causes the basic distinguisher to be completely useless as it needs infinite data complexity, in contrast, the smart distinguisher would still work as long as* $D_{f_{\mathcal{B}}}$ *is statistically dependent on* $\mathcal{B}$*, i.e.* $\mathrm{E}[SEI(f_{\mathcal{B}})] > 0$*. In Section 5.7.2, we give illustrative examples* $\mathrm{E}[SEI(f_{\mathcal{B}})]$ *on the core of Bluetooth E0 to be compared with their counterparts* $SEI(f)$*.*

Proof. By Eq.(5.5), we have

$$\mathrm{E}[\mathrm{SEI}(f_{\mathcal{B}})] = 2^r \sum_{A \in GF(2)^r} \mathrm{E}\left[\left(D_{f_{\mathcal{B}}}(A) - \frac{1}{2^r}\right)^2\right],$$

where the expectation is taken over uniformly distributed $\mathcal{B}$ for the fixed A. On the other hand, since $D_f(A) = \mathrm{E}[D_{f_{\mathcal{B}}}(A)]$ for any fixed A, we have

$$\begin{aligned}
\mathrm{SEI}(f) &= 2^r \sum_{A \in GF(2)^r} \left(D_f(A) - \frac{1}{2^r}\right)^2 \\
&= 2^r \sum_{A \in GF(2)^r} \left(\mathrm{E}\left[D_{f_{\mathcal{B}}}(A)\right] - \frac{1}{2^r}\right)^2 \\
&= 2^r \sum_{A \in GF(2)^r} \mathrm{E}^2\left[D_{f_{\mathcal{B}}}(A) - \frac{1}{2^r}\right],
\end{aligned}$$

by definition of Eq.(5.1), with all the expectation taken over uniformly distributed $\mathcal{B}$ for the fixed A. As we know from theory of statistics that for any fixed A,

$$0 \leq \mathrm{Var}\left[D_{f_{\mathcal{B}}}(A) - \frac{1}{2^r}\right] = \mathrm{E}\left[\left(D_{f_{\mathcal{B}}}(A) - \frac{1}{2^r}\right)^2\right] - \mathrm{E}^2\left[D_{f_{\mathcal{B}}}(A) - \frac{1}{2^r}\right]$$

always holds, where equality holds iff $D_{f_{\mathcal{B}}}(A)$ is statistically independent of $\mathcal{B}$.

□

5.7 Case Study: Attack on Bluetooth Two-level E0

5.7.1 Preliminaries and Notations

In order to review the reinitialization flaw introduced in Chapter 4 for our purposes, we first introduce some notations. Define the binary vector $\alpha = (\alpha_0, \alpha_1, \ldots, \alpha_{\ell-1})$ of length $|\alpha| = \ell \geq 3$ with $\alpha_0 = \alpha_{\ell-1} = 1$ and let $\bar{\alpha} \stackrel{\text{def}}{=} (\alpha_{\ell-1}, \alpha_{\ell-2}, \ldots, \alpha_0)$ represent the vector in reverse order of α. Let $B_t \stackrel{\text{def}}{=} x_t \in GF(2)^4$ be the four output bits of LFSRs at time instance t, and $X_t \stackrel{\text{def}}{=} \sigma_t \in GF(2)^4$ be the FSM state[3] at time instance t. Given ℓ and t, for the one-level E0, we define $\mathcal{B}_{t+1} = B_{t+1}B_{t+2}\ldots B_{t+\ell-2}$ and $C_t = (c_t^0, \ldots, c_{t+\ell-1}^0)$. Recall from Section 3.6.1 that the computation of the next state X_{t+1} of the FSM only depends on its current state X_t together with the Hamming weight $w(B_t)$ of B_t. Therefore, the function $h^{\alpha}_{\mathcal{B}_{t+1}} : X_{t+1} \mapsto \alpha \cdot C_t$ is well defined[4] for all t, where the dot operator between two vectors represents the inner product. For clarity, we refer the time instance t and t' to the context of E0 level one and E0 level two respectively. And we let $(\mathcal{B}^i_{t+1}, X^i_{t+1})$ (resp. $(\mathcal{B}^i_{t'+1}, X^i_{t'+1})$) control the FSM to compute C^i_t (resp. $C^i_{t'}$) at E0 first (resp. second) level for the i-th frame.

Recall from Section 4.3.1 that the initialization of LFSRs at E0 level one by an affine transformation of the effective encryption key $\mathcal{K}$ and public nonce $\mathcal{P}^i$ implies

$$\mathcal{B}^i_t = \mathcal{G}_t(\mathcal{K}) \oplus \mathcal{G}'_t(\mathcal{P}^i), \tag{5.12}$$

where $\mathcal{G}_t, \mathcal{G}'_t$ are public affine transformations (which are dependent on ℓ but omitted from notations for simplicity). Let $Z^i_{t'} = (z^i_{t'}, \ldots, z^i_{t'+\ell-1})$. Then, as detailed in Section 4.3.3, the critical reinitialization flaw in Eq.(4.14) of Bluetooth two-level E0 can be rewritten as

$$\bar{\alpha} \cdot (Z^i_{t'} \oplus \mathcal{L}_{t'}(\mathcal{K}) \oplus \mathcal{L}'_{t'}(\mathcal{P}^i)) = \bigoplus_{j=1}^{4} (\alpha \cdot C^i_{t_j}) \oplus (\bar{\alpha} \cdot C^i_{t'}), \tag{5.13}$$

for any i and α of length ℓ such that $3 \leq \ell \leq 8$, and $t' \in \bigcup_{k=0}^{2}\{8k+1, \ldots, 8k+$

[3] Note that X_t contains the bit c_t^0 as well as the bit c_{t-1}^0.

[4] because c_t^0, c_{t+1}^0 are contained in X_{t+1} already and we can compute $c_{t+2}^0, \ldots, c_{t+\ell-1}^0$ by $\mathcal{B}_{t+1}, X_{t+1}$. Actually, the prerequisite $\alpha_0 = \alpha_{\ell-1} = 1$ on α is to guarantee that knowledge of $\mathcal{B}_{t+1}, X_{t+1}$ is necessary and sufficient to compute $\alpha \cdot C_t$.

$9-\ell\}$, where $t_1, \ldots, t_4$ are functions[5] in terms of t' only, and $C^i_{t_1}, \ldots, C^i_{t_4}$ share no common coordinate, and $\mathcal{L}_{t'}, \mathcal{L}'_{t'}$ are fixed linear functions which can be expressed by t', ℓ from the standard. By definition of h, Eq.(5.13) can be put equivalently as:

$$\bar{\alpha} \cdot (Z^i_{t'} \oplus \mathcal{L}_{t'}(\mathcal{K}) \oplus \mathcal{L}'_{t'}(\mathcal{P}^i)) = \bigoplus_{j=1}^{4} h^{\alpha}_{\mathcal{B}^i_{t_j+1}}(X^i_{t_j+1}) \oplus h^{\bar{\alpha}}_{\mathcal{B}^i_{t'+1}}(X^i_{t'+1}), \qquad (5.14)$$

for any i, α with $3 \leq \ell \leq 8$ and $t' \in \bigcup_{k=0}^{2}\{8k+1, \ldots, 8k+9-\ell\}$.

5.7.2 Correlations Conditioned on Input Weights of FSM

Recall from Remark 3, Section 3.6.2 we know that if $w(B_{t+1})w(B_{t+2})w(B_{t+3}) = 222$ is satisfied, then, we always have

$$c^0_t \oplus c^0_{t+1} \oplus c^0_{t+2} \oplus c^0_{t+3} \oplus c^0_{t+4} = 1. \qquad (5.15)$$

From Section 5.7.1 we know that with $\alpha = (1,1,1,1,1)$ and $\ell = 5$, the function $h^{\alpha}_{\mathcal{B}_{t+1}} : X_{t+1} \mapsto \alpha \cdot C_t$ is well defined for all t, given $\mathcal{B}_{t+1} = B_{t+1}B_{t+2}B_{t+3} \in GF(2)^{12}$. In contrast to the (unconditional) correlation as mentioned in Section 5.2, Eq.(5.15) allows us to deduct a conditional correlation[6] on one-level E0, i.e. the correlation $\Delta(h^{\alpha}_{\mathcal{B}_{t+1}}) = -1$ conditioned on $W(\mathcal{B}_{t+1}) = 222$, where $W(\mathcal{B}_{t+1}) \stackrel{\text{def}}{=} w(B_{t+1})w(B_{t+2}) \cdots w(B_{t+\ell-2})$.

This motivates us to study the general correlation $\Delta(h^{\alpha}_{\mathcal{B}_{t+1}})$ conditioned on $\mathcal{B}_{t+1}$, or more precisely, $W(\mathcal{B}_{t+1})$, when X_{t+1} is uniformly distributed. All the nonzero conditional correlations $\Delta(h^{\alpha}_{\mathcal{B}_{t+1}})$ are shown in Table 5.1 in descending order of the absolute value, where $\#\mathcal{B}_{t+1}$ denotes the cardinality of $\mathcal{B}_{t+1}$ admitting any weight triplet in the group. As the unconditioned correlation $\Delta(h^{\alpha})$ of the bit $\alpha \cdot C_t$ (i.e. $c^0_t \oplus c^0_{t+1} \oplus c^0_{t+2} \oplus c^0_{t+3} \oplus c^0_{t+4}$) always equals the mean value[7] $\mathrm{E}[\Delta(h^{\alpha}_{\mathcal{B}_{t+1}})]$ over the uniformly distributed $\mathcal{B}_{t+1}$, we can use Table 5.1 to verify

[5] Additionally, recall the fact that given t', the relation $t_1 < t_2 < t_3 < t_4$ always holds that satisfies $t_2 - t_1 = t_4 - t_3 = 8$ and $t_3 - t_2 \geq 32$ from Section 4.3.3.

[6] Note that earlier in [53], correlations conditioned on keystream bits (both with and without one LFSR outputs) were well studied for one-level E0, which differ from our conditional correlations and do not fit in the context of two-level E0 if the initial state of E0 is not recovered level by level.

[7] Note that $\mathrm{E}[\Delta(h^{\alpha}_{\mathcal{B}_{t+1}})]$ is computed by an exhaustive search over all possible $X_{t+1} \in GF(2)^4$, $\mathcal{B}_{t+1} \in GF(2)^{12}$ and thus does not depend on t.

$\Delta(h^\alpha) = \Omega(\alpha) = -\frac{25}{256} \stackrel{\text{def}}{=} \lambda$, as proved in Property 5, Section 3.6.2 to be one of the two largest unconditioned correlations up to 27-bit output sequence of the FSM. Let $f_{\mathcal{B}} = h^\alpha_{\mathcal{B}_{t+1}}$ with $\alpha = (1, 1, 1, 1, 1)$. Now, to verify Theorem 6 in Section 5.6 we compute $\mathrm{E}[\mathsf{SEI}(f_{\mathcal{B}})] = \frac{544}{2^{12}} \approx 2^{-2.9}$, which is significantly larger than $\mathsf{SEI}(f) = \lambda^2 \approx 2^{-6.67}$.

Table 5.1: Weight triplets to generate the biased bit $\alpha \cdot C_t$ (i.e. $c^0_t \oplus c^0_{t+1} \oplus c^0_{t+2} \oplus c^0_{t+3} \oplus c^0_{t+4}$) with $\alpha = (1, 1, 1, 1, 1)$ and $\ell = 5$

bias of $\alpha \cdot C_t$ $\Delta(h^\alpha_{\mathcal{B}_{t+1}})$	weight triplet(s) $W(\mathcal{B}_{t+1})$	cardinality $\#\mathcal{B}_{t+1}$
1	$220, 224$	72
-1	222	216
0.5	$120, 124, 210, 214$ $230, 234, 320, 324$	192
-0.5	$122, 212, 322, 232$	576
0.25	$110, 111, 114, 130$ $131, 134, 310, 311$ $314, 330, 331, 334$	384
-0.25	$112, 113, 132, 133$ $312, 313, 332, 333$	640

As another example, consider now $f_{\mathcal{B}} = h^\alpha_{\mathcal{B}_{t+1}}$ with $\alpha = (1, 1, 0, 1)$ and $u = 8, v = 4, r = 1$. Similarly, the conditioned correlation of the corresponding sum $\alpha \cdot C_t$ (i.e. $c^0_t \oplus c^0_{t+1} \oplus c^0_{t+3}$) is shown in Table 5.2. From Table 5.2, we get a quite large $\mathrm{E}[\mathsf{SEI}(f_{\mathcal{B}})] = 2^{-3}$ as well; in contrast, we can check that as already pointed out in Remark 6, Section 5.6, the unconditional correlation $\mathsf{SEI}(f) = 0$ from Table 5.2. Note that on the other hand, from Section 3.6.2 we know that the unconditional correlation $\Delta(h^{\bar{\alpha}}) = \Omega(\bar{\alpha}) = -2^{-4} \stackrel{\text{def}}{=} \lambda'$ is the only second largest unconditioned correlations up to 27-bit output sequence of the FSM. In Table 5.3 we compare conditional correlations with unconditional correlation counterparts on one-level E0 corresponding to those interesting choices of α.

Table 5.2: Weight pairs to generate the biased bit $\alpha \cdot C_t$ (i.e. $c_t^0 \oplus c_{t+1}^0 \oplus c_{t+3}^0$) with $\alpha = (1, 1, 0, 1)$ and $\ell = 4$

bias of $\alpha \cdot C_t$ $\Delta(h^\alpha_{\mathcal{B}_{t+1}})$	weight pairs $W(\mathcal{B}_{t+1})$	cardinality $\#\mathcal{B}_{t+1}$
1	01, 43	8
-1	03, 41	8
0.5	11, 33	32
-0.5	13, 31	32

Table 5.3: Comparison of conditional correlations vs. unconditional correlations on one-level E0

α	$(1,1,0,1)$	$(1,0,1,1)$	$(1,1,1,1,1)$	$(1,0,0,0,0,1)$
SEI(f)	0	2^{-8}	$\approx 2^{-6.7}$	$\approx 2^{-6.7}$
E[SEI(f_B)]	2^{-3}	2^{-4}	$\approx 2^{-2.9}$	$\approx 2^{-2.5}$

5.7.3 Basic Partial Key-recovery Attack

Given the binary vector α (to be determined later) with $3 \le \ell \le 8$, for all $\mathcal{B} \in GF(2)^{4(\ell-2)}$ such that $\Delta(h^\alpha_{\mathcal{B}}) \neq 0$, define the function

$$g^\alpha(\mathcal{B}) = \begin{cases} 0, & \text{if } \Delta(h^\alpha_{\mathcal{B}}) > 0 \\ 1, & \text{if } \Delta(h^\alpha_{\mathcal{B}}) < 0 \end{cases}$$

to estimate the effective value of $h^\alpha_{\mathcal{B}}(X)$ (defined in Section 5.7.1) for some unknown $X \in GF(2)^4$. For a fixed $t' \in \bigcup_{k=0}^{2}\{8k+1, \ldots, 8k+9-|\alpha|\}$, let us guess the subkey $K_1 \stackrel{\text{def}}{=} (\mathcal{G}_{t_1}(\mathcal{K}), \ldots, \mathcal{G}_{t_4}(\mathcal{K}))$ of $16(\ell-2)$ bits by $\widehat{K_1}$ and the one-bit subkey $K_2 \stackrel{\text{def}}{=} \bar{\alpha} \cdot \mathcal{L}_{t'}(\mathcal{K})$ by $\widehat{K_2}$. We set $K = (K_1, K_2)$, $\widehat{K} = (\widehat{K_1}, \widehat{K_2})$. As $\mathcal{P}^i$'s are public, for every frame i, we can use Eq.(5.12) to compute the estimate $\widehat{\mathcal{B}^i_{t_j+1}}$ for $\mathcal{B}^i_{t_j+1}$ for $j = 1, \ldots, 4$ with $\widehat{K_1}$. Denote

$$\begin{aligned} \mathcal{B}^i &= (\mathcal{B}^i_{t_1+1}, \mathcal{B}^i_{t_2+1}, \mathcal{B}^i_{t_3+1}, \mathcal{B}^i_{t_4+1}), \\ \mathcal{X}^i &= (X^i_{t_1+1}, X^i_{t_2+1}, X^i_{t_3+1}, X^i_{t_4+1}, X^i_{t'+1}, \mathcal{B}^i_{t'+1}, \widehat{K}). \end{aligned}$$

Define the probabilistic mapping $\mathcal{F}^{\alpha}_{\mathcal{B}^i}(\mathcal{X}^i)$ to be a truly random bit with uniform distribution for all i such that $\prod_{j=1}^{4}\Delta(h^{\alpha}_{\widehat{\mathcal{B}^i_{t_j+1}}})=0$; otherwise, we let

$$\mathcal{F}^{\alpha}_{\mathcal{B}^i}(\mathcal{X}^i)=\bigoplus_{j=1}^{4}\left(h^{\alpha}_{\mathcal{B}^i_{t_j+1}}(X^i_{t_j+1})\oplus g^{\alpha}(\widehat{\mathcal{B}^i_{t_j+1}})\right)\oplus h^{\bar{\alpha}}(\mathcal{B}^i_{t'+1},X^i_{t'+1}). \tag{5.16}$$

Note that given $\widehat{K_2}$, $\mathcal{F}^{\alpha}_{\mathcal{B}^i}(\mathcal{X}^i)$ is accessible in the latter case as we have

$$\mathcal{F}^{\alpha}_{\mathcal{B}^i}(\mathcal{X}^i)=\bar{\alpha}\cdot\left(Z^i_{t'}\oplus\mathcal{L}'_{t'}(\mathcal{P}^i)\right)\oplus\widehat{K_2}\oplus\bigoplus_{j=1}^{4}g^{\alpha}(\widehat{\mathcal{B}^i_{t_j+1}}),$$

for all i such that $\prod_{j=1}^{4}\Delta(h^{\alpha}_{\widehat{\mathcal{B}^i_{t_j+1}}})\neq 0$ according to Eq.(5.14). With the correct guess $\widehat{K}=K$, Eq.(5.16) reduces to

$$\mathcal{F}^{\alpha}_{\mathcal{B}^i}(\mathcal{X}^i)=\bigoplus_{j=1}^{4}\left(h^{\alpha}_{\mathcal{B}^i_{t_j+1}}(X^i_{t_j+1})\oplus g^{\alpha}(\mathcal{B}^i_{t_j+1})\right)\oplus h^{\bar{\alpha}}(\mathcal{B}^i_{t'+1},X^i_{t'+1}), \tag{5.17}$$

for all i such that $\prod_{j=1}^{4}\Delta(h^{\alpha}_{\mathcal{B}^i_{t_j+1}})\neq 0$. As the right-hand side of Eq.(5.17) only involves the unknown $X^i=(X^i_{t_1+1},X^i_{t_2+1},X^i_{t_3+1},X^i_{t_4+1},X^i_{t'+1},\mathcal{B}^i_{t'+1})$, we denote the mapping in this case by $f^{\alpha}_{\mathcal{B}^i}(X^i)$. With the appropriate choice of α as discussed immediately next in Section 5.7.4, we can have $\mathrm{E}[\mathsf{SEI}(f^{\alpha}_{\mathcal{B}^i})]>0$.

With each wrong guess $\widehat{K}\neq K$, however, we estimate $\mathcal{F}^{\alpha}_{\mathcal{B}^i}(\mathcal{X}^i)$ to be uniformly and independently distributed for all i (i.e. $\mathrm{E}[\mathsf{SEI}(\mathcal{F}^{\alpha}_{\mathcal{B}^i})]=0$). The reason can be explained when we separate those wrong guesses into two cases as follows.

Case 1: $\widehat{K_1}\neq K_1$. Let us check the distribution of $\mathcal{F}^{\alpha}_{\mathcal{B}^i}(\mathcal{X}^i)$ when[8] $\prod_{j=1}^{4}\Delta(h^{\alpha}_{\mathcal{B}^i_{t_j+1}})\neq 0$. Assuming that $\mathcal{P}^i$'s are uniformly and independently distributed, we deduct by Eq.(5.12) that so are $\widehat{\mathcal{B}^i}$'s for every $\widehat{K}$, where $\widehat{\mathcal{B}^i}=(\widehat{\mathcal{B}^i_{t_1+1}},\ldots,\widehat{\mathcal{B}^i_{t_4+1}})$. Hence, we estimate $g^{\alpha}(\widehat{\mathcal{B}^i_{t_j+1}})$ for $j=1,\ldots,4$ are also uniformly and independently distributed. By Eq.(5.16), we complete our justification.

Case 2: $\widehat{K_1}=K_1$ and $\widehat{K_2}\neq K_2$. $\mathcal{F}^{\alpha}_{\mathcal{B}^i}(\mathcal{X}^i)$ is no longer uniformly distributed. But, because we have $\mathcal{F}^{\alpha}_{\mathcal{B}^i}(\mathcal{X}^i)=f^{\alpha}_{\mathcal{B}^i}(X^i)\oplus 1$ for all i such that

[8] By definition of $\mathcal{F}^{\alpha}_{\mathcal{B}^i}$, this is trivial for the cases when $\prod_{j=1}^{4}\Delta(h^{\alpha}_{\mathcal{B}^i_{t_j+1}})=0$.

$\prod_{j=1}^{4} \Delta(h^{\alpha}_{\mathcal{B}^i_{t_j+1}}) \neq 0$, its distribution $D_{\mathcal{F}^{\alpha}_{\mathcal{B}^i}}$ has larger Kullback Leibler distance (see [31]) to $D_{f^{\alpha}_{\mathcal{B}^i}}$ than a uniform distribution does according to [9]. Thus, this case is more favorable to us.

Therefore, we pessimistically approximate $D_{\mathcal{F}^{\alpha}_{\mathcal{B}^i}}$ by a uniform distribution for each wrong guess $\widehat{K} \neq K$. As we are interested in small ℓ for low time complexity, e.g. $\ell < 6$ as explained immediately next, we can assume from this constraint[9] that X^i's are uniformly distributed and that all X^i's, $\mathcal{B}^i$'s are independent. Submitting 2^L sequences of n pairs $(\mathcal{F}^{\alpha}_{\mathcal{B}^i}(\mathcal{X}^i), \widehat{\mathcal{B}^i})$ (for $i = 1, 2, \ldots, n$) to the distinguisher, we can fit in the smart distinguisher of Section 5.4 with $L = 16(\ell - 2) + 1, u = 16(\ell - 2), v = 20 + 4(\ell - 2), r = 1$ and expect it to successfully recover L-bit K with data complexity n sufficiently large as analyzed later. Note that the favorable $L < 64$ necessitates that $\ell < 6$.

5.7.4 Complexity Analysis and Optimization

From Theorem 4 in Section 5.4, the smart distinguisher needs data complexity

$$n = \frac{4L \log 2}{\mathrm{E}\left[\mathsf{SEI}\left(f^{\alpha}_{\mathcal{B}^i}\right)\right]}. \tag{5.18}$$

To compute n, we introduce another probabilistic mapping $f'^{\alpha}_{\mathcal{B}^i}$ similar to $f^{\alpha}_{\mathcal{B}^i}$:

$$f'^{\alpha}_{\mathcal{B}^i}(\mathcal{X}^i) \stackrel{\mathsf{def}}{=} \bigoplus_{j=1}^{4} h^{\alpha}_{\mathcal{B}^i_{t_j+1}}(X^i_{t_j+1}) \oplus h^{\bar{\alpha}}(\mathcal{B}^i_{t'+1}, X^i_{t'+1}). \tag{5.19}$$

Theorem 7 *For all $\mathcal{B}^i = (\mathcal{B}^i_{t_1+1}, \mathcal{B}^i_{t_2+1}, \mathcal{B}^i_{t_3+1}, \mathcal{B}^i_{t_4+1}) \in GF(2)^{16(\ell-2)}$, we always have*

$$\mathit{SEI}(f^{\alpha}_{\mathcal{B}^i}) = \mathit{SEI}(f'^{\alpha}_{\mathcal{B}^i}).$$

Proof. This is trivial for the case where $\prod_{j=1}^{4} \Delta(h^{\alpha}_{\mathcal{B}^i_{t_j+1}}) = 0$, because by definition $D_{f^{\alpha}_{\mathcal{B}^i}}$ is a uniform distribution and so is $D_{f'^{\alpha}_{\mathcal{B}^i}}$ by the famous Piling-up lemma (see [77]). Let us discuss the case where $\prod_{j=1}^{4} \Delta(h^{\alpha}_{\mathcal{B}^i_{t_j+1}}) \neq 0$. In this case we

[9] However, the assumption does not hold for $\ell = 7, 8$: with $\ell = 8$, we know that $X^i_{t_2+1}$ is fixed given $X^i_{t_1+1}$ and $\mathcal{B}^i_{t_1+1}$ as we have $t_2 = t_1 + 8$ from Chapter 4; with $\ell = 7$, two bits of $X^i_{t_2+1}$ are fixed given $X^i_{t_1+1}$ and $\mathcal{B}^i_{t_1+1}$. Similar statements hold concerning $X^i_{t_3+1}, \mathcal{B}^i_{t_3+1}$ and $X^i_{t_4+1}$.

know that given $\mathcal{B}^i$, $\bigoplus_{j=1}^{4} g^{\alpha}(\mathcal{B}^i_{t_j+1})$ is well-defined and it is a fixed value that doesn't depend on the unknown X^i. Consequently, we have

$$\mathsf{SEI}(f^{\alpha}_{\mathcal{B}^i}) = \mathsf{SEI}(f'^{\alpha}_{\mathcal{B}^i} \oplus \text{const.}) = \mathsf{SEI}(f'^{\alpha}_{\mathcal{B}^i}).$$

□

We can use Theorem 7 to compute $\frac{4L\log 2}{n}$ from Eq.(5.18) as

$$\frac{4L\log 2}{n} = \mathrm{E}[\mathsf{SEI}(f^{\alpha}_{\mathcal{B}^i})] = \mathrm{E}[\mathsf{SEI}(f'^{\alpha}_{\mathcal{B}^i})].$$

Next, the independence of $\mathcal{B}^i$'s allows us to apply Piling-up Lemma [77] to continue as follows,

$$\frac{4L\log 2}{n} = \mathrm{E}\left[\mathsf{SEI}(h^{\bar{\alpha}}) \prod_{j=1}^{4} \mathsf{SEI}\left(h^{\alpha}_{\mathcal{B}^i_{t_j+1}}\right)\right] = \mathsf{SEI}(h^{\bar{\alpha}}) \prod_{j=1}^{4} \mathrm{E}\left[\mathsf{SEI}\left(h^{\alpha}_{\mathcal{B}^i_{t_j+1}}\right)\right].$$

Because we know from Section 5.7.2 that $\mathrm{E}[\mathsf{SEI}(h^{\alpha}_{\mathcal{B}^i_{t+1}})]$ does not depend on t and i, we finally have

$$\frac{4L\log 2}{n} = \mathsf{SEI}(h^{\bar{\alpha}}) \cdot \mathrm{E}^4\left[\mathsf{SEI}\left(h^{\alpha}_{\mathcal{B}_{t+1}}\right)\right]. \tag{5.20}$$

As we want to minimize n, according to Eq.(5.18), we would like to find some α $(3 \le |\alpha| < 6)$ such that $\mathrm{E}[\mathsf{SEI}(f^{\alpha}_{\mathcal{B}^i})]$ is large, and above all, strictly positive. In order to have $\mathrm{E}[\mathsf{SEI}(f^{\alpha}_{\mathcal{B}^i})] > 0$, we must have $\mathsf{SEI}(h^{\bar{\alpha}}) > 0$ first, by Eq.(5.20). According to Section 3.6.2, only two aforementioned choices satisfy our predefined prerequisite about α (i.e. both the first and last coordinates of α are one): either $\alpha = (1,1,1,1,1)$ with $\mathsf{SEI}(h^{\bar{\alpha}}) = \lambda^2 \approx 2^{-6.71}$, or $\alpha = (1,1,0,1)$ with $\mathsf{SEI}(h^{\bar{\alpha}}) = \lambda'^2 = 2^{-8}$.

For $\alpha = (1,1,1,1,1)$, we know from Section 5.7.2 that the conditional correlation $\mathrm{E}[\mathsf{SEI}(h^{\alpha}_{\mathcal{B}_{t+1}})]$ on one-level E0 is approximately $\mathrm{E}[\mathsf{SEI}(h^{\alpha}_{\mathcal{B}_{t+1}})] \approx 2^{-2.9}$. So we conclude from Eq.(5.20) that the corresponding conditional correlation $\mathrm{E}[\mathsf{SEI}(f^{\alpha}_{\mathcal{B}^i})]$ on two-level E0 is approximately $\mathrm{E}[\mathsf{SEI}(f^{\alpha}_{\mathcal{B}^i})] \approx 2^{-18.3}$ (in contrast to its unconditional correlation counterpart $\mathsf{SEI}(f^{\alpha}) \approx 2^{-33.5}$). Hence, $n \approx 2^{25.4}$ frames of keystreams generated by the same key $\mathcal{K}$ suffice to recover the $L = 49$-bit subkey K from Eq.(5.18).

Analogously, for $\alpha = (1,1,0,1)$, we have $\mathrm{E}[\mathsf{SEI}(h^{\alpha}_{\mathcal{B}_{t+1}})] = 2^{-3}$ from Section 5.7.2. And it implies that the corresponding conditional correlation $\mathrm{E}[\mathsf{SEI}(f^{\alpha}_{\mathcal{B}^i})]$

on two-level E0 is $\mathrm{E}[\mathsf{SEI}(f^{\alpha}_{\mathcal{B}^i})] = 2^{-20}$ (in contrast to its remarkable unconditional correlation counterpart $\mathsf{SEI}(f^{\alpha}) = 0$ as pointed out in Remark 6 in Section 5.6). It results in $n \approx 2^{26.5}$ frames to recover $L = 33$-bit subkey. Table 5.4 compares conditional correlations $\mathrm{E}[\mathsf{SEI}(f^{\alpha}_{\mathcal{B}^i})]$ with the unconditional correlation counterparts $\mathsf{SEI}(f^{\alpha})$ on two-level E0 corresponding to the interesting choices of α (the last row lists the size of the involved partial input of conditional correlations). From the table we can check that $\alpha = (1, 1, 0, 1)$ is the best choice for our attack with respect to time and data complexities.

Table 5.4: Comparison of conditional correlation vs. unconditional correlation on two-level E0

α	$(1,1,0,1)$	$(1,0,1,1)$	$(1,1,1,1,1)$	$(1,0,0,0,0,1)$
$\mathsf{SEI}(f^{\alpha})$	0	0	$\approx 2^{-33.5}$	$\approx 2^{-33.5}$
$\mathrm{E}[\mathsf{SEI}(f^{\alpha}_{\mathcal{B}^i})]$	2^{-20}	0	$\approx 2^{-18.3}$	$\approx 2^{-16.7}$
$\log_2 \#(\mathcal{B}^i)$	33	33	49	65

Let us discuss the time complexity of the attack now. For all $J = (J_1, J_2) \in GF(2)^{L-1} \times GF(2)$, and let $J_1 = (J_{1,1}, \ldots, J_{1,4})$ where $J_{1,i} \in GF(2)^{4(\ell-2)}$, we define $\mathcal{H}, \mathcal{H}'$:

$$\mathcal{H}(J) = \sum_{i=1}^{n} \mathbf{1}_{\mathcal{G}'_{t_1}(\mathcal{P}^i),\ldots,\mathcal{G}'_{t_4}(\mathcal{P}^i)=J_1 \text{ and } \bar{\alpha}\cdot(Z^i_{t'} \oplus \mathcal{L}'_{t'}(\mathcal{P}^i))=J_2},$$

$$\mathcal{H}'(J) = \begin{cases} 0, & \text{if } \prod_{i=1}^{4} \Delta(h^{\alpha}_{J_{1,i}}) = 0 \\ \log 2^r \cdot D_{J_1}\left(J_2 \oplus \bigoplus_{i=1}^{4} g^{\alpha}\left(J_{1,i}\right)\right), & \text{otherwise} \end{cases}$$

where $D_{J_1} = D_{h^{\alpha}_{J_{1,1}}} \otimes D_{h^{\alpha}_{J_{1,2}}} \otimes D_{h^{\alpha}_{J_{1,3}}} \otimes D_{h^{\alpha}_{J_{1,4}}}$. Let

$$\mathcal{H}''(K) \stackrel{\text{def}}{=} \widehat{\mathcal{H}}(K) \cdot \widehat{\mathcal{H}'}(K).$$

By Theorem 5 in Section 5.5, we have

$$G(K) = \frac{1}{2^L}\widehat{\mathcal{H}''}(K).$$

This means that after precomputing $\widehat{\mathcal{H}'}$ in time $O(L \cdot 2^L)$, our partial key-recovery attack would be dominated by twice FWT, i.e. computing $\widehat{\mathcal{H}}, \widehat{\mathcal{H}''}$ with time

$O(L \cdot 2^{L+1})$. Algorithm 7 illustrates the above basic partial key-recovery attack. Note that without the optimization technique of Theorem 5, the deterministic smart distinguisher has to perform $O(n \cdot 2^L)$ operations otherwise, which makes our attack impractical.

Algorithm 7 The basic partial key-recovery attack on two-level E0

Parameters:
1: $\alpha, t', t_1, t_2, t_3, t_4, L$
2: n set by Eq.(5.20)
Inputs:
3: $\mathcal{P}^i$ for $i = 1, 2, \ldots, n$
4: $Z^i_{t'}$ for $i = 1, 2, \ldots, n$
Preprocessing:
5: compute $H', \widehat{H'}$
Processing:
6: compute $H, \widehat{H}$
7: compute $H'' = \widehat{H} \cdot \widehat{H'}$ and $\widehat{H''}$
8: output K with the maximum $\widehat{H''}(K)$

5.7.5 Equivalent Key Candidates

Taking a closer look at Table 5.2, we discovered a special property

$$\Delta(h^{\alpha}_{B_{t+1}B_{t+2}}) \equiv \Delta(h^{\alpha}_{\overline{B}_{t+1}\overline{B}_{t+2}}) \equiv -\Delta(h^{\alpha}_{\overline{B}_{t+1}B_{t+2}}) \equiv -\Delta(h^{\alpha}_{B_{t+1}\overline{B}_{t+2}})$$

for all $\mathcal{B}_{t+1} = B_{t+1}B_{t+2} \in GF(2)^8$ with $\alpha = (1, 1, 0, 1)$, where the bar operator denotes the bitwise complement of the 4-bit binary vector. This means that for our 33-bit partial key-recovery attack, we always have $4^4 = 256$ equivalent key candidates[10]. Recall that in Section 5.7.3 we have the 33-bit key $K = (K_1, K_2)$, with $K_1 = (\mathcal{G}_{t_1}(\mathcal{K}), \ldots, \mathcal{G}_{t_4}(\mathcal{K}))$. For simplicity, we let $K_{1,i} = \mathcal{G}_{t_i}(\mathcal{K})$. Define the following 8-bit masks (in hexadecimal):

$$\mathsf{mask}_0 = 0x00,\ \mathsf{mask}_1 = 0xff,\ \mathsf{mask}_2 = 0x0f,\ \mathsf{mask}_3 = 0xf0.$$

Then for any K, we can replace $K_{1,i}$ by $K_{1,i} \oplus \mathsf{mask}_j$ for any $i = 1, 2, \ldots, 4$ and $j \in \{0, 1, 2, 3\}$ and replace K_2 by $K_2 \oplus \lceil \frac{j}{2} \rceil$. Denote this set containing $4^4 = 2^8$

[10] The term "equivalent key candidate" is exclusively used for our attack, which does not mean that they are equivalent keys for the Bluetooth encryption.

elements by $\langle K \rangle$. We can easily verify that the Walsh coefficients $\widehat{\mathcal{H}''}$ of the element in the set equals by following the definition of convolution between $\mathcal{H}$ and $\mathcal{H}'$:

$$\mathcal{H} \otimes \mathcal{H}'(K) = \sum_{K'} \mathcal{H}(K')\mathcal{H}'(K \oplus K').$$

Since if $R \in \langle K \rangle$ then $R \oplus K' \in \langle K \oplus K' \rangle$ for all K'. And $\mathcal{H}'$ maps all the elements of the same set to the same value from Section 5.7.4, we conclude that the set defined above form an equivalent class of the candidate keys. Thus, we have 2^8 equivalent 33-bit keys. This helps to decrease the computation time on $\widehat{H''}$ (see Section 3.3.4) from $33 \times 2^{33} \approx 2^{38}$ to $25 \times 2^{25} \approx 2^{30}$. In total we have the running time $2^{38} + 2^{30} \approx 2^{38}$ for Algorithm 7.

5.7.6 Experiments

We have implemented the full Algorithm 7 with $\alpha = (1, 1, 0, 1), t' = 1, n = 2^{26}$ frames (slightly less than the theoretical estimate $2^{26.5}$) on the Linux platform, 2.4G CPU, 2G RAM, 128GB hard disk with the external data transfer rate[11] 32MB/s between the hard disk and PC's main memory. It turned out that after one run of a 37-hour precomputation (i.e. Line 5 in Algorithm 7 which stores a 64GB table in the hard disk), of all the 30 runs tested so far, our attack never fails to successfully recover the right 25-bit key in about 19 hours. Computing $H, \widehat{H}, H'', \widehat{H''}$ takes time 27 minutes, 18 hours, 45 minutes and 20 seconds respectively. The running time is dominated by FWT[12] $\widehat{H}$, which only takes a negligible portion of CPU time and depends dominantly on the performance of the hardware, i.e. the external data transfer rate between the hard disk and PC's main memory.

5.7.7 Advanced Partial Key-recovery Attack

Having studied how to apply the smart distinguisher of Section 5.4 with $r = 1$ (namely the uni-bias-based approach) for an attack to two-level E0 previously, now we wonder the possibility of improvement based on multi-biases, inspired by the traditional multi-bias-based distinguisher in Section 3.3.4.

For the reason of low time complexity of the attack, we still focus on the analysis of 4-bit biases; additionally, we restrict ourselves to bi-biases analysis (i.e.

[11] The external data transfer rate 32MB/s of the hard disk is common nowadays.

[12] The result is stored in a 32GB table in the hard disk.

$r = 2$) to simplify the presentation, which will be shown later to be optimal. Let $\alpha = (\alpha_1, \alpha_2)$, where α_1 is fixed to $(1, 1, 0, 1)$ and α_2 with length $\ell_2 \stackrel{\text{def}}{=} |\alpha_2| = 4$ remains to be determined later such that the data complexity is lowered when we analyze the characteristics of bi-biases simultaneously for each frame instead of conducting the previous uni-bias-based analysis.

Recall that $g^{\alpha_1}(\mathcal{B}) : GF(2)^8 \rightarrow GF(2)$ in Section 5.7.3 was defined to be the most likely bit of $h^{\alpha_1}_{\mathcal{B}}(X)$ for a uniformly distributed $X \in GF(2)^4$ if it exists (i.e. $\Delta(h^{\alpha}_{\mathcal{B}}) \neq 0$). We extend $g^{\alpha_1}(\mathcal{B}) : GF(2)^8 \rightarrow GF(2)$ to $g^{\alpha}(\mathcal{B}) : GF(2)^8 \rightarrow GF(2)^2$ over all $\mathcal{B} \in GF(2)^8$ such that $\Delta(h^{\alpha_1}_{\mathcal{B}}) \neq 0$, and let $g^{\alpha}(\mathcal{B})$ be the most likely 2-bit binary vector $\tau = (\tau_1, \tau_2)$. Note that we can always easily determine the first bit τ_1 because of the assumption $\Delta(h^{\alpha_1}_{\mathcal{B}}) \neq 0$; with regards to determining the second bit τ_2 in case that a tie occurs, we just let τ_2 be a uniformly distributed bit. Let

$$\begin{aligned} h^{\alpha}_{\mathcal{B}}(X) &= (h^{\alpha_1}_{\mathcal{B}}(X), h^{\alpha_2}_{\mathcal{B}}(X)), \\ h^{\bar{\alpha}}(\mathcal{B}, X) &= (h^{\bar{\alpha}_1}(\mathcal{B}, X), h^{\bar{\alpha}_2}(\mathcal{B}, X)). \end{aligned}$$

Note that $h^{\alpha}_{\mathcal{B}}(X)$ outputs the two bits which are generated by the same unknown X given $\mathcal{B}$; by contrast, $h^{\bar{\alpha}}(\mathcal{B}, X)$ outputs the two bits which are generated by the unknown X and $\mathcal{B}$. We can extend $\mathcal{F}^{\alpha_1}_{\mathcal{B}^i}(\mathcal{X}^i)$ in Eq.(5.16) to $\mathcal{F}^{\alpha}_{\mathcal{B}^i}(\mathcal{X}^i)$ by letting

$$\begin{aligned} \mathcal{F}^{\alpha}_{\mathcal{B}^i}(\mathcal{X}^i) = \Bigg(& \bigoplus_{j=1}^{4} h^{\alpha_1}_{\mathcal{B}^i_{t_j+1}}(X^i_{t_j+1}) \oplus h^{\bar{\alpha_1}}(\mathcal{B}^i_{t'+1}, X^i_{t'+1}), \\ & \bigoplus_{j=1}^{4} h^{\alpha_2}_{\mathcal{B}^i_{t_j+1}}(X^i_{t_j+1}) \oplus h^{\bar{\alpha_2}}(\mathcal{B}^i_{t'+1}, X^i_{t'+1}) \Bigg) \oplus g^{\alpha}(\widehat{\mathcal{B}^i_{t_j+1}}), \end{aligned}$$

if $\prod_{j=1}^{4} \Delta(h^{\alpha_1}_{\widehat{\mathcal{B}^i_{t_j+1}}}) \neq 0$; otherwise, we let it be a uniformly distributed two-bit vector. Similarly, we denote $\mathcal{F}^{\alpha}_{\mathcal{B}^i}(\mathcal{X}^i)$ corresponding to the correct guess by $f^{\alpha}_{\mathcal{B}^i}$.

It is easy to verify the assumption holds to apply Section 5.4 that says $D_{\mathcal{F}^{\alpha}_{\mathcal{B}^i}}$ can still be approximated by a uniform distribution for each wrong guess on the key $\widehat{K} \neq K$. Moreover, by introducing the extended $f'^{\alpha}_{\mathcal{B}^i}$ from $f'^{\alpha_1}_{\mathcal{B}^i}$ in

Eq.(5.19) as

$$\begin{aligned} f'^{\alpha}_{\mathcal{B}^i}(\mathcal{X}^i) &\stackrel{\text{def}}{=} \left(f'^{\alpha_1}_{\mathcal{B}^i}(\mathcal{X}^i), f'^{\alpha_2}_{\mathcal{B}^i}(\mathcal{X}^i)\right) \\ &= \left(\bigoplus_{j=1}^{4} h^{\alpha_1}_{\mathcal{B}^i_{t_j+1}}(X^i_{t_j+1}) \oplus h^{\bar{\alpha}_1}(\mathcal{B}^i_{t'+1}, X^i_{t'+1}),\right. \\ &\quad \left.\bigoplus_{j=1}^{4} h^{\alpha_2}_{\mathcal{B}^i_{t_j+1}}(X^i_{t_j+1}) \oplus h^{\bar{\alpha}_2}(\mathcal{B}^i_{t'+1}, X^i_{t'+1})\right). \end{aligned}$$

Theorem 7 in Section 5.7.4 can be similarly extended to the following: for all $\mathcal{B}^i = (\mathcal{B}^i_{t_1+1}, \mathcal{B}^i_{t_2+1}, \mathcal{B}^i_{t_3+1}, \mathcal{B}^i_{t_4+1}) \in GF(2)^{32}$, we always have

$$\mathsf{SEI}(f^{\alpha}_{\mathcal{B}^i}) = \mathsf{SEI}(f'^{\alpha}_{\mathcal{B}^i}).$$

Similar computation yields the same formula for data complexity we need as in Eq.(5.20)

$$\frac{4L\log 2}{n} = \mathsf{SEI}(h^{\bar{\alpha}}) \cdot \mathrm{E}^4\left[\mathsf{SEI}\left(h^{\alpha}_{\mathcal{B}_{t+1}}\right)\right].$$

Experimental result shows that with $\alpha_1 = (1,1,0,1), \alpha_2 = (1,0,1,1)$, we achieve optimum $\mathsf{SEI}(h^{\alpha}_{\mathcal{B}_{t+1}}) \approx 2^{-2.415}$ (in comparison to $\mathsf{SEI}(h^{\alpha_1}_{\mathcal{B}_{t+1}}) = 2^{-3}$ in Section 5.7.2), though $\mathsf{SEI}(h^{\bar{\alpha}})$ always equals $\mathsf{SEI}(h^{\bar{\alpha_1}})$ regardless of the choice of α_2; additionally, $\mathsf{SEI}(h^{\bar{\alpha}}) \equiv 0$ if $\alpha_1, \alpha_2 \neq (1,1,0,1)$. Therefore, we have the minimum data complexity $n \approx 2^{23.8}$ (in comparison to $n \approx 2^{26.5}$ in Section 5.7.4) frames with the same time complexity.

5.7.8 Full Attack

Once we recover the first $(33-8) = 25$-bit subkey, we just increment (or decrement) t' by one and use the knowledge of those subkey bits to reiterate Algorithm 7 to recover more key bits similarly as was done in Chapter 4. Since only 17 new key bits are involved, which reduce to the 13-bit equivalent key, it is much faster to recover those key bits. Finally, we perform an exhaustive search over the equivalent key candidates in negligible time, whose total number is upper bounded by $2^{\frac{8|\mathcal{K}|}{32}} = 2^{\frac{|\mathcal{K}|}{4}}$. The final complexity of the complete key-recovery attack is bounded by one run of Algorithm 7, i.e. $O(2^{38})$. Table 5.5 compares our attacks with the best known attacks [46,47,53] on two-level E0 (for effective key size $|\mathcal{K}| = 128$) as well as our previous attack in Chapter 4 using unconditional correlations, where † means that the frame number is not critical for the

attack [47]. Since a maximum of 2^{26} frames are available and the memory or time complexity 2^{50} is beyond the current technology, our advanced attack is clearly the *only* practical attack so far. Note that with $|\mathcal{K}| = 64$, Bluetooth key loading at E0 level one makes the bits of the subkey K linearly independent for all $t' \in \bigcup_{k=0}^{2}\{8k+1, \ldots, 8k+5\}$. Therefore, the attack complexities remain to be on the same order.

Table 5.5: Comparison of our attacks with the best attacks on two-level E0 for $|\mathcal{K}| = 128$

attacks		precomputation	time	frames	data	memory
Fluhrer-Lucks	[47]	-	2^{73}	†	2^{43}	2^{51}
Golić et al.	[53]	2^{80}	2^{70}	45	2^{17}	2^{80}
Fluhrer	[46]	2^{80}	2^{65}	2	$2^{12.4}$	2^{80}
Former attack	Chap. 4	-	2^{56}	2^{35}	$2^{39.6}$	2^{35}
Attacks in this chapter	basic	2^{38}	2^{38}	$2^{26.5}$	$2^{31.1}$	2^{33}
	advanced	2^{38}	2^{38}	$2^{23.8}$	$2^{28.4}$	2^{33}

5.8 Summary

In this chapter, we have generalized the concept of conditional correlations in [2, 71, 73] to study conditional correlation attacks against stream ciphers and other cryptosystems, in case the computation of the output allows for side information related to correlations conditioned on the input (e.g. see [7]). A general framework has been developed for smart distinguishers, which exploit those generalized conditional correlations. In particular, based on the theory of the traditional distinguisher [9] we derive the number of samples necessary for a smart distinguisher to succeed. It is demonstrated that the generalized conditional correlation is no smaller than the unconditional correlation. Consequently, the smart distinguisher improves on the traditional basic distinguisher (in the worst case the smart distinguisher degrades into the traditional one); the smart distinguisher could be efficient even if no ordinary correlations exist. As an application of our generalized conditional correlations, a conditional correlation attack on the two-level Bluetooth E0 is developed and optimized. Whereas our previous analysis in Chapter 4 was based on a traditional distinguishing attack using the

strongest (unconditional) 5-bit correlation, we have successfully demonstrated the superiority of our attack over the attack in Chapter 4 by showing a best attack using 4-bit conditional correlations, which are not suitable for the attack in Chapter 4 as the corresponding ordinary correlations are all *zeros*. Our best attack fully recovers the original encryption key using the first 24 bits of $2^{23.8}$ frames and with 2^{38} computations. Compared with all existing attacks [46,47,53,70,99] as well as our attack in the previous chapter, this is clearly the fastest and *only* practical known-plaintext attack on Bluetooth encryption so far.

Chapter 6

Conclusion

In this dissertation, we extend Bluetooth E0 stream cipher to a class of dedicated stream ciphers, i.e. the E0-like combiner with memory used with one-level or two-level initialization scheme. In-depth studies are done to investigate the possibility of one of the mainstream attacks on stream ciphers (i.e. correlation attacks) with focus on the deterministic distinguisher. We establish a design criterion for the core stream cipher to resist our simple correlation attacks. As one-level initialization scheme turns out to be less strong protection for the core stream cipher than the two-level initialization scheme, the former is not recommended for use in the dedicated stream ciphers. The case study yields the fastest and only practical known-plaintext attack on Bluetooth encryption.

It is well known that the Maximum Likelihood Decoding problem for general linear codes is NP-complete. Hence, except the original version of basic correlation attacks, the subsequent correlation attacks (called as fast correlation attacks) tend to solve the general problem (i.e. codes with large dimension) by various probabilistic algorithms. Prior to our work, it remained an unsolved problem to decode the linear code with not so large dimension but very large length where the naive exhaustive decoding is still impossible. As we showed that the complexity of the MLD problem grows linear in the code length, it makes the deterministic distinguisher appealing in those situations where exhaustive search is within the critical computing point $O(2^{38}) \sim O(2^{40})$ of today's modern computer. Moreover, the simplicity of the deterministic distinguisher also allows to easily judge the performance comparison between (usual) correlation attacks and conditional correlation attacks, which was unknown before.

As a consequence, it remains an interesting open work to efficiently combine the two approaches (deterministic and probabilistic distinguishers) into a hybrid

correlation attack, especially in the case of comparatively young conditional correlation attacks; the design criterion should be updated accordingly. Meanwhile, it is an imperative question that whether or not a secure core stream cipher is sufficient to make a secure dedicated stream cipher (with two-level initialization scheme).

Appendix A

Linear Feedback Shift Register

We briefly review the basics of Linear Feedback Shift Register (LFSR) from [85].

An LFSR of length L with the connection polynomial $p(x) = 1 + \sum_{i=1}^{L} p_i x^i$ (where $p_i \in \{0, 1\}$) consists of L stages (or delay elements) $s_0, \ldots, s_{L-1} \in \{0, 1\}$. Note that the reciprocal polynomial of the connection polynomial $p(x)$ is called the feedback polynomial of the LFSR. During each clock cycle it performs the following:

1. outputs the bit s_0;
2. compute the feedback bit $\bigoplus_{i=1}^{L} p_i \cdot s_{L-i}$;
3. let $s_i = s_{i+1}$ for $i = 0, \ldots, L-2$;
4. store the precomputed feedback bit to s_{L-1}.

And the initial values of $(s_{L-1}, \ldots, s_0)$ is called the initial state of the LFSR. If $p(x)$ is a primitive polynomial of degree L, then the LFSR is called a maximum-length LFSR and the output of a maximum-length LFSR with nonzero initial state is called an m-sequence.

Property 7 *Each nonzero initial state of a maximum-length LFSR with length L produces an output sequence with maximum possible period $2^L - 1$.*

For example, an LFSR of length 4 with the connection polynomial $p(x) = x^4 + x + 1$ (which is a primitive polynomial) is a maximum-length LFSR. And thus, with any of the 15 nonzero initial states, the LFSR output sequence has a period of 15. In Table A.1, we list the state transition of the LFSR for $t = 0, \ldots, 15$

Table A.1: The state transition of an example maximum-length LFSR with length 4 for $t = 0, \ldots, 15$

t	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
s_3	1	1	1	1	0	1	0	1	1	0	0	1	0	0	0	1
s_2	0	1	1	1	1	0	1	0	1	1	0	0	1	0	0	0
s_1	0	0	1	1	1	1	0	1	0	1	1	0	0	1	0	0
s_0	0	0	0	1	1	1	1	0	1	0	1	1	0	0	1	0

starting from the initial state $(1, 0, 0, 0)$. One can verify that the state of the LFSR at $t = 15$ is identical to the one at $t = 0$.

For a more comprehensive survey, we refer to [56, 96].

List of Tables

List of Figures

List of Algorithms

Bibliography

[1] Specification of the A5/3 encryption algorithms for GSM and EDGE, and the GEA3 encryption algorithm for GPRS. available on-line at `http://www.gsmworld.com/using/algorithms/index.shtml`.

[2] Ross Anderson. Searching for the optimum correlation attack. In B. Preneel, editor, *Fast Software Encryption'94*, volume 1008 of *Lecture Notes in Computer Science*, pages 137–143. Springer-Verlag, 1994.

[3] Frederik Armknecht. Improving fast algebraic attacks. In B. Roy and W. Meier, editors, *Fast Software Encryption2004*, volume 3017 of *Lecture Notes in Computer Science*, pages 65–82. Springer-Verlag, 2004.

[4] Frederik Armknecht and Matthias Krause. Algebraic attacks on combiners with memory. In D. Boneh, editor, *Advances in Cryptology - CRYPTO2003*, volume 2729 of *Lecture Notes in Computer Science*, pages 162–175. Springer-Verlag, 2003.

[5] Frederik Armknecht, Matthias Krause, and Dirk Stegemann. Design principles for combiners with memory. In S. Maitra, C. E. V. Madhavan, and R. Venkatesan, editors, *to appear in the proceedings of Progress in Cryptology - INDOCRYPT2005*, Lecture Notes in Computer Science. Springer-Verlag.

[6] Frederik Armknecht, Joseph Lano, and Bart Preneel. Extending the resynchronization attack. In H. Handschuh and A. Hasan, editors, *Selected Areas in Cryptography2004*, volume 3357 of *Lecture Notes in Computer Science*, pages 19–38. Springer-Verlag, 2005.

[7] Frederik Armknecht and Willi Meier. Fault attacks on combiners with memory. In B. Preneel and S. Tavares, editors, *Selected Areas in Cryptography*

2005, Lecture Notes in Computer Science, pages 38–53. Springer-Verlag, 2006.

[8] Steve Babbage. Improved exhaustive search attacks on stream ciphers. *European Convention on Security and Detection, IEE conference publication, No. 408*, pages 161–166, May 1995.

[9] Thomas Baignères, Pascal Junod, and Serge Vaudenay. How far can we go beyond linear cryptanalysis? In P. J Lee, editor, *Advances in Cryptology - ASIACRYPT2004*, volume 3329 of *Lecture Notes in Computer Science*, pages 432–450. Springer-Verlag, 2004.

[10] Elad Barkan and Eli Biham. Conditional estimators: an effective attack on A5/1. In B. Preneel and S. Tavares, editors, *Selected Areas in Cryptography 2005*, Lecture Notes in Computer Science, pages 1–18. Springer-Verlag, 2006.

[11] Elad Barkan, Eli Biham, and Nathan Keller. Instant ciphertext-only cryptanalysis of GSM encrypted communication. In D. Boneh, editor, *Advances in Cryptology - CRYPTO2003*, volume 2729 of *Lecture Notes in Computer Science*, pages 600–616. Springer-Verlag, 2003.

[12] Elwyn R. Berlekamp, Robert J. McEliece, and Henk C. A. Van Tilborg. On the inherent intractability of certain coding problems. *IEEE Transactions on Information Theory*, IT-24(3):384–386, May 1978.

[13] Eli Biham. New types of cryptanalytic attacks using related keys. In T. Helleseth, editor, *Advances in Cryptology - EUROCRYPT'93*, volume 765 of *Lecture Notes in Computer Science*, pages 398–409. Springer-Verlag, 1994.

[14] Eli Biham. New types of cryptanalytic attacks using related keys. *Journal of Cryptology*, 7:229–246, 1994.

[15] Alex Biryukov. Block ciphers and stream ciphers: The state of the art, 2004. available on-line at `http://eprint.iacr.org/2004/094`.

[16] Alex Biryukov and Adi Shamir. Cryptanalytic time/memory/data tradeoffs for stream ciphers. In T. Okamoto, editor, *Advances in Cryptology - ASIACRYPT2000*, volume 1976 of *Lecture Notes in Computer Science*, pages 1–13. Springer-Verlag, 2000.

[17] Bluetooth specification (version 2.0 + EDR), Nov. 2004. available on-line at `http://www.bluetooth.org`.

[18] Anne Canteaut and Florent Chabaud. A new algorithm for finding minimum-weight words in a linear code: Application to McEliece's cryptosystem and to narrow-sense BCH codes of length 511. *IEEE Transactions on Information Theory*, 44(1):367–378, Jan. 1998.

[19] Anne Canteaut and Eric Filiol. Ciphertext only reconstruction of stream ciphers based on combination generators. In B. Schneier, editor, *Fast Software Encryption2000*, volume 1978 of *Lecture Notes in Computer Science*, pages 165–180. Springer-Verlag, 2001.

[20] Anne Canteaut and Michaël Trabbia. Improved fast correlation attacks using parity-check equations of weight 4 and 5. In B. Preneel, editor, *Advances in Cryptology - EUROCRYPT2000*, volume 1807 of *Lecture Notes in Computer Science*, pages 573–588. Springer-Verlag, 2000.

[21] Claude Carlet. Improving the algebraic immunity of resilient and nonlinear functions and constructing bent functions, 2004. available on-line at `http://eprint.iacr.org/2004/276`.

[22] Vladimir Chepyzhov and Ben Smeets. On a fast correlation attack on certain stream ciphers. In D. W. Davies, editor, *Advances in Cryptology - EUROCRYPT'91*, volume 547 of *Lecture Notes in Computer Science*, pages 176–185. Springer-Verlag, 1991.

[23] Vladimir V. Chepyzhov, Thomas Johansson, and Ben Smeets. A simple algorithm for fast correlation attacks on stream ciphers. In B. Schneier, editor, *Fast Software Encryption2000*, volume 1978 of *Lecture Notes in Computer Science*, pages 181–195. Springer-Verlag, 2001.

[24] Philippe Chose, Antoine Joux, and Michel Mitton. Fast correlation attacks: An algorithmic point of view. In L. R. Knudsen, editor, *Advances in Cryptology - EUROCRYPT2002*, volume 2332 of *Lecture Notes in Computer Science*, pages 209–221. Springer-Verlag, 2002.

[25] Andrew Clark, Jovan Dj. Golić, and Ed Dawson. A comparison of fast correlation attaks. In D. Gollmann, editor, *Fast Software Encryption'96*, volume 1039 of *Lecture Notes in Computer Science*, pages 145–157. Springer-Verlag, 1996.

[26] Nicolas Courtois, Alexander Klimov, Jacques Patarin, and Adi Shamir. Efficient algorithms for solving overdefined systems of multivariate polynomial equations. In B. Preneel, editor, *Advances in Cryptology - EUROCRYPT2000*, volume 1807 of *Lecture Notes in Computer Science*, pages 392–407. Springer-Verlag, 2000.

[27] Nicolas T. Courtois. Higher order correlation attacks, XL algorithm and cryptanalysis of Toyocrypt. In P. J. Lee and C. H. Lim, editors, *Information Security and Cryptology - ICISC 2002*, volume 2587 of *Lecture Notes in Computer Science*, pages 182–199. Springer-Verlag, 2002.

[28] Nicolas T. Courtois. Fast algebraic attacks on stream ciphers with linear feedback. In D. Boneh, editor, *Advances in Cryptology - CRYPTO2003*, volume 2729 of *Lecture Notes in Computer Science*, pages 176–194. Springer-Verlag, 2003.

[29] Nicolas T. Courtois and Willi Meier. Algebraic attacks on stream ciphers with linear feedback. In E. Biham, editor, *Advances in Cryptology - EUROCRYPT2003*, volume 2656 of *Lecture Notes in Computer Science*, pages 345–359. Springer-Verlag, 2003.

[30] Nicolas T. Courtois and Josef Pieprzyk. Cryptanalysis of block ciphers with overdefined systems of equations. In Y. Zheng, editor, *Advances in Cryptology - ASIACRYPT2002*, volume 2501 of *Lecture Notes in Computer Science*, pages 267–287. Springer-Verlag, 2002.

[31] Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory*. Wiley, 1991.

[32] Joan Daemen, René Govaerts, and Joos Vandewalle. Resynchronization weaknesses in synchronous stream ciphers. In T. Helleseth, editor, *Advances in Cryptology - EUROCRYPT'93*, volume 765 of *Lecture Notes in Computer Science*, pages 159–167. Springer-Verlag, 1994.

[33] Deepak K. Dalai, Kishan C. Gupta, and Subhamoy Maitra. Cryptographically significant boolean functions: Construction and analysis in terms of algebraic immunity. In H. Gilbert and H. Handschuh, editors, *Fast Software Encryption2005*, volume 3557 of *Lecture Notes in Computer Science*, pages 98–111. Springer-Verlag, 2005.

[34] Magnus Daum. Narrow t-functions. In H. Gilbert and H. Handschuh, editors, *Fast Software Encryption2005*, volume 3557 of *Lecture Notes in Computer Science*, pages 50–67. Springer-Verlag, 2005.

[35] Alexander W. Dent and Chris J. Mitchell. *User's Guide to Cryptography and Standards*. Artech House, 2005.

[36] Whitfield Diffie and Martin E. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, IT-22(6):644–654, Nov. 1976.

[37] Cunsheng Ding, Guozhen Xiao, and Weijuan Shan. *The Stability Theory of Stream Ciphers*, volume 561 of *Lecture Notes in Computer Science*. Springer-Verlag, 1991.

[38] ECRYPT - European Network of Excellence for Cryptology. available on-line at `http://www.ecrypt.eu.org`.

[39] Patrik Ekdahl. *On LFSR Based Stream Ciphers: Analysis and Design*. PhD thesis, Lund Univ., Nov. 2003.

[40] Patrik Ekdahl and Thomas Johansson. Some results on correlations in the Bluetooth stream cipher. In *Proceedings of the 10th Joint Conference on Communications and Coding, Austria*, 2000.

[41] Patrik Ekdahl and Thomas Johansson. Another attack on A5/1. *IEEE Transactions on Information Theory*, 49(1):284–289, Jan. 2003.

[42] eSTREAM - the ECRYPT Stream Cipher Project. available on-line at `http://www.ecrypt.eu.org/stream`.

[43] Jean-Charles Faugère and Gwénolé Ars. An algebraic cryptanalysis of non-linear filter generators using Gröbner bases. Technical Report 4739, INRIA, 2003.

[44] Eric Filiol and Caroline Fontaine. Highly nonlinear balanced boolean functions with a good correlation-immunity. In K. Nyberg, editor, *Advances in Cryptology - EUROCRYPT'98*, volume 1403 of *Lecture Notes in Computer Science*, pages 475–488. Springer-Verlag, 1998.

[45] FIPS 140-1, Security Requirements for Cryptographic Modules, Federal Information Processing Standards Publication 140-1. U.S. Department of

Commerce/N.I.S.T., National Technical Information Service, Springfield, Virginia, 1994.

[46] Scott Fluhrer. Improved key recovery of level 1 of the Bluetooth encryption system, 2002. available on-line at `http://eprint.iacr.org/2002/068`.

[47] Scott Fluhrer and Stefan Lucks. Analysis of the E0 encryption system. In S. Vaudenay and A. Youssef, editors, *Selected Areas in Cryptography 2001*, volume 2259 of *Lecture Notes in Computer Science*, pages 38–48. Springer-Verlag, 2002.

[48] Scott Fluhrer, Itsik Mantin, and Adi Shamir. Weakness in the key scheduling algorithm of RC4. In S. Vaudenay and A. Youssef, editors, *Selected Areas in Cryptography 2001*, volume 2259 of *Lecture Notes in Computer Science*, pages 1–24. Springer-Verlag, 2002.

[49] Réjane Forré. A fast correlation attack on nonlinearly feedforward filtered shift-register sequences. In J. J. Quisquater and J. Vandewalle, editors, *Advances in Cryptology - EUROCRYPT'89*, volume 434 of *Lecture Notes in Computer Science*, pages 586–595. Springer-Verlag, 1990.

[50] Michael R. Garey and David S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. Freeman and company, 2000.

[51] Jovan Dj. Golić. Correlation properties of a general binary combiner with memory. *Journal of Cryptology*, 9:111–126, 1996.

[52] Jovan Dj. Golić. On the security of nonlinear filter generators. In D. Gollmann, editor, *Fast Software Encryption'96*, volume 1039 of *Lecture Notes in Computer Science*, pages 173–188. Springer-Verlag, 1996.

[53] Jovan Dj. Golić, Vittorio Bagini, and Guglielmo Morgari. Linear cryptanalysis of Bluetooth stream cipher. In L. R. Knudsen, editor, *Advances in Cryptology - EUROCRYPT2002*, volume 2332 of *Lecture Notes in Computer Science*, pages 238–255. Springer-Verlag, 2002.

[54] Jovan Dj. Golić and Guglielmo Morgari. On the resynchronization attack. In T. Johansson, editor, *Fast Software Encryption2003*, volume 2887 of *Lecture Notes in Computer Science*, pages 100–110. Springer-Verlag, 2003.

[55] Dieter Gollmann and William G. Chambers. Clock-controlled shift registers: A review. *IEEE Journal on Selected Areas in Communications*, 7(4):525–533, May 1989.

[56] Solomon W. Golomb. *Shift Register Sequences*. Aegean Park, 1982.

[57] Philip Hawkes and Gregory G. Rose. Rewriting variables: The complexity of fast algebraic attacks on stream ciphers. In M. Franklin, editor, *Advances in Cryptology - CRYPTO2004*, volume 3152 of *Lecture Notes in Computer Science*, pages 390–406. Springer-Verlag, 2004.

[58] Martin E. Hellman. A cryptanalytic time-memory trade-off. *IEEE Transactions on Information Theory*, IT-26(4):401–406, July 1980.

[59] Miia Hermelin and Kaisa Nyberg. Correlation properties of the Bluetooth combiner. In J. Song, editor, *Information Security and Cryptology - ICISC'99*, volume 1787 of *Lecture Notes in Computer Science*, pages 17–29. Springer-Verlag, 2000.

[60] Jin Hong, Dong H. Lee, Yongjin Yeom, and Daewan Han. A new class of single cycle t-functions. In H. Gilbert and H. Handschuh, editors, *Fast Software Encryption2005*, volume 3557 of *Lecture Notes in Computer Science*, pages 68–82. Springer-Verlag, 2005.

[61] Jin Hong and Palash Sarkar. New applications of time memory data tradeoffs. In Bimal Roy, editor, *to appear in the proceedings of Advances in Cryptology - ASIACRYPT2005*, Lecture Notes in Computer Science. Springer-Verlag.

[62] Thomas Johansson and Frederik Jönsson. Fast correlation attacks based on turbo code techniques. In M. Wiener, editor, *Advances in Cryptology - CRYPTO'99*, volume 1666 of *Lecture Notes in Computer Science*, pages 181–197. Springer-Verlag, 1999.

[63] Thomas Johansson and Frederik Jönsson. Improved fast correlation attacks on stream ciphers via convolutional codes. In J. Stern, editor, *Advances in Cryptology - EUROCRYPT'99*, volume 1592 of *Lecture Notes in Computer Science*, pages 347–362. Springer-Verlag, 1999.

[64] Thomas Johansson and Frederik Jönsson. Fast correlation attacks through reconstruction of linear polynomials. In M. Bellare, editor, *Advances in*

Cryptology - CRYPTO2000, volume 1880 of *Lecture Notes in Computer Science*, pages 300–315. Springer-Verlag, 2000.

[65] Pascal Junod. *Statistical Cryptanalysis of Block Ciphers*. PhD thesis, EPFL, 2004.

[66] Aviad Kipnis and Adi Shamir. Cryptanalysis of the HFE public key cryptosystem by relinearization. In M. Wiener, editor, *Advances in Cryptology - CRYPTO'99*, volume 1666 of *Lecture Notes in Computer Science*, pages 19–30. Springer-Verlag, 1999.

[67] Alexander Klimov and Adi Shamir. A new class of invertible mappings. In B. S. Kaliski Jr., Ç. K. Koç, and C. Paar, editors, *Cryptographic Hardware and Embedded Systems - CHES2002*, volume 2523 of *Lecture Notes in Computer Science*, pages 470–483. Springer-Verlag, 2003.

[68] Alexander Klimov and Adi Shamir. Cryptographic applications of t-functions. In M. Matsui and R. Zuccherato, editors, *Selected Areas in Cryptography2003*, volume 3006 of *Lecture Notes in Computer Science*, pages 248–261. Springer-Verlag, 2004.

[69] Alexander Klimov and Adi Shamir. New cryptographic primitives based on multiword t-functions. In B. Roy and W. Meier, editors, *Fast Software Encryption2004*, volume 3017 of *Lecture Notes in Computer Science*, pages 1–15. Springer-Verlag, 2004.

[70] Matthias Krause. BDD-based cryptanalysis of keystream generators. In L. R. Knudsen, editor, *Advances in Cryptology - EUROCRYPT2002*, volume 2332 of *Lecture Notes in Computer Science*, pages 222–237. Springer-Verlag, 2002.

[71] Sangjin Lee, Seongtaek Chee, Sangjoon Park, and Sungmo Park. Conditional correlation attack on nonlinear filter generators. In K. Kim and T. Matsumoto, editors, *Advances in Cryptology - ASIACRYPT'96*, volume 1163 of *Lecture Notes in Computer Science*, pages 360–367. Springer-Verlag, 1996.

[72] Rudolf Lidl and Herald Niederreiter. *Introduction to Finite Fields and Their Applications*. Cambridge, 1986.

[73] Bernhard Löhlein. Attacks based on conditional correlations against the nonlinear filter generator, 2003. available on-line at `http://eprint.iacr.org/2003/020`.

[74] Florence Jessie MacWilliams and Neil J. A. Sloane. *The Theory of Error-correcting Codes*. North-Holland, 1996.

[75] Itsik Mantin. A practical attack on the fixed RC4 in the WEP mode. In Bimal Roy, editor, *to appear in the proceedings of Advances in Cryptology - ASIACRYPT2005*, Lecture Notes in Computer Science. Springer-Verlag.

[76] James L. Massey. Shift-register synthesis and BCH decoding. *IEEE Transactions on Information Theory*, IT-15(1):122–127, Jan. 1969.

[77] Mitsuru Matsui. Linear cryptanalysis method for DES cipher. In T. Helleseth, editor, *Advances in Cryptology - EUROCRYPT'93*, volume 765 of *Lecture Notes in Computer Science*, pages 386–397. Springer-Verlag, 1994.

[78] Alexander Maximov, Thomas Johansson, and Steve Babbage. An improved correlation attack on A5/1. In H. Handschuh and A. Hasan, editors, *Selected Areas in Cryptography 2004*, Lecture Notes in Computer Science, pages 1–18. Springer-Verlag, 2005.

[79] Willi Meier, Enes Pasalic, and Claude Carlet. Algebraic attacks and decomposition of boolean functions. In C. Cschin and J. Camenisch, editors, *Advances in Cryptology - EUROCRYPT2004*, volume 3027 of *Lecture Notes in Computer Science*, pages 474–491. Springer-Verlag, 2004.

[80] Willi Meier and Othmar Staffelbach. Fast correlation attacks on stream ciphers (extended abstract). In C. Günther, editor, *Advances in Cryptology - EUROCRYPT'88*, volume 330 of *Lecture Notes in Computer Science*, pages 301–314. Springer-Verlag, 1988.

[81] Willi Meier and Othmar Staffelbach. Fast correlation attacks on certain stream ciphers. *Journal of Cryptology*, 1(3):159–176, 1989.

[82] Willi Meier and Othmar Staffelbach. Nonlinearity criteria for cryptographic functions. In J. J. Quisquater and J. Vandewalle, editors, *Advances in Cryptology - EUROCRYPT'89*, volume 434 of *Lecture Notes in Computer Science*, pages 549–562. Springer-Verlag, 1990.

[83] Willi Meier and Othmar Staffelbach. Analysis of pseudo random sequences generated by cellular automata. In D. W. Davies, editor, *Advances in Cryptology - EUROCRYPT'91*, volume 547 of *Lecture Notes in Computer Science*, pages 186–199. Springer-Verlag, 1991.

[84] Willi Meier and Othmar Staffelbach. Correlation properties of combiners with memory in stream ciphers. *Journal of Cryptology*, 5:67–86, Nov. 1992.

[85] Alfred J. Menezes, Paul C. van. Oorschot, and Scott A. Vanstone. *Handbook of Applied Cryptography*. CRC, 1996.

[86] Miodrag J. Mihaljević, Marc P. C. Fossorier, and Hideki Imai. A low-complexity and high-performance algorithm for the fast correlation attack. In B. Schneier, editor, *Fast Software Encryption2000*, volume 1978 of *Lecture Notes in Computer Science*, pages 196–212. Springer-Verlag, 2001.

[87] Miodrag J. Mihaljević, Marc P. C. Fossorier, and Hideki Imai. Fast correlation attack algorithm with list decoding and an application. In M. Matsui, editor, *Fast Software Encryption2001*, volume 2355 of *Lecture Notes in Computer Science*, pages 196–210. Springer-Verlag, 2002.

[88] Miodrag J. Mihaljević and Jovan Dj. Golić. A fast iterative algorithm for a shift register initial state reconstruction given the noisy output sequence. In J. Seberry and J. Pieprzyk, editors, *Advances in Cryptology - AUSCRYPT'90*, volume 453 of *Lecture Notes in Computer Science*, pages 165–175. Springer-Verlag, 1990.

[89] Miodrag J. Mihaljević and Jovan Dj. Golić. A comparison of cryptanalytic principles based on iterative error-correction. In D. W. Davies, editor, *Advances in Cryptology - EUROCRYPT'91*, volume 547 of *Lecture Notes in Computer Science*, pages 527–531. Springer-Verlag, 1991.

[90] Miodrag J. Mihaljevic and Hideki Imai. Cryptanalysis of TOYOCRYPT-HS1 stream cipher. *IEICE Transactions on Fundamentals*, E85-A(1):66–73, Jan. 2002.

[91] Chris J. Mitchell and Alexander W. Dent. International standards for stream ciphers: A progress report. SASC - The State of the Art of

Stream Ciphers, special workshop hosted by the ECRYPT Network of Excellence, Oct. 2004. available on-line at `http://www.isg.rhul.ac.uk/research/projects/ecrypt/stvl/sasc.html`.

[92] Michel Mouly and Marie Bernadette Pautet. *The GSM System for Mobile Communications*. 1992.

[93] New European Schemes for Signatures, Integrity, and Encryption (NESSIE). available on-line at `http://www.cryptonessie.org`.

[94] Philippe Oechslin. Making a faster cryptanalytic time-memory trade-off. In D. Boneh, editor, *Advances in Cryptology - CRYPTO2003*, volume 2729 of *Lecture Notes in Computer Science*, pages 617–630. Springer-Verlag, 2003.

[95] Walter T. Penzhorn. Correlation attaks on stream ciphers: Computing low-weight parity checks based on error-correcting codes. In D. Gollmann, editor, *Fast Software Encryption'96*, volume 1039 of *Lecture Notes in Computer Science*, pages 159–172. Springer-Verlag, 1996.

[96] Rainer A. Rueppel. *Analysis and Design of Stream Ciphers*. Springer-Verlag, 1986.

[97] Rainer A. Rueppel. Correlation immunity and the summation generator. In H. C. Williams, editor, *Advances in Cryptology - CRYPTO'85*, volume 218 of *Lecture Notes in Computer Science*, pages 260–272. Springer-Verlag, 1986.

[98] Rainer A. Rueppel. Stream ciphers. In Gustavus J. Simmons, editor, *Contemporary cryptology: the Science of Information Integrity*, chapter 2, pages 65–134. IEEE Press, 1992.

[99] Markku Saarinen. Re: Bluetooth and E0, 2000. posted at `http://sci.crypt.research`.

[100] Palash Sarkar and Subhamoy Maitra. Construction of nonlinear boolean functions with important cryptographic properties. In B. Preneel, editor, *Advances in Cryptology - EUROCRYPT2000*, volume 1807 of *Lecture Notes in Computer Science*, pages 485–506. Springer-Verlag, 2000.

[101] Palash Sarkar and Subhamoy Maitra. Nonlinearity bounds and constructions of resilient boolean functions. In M. Bellare, editor, *Advances in Cryptology - CRYPTO2000*, volume 1880 of *Lecture Notes in Computer Science*, pages 515–532. Springer-Verlag, 2000.

[102] Bruce Schneier. *Applied Cryptography: Protocols, Algorithms, and Source Code in C*. John Wiley & Sons, 1996.

[103] Jennifer Seberry, Xian-Mo Zhang, and Yuliang Zheng. On constructions and nonlinearity of correlation immune functions (extended abstract). In T. Helleseth, editor, *Advances in Cryptology - EUROCRYPT'93*, volume 765 of *Lecture Notes in Computer Science*, pages 181–199. Springer-Verlag, 1994.

[104] Jennifer Seberry, Xian-Mo Zhang, and Yuliang Zheng. Relationships among nonlinearity criteria (extended abstract). In A. De Santis, editor, *Advances in Cryptology - EUROCRYPT'94*, volume 950 of *Lecture Notes in Computer Science*, pages 376–388. Springer-Verlag, 1995.

[105] Yaniv Shaked and Avishai Wool. Cracking the Bluetooth PIN. Proceedings of the 3rd International Conference on Mobile Systems, Applications, and Services (MobiSys) 2005, Seattle, Washington, 2005.

[106] Claude E. Shannon. Communication theory of secrecy systems. *Bell System Technical Journal*, 28:656–715, 1949.

[107] Thomas Siegenthaler. Correlation-immunity of nonlinear combining functions for cryptographic applications. *IEEE Transactions on Information Theory*, IT-30(5):776–780, Sep. 1984.

[108] Thomas Siegenthaler. Decrypting a class of stream ciphers using ciphertext only. *IEEE Transactions on Computers*, C-34(1):81–85, Jan. 1985.

[109] Thomas Siegenthaler. Cryptanalysts representation of nonlinearly filtered ML-sequences. In F. Pichler, editor, *Advances in Cryptology - EUROCRYPT'85*, volume 219 of *Lecture Notes in Computer Science*, pages 103–110. Springer-Verlag, 1986.

[110] Adam Stubblefield, John Ioannidis, and Aviel D. Rubin. A key recovery attack on the 802.11b wired equivalent privacy protocol (WEP). *ACM Transactions on Information and System Security*, 7:319–332, 2004.

[111] Serge Vaudenay. An experiment on DES - statistical cryptanalysis. In *Proceedings of the 3rd ACM Conferences on Computer Security*, pages 139–147, 1996.

[112] Gilbert S. Vernam. Cipher printing telegraph systems for secret wire and radio telegraphic communications. *Journal of the IEEE*, 55:109–115, 1926.

[113] David Wagner. A generalized birthday problem. In M. Yung, editor, *Advances in Cryptology - CRYPTO2002*, volume 2442 of *Lecture Notes in Computer Science*, pages 288–304. Springer-Verlag, 2002.

[114] Markus Jakobsson Susanne Wetzel. Security weakness in Bluetooth. In D. Naccache, editor, *Topics in Cryptology - CT-RSA 2001*, volume 2020 of *Lecture Notes in Computer Science*, pages 176–191. Springer-Verlag, 2001.

[115] Stephen Wolfram. Cryptography with cellular automata. In H. C. Williams, editor, *Advances in Cryptology - CRYPTO'85*, volume 218 of *Lecture Notes in Computer Science*, pages 429–432. Springer-Verlag, 1986.

[116] Stephen Wolfram. *A New Kind of Science*. Wolfram Media, 2002.

[117] Rao K. Yarlagadda and John E. Hershey. *Hadamard Matrix Analysis and Synthesis with Applications to Communications and Signal/Image Processing*. Kluwer Academic, 1997.

[118] Kencheng Zeng and Minqiang Huang. On the linear syndrome method in cryptanalysis. In S. Goldwasser, editor, *Advances in Cryptology - CRYPTO'88*, volume 403 of *Lecture Notes in Computer Science*, pages 469–478. Springer-Verlag, 1990.

[119] Erik Zenner. On the role of the inner state size in stream ciphers, 2004. available on-line at `http://eprint.iacr.org/2004/003`.

[120] Yuliang Zheng and Xian-Mo Zhang. On relationships among avalanche, nonlinearity, and correlation immunity. In T. Okamoto, editor, *Advances in Cryptology - ASIACRYPT2000*, volume 1976 of *Lecture Notes in Computer Science*, pages 470–482. Springer-Verlag, 2000.

CV

Personal Data

Family Name: LU
First Name: YI
Nationality: Chinese
Gender: female

Education

Oct. 2002–present, Ph.D. in Sciences, Laboratory of Cryptography and Security (LASEC), Department of Computer & Communication Sciences, École Polytechnique Fédérale de Lausanne (EPFL), Switzerland

Oct. 2001–Jul. 2002, postgraduate at Department of Computer & Communication Sciences, EPFL, Switzerland

Sep. 1996–Jul. 2001, Bachelor of Engineering, Department of Computer, Beijing Polytechnic University, China

Awards

One-year scholarship at EPFL for postgraduate studies, 2001-2002

The first class of scholarship at Beijing Polytechnic University in 1998, and the third class of scholarship in 1997, 1999 and 2000

No. 31 in Beijing Mathematics Contest in 1992, 1993

The third class of prize in Beijing Mathematics Contest of Ying Chun Cup in 1990, 1991

Professional Experience

Oct. 2002–present, Ph.D. thesis entitled "Applied Stream Ciphers in Mobile Communications" under supervision of Prof. Serge Vaudenay, LASEC, Department of Computer & Communication Sciences, EPFL

Mar. 2005–Jul. 2005, teaching assistant for the course "Advanced Cryptography", Department of Computer & Communication Sciences, EPFL

Mar. 2004–Jul. 2004, teaching assistant for the course "Communication Security", Department of Computer & Communication Sciences, EPFL

Mar. 2004–Jul. 2004, supervision of one undergraduate student's semester project entitled "A Practical Attack against A5/1", LASEC, Department of Computer & Communication Sciences, EPFL

Feb. 2002–Jun. 2002, semester project on cryptanalysis of GSM encryption algorithm A5/1, LASEC, Department of Computer & Communication Sciences, EPFL

Feb. 2001–Jun. 2001, as the project for diploma at Beijing Polytechnic University, developed the software application "Real-time Traffic Flow Simulation on the Crossroad Using Multi-threads Technique"

Sep. 2000–Aug. 2001, participated in the research work of "Modelling and Simulation of Reactive Ion Etching Technology by Artificial Neural Networks", part of the project "TCAD for Optoelectronic Devices" at the State Key Laboratory on Integrated Optoelectronics, Chinese Academy of Sciences, Beijing

Publications

Yi Lu, Serge Vaudenay, "Faster Correlation Attack on Bluetooth Keystream Generator E0", Advances in Cryptology - CRYPTO 2004, Lecture Notes in Computer Science, vol.3152, M. Franklin Ed., Springer-Verlag, pp. 407-425, 2004.

Yi Lu, Serge Vaudenay, "Cryptanalysis of Bluetooth Keystream Generator Two-level E0", Advances in Cryptology - ASIACRYPT 2004, Lecture Notes in Computer Science, vol.3329, P. J. Lee Ed., Springer-Verlag, pp. 483-499, 2004.

Yi Lu, Willi Meier, Serge Vaudenay, "The Conditional Correlation Attack: A Practical Attack on Bluetooth Encryption", Advances in Cryptology - CRYPTO 2005, Lecture Notes in Computer Science, vol.3621, V. Shoup Ed., Springer-Verlag, pp. 97-117, 2005.