An Attempt to Cryptanalyze A Partially Known Cipher Algorithm

Juay Guan Hee
hjuaygua@gmail.com

10 Jun 2017

Abstract

This paper presents an empirical crypt-analytical method to analyse a partially known cipher algorithm. During cipher evaluation, it is always a challenge to make any decision on the strength of a partially known cipher algorithm, and if the algorithm is suitable for deployment. The core concept will be presented first, followed by an example to illustrate the idea. The idea is to focus on one input bit at a time using a known keystream attack, assuming this bit is independent from the rest. By computing the statistics of related keystream bits and using the correlation method, one can derive this input bit with certain confidence.

keywords Linear Feedback Shift Register, Correlation Coefficient, Stream Cipher.

1 Introduction

During the cipher algorithm evaluation process, one of the greatest challenges is to evaluate a product when a potential vendor does not wish to disclose all the details of the cipher algorithm before any commercial commitment. For the purpose of evaluation, a vendor is usually willing to release partial information such as a partially known cipher algorithm, and the keystreams generated using different based keys (secret keys) as a common practice. This gives rise to the need to provide a fair assessment technique without full disclosure of the cipher algorithm from the vendor. This paper describesa novel approach using an empirical method as an attempt to analyse a partially known cipher algorithm in this situation.

2 Core Concept

To illustrate the idea, this core method will assume that the given cipher system has a group of linear feedback shift registers (LFSRs) as a keystream

generator. Each of the LFSRs is tapped at one point to extract out a bit as input bit into a mixing function, or a lookup table to form keystream bits in one clock cycle. Usually, these generated keystream bits will form an Exclusive OR (XOR) with the plain text bits to form the cipher bits for sending out. Again, these processes are done within the same clock cycle.

2.1 Assumptions of the stream cipher system

This stream cipher system has a group of LFSRs, where the length of each LFSR and its feedback points are given for the purpose of computing the keystream cycle length of the entire system. This is to ensure that the keystream length is long enough for the required application. The keystream length is usually 2 to the power of a few hundred bits. Roughly speaking, different lengths of LFSRs are used to generate the largest keystream cycle. Next, these LFSRs are filled with a base key (BK) and there is no pre-run process. In addition, the tapping points for extracting out the bits in generating the keystream bits are known. The generated keystream bits are used to encrypt the plaintext bits to form the ciphertext bits. Other than the above, no further details are given on the cipher algorithm.

As the mixing function or look-up-table is unknown, it can be assumed that there is a correlation between the bit extracted from LFSRs and the keystream bits. As long as the mixing is not balanced, this method can be applied. Otherwise, this method will fail, as a decision cannot be made. This is not a problem as a balanced correlation implies that the keystream is a linear combination of the input based key (BK), wich is a designed to avoid being attacked. For this method, it is also assumed that the unknown mixing is a memory-less type of circuitry representing the mixing from the LFSRs output.

Mathematically, the known M bits of BK content are used to fill up all the bits in LFSRs.

Let $SR[i, j]$ denotes the j^{th} bit of the i^{th} LFSR where there are n LFSRs, and each LFSR is of length m_i. Here, $M = \sum_{i=1}^{n} m_i$.

For the purpose of illustration, it can be assumed that the n bits from each of the LFSR, $SR[i, m_i]$ where $i = 1, ..., n$ are extracted for mixing to generate keystream bits, $KS[t]$ where time $t = 1, ..., N$ is the clock cycle.

2.2 Method

In this method, the content of all LFSRs are assumed to be unknown. That is, the base key, BK is assumed to be unknown. The unknown BK will be denoted as BK'. The following steps will be used to derive the BK' for filling up all the LFSRs. If the derived BK' is found to be identical to BK, or the complement of BK, then the cipher algorithm is assessed to be weak.

Correlation **P**	Computed **P'**	Estimated bit **b**
< 0.5	< 0.5	0
	> 0.5	1
> 0.5	< 0.5	1
	> 0.5	0

Table 1: Estimated bit **b** content

2.2.1 Selection of bit

This method will compute the statistics of a selected single bit, say $SR[i, j]$ for a fixed i and j. One bit will be chosen at a time starting from one of the LFSRs until all bits from the same LFSR are exhusted. Similar steps will be done for the rest of the LFSRs.

2.2.2 Computation of statistics

For the chosen bit, **b**, statistics will be computed. The method of computing statistics will be illustrated in section 3. As long as **b** is involved in the mixing function/computation, the output keystream bits will be collected and used for computing the statistics, **P'**.

2.2.3 Correlation method

Design of a good cipher algorithm will usually ensure that the correlation of the keystream bits and the base key is not 0.5. This is to prevent a linear relationship between the base key and the keystream from forming. Therefore, it is safe to assume that the mixing circuitry is not a balanced one; that is, the correlation, **P**, is not 0.5. For the purpose of computation, we shall assume the correlation, **P**, is less than 0.5. We should be able to estimate **b** by using the above statistics. As summarized in Table 1, if **P'** is less than 0.5, then **b** is probably a 0. Otherwise, it is a 1.

2.2.4 Comparison and assessment

With the BK' found through the above process, one can compare if this is the same as the given BK, or if it is a complement of the given BK (i.e. **P** is greater than 0.5). For both cases, the cipher algorithm is considered as weak.

3 Examples in computing statistics

In the following examples, a partially known cipher algorithm of a tactical communication system will be used to illustrate the above concept. This system has a group of LFSRs as a keystream generator for speedy generation

of keystreams. This example will show the steps to compute a selected bit, **b**. It is also assumed that the tapping points are extracted from all the least significant bit of the LFSRs. Two of the LFSRs have lengths of 137 bits and 124 bits for this example. The primitive polynomials used on each of the two LFSRs are $x^{137} + x^{21} + 1$, and $x^{124} + x^{37} + 1$, respectively [1].

3.1 Solving selected bits in LFSR $x^{137} + x^{21} + 1$

For this primitive polynomial, the selected bit, **b** is the first output bit, or the least significant bit of this LFSR for a start of this method. The keystream bits $KS[1], KS[138], KS[254], KS[275], KS[370], KS[412], KS[486]$, etc are collected, as these bits are derived from the selected bit, **b**. These bits will be summed to obtain the average, **P'**. By using **P'** and Table 1, the least significant bit, **b** from this LFSR can be derived.
As for the second least significant bit, **b** of this LFSR, the keystream bits $KS[2], KS[139], KS[255], KS[276], KS[371], KS[413], KS[487]$, etc. will be collected. These bits will be summed to compute the average, **P'**, in the same manner. By using **P'** and the Table 1, the second least significant bit, **b** from this LFSR can be derived too.

3.2 Solving selected bits in LFSR $x^{124} + x^{37} + 1$

For this primitive polynomial, the selected bit, **b** is the first output bit, or the least significant bit of this LFSR to start with. The keystream bits $KS[1], KS[125], KS[212], KS[249], KS[299]$, etc. will be collected. The average, **P'**, will be computed. By using **P'** and the Table 1, the least significant bit, **b** of this LFSR can be derived also.

4 Future Research

The concept introduced in this paper provides many interesting issues to be addressed in future research. Based on the above testing, it is generally observed that the estimated value of the selected bit **b** is rather accurate when the sample size is small. A future project could explore the possibility of applying this concept to a stream cipher system with some intermediate buffer and determining a design that could resist this attack.

5 Conclusions

This method is efficient. The complexity is linear with order $(O(M))$, as each bit and each linear shift register can be computed independently. The conventional method will yield a complexity of $(O(2^M))$ where $M = \sum_{i=1}^{n} m_i$.

Although the method focuses only on keystream attacks, it could be extended to ciphertext attacks for (known) formatted plain text bits, such as formatted messages with a fixed format header.

References

[1] Miodrag Živković. A Table of Primitive Binary Polynomials, *Mathematics of Computation*(Jan. 1994), 62(205), pp. 385–386.