Computational Sciences and Engineering 3(1) (2023) 115-128

University of Guilan

Computational Sciences and Engineering

journal homepage: https://cse.guilan.ac.ir/

Accelerated brute force attack on sequential cipher systems on LFSR

Alireza Babaei [a], Hamid Haj Seyyed Javadi [b,*], Nader Jafari Rad [a]

[a] Department of Mathematics, Faculty of Basic Sciences, Shahed University, Tehran, Iran
[b] Department of Computer Engineering, Faculty of Computer Engineering, Shahed University, Tehran, Iran

ARTICLE INFO

Article history:
Received 7 November 2023
Received in revised form 15 February 2024
Accepted 19 February 2024
Available online 1 April 2024

Keywords:
Algebraic attack
Stream cipher
Extended linearization algorithm

ABSTRACT

Algebraic attack is an emergent decryption method. The main objective in this decryption is to form and solve a set of multivariate polynomial equations on finite fields. The present findings show that algebraic attacks have been significantly successful and effective on a specific type of stream ciphers system and linear-feedback shift register systems (LFSRs). One of the reasons for this is that linear functions are used for updating LFSRs, although the nonlinear types can also be approximated by an appropriate linear function, and this increases the necessity of paying attention to it. In the present article, an attempt is made to present the main idea of algebraic attacks on stream ciphers systems, and to explain these ideas by certain concrete examples. Particularly, a synchronous stream cipher system based on LFSRs, entitled the LILI stream ciphers, and algebraic attacks on them, will be discussed. In this research, the extended linearization algorithm (XL) will be used to deal with an attained set of equations. Additionally, some of the accelerated extended algorithms (XL) for dealing with the set of equations algebraic resulted from the attacks on stream cipher systems, will be analyzed and their efficiency will be examined in the frame of certain examples.

1. Introduction

The advancement of technology and more powerful processing hardware caused more attention to encryption algorithms. Therefore, the importance of algebraic attacks, including effective attacks on symmetric cipher systems and especially sequential cipher systems, was revealed. Sequential encryption systems have a high encryption speed and are mostly used for cases that need to encrypt a lot of information in a logical unit of time. Methods are based on the production of polynomial multivariable equations on finite fields. One of the most important categories of

* Corresponding author.
E-mail addresses: h.s.javadi@shahed.ac.ir (H. Haj Seyyed Javadi)

https://doi.org/10.22124/cse.2024.25975.1065

sequential ciphers are ciphers based on LFSR. The basic step in algebraic attacks on sequential cipher systems is to create and solve a system of multivariable polynomial equations. The structure of this article is as follows: an efficient and powerful numerical algorithm known as the XL algorithm is introduced to solve this system of equations. In the second part, in order to speed up the execution of brute force attacks and increase their efficiency, we will introduce and examine some developments method of the standard forms of the XL algorithm, which are known as the FXL.

Research records: Capabilities and challenges of searching on encrypted data were reviewed by Najafi et al. [16]. Erfani et al. [8] presented the use of a residual scheme on key management in a hierarchical wireless sensor network. Fuzzy classification system was investigated in order to detect intrusion detection in computer networks (see Molin [15]). Several researches have worked on the security of authentication systems for sharing data in the cloud (see [1]). Key pre-distribution schemes for symmetrical combination keys have been presented by Anzani et al. [2]. Algebraic attacks based on non-linear criteria were presented by Eskandari et al. [9]. Other studies have been presented by some researchers. The research done by Krause et al. [12] showed that this category of efficiency is followed by this category of ciphers. In the following, this program is specifically used based on sequential cipher systems in LFSR, and the behaviors of this class of operations include ensuring the existence of algebraic equations related to key bits and key bits, as well as computational complexity. This program and the calculation of upper bounds for this category of calculations have been directly investigated in [11].

In recent years, in order to increase the efficiency of algebraic operations on sequential cipher systems based on LFSR, acceleration algorithms have been presented and an example of the research done in [4] has been published. Different researches have been done by other researchers in encryption systems, their security and types of operations, which can be mentioned as an example [13]. In the following, we will try to give a brief introduction of XL and FXL algorithms to enter the topic.

History and Background: The XL and FXL algorithms are powerful methods used in data analysis and machine learning tasks, particularly for classification and regression problems. These algorithms were developed as extensions of the popular XGBoost (eXtreme Gradient Boosting) algorithm, which itself is based on gradient boosting techniques. XGBoost gained popularity for its ability to handle complex data, large-scale datasets, and diverse types of features effectively. It leverages boosting, an ensemble learning technique, to create a strong predictive model by combining multiple weak models, such as decision trees. Building upon the success of XGBoost, the XL and FXL algorithms were introduced to enhance its capabilities further.

XL algorithm: The XL algorithm, short for eXtreme Learning, extends XGBoost by incorporating a novel approach for the ensemble construction. It introduces an additional randomization component to generate diverse and accurate base models. This randomization reduces overfitting and enhances the model's generalization capabilities. Unlike traditional boosting algorithms that sequentially add trees one by one, the XL algorithm builds the ensemble using a two-step process. At its core, it starts with the same principles as XGBoost—using a loss function and optimizing the objective. In the first step, it randomly initializes a subset of base models as regressors or classifiers. Then, in the second step, it optimizes the ensemble coefficients through a ridge regression process, resulting in an ensemble of base models.

FXL algorithm: The FXL algorithm, which stands for fully eXtreme Learning, is an improved version of the xl algorithm. It addresses the limitation of the initial version by refining the construction process and enhancing the model's robustness and interpretability. The FXL algorithm introduces an additional regularization term during the ensemble coefficient optimization process. This term encourages sparsity in the ensemble coefficients, leading to a more interpretable final model. Furthermore, the FXL algorithm incorporates L1 and L2 regularization to balance model complexity and predictive performance effectively.

Examples: To illustrate the application of xl and FXL algorithms, let's consider a classification problem. Suppose we have a dataset with various features used to predict whether a customer will churn or not in a telecommunications company. By applying XL or FXL algorithms, we can construct a powerful ensemble model that combines multiple weak models (decision trees) to make accurate predictions. The XL and FXL algorithms automatically handle feature selection, non-linearity, and interaction effects, making them suitable for a wide range of potential applications. Their versatility, interpretability, and robustness have made them popular choices in various domains, including finance, healthcare, and natural language processing. In conclusion, the xl and FXL algorithms are extensions of XGBoost that aim to improve its performance, flexibility, and interpretability. These algorithms have gained traction in the machine learning community, offering state-of-the-art solutions for challenging data analysis tasks.

Preliminary definitions: The XL and FXL (eXtended and Fast eXtended Learning) algorithms are advanced algorithms used for incremental learning in the field of machine learning. These algorithms are specifically designed to handle streaming data, where new information arrives in a continuous and sequential manner, making it challenging to train models in a traditional batch learning setting. The idea behind these algorithms is to update and adapt the model's parameters or structure incrementally, without having to retrain the entire model from scratch. This allows for efficient processing of large volumes of streaming data and the ability to adapt to concept drift (changes in the underlying data distribution over time).

XL and FXL algorithms build upon the popular online learning technique known as Extreme Learning Machine (ELM), which is characterized by its simplicity and fast learning speed. ELM typically trains a single hidden layer feed forward neural network (SLFN) by randomly initializing the input weights and analytically computing the output weights using Moore-Penrose pseudo inverse.

The XL algorithm extends ELM by introducing a hidden layer selection strategy, which dynamically adjusts the size and connectivity of the hidden layer based on the importance and relevance of each input feature. In this way, XL optimizes the network's architecture to better capture the patterns in the streaming data.

On the other hand, FXL is an even faster variant of XL, designed to handle high-dimensional data and improve computational efficiency. It achieves this by incorporating random Fourier features, which approximate the kernel trick commonly used in Support Vector Machines (SVMs). This approximation allows the FXL algorithm to achieve similar performance to traditional kernel methods while being much faster in practice. These algorithms have found applications in various domains, including online prediction, time-series forecasting, and real-time anomaly detection. They have been proven to be effective in situations where data arrives sequentially and is subject to concept drift. To illustrate their usage, let's consider a stock market prediction scenario.

Suppose you want to predict the future price of a particular stock based on historical data. Using the XL or FXL algorithms, you can construct a model that incrementally learns from the continuously streaming stock price data. As new stock prices come in, the model adapts and updates its parameters, allowing it to capture changing trends and patterns in the market over time. In this way, XL and FXL algorithms provide powerful tools for handling streaming data and adapting machine learning models in dynamic environments. Their ability to handle concept drift and process large volumes of data efficiently makes them a valuable addition to the field of incremental learning.

Expression of research innovation: Algebraic attacks on cryptographic systems are, in fact, an attempt to find the key of a cryptosystem using a set of algebraic equations. Algebraic attacks were used for the first time in the form of linearization algorithms to find the public key of cryptographic systems (see [3]). Later, this idea was developed in the form of XL algorithm to solve polynomial equations (see [4]). Next, the algebraic attack based on the XL algorithm was successfully used against AES cipher systems, (see [14]). Although subsequent studies showed that algebraic attacks do not have acceptable efficiency in dealing with some block cipher systems including S-boxes, however, they are a powerful tool in dealing with sequential cipher systems. The effectiveness of this type of attacks was first implemented well on a special class of sequence cipher systems, the Toyocrypt cipher system, and then these attacks were also developed on LiLI-128 sequence cipher systems ([5]). Next, it was predicted that the memory-based sequence code systems would be resistant to this type of attacks.

In this research, an attempt is made to comprehensively investigate the characteristics of algebraic attacks on LFSR-based sequence cipher systems and in addition, we will analyze the new accelerated algorithms. It is expected that after conducting this research, a sufficient knowledge regarding be achieved by the impact of brute-force attacks on at least one specific class of LILI sequential cipher systems based on $LFSR_s$. In addition, let's achieve an accurate measurement of the resistance of this particular class of sequential cipher systems to brute force attacks and their accelerated forms. It is tried to analyze the steps in such a way that this method is not only limited to this category, but by reviewing the analysis, this method can be implemented on other sequence codes as well.

An algebraic attack is a code breaking method that reduces the problem of decoding the attacked code to a problem of generating and solving a set of multivariate polynomial equations. Therefore, an algebraic attack on a cryptosystem is equivalent to the problem of finding and solving a system of nonlinear equations on a finite field. In most of the attacks that take place on encryption systems, one of the basic assumptions is that the structure of the encryption algorithm is known, and brute force attacks are no exception to this rule. In addition, we also need a number of sequential execution keys.

2. Sequential ciphers based on LFSR

We consider the following subsections.

2.1. History of cryptography

At first, there is a shift register of length $L \in N$, which contains a row of L registers from left to right labeled $R_{L-1}, R_{L-2}, \ldots . R_1, R_0$ each of which contains one bit. Suppose K_0 is the binary value in the first register on the right and $K_1$is the value of the second register from the right, and thus $K_{L-1}$is the value of the left register, then the initial state will be as follows.

Figure 1. The initial state

In the first pulse of the clock, it causes the first register from the left, K_{L-1}in the register $R_{L-1}$to move to the register R_{L-2} and the register K_{L-2} in R_{L-2} move to register R_{L-3} in the same way so that K_1 moves to R_0 and K_0 is placed in the output sequence. This move clears the R_{L-1} register. To fill this register, we need a linear function with variables kj for $j = 0, .., L-1$.

To achieve this, we need the Tap Sequence, which is an ordered $L-tuple$ of bits: $(C_{L-1}, \ldots, C_1, C_0)$with $C_0 = 1$. The above function is called linear feedback and we place the register K_L in the register R_{L-1}. ***Figure 2*** shows a simple design of a clock pulse.

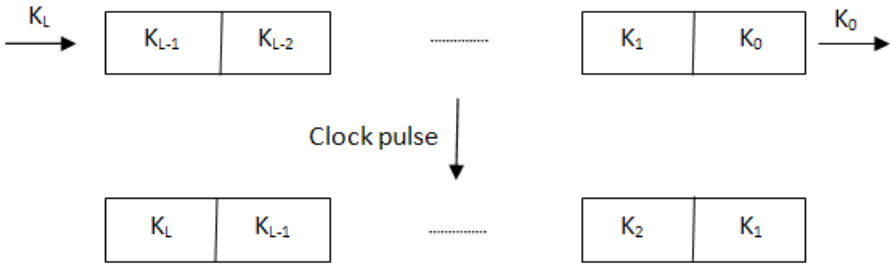

Figure 2. A simple design of a clock pulse

Sm state of LFSR is a bit string that describes all the registers Rj after $m+1$ clock pulse and $m \geq 0$. The initial state is called Seed, which cannot be a zero vector

$S_0 = (K_{L-1}, K_{L-2}, \ldots, K)$. After one clock pulse $S_1 = (K_L, K_{L-1}, \ldots, K_1)$. In the general case m≥0 and S_m= (K_{m+L-1}, K_{m+L-2}, K_{m+1}, K_m). Then linear feedback in the form $K_{m+L} = \sum_{J=0}^{L-1} C_j K_{m+j}$. This equation is also called binary regression relation. ***Figure 3*** is a simple design of a shift register with linear feedback.

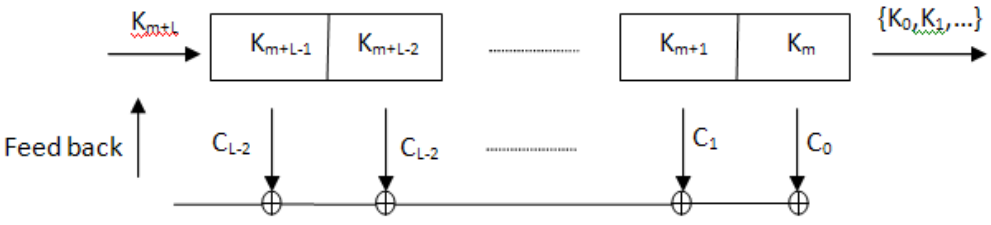

Figure 3. A simple design of a shift register with linear feedback

2.2. linear complexity of LFSR

If $t(x)$ is an irreducible polynomial of degree L. The corresponding LFSR has a linear complexity L for non-zero initial state. A necessary property for the security of an LFSR is that it has a large linear complexity.

2.3. Sequential codes based on FCSR

Suppose N>1 and is constant and $q_1, \ldots, q_m \in S, S = \{0,1,\ldots,N-1\}$. An N-tuple LFSR of length m with coefficients $q_1, \ldots, q_m$ (or Tap) is a discrete state machine so that the state is a convergence $a_0, \ldots, a_{m-1}, z$ and $z \in Z$, $a_j \in S$. The change of modes is explained below:

We put $\partial = \sum_{i=1}^{m} q_{iii} i_{ii} a m q_1 z$ and instead of $(a_0, a_1, \ldots, a_{m-1}, z)$ we put $a_1, a_2, \ldots$, δ(mod N); δ(div N). ***Figure 4*** is a simple design of FCSR (z is calculated as N/δ in each step)

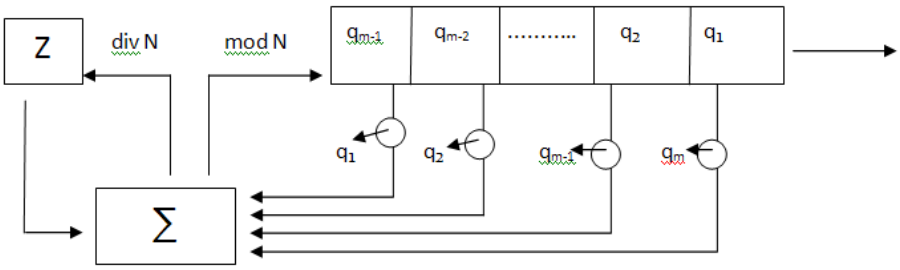

Figure 4. A simple design of FCSR

2.4. SNOW 3G serial code systems

SNOW 3G includes an LFSR and FSM. LFSR is made of 16 registers, each of which is 32 bits, and the feedback is defined using elementary polynomials on the finite field $GF(2^{32})$. FSM is based on 32-bit 3 registers $R3, R2, R1$.The operation performed in FSM is that the inputs come from LFSR and are replaced by using 2 insertion tables S_2, S_1. ***Figure 5*** is a plan of SNOW 3G.

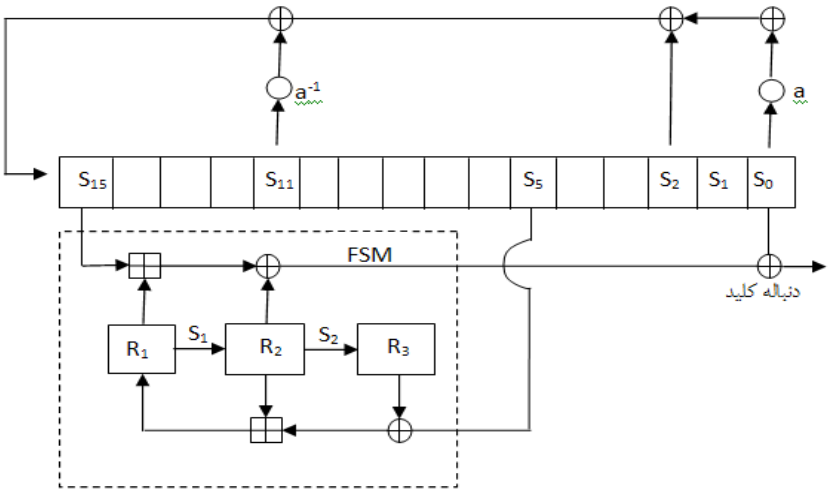

Figure 5. A plan of SNOW 3G

3. Algebraic attack against symmetric and sequential cipher systems

Algebraic attack methods are based on the formation and solution of a system of nonlinear equations that describe the encryption system. The complexity of the data is of the order of O(M), respectively, where

$$M = \sum_{i=0}^{d} C(n, i)$$

And n is the size of the mode and d is the degree of the output function, and $C(\dots,\dots)$ represents the binomial coefficients [3].

3.1. Linearization of brute-force attack

Suppose we have a quadratic device with m equations and n variables $\binom{n}{2}$ state of the second degree expression and n states of the first degree expression, which in total becomes $\frac{n^2+n}{2}$.

It is clear that if there is a solution for the device of the first few variables, then it will be a solution for the device resulting from linearization, and the opposite case is not true. That is, there may be many answers for the linear device, but the original (first) device does not have an answer (K. Alimohammadi, A secure key-aggregate authentication cryptosystem for data sharing in dynamic cloud storage, (2020)) .

3.2. Solving nonlinear equations using XL algorithm

XL algorithm is an equation solving method related to Grubner bases. The cryptanalysts claim that this method is effective for all basic sequential cipher systems such as Toyo crypt and $E0$ (Bluetooth) (K. Alimohammadi, A secure key-aggregate authentication cryptosystem for data sharing in dynamic cloud storage, (2020)) .

XL algorithm requires more equations. The XL algorithm is as follows:

Input: a set of m linear equations with n variables, of degree n.

Output: One or more solutions for the equation machine if the number of independent equations is sufficient.

Algorithm process:

1- A degree is chosen (by the person), usually ($D = d + 1$) $D > d$

2- Prepare a list like L of all monographs of degree D-d or less which includes a monograph and is of zero degree.

3- Multiply all the equations in each term of L. (After this step we will have m|L| equation).

4- Linearize the system.

5 -Solve the machine using linear algebra.

Example: Suppose the device has given the following equations and asked for their answers.

$$1 + X + Y + Z + WZ + YZ = 0.$$

$$X + Z + WX + WY + WZ + XY + XZ + YZ = 1.$$

$$W + Y + WX + XZ + YZ = 0.$$

$$X + WX + WY + WZ + YZ = 1.$$

At first, it seems that linearization can help us. We have four variables, so there are 10 terms, so that 6 terms are second order and 4 terms are linear. Although we only have 4 equations much less than 10.

Now suppose we made a cubic machine (the highest order of terms is 3) cubic machine $\binom{n}{3} + \binom{n}{2} + \binom{n}{1} = \frac{n^3}{6} + \frac{5}{6}n$ has n possible sentences. In our example $\binom{4}{3} = 4$ there are cubic terms and not 14 but only 10 equations are needed.

3.3. Linear complexity of the XL algorithm

Naturally, it is easy to check all 2^4 states for X, Y, Z, and W, but it takes 2^n steps.

For $d = 2$ and $D = 3$ and the second order device m, the equation of n unknowns is the result of $\frac{n^3}{6} + \frac{5}{6}n$ terms, which $m(n+1)$ of the equation is the result of a matrix $(mn + m) \times \left(\frac{n^3}{6} + \frac{5}{6}n\right)$ is obtained. With the Gauss elimination method, it will require approximately $\frac{n}{2^{16}}$ operations $\left(m \approx \frac{n^2}{6}\right)$.

3.4. Enough number of equations

If the full-rank matrix is a column, then the answer will be unique, and if the number of rows is less than the number of columns, then the full-rank matrix will not be a column. Although the condition that the number of rows is equal to the number of columns is not enough for full rank columns, because the rows may be the same. The matrix for D = d+1 will be square when $m \approx \frac{n^2}{6}$ or $(n+1)\mathrm{m} = \frac{n^3}{6} + \frac{5}{6}\mathrm{n}$.

3.5. FXL algorithm

In order to speed up the execution of brute force attacks and increase their efficiency, some developments use the standard forms of the XL algorithm, which are known as the FXL algorithm. Guessing to help solve the equation, the attacker gives random values to some variables to guess them, hoping that the degree of D will decrease. After guessing, he performs the XL algorithm and finally checks. It determines whether the answers are acceptable or not.

3.6. XL2 algorithm: Benefit from additional equations with the method $\acute{T}$

We put it (belongs to the new device in XL)

$\tau = \{X^b[X^b = X_1^{b_1}, \dots X_n^{b_n}, |b| \leq D, X^b]\}$.

We call $R = R^{(\mathfrak{v})}$the new device in XL. Suppose the number of equations and variables are R and T respectively. (In the new device in XL)

If the first device has an answer, the number of independent equations (we denote the rank of the device by I) does not exceed $T - 1$. If $I = T - 1$ then we expect the XL method to have a unique answer. Assume that $\acute{T}$ the total number of sentences, when multiplied by the variables, still belongs to τ. $\acute{T} = |\acute{\tau}_i|$for each i so that $\acute{\tau}_i = \{X^b = X_i X^b e\tau\}$.

Suppose I is not as big as T-D but $c = T' + I - T > 0$. Then

1- First, we remove all singularities that are not in $\acute{\tau}_1$from R=R$^{(\mathfrak{v})}$. Then, we obtain R_1 relations such that every monograph in $\tau/\acute{\tau}$ is a linear combination of monographs in $\acute{\tau}_1$.

2- for $\acute{\tau}_2$ to obtain the equations $\acute{R}_2, R_2$ (we should also have $|\acute{R}_2| = C$).

3 -For each $L \in \acute{R}_1$، the monograph that is in $X1L = 0$ or in $\acute{\tau}_2$ can be reduced to $\acute{\tau}_2$ (using R2).

4. Efficiency and comparison of brute force attack methods

Sequential encryption systems play an important and influential role in new symmetric encryptions which are significantly used in practice due to their efficiency and related hardware issues, which can be, for example, sequence codes that are predominantly used in global mobile communication systems (GSM) and it is currently used by more than two billion users. As another example, we can mention SNOW serial code systems, which are widely used in mobile recently mentioned. That the domestic operators refer to it by different names such as 3G and 4G. Therefore, due to the high efficiency and widespread use of this category of encryption systems, it is necessary to examine the sequence codes from different aspects, including brute force attacks. We try to solidly investigate the properties of algebraic attacks on sequence cipher systems based on LFSRs, and in addition, we will analyze the new accelerated algorithms.

The effectiveness of comparing this algebraic attack based on articles by Shamir, Kipnis, Courtois, XL algorithms for solving a system of multivariable equations and also methods to reduce the degree of a system of multivariable polynomial equations such as algebraic attack with correlation with Latter and algebraic attack Fast is based on the effect of guessing on the complexity of solving a system of polynomial equations.

4.1. Key Generation Mechanisms of LILI Systems and Algebraic Attacks Analysis

The LILI stream cipher family uses two primary linear feedback shift registers (LFSR), denoted as $LFSR_c$ and $LFSR_d$, to generate key streams. The configurations and operations of these ciphers are as follows:

LILI-128 Key Generator

1. Register Configuration:
    - $LFSR_c$ of length L=39.

- $LFSR_d$ of length m=89.
- Total internal state size: 128 bits.

2. Key Stream Generation:
   - At time t, registers 12 and 20 of $LFSR_c$ are inputs to a function f_c, $F_c\left(C_{12}^t, C_2^t\right) = 2\left(C_{12}\right)^t + C_{20}^t + 1$ ،which outputs an integer between 1 and 4.
   - F_d , the filter function, takes 10 inputs from $LFSR_d$ (specific positions: 80, 65, 44, 30, 20, 12, 7, 3, 1, 0) and has an algebraic degree of 6.
   - The initial state is determined by XORing a 128-bit key K with a 128-bit IV. If V is shorter than 128 bits, it is repeated to fill the size.
3. Initialization Process:
   - $LFSR_c$ is loaded with the first 39 bits, and $LFSR_d$ with the next 89 bits.
   - Parameters a and b define initialization cycles and the number of bits discarded during key stream generation, respectively.
   - Suggested values: $a \geq 1$, $b \in \{32,64,128\}$.

LILI-II Key Generator

1. Register Configuration:
   - $LFSR_c$ of length L=128.
   - $LFSR_d$ of length m=127.
   - Total internal state size: 255 bits.
2. Key Stream Generation:
   - At time t, registers 1 and 127 of $LFSR_c$ are inputs to F_c :

$F_C(C_1^t, C_{127}^t) = 2(C_1)^t + C_{127}^t + 1$.

   - F_d , the filter function, uses 12 inputs from $LFSR_d$ (positions: 122, 96, 80, 65, 44, 30, 20, 12, 7, 3, 1, 0) and has an algebraic degree of 10.
3. Initialization Process:
   - Initial states of $LFSR_c$ and $LFSR_d$ are derived by XORing K and V, with some bits dropped and rearranged to form a 255-bit internal state.

The generator undergoes multiple setup phases before being ready to produce a key stream.

4.2. Algebraic Attacks on LILI Systems

Algebraic attacks on the LILI family involve solving polynomial equations formed from the cipher's internal state and key stream bits. The attack's complexity depends on the degree of the equations.

Attack 1: Guessing Controller States

1. Method:
    - Guess the initial state of $LFSR_c$.
    - Use the feedback pattern to generate equations for $LFSR_d$'s initial state and key stream bits.
    - Solve these equations to verify the guess.
    - If the equations are inconsistent, the guess is incorrect.
2. Complexity:
    - Requires 2^{18} key stream bits and 2^{102} equations for LILI−128.
    - Complexity: 2^L-1, where L is the length of $LFSR_c$.
3. Analysis:
    - Effective for systems with low algebraic degrees in F_d .
    - For higher degrees, resistance improves due to increased equation complexity.

Attack 2: Extracting Sub sequences

1. Method:
    - Avoid guessing by analyzing independent sub-sequences of the key stream.
    - Use known properties of F_d to form and solve equations.
2. Complexity:
    - Requires 2^{57}key stream bits and 2^{63} equations for LILI−128.
    - Complexity depends on the number of taps in $LFSR_d$ and the algebraic degree of F_d .
3. Real-time Implementation:
    - Involves pre-computation of F_d 's low-degree multiples.
    - Real-time phase substitutes key stream bits into the equations.

Factors Affecting Resistance

The resistance of LILI ciphers to algebraic attacks depends on:

- Number of taps in $LFSR_d$.
- Algebraic degree of F_d .
- Lengths L and m of $LFSR_c$ and $LFSR_d$, respectively.

Comparison of Attacks

- Attack 1 has higher computational complexity but requires fewer key stream bits.

- Attack 2 is computationally less demanding but needs more key stream bits.

Conclusions

LILI systems' security can be improved by:

1. Increasing the algebraic degree of F_d .
2. Using larger LFSR lengths.
3. Introducing nonlinear functions to increase resistance to algebraic attacks.

Further research and optimization are necessary to balance performance and security effectively.

4. Conclusion

A brute force attack is a relatively new method of decryption, which is the problem of decrypting the attacked cipher by reduces polynomial equations and solutions to a multivariable product problem

An algebraic attack on a cryptosystem is equivalent to the problem of finding and solving a system of nonlinear equations on a finite field. In most of the attacks that are carried out on encryption systems, one of the basic assumptions is that the structure of the encryption algorithm is clear, and brute force attacks are no exception to this rule. In addition, we also need a number of execution keys.

Regarding the reasons for the effectiveness of brute force attack, the following can be mentioned:

1. Ability to run on a wide range of encryption systems.

2. Ability to do algorithm.

3. Simplicity in its implementation by computer.

4. The possibility of using secondary information about the password system.

5. It will be very efficient if existing computers improve.

A detailed examination of LILI stream ciphers and the analysis of various algorithms for solving systems of equations resulting from these attacks will lead to a better understanding of the efficiency and vulnerabilities of LFSR-based stream cipher systems.

The conclusions of this paper, based on findings and analyses of algebraic attacks on LFSR-based stream cipher systems, include several key points:

1. Efficiency of Algebraic Attacks on LFSRs

Analyses demonstrate that algebraic attacks can effectively compromise LFSR-based stream cipher systems. The primary reason for this success lies in the linear nature of LFSRs, which simplifies the resulting algebraic equations. Even in cases where nonlinear functions are employed, these functions can often be approximated by suitable linear ones, increasing the vulnerability of such systems to algebraic attacks.

2. Application of Numerical Algorithms

Numerical algorithms such as the XL (Extended Linearization) algorithm have proven highly effective in solving systems of multivariate polynomial equations. The XL algorithm and its accelerated variants can efficiently solve equations resulting from algebraic attacks, facilitating faster and more efficient decryption. These algorithms play a crucial role in advancing algebraic attacks by reducing computational time and increasing accuracy.

3. Analysis of LILI Stream Ciphers

The examination of LILI stream ciphers revealed that they are also susceptible to algebraic attacks. The structure, which relies on multiple LFSRs, allows for the formation of complex multivariate polynomial systems. However, using appropriate equation-solving methods can yield satisfactory results. These analyses highlight the need for a closer examination and enhanced security for this type of cipher.

4. The Need to Enhance Security in Stream Cipher Systems The results of this research underscore the necessity of improving the resistance of LFSR-based stream cipher systems against algebraic attacks. This can be achieved by employing more complex nonlinear functions, increasing the number of LFSRs, and leveraging new cryptographic techniques. Moreover, ongoing security analysis and evaluation are essential to ensure the robustness of these systems against attacks.

5. The Future of Algebraic Attacks

Given the success of algebraic attacks on stream cipher systems and the advancement of numerical algorithms, it is anticipated that such attacks will play a more significant role in analyzing cryptographic system security in the future. Thus, further research and the development of novel methods to counteract these types of attacks are essential.

References

[1] K. Alimohammadi, (2020). A secure key-aggregate authentication cryptosystem for data sharing in dynamic cloud storage. *Multimedia Tools and Applications*, 79, 2855–2872.
[2] M. Anzani, H. Haj Seyyed Javadi, V. Modirir, (2018). Key-management scheme for wireless sensor networks based on merging blocks of symmetric design. Wireless Networks, 24(8), 2867–2879.
[3] N. T. Courtois, (2001). The security of Hidden Field Equations (HFE). *Topics in Cryptology - CT-RSA 2001*, LNCS 2020, Springer-Verlag, 266 -281.
[4] N. T. Courtois (2003). Fast algebraic attack on stream ciphers with linear feedback. *Advances in Cryptology - Crypto 2003*, LNCS 2729, Springer-Verlag, 176-194.
[5] N. T. Courtois, W. Meier. (2003). Algebraic attacks on stream ciphers with linear feedback. *Advances in Cryptology - EUROCRYPT 2003*, LNCS 2656, Springer-Verlag, 345-359.
[6] N. T. Courtois, A. Klimov, J. Patarin, A. Shamir (2000). Efficient Algorithms for Solving Overdefined Systems of Multivariate Polynomial Equations. *Advances in Cryptology - EUROCRYPT 2000*, LNCS 1807, Springer-Verlag, 392-407.
[7] K. Diem (2004). The XL-algorithm and a conjecture from commutative algebra. *Advances in cryptology - ASIACRYPT 2004*, LNCS 3329, Springer-Verlag, 323-337.
[8] S. H. Erfani, H. Haj Seyyed Javadi, A. M. Rahmani (2015). A dynamic key management scheme for dynamic wireless sensor networks. *Security and Communication Networks*, 8(6), 1040-1049.
[9] Z. Eskandari, A. Ghaemi Bafghi (2020). Extension of Cube Attack with Probabilistic Equations and it's Application on Cryptanalysis of KATAN Cipher. *ISeCure*, 12(1), 1-12.
[10] M. Javanbakht, S. H. Erfani, H. Haj Seyyed Javadi, P. Daneshjoo (2014). Key Predistribution Scheme for Clustered Hierarchical Wireless Sensor Networks based on Combinatorial Designs. Security and Communication Networks, 7(11), 2003-2014.

[11] P. Hawkes, G. Rose (2004). The complexity of fast algebraic attacks onstream ciphers. *Advances in Cryptology - Crypto 2004*, LNCS 3152, Springer-Verlag, 390-406.
[12] M. Krause, F. ArmKnecht (2003). Algebraic attacks on combiners with memory. *Advances in Cryptology - Crypto 2003*, LNCS 2729, Springer-Verlag, 162-175.
[13] C. Li, B. Preneel (2019). Improved Interpolation Attacks on Cryptographic Primitives of Low Algebraic Degree. *Selected Areas in Cryptography 2019*, Available on IACR Cryptology ePrint Archive 2019: 812, 1-23.
[14] J. Pieprzyk, N. T. Courtois (2002). Cryptanalysis of block ciphers with overdened systems ofequations. *Advances in cryptology - ASIACRYPT 2002*, LNCS 2501, Springer-Verlag, 267-287.
[15] R. A. Mollin (2006). *An Introduction to Cryptography*. New York: Chapman and Hall/CRC.
[16] A. Najafi, M. Bayat, H. Haj Seyyed Javadi. (2018). Search over Encrypted Data: Functionalities and Challenges. *Biannual Journal Monadi for Cyberspace Security (AFTA)*, *7*(1), 21-44.
[17] M. Voros, (2007). *Algebraic attack on stream ciphers*. Master Thesis, Comenius University.