Model Design for a Reduced Variant of a Trivium Type Stream Cipher

Antonio Castro Lechtaler[1, 2], Marcelo Cipriano[1], Edith García[1], Julio Liporace[1], Ariel Maiorano[1], Eduardo Malvacio[1]

CriptoLab[1], EST, Instituto Universitario del Ejército, Buenos Aires, C1426AAA; FCE[1, 2], Universidad de Buenos Aires, Buenos Aires, C1120. República Argentina.

ABSTRACT.

We analyze the family of stream ciphers N-viums: Trivium and Bivium. We present the Trivium algorithm and its variants. In particular, we study the NLFSRs used in these generators, their feedback functions and their combination. Two reduced variants of these models are presented, labeled Toys. Finally, we delve into the open problems ingrained in these cryptosystems.

Keywords. LFSR, NLSFR, Trivium, Bivium, Trivium-Toy, Bivium-Toy.

1. INTRODUCTION.

The revolution of communications and technology has taken cryptology from the military and diplomatic realm into everyday life.

E-mailing, home banking, user authentication in social networks, mobile communications, and wireless technology have increased the requirements for confidentiality while data is transferred via insecure channels.

Some ciphering systems meet the requirements to protect data satisfactorily. However, they do not meet the increasing demand for higher transfer rates.

Because of the resources used and the processing power required, the existing algorithms lag behind the increasing needs for data transfer security.

Stream ciphers may prove suitable to use in portable devices. Their hardware adaptability turns them into feasible solutions, responding to the increasing demand and high transfer rate standards.

Stream Ciphers.

A perfect cryptosystem entails the capacity for an algorithm to cipher a message which can be deciphered only by the intended receiver.

Vernan and Mauborge created such a system in 1917 at the AT&T labs. In this system, the required key is as long as the length of the message. Both, transmitter and receiver must have the key which must be destroyed after use. Otherwise, security is jeopardized.

Because of this feature, the system is known as One-Time-Pad. The key must be random and is used for both processes: ciphering and deciphering. Hence, users need to share it at both ends. Cryptosystems under this particular secret key configuration belong to a class known as symmetric-key algorithms.

In 1949, Shannon demonstrated the invulnerability of this system by satisfying the requirements for perfect secrecy established by the rising field of Information Theory.

Nonetheless, two weaknesses become apparent, not in the algorithm itself, but in its application. On one hand, a problem arises in the generation of the secret key; and, on the other, in the security of key distribution.

A possible solution is to find a deterministic procedure to generate the key. Such a key would not be random, but pseudorandom, and shall meet additional requirements to be considered secure.

LFSRs and Non-LFSRs.

Currently, *Linear Feedback Shift Registers* (LFSRs) are used extensively to generate pseudorandom sequences with controlled period and linear complexity.

Research on LFSRs began in the 60s [6] and continued through several years. A significant number of results and applications have been produced: algorithm design, error control codes, and linear complexity analysis of binary sequences with the Berlekamp-Massey algorithm [7].

Because of their linearity, LFSRs alone are insecure. It is widely known that, when *2n-1* consecutive bits of an outbound sequence are known, it becomes predictable. Attempts to add linear complexity by combining LFSRs with, among other things, nonlinear functions have not met the desired standards yet.

Nonlinear Feedback Shift Registers (NLFSRs), a generalization of their linear counterparts, have been relegated for a long time. While LFSR theory is robust and well understood, many fundamental problems with NLFSRs remain unanswered.

One such problem is the determination of the period of outbound sequences in NLFSRs. In recent years, research has focused on nonlinear registers and stream ciphers using NLFSRs in some form. This is the case for the class TRIVIUM [1, 2], BIVIUM [10].

Our research focuses in the development of a new family of the TRIVIUM-BIVIUM stream cipher class, designated as *Toys*.

In our Toys, in which the sizes of the NLFSRs are reduced significantly, we have modified their taps while maintaining the original design principles.

With these models, observation in a constrained environment may foster more realistic research projects, as well as allow researchers to compare results within smaller samples and conduct tests in a reduced space.

In the future, the Toy family may help contribute in the development of a solid algebra involving NLFSRs, in particular for generators of the TRIVIUM-BIVIUM class.

2. FSR OVERVIEW.

An n-bit *feedback shift register* (FSR) is an n-bit length register with a feedback function:

$$f:\{0,1\}^n \rightarrow \{0,1\} \qquad (1)$$

where the feedback bit (at the tap positions of the register) or the output bit is of the form:

$$x_{n+t} = f(x_{n-1+t}, x_{n-2+t}, \ldots\ldots, x_t)\ (t \geq 0) \qquad (2)$$

For each step *t*, the register bits shift one position to the right and the taps are fed into the function and become the bit input for the following step. The *n* bits of the register constitute the state of the register at step *t*. The initial state is defined when *t=0*. The period of a FSR is the length of the largest cycle generated by the output sequence of the register.

If the feedback function is linear, i.e.:

$$f(x_{n-1}, x_{n-2}, .., x_0) = c_0x_0 + c_1x_1 + c_2x_2 + \ldots + c_{n-1}x_{n-1} (c_i \in \{0,1\}) \qquad (3)$$

We say that the registry is an **LFSR (Linear Feedback Shift Register)**. Otherwise, with a nonlinear feedback function, we have a **NLFSR (Nonlinear Feedback Shift Register)**.

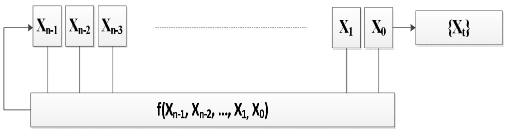

Fig.1: n-bit FSR Structure.

In the LFSR case, when the coefficients c_i belong to a primitive polynomial, the LFSR output sequence has a maximum length of 2^n-1, regardless of the chosen initial (non-trivial) state. The LFSR output sequences of maximum length are called *maximal sequences or m-sequences* [6]. If $2n-1$ output bits of an n-length LFSR are known, then the sequence becomes predictable using the Berlekamp-Massey algorithm [10].

NLFSRs are more robust to algebraic attacks. However, no systematic and efficient method is known to construct secure NLFSRs [3][4]. Furthermore, for a given nonlinear feedback function, it is difficult to predict the period of the output sequence.

A **stream cipher** is a symmetric ciphering system which takes a sequence of plaintext and a secret key, and operates on the plaintext, generally bit by bit with the **key bit stream,** generated by the secret key and the algorithm.

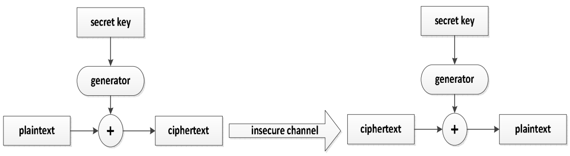

Fig.2: Stream Cipher Example.

The key bit stream must meet certain cryptologic security conditions, i.e.: the length of the sequence and the linear complexity must be sufficiently large, and the binary sequence must satisfy a series of pseudo-random tests [6].

3. TRIVIUM AND BIVIUM.

The stream algorithm TRIVIUM was designed by Christophe De Cannière and Bart Preneel. It was selected as a finalist algorithm in the e-STREAM Project [5]. It was designed to generate at least 2^{64} bits with the use of a 80-bit secret key and an initialization vector (IV) of also 80 bits.

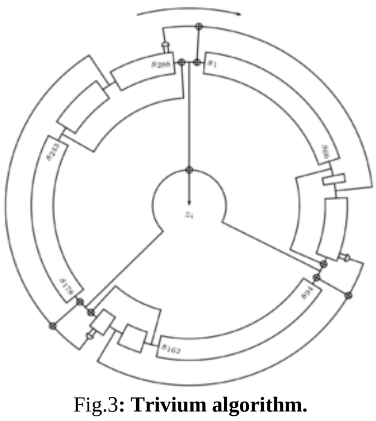

Fig.3**: Trivium algorithm.**

It consists of three combined NLFSRs. The first register controls the second, the second controls the third, and this last register controls the first.

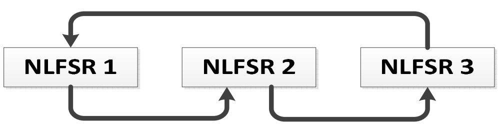

Fig.4: Trivium-Like Structure.

The core idea behind the design focuses on using the principles of block cipher construction in order to create equivalent components in stream ciphers.

The output consists of three combined non-linear shift registers of length *93*, *84*, and *111*, and where specific positions are selected to obtain a key bit stream. Whereas no efficient attack has been encountered to break this generator so far [8][9], its period remains undetermined and an open research problem.

A complete description is given by the following simple pseudo-code:

```
INPUT: s0, s1,…,s287 initial state, integer
n, si ∈{0,1}.
OUTPUT: binary sequence {kt}
    1. Initialization
       t1 ← s65 ⊕ s92
       t2 ← s161 ⊕ s176
       t3 ← s242 ⊕ s287

    2. While(t<n)do the following:
     2.1 kt ← t1⊕t2⊕t3
     2.2 t1 ← t1⊕s90⊗s91⊕s170
         t2 ← t2⊕s174 ⊗s175⊕s263
         t3 ← t3⊕s285 ⊗s286⊕s68
     2.3(s0,s1,…,s92)←(t3,s0,…,s91)
     (s93,s94,…,s176)←(t1,s93,…,s175)
     (s177,s178,…,s287)←(t2, s177,…,s285)

     3. Return {kt}
```

Note that ⊕ is the XOR operation and ⊗ the AND operation.

BIVIUM was designed by Hårvard Raddum to obtain a reduced sized version of TRIVIUM. It consists of two combined NLFSRs (while TRIVIUM has three) of lengths *93* and *84*.

Despite the improved security under specific attacks granted by this model, the results are not entirely satisfactory.

4. THE TOY MODEL

We present reduced variants of TRIVIUM and BIVIUM algorithms as a strategy to tackle the open problems discussed and the mathematical theory behind the behavior of NLFSRs. The reduced models (decimated by 3) are based on previous work by Yun Tian et al, who developed an extended model of the TRIVIUM structure [11]. We have named these models Toys, considering they are miniatures of the originals.

Let´s focus on the operation and pseudo-code of Trivium:

$$\begin{aligned} \mathbf{t_1} &\leftarrow \mathbf{s_{65} \oplus s_{90} \otimes s_{91} \oplus s_{92} \oplus s_{170}} \\ \mathbf{t_2} &\leftarrow \mathbf{s_{161} \oplus s_{174} \otimes s_{175} \oplus s_{176} \oplus s_{263}} \\ \mathbf{t_3} &\leftarrow \mathbf{s_{242} \oplus s_{285} \otimes s_{286} \oplus s_{287} \oplus s_{68}} \end{aligned} \quad \mathbf{(4)}$$

Shifting left one position the index numbers, is not difficult to find that each number has a factor of 3. Index numbers of active bits in this part are rewritten in Table 1.

Taps Original Trivium	Number Value	Taps Trivium Toy
66	66=3*22	$u_{1=}22$
69	69=3*23	$u_{2=}23$
162	162=3*54	$u_{3=}54$
171	171=3*57	$u_{4=}57$
243	243=3*81	$u_{5=}81$
264	264=3*88	$u_{6=}88$

Table 1: Taps of original Trivium and Toy.

Trivium-model stream cipher can be noted as

$$\{3u_1, 3u_2, 3n_1\}, \{3u_3, 3u_4, 3n_2\}, \{3u_5, 3u_6, 3n_3\} \quad (5)$$

where u_i (i = 1,…,6), n_i (i = 1,2,3) are parameters and

$$u_1 < u_2 < n_1 < u_3 < u_4 < n_2 < u_5 < u_6 < n_3. \quad (6)$$

In order to maintain the design of Trivium, the algorithm can be reduced (decimated by 3) into a *Toy* model as following:

$$\begin{aligned} \{u_1,u_2,n_1\} &\leftarrow \{21,22,30\} \\ \{u_3,u_4,n_2\} &\leftarrow \{53,56,58\} \\ \{u_5,u_6,n_3\} &\leftarrow \{80,87,95\} \end{aligned} \quad (7)$$

Length Trivium´s Registers	Number Value	Length Toy´s Registers
n_1=93	66=3*22	n_1=31
n_2=84	69=3*23	n_2=28
n_3=111	162=3*54	n_3=37

Table 2: Length of Trivium´s Registers and Toy´s Registers.

It is noted that every reduction of a model focuses on a quest for simplicity in its mathematical study and it is not meant to be used in operative information security environments.

We assume the following:

A1) *Property invariance after size reduction*: the reduced size structure of the models maintains the mathematical properties of the original model.

A2) *Computational complexity reduction:* The reduction in size contributes to a reduction of the problem, making the model more manageable under computational as well as algebraic considerations.

A3) *Property invariance after size increase:* In the case of identified patterns in the behavior and mathematical properties in the reduced model, they may be extrapolated to the original model.

These assumptions need to hold throughout the entire research. In case one of them does not hold or inconsistences among them are encountered, the procedure presented here ought to be revised.

Trivium-Toy.

The model consists of three NLFSRs *X, Y,* and *Z* of lengths ***31***, ***28*** and ***37*** with the following states:

$$\begin{aligned} &X(31): X_0, X_1, \ldots\ldots\ldots\ldots, X_{30} \\ &Y(28): Y_0, Y_1, \ldots\ldots\ldots\ldots, Y_{27} \\ &Z(37): Z_0, Z_1, \ldots\ldots\ldots\ldots, Z_{36} \end{aligned} \quad (8)$$

Being the feedback of each register, i.e. the bit input in each:

$$\begin{aligned} &X_0: Z_{21} \oplus Z_{36} \oplus Z_{35} \otimes Z_{34} \oplus X_{22} \\ &Y_0: X_{21} \oplus X_{30} \oplus X_{29} \otimes X_{28} \oplus Y_{25} \\ &Z_0: Y_{22} \oplus Y_{27} \oplus Y_{26} \otimes Y_{25} \oplus Z_{28} \end{aligned} \quad (9)$$

and the key bit stream:

$$K_t: \quad X_{21} \oplus X_{30} \oplus Y_{22} \oplus Y_{27} \oplus Z_{21} \oplus Z_{36} \quad (10)$$

Also, the cipher of the plaintext with the key bit stream is:

$$C_t = P_t \oplus K_t \quad (11)$$

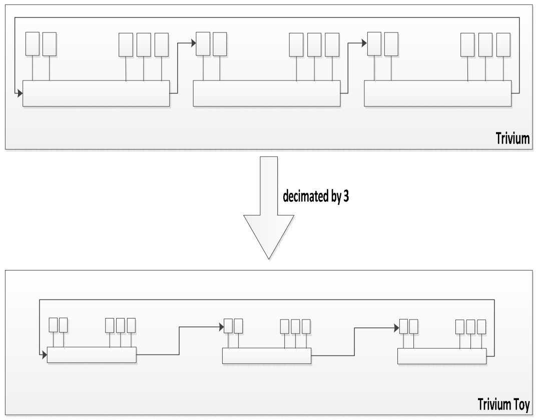

Fig.5: Trivium vs Trivium Toy

Pseudo-code of the Trivium is changed to a reduced form as following:

```
INPUT: s0, s1,…,s95 initial state, integer
n, si ∈ {0,1}.
OUTPUT: binary sequence {kt}
   1.Initialization.
       t1 ← s21⊕s30
       t2 ← s53⊕s58
       t3 ← s80⊕s95

   2. While(t<n)do the following:
       2.1.kt ← t1⊕t2⊕t3
       2.2.t1 ← t1⊕s28⊗s29⊕s55
           t2 ← t2⊕s56⊗s57⊕s87
           t3 ← t3⊕s93⊗s94⊕s22
       2.3(s0,s1,…,s30)←(t3,s0,…,s29)
          (s31,s32,…,s58)←(t1,s31,…,s57)
          (s59, s60,…,s95)←(t2, s59,…,s94)

   3. Return {kt}
```

Bivium-Toy

The model consists of two **NLFSRs** *X, and Y* of lengths ***31*** and ***28*** respectively with the following states:

$$\begin{aligned} &X(31): \quad X_0, X_1, \ldots\ldots\ldots\ldots, X_{30} \\ &Y(28): \quad Y_0, Y_1, \ldots\ldots\ldots\ldots, Y_{27} \end{aligned} \quad (12)$$

Being the feedback of each register:

$$\begin{aligned} &X_0: Y_{22} \oplus Y_{27} \oplus Y_{26} \otimes Y_{25} \oplus X_{22} \\ &Y_0: X_{21} \oplus X_{30} \oplus X_{29} \otimes X_{28} \oplus Y_{25} \end{aligned} \quad (13)$$

and the key bit stream:

$$K_t: \quad X_{21} \oplus X_{30} \oplus Y_{22} \oplus Y_{27} \quad (14)$$

The cipher process is the same as detailed in formulae **(11).**

Pseudo-code of this reduced cipher is given below:

```
INPUT: s0,s1,…,s58 initial state, integer
n., si ∈ {0,1}.
OUTPUT: binary sequence {kt}
   1.Initialization.
       t1 ← s21 ⊕ s30
       t2 ← s53 ⊕ s58

     2. While(t<n) do the following:
       2.1 kt ←t1⊕t2
```

```
    2.2 t1 ←t1⊕s28⊗s29⊕s55
        t2 ←t2⊕s56⊗s57⊕s22
    2.3(s0,s1,…,s30)←(t2,s0,…,s29)
       (s31,s32,…,s58)←(t1,s31,…,s57)
3. Return {kt}
```

5. CONCLUSIONS.

In this article we present the class of Trivium-Bivium random sequence generators using non-linear shift registers (NLFSR).

Because of their size, several research problems remain unanswered: patterns of behavior, algebraic properties, period lengths, and weak keys among others.

Under this framework, we present reduced sized variants of these generators for research and applications in cryptology, laying out the formulae of the feedback functions as well as the key bit streams. We assume that the properties identified in the reduced sized models would remain invariant in the original ones.

6. FURTHER RESEARCH.

The Toy family may foster additional research in the following areas:

- Search for length of the period or cycles.

-Distribution of taps and their changes to determine algebraic properties and personalization of N-viums.

- Algebraic analysis of the non-linear functions used in the models.

- Search for possible weak keys.

7. ACKNOWLEDGEMENTS

The financial support provided by Agencia Nacional para la Promoción Científica y Tecnológica (Project PICTO 11- PICTO 11-18621) is gratefully acknowledged.

8. REFERENCES

[1] De Canniére, C. and Preneel, B. "TRIVIUM A Stream Cipher Construction Inspired by Block Cipher Design Principles". In Workshop on Stream Ciphers Revisited (SASC2006), 2006.

[2] De Canniére, C. and Preneel, B. "TRIVIUM Specifications". eSTREAM, ECRYPT Stream Cipher Project, Report. 2008.

[3] Dubrova, E. "A List of Maximum-Period NLFSRs", Cryptology ePrint Archive, Report 2012/166, March 2012, http://eprint.iacr.org/2012/166

[4] Dubrova, E. "A scalable method for constructing Galois NLFSRs with period 2n −1 using cross-join pairs". Technical Report 2011/632, Cryptology ePrint Archive, November 2011. http://eprint.iacr.org/2011/632.

[5] eSTREAM: eSTREAM – The ECRYPT Stream Cipher Project: http://www.ecrypt.eu.org/stream/

[6] Golomb. "*Shift Register Sequences*". Aegean Park Press, 1982.

[7] Massey, J.L. "Shift-register synthesis and BCH decoding". IEEE Transactions on Information Theory 15, 1969.

[8] Maximov, A. and Biryukov, A. "Two Trivial Attacks on Trivium", Selected Areas in Cryptography, Lecture Notes in Computer Science, Vol.4876, Springer, 2007.

[9] McDonald, C. and Pieprzyk, C. "Attacking Bivium with MiniSat", Cryptology ePrint Archive,Report 2007/040, 2007.

[10] Raddum, H."Cryptanalytic Results on Trivium", eSTREAM, ECRYPT Stream Cipher Project, Report 2006/039, 2006.

[11] Yun Tian, Gongliang Chen, Jianhua Li: "On the Design of Trivium". IACR Cryptology ePrint Archive 2009.