Tatra Mt. Math. Publ. **45** (2010), 51–63
DOI: 10.2478/v10127-010-0005-x

TATRA MOUNTAINS
Mathematical Publications

ON THE CALCULATION OF THE LINEAR EQUIVALENCE BIAS OF JUMP CONTROLLED LINEAR FINITE STATE MACHINES

Cees J. A. Jansen

ABSTRACT. Jump controlled linear finite state machines were introduced several years ago as building blocks for stream ciphers that can efficiently be implemented in hardware and have intrinsically good side channel resistance. These constructions have found their way in concrete stream cipher designs. The bias in the distribution of linear relations of low degree in the key stream is important for the cryptographic strength of these stream ciphers. Recently, an algorithm was presented by the author to determine this bias. In this paper a new algorithm is introduced, that makes use of the properties of jump registers and has sub exponential order in the degree of the characteristic polynomial of the linear finite state machine.

1. Introduction

Jump controlled linear finite state machines are introduced in [2], [3], [4] as efficient building blocks for stream ciphers. As is discussed in [6], the bias in the distribution of linear relations of low degree is important for the cryptographic strength of stream ciphers based on irregularly jumping linear finite state machines (LFSMs). If this bias is too high, a key recovery attack could be feasible, which breaks the cipher. Research has shown that this Linear Equivalence Bias (LEB) depends solely on the characteristic polynomial of the LFSM. In [7] an efficient algorithm is given to determine the LEB for polynomials of degrees up to 30. This limit comes from the exponential memory usage of the algorithm. Although the described algorithm has proved its usefulness in practice, the quest for a more efficient algorithm has led to a new algorithm. The basis for this new algorithm is formed by the fact that in a jump controlled LFSM the coefficients

2010 Mathematics Subject Classification: Primary: 94A60, 68P25; Secondary 14G50.
Keywords: linear finite state machine, linear relation, characteristic polynomial, linear equivalence bias, jump register, irregular clocking.
Supported by the grant NIL-I-004 from Iceland, Liechtenstein and Norway through the EEA Financial Mechanism and the Norwegian Financial Mechanism.

of the linear relations in the output stream turn out to be symmetric Boolean functions (SBFs) of the jump control bits. It has been observed in [8] that the functions of the jump control signals in the matrices, given by eqations (5) and (8) in [7] are symmetric in their variables. These symmetries make evaluation of these functions quite simple and, hence, avoid the necessity of storing long truth table vectors. As a consequence, it turns out to be feasible to find short descriptions of the linear relation coefficients in terms of symmetric Boolean functions represented by i-bit vectors for each coefficient a_i. These symmetry properties and their implications are shown in [8]. Exploiting specific properties of these SBFs and jump controlled LFSMs paves the way for an extremely efficient algorithm to determine the LEB of high degree polynomials (≥ 120) in a matter of seconds on a common laptop.

In this paper we describe the new algorithm and illustrate that its time and memory orders are sub exponential with small enough constants to process polynomials of high degree. Concrete performance figures will demonstrate the usefulness of the new algorithm. Section 2 recaps the main results of [8]. The efficient algorithm to determine the LEB is developed in Section 3. In Section 4 a statistical experiment is described, which illustrates the sub exponential order of the new algorithm. The paper ends with some conclusions in Section 5.

2. Linear relation coefficients and symmetric Boolean functions

A symmetric Boolean function, SBF for short, is a function defined here as follows.

Definition 1. Let $\mathcal{S}_n(x_1, x_2, \ldots, x_n)$ be a function of n binary variables $x_1, \ldots$ $\ldots, x_n$, mapping binary n-tuples to a binary output value and let π denote a permutation on n elements. The function $\mathcal{S}_n$ is called symmetric if and only if

$$\forall_\pi \forall_{(x_1,x_2,\ldots,x_n)} \Big(\mathcal{S}_n\big(\pi(x_1, x_2, \ldots, x_n)\big) = \mathcal{S}_n(x_1, x_2, \ldots, x_n) \Big).$$

In [8] it is shown that the coefficients of the linear relations that occur in the output stream of jump controlled linear finite state machines are symmetric functions of their two-valued jump control variables. As the most important consequence, a table similar to Table 1 of [7] is now constructed. Instead of hexadecimal representations of truth tables in each entry, it suffices to place a representation of the SBF in each entry as shown in Table 1. In this table the SBFs are represented by their vectors $(\psi_{i+1}, \ldots, \psi_1, \psi_0)$ as hexadecimal integers $f^i = \psi_0 + 2\psi_1 + \cdots + 2^{i+1}\psi_{i+1}$. The series of column entries $1, 2, 6, \ldots$ is easily calculated. In the column of coefficient a_i let $f^i_0 = 1$, $f^i_1 = 2$, then

$$f^i_k = \big(2f^i_{k-1} \bmod 2^{i+2}\big) \oplus f^i_{k-1}, \qquad \text{for} \quad k > 1,$$

Table 1. Linear relation coefficients expressed as symmetric Boolean functions.

	a_0	a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8	a_9
c_0	1	0	0	0	0	0	0	0	0	
c_1	2	1	0	0	0	0	0	0	0	
c_2	2	2	1	0	0	0	0	0	0	
c_3	2	6	2	1	0	0	0	0	0	
c_4	2	2	6	2	1	0	0	0	0	
c_5	2	6	A	6	2	1	0	0	0	
c_6	2	2	E	A	6	2	1	0	0	
c_7	2	6	2	1E	A	6	2	1	0	
c_8	2	2	6	2	1E	A	6	2	1	
c_9	2	6	A	6	22	1E	A	6	2	$\cdots$
c_{10}	2	2	E	A	26	22	1E	A	6	
c_{11}	2	6	2	1E	2A	66	22	1E	A	
c_{12}	2	2	6	2	3E	2A	66	22	1E	
c_{13}	2	6	A	6	2	7E	AA	66	22	
c_{14}	2	2	E	A	6	2	FE	AA	66	
c_{15}	2	6	2	1E	A	6	2	1FE	AA	
c_{16}	2	2	6	2	1E	A	6	2	1FE	
c_{17}	2	6	A	6	22	1E	A	6	202	
c_{18}	2	2	E	A	26	22	1E	A	206	
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	

where $\oplus$ denotes bitwise modulo 2 addition (XOR). For a given characteristic polynomial of degree L and coefficients $c_0, c_1, \dots, c_L$ the linear relation coefficients are calculated by taking a linear combination of the SBFs, as given by eqation (1).

$$a_i = \sum_{k=i}^{L} c_k f_{k-i}^{i} \, . \tag{1}$$

3. Calculating the LEB using SBFs

The result of the previous section implies that the huge tables introduced in Section 3 of [7] are not needed to calculate the linear relation coefficients. But how does this affect the calculation of the LEB? At first sight, one might think that only $i+2$ values of a_i need to be calculated, as the values of SBFs depend only on the weight of their arguments. However, the LEB is defined as the number of the most often occurring linear relation. Does this mean that we still have to go through all 2^L combinations of jump control variables, evaluate all

SBFs and count the occurrences of all $(L+1)$-bit patterns? No, we can do much better by traversing the Binary Weight Triangle (BWT), a matrix structure that contains the values of the SBFs representing the linear relation coefficients $a_0, a_1, \ldots, a_L$, which are SBFs of $1, 2, \ldots, L+1$ variables, respectively, as given by (2). The notation $a_i(w)$ is used to denote the value of a_i for argument vectors with Hamming weight w.

$$
\begin{array}{|c|cccc|cc|}
\hline
\text{Weight} & a_0 & a_1 & \ldots & a_{L-2} & a_{L-1} & a_L \\
\hline
0 & a_0(0) & a_1(0) & \cdots & a_{L-2}(0) & a_{L-1}(0) & 1 \\
1 & a_0(1) & a_1(1) & \cdots & a_{L-2}(1) & a_{L-1}(1) & 1 \\
2 & & a_1(2) & \cdots & a_{L-2}(2) & a_{L-1}(2) & 1 \\
\vdots & & & \ddots & \vdots & \vdots & \vdots \\
L-1 & & & & a_{L-2}(L-1) & a_{L-1}(L-1) & 1 \\
L & & & & & a_{L-1}(L) & 1 \\
\hline
\end{array}
\tag{2}
$$

The BWT is computed directly from the coefficients of the characteristic polynomial of degree L using Table 1. The computational effort is of order L^2. Clearly, the BWT contains all linear relations by its definition. For the explanation of the BWT traversing algorithm, the following definitions are given.

Weight set: A weight set $E_i = \{w_0, w_1, \ldots\}$, $i = 0, 1, \ldots, L-1$. A set of all weights w_k (k is an enumeration variable) of argument vectors, where argument vectors are vectors of values of jump control variables. All argument vectors having weights in a set result in the same value of a linear relation coefficient a_i. In general, weight sets do not contain all weights that result in the same value of a coefficient a_i.

Ensemble of weight sets: $\mathcal{E}_i = \left\{E_i^0, E_i^1, \ldots\right\}$, $i = 0, 1, \ldots, L-1$. A set of all sets E_i^k, with k an enumeration variable.

Extended weight set: $\widetilde{E}_i^j = \{w_k\} \cup \{w_k + 1\} = E_i^j \cup (E_i^j + 1)$. The union of the set of weights $\{w_k\}$ and the same set with all its weights increased by one. This extension arises when an additional argument variable is considered going from a_i to a_{i+1}, implying that weights stay the same if this variable has the value 0, and weights are increased by one otherwise.

Conditional splitting: Extended weight sets $\widetilde{E}_i$ are split into two successor sets E'_{i+1} and E''_{i+1}, one with weights resulting in $a_{i+1} = 0$ and one with weights resulting in $a_{i+1} = 1$, according to the values in the BWT. If for all weights in some $\widetilde{E}_i^j$, the corresponding a_{i+1} assumes only one value, then $E_{i+1}^j = \widetilde{E}_i^j$, else two disjoint sets E_{i+1}^k and $E_{i+1}^{k'}$ result.

Multiplicity set: A multiplicity set $M_i = (m_0, m_1, \ldots)$, $i = 0, 1, \ldots, L-1$. A set of multiplicities m_k, where k is an enumeration variable, corresponding to weights of argument vectors. In order to determine the LEB, which is the multiplicity of the linear relation that occurs most, the m_k count the number of argument vectors that result in one and the same linear relation with a coefficient a_i of some binary value for all weights given by its weight set E_i. The multiplicity of a weight w_k in a set E^l_{i+1} is equal to the sum of the multiplicities of weights w_k and $w_k - 1$ of the preceding weight set E^j_i. The sum of all multiplicities in all multiplicity sets corresponding to the weight sets in an ensemble sum up to 2^{i+1}, the total number of value combinations of the $(i + 1)$ jump control variables.

Traversing the BWT: Starting with $i = 0$ and proceeding from left to right in the Binary Weight Triangle until $i = L-1$, successive ensembles $\mathcal{E}_i$ are determined recursively by the three steps:

(1) Extending all weight sets in the ensemble.

(2) Conditionally splitting all weight sets in the ensemble using the values of the coefficients $a_{i+1}(k)$ in the BWT and adding the resulting sets to $\mathcal{E}_{i+1}$.

(3) Update the multiplicities in the corresponding multiplicity sets.

From the above definitions it is clear that the number of weight sets in $\mathcal{E}_{i+1}$ can be any number ranging from the number of weight sets in $\mathcal{E}_i$ up to twice that number. However, a doubling of weight sets only occurs if and only if some a_i is a linear or affine function of all its variables, which, as a consequence of the jump mechanism, does not occur, except for a_{L-1}. Also, the number of weight sets stays the same if and only if some a_i is a constant function, which occurs for a_L. Next, consider the weight sets in all ensembles as nodes in a directed graph, and draw edges between nodes if and only if the corresponding weight sets are related by set extension and conditional splitting. The resulting graph is an (incomplete) binary tree containing all linear relations that occur for the characteristic polynomial, used to construct the BWT. Now the following algorithm is evident.

Algorithm 1.

```
1. Calculate the BWT of $C(x)$.
2. Traverse the BWT.
3. Create a binary tree with linear relations.
```

Figure 1 shows a partial result $(i = 0, \ldots, 6)$ of Algorithm 1 for the polynomial $x^{14}+x^{13}+x^{12}+x^{11}+x^9+x^7+x^5+x^3+x^2+x+1$, 75267_O in octal notation, that was used in version 2 of the Pomaranch stream cipher [6]. From Figure 1

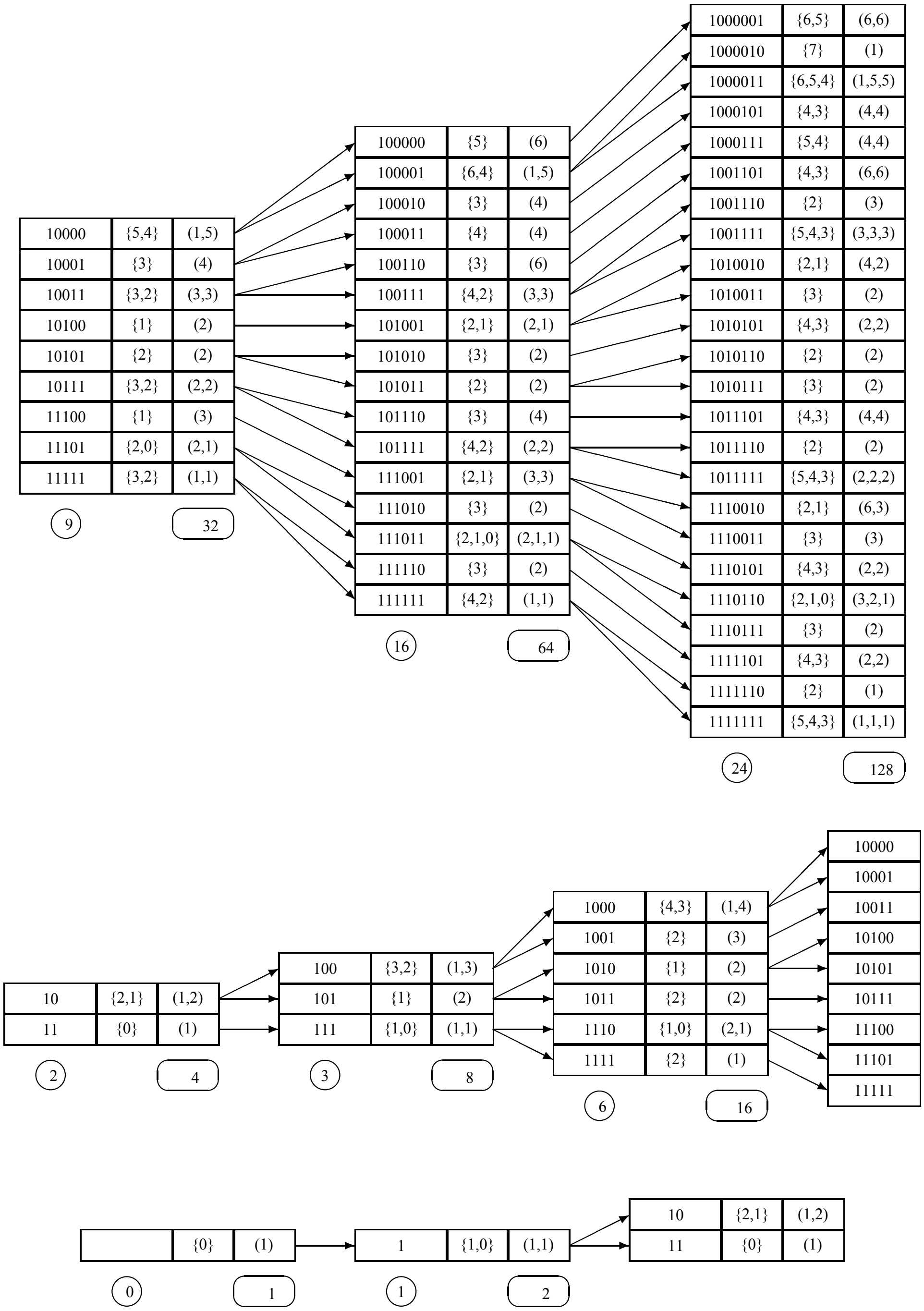

Figure 1. Linear relations from the binary weight triangle of polynomial 75267.

two things are immediately clear. Firstly, it can be seen that the order is quadratic with the number of linear relations, and therefore the number of weight sets grows exponentially. Secondly, multiple copies of weight sets occur in ensembles. A dramatic improvement of the algorithm is achieved if we apply the convention that weight sets cannot occur more than once in an ensemble. However, in order to count the number of linear relations a Linear Relations Counter (LRC) is maintained for each weight set in an ensemble. Moreover, as a consequence of this weight set unicity, the processing of the multiplicity sets needs to be adapted, because identical copies of weight sets may have different multiplicity sets associated with them. Finally, weight sets can have more than one predecessor sets in this case, making it necessary to select one out of several multiplicity sets.

Linear relations counter: A linear relations counter R_i is the value that indicates the number of linear relations associated with a weight set E_i. Splitting an extended set results in two sets with identical LRC values. Also, if a weight set E_{i+1} has more than one preceding weight sets $E_i^{k_1}$, $E_i^{k_2}, \ldots$, the resulting LRC value is the sum of the corresponding preceding LRC values, i.e., $R_{i+1} = R_i^{k_1} + R_i^{k_2} + \cdots$. In this way, the total number of Linear Relations is the sum of all LRCs in $\mathcal{E}_L$. By keeping track of which set originates from a preceding set with $a_i(w_k) = 0$ and $a_i(w_k) = 1$, the actual values of the linear relation coefficients are obtained.

Multiplicity set: The previous definition is modified as follows. The multiplicity of a weight w_k in a set E_{i+1}^l is equal to the sum of the multiplicities of weights w_k and $w_k - 1$ of the preceding weight set E_i^j. If more than one weight set precedes the current weight set, then the multiplicities of the weight set with the highest total multiplicity over all weights in the weight set are taken. This is allowed, because only the highest total multiplicity is of interest for the LEB. By examining all weight sets for $i = L$ the weight set with the highest total provides the LEB.

The following algorithm shows the steps resulting from Algorithm 1 with the weight set unicity modifications discussed above.

Algorithm 2.

```
1. Calculate the BWT of $C(x)$.
2. Traverse the BWT with modified update of multiplicities.
3. Remove multiple copies of weight sets.
4. Create a graph with weight sets as nodes and extension relations as edges.
```

As a further improvement to the algorithm, if the degree of the characteristic polynomial is L, then $a_L = 1$ and need not be considered in calculating the

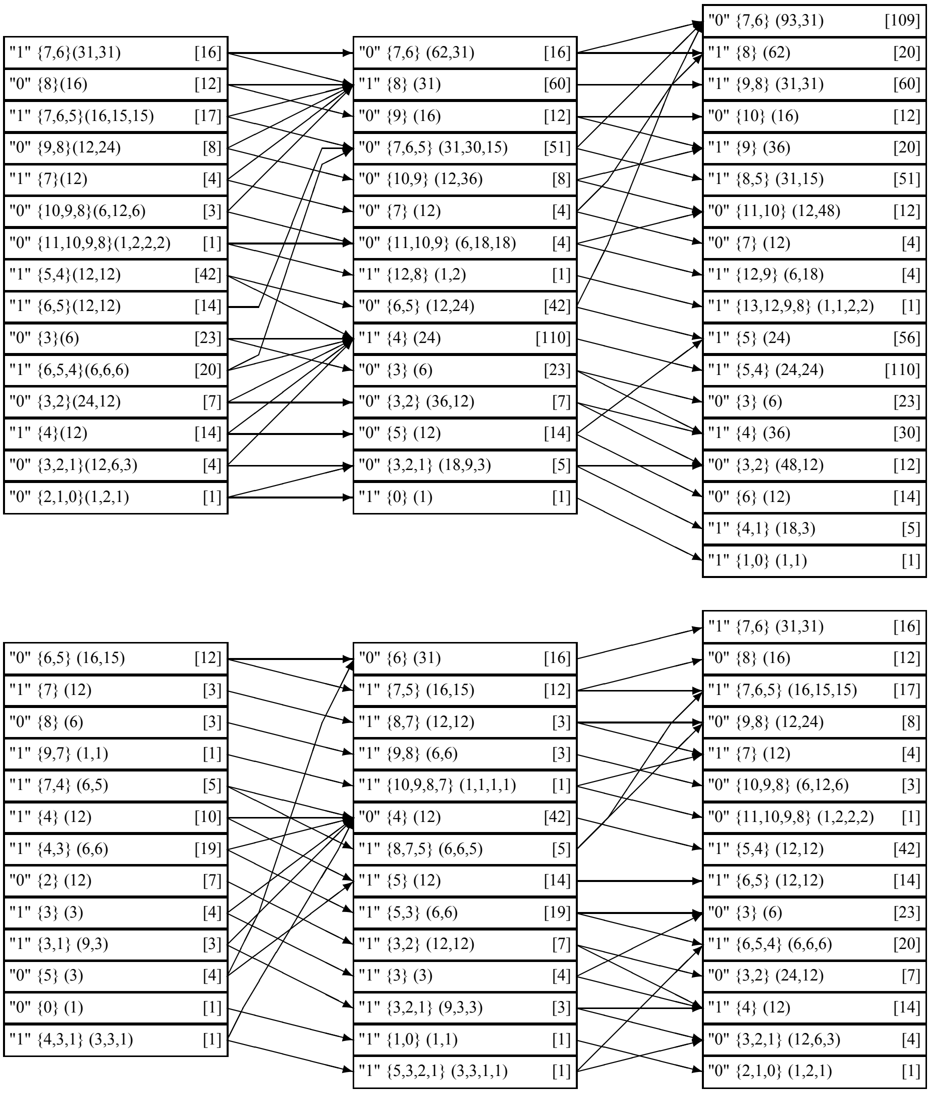

FIGURE 2. Algorithm 2 output stages 8–12 for polynomial 75267.

LEB. Taking *irreducible* polynomials of degree $L > 1$ implies that $c_0 = 1$ and also $\sum c_i = 1$, so that $a_0 = 1$. Also, from the Doubling Rule (see [6]) it follows that linear relations with $a_{L-1} = 0$ and $a_{L-1} = 1$ occur equally often. Consequently, a_L, a_{L-1} and a_0 need not be considered. Figures 2 and 3 show the

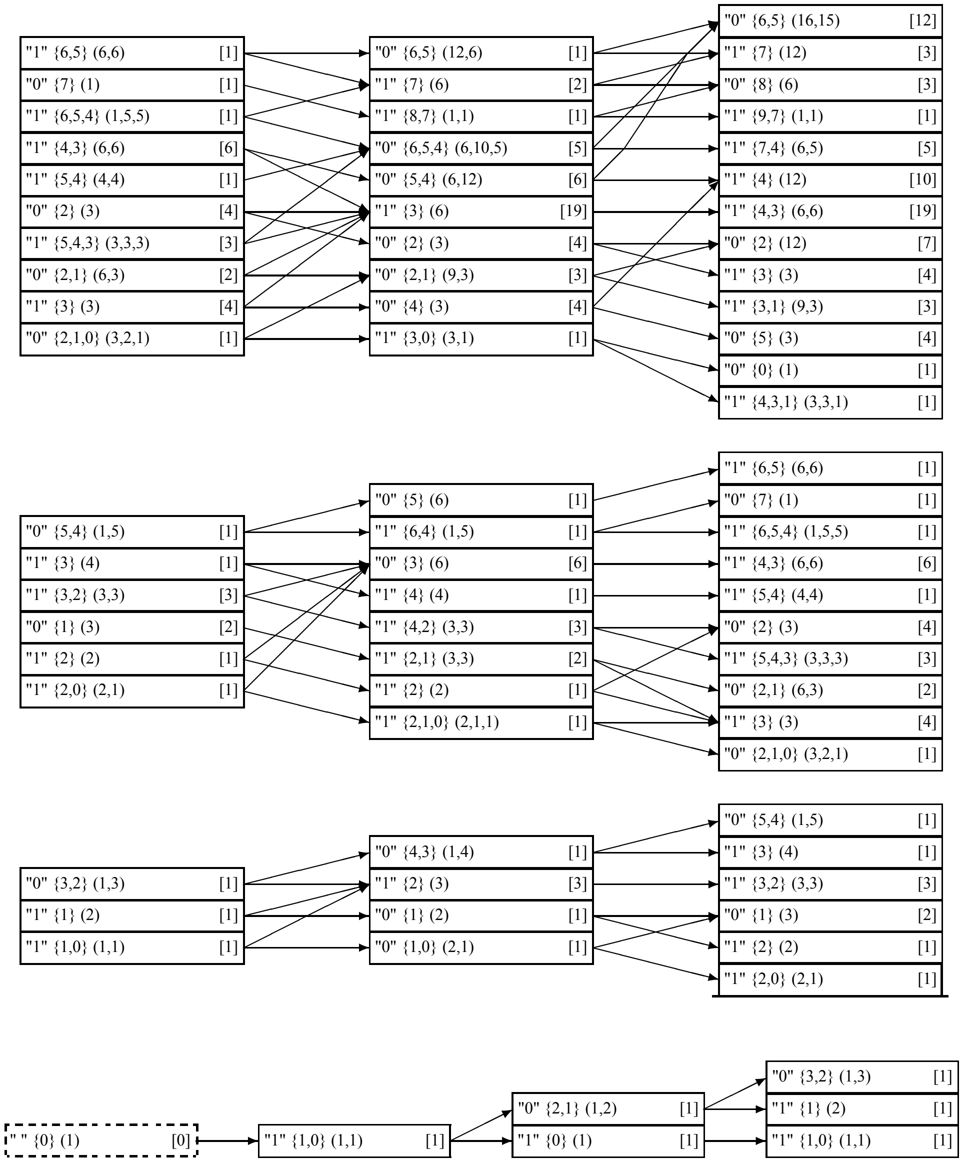

FIGURE 3. Algorithm 2 output stages 0–8 for polynomial 75267.

resulting graph for the polynomial 75267_O up to $\mathcal{E}_{L-2}$. Taking the maximum over all sums of multiplicities in multiplicity sets in $\mathcal{E}_{12}$ identifies the value 124 $(= 93 + 31)$ as the LEB. Summing up all all LRC values in this ensemble yields

544 linear relations at that stage, in agreement with the total of 1088 linear relations for this polynomial. Backtracking through the graph reveals the linear relations 1000011000100(01) and 1000011000100(11) that occur 124 times.

4. The order of Algorithm 2

It is not straightforward to assess the order of the described algorithm on theoretical grounds, as the relations between the number of weights in a weight set and the number of weight sets in an ensemble on the one hand, and the degree of the characteristic polynomial L on the other hand, seem quite complex. In order to get a realistic impression of the complexity behaviour of Algorithm 2, a statistical experiment was set up. The algorithm was run for irreducible polynomials of various degrees from 5 through 120, that were taken at random from [1]. For certain values of the degree L of the polynomials ($L = 10, 20, \ldots, 100, 120$) ten different polynomials were processed with Algorithm 2 and the $\mathcal{E}_{L-2}$ determined. For these degrees the geometric means of the number of weight sets in the ensembles $\mathcal{E}_{L-2}$ were calculated. These values were then plotted in a graph. The results are depicted in Figure 4.

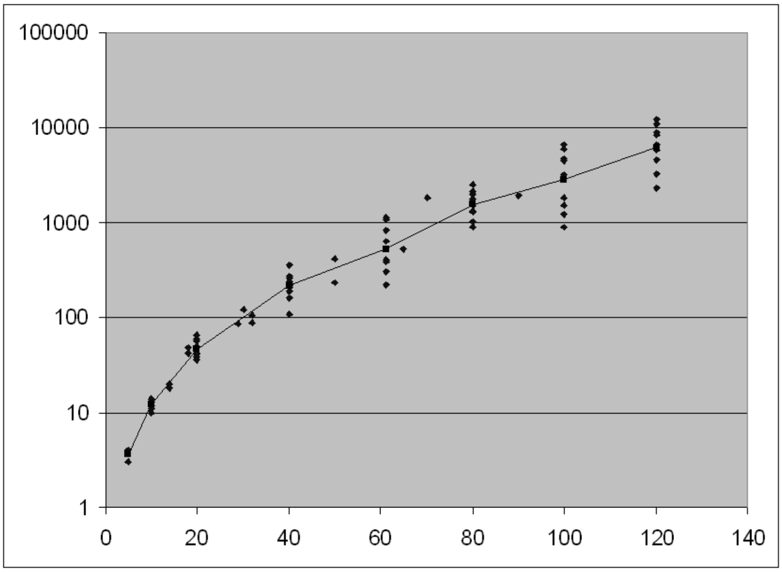

FIGURE 4. Cardinalities of ensembles $\mathcal{E}_{L-2}$ from the statistical experiment.

Table 2. Statistical experiment weight sets of ensemble $\mathcal{E}_{L-2}$.

		$\mathcal{E}_{L-2}$			
deg L	# pol	min	avg	max	Calc $\mathcal{E}_{L-2}$
5	3	3	3.634	4	7.83
10	10	10	12.34	14	16.76
14	2	18		20	
18	2	42		48	
20	10	36	46.88	66	47.54
29	1	85			
30	1	120			
32	2	87		105	
40	10	109	213.52	353	198.29
50	2	231		420	
61	8	222	533.62	1122	605.86
65	1	524			
70	1	1813			
80	10	903	1566.28	2518	1402.30
90	1	1928			
100	10	887	2863.76	6567	3036.15
120	10	2297	6167.97	12017	6064.77

From this figure it can already be seen that the complexity order is sub exponential. This is further supported by the fact that an almost straight line is obtained if one plots the values of $\log\log\mathcal{E}_{L-2}$ as a function of $\log L$. Next, least squares curve fitting was used to obtain the best matching straight line. The number of weight sets in the ensemble $\mathcal{E}_{L-2}$ as a function of the polynomial degree L, obtained as best match, is given by (3) below.

$$\mathcal{E}_{L-2} = \exp(0.9911L^{0.4540}). \tag{3}$$

Table 2 lists all obtained data with the rightmost column containing the calculated values using the least squares fit parameters. For comparison, the best exponential curve fit was also calculated from the same data. The best exponential curve fit is given by (4).

$$\mathcal{E}_{L-2} = 25.76(1.04875)^{L}. \tag{4}$$

The results are shown in Figure 5, with $\log\log\mathcal{E}_{L-2}$ on the y-axis and L on the x-axis in a log scale. This figure illustrates the sub exponential behaviour of Algorithm 2.

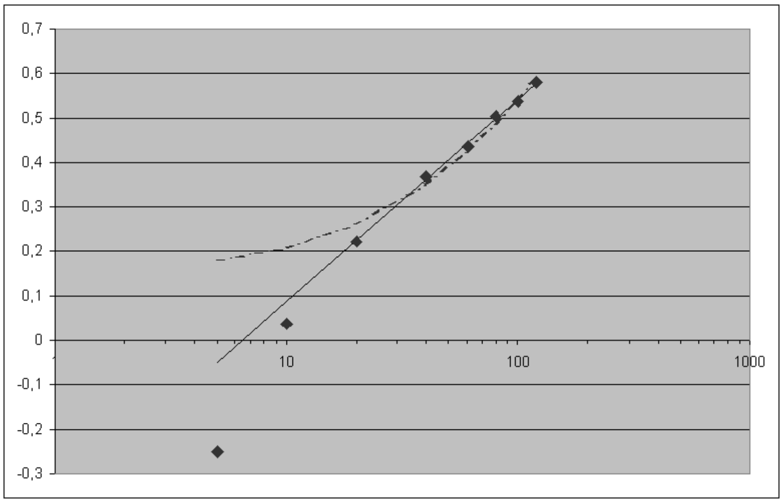

FIGURE 5. Least squares fitted curves of $\log\log \mathcal{E}_{L-2}$ versus $\log L$.

5. Conclusions

This paper presents a new and efficient algorithm to solving the problem of finding the linear equivalence bias of jump controlled linear finite state machines. A statistical experiment provides evidence that the algorithm is sub exponential in the degree of the characteristic polynomial of the linear finite state machine. In particular, implementing Algorithm 2 in software on a PC to calculate the linear equivalence bias of high degree polynomials is quite straightforward. Polynomials of degree 120 are processed in a matter of seconds on a standard laptop.

Future research includes generalizations in two directions: 1) general clock control in stead of jumping, and 2) extended linear relation, i.e., linear relations of length greater than the degree of the characteristic polynomial.

REFERENCES

[1] CHABAUD, F.: *Polynomials over Galois fields,* `http://fchabaud.free.fr/English/default.php?COUNT=1&FILE0=Poly`.

[2] JANSEN, C. J. A.: *Modern stream cipher design: A new view on multiple clocking and irreducible polynomials,* in: Actas de la VII Reunión Española sobre Criptología y Seguridad de la Información, Volume Tomo I (S. González, C. Martínez, eds.), Servicio de Publicaciones de la Universidad de Oviedo, 2002, pp. 11–29.

[3] JANSEN, C. J. A.: *Streamcipher design: Make your LFSRs jump!* in: ECRYPT The State of the Art of Stream Ciphers (SASC), Workshop Record, Network of Excellence in Cryptology, 2004, pp. 94–108.

[4] JANSEN, C. J. A.: *Stream cipher design based on jumping finite state machines,* Cryptology ePrint Archive, Report 2005/267, `http://eprint.iacr.org/2005/267/`.

[5] JANSEN, C. J. A.—HELLESETH, T.—KHOLOSHA, A.: *Cascade jump controlled sequence generator and Pomaranch stream cipher (Version 3).* eSTREAM, ECRYPT Stream Cipher Project, End of 2nd Phase, March 2007,
`http://www.ecrypt.eu.org/stream/p3ciphers/pomaranch/pomaranch_p3.pdf`.

[6] JANSEN, C. J. A.—KHOLOSHA, A.: *Countering the correlation attack on Pomaranch.* eSTREAM, ECRYPT Stream Cipher Project, Phase 1, October 2005,
`http://www.ecrypt.eu.org/stream/papersdir/070.pdf`.

[7] JANSEN, C. J. A.: *Linear relations in irregularly clocked linear finite state machines.* in: 29th Symposium on Information Theory in the Benelux, Leuven (L. Van der Perre et al., eds.), Werkgemeenschap voor Informatie- en Communicatietheorie IMEC, Leuven, 2008, pp. 223–229.

[8] JANSEN, C. J. A.: *Linearities in cascade jump controlled stream ciphers,* in: Proc. of the NATO, Advanced Research Workshop on Enhancing Cryptographic Primitives with Techniques from Error Correcting Codes (B. Preneel et al., eds.) Veliko Tarnovo, Bulgaria, 2008, IOS Press, Amsterdam, 2009, pp. 179–191.

Received May 20, 2010

DeltaCrypto BV
Riebeeckstraat 10
5684ej Best
THE NETHERLANDS
E-mail: cees.jansen@deltacrypto.nl