Correlation Properties of Combiners with Memory in Stream Ciphers

(Extended Abstract) [1)]

Willi Meier [2)] Othmar Staffelbach [3)]

[2)] HTL Brugg-Windisch
CH-5200 Windisch, Switzerland

[3)] GRETAG, Althardstrasse 70
CH-8105 Regensdorf, Switzerland

Abstract

In stream cipher design pseudo random generators have been proposed which combine the output of one or several LFSRs in order to produce the key stream. For memoryless combiners it is known that the produced sequence has correlation to sums of certain LFSR-sequences whose correlation coefficients c_i satisfy the equation $\sum_i {c_i}^2 = 1$. It is proved that a corresponding result also holds for combiners with memory.

If correlation probabilities are conditioned on side information, e.g. on known output digits, it is shown that new or stronger correlations may occur. This is exemplified for the summation cipher with two LFSRs where such correlations can be exploited in a known plaintext attack. A cryptanalytic algorithm is given which is shown to be successful for LFSRs of considerable length and with arbitrary feedback connection.

1 Introduction

Cryptographic transformations are usually designed by appropriate composition of nonlinear functions. In stream cipher design such functions have been applied to combine the output of linear feedback shift registers (LFSRs) in order to produce the key stream. In this design the combining functions should not leak information about the individual LFSR-sequences into the key stream. For this purpose the concept of correlation immunity has been introduced in [7,5] in order to prevent divide and conquer correlation attacks. For nonlinear combiners there is a tradeoff between the nonlinear order of the Boolean function and its order of correlation immunity. As has been pointed out in [5], this tradeoff can be avoided if the function is allowed to have memory.

[1)] Full paper to appear in the *Journal of Cryptology*.

I.B. Damgard (Ed.): Advances in Cryptology - EUROCRYPT '90, LNCS 473, pp. 204-213, 1991.

For a memoryless combiner the output always has correlation to certain linear functions of the inputs, and the "total" correlation is independent of the combining function. In fact it has been shown in [3] that the sum of the squares of the correlation coefficients is always 1. If such a combiner is applied to the output of LFSRs, there result correlations to sums of certain LFSR-sequences whose correlation coefficients c_i satisfy

$$\sum_i c_i^2 = 1 \tag{1}$$

Choosing the combiner to be correlation immune of some order means that certain of these c_i's vanish. In particular, to prevent divide and conquer, one can achieve that there is no correlation to sums of outputs of only few LFSRs. However by (1) there must be stronger correlation to certain other sums of LFSR-sequences, which have to be considered with regard to the cryptanalytic algorithms described in [2]. A first goal of the present paper is to show that a result similar to (1) remains valid for combiners with memory. This implies that memory does not affect the total correlation. In fact the total correlation is independent of the combining functions as for memoryless combiners.

The correlation coefficients in (1) are derived from unconditioned probabilities. However it is often the case that the cryptanalyst has access to side information, e.g. he may know portions of the output sequence. In fact, if correlation is conditioned on the output, new or much stronger correlations may occur. This is exemplified for the basic summation combiner with two inputs, where the resulting correlations can be cryptanalytically exploited in a known plaintext attack on the basic summation cipher with two LFSRs. A cryptanalytic algorithm is given which is shown to be successful for LFSRs of considerable length and with arbitrary feedback connection. As a consequence for the design of summation ciphers, it is recommended to take several LFSRs of moderate length rather than just few long LFSRs.

2 Basic Summation Combiner

The summation combiner has been considered in [5] as a main example of a combiner with memory, in order to generate cryptographically strong binary sequences out of given (cryptographically weak) sequences. It is based on integer addition which, when viewed over $GF(2)$, defines a nonlinear function with memory whose correlation immunity is maximum.

To describe this combiner, consider two binary sequences $\mathcal{A} = (a_0, a_1, \ldots)$ and $\mathcal{B} = (b_0, b_1, \ldots)$. For every n the first n digits are viewed as the binary representation of an integer, i.e. $a = a_{n-1}2^{n-1} + \cdots + a_1 2 + a_0$ and $b = b_{n-1}2^{n-1} + \cdots + b_1 2 + b_0$. Then the integer sum $z = a + b$ defines the first n digits of the resulting sequence $\mathcal{Z} = (z_0, z_1, \ldots)$. If $\mathcal{A}$ and $\mathcal{B}$ are semi-infinite then $\mathcal{Z}$ is also defined as a semi-infinite sequence. The digit z_j is recursively computed by

$$z_j = f_0(a_j, b_j, \sigma_{j-1}) = a_j + b_j + \sigma_{j-1} \tag{2}$$

$$\sigma_j = f_1(a_j, b_j, \sigma_{j-1}) = a_j b_j + a_j \sigma_{j-1} + b_j \sigma_{j-1} \tag{3}$$

where in (2) σ_{j-1} denotes the carry bit, and $\sigma_{-1} = 0$. For this basic summation combiner it is possible to give an explicit description of all correlations that exist between its output and linear functions of the inputs.

If $\mathcal{A}$ and $\mathcal{B}$ are considered to be independent and uniformly distributed sequences of random variables, the output sequence $\mathcal{Z} = (z_0, z_1, z_2, \ldots)$ is also uniformly distributed. Moreover z_j is independent of a_j, b_j and the sum $a_j + b_j$. However it can be shown (cf. [4]) that z_j is correlated to $a_j + b_j + a_{j-1}$ and $a_j + b_j + b_{j-1}$, with probability

$$p = P(z_j = a_j + b_j + a_{j-1}) = P(z_j = a_j + b_j + b_{j-1}) = 0.75 \tag{4}$$

This means that the corresponding correlation coefficient (cf. [3]) amounts to $c = 2p - 1 = 0.5$. More generally, it is proved in [4] that for every i, $1 \le i \le j$, there are correlations to $N = 2^{i+1} - 2$ linear functions of the form

$$s = \sum_{k=j-i}^{j} \alpha_k a_k + \beta_k b_k, \tag{5}$$

and the corresponding correlation coefficients c_h satisfy

$$\sum_{h=1}^{N} {c_h}^2 = 1 - \frac{1}{2^i} \tag{6}$$

Note that the right hand side of (6) tends to 1 as i tends to ∞. This means that the total correlation for the basic summation combiner approaches 1, similar to the case of memoryless combiners. This remarkable fact can be extended to completely general combiners with 1 bit memory.

3 General Combiner with One Bit Memory

A general combiner with 1 bit memory is described by two balanced functions f_0 and f_1 as follows:

$$z_j = f_0(x_{1j}, \ldots, x_{nj}, \sigma_{j-1}) \tag{7}$$

$$\sigma_j = f_1(x_{1j}, \ldots, x_{nj}, \sigma_{j-1}) \tag{8}$$

Hereby σ_j denotes the state of the memory, and the inputs $\mathcal{X}_m = (x_{m0}, x_{m1}, x_{m2}, \ldots)$, $1 \le m \le n$, are assumed to be independent and uniformly distributed sequences of random variables.

In order to study the correlation properties of this combiner we investigate correlations of the combining functions $f_0, f_1 : GF(2)^{n+1} \to GF(2)$ to linear functions. The correlation of an arbitrary function $f : GF(2)^{n+1} \to GF(2)$ to the linear function $L_{\mathbf{w}}(\mathbf{x}) = \mathbf{w} \cdot \mathbf{x}$ ($\mathbf{w}, \mathbf{x} \in GF(2)^{n+1}$) is computed by the Walsh transform

$$F(\mathbf{w}) = \sum_{\mathbf{x} \in GF(2)^{n+1}} f(\mathbf{x})(-1)^{\mathbf{w} \cdot \mathbf{x}} \tag{9}$$

Here, in connection with Walsh transforms, all Boolean functions are considered with values +1 and −1 (i.e. $f(\mathbf{x})$ is replaced by $(-1)^{f(\mathbf{x})}$). Then the correlation between f and $L_{\mathbf{w}}$ (cf. [3]) is computed as

$$c(f, L_{\mathbf{w}}) = \frac{F(\mathbf{w})}{2^{n+1}} \tag{10}$$

For the combining functions $f_0(\mathbf{x},\sigma)$ and $f_1(\mathbf{x},\sigma)$, $\mathbf{x} \in GF(2)^n$, we distinguish between correlation to linear functions of the form

$$L(\mathbf{x},\sigma) = \mathbf{w}\cdot\mathbf{x} \tag{11}$$

and

$$L(\mathbf{x},\sigma) = \mathbf{w}\cdot\mathbf{x} + \sigma \tag{12}$$

For the function f_0 the corresponding correlation coefficients are given by $c_0(\mathbf{w}) = F_0(\mathbf{w},0)/2^{n+1}$ and $c_1(\mathbf{w}) = F_0(\mathbf{w},1)/2^{n+1}$, where F_0 denotes the Walsh transform of f_0. In order to distinguish between linear functions of the form (11) and (12) we introduce

$$C_0{}^2 = \sum_{\mathbf{w}\in GF(2)^n} c_0(\mathbf{w})^2, \quad C_1{}^2 = \sum_{\mathbf{w}\in GF(2)^n} c_1(\mathbf{w})^2 \tag{13}$$

In a similar way, for the function f_1, we introduce $d_0(\mathbf{w}) = F_1(\mathbf{w},0)/2^{n+1}$, $d_1(\mathbf{w}) = F_1(\mathbf{w},1)/2^{n+1}$ and

$$D_0{}^2 = \sum_{\mathbf{w}\in GF(2)^n} d_0(\mathbf{w})^2, \quad D_1{}^2 = \sum_{\mathbf{w}\in GF(2)^n} d_1(\mathbf{w})^2 \tag{14}$$

Then by Parseval's theorem

$$C_0{}^2 + C_1{}^2 = 1 \quad \text{and} \quad D_0{}^2 + D_1{}^2 = 1 \tag{15}$$

In this framework we can determine all correlations of the output z_j of the general combiner (7,8) to linear functions of the form

$$s = \sum_{k=j-i}^{j} \sum_{m=1}^{n} w_{mk} x_{mk}. \tag{16}$$

There are $N = 2^{(i+1)n}$ such functions. As a generalization of equation (6) we obtain the following theorem, proved in [4].

Theorem 1 *Let $1 \leq i \leq j$. Then the output digit z_j of the general combiner with one bit memory is correlated to linear functions $s_1, s_2, \ldots, s_N$ of the form (16), and the corresponding correlation coefficients c_h satisfy*

$$\sum_{h=1}^{N} c_h{}^2 = C_0{}^2 + C_1{}^2(1 - (D_1{}^2)^i). \tag{17}$$

Theorem 1 is our main result for general combiners with 1 bit memory and has several implications on the design of stream ciphers using combiners with memory. First observe that the sum of the squares of the correlation coefficients (17) converges to 1 as i tends to ∞, except in the (singular) case $D_0 = 0$, where the limit is ${C_0}^2$.

If the input sequences $\mathcal{X}_m = (x_{m0}, x_{m1}, x_{m2}, \ldots)$, $1 \leq m \leq n$, to a combiner with memory are generated by LFSRs, the correlation of z_j to linear functions of the form (16) leads to correlation to sums of LFSR–sequences. By (16) these sums are given by $s = \sum_{m=1}^{n}(\sum_{k=j-i}^{j} w_{mk}x_{mk})$. Note that for each m the inner sum $s_m = \sum_{k=j-i}^{j} w_{mk}x_{mk}$ is in fact a phase of the m–th LFSR. If certain of these s_m's vanish, a divide and conquer correlation attack is possible. To prevent divide and conquer *maximum order correlation immunity* has been postulated in [6,3]. According to Theorem 1 the combiner is maximum order correlation immune if for every linear function of the form (16) with nonvanishing correlation coefficient, and for every m, $1 \leq m \leq n$, there is at least one index k with $w_{mk} \neq 0$. Note that this coincides with condition MCI as introduced in [3] for memoryless combiners.

Theorem 1 extends Rueppels's treatment of maximum order correlation immunity in [5], as it covers every kind of correlation to LFSR–sequences. Such correlations exist even if the combiner is chosen to be maximum order correlation immune. In fact in the case $D_0 \neq 0$ the "total" correlation is independent of the combiners f_0 and f_1 as expression (17) converges to 1. This generalizes a corresponding result in [3] for memoryless combiners.

Motivated by these results one might be tempted to choose a maximum order correlation immune combiner (7,8) satisfying $D_0 = 0$. However in this case it can be shown that the sequence $z'_j = z_j + z_{j-1}$ is generated even by a memoryless combiner. Hence by a result in [3] z'_j is correlated to LFSR–sequences with correlation coefficients c_i with the property that

$$\sum_i {c_i}^2 = 1$$

Thus one can well achieve that all correlation coefficients in Theorem 1 vanish. However the sequence $\mathcal{Z}'$, which is easily obtained from $\mathcal{Z}$, has correlation to LFSR–sequences, as in the case where $D_0 \neq 0$.

4 Correlation Conditioned on Known Output Sequence

So far correlation was not conditioned on the output of the combiner. A completely different situation results if correlation is conditioned on the events $z_j = 0$ or $z_j = 1$. This is exemplified for the basic summation combiner with two inputs, where knowledge of portions of the output sequence can considerably reduce the uncertainty about the carry bit. This effects correlation of z_j to the input sum $a_j + b_j$, although z_j and $a_j + b_j$ are uncorrelated in the average.

It can be shown that in a run of s consecutive output digits 0 the carries tend to be 1. For example assume that $z_{j+1} = z_{j+2} = \cdots = z_{j+s} = 0$. Then at the end of the run the carry bit σ_{j+s} is 1 with probability at least $1-2^{-s}$. More generally for every t with $1 \leq t \leq s$, the conditional probability satisfies $P(\sigma_{j+s} = \sigma_{j+s-1} = \cdots = \sigma_{j+t} = 1) \geq 1 - 2^{-t}$. Similarly, in a run of s consecutive output digits 1 the carries tend to be 0. As a consequence the following result can be derived (cf. [4]).

Theorem 2 *(1) Suppose that the output of the basic summation combiner satisfies $z_{j+1} = z_{j+2} = \cdots = z_{j+s} = 0$, and $z_{j+s+1} = 1$. Then for every t with $1 \leq t \leq s$ the following $s-t+2$ equations*

$$\begin{aligned} z_{j+t+1} &= a_{j+t+1} + b_{j+t+1} + 1 = 0 \\ z_{j+t+2} &= a_{j+t+2} + b_{j+t+2} + 1 = 0 \\ &\vdots \\ z_{j+s+1} &= a_{j+s+1} + b_{j+s+1} + 1 = 1 \\ z_{j+s+2} &= a_{j+s+2} + b_{j+s+2} + a_{j+s+1} \end{aligned} \tag{18}$$

are simultaneously satisfied with probability at least $1-2^{-t}$.

(2) Suppose that the output of the basic summation combiner satisfies $z_{j+1} = z_{j+2} = \cdots = z_{j+s} = 1$, and $z_{j+s+1} = 0$. Then for every t with $1 \leq t \leq s$ the following $s-t+2$ equations

$$\begin{aligned} z_{j+t+1} &= a_{j+t+1} + b_{j+t+1} = 1 \\ z_{j+t+2} &= a_{j+t+2} + b_{j+t+2} = 1 \\ &\vdots \\ z_{j+s+1} &= a_{j+s+1} + b_{j+s+1} = 0 \\ z_{j+s+2} &= a_{j+s+2} + b_{j+s+2} + a_{j+s+1} \end{aligned} \tag{19}$$

are simultaneously satisfied with probability at least $1-2^{-t}$.

Observe that Theorem 2, stating that the equations (18, 19) are simultaneously satisfied with a certain probability, is much stronger than the statement that these equations are individually satisfied with the same probability. This fact can be cryptanalytically exploited as described in the next section.

5 Cryptanalysis of the Summation Cipher with Two LFSRs

In the basic summation cipher the two input sequences to the adder are produced by LFSRs. Then the systems of equations in Theorem 2 can be cryptanalytically exploited in a known plaintext attack.

Suppose that a run $z_{j+1}, \ldots, z_{j+s}$ of s consecutive 0's or 1's has been observed in the key stream sequence. By considering the digits $z_{j+t+1}, \ldots, z_{j+s+2}$ one obtains $s-t+2$ equations of the form (18) or (19) which are simultaneously satisfied with probability at least $1-2^{-t}$. The actual value of t, which is a parameter for the reliability of the equations, may be chosen depending on the length of known portion of the key stream. Since the digits a_j and b_j in (18, 19) are linearly expressed in terms of the initial state of the two LFSRs, one obtains a system of $s-t+2$ linear equations for the initial digits of the LFSRs. Our aim is to find sufficiently many such systems with highest reliability, which are suitably combined to a system of linear equations for the initial digits.

Let N be the length of the known key stream sequence and k be the key size (which is the sum of the LFSR-lengths). The key stream is scanned for runs of at least s consecutive 0's or 1's. Suppose that a total number of n such runs have been found. According to the desired reliability we choose the parameter t, and we obtain, as described in Theorem 2, a "block" of $d = s-t+2$ equations for each run. Thus we get at least nd equations for the initial digits. We assume that $nd > k$, i.e. $nd = \alpha k$ where $\alpha > 1$. To solve for the key we only need $m = \lceil k/d \rceil \approx \alpha^{-1} n$ "correct" blocks of equations. In order to find m correct blocks proceed as follows.

1. Randomly choose m out of the n available blocks, and solve the resulting system of linear equations for the k unknowns.

2. Test all possible solutions obtained in step 1 whether they produce the correct key stream. If there is a correct solution terminate, else go to step 1.

The complexity of this cryptanalytic algorithm is dominated by the total number of trials. To get an estimate of this number, observe that each block has probability ρ of being incorrect, where $\rho \leq 2^{-t}$. Then the expected number of trials needed is the reciprocal value of the probability q that, by sampling without replacement, m randomly chosen blocks are correct. We estimate this probability in a typical situation where ρn blocks are incorrect. (Assume here for simplicity that ρn is an integer.) Then q is estimated as

$$q = \left(1-\frac{\rho n}{n}\right)\left(1-\frac{\rho n}{n-1}\right)\cdots\left(1-\frac{\rho n}{n-(m-1)}\right) \quad (20)$$

$$> \left(1-\frac{\rho n}{n-m}\right)^m = \left(1-\frac{\alpha\rho}{\alpha-1}\right)^m \quad (21)$$

As an illustration we consider the following example.

Example. Consider a basic summation cipher with two LFSRs of length approximately 200, i.e. $k = 400$. Suppose that we have $N = 50,000$ digits of the key stream sequence. If this sequence is scanned for runs, then in the average

$$n \approx \frac{N}{2^s} \quad (22)$$

runs of length at least s are to be expected (see [1], p. 322 ff.). If we choose $s = 7$ we obtain $n = 390$ runs of length at least 7. Take $t = 4$. Then $d = s-t+2 = 5$ is the length of a block, and $\rho = 2^{-4} = 1/16$ is the probability of a block being incorrect.

Moreover $m = k/d = 80$ blocks of equations are needed to solve for the key. The value of α is obtained as $\alpha = n/m = 390/80 = 4.88$. Thus

$$q > \left(1 - \frac{4.88}{3.88} \cdot \frac{1}{16}\right)^{80} = 0.0014 \quad \text{and} \quad q^{-1} < 699$$

and therefore less than 700 trials are already sufficient in a typical situation.

This example shows that the summation cipher with two LFSRs can be successfully cryptanalyzed for LFSRs of considerable length with arbitrary feedback connection. Note that our algorithm also works if the known portion of the key stream has some (but not too many) gaps.

6 Comments on the Cryptanalytic Algorithm

The success of our algorithm rests on the weakness of the basic summation combiner as observed in Theorem 2. It is shown in [8] that a similar cryptanalysis is no longer possible for a summation cipher with more than 2 LFSRs. From this point of view it is recommended to take several LFSRs of moderate length rather than just few long LFSRs.

The method of our algorithm can be described in more general terms. Basically the cryptanalytic problem consists in finding a k-bit key. Observing that this key is determined by a number $m = k/d$ of (correct) blocks of equations, we search for $m = k/d$ such blocks instead of the k unknown bits. Since a block is correct with probability at least $1 - 2^{-t} \geq 0.5$, this procedure may be compared with an exhaustive search over only k/d bits instead of k bits. This is similar to the effect of a reduction of the key size by the factor d. However if a block is incorrect it cannot be corrected by complementing like a single bit. Therefore a set S of more than m blocks is required in order to find m correct blocks.

Let n denote the total number of available blocks and ρ the probability of a block being incorrect. Then $n - \rho n$ blocks are expected to be correct. Hence it is necessary that $n - \rho n$ is larger than m. Therefore it is favourable to have ρ small and $\alpha = n/m$ large. In fact already for $\alpha \approx 5$ and ρ only slightly smaller than 0.5, our cryptanalytic algorithm is much faster than an exhaustive search, even if the blocks consist of single bits, i.e. if $d = 1$ and $k = m$.

In the case $d = 1$ our method leads to a procedure to find k correct bits out of a set of n bits, where each bit in the set is assumed to be incorrect with probability ρ. This is exactly the situation one is faced with in the general correlation problem in cryptanalysis. In this direction our method applies to increase the efficiency of a cryptanalytic algorithm described in [2].

Algorithm A in [2] addresses the problem of determining the initial digits of a k-bit LFSR (with few feedback taps) from a disturbed output sequence of the LFSR. The algorithm describes a method to find k digits with highest probability of being undisturbed. These digits are taken as an estimate of the LFSR-sequence at the corresponding positions. Then the correct sequence is found by testing modifications of this estimate. Again denote by ρ the probability of a selected bit to be incorrect. It is shown in [2] that in the average

$$W_0 = 2^{h(\rho)k} \tag{23}$$

trials are necessary, where $h(\rho)$ denotes the binary entropy function.

We can improve algorithm A by applying the method as introduced in Section 5. According to this method we start with a set S of more than k digits having high probability of being undisturbed. Then we randomly choose k digits from this set and test these whether they are correct, i.e. whether they determine the correct LFSR–sequence. This process is repeated until k correct digits have been found.

We express the cardinality of the set S as a multiple of k, i.e. $|S| = \alpha k$ where $\alpha > 1$. According to (21) it is favourable to choose S (or α) as large as possible. On the other hand for increasing cardinality of S the reliability of the selected digits will decrease (cf. [2]). However for moderate α (e.g. $\alpha = 4$ or 5) the error probability ρ turns out to be roughly the same as for $\alpha = 1$. Therefore according to (21), in a typical situation the average number of trials is less than

$$W_1 = \left(1 - \frac{\alpha}{\alpha - 1}\rho\right)^{-k} = 2^{-\log_2(1-\frac{\alpha}{\alpha-1}\rho)\,k} \tag{24}$$

For a comparison of the two work factors W_0 and W_1 we may assume that the fraction $\alpha/(\alpha - 1)$ in (24) is close to 1. Thus (24) can be replaced by

$$W_1 = 2^{\ell(\rho)k} \tag{25}$$

where $\ell(\rho) = -\log_2(1-\rho)$. Formulas (23) and (25) show that both methods have exponential complexity. However the exponent in (25) is smaller than that in (23). In particular, for small ρ the value $\ell(\rho)$ is a small fraction of $h(\rho)$. In fact

$$\lim_{\rho \to 0} \frac{h(\rho)}{\ell(\rho)} = \infty \tag{26}$$

Thus for small ρ the method of Section 5 leads to a substantial improvement of algorithm A, as is also illustrated in the following example.

Example. Consider a LFSR of length k = 200 with few feedback taps. Then with the method of algorithm A it is feasible to find e.g. a set S of 1000 digits with error probability lower than 0.1, i.e. with $\alpha = 5$ and $\rho \leq 0.1$. Then, in order to find the LFSR–sequence with a search as in the original algorithm A, formula (23) shows that 2^{94} trials would be necessary in the average. However if the improved algorithm A is applied, the number of trials according to (20) can be estimated as 2^{34}.

In order to find sufficiently many digits with small ρ, it has to be assumed (as in [2]) that the number of feedback taps is small. In fact for LFSRs with more than 10 feedback taps the feasibility of the improved algorithm is roughly limited to the same conditions as the original algorithm A.

References

[1] W. Feller, *An Introduction to Probability Theory and its Applications*, Vol 1, John Wiley & Sons, Inc., 1968.

[2] W. Meier, O. Staffelbach, *Fast Correlation Attacks on Certain Stream Ciphers*, Journal of Cryptology, Vol 1, No. 3, pp. 159–176, 1989.

[3] W. Meier, O. Staffelbach, *Nonlinearity Criteria for Cryptographic Functions*, Proceedings of Eurocrypt'89, Springer-Verlag, to appear.

[4] W. Meier, O. Staffelbach, *Correlation Properties of Combiners with Memory in Stream Ciphers*, full paper to appear in the Journal of Cryptology.

[5] R.A. Rueppel, *Correlation Immunity and the Summation Generator*, Advances in Cryptology—Crypto'85, Proceedings, pp. 260–272, Springer-Verlag, 1986.

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

[7] T. Siegenthaler, *Correlation-Immunity of Nonlinear Combining Functions for Cryptographic Applications*, IEEE Trans. Inform. Theory, Vol IT-30, pp. 776–780, 1984.

[8] O. Staffelbach, W. Meier, *Cryptographic Significance of the Carry for Ciphers Based on Integer Addition*, Proceedings of Crypto'90, Springer-Verlag, to appear.