On the Existence of Secure Feedback Registers (Extended Abstract)

Andrew Klapper*

Department of Computer Science
763H Anderson Hall
University of Kentucky
Lexington KY 40506-0046.
Phone: (606) 257-6743; FAX: (606) 323-1971
E-mail: klapper@cs.engr.uky.edu
WWW: http://al.cs.engr.uky.edu/~klapper/andy.html

Abstract. Designers of stream ciphers have generally used ad hoc methods to build systems that are secure against known attacks. There is often a sense that this is the best that can be done, that any system will eventually fall to a practical attack. In this paper we show that there are families of keystream generators that resist all possible attacks of a very general type in which a small number of known bits of a keystream are used to synthesize a generator of the keystream (called a synthesizing algorithm). Such attacks are exemplified by the Berlekamp-Massey attack. We first formalize the notions of a family of feedback registers and of a synthesizing algorithm. We then show that for any function $h(n)$ that is in $\mathcal{O}(2^{n/d})$ for every $d > 0$, there is a secure family $\mathcal{B}$ of periodic sequences in the sense that any efficient synthesizing algorithm outputs a register of size $h(\log(\text{period}(B)))$ given the required number of bits of a sequence $B \in \mathcal{B}$ of large enough period. This result is tight in the sense it fails for any faster growing function $h(n)$. We also consider several variations on this scenario.

Index Terms – Binary sequences, nonlinear feedback registers, security, cryptography, stream ciphers.

1 Introduction

Historically, the design of stream ciphers has been largely a matter of finding ad hoc methods of foiling existing cryptanalytic attacks. Having their roots in Shannon's information theory, designers often feel that seeking a truly secure stream cipher is hopeless, that the best they can do is design a system that resists known attacks. The purpose of this paper is to explore the possibility that there exist families of stream ciphers that resist cryptanalysis by very large classes of attacks. We use asymptotic complexity rather than Shannon theory as

* This research funded in part by NSF grant #NCR-9400762.

U. Maurer (Ed.): Advances in Cryptology - EUROCRYPT '96, LNCS 1070, pp. 256-267, 1996.

the basis for notions of security. A family of stream ciphers is secure against all attacks of a certain general type if all such attacks require asymptotically large numbers of bits of the keystream. Our approach to the construction of these families of stream ciphers, while recursive, gives no practical construction for such a family of stream ciphers.

The sort of attack we are concerned with uses a small number of known bits of a keystream to synthesize a fast generator for the keystream. Perhaps the best known example of such an attack is the Berlekamp-Massey algorithm [13]. If the keystream can be generated by a linear feedback shift register (or LFSR) of length n, then $2n$ bits of the sequence suffice for the Berlekamp-Massey algorithm to determine the LFSR that generates the keystream. The smallest such n is called the linear span of the keystream and is a well studied measure of security. The ingredients that make this attack of concern are as follows.

1. A class of fast devices (LFSRs) that generate all possible eventually periodic sequences.
2. A polynomial time algorithm A and a polynomial $p(n)$ such that if a sequence can be generated by a device of size n, then $p(n)$ bits of the sequence suffice for A to determine the device.

A great deal of energy has gone into the design of (nonlinear) feedback registers that resist the Berlekamp-Massey attack (see [4, 7, 8, 15, 16], to name but a few). Similar attacks exist in the literature. For example, the Berlekamp-Massey algorithm works in any odd characteristic, and a binary sequence can be treated as a sequence over any field [9]. Also, recently Klapper and Goresky have designed a class of devices called feedback with carry shift registers or FCSRs [10, 11], and an algorithm for synthesizing them based on 2-adic rational approximation theory [12]. Both these approaches have been used to cryptanalyze certain sequences that had previously been shown to resist the Berlekamp-Massey attack (in the first case, geometric sequences [4], in the second case summation combiners [15]).

In this paper we ask whether there is a family of efficiently generated sequences that resist all such attacks. The answer is affirmative. However, the techniques used to show their existence, while recursive, give no practical method for finding such a family. Even if we could give a reasonable description of such a family, it might not be possible to give a computationally effective description of its generators. This should not be seen as too much of a difficulty, however, as this is the case even for m-sequences. Finding generators for m-sequences amounts to finding primitive polynomials over $GF(2)$, and this is apparently a computationally hard problem unless one knows a factorization of $2^n - 1$ (although it is not hard for currently practical sizes). Yet m-sequences are commonly used in practice as the bases for keystream generators.

Related questions have been studied previously by Yao [18] and by Blum and Micali [2]. Their models and results were different, however, in a number of regards. First, their sequence generators were arbitrary polynomial time computable generators (in the size of the seed). We use a much more restrictive model based on the use of fast feedback registers for generation of bit streams.

Second, the attacks they were concerned with required the availability of all previously generated bits to predict the next bit (by a so-called next bit test). The attacks considered here require that only a small number (polynomially many in the size of the resulting generator) of bits be available to generate all remaining bits. Third, the attacks consider by Yao and Blum and Micali were probabilistic while those considered here are deterministic. Finally, the existence results they gave were based on unproved complexity theoretic assumptions, such as the intractability of the discrete logarithm problem. Our results hold independent of any such assumptions.

Maurer also considered the design of private key systems that resist all attacks [14]. His point of view differed from ours in that the system he designed required a globally accessible source of public randomness. Also, the notion of security was probabilistic – the probability that an enemy could obtain information was shown to be exceedingly small.

In Section 2 we abstract the notions of fast keystream generator and of efficient algorithms for synthesizing such generators given a small number of initial bits. In Section 3 we first show that there is a family of keystream generators that admits no such synthesizing algorithm. We then show that an efficient, secure family $\mathcal{B}$ of sequences exists. This family is secure in the sense that, for every family of keystream generators $\mathcal{F}$ that admits a synthesizing algorithm, the size of the smallest generator in $\mathcal{F}$ that outputs a given sequence B in $\mathcal{B}$ grows at a superpolynomial rate in the size of smallest efficient generator for B. In Section 4 we show that the bounds in Section 3 are optimal. In Sections 5 and 6 we consider two variants: the case where the cryptanalyst is only required to generate a fraction of the keystream; and the case where the number of bits the cryptanalyst has access to is linear in the size of the smallest generator.

2 Definitions

In this section we describe feedback registers, the basic objects of study of this paper, and notions of security for families of feedback registers.

Definition 1. A *feedback register* of length n is determined by a function $F : \{0,1\}^n \to \{0,1\}^n$, called the *feedback function*. The *state* of the register is an n-bit vector $\bar{x} = (x_0, \cdots, x_{n-1})$. The *output* from the state $\bar{x}$ is x_0, and the *next state* is $F(\bar{x})$.

Thus from a given state $\bar{x}$ a feedback register outputs an infinite eventually periodic binary sequence by iterating the output and next state operations. In algorithms dealing with descriptions of feedback functions, we assume that the functions are described by circuits using binary AND (denoted $\wedge$), binary XOR (denoted $\oplus$), and NOT (denoted $\neg$) gates. Binary OR gates could have been used instead of XOR gates, but for the functions described here we find XOR more convenient. Changing the types of gates only changes complexity measures by a constant multiple. The AND and XOR gates are assumed to have fan-in two.

Such circuits can be encoded as binary strings [1]. The *size* of a register F is the minimum number of gates in a circuit that computes the function F. The *depth* of a register F is the depth of the minimum depth circuit that computes F. In a software implementation, the time it takes to evaluate a circuit is proportional to its size. In a hardware implementation, evaluation time is proportional to depth.

Note that feedback registers are really a type of finite state machine *without input*. One can consider more general finite state machines without input, where the output is computed as a function of the state rather than as the rightmost bit of the state vector. However, any such machine can be easily converted into an equivalent machine (i.e., producing the same output sequence) of the type used here by increasing the length by one. This new bit is then updated by the composition of the original output and state change functions. The register has depth and size at most twice those of the original machine. Thus our results apply to the more general model as well. The simpler model is, however, easier to work with.

A *family of feedback registers*, $\mathcal{F}$, is an infinite collection of feedback registers such that every eventually periodic binary sequence can be output by at least one register in $\mathcal{F}$. We let $\mathcal{F}_n$ denote the set of feedback registers in $\mathcal{F}$ of length n. If B is an infinite eventually periodic binary sequence, then the $\mathcal{F}$-span of B, denoted $\lambda_{\mathcal{F}}(B)$, is the least integer n such that B can be output by a register in $\mathcal{F}_n$.

We are concerned with registers whose feedback functions can be computed quickly. Let $\delta(n)$ be the maximum over all F in $\mathcal{F}_n$ of the depth of F. We say $\mathcal{F}$ is *fast* if $\delta(n) \in \mathcal{O}(\log(n))$. Note that this implies that the sizes (numbers of gates) of the registers in $\mathcal{F}$ are polynomial in the lengths of the registers. In fact, in all but one case in this paper, the sizes of fast families of registers described are linear in the lengths of the registers.

Our basic concern is whether, given a small number of bits of a sequence B, we can efficiently synthesize the smallest register in $\mathcal{F}$ that outputs B.

Definition 2. Algorithm T is an *$\mathcal{F}$-synthesizing algorithm* if, when given the input $b_0, \cdots, b_{k-1}$,

1. T outputs the encoding of a circuit that computes a feedback function $F \in \mathcal{F}$; and
2. if n is the length of F, T also outputs an n-bit vector $\bar{a}$ such that the first k bits of the output of F with start state $\bar{a}$ are $b_0, \cdots, b_{k-1}$.

The algorithm may or may not be assumed to know the period of the sequence B. T is *effective* if:

1. It runs in polynomial time; and
2. There is a polynomial $p(n)$ such that if $n = \lambda_{\mathcal{F}}(B)$, on input $b_0, \cdots, b_{k-1}$ with $k \geq p(n)$, T outputs an $F \in \mathcal{F}$ of length n that generates B.

A family $\mathcal{F}$ of registers is *synthetic* if there is an effective $\mathcal{F}$-synthesizing algorithm.

Certain families of sequences that have played a significant role in cryptography are known to be synthetic.

Fact 3 *The family of LFSRs and the family of FCSRs are synthetic.*

We say that a family of sequences is secure with respect to a family of registers if there is either no way to synthesize the best register in the family for a given sequence, or the length of the best register grows quickly with period of the sequence.

Definition 4. Let $\mathcal{B} = B^1, B^2, \cdots$ be a sequence of binary sequences of increasing periods, and let $p_{\mathcal{B}}(n)$ be the period of B^n. Let $\Lambda_{\mathcal{F},\mathcal{B}}(n) = \lambda_{\mathcal{F}}(B^n)$. Then $\mathcal{B}$ is *$\mathcal{F}$-secure* if either

1. $\mathcal{F}$ is not synthetic; or
2. For every $k > 0$, we have

$$\Lambda_{\mathcal{F},\mathcal{B}}(n) \in \Omega(\log(p_{\mathcal{B}}(n))^k).$$

In either case, for large enough n a short register in $\mathcal{F}$ generating B^n cannot be found effectively. Our goal is to show the existence of a sequence of periodic binary sequences $\mathcal{B}$ such that

1. $\mathcal{B}$ is $\mathcal{F}$-secure for every family of feedback registers;
2. There is a family $\mathcal{F}$ of fast registers containing generators for the sequences in $\mathcal{B}$ whose lengths are logarithmic in the periods of the sequences.

In describing the growth rates of functions, we use the following terminology.

Definition 5. Let $f(n)$ be a function.

1. We say $f(n)$ is *subexponential* if for every $d > 0$, $f(n) \in \mathcal{O}(2^{n/d})$.
2. We say $f(n)$ is *superpolynomial* if for every $k > 0$, $f(n) \in \Omega(n^k)$.

Thus by our definition, a family is secure if it achieves superpolynomial security. In fact, we show that we can find families that achieve arbitrary subexponential security, not simply superpolynomial security. It is well known that there are subexponential superpolynomial functions.

3 Existence of Secure Feedback Registers

In this section we give theorems on the existence of feedback registers and sequences that resist all synthesis attacks in the sense that any such attack outputs inefficient generators.

Theorem 6. *Let $h(n)$ be any subexponential function. There exists a fast family $\mathcal{F}$ of registers such that for every synthetic family $\mathcal{F}'$, there are infinitely many registers F in $\mathcal{F}$ with output sequence S satisfying*

$$\lambda_{\mathcal{F}'}(S) \geq h(\lambda_{\mathcal{F}}(S)).$$

Proof: For each synthesis algorithm T, let $\mathcal{F}_T$ be the family of registers that is output by T. Let $\mathcal{F}^1, \mathcal{F}^2, \cdots$ be an enumeration of the synthetic families of registers such that each $\mathcal{F}_T$ occurs infinitely often. Let the corresponding synthesis algorithms be $T^1, T^2, \cdots$.

We construct $\mathcal{F}$ in stages by a kind of diagonalization argument. At the end of stage $i-1$ we will have constructed $\mathcal{F}$ for registers of lengths up to k_{i-1} so that the theorem holds for $\mathcal{F}^1, \cdots, \mathcal{F}^{i-1}$. We now show how to extend the construction to stage i.

Let $p(n)$ be a polynomial such that for every sequence B, T^i outputs a register that generates B given at least $p(\lambda_{\mathcal{F}^i}(B))$ bits of B. We may assume $p(n) = n^d$. Let r be larger than the period of any sequence generated by any register already in $\mathcal{F}$, and be large enough that for every $k \geq r$, we have

$$2^{k/d} > h(k+3).$$

Let n satisfy $2^r < p(n)$. Let k satisfy $2^k \leq p(n) < 2^{k+1}$. We construct a pair of registers F and G of length at most $k+3$ and depth at most $\lceil \log(k) \rceil + 1$, whose outputs agree on the first $2^{k+1}-1$ terms, but not on the 2^{k+1}st term.

For F we choose a linear feedback shift register of length $k+1$ that outputs an m-sequence (of period $2^{k+1}-1$). The feedback function of such a register is just a shift for k bits, and an XOR of at most $k+1$ bits. Thus it can be computed in depth $\lceil \log(k) \rceil$.

For G we take a length $k+3$ register with bits labeled x_{k+2} to x_0 from left to right. The leftmost $k+1$ bits are updated exactly as the $k+1$ bits of F. Bit x_1 changes only when the leftmost $k+1$ bits all equal 1. Bit x_0 (the output bit) always equals $x_1 \oplus x_2$. Suppose x_1 is initially 0, the leftmost $k+1$ bits of G are initially identical to the $k+1$ bits of F, and $x_0 = x_1 \oplus x_2$ initially. Then the output from G is strictly periodic with period $2(2^{k+1}-1)$. The first half of one period equals a period of the output of F, while the second half equals the complement of a period of the output of F. The additional circuitry required for G has depth $\lceil \log(k) \rceil + 1$ since $x_1 = x_1 \oplus (x_2 \wedge \cdots \wedge x_{k+3})$, and $x_0 = (x_1 \oplus (x_2 \wedge \cdots \wedge x_{k+3})) \oplus x_3$ (remember: the new value of x_2 is the old value of x_3).

For initial values, we take $0, 1, 1, \cdots, 1$ for F, and $0, 1, 1, \cdots, 1, 0, 1$ for G. This gives the desired behavior. Let B^1 and B^2 be the resulting output sequences. Since $p(n) < 2^{k+1}$, T^i cannot distinguish these sequences with only $p(n)$ bits available. Thus at least one of the sequences, say B^m, has

$$\lambda_{\mathcal{F}^i}(B^m) > n \geq 2^{k/d} > h(k+3).$$

We put the corresponding register in $\mathcal{F}$ and let k_i be the length of that register. Thus

$$\lambda_{\mathcal{F}^i}(B^m) > h(k+3) \geq h(k_i) = h(\lambda_{\mathcal{F}}(B^m))$$

as desired. This concludes stage i.

To make $\mathcal{F}$ a family of registers (i.e., capable of generating every sequence), we add to $\mathcal{F}$ every linear feedback shift register with only a single tap in its feedback function. □

In particular, for the sequence $\mathcal{B}$ of registers constructed in the proof, there is no i such that $\Lambda_{\mathcal{F}^i,\mathcal{B}}(m)$ is in $\mathcal{O}(h(\Lambda_{\mathcal{F},\mathcal{B}}(m)))$, and $h(n)$ can be taken to be a superpolynomial function.

Corollary 7. *There exist (uncountably many) fast nonsynthetic families of registers.*

However, we want a stronger statement than that given by Theorem 6. All we know is that the above bound holds for infinitely many sequences in $\mathcal{B}$. We want it to hold for all sequences in $\mathcal{B}$ of sufficiently large period. That is, we want $\Lambda_{\mathcal{F}^i,\mathcal{B}}(m) \in \Omega(h(\Lambda_{\mathcal{F},\mathcal{B}}(m)))$. We can modify the above construction to achieve this, as follows.

Theorem 8. *Let $h(n)$ be any subexponential function. There exists a sequence of binary periodic sequences $\mathcal{B} = B^1, B^2, \cdots$ such that*

a. $\mathcal{B}$ can be generated by a fast family $\mathcal{F}$ of registers such that the length of the register generating B^m is at most twice the log of the period of B^m;
b. For every synthetic family $\mathcal{F}'$, if m is sufficiently large, then

$$\Lambda_{\mathcal{F}',\mathcal{B}}(m) \geq h(\log(period(B^m))).$$

In particular, if we let h be superpolynomial, then $\mathcal{B}$ satisfies the requirements at the end of Section 2.

Proof: We begin with an enumeration $\mathcal{F}^i$ of the synthetic families of registers and construct $\mathcal{B}$ by stages, as in the preceding proof. At the ith stage we construct B^i to have appropriate properties with respect to $\mathcal{F}^1, \cdots, \mathcal{F}^i$. Let $p(n) = n^d$ be so that for $1 \leq j \leq i$, T^j synthesizes a register in $\mathcal{F}^j$ that outputs any sequence S given $p(\lambda_{\mathcal{F}^j}(S))$ bits of S.

Let $t = \lceil \log(i) \rceil$. We construct B^i so that it has period $2^{k+1}+t-1$, and can be generated by a register of length $k+t+2$ and depth $\max(\lceil \log(k+1) \rceil, \lceil \log(t) \rceil)+3$ for some k.

Let r be

1. larger than the period of any previous B^j;
2. larger than t; and
3. large enough that, for every $k \geq r$, we have

$$2^{k/d} > h(\log(2^k + t)).$$

Choose n so that $n^d > 2^r$. Let k satisfy $2^k \leq n^d < 2^{k+1}$. We construct $i+1$ registers whose outputs are identical to one period of an m-sequence for $2^{k+1}-1$ bits. The jth register then outputs the binary expansion of the integer j. It

is straightforward to check that this can be done within the stated bounds on length and depth.

There must be at least one of these sequences, which we denote B^i, that satisfies

$$\lambda_{\mathcal{F}^j}(B^i) > n \geq 2^{k/d} > h(\log(2^k + t)) = h(\log(\text{period}(B^i)))$$

for $1 \leq j \leq i$. This concludes stage i. □

The constructions in Theorems 6 and 8 can be made recursive. That is, there is an effective procedure which, given i, outputs a list of the registers in $\mathcal{F}_i$ in the first case or B^i (or a generator of B^i) in the second case. Such a procedure, however, is likely to be impractically slow.

4 Exponential Bounds Are Impossible

In this section we show that Theorem 8 is sharp in the sense that the function h cannot be replaced by an exponential function.

Theorem 9. *Let $h(n) = 2^{n/d}$ be an exponential function, and let $\mathcal{B} = B^1, B^2, \cdots$ be any sequence of periodic binary sequences. There exists a fast synthetic family of sequences $\mathcal{F}$ such that for every i,*

$$\lambda_{\mathcal{F}}(B^i) \leq h(\log(\textit{period}(B^i))).$$

The synthesis algorithm for $\mathcal{F}$ is assumed to know the period of the sequence.

Proof: We construct the family $\mathcal{F}$ by describing a register synthesis algorithm T. $\mathcal{F}$ is then the set of registers output by T.

A k bit register generated by T has the following form. The leftmost $k-1$ bits operate independently of the rightmost bit. The rightmost bit is computed as a function of the leftmost $k-1$ bits. Thus the registers are, in effect, nonlinear feedback registers with nonlinear feedforward functions.

Algorithm T will produce a register of length $\lfloor p^{1/d} \rfloor$ when acting on a sequence of period p. Thus

$$\lambda_{\mathcal{F}}(B^i) = \left\lfloor \text{period}(B^i)^{1/d} \right\rfloor \leq h(\log(\text{period}(B^i))).$$

Since the number of bits the algorithm can have access to is polynomial in $\lambda_{\mathcal{F}}(B^i)$, we can assume T knows a complete period of period(B^i).

The first step is to construct a fast feedback register whose state sequence has period p. This can be done, for example, by constructing a maximal period LFSR of length k, with $2^{k-1} \leq p < 2^k$. Thus the period of this LFSR is $2^k - 1$. Such a LFSR can be found by an exhaustive search for a primitive polynomial of degree k. There are $2^k \leq 2p$ polynomials of degree k, and each can be checked for primitivity in time quasi-linear in p. Thus such a LFSR can be found in polynomial time. It can then be modified to switch back to its initial state after

p states by using a k-bit AND to check for the pth state. We call the resulting register G.

The construction is completed by finding a binary function on k bits that has the bits of the sequence as values on the p states of G. This can be written as an XOR of p terms, each an AND of k bits. Such an expression can be implemented as a circuit of depth $\lceil \log(p) \rceil + \lceil \log(k) \rceil$.

Finally, the resulting register is extended to length $\lfloor p^{1/d} \rfloor$ by padding it with $\lfloor p^{1/d} \rfloor - k$ dummy bits on the left. □

Theorem 9 is in fact true without knowledge of the period, but the polynomial bound on the number of bits available must be squared. This ensures that, for period p, p^2 bits are available. These are enough bits to determine the period unambiguously.

5 Partial Attacks

For many purposes the attacks considered in the preceding sections are too weak. A system is also vulnerable if an adversary can find a substantial number of bits of the keystream. This is especially true if there is enough context in the message to recover the remaining bits. If $\mathcal{F}$ is a family of registers, B is a sequence of (eventual) period m, and $0 < r \leq m$, then $\lambda_{\mathcal{F},r}(B)$ is the size of the smallest register F in $\mathcal{F}$ whose output agrees with B on at least r bits of each period[2] of B.

Definition 10. Let T be an $\mathcal{F}$-synthesizing algorithm and $0 < r(m) \leq m$. We say that T is $r(m)$-*effective for* $\mathcal{F}$ if

1. It runs in polynomial time; and
2. There is a polynomial $p(n)$ such that if B is a sequence with (eventual) period m and $n = \lambda_{\mathcal{F},r(m)}(B)$, then on input $b_0, \cdots, b_{k-1}$ with $k \geq p(n)$, T outputs an $F \in \mathcal{F}$ of length n. If the sequence generated by F is B', then for any k,
$$|\{i, k \leq i \leq k + m - 1 : b_i = b'_i\}| \geq r(m).$$

$\mathcal{F}$ is $r(m)$-*synthetic* if there is an $r(m)$-effective algorithm for $\mathcal{F}$.

Theorem 11. *Let $h(n)$ be subexponential and let $r(m) \in (m + \mathcal{O}(m^{1/2}))/2$. There exists a fast family $\mathcal{F}$ of registers such that for every $r(m)$-synthetic family $\mathcal{F}'$, there are infinitely many registers F in $\mathcal{F}$ with output sequence B of eventual period m satisfying*
$$\lambda_{\mathcal{F}',r(m)}(B) \geq h(\lambda_{\mathcal{F}}(B)).$$

[2] Some care must be taken here. The sequence B and the output sequence B' of F may have different periods, and in fact may not be strictly periodic, only eventually periodic. A reasonable interpretation is that $r/\text{period}(B)$ is less than or equal to the limit as n goes to ∞ of $|\{i, 1 \leq i \leq n : b_i = b'_i\}|/n$.

Proof Sketch: It is possible to construct a fast register such that one output period consists of an m-sequence followed by a degree one Reed-Muller code word. It is well known that the covering radius of the Reed-Muller code $RM(n,k)$ for fixed k is at most $(2^n - c2^{n/2})/2$, where c depends only on k [3]. The constant c can be made arbitrarily large by the choice of k. Taking complements, we see that there is a Reed-Muller codeword whose distance from any given sequence of length 2^n is at least $(2^n + c2^{n/2})/2$. $R(n,k)$ codewords can be generated by registers of length n and depth $\mathcal{O}(\log(n))$.

Let T be an $r(m)$-effective synthesis algorithm that is successful when given $p(\lambda_{\mathcal{F}',r(m)}(S))$ bits of any sequence S. We choose the sizes of the two parts of the above register so that the length $2^k - 1$ of the m-sequence is at least $p(h(\log(m)))$, where m is the period. We then choose the Reed-Muller codeword so that whatever sequence T outputs given $2^k - 1$ bits, the last 2^n bits disagree with the codeword on at least $(2^n + c2^{n/2})/2$ bits. It is possible to choose n and k so that this is more than $(m + \mathcal{O}(m^{1/2}))/2$. This is used to construct the family $\mathcal{F}$ in a manner similar to that in Theorem 6. □

This result will be improved if easily generated codes with small covering radii can be constructed. Coding theorists have studied covering radii for some years, but good asymptotic bounds are difficult to obtain and they seem to have not considered the question of fast generation of the codewords. It would also be desirable to find a sequence of sequences B^i so that B^i resists the first i $r(m)$-synthetic attacks. Using our techniques, such a construction would depend on finding easily generated codes with small *multicovering radii*, i.e., the smallest d such that every sequence is within distance d of at least i codewords. This concept appears to have not been studied by coding theorists.

6 Linear Synthesis Attacks

In this section we discuss the effect on our results of restricting the power of the synthesis algorithms.

As defined, synthesis algorithms depend on polynomial bounds. A synthesis algorithm for a family $\mathcal{F}$ of registers must work correctly if the number of bits available is at least a fixed polynomial in the $\mathcal{F}$-span, and the running time must be polynomially bounded in the number of bits available. If the degree of the polynomial is large, however, it is questionable whether such an attack should be considered strong enough to be of practical concern. By contrast, the Berlekamp-Massey and the 2-adic rational approximation algorithms work correctly if at least a linear number of bits are available. The former algorithm has quadratic running time, while the latter has quasi-quadratic running time. An algorithm is said to be a *linear synthesis algorithm* for a family $\mathcal{F}$ if it requires only a linear number of bits in $\lambda_{\mathcal{F}}(B)$ to synthesize a register in $\mathcal{F}$ that outputs B. Then $\mathcal{F}$ is said to be *linearly synthetic*. Theorems 6 and 8 can be improved if we restrict our attention to linear synthesis.

Theorem 12. *Let $h(n) \in o(2^n)$. There exists a sequence of binary periodic sequences $\mathcal{B} = B^1, B^2, \cdots$ such that*

a. *$\mathcal{B}$ can be generated by a fast family $\mathcal{F}$ of registers such that the length of the register generating B^i is at most twice the log of the period of B^i;*
b. *Let $\mathcal{F}'$ be a linearly synthetic family. For every sufficiently large i we have*

$$\lambda_{\mathcal{F}'}(B^i) \geq h(\log(period(B^i))).$$

This result can then be shown to be tight.

Theorem 13. *Let $h(n) = c2^n$, for some constant c. Let $\mathcal{B} = B^1, B^2, \cdots$ be a sequence of periodic binary sequences. There exists a fast linearly synthetic family of sequences $\mathcal{F}$ such that for every i,*

$$\lambda_{\mathcal{F}}(B^i) \leq h(\log(period(B^i))).$$

The synthesis algorithm for $\mathcal{F}$ is assumed to know the period of the sequence.

7 Conclusions and Open Questions

We have described a general model for attacks on stream ciphers of a very general type. Using this model, we have proved the existence of families of sequence generators that resist all such attacks. The proof, however, does not give a practical construction. We hope to inspire researchers to search for such highly secure sequence generators with more natural descriptions. The basic open question we leave is whether practical constructions can be found for register and sequence families that satisfy the conclusions of Theorem 8 and 6.

We have also considered only one class of attack on stream ciphers. Other attacks are possible, for example probabilistic attacks such as correlation attacks [17], differential cryptanalysis [5], and linear cryptanalysis [6]. It is desirable to formalize such probabilistic attacks and (hopefully) prove the existence of classes of keystream generators that universally resist them.

We have concentrated on the depth of circuits as a measure of feasibility. This corresponds to time of evaluation. Size (number of gates) is also a concern as it impacts area and hence cost. In all our theorems except Theorem 9 the sizes of the resulting fast registers are linear in the lengths of the registers. In Theorem 9, the sizes of the fast registers are polynomial in their lengths. We leave open the question as to whether one can achieve smaller sizes than these.

References

1. J. Balcázar, J. Díaz, and J. Gabarró: Structural Complexity I. Berlin: Springer 1988.
2. M. Blum and S. Micali: How to generate cryptographically strong sequences of pseudorandom bits. SIAM Journal on Computing 13, 850-864 (1984).
3. G. Cohen and S. Litsyn: On the covering radius of Reed-Muller codes. Discrete Mathematics 106-107, 147-155 (1992).
4. A.H. Chan and R.A. Games: On the linear span of binary sequences from finite geometries, q odd. IEEE Transactions on Information Theory 36, 548-552 (1990).

5. C. Ding: The differential cryptanalysis and design of natural stream ciphers. In: R. Anderson (ed.): Fast Software Encryption: Proceedings of 1993 Cambridge Security Workshop. Lecture Notes in Computer Science 809. Berlin: Springer 1994, pp. 101-120.
6. J. Golić: Linear cryptanalyis of stream ciphers. In: B. Preneel (ed.): Fast Software Encryption: Proceedings of 1994 Leuven Security Workshop. Lecture Notes in Computer Science 1008. Berlin: Springer 1995, pp. 154-169.
7. E.J. Groth: Generation of binary sequences with controllable complexity. IEEE Transactions on Information Theory IT-17, 288-296 (1971).
8. E.L. Key: An Analysis of the structure and complexity of nonlinear binary sequence generators. IEEE Transactions on Information Theory IT-22, 732-736 (1976).
9. A. Klapper: The Vulnerability of Geometric Sequences Based on Fields of Odd Characteristic. Journal of Cryptology 7, 33-51 (1994).
10. A. Klapper and M. Goresky: 2-Adic Shift Registers. In : R. Anderson (ed.): Fast Software Encryption: Proceedings of 1993 Cambridge Security Workshop. Lecture Notes in Computer Science 809. Berlin: Springer 1994, pp. 174-178.
11. A. Klapper and M. Goresky: Feedback Shift Registers, Combiners with Memory, and 2-Adic Span, University of Kentucky, Deptartment of Computer Science Technical Report, 1995.
12. A. Klapper and M. Goresky: Cryptanalysis Based on 2-Adic Rational Approximation. In: D. Coppersmith (ed.) Advances in Cryptology – CRYPTO '95. Lecture Notes in Computer Science 963. Berlin: Springer 1994, pp. 262-273.
13. J.L. Massey: Shift register sequences and BCH decoding. IEEE Transactions on Information Theory IT-15, 122-127 (1969).
14. U. M. Maurer: A Provably-Secure Strongly-Randomized Cipher. In: S. Vanstone (ed.) Advances in Cryptology – CRYPTO '90. Lecture Notes in Computer Science 473. Berlin: Springer 1991, pp. 361-73.
15. R. Rueppel: Analysis and Design of Stream Ciphers. New York Springer, 1986.
16. R.A. Rueppel and O.J. Staffelbach: Products of linear recurring sequences sith maximum complexity, IEEE Transactions on Information Theory IT-33, 124-129 (1987).
17. T. Siegenthaler: Decrypting a class of stream ciphers using cipertext only. IEEE Transactions on Computing 34, 81-85 (1985).
18. A. Yao: Theory and applications of trapdoor functions. In: Proceedings, 23rd IEEE Symposium on Foundations of Computer Science, 1982, pp. 80-91.