Math. J. Okayama Univ. 30 (1988), 121－149

SOME NEW CRYPTOSYSTEMS BASED ON FEEDBACK SHIFT REGISTER SEQUENCES

HARALD NIEDERREITER*

1. Introduction.

A cryptosystem is a system designed for the communication of secret information and for data security. Cryptography——the art of designing cryptosystems——entered a new age with the seminal paper of Diffie and Hellman [7]. On the other hand, cryptanalysis——the art of breaking cryptosystems——is rapidly catching up with new advances in cryptography. As cryptanalysis progresses, the design of secure cryptosystems will require more and more sophisticated techniques of discrete mathematics.

The algebraic structure known as a finite field plays an increasingly important role in the design of cryptosystems. Typical examples of cryptosystems using finite field arithmetic include the RSA-type cryptosystems based on polynomial functions and rational functions over finite fields (see Lidl and Müller [15], [16], Müller and W. Nöbauer [19], and R. Nöbauer [22]), the knapsack-type public-key cryptosystems of Chor and Rivest [4] and the author [21], and the various cryptosystems based on discrete exponentiation in finite fields (see Diffie and Hellman [7], ElGamal [9], Lidl and Niederreiter [18, Ch. 9], Odlyzko [23], and Wah and Wang [30]). The construction of stream ciphers via multiplexed sequences also makes heavy use of the structure of finite fields (see Beker and Piper [1], Beth *et al.* [2], and Lidl and Niederreiter [18, Ch. 9]).

The security of cryptosystems based on discrete exponentiation in finite fields has recently been diminished by significant progress achieved on the inverse problem, namely that of calculating discrete logarithms in finite fields (see Blake *et al.* [3], Coppersmith [5], Coppersmith *et al.* [6], and Odlyzko [23]). Given the present hardware and software, such cryptosystems appear to be safe only if we use finite fields of very large order, say 2^n with $n \geq 800$. In this paper we propose some new cryptosystems for various purposes that are more complex than cryptosystems based on discrete exponentiation and therefore seem to be harder to break. All the cryptosystems described here use sequences that can be generated on

* The author gratefully acknowledges support for this research project by the Austrian Ministry for Science and Research.

a feedback shift register (abbreviated FSR) with finite field arithmetic. Therefore, these cryptosystems allow a fast and direct hardware implementation. Encryption and decryption are in general somewhat slower than for comparable cryptosystems based on discrete exponentiation, but there is a possible gain in security. A brief announcement of one of our cryptosystems is already contained in [20].

In Section 2 we give the necessary background on FSR sequences and in Section 3 we describe the cryptosystems in detail. Section 4 provides information on the algorithms that are needed for encryption and decryption in the context of these cryptosystems. A heuristic discussion of the complexity of these cryptosystems and of possible attacks is contained in Section 5. The proofs of some theoretical results on FSR sequences on which our cryptosystems rely are presented in Section 6.

2. Background on FSR sequences.

Let F_q denote the finite field with q elements. A sequence $s_0, s_1, \ldots$ of elements of F_q is called an *n-stage FSR* (or *nth-order linear recurring*) *sequence* if it satisfies a linear recurrence relation

$$(1) \qquad s_{i+n} = a_{n-1}s_{i+n-1} + \cdots + a_1 s_{i+1} + a_0 s_i \text{ for } i = 0, 1, \ldots$$

with constant coefficients $a_0, a_1, \ldots, a_{n-1} \in F_q$. The sequence is completely determined by the relation (1) and by the initial values $s_0, s_1, \ldots, s_{n-1}$. An n-stage FSR sequence can be generated by an FSR with n delay elements, at most n constant multipliers corresponding to the nonzero coefficients among $a_0, a_1, \ldots, a_{n-1}$, and at most $n-1$ adders, with the arithmetic being that of F_q. In the binary case $q = 2$, where all $a_j = 0$ or 1, simple wire connections or disconnections suffice to simulate the effect of the constant multipliers.

We collect here some standard notions and results for FSR sequences. We refer to the collection of articles in Golomb [12], the lecture notes of Selmer [27], and the books of Lidl and Niederreiter [17], [18] for more information on FSR sequences. The monic polynomial

$$(2) \qquad f(x) = x^n - a_{n-1}x^{n-1} - \cdots - a_1 x - a_0 \in F_q[x],$$

whose coefficients we read off from (1), is called a *characteristic polynomial* of the FSR sequence (s_i). A given FSR sequence can have many different characteristic polynomials since it can satisfy many different linear recur-

rence relations. The most extreme case is that of the zero sequence (i.e., the sequence all of whose terms are 0) which satisfies any linear recurrence relation. For a nonzero FSR sequence there exists a characteristic polynomial over F_q of least positive degree, and this uniquely determined monic polynomial is called the *minimal polynomial* of the FSR sequence. The minimal polynomial of the zero sequence is by definition the constant polynomial 1.

The minimal polynomial $m(x)$ of an FSR sequence contains a lot of information about the sequence. First of all, the characteristic polynomials of the sequence are exactly all the monic polynomials of positive degree that are divisible by $m(x)$. Furthermore, the minimal polynomial describes the periodicity properties of the sequence. Any FSR sequence is ultimately periodic with a possible preperiod. Let

$$m(x) = x^{e_0} m_1(x)^{e_1} \cdots m_t(x)^{e_t} \tag{3}$$

be the canonical factorization of $m(x)$ in $F_q[x]$ with integers $e_0 \geq 0$, $e_j \geq 1$ for $1 \leq j \leq t$, and $m_1(x), \ldots, m_t(x)$ distinct monic irreducible polynomials over F_q with $m_j(0) \neq 0$ for $1 \leq j \leq t$. For a polynomial $f(x)$ over F_q with $f(0) \neq 0$ we define $\mathrm{ord}(f(x))$ to be the least positive integer e such that $f(x)$ divides $x^e - 1$, and for a nonzero $f(x)$ with $f(0) = 0$ we write $f(x) = x^{e_0} f_1(x)$ with $f_1(0) \neq 0$ and define $\mathrm{ord}(f(x)) = \mathrm{ord}(f_1(x))$. The following facts are standard.

Lemma 1. *Let (s_i) be an FSR sequence with minimal polynomial $m(x)$ having the canonical factorization* (3). *Then:*

(a) *The preperiod of (s_i) is e_0.*

(b) *The least period of (s_i) is* $\mathrm{ord}(m(x))$.

(c) $\mathrm{ord}(m(x)) = p^u \mathrm{lcm}(\mathrm{ord}(m_1(x)), \ldots, \mathrm{ord}(m_t(x)))$, *where u is the least integer with $p^u \geq \max(e_1, \ldots, e_t)$.*

(d) *If $f(x)$ is irreducible over F_q and of degree d, then* $\mathrm{ord}(f(x))$ *divides $q^d - 1$.*

If (s_i) is an FSR sequence satisfying the linear recurrence relation (1) with initial values $s_0 = s_1 = \cdots = s_{n-2} = 0$, $s_{n-1} = 1$ ($s_0 = 1$ if $n = 1$), then (s_i) is called an *impulse response sequence*. The significance of such a sequence stems from the fact that the characteristic polynomial $f(x)$ in (2) is also the minimal polynomial of an impulse response sequence.

Another special FSR sequence satisfying (1) is the *power-sum sequence*,

defined in the following way. Let

$$f(x) = \prod_{j=1}^{n} (x-\alpha_j)$$

be the factorization of the characteristic polynomial $f(x)$ in its splitting field over F_q. Then we set

$$(4) \qquad s_i = \sum_{j=1}^{n} \alpha_j^i \text{ for } i = 0, 1, \ldots,$$

where we observe the convention $0^0 = 1$. To show that this sequence (s_i) satisfies (1), we note that

$$\begin{aligned} &s_{i+n} - a_{n-1}s_{i+n-1} - \cdots - a_1 s_{i+1} - a_0 s_i \\ &= \sum_{j=1}^{n} \alpha_j^{i+n} - a_{n-1} \sum_{j=1}^{n} \alpha_j^{i+n-1} - \cdots - a_1 \sum_{j=1}^{n} \alpha_j^{i+1} - a_0 \sum_{j=1}^{n} \alpha_j^i \\ &= \sum_{j=1}^{n} \alpha_j^i (\alpha_j^n - a_{n-1}\alpha_j^{n-1} - \cdots - a_1 \alpha_j - a_0) = \sum_{j=1}^{n} \alpha_j^i f(\alpha_j) = 0 \end{aligned}$$

for $i = 0, 1, \ldots$, so that (1) holds indeed. The minimal polynomial of the power-sum sequence is obtained from Theorem 3 in Section 6.

For our purposes, an important operation on sequences is that of *decimation*. If σ is the sequence $s_0, s_1, \ldots$ of elements of F_q and $k \geq 1$ and $d \geq 0$ are integers, then the *decimated sequence* $\sigma_k^{(d)}$ has the terms $s_d, s_{k+d}, s_{2k+d}, \ldots$, that is, $\sigma_k^{(d)}$ is obtained by taking every kth term of σ, starting from s_d. The study of this operation was initiated by Golomb [11] and Zierler [32].

3. Description of the cryptosystems.

In all the cryptosystems to be described here the alphabet for both plaintext and ciphertext is a finite field F_p, where p is a prime number. The plaintext messages are strings $a_0 a_1 \cdots a_{n-1}$ of elements of F_p of length $n \geq 2$. One of the principles is to use the plaintext message $a_0 a_1 \cdots a_{n-1}$ to form the polynomial

$$f(x) = x^n - a_{n-1}x^{n-1} - \cdots - a_1 x - a_0 \in F_p[x],$$

which serves then as the characteristic polynomial of a linear recurrence relation. In this context we may call $f(x)$ the *message polynomial*. The cryptosystems come in two versions, depending on which specific FSR

sequence satisfying the linear recurrence relation (1) we use. In Version 1 we consider the impulse response sequence associated with (1) and in Version 2 the power-sum sequence associated with (1). Version 1 has the advantage that it works for any prime p, so in particular for the prime $p = 2$ that allows the easiest implementation, whereas Version 2 works only for primes $p > n$. On the other hand, the transmission rates in Version 2 are about twice as fast as for the corresponding system in Version 1 since the ciphertexts are only about half as long.

Another principle that we employ is the fact that a characteristic polynomial of an n-stage FSR sequence is determined by the first $2n$ terms of the sequence (see Lidl and Niederreiter [17, p. 439]). In fact, standard algorithms such as the Berlekamp-Massey algorithm or continued fraction algorithms even allow the calculation of the minimal polynomial of the sequence (see [17, Ch. 8]). Thus, the first $2n$ terms of the sequence determine the rest of the sequence. If the sequence is known to be a power-sum sequence and we have $p > n$, then the first $n+1$ terms of the sequence already suffice to determine the rest of the sequence, as we will show in Section 4.

A1. FSR One-Key Cryptosystem (Version 1). Let p be an arbitrary prime and $n \geq 2$. The key is a random integer k with

$$\gcd(k, p) = 1, \ \gcd(k, p^j - 1) = 1 \text{ for } 1 \leq j \leq n, \tag{5}$$

and

$$1 < k < R = p^u \operatorname{lcm}(p-1, p^2-1, \ldots, p^n-1), \tag{6}$$

where u is the least integer with $p^u \geq n$.

Encryption: Given the plaintext message $a_0 a_1 \cdots a_{n-1}$, form the message polynomial

$$f(x) = x^n - a_{n-1}x^{n-1} - \cdots - a_1 x - a_0 \in F_p[x].$$

Let (s_i) be the impulse response sequence with characteristic polynomial $f(x)$. Then the ciphertext is the string

$$s_k s_{2k} \cdots s_{(2n-1)k}$$

of $2n-1$ elements of F_p.

Decryption: Upon receipt of the ciphertext $s_k s_{2k} \cdots s_{(2n-1)k}$ we consider the decimated sequence $(t_i) = (s_{ik})$. We know t_i for $1 \leq i \leq 2n-1$ and

also $t_0 = s_0 = 0$, and so the first $2n$ terms of (t_i). Now (t_i) has a minimal polynomial of degree $\leq n$, so that this minimal polynomial can be calculated and the rest of the sequence (t_i) is determined. For $0 \leq j \leq n-1$ let $d_j \geq 1$ be a solution of the congruence

$$kd_j \equiv n+j \bmod R. \tag{7}$$

Then

$$s_{n+j} = s_{kd_j} = t_{d_j} \text{ for } 0 \leq j \leq n-1. \tag{8}$$

Thus we know $s_0 = s_1 = \cdots = s_{n-2} = 0$, $s_{n-1} = 1$, as well as $s_n, s_{n+1}, \ldots, s_{2n-1}$, and so the first $2n$ terms of the sequence (s_i). On the basis of this information we can calculate the minimal polynomial $f(x)$ of (s_i) and thus recover the original message.

Some further comments on the deciphering procedure are in order. The fact that (t_i) has a minimal polynomial of degree $\leq n$ follows from Theorem 1 in Section 6, where a characteristic polynomial of (t_i) of degree n is given. Next we note that the conditions on k in (5) and the definition of R in (6) imply that $\gcd(k, R) = 1$, so that the congruence (7) can be solved for d_j. From $p^u \geq n$ it follows that $R > 2n > n+j$, hence

$$kd_j \geq n+j \text{ for } 0 \leq j \leq n-1. \tag{9}$$

Since the impulse response sequence (s_i) has the minimal polynomial $f(x)$ of degree n, Lemma 1(a) implies that the preperiod of (s_i) is $\leq n$. Thus, if r is the least period of (s_i), then we have

$$s_{i+vr} = s_i \text{ for all integers } i \geq n \text{ and } v \geq 0. \tag{10}$$

Now Lemma 1(b) shows that $r = \operatorname{ord}(f(x))$, and from Lemma 1(c), (d) and the definition of R it follows then that r divides R. Consequently, (7) implies

$$kd_j \equiv n+j \bmod r \text{ for } 0 \leq j \leq n-1.$$

Together with (9) and (10) this yields

$$s_{kd_j} = s_{n+j} \text{ for } 0 \leq j \leq n-1,$$

which verifies the first identity in (8). The second identity in (8) follows from the definition of the elements t_i.

A2. FSR One-Key Cryptosystem (Version 2). Let p be a prime and $2 \leq n < p$. The key is a random integer k with

$$\gcd(k,\ p^j-1) = 1 \text{ for } 1 \leq j \leq n \tag{11}$$

and

$$1 < k < M = \operatorname{lcm}(p-1, p^2-1, \ldots, p^n-1). \tag{12}$$

Encryption: Given the plaintext message $a_0a_1\cdots a_{n-1}$, form the message polynomial

$$f(x) = x^n - a_{n-1}x^{n-1} - \cdots - a_1x - a_0 \in F_p[x].$$

Let (s_i) be the power-sum sequence with characteristic polynomial $f(x)$. Then the ciphertext is the string

$$s_k s_{2k} \cdots s_{nk}$$

of n elements of F_p.

Decryption: Upon receipt of the ciphertext $s_k s_{2k} \cdots s_{nk}$ we consider the decimated sequence $(t_i) = (s_{ik})$. We know t_i for $1 \leq i \leq n$ and also $t_0 = s_0 = n \in F_p$, and so the first $n+1$ terms of (t_i). Now (t_i) is a power-sum sequence with a characteristic polynomial of degree n, so that this characteristic polynomial can be calculated and the rest of the sequence (t_i) is determined. For $1 \leq j \leq n$ let $d_j \geq 1$ be a solution of the congruence

$$kd_j \equiv j \bmod M. \tag{13}$$

Then

$$s_j = s_{kd_j} = t_{d_j} \text{ for } 1 \leq j \leq n. \tag{14}$$

Thus we know $s_0 = n \in F_p$ as well as $s_1, s_2, \ldots, s_n$, and so the first $n+1$ terms of the sequence (s_i). On the basis of this information we can calculate the polynomial $f(x)$ and thus recover the original message.

We comment on some points in the deciphering procedure. The fact that the decimated sequence (t_i) has a characteristic polynomial of degree n follows from Theorem 1 in Section 6. But it can also be seen directly that if

$$f(x) = \prod_{j=1}^{n} (x-\alpha_j)$$

is the factorization of $f(x)$ in its splitting field over F_p, then

$$t_i = s_{ik} = \sum_{j=1}^{n} \alpha_j^{ik} = \sum_{j=1}^{n} (\alpha_j^k)^i \text{ for } i = 0, 1, \ldots,$$

so that (t_i) is the power-sum sequence with characteristic polynomial

$$f_k(x) = \prod_{j=1}^{n} (x - \alpha_j^k).$$

We note also that the conditions on k in (11) and the definition of M in (12) imply that $\gcd(k, M) = 1$, so that the congruence (13) can be solved for d_j. By Theorem 3 in Section 6, the minimal polynomial $m(x)$ of the power-sum sequence (s_i) has no multiple roots, so that $e_0 \leq 1$ and $e_j = 1$ for $1 \leq j \leq t$ in (3). By Lemma 1(a), the preperiod of (s_i) is ≤ 1. Thus, if r is the least period of (s_i), we have

(15) $$s_{i+vr} = s_i \text{ for all integers } i \geq 1 \text{ and } v \geq 0.$$

Furthermore, Lemma 1(b) yields $r = \operatorname{ord}(m(x))$, and Lemma 1(c), (d) and the definition of M imply that r divides M. Thus from (13) we infer

$$kd_j \equiv j \bmod r \text{ for } 1 \leq j \leq n.$$

Together with (15) we get

$$s_{kd_j} = s_j \text{ for } 1 \leq j \leq n,$$

which verifies the first identity in (14). The second identity in (14) follows from the definition of the elements t_i.

The following key-exchange systems, which are analogs of the well-known Diffie-Hellman scheme (see [7]), rely on Theorem 1 in Section 6 and the fact that the minimal polynomial of degree $\leq n$ of an FSR sequence is determined by the first $2n$ terms of the sequence, whereas for a power-sum sequence it suffices to know the first $n+1$ terms provided that $p > n$ (see Section 4). In both systems B1 and B2, the correspondents A and B who want to have a common key know the polynomial

$$g(x) = x^n - b_{n-1}x^{n-1} - \cdots - b_1 x - b_0 \in F_p[x]$$

and the positive integer m that serves as the length of the key.

B1. FSR Key-Exchange System (Version 1). Let p be an arbitrary prime, $n \geq 2$, and $m \leq 2n-1$. Then A and B choose random integers h

and k, respectively, with

$$1 < h,\ k < \mathrm{ord}(g(x)).$$

Each one considers the impulse response sequence (s_i) with characteristic polynomial $g(x)$. Now A calculates the string

$$s_h s_{2h} \cdots s_{(2n-1)h}$$

and sends it to B, while B calculates the string

$$s_k s_{2k} \cdots s_{(2n-1)k}$$

and sends it to A. Then A takes $t_i = s_{ik}$, $0 \leq i \leq 2n-1$, determines the minimal polynomial of $(t_i) = (s_{ik})$, and then calculates the string

$$t_h t_{2h} \cdots t_{mh}.$$

Similarly, B takes $u_i = s_{ih}$, $0 \leq i \leq 2n-1$, determines the minimal polynomial of $(u_i) = (s_{ih})$, and then calculates the string

$$u_k u_{2k} \cdots u_{mk}.$$

Since $t_{ih} = s_{ihk} = u_{ik}$ for $1 \leq i \leq m$, both A and B have the same key $t_h t_{2h} \cdots t_{mh}$ consisting of a string of m elements of F_p.

B2. FSR Key-Exchange System (Version 2). Let p be a prime, $2 \leq n < p$, and $m \leq n$. Then A and B choose random integers h and k, respectively, with

$$1 < h,\ k < \mathrm{ord}(g(x)).$$

Each one considers the power-sum sequence (s_i) with characteristic polynomial $g(x)$. Now A calculates the string

$$s_h s_{2h} \cdots s_{nh}$$

and sends it to B, while B calculates the string

$$s_k s_{2k} \cdots s_{nk}$$

and sends it to A. Then A takes $t_i = s_{ik}$, $0 \leq i \leq n$, determines a characteristic polynomial of degree n of $(t_i) = (s_{ik})$, and then calculates the string

$$t_h t_{2h} \cdots t_{mh}.$$

Similarly, B takes $u_i = s_{ih}$, $0 \leq i \leq n$, determines a characteristic polynomial of degree n of $(u_i) = (s_{ih})$, and then calculates the string

$$u_k u_{2k} \cdots u_{mk}.$$

Since $t_{ih} = s_{ihk} = u_{ik}$ for $1 \leq i \leq m$, both A and B have the same key $t_h t_{2h} \cdots t_{mh}$ consisting of a string of m elements of F_p.

Next we describe variants of Shamir's no-key algorithm (see [14, pp. 345−346]). Suppose correspondent A wants to send a message to correspondent B that consists of a string $a_0 a_1 \cdots a_{n-1}$ of n elements of F_p. As before, we identify the plaintext message with the message polynomial

$$f(x) = x^n - a_{n-1}x^{n-1} - \cdots - a_1 x - a_0 \in F_p[x].$$

C1. FSR No-Key Algorithm (Version 1). Let p be an arbitrary prime and $n \geq 2$. Then A chooses a random integer h with

$$\gcd(h, p) = 1, \ \gcd(h, p^j - 1) = 1 \text{ for } 1 \leq j \leq n,$$

and $1 < h < R$, where R is as in (6). Now A considers the impulse response sequence (s_i) with characteristic polynomial $f(x)$ and transmits the string

$$s_h s_{2h} \cdots s_{(2n-1)h}$$

to B. On the basis of this information, B calculates the minimal polynomial of the decimated sequence $(t_i) = (s_{ih})$. Then B chooses a random integer k with

$$\gcd(k, p) = 1, \ \gcd(k, p^j - 1) = 1 \text{ for } 1 \leq j \leq n,$$

and $1 < k < R$. Now B transmits the string

$$t_k t_{2k} \cdots t_{(2n-1)k}$$

to A. On the basis of this information, A calculates the minimal polynomial of the decimated sequence $(u_i) = (t_{ik})$. Furthermore, A determines a solution $m \geq 1$ of the congruence

$$hm \equiv 1 \bmod R$$

and transmits the string

$$u_m u_{2m} \cdots u_{(2n-1)m}$$

to B. This allows B to calculate the minimal polynomial of the decimated sequence $(v_i) = (u_{im})$. Then B determines for $0 \leq j \leq n-1$ a solution $d_j \geq 1$ of the congruence

$$kd_j \equiv n+j \bmod R.$$

Then

$$s_{n+j} = v_{d_j} \text{ for } 0 \leq j \leq n-1, \tag{16}$$

so that B can calculate $s_n, s_{n+1}, \ldots, s_{2n-1}$ and thus recovers the original message as in the cryptosystem A1.

The comments following the cryptosystem A1 apply, mutatis mutandis, to the case of the cryptosystem C1 as well. In particular, the identity (16) is shown by the same method as for (8), namely by noting that

$$hm\,kd_j \equiv n+j \bmod R,$$

hence

$$s_{n+j} = s_{hm\,kd_j} = t_{m\,kd_j} = u_{m\,d_j} = v_{d_j}.$$

C2. FSR No-Key Algorithm (Version 2). Let p be a prime and $2 \leq n < p$. Then A chooses a random integer h with

$$\gcd(h, p^j-1) = 1 \text{ for } 1 \leq j \leq n$$

and $1 < h < M$, where M is as in (12). Now A considers the power-sum sequence (s_i) with characteristic polynomial $f(x)$ and transmits the string

$$s_h s_{2h} \cdots s_{nh}$$

to B. On the basis of this information, B calculates a characteristic polynomial of degree n of the decimated sequence $(t_i) = (s_{ih})$. Then B chooses a random integer k with

$$\gcd(k, p^j-1) = 1 \text{ for } 1 \leq j \leq n$$

and $1 < k < M$. Now B transmits the string

$$t_k t_{2k} \cdots t_{nk}$$

to A. On the basis of this information, A calculates a characteristic polynomial of degree n of the decimated sequence $(u_i) = (t_{ik})$. Furthermore, A determines a solution $m \geq 1$ of the congruence

$$hm \equiv 1 \bmod M$$

and transmits the string

$$u_m u_{2m} \cdots u_{nm}$$

to B. This allows B to calculate a characteristic polynomial of degree n of the decimated sequence $(v_i) = (u_{im})$. Then B determines for $1 \leq j \leq n$ a solution $d_j \geq 1$ of the congruence

$$kd_j \equiv j \bmod M.$$

Then

$$s_j = v_{d_j} \text{ for } 1 \leq j \leq n, \tag{17}$$

so that B can calculate $s_1, s_2, \ldots, s_n$ and thus recovers the original message as in the cryptosystem A2.

The remarks following the cryptosystem A2 apply also in the case of the cryptosystem C2. In particular, the identity (17) is shown by the same method as (14). We emphasize that the decimations employed in the cryptosystem C2 yield again power-sum sequences. The calculation of the required characteristic polynomials will be discussed in Section 4.

Finally, we present a public-key cryptosystem based on FSR sequences. This system relies on a publicly known polynomial

$$g(x) = x^n - b_{n-1}x^{n-1} - \cdots - b_1 x - b_0 \in F_p[x]$$

of degree $n \geq 2$ which satisfies $(b_0, b_1) \neq (0, 0)$, and on a publicly known FSR sequence (s_i) with minimal polynomial $g(x)$, e.g. the impulse response sequence.

D. FSR Public-Key Cryptosystem. Let p be an arbitrary prime, let $n \geq 2$, and let $g(x)$ and (s_i) be as above. The private key of correspondent A is a random integer h with $1 < h < \operatorname{ord}(g(x))$ and

$$\gcd(h, \operatorname{ord}(g(x))) = 1. \tag{18}$$

The public key of A consists of the minimal polynomial $g_h(x)$ and the string

$$t_0 t_1 \cdots t_{n-1}$$

of initial values of the decimated sequence $(t_i) = (s_{ih})$. Similarly, the pri-

vate key of correspondent B is a random integer k with $1 < k < \mathrm{ord}(g(x))$ and

$$\gcd(k,\ \mathrm{ord}(g(x))) = 1. \tag{19}$$

The public key of B consists of the minimal polynomial $g_k(x)$ and the string

$$u_0 u_1 \cdots u_{n-1}$$

of initial values of the decimated sequence $(u_i) = (s_{ik})$.

Encryption: Suppose correspondent A wants to send a message to B that consists of a string $a_0 a_1 \cdots a_{n-1}$ of n elements of F_p not all of which are 0. From B's public key, A can calculate any $v_i = u_{ih}$. Now A forms the Hankel matrix

$$V = \begin{pmatrix} v_0 & v_1 \cdots v_{n-1} \\ v_1 & v_2 \cdots v_n \\ \vdots & \vdots \quad \vdots \\ v_{n-1} & v_n \cdots v_{2n-2} \end{pmatrix}$$

and transmits to B the ciphertext consisting of the row vector

$$(a_0 a_1 \cdots a_{n-1}) V.$$

Decryption: From A's public key, B can calculate any $v_i = s_{ihk} = t_{ik}$ and so find the matrix V. Since V is nonsingular, B can recover the plaintext message $a_0 a_1 \cdots a_{n-1}$ from the ciphertext $(a_0 a_1 \cdots a_{n-1}) V$.

The only point that requires justification is the nonsingularity of the Hankel matrix V. Note that $(v_i) = (s_{ihk})$ is a decimated sequence of the sequence (s_i) with minimal polynomial $g(x)$ and that

$$\gcd(hk,\ \mathrm{ord}(g(x))) = 1$$

by (18) and (19). Furthermore, the condition $(b_0,\ b_1) \neq (0,\ 0)$ means that $g(x)$ is not divisible by x^2. It follows then from Theorem 2 in Section 6 that the minimal polynomial of (v_i) has degree n. The nonsingularity of V is now a consequence of Theorem 8.75 in Lidl and Niederreiter [17].

In the cryptosystems A1—C2, strings of consecutive terms of certain FSR sequences have to be transmitted, such as the string $t_1 t_2 \cdots t_n$ of terms of the decimated sequence $(t_i) = (s_{ik})$ in the cryptosystem A2. If n is large, this may lead to a nonnegligible probability of transmission errors. This error probability can be lowered by using the standard device of coding theory, namely to add a certain number of check symbols to the string that

we want to transmit. The natural choice in our context is to add a few subsequent terms of the FSR sequence as check symbols, such as a string $t_{n+1}\cdots t_{n+m}$ in our example. The receiver still determines the characteristic polynomial from the string $t_1 t_2 \cdots t_n$, and if the check symbols do not fit, he can ask for a retransmission. In the public-key cryptosystem D the entries of the row vector $(w_0 w_1 \cdots w_{n-1}) = (a_0 a_1 \cdots a_{n-1})V$ can be viewed as the initial values of an FSR sequence (w_i) satisfying the same recurrence as (v_i), so that subsequent terms of (w_i) can be added as check symbols. Therefore, all our FSR cryptosystems allow the possibility of *error-detecting cryptography*.

4. Computational aspects.

In this section we discuss various computational tasks that arise in the implementation of our cryptosystems. The task that occurs most frequently is that of calculating remote terms of an FSR sequence. This problem has been studied intensively in the literature. It is known that the ith term of an n-stage FSR sequence can be calculated in $O(n(\log n)\log i)$ steps. See e. g. Fiduccia [10], Gries and Levin [13], Pettorossi [24], Pettorossi and Burstall [25], Urbanek [29], and Wilson and Shortt [31] for various algorithms.

A special problem arises for power-sum sequences, which are used in Version 2 of our cryptosystems. Let (s_i) be the power-sum sequence with characteristic polynomial

$$(20) \qquad f(x) = \prod_{j=1}^{n} (x - a_j) = x^n - a_{n-1}x^{n-1} - \cdots - a_1 x - a_0 \in F_p[x],$$

so that

$$(21) \qquad s_i = \sum_{j=1}^{n} a_j^i \text{ for } i = 0, 1, \ldots.$$

We have already shown in Section 2 that (s_i) satisfies the linear recurrence relation

$$s_{i+n} = a_{n-1}s_{i+n-1} + \cdots + a_1 s_{i+1} + a_0 s_i \text{ for } i = 0, 1, \ldots.$$

For a complete numerical determination of the sequence we also need the initial values $s_0, s_1, \ldots, s_{n-1}$. In general, the definition (21) will not be convenient for computational purposes. Direct methods of calculating the

initial values in terms of the coefficients a_j in (20) would be preferable. One such method is obtained from Newton's formula (see [17, Theorem 1.75]). First we note that the jth elementary symmetric polynomial c_j in the roots $\alpha_1, \ldots, \alpha_n$ is given by

$$c_j = (-1)^{j+1} a_{n-j} \text{ for } 1 \leq j \leq n$$

according to (20). Then by Newton's formula,

$$(22) \quad s_i - s_{i-1}a_{n-1} - s_{i-2}a_{n-2} - \cdots - s_1 a_{n-i+1} - i a_{n-i} = 0 \text{ for } 1 \leq i \leq n.$$

Substituting $i = 1, 2, \ldots, n-1$ in this formula, we obtain successively the values $s_1, s_2, \ldots, s_{n-1}$, and together with $s_0 = n \in F_p$ we have then all the desired initial values.

The following method of Selmer [26], [27] for the calculation of the initial values $s_0, s_1, \ldots, s_{n-1}$ of the power-sum sequence above may also be of interest. For $0 \leq j \leq n-1$, let $(t_i^{(j)})$, $i = 0, 1, \ldots$, be the FSR sequence with characteristic polynomial $f(x)$ in (20) and with the initial values

$$t_i^{(j)} = \begin{cases} 1 \text{ if } i = j, \\ 0 \text{ if } i \neq j \text{ and } 0 \leq i \leq n-1. \end{cases}$$

Then Selmer's formula yields

$$s_i = \sum_{j=0}^{n-1} t_{i+j}^{(j)} \text{ for } i = 0, 1, \ldots.$$

Thus, the required initial values $s_0, s_1, \ldots, s_{n-1}$ can be calculated on the same FSR that is used for encryption in several of our cryptosystems.

In Version 2 of our cryptosystems we often have to solve also the inverse problem, namely that of determining a characteristic polynomial $f(x)$ of degree n of a power-sum sequence (s_i) if we are given the first $n+1$ terms $s_0, s_1, \ldots, s_n$. Here we rely again on Newton's formula. Substituting $i = 1, 2, \ldots, n$ in (22), we obtain successively the coefficients $a_{n-1}, a_{n-2}, \ldots, a_0$ of $f(x)$ since the condition $n < p$ in Version 2 guarantees that the coefficient of a_{n-i} in (22) is always a nonzero element of F_p.

In the cryptosystems A1 and C1 there arises the problem of finding the minimal polynomial $f(x)$ of degree n of an impulse response sequence (s_i), given the terms $s_n, s_{n+1}, \ldots, s_{2n-1}$. This, however, is an easy task since we have a linear recurrence relation of the form

$$s_{i+n} = a_{n-1}s_{i+n-1} + \cdots + a_1 s_{i+1} + a_0 s_i \text{ for } i = 0, 1, \ldots$$

and initial values $s_0 = \cdots = s_{n-2} = 0$, $s_{n-1} = 1$, so that substituting $i = 0, 1, \ldots, n-1$ yields successively the coefficients $a_{n-1}, a_{n-2}, \ldots, a_0$ of $f(x)$.

In several cryptosystems there occurs the problem of solving congruences modulo a large integer, such as (7) and (13). Note that these congruences are independent of the messages to be sent, so that the problem can be solved once and for all as part of the precomputation. For the sake of definiteness we discuss (7), namely

$$kd_j \equiv n+j \bmod R$$

with

$$R = p^u \operatorname{lcm}(p-1, p^2-1, \ldots, p^n-1).$$

It suffices to solve

$$ke \equiv 1 \bmod R, \tag{23}$$

for then $d_j \equiv (n+j)e \bmod R$ solves the original congruence. To solve (23) efficiently, first solve

$$ke_0 \equiv 1 \bmod p^u,\ ke_j \equiv 1 \bmod (p^j-1) \text{ for } 1 \leq j \leq n. \tag{24}$$

Then we observe that every prime power Q appearing in the factorization of R is either p^u or a factor of some p^j-1, so that a solution modulo Q is obtained from the appropriate congruence in (24). These solutions modulo the various prime powers Q can then be combined by the Chinese remainder theorem to yield a solution of (23). The task of solving the congruences in (24) can be simplified in various ways. For instance, if for some j we know a solution e_j modulo p^j-1, then for any positive divisor i of j we can use $e_i = e_j$ since p^i-1 divides p^j-1. Moreover, it is useful to write the given integer k in its digit expansion to the base p, say

$$k = k_0 + k_1 p + \cdots + k_t p^t,$$

since we have then

$$\begin{aligned} k &\equiv k_0 + k_1 p + \cdots + k_{u-1} p^{u-1} \bmod p^u, \\ k &\equiv k_0 + k_1 + \cdots + k_t \bmod (p-1), \\ k &\equiv (k_0 + k_2 + \cdots) + (k_1 + k_3 + \cdots) p \bmod (p^2-1), \end{aligned}$$

and so on. If we use the smaller values on the right-hand side of the congruences above, then the Euclidean algorithm for solving the congruences in (24) is speeded up considerably.

5. Possible attacks.

A cryptanalyst intent on attacking our cryptosystems will meet several obstacles. The obstacles are in principle the same for all the cryptosystems in Section 3, and we will discuss them in the context of the cryptosystem A2.

Suppose an attacker of the cryptosystem A2 knows a plaintext-ciphertext pair $a_0a_1\cdots a_{n-1}$ and $s_k s_{2k}\cdots s_{nk}$. Then he can form the message polynomial $f(x)$, and by Newton's formula (see Section 4) he can determine the characteristic polynomial $f_k(x)$ of degree n of the decimated power-sum sequence (s_{ik}). Next, he calculates all the roots of $f(x)$ and $f_k(x)$ in order to obtain the complete factorizations

$$f(x) = \prod_{j=1}^{n}(x-\alpha_j) \text{ and } f_k(x) = \prod_{j=1}^{n}(x-\beta_j)$$

of these polynomials in their splitting field over F_p. For large n this can already be a time-consuming exercise, even if one employs the best root-finding algorithms available at present (see Lidl and Niederreiter [17, Ch. 4]). The cryptographer knows that $\beta_j = \alpha_j^k$ for $1 \leq j \leq n$, where k is the key, but this holds only if the roots of $f(x)$ and $f_k(x)$ are paired off correctly! This is the next serious problem faced by the cryptanalyst. Of course, the roots of $f(x)$ and $f_k(x)$ come in sets of conjugates over F_p, so that actually these sets of conjugates need to be paired off correctly.

As an example, consider the situation where n is a multiple of 5 and $f(x)$ factors in the form

$$f(x) = g_1(x)g_2(x)g_3(x)g_4(x)g_5(x)$$

with $g_1(x),\ldots, g_5(x)$ being 5 distinct monic irreducible polynomials over F_p of degree $d = n/5$. If α is a root of $g_i(x)$, then $F_p(\alpha^k) \subseteq F_p(\alpha) = F_{p^d}$. But $\gcd(k, p^d-1) = 1$ by (11), thus $ke \equiv 1 \bmod (p^d-1)$ for some e, and so $(\alpha^k)^e = \alpha$ and $F_p(\alpha) \subseteq F_p(\alpha^k)$. Hence $F_p(\alpha^k) = F_{p^d}$ and α^k is a root of a monic irreducible polynomial over F_p of degree d. This means that $f_k(x)$ factors in the form

$$f_k(x) = h_1(x)h_2(x)h_3(x)h_4(x)h_5(x)$$

with $h_1(x),\ldots, h_5(x)$ being 5 distinct monic irreducible polynomials over F_p of degree d. If, say,

$$g_1(x) = \prod_{j=1}^{d} (x-\alpha_j),$$

then for some $h_i(x)$ we have

$$h_i(x) = \prod_{j=1}^{d} (x-\alpha_j^k).$$

Thus the factors $g_i(x)$ and $h_i(x)$ correspond to each other in a certain way. There are $5! = 120$ possibilities for such correspondences. It is clear how the principle contained in this example can be generalized.

If the attacker has managed to settle the pairing-off problem, so that he knows the relations

$$\beta_j = \alpha_j^k \text{ for } 1 \leq j \leq n,$$

he has to solve a discrete logarithm problem. But even if this can be solved in a reasonable amount of time, this does not in general fully determine k. For instance, in our example above where all $\alpha_j \in F_{p^d}$ with $d = n/5$, the value of k is only determined modulo the orders of the α_j in the multiplicative group $F_{p^d}^*$. Since all these orders are divisors of p^d-1, the value of k is at best determined modulo p^d-1. In general, to determine k modulo $M = \operatorname{lcm}(p-1, p^2-1, \ldots, p^n-1)$, the attacker will have to know quite a number of plaintext-ciphertext pairs.

If the attacker is "lucky", then the message polynomial $f(x)$ is irreducible over F_p. Then $f_k(x)$ is also irreducible over F_p and the pairing-off problem does not arise. The attacker simply selects an arbitrary root α_1 of $f(x)$ and an arbitrary root β_1 of $f_k(x)$, notes that

$$\beta_1 = \alpha_j^k \text{ for some } 1 \leq j \leq n$$

and that

$$\alpha_j = \alpha_1^{p^b} \text{ for some } 0 \leq b \leq n-1,$$

and so concludes that

$$\beta_1 = \alpha_1^{kp^b}. \tag{25}$$

Thus the exponent in (25) may differ from the correct value of k by a factor that is a power of p, but this does not matter in Version 2 of our cryptosystems since

$$(26) \qquad s_{ip} = \sum_{j=1}^{n} \alpha_j^{ip} = \left(\sum_{j=1}^{n} \alpha_j^{i} \right)^{p} = s_i^p = s_i \text{ for } i = 0, 1, \ldots,$$

and similarly for decimations by higher powers of p. On the other hand, it should be kept in mind that (25) at best determines k modulo p^n-1 (the "best" case arises when α_1 is a primitive element of F_{p^n}). We note also that this "lucky" case occurs rarely, since the probability that a random monic polynomial over F_p of degree n is irreducible over F_p is roughly $\frac{1}{n}$ (see Lidl and Niederreiter [17, Theorem 3.25]).

The identity (26) suggests that in Version 2 of our cryptosystems we may as well impose the additional condition $\gcd(k, p) = 1$ on the key k, since a decimation by p does not hide information. This condition appears also in Version 1 of our cryptosystems, but for a different reason.

It is evident from the discussion above that the effort of breaking our cryptosystems is greater than for cryptosystems based on discrete exponentiation, where the only task to be performed by the cryptanalyst is to solve the discrete logarithm problem. The security of our cryptosystems rests on the combined complexity of the root-finding problem, the pairing-off problem, and the discrete logarithm problem. Only in the special case $n = 1$ (which is trivial in our context and which we have therefore excluded) would the security of these cryptosystems be based solely on the presumed intractability of the discrete logarithm problem. In essence, breaking our cryptosystems means inferring the value of k from the knowledge of the polynomials $f(x)$ and $f_k(x)$. It can, of course, not be ruled out that there might be a feasible way of inferring k without determining the roots of $f(x)$ and $f_k(x)$, but it seems unlikely. In principle, the coefficients of $f_k(x)$, which are (up to sign) the elementary symmetric polynomials in $\alpha_1^k, \ldots, \alpha_n^k$ and thus symmetric polynomials in $\alpha_1, \ldots, \alpha_n$, can be expressed algebraically in terms of the coefficients of $f(x)$, which are (up to sign) the elementary symmetric polynomials in $\alpha_1, \ldots, \alpha_n$. But already for the first elementary symmetric polynomial $\alpha_1^k + \cdots + \alpha_n^k$ this expression, which is known as Waring's formula (see Lidl and Niederreiter [17, Theorem 1.76]), is very complicated.

If we work with Version 1 of our cryptosystems, then a choice such as $p = 2$ and $n = 800$ seems to offer a sufficient amount of security under present hardware and software constraints. For Version 2, where we have the condition $p > n$, it is advisable to choose p as small as we can, while having p^n roughly of the order of magnitude 2^{800}. This suggests choices

such as $p = 113$, $n = 112$, or $p = 127$, $n = 126$.

Somewhat different aspects occur in the public-key cryptosystem D in which the original message is recovered as the solution vector of a system of linear equations with coefficient matrix V. Here the attacker can break the system by determining the matrix V. If the attacker is "lucky", then there are two nonzero message vectors $(a_0 a_1 \cdots a_{n-1})$ that are scalar multiples of the unit vectors (1 0...0) and (0...0 1), respectively. Knowing

$$(a_0 a_1 \cdots a_{n-1}) V \tag{27}$$

for these two message vectors, the attacker can infer the matrix V because of the special form of a Hankel matrix. But, in general, the attacker will need to know more message vectors and the corresponding vectors in (27) in order to determine V. At any rate, low-weight messages should be avoided in this cryptosystem. We note that there are public-key cryptosystems designed especially for low-weight messages (see Chor and Rivest [4] and the author [21]).

The cryptographer can defend himself against an attack based on chosen low-weight plaintexts by only allowing message vectors that belong to a linear code L in F_p^n with relatively large minimum Hamming distance (in the sense of algebraic coding theory). This is achieved by starting from shorter message strings of length equal to the dimension of L, then using the coding scheme of L to find the corresponding code words of length n, and subsequently applying the encryption procedure to these code words. By inverting these procedures, we recover the original message. At any rate, it is clear that there are limits to this defense, for if the attacker knows the vectors in (27) for n linearly independent message vectors $(a_0 a_1 \cdots a_{n-1})$, then he can always determine V. Therefore, the cryptosystem D should only be used for the transmission of a few important messages, after which at least the key k should be changed.

We add some remarks on the choice of the polynomial $g(x)$ in the public-key cryptosystem D. This polynomial should be selected in such a way that it is difficult to infer the value of k from the knowledge of $g(x)$ and $g_k(x)$. For the reasons explained in the discussion leading to (25), the polynomial $g(x)$ should not be irreducible over F_p. This was already pointed out in [20] (see also Smeets [28]). In fact, it is preferable if $g(x)$ factors into many polynomials over F_p of small degree since this complicates the pairing-off problem. A suitable choice will be to select for p a fairly large prime and to let $g(x)$ be a product of n distinct linear factors $x - \alpha_j$

with $\alpha_j \in F_p$ for $1 \leq j \leq n$.

6. Theoretical results.

We prove three results on FSR sequences that are needed in earlier sections. We show these results for arbitrary finite fields F_q, although we only apply them for finite prime fields. Theorem 1 provides a characteristic polynomial for any decimated sequence of any FSR sequence. Such a characteristic polynomial was also given by Duvall and Mortick [8] (though a serious misprint in their main Theorem 6 should be noted), but we use a form that is more convenient for our purposes and we give a much simpler proof.

Theorem 1. *Let $\sigma = (s_i)$, $i = 0, 1, \ldots$, be an n-stage FSR sequence in F_q with characteristic polynomial $f(x) \in F_q[x]$, and let*

$$f(x) = \prod_{j=1}^{n} (x - \alpha_j)$$

be the factorization of $f(x)$ in its splitting field over F_q. Then for any integers $k \geq 1$ and $d \geq 0$ the decimated sequence $\sigma_k^{(d)} = (s_{ik+d})$, $i = 0, 1, \ldots$, is an n-stage FSR sequence with characteristic polynomial

$$f_k(x) = \prod_{j=1}^{n} (x - \alpha_j^k) \in F_q[x].$$

Proof. Since $\sigma_k^{(d)} = (\sigma_1^{(d)})_k^{(0)}$, i.e. $\sigma_k^{(d)}$ is obtained by first shifting σ by d and then decimating by k, and since $\sigma_1^{(d)}$ has again characteristic polynomial $f(x)$, it suffices to consider the case $d = 0$. Furthermore, for iteration of decimations we have $(\sigma_k^{(0)})_m^{(0)} = \sigma_{km}^{(0)}$, which corresponds to passing from $f(x)$ to $f_k(x)$ and then to $(f_k(x))_m = f_{km}(x)$. Therefore, it suffices to prove the theorem separately for $\gcd(k, p) = 1$ and for $k = p$, where p is the characteristic of the field F_q.

We first consider the case $\gcd(k, p) = 1$. Let the generating function of σ be the formal power series

$$G(x) = \sum_{i=0}^{\infty} s_i x^i.$$

Then the generating function of $\sigma_k^{(0)}$ is

$$H(x) = \sum_{i=0}^{\infty} s_{ik} x^i.$$

Since $\gcd(k, p) = 1$, there exists a primitive kth root of unity ζ over F_q, and the kth roots of unity over F_q are given by $\omega_r = \zeta^r$ for $1 \leq r \leq k$. Now

$$\sum_{r=1}^{k} \omega_r^i = \sum_{r=1}^{k} (\zeta^i)^r = \begin{cases} 0 & \text{if } i \not\equiv 0 \bmod k, \\ k & \text{if } i \equiv 0 \bmod k. \end{cases}$$

Thus

$$\sum_{r=1}^{k} G(\omega_r x) = \sum_{r=1}^{k} \sum_{i=0}^{\infty} s_i \omega_r^i x^i = \sum_{i=0}^{\infty} \left(\sum_{r=1}^{k} \omega_r^i \right) s_i x^i$$

$$= k \sum_{i=0}^{\infty} s_{ik} x^{ik} = kH(x^k).$$

By the first part of Theorem 8.40 in [17] we have

$$G(x) = \frac{g(x)}{f^*(x)}$$

with $g(x) \in F_q[x]$, $\deg(g(x)) < n$, and $f^*(x) = x^n f(1/x)$ being the reciprocal polynomial of $f(x)$. Therefore

$$kH(x^k) = \sum_{r=1}^{k} G(\omega_r x) = \sum_{r=1}^{k} \frac{g(\omega_r x)}{f^*(\omega_r x)} = \frac{h(x)}{\prod_{r=1}^{k} f^*(\omega_r x)}, \tag{28}$$

where $h(x)$ is a polynomial over $F_q(\zeta)$ with $\deg(h(x)) < kn$. Now

$$f^*(x) = x^n f\left(\frac{1}{x}\right) = x^n \prod_{j=1}^{n} \left(\frac{1}{x} - \alpha_j\right) = \prod_{j=1}^{n} (1 - \alpha_j x),$$

so that

$$\prod_{r=1}^{k} f^*(\omega_r x) = \prod_{r=1}^{k} \prod_{j=1}^{n} (1 - \alpha_j \omega_r x) = \prod_{j=1}^{n} \prod_{r=1}^{k} (1 - \alpha_j \omega_r x).$$

From

$$x^k - 1 = \prod_{r=1}^{k} (x - \omega_r)$$

we get, replacing x by $1/x$, then multiplying by x^k, and then replacing x by

$\alpha_j x$,

$$1-\alpha_j^k x^k = \prod_{r=1}^{k} (1-\alpha_j \omega_r x).$$

Therefore

$$\prod_{r=1}^{k} f^*(\omega_r x) = \prod_{j=1}^{n} (1-\alpha_j^k x^k).$$

On the other hand,

$$f_k^*(x) = x^n f_k\left(\frac{1}{x}\right) = x^n \prod_{j=1}^{n}\left(\frac{1}{x}-\alpha_j^k\right) = \prod_{j=1}^{n}(1-\alpha_j^k x),$$

so that

$$\prod_{r=1}^{k} f^*(\omega_r x) = f_k^*(x^k).$$

From (28) we get then

$$kH(x^k) = \frac{h(x)}{f_k^*(x^k)}. \tag{29}$$

The coefficients of $f_k(x)$, being symmetric polynomials in $\alpha_1, \ldots, \alpha_n$, belong to F_q, and therefore (29) shows that $h(x) = h_1(x^k)$ with $h_1(x) \in F_q[x]$, $\deg(h_1(x)) < n$. Hence

$$H(x) = \frac{h_2(x)}{f_k^*(x)}$$

with $h_2(x) \in F_q[x]$, $\deg(h_2(x)) < n$. By the second part of Theorem 8.40 in [17] it follows that $\sigma_k^{(0)}$ is an FSR sequence with characteristic polynomial $f_k(x)$.

Now we consider the case $k = p$. Let e_0 be the multiplicity of 0 as a root of $f(x)$ (where we can have $e_0 = 0$), and let $\alpha_1, \ldots, \alpha_m$ be the distinct nonzero roots of $f(x)$ with multiplicities $e_1, \ldots, e_m$, respectively. By the general formula for the terms of an FSR sequence (see [18, Ch. 6]) we have

$$s_i = u_i + \sum_{h=1}^{m} \sum_{j=0}^{e_h-1} \binom{i+j}{j} \beta_{hj} \alpha_h^i \text{ for } i = 0, 1, \ldots,$$

where $u_i \in F_q$, $u_i = 0$ for $i \geq e_0$, and β_{hj} belongs to the splitting field K

of $f(x)$ over F_q. Therefore the terms of $\sigma_p^{(0)}$ are given by

$$s_{ip} = u_{ip} + \sum_{h=1}^{m} \sum_{j=0}^{e_h-1} \binom{ip+j}{j} \beta_{hj} \theta_h^i \text{ for } i = 0, 1, \ldots, \tag{30}$$

where $\theta_h = \alpha_h^p$ for $1 \leq h \leq m$. If $rp \leq j < (r+1)p$ for some integer $r \geq 0$, i.e. $r = \lfloor j/p \rfloor$, then

$$\binom{ip+j}{j} = \frac{(ip+1)\cdots(ip+p)\cdots(ip+2p)\cdots(ip+rp)\cdots(ip+j)}{1 \quad \cdots \quad p \quad \cdots \quad 2p \quad \cdots \quad rp \quad \cdots \quad j}.$$

Canceling corresponding factors p, we get

$$\binom{ip+j}{j} = \binom{i+r}{r} A$$

with $A \equiv 1 \bmod p$, and therefore

$$\binom{ip+j}{j} \equiv \binom{i+r}{r} \bmod p.$$

Using this in (30), we obtain

$$s_{ip} = u_{ip} + \sum_{h=1}^{m} \sum_{r=0}^{\lfloor (e_h-1)/p \rfloor} \binom{i+r}{r} \theta_h^i \sum_{j=rp}^{rp+p-1} \beta_{hj}$$

with $\beta_{hj} = 0$ for $j \geq e_h$. Writing δ_{hr} for the innermost sum and putting $\delta_{hr} = 0$ for $r > \lfloor (e_h-1)/p \rfloor$, we get

$$s_{ip} = u_{ip} + \sum_{h=1}^{m} \sum_{r=0}^{e_h-1} \binom{i+r}{r} \delta_{hr} \theta_h^i.$$

For the generating function of $\sigma_p^{(0)}$ we have then

$$\begin{aligned} H(x) &= \sum_{i=0}^{\infty} s_{ip} x^i = \sum_{i=0}^{\infty} u_{ip} x^i + \sum_{h=1}^{m} \sum_{r=0}^{e_h-1} \delta_{hr} \sum_{i=0}^{\infty} \binom{i+r}{r} (\theta_h x)^i \\ &= \sum_{i=0}^{\infty} u_{ip} x^i + \sum_{h=1}^{m} \sum_{r=0}^{e_h-1} \delta_{hr} (1-\theta_h x)^{-r-1}. \end{aligned}$$

Since $u_{ip} = 0$ for $i \geq e_0$, we can write

$$H(x) = g_0(x) + \sum_{h=1}^{m} \sum_{r=1}^{e_h} \frac{\delta_{h,r-1}}{(1-\theta_h x)^r} = g_0(x) + \sum_{h=1}^{m} \frac{g_h(x)}{(1-\theta_h x)^{e_h}},$$

where $g_0(x) \in F_q[x]$, $\deg(g_0(x)) < e_0$, and $g_h(x) \in K[x]$, $\deg(g_h(x)) < e_h$ for $1 \leq h \leq m$. Bringing the last expression on a common denominator,

we get

$$H(x) = \frac{g(x)}{\prod_{h=1}^{m}(1-\theta_h x)^{e_h}}$$

with $g(x) \in K[x]$ and

$$\deg(g(x)) < \sum_{h=0}^{m} e_h = n.$$

Now

$$f_\rho(x) = \prod_{j=1}^{n}(x-\alpha_j^\rho) = x^{e_0}\prod_{h=1}^{m}(x-\theta_h)^{e_h},$$

so that

$$f_\rho^*(x) = x^n f_\rho\left(\frac{1}{x}\right) = x^n x^{-e_0}\prod_{h=1}^{m}\left(\frac{1}{x}-\theta_h\right)^{e_h} = \prod_{h=1}^{m}(1-\theta_h x)^{e_h}$$

and

$$H(x) = \frac{g(x)}{f_\rho^*(x)}.$$

Since $f_\rho(x)$ is a polynomial over F_q, it follows from the last identity that the coefficients of $g(x)$ belong to F_q, and then the second part of Theorem 8.40 in [17] shows that $\sigma_\rho^{(0)}$ is an FSR sequence with characteristic polynomial $f_\rho(x)$.

Theorem 2. *Let $\sigma = (s_i)$, $i = 0, 1, \ldots$, be an FSR sequence in F_q with minimal polynomial $m(x) \in F_q[x]$ not divisible by x^2, and let*

$$m(x) = \prod_{j=1}^{n}(x-\alpha_j)$$

be the factorization of $m(x)$ in its splitting field over F_q. Then for any integer $k \geq 1$ with $\gcd(k, \operatorname{ord}(m(x))) = 1$, the minimal polynomial of the decimated sequence $\sigma_k^{(0)} = (s_{ik})$, $i = 0, 1, \ldots$, is given by

$$m_k(x) = \prod_{j=1}^{n}(x-\alpha_j^k) \in F_q[x].$$

Proof. σ has preperiod ≤ 1 by Lemma 1(a) and least period $r = \operatorname{ord}(m(x))$ by Lemma 1(b). Since $\gcd(k, r) = 1$, there exists an integer $h \geq 1$ with $kh \equiv 1 \bmod r$. Let $\tau = \sigma_k^{(0)}$ and consider the decimated sequence $\tau_h^{(0)} = (s_{ikh})$. For $i \geq 1$, $ikh \equiv i \bmod r$ implies $s_{ikh} = s_i$ since σ has preperiod ≤ 1, and $s_{ikh} = s_i$ is trivial for $i = 0$. Therefore $\tau_h^{(0)} = \sigma$. Since we can assume that σ is not the zero sequence (the result being trivial otherwise), it follows that $\tau_h^{(0)}$ is not the zero sequence, and so τ cannot be the zero sequence. From Theorem 1 it follows that $m_k(x)$ is a characteristic polynomial of τ of degree $n \geq 1$. If τ had a characteristic polynomial of degree $< n$, then Theorem 1 implies that $\tau_h^{(0)} = \sigma$ has a characteristic polynomial of degree $< n$, a contradiction. Therefore $m_k(x)$ is the minimal polynomial of τ.

We remark that the condition that x^2 does not divide $m(x)$ is needed in Theorem 2. Let $\sigma = (s_i)$ be the sequence in F_q with $s_0 = 0$, $s_1 = 1$, and $s_i = 0$ for $i \geq 2$. Then σ is an FSR sequence with minimal polynomial $m(x) = x^2$. However, any decimated sequence $\sigma_k^{(0)}$ with $k \geq 2$ is the zero sequence and thus has minimal polynomial 1 and not $m_k(x) = x^2$.

Finally we determine the minimal polynomial of a power-sum sequence. We use the notation in (4) for such sequences.

Theorem 3. *Let (s_i) be the power-sum sequence with characteristic polynomial $f(x) \in F_q[x]$ having the canonical factorization*

$$f(x) = g_1(x)^{e_1} \cdots g_t(x)^{e_t},$$

where $e_j \geq 1$ for $1 \leq j \leq t$ and $g_1(x), \ldots, g_t(x)$ are distinct monic irreducible polynomials over F_q. Then the minimal polynomial of (s_i) is given by

$$m(x) = \prod_{j=1}^{t}{}^{*} g_j(x),$$

where the asterisk indicates that we only consider those j for which e_j is not divisible by the characteristic p of F_q, and where an empty product is interpreted to be 1.

Proof. Arrange the notation in such a way that none of $e_1, \ldots, e_u$ is divisible by p, whereas $e_{u+1}, \ldots, e_t$ are all divisible by p (here $0 \leq u \leq t$). For $1 \leq j \leq t$ let the roots of $g_j(x)$ be denoted by β_{hj} with $1 \leq h \leq d_j = \deg(g_j(x))$. Then the roots $\alpha_1, \ldots, \alpha_n$ of $f(x)$ are the elements β_{hj}, $1 \leq h \leq d_j$, $1 \leq j \leq t$, each repeated with multiplicity e_j. Now let

$$g(x) = x^k - b_{k-1}x^{k-1} - \cdots - b_1 x - b_0$$

be a polynomial over F_q of positive degree. Then in the same way as in the calculation following (4) we get for all $i \geq 0$,

$$(31) \qquad s_{i+k} - b_{k-1}s_{i+k-1} - \cdots - b_1 s_{i+1} - b_0 s_i = \sum_{j=1}^{n} \alpha_j^i g(\alpha_j)$$

$$= \sum_{j=1}^{t} \sum_{h=1}^{d_j} e_j \beta_{hj}^i g(\beta_{hj}) = \sum_{j=1}^{u} \sum_{h=1}^{d_j} e_j \beta_{hj}^i g(\beta_{hj}),$$

since for $u+1 \leq j \leq t$ we have $e_j = 0$ as an element of F_q. If $u = 0$, then this shows that any $g(x)$ is a characteristic polynomial of (s_i), hence (s_i) is the zero sequence and its minimal polynomial is 1, in accordance with our claim. If $u \geq 1$, then it follows from (31) that any $g(x)$ satisfying $g(\beta_{hj}) = 0$ for all $1 \leq h \leq d_j$, $1 \leq j \leq u$, is a characteristic polynomial of (s_i). Conversely, if $g(x)$ is a characteristic polynomial of (s_i), then with $d = d_1 + \cdots + d_u$ we get from (31),

$$\sum_{j=1}^{u} \sum_{h=1}^{d_j} \beta_{hj}^i e_j g(\beta_{hj}) = 0 \text{ for } 0 \leq i \leq d-1.$$

This can be viewed as a system of linear equations for the d elements $e_j g(\beta_{hj})$, $1 \leq h \leq d_j$, $1 \leq j \leq u$, with coefficient matrix being a Vandermonde matrix. Since the roots β_{hj}, $1 \leq h \leq d_j$, $1 \leq j \leq u$, are distinct, the Vandermonde matrix is nonsingular, and so $e_j g(\beta_{hj}) = 0$ for all $1 \leq h \leq d_j$, $1 \leq j \leq u$. For $1 \leq j \leq u$ we have $e_j \neq 0$ as an element of F_q, hence $g(\beta_{hj}) = 0$ for all $1 \leq h \leq d_j$, $1 \leq j \leq u$. Thus we have shown that $g(x)$ is a characteristic polynomial of (s_i) if and only if $g(x)$ has all the elements β_{hj} with $1 \leq h \leq d_j$, $1 \leq j \leq u$, as roots. The polynomial of least positive degree with this property is $g_1(x) \cdots g_u(x)$, and this is therefore the minimal polynomial of (s_i). Note that in our notation $g_1(x) \cdots g_u(x)$ is equal to the polynomial $m(x)$ in the statement of the theorem, and so the proof is complete.

REFERENCES

[1] H. BEKER and F. PIPER : Cipher Systems. The Protection of Communications, Northwood, London, 1982.

[2] T. BETH, P. HESS and K. WIRL : Kryptographie, Teubner, Stuttgart, 1983.

[3] I. F. BLAKE, R. FUJI-HARA, R. C. MULLIN and S. A. VANSTONE : Computing logarithms in finite fields of characteristic two, SIAM J. Algebraic Discrete Methods 5 (1984), 276 –

285.

[4] B. CHOR and R. L. RIVEST : A knapsack type public key cryptosystem based on arithmetic in finite fields, Advances in Cryptology : Proceedings of CRYPTO 84 (G. R. Blakley and D. Chaum, eds.), pp. 54－65, Lecture Notes in Computer Science, Vol. 196, Springer-Verlag, Berlin, 1985.

[5] D. COPPERSMITH : Fast evaluation of logarithms in fields of characteristic two, IEEE Trans. Information Theory **30** (1984), 587－594.

[6] D. COPPERSMITH, A. M. ODLYZKO and R. SCHROEPPEL : Discrete logarithms in GF(p), Algorithmica **1** (1986), 1－15.

[7] W. DIFFIE and M. E. HELLMAN : New directions in cryptography, IEEE Trans. Information Theory **22** (1976), 644－654.

[8] P. F. DUVALL and J. C. MORTICK : Decimation of periodic sequences, SIAM J. Appl. Math. **21** (1971), 367－372.

[9] T. ELGAMAL : A public key cryptosystem and a signature scheme based on discrete logarithms, IEEE Trans. Information Theory **31** (1985), 469－472.

[10] C. M. FIDUCCIA : An efficient formula for linear recurrences, SIAM J. Comput. **14** (1985), 106－112.

[11] S. W. GOLOMB : Sequences with randomness properties, Glenn L. Martin Co. Final Report, Baltimore, Md., 1955 ; reprinted in Golomb [12].

[12] S. W. GOLOMB : Shift Register Sequences, Holden-Day, San Francisco, 1967.

[13] D. GRIES and G. LEVIN : Computing Fibonacci numbers (and similarly defined functions) in log time, Inform. Process. Lett. **11** (1980), no. 2, 68－69.

[14] A. G. KONHEIM : Cryptography : A Primer, Wiley, New York, 1981.

[15] R. LIDL and W. B. MÜLLER : Permutation polynomials in RSA-cryptosystems, Proc. CRYPTO '83 (Santa Barbara, Cal., 1983), pp. 293－301, Plenum, New York, 1984.

[16] R. LIDL and W. B. MÜLLER : A note on polynomials and functions in algebraic cryptography, Ars Combinatoria **17A** (1984), 223－229.

[17] R. LIDL and H. NIEDERREITER : Finite Fields, Addison-Wesley, Reading, Mass., 1983.

[18] R. LIDL and H. NIEDERREITER : Introduction to Finite Fields and Their Applications, Cambridge Univ. Press, Cambridge, 1986.

[19] W. B. MÜLLER and W. NÖBAUER : Some remarks on public-key cryptosystems, Studia Sci. Math. Hungar. **16** (1981), 71－76.

[20] H. NIEDERREITER : A public-key cryptosystem based on shift register sequences, Advances in Cryptology－EUROCRYPT '85 (F. Pichler, ed.), pp. 35－39, Lecture Notes in Computer Science, Vol. 219, Springer-Verlag, Berlin, 1986.

[21] H. NIEDERREITER : Knapsack-type cryptosystems and algebraic coding theory, Problems of Control and Information Theory **15** (1986), 159－166.

[22] R. NÖBAUER : Rédei-Funktionen und ihre Anwendung in der Kryptographie, Acta Sci. Math. Szeged, **50** (1986), 287－298.

[23] A. M. ODLYZKO : Discrete logarithms in finite fields and their cryptographic significance, Advances in Cryptology : Proceedings of EUROCRYPT 84 (T. Beth, N. Cot and I. Ingemarsson, eds.), pp. 224－314, Lecture Notes in Computer Science, Vol. 209, Springer-Verlag, Berlin, 1985.

[24] A. PETTOROSSI : Derivation of an $O(k^2 \log n)$ algorithm for computing order-k Fibonacci numbers from the $O(k^3 \log n)$ matrix multiplication method, Inform. Process. Lett. **11** (1980), no. 4－5, 172－179.

[25] A. PETTOROSSI and R. M. BURSTALL : Deriving very efficient algorithms for evaluating linear recurrence relations using the program transformation technique, Acta Informatica **18** (1982), 181－206.

[26] E. S. SELMER : On Newton's equations for the power sums, BIT **6** (1966), 158－160.

[27] E. S. SELMER : Linear Recurrence Relations over Finite Fields, Lecture Notes, Univ. of Bergen, 1966.

[28] B. SMEETS : A comment on Niederreiter's public key cryptosystem, Advances in Cryptology —EUROCRYPT '85 (F. Pichler, ed.), pp. 40—42, Lecture Notes in Computer Science, Vol. 219, Springer-Verlag, Berlin, 1986.

[29] F. J. URBANEK : An $O(\log n)$ algorithm for computing the nth element of the solution of a difference equation, Inform. Process. Lett. **11** (1980), no. 2, 66—67.

[30] P. K. S. WAH and M. Z. WANG : Realization and application of the Massey-Omura lock, Proc. Internat. Seminar on Digital Communications (Zürich, 1984), pp. 175—182.

[31] T. C. WILSON and J. SHORTT : An $O(\log n)$ algorithm for computing general order-k Fibonacci numbers, Inform. Process. Lett. **10** (1980), no. 2, 68—75.

[32] N. ZIERLER : Linear recurring sequences, J. Soc. Indust. Appl. Math. **7** (1959), 31—48.

MATHEMATICAL INSTITUTE
AUSTRIAN ACADEMY OF SCIENCES
DR. IGNAZ-SEIPEL-PLATZ 2
A-1010 VIENNA, AUSTRIA

(*Received May 1, 1987*)