# A PUBLIC-KEY CRYPTOSYSTEM BASED ON SHIFT REGISTER SEQUENCES

Harald Niederreiter
Mathematical Institute
Austrian Academy of Sciences
A-1010 Vienna/Austria

Various cryptosystems using finite field arithmetic have been introduced recently, e.g. cryptosystems based on permutations of finite fields (Lidl and Müller [8], Nöbauer [12]), cryptosystems of the knapsack type (Chor and Rivest [4], Niederreiter [11]), and cryptosystems based on discrete exponentiation in finite fields (Odlyzko [13], Wah and Wang [14]). Finite fields also play a role in the construction of stream ciphers (Beker and Piper [1], Beth et al. [2], Lidl and Niederreiter [10]). The security of cryptosystems based on discrete exponentiation has recently been diminished by significant progress on the discrete logarithm problem (Blake et al. [3], Coppersmith [5], Coppersmith et al. [6]). In this paper we propose a public-key cryptosystem that has a more complex structure than the corresponding discrete-exponentiation cryptosystem and is therefore potentially harder to break. This cryptosystem uses feedback shift register (FSR) sequences in finite fields and is thus easy to implement.

To set up the cryptosystem, let $q$ be a prime power, let $F_q$ be the finite field with $q$ elements, and let

$$g(x) = x^n - b_{n-1}x^{n-1} - \ldots - b_1x - b_0$$

be a publicly known polynomial over $F_q$ with $n \geq 1$ and $b_0 \neq 0$. Let $(s_i)$ be an FSR sequence in $F_q$ with

$$s_{i+n} = b_{n-1}s_{i+n-1} + \ldots + b_1s_{i+1} + b_0s_i \quad \text{for} \quad i = 0,1,\ldots .$$

This sequence can be generated by an n-stage FSR and has characteristic polynomial $g(x)$. The basic idea of our cryptosystem is to replace discrete exponentiation by the operation of decimation for FSR sequences. By definition, the decimation of $(s_i)$ by the factor $k$ yields the sequence $(s_{ik})$, i.e. we take every $k$th term of $(s_i)$ starting from $s_0$. Let $M$ be the least positive integer such that $g(x)$ divides $x^M - 1$. If $g(x)$ is also the minimal polynomial of $(s_i)$, then the least

F. Pichler (Ed.): Advances in Cryptology - EUROCRYPT '85, LNCS 219, pp. 35-39, 1986.

period of $(s_i)$ is equal to M. We refer to [9, Ch. 8] for the necessary background on FSR sequences.

FSR Public-Key Cryptosystem. Let A and B be two correspondents in a communication system. The private key of A is a random integer h with $1 < h < M$ and $\gcd(h,M) = 1$. Let $(s_i)$ be the FSR sequence with characteristic polynomial $g(x)$ and initial values $s_0 = \ldots = s_{n-2} = 0, s_{n-1} = 1$ ($s_0 = 1$ if $n = 1$). Then the public key of A is the string $s_h s_{2h} \cdots s_{(2n-1)h}$ of $2n - 1$ elements of $F_q$.
Encryption: If B wants to send a message to A that consists of a string $a_0 a_1 \cdots a_{n-1}$ of n elements of $F_q$ which are not all 0, then B picks a random integer k with $1 < k < M$ and $\gcd(k,M) = 1$. From A's public key, B determines the minimal polynomial of the decimated sequence $(t_i) = (s_{ih})$. Thus B can calculate any $u_i = t_{ik}$. Now B forms the Hankel matrix

$$U = \begin{pmatrix} u_0 & u_1 & \cdots & u_{n-1} \\ u_1 & u_2 & \cdots & u_n \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ u_{n-1} & u_n & \cdots & u_{2n-2} \end{pmatrix}$$

and transmits to A the following two strings as ciphertexts: $s_k s_{2k} \cdots s_{(2n-1)k}$ and $(a_0 a_1 \cdots a_{n-1})U$.
Decryption: From the ciphertext $s_k s_{2k} \cdots s_{(2n-1)k}$, A determines the minimal polynomial of the decimated sequence $(v_i) = (s_{ik})$. Thus A can calculate any $u_i = s_{ihk} = v_{ih}$ and so finds the matrix U. Then A recovers the plaintext $a_0 a_1 \cdots a_{n-1}$ by postmultiplying the ciphertext $(a_0 a_1 \cdots a_{n-1})U$ by $U^{-1}$.

Some comments on this cryptosystem are in order. A task we face several times is the calculation of remote terms in an FSR sequence. This task can be solved by very efficient algorithms. For instance, a recent algorithm of Fiduccia [7] allows the calculation of the ith term of an n-stage FSR sequence in $F_q$ by $O(n(\log n)(\log i))$ arithmetic operations in $F_q$; for earlier algorithms see the references in [9, p. 458]. We note further that the given initial values of the FSR sequence $(s_i)$ guarantee that $g(x)$ is the minimal polynomial of $(s_i)$. The following general results on FSR sequences are also needed.

Lemma 1. If the characteristic polynomial $g(x)$ of an arbitrary FSR sequence $(s_i)$ in $F_q$ has the factorization $g(x) = \prod_{j=1}^{n} (x - \beta_j)$ in its splitting field over $F_q$, then the decimated sequence $(s_{ik})$ has the characteristic polynomial $g_k(x) = \prod_{j=1}^{n} (x - \beta_j^k)$, which is again a polynomial over $F_q$.

Lemma 2. If $(s_i)$ is an arbitrary FSR sequence in $F_q$ with minimal polynomial $g(x)$, if $\gcd(k,M) = 1$ and $x^2$ does not divide $g(x)$, then $(s_{ik})$ has the minimal polynomial $g_k(x)$.

These two lemmas show that all the decimated sequences appearing in our cryptosystem are again n-stage FSR sequences in $F_q$ with minimal polynomials of degree n. It is known that the minimal polynomial of an n-stage FSR sequence can be calculated quickly from the first $2n$ terms of the sequence, e.g. by the Berlekamp-Massey algorithm (see [9, Ch. 8]). In our case, the first $2n$ terms of the relevant sequences are always available since the first term $s_0$ is known anyway and the next $2n - 1$ terms are either published or transmitted over the channel. In the deciphering procedure we have to make use of the nonsingularity of the matrix U, which follows from the fact that the sequence $(u_i)$ has a minimal polynomial of degree n and from [9, Theorem 8.75].

In the special case $n = 1$ our cryptosystem reduces to one based on discrete exponentiation. The presence of the free parameter n allows for a greater flexibility and for a more complex structure as compared to a discrete-exponentiation cryptosystem. An additional advantage is the possibility of <u>error-correcting cryptography.</u> This means that if the channel $B \rightarrow A$ is noisy, then we can add some check symbols to the ciphertexts in a natural way to reduce the probability of transmission errors. In detail, we take the ciphertext $v_1 v_2 \cdots v_{2n-1}$ and add to it a string $v_{2n} v_{2n+1} \cdots v_{2n+m}$ of subsequent terms of the FSR sequence $(v_i)$ as check symbols. The receiver A still determines the minimal polynomial of $(v_i)$ from the string $v_1 v_2 \cdots v_{2n-1}$, and if the check symbols do not fit, he can ask for a retransmission. We note also that there is a <u>second version</u> of the FSR public-key cryptosystem in which the sequence $(s_i)$ is replaced by the power-sum sequence associated with $g(x)$, i.e. the sequence which in the notation of Lemma 1 is given by $s_i = \sum_{j=1}^{n} \beta_j^i$ for $i = 0,1,\ldots$ . In this case the strings of length $2n - 1$ can be replaced by strings of length n, but on the other hand we can only work with fields $F_q$ of characteristic $p > n$.

A cryptanalyst can basically pursue two lines of attack against the FSR public-key cryptosystem. The first type of attack is directed against the keys h and k. This amounts to inferring the value of k from knowledge of the polynomials $g(x)$ and $g_k(x)$ in Lemma 1. The following three steps are required:

(i) calculating the roots of $g(x)$ and $g_k(x)$;

(ii) pairing off the roots of $g(x)$ with those of $g_k(x)$ in a correct manner (which may require up to $n!$ trials);

(iii) solving discrete logarithm problems in various extensions of $F_q$.

The least favorable case is the one where $g(x)$ is irreducible over $F_q$, for then step (ii) is not needed and the problem of inferring k is equivalent to a discrete logarithm problem. The polynomial $g(x)$ should be chosen in such a way that a large value of M is obtained and the factorization of $g(x)$ into irreducibles over $F_q$ is fairly complicated. A good choice for $g(x)$ appears to be the following: let q be a large prime and let $g(x)$ be a product of many irreducibles over $F_q$ of small degree such that a large value of M is obtained. Alternatively, we could use $q = 2$

and let $g(x)$ be a product of many irreducibles over $F_2$ of moderately large degree such that a large value of $M$ is obtained.

The second line of attack is aimed at a direct determination of the matrix $U$. This succeeds immediately if there are two message strings $a_0a_1 \ldots a_{n-1}$ that are scalar multiples of $10 \ldots 0$ and $0 \ldots 01$, respectively, for then a knowledge of the corresponding ciphertexts $(a_0a_1 \ldots a_{n-1})U$ determines $U$ completely (because of the special structure of a Hankel matrix). In general, the system is vulnerable to this type of attack if there are several messages of low weight. We can defend against this attack by using a different enciphering scheme for low-weight messages, such as those in [4] and [11] designed especially for low-weight messages. Another defense is based on first encoding all messages via a linear code $C$ over $F_q$ of length $n$, dimension $d < n$, and large minimum distance (in the sense of algebraic coding theory). The resulting system works as follows. The acceptable messages consist of nonzero strings of $d$ elements of $F_q$. Each message is transformed via the coding scheme of $C$ into a code word $a_0a_1 \ldots a_{n-1}$, which is then postmultiplied by $U$ to get the second ciphertext in the FSR public-key cryptosystem. Decryption proceeds by determining $U$ as before from the first ciphertext, postmultiplying the second ciphertext by $U^{-1}$ to recover $a_0a_1 \ldots a_{n-1}$, and then applying the inverse coding scheme to recover the original message. By using the code $C$ we make sure that each string $a_0a_1 \ldots a_{n-1}$ entering the encryption phase has a relatively large weight. In all cases it should be noted that $U$ is determined once we have used $n$ linearly independent message strings, and so the value of the key $k$ should be changed before that.

The principle employed in the design of our cryptosystem, namely to replace discrete exponentiation by the more complex operation of decimation of FSR sequences, can be used to construct other cryptosystems. We have obtained in this way new conventional cryptosystems, key-exchange schemes, and variants of Shamir's no-key algorithm. The most intricate of these cryptosystems work with message-dependent FSR sequences.

## References


1. H. Beker and F. Piper: Cipher Systems. The Protection of Communications, Northwood Books, 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. Alg. Discr. Methods 5, 276-285 (1984).
4. B. Chor and R. L. Rivest: A knapsack type public key cryptosystem based on arithmetic in finite fields, Proc. CRYPTO '84, to appear.
5. D. Coppersmith: Fast evaluation of logarithms in fields of characteristic two, IEEE Trans. Inform. Theory 30, 587-594 (1984).
6. D. Coppersmith, A. M. Odlyzko, and R. Schroeppel: Discrete logarithms in GF(p), preprint.
7. C. M. Fiduccia: An efficient formula for linear recurrences, SIAM J. Comput. 14, 106-112 (1985).

8. R. Lidl and W. B. Müller: A note on polynomials and functions in algebraic cryptography, Ars Combin. 17A, 223-229 (1984).
9. R. Lidl and H. Niederreiter: Finite Fields, Addison-Wesley, Reading, Mass., 1983.
10. R. Lidl and H. Niederreiter: Introduction to Finite Fields and Their Applications, Cambridge Univ. Press, in press.
11. H. Niederreiter: Knapsack-type cryptosystems and algebraic coding theory, Problems of Control and Information Theory, to appear.
12. R. Nöbauer: Rédei-Funktionen und ihre Anwendung in der Kryptographie, Acta Sci. Math. Szeged, to appear.
13. A. M. Odlyzko: Discrete logarithms in finite fields and their cryptographic significance, Proc. EUROCRYPT '84, to appear.
14. P. K. S. Wah and M. Z. Wang: Realization and application of the Massey-Omura lock, Proc. Intern. Sem. on Digital Communications (Zürich, 1984), pp. 175-182.