1989

Algebraic coding for communications security /

Andrew Cameron Duke
Lehigh University

ALGEBRAIC CODING FOR COMMUNICATIONS SECURITY

by

Andrew Cameron Duke

A Thesis
Presented to the Graduate Committee
of Lehigh University
in Candidacy for the Degree of
Master of Science
in
Electrical Engineering

Lehigh University
1989

The following thesis is approved for partial fulfillment of the Master's of Science in Electrical Engineering.

Kenneth K. Tzeng

Professor Kenneth K. Tzeng
Advisor
CSEE Department

Donald Talhel

Professor Donald T. Talhelm
Division Head
Electrical Engineering

Lawrence J. Varnerin Jr.

Professor Lawrence J. Varnerin Jr.
Department Chair
CSEE Department

ACKNOWLEDGEMENTS

I would like to thank my advisor, Professor Kenneth Tzeng, for his help and guidance in the preparation in the course of my research. I would also like to thank René Struik for his help by sending a copy of the article "The Rao-Nam Scheme is Insecure Against a Chosen-Plaintext Attack." Finally I would like to thank C.S. Park for his help in compiling source articles.

The support of the National Science Foundation is also gratefully acknowledged.

A.C. Duke

TABLE OF CONTENTS

FIGURES

ABSTRACT

Most modern communication systems require that messages be sent securely. While there exist many cryptographic methods with which to provide this security, implementation making use of error-control codes lends itself to the possibility of also supplying reliability to the system. There are two prevalent methods for using error-control codes in the design of cryptographic systems. The first, based on a scheme proposed by McEliece and adapted by Rao-Nam, relies on the use of a scrambled code generator matrix to provide encryption, while the intractability of decoding arbitrary codes provides the security. The second, proposed by Niederreiter, relies on the decimation of feedback shift register sequences to generate an encryption matrix, whereas the difficulty of determining these factors provides the security. A characterization of the cryptosystems based on the McEliece system is proposed. A generalization of the Niederreiter Feedback Shift Register Cryptosystem that makes use of multiple sequences is also proposed. The suitability of both general types of systems towards solving the authentication problem as well as their application to joint encryption and error correction is discussed.

Chapter I - Introduction

Since the introduction of modern communications in the nineteenth century the study of cryptography, the science of protecting communications, has grown. With society today having an increased dependence upon electronic commincations such as radio, facsimile and high-speed data communications, communication security has become even more important.

There are two primary methods of providing security. The first is to have impenetrable lines between every pair of users in the system. However; this is impractical from both a cost and implementation viewpoint. The second more practical method is to alter in some way the appearance of any message so that its content is concealed from all but the intended recipient. It is this method that will be the focus of discussion. Such systems are referred to as cipher or cryptosystems.

While there are many possible methods for implementing the second option, cryptographic systems that rely on ideas from error-control coding for their implementation are of value for they also suggest the ability to provide reliability for the communication system as well.

This type of cryptosystem was first proposed by McEliece [18] in 1978, and used the intractability of decoding general error-correcting codes as the basis for its security. In 1983 Jordan [11] proposed a variant of this system. Both of these systems are public-key cryptosystems, a form of cryptosystem proposed by Diffie and Hellman [7] in 1976. Rao and Nam [27] offered a private-key or traditional cryptosystem variation of the McEliece system in 1986. Their system has since has modifications proposed by Struik and van Tilburg [29] in 1987, as well as Denny and Rao [6] in 1988, in efforts to improve upon the security of the system.

A second type of cryptosystem based on the ideas of algebraic codes was proposed by Niederreiter [21] in 1985. His system involves making use of feedback shift register (FSR) sequences and decimating them by privately known factors in order to provide the system security. This system is a generalization of the discrete logarithm cryptosystem.

A third type of system, prposed by Niederreiter [20] in 1986, is based on the knapsack problem. Again this system relies on the difficulty in decoding a general algebraic code.

While an attempt has been made to be exhaustive, recent developments such as the work by Park and Tzeng [22], using concatanated codes on an attempt to solve the JOEEC problem, are not included.

The remainder of this thesis is organized as follows.

Chapter II is a review of the background material needed in order to describe the cryptosystems based on error-control coding. Section one is devoted to error-control coding. Included are reviews of the basics of linear and cyclic codes as well as the encryption and decryption of BCH, Goppa and non-linear codes. Section two is a review of the basics of cryptology. Included are discussions of both private and public key cryptosystems, as well as the rudiments of cryptanalysis specifically what is meant by a system being insecure. Mention is also made of the authentication problem and joint encryption and error-correcting. Chapter two concludes with section three, where feedback shift registers are described. Included in this section are the notion of the characteristic polynomial, families of sequences, and the synthesis of a feedback shift register to generate a multiple sequence system.

Chapter III is an analysis of the feedback shift register cryptosystem as originally suggested by Niederreiter. Section one is a description of his system. Section two is a description of our proposed generalization relying on multiple sequences, for which an example is given. Section three is a cryptanalysis of the generalized system. The chapter concludes with a mention of the systems applicability to the authentication problem.

Chapter IV is devoted to the McEliece public key cryptosystem including the variation by Jordan. Mention is also made of its private key counterpart suggested by Rao-Nam and its variants by Struik-Van Tilburg and Rao-Denny. A characterization of these systems is made so that all possible variants, both public and private key can be discussed with one encryption algorithm. Finally the suitability of this family of cryptosystems to the joint encryption error-correction problem and their insuitability to solving the authentication problem is mentioned.

Chapter V deals with Niederreiter's knapsack cryptosystem. Following a brief introduction to the knapsack problem, section one is a presentation of the cryptosystem. Section two presents two attacks on the system, showing its vulnerability.

Chapter VI is a summary of results. A comparison of the three basic types of cryptosystems, FSR, McEliece, and Niederreiter knapsack is also offered.

For further information regarding algebraic coding the reader is referred to sources 14, 15, 17, 30 and 31. For further information regarding cryptology the reader is referred to sources 2, 4, and 23. For information regarding feedback register sequences the reader is referred to source 10. Finally for general background sources 12 and 13 are recommended.

Chapter II - Preliminaries

In this chapter some preliminary concepts that are needed to discuss communication security using algebraic codes are presented. We begin with a review of error-correcting codes, including Goppa codes, BCH and Reed-Solomon codes as well as non-linear codes. Both encoding and decoding for these classes of codes will be described. Goppa codes are essential as they are used in the McEliece system, while non-linear codes are used in a variation of the Rao-Nam scheme. BCH and Reed-Solomon codes are used in applications to the joint encryption error-correction (JOECC) problem.

Following a discussion of coding theory the basic concepts of cryptology are described. Included in this is the difference between public and private-key encryption. Also included in this area is the notion of authentication. The idea of JOEEC is also presented.

This chapter concludes with a presentation of feedback shift register sequences, which are used by Niederreiter to perform public key encryption. Various important properties of these sequences will be discussed.

II.1 Fundamentals of Error-Correcting Codes

In communication systems (see figure 2.1) it is desirable to have the ability to correct any errors introduced due to noise in the channel. It is for this reason that error-correcting codes were developed.

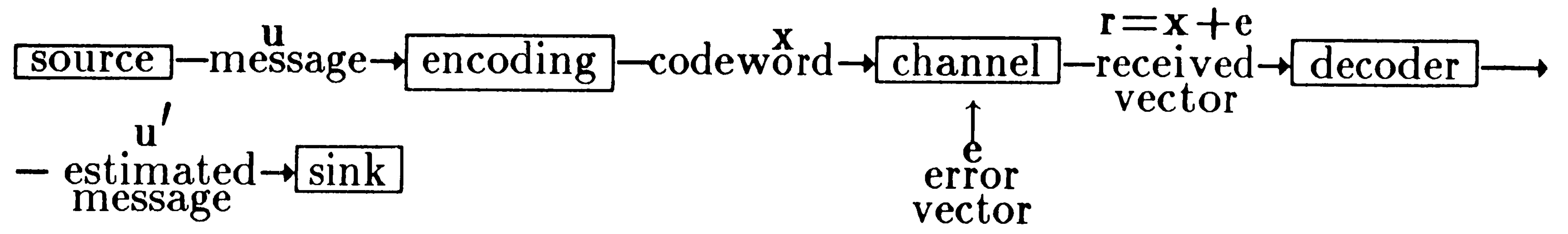

Figure 2.1 Communication System

Let u and v be elements of $\mathbb{F}_q^n$. Define the Hamming Weight of x, w(u), to be the number of non-zero entries of u. The Hamming Distance between u and v, denoted d(u,v), is defined to be the number of cooridinates in u and v that are different, that is $d(u, v)=|\{i \mid u_i \neq v_i,\ 1 \leq i \leq n\}|$. We then have $w(u)=d(u, 0)$.

Over a finite field $\mathbb{F}_q$, a linear (n, k, d) code C is a k-dimensional subspace of $\mathbb{F}_q^n$ whose elements have minimum distance d. The generator matrix, G, of the code is an $k \times n$ matrix whose rowspace is C. If G is of the form $(P\ I_k)$ then it is said to be in standard form, and from G we obtain the parity check matrix $H=(\ I_{n\text{-}k}\ \ -P^T)$. For any $u \in C$, $uH^T = 0$ and for $u \in \mathbb{F}_q^n$, uH^T is called the syndrome of u. A code capable of correcting up to t errors is known as a t-error-correcting code; note that $d_{min} \geq 2t+1$.

For a code C define the minimum distance $d_{min}=\min\{d(u, v)|u, v \in C\}$, and minimum weight $w_{min}=\min\{w(u) \mid u \in C, u \neq 0\}$. If the code is linear, since $d(u, v) = d(u-v, 0) = w(u-v)$ and $u, y \in C$ implies $u-v \in C$ the minimum distance is equal to the minimum weight of C.

A code C is *cyclic* if $(c_0, c_1, \ldots, c_{n-1}) \in C$ implies $(c_1, c_2, \ldots, c_{n-1}, c_0) \in C$. For cyclic codes, a codeword can be viewed as a polynomial, namely $(c_0, c_1, \ldots, c_{n-1})$ can be viewed as $c(x) = c_0 + c_1x + \cdots + c_{n-1}x^{n-1}$. Then a linear code C is cyclic if and only if C is an ideal in $\mathbb{F}_q[x]/(x^n-1)$. Since $\mathbb{F}_q[x]/(x^n-1)$ is a principal ideal ring, it has a generator polynomial, this polynomial is the generator polynomial of the code and is a divisor of x^n-1. If the code has as its generator polynomial $g(x) = g_0 + g_1x + \cdots + g_{n\text{-}k}x^{n\text{-}k}$, then it has as its parity check polynomial

$h(x) = \frac{x^n - 1}{g(x)} = h_0 + h_1 x + \cdots + h_k x^k$. The generator and check matrices are determined as below.

$$G = \left[\begin{array}{cccccccc} g_0 & g_1 & \cdots & g_{n-k} & 0 & 0 & \cdots & 0 \\ 0 & g_0 & g_1 & \cdots & g_{n-k} & 0 & \cdots & 0 \\ 0 & \vdots & & & \cdots & & & 0 \\ 0 & 0 & \cdots & & g_0 & g_1 & \cdots & g_{n-k} \end{array}\right]$$

$$H = \left[\begin{array}{cccccccc} 0 & 0 & \cdots & 0 & h_k & \cdots & h_1 & h_0 \\ 0 & 0 & \cdots & h_k & h_{k-1} & \cdots & h_0 & 0 \\ \vdots & & & & & & & \vdots \\ h_k & \cdots & h_1 & h_0 & 0 & 0 & \cdots & 0 \end{array}\right]$$

II.1.1 BCH and Reed-Solomon Codes

An important class of cyclic codes are the BCH codes. To generate these codes let $\mathbb{F}_q$ be an arbitrary field, let g(z) be a generator polynomial that is the least common multiple of the minimal polynomials of α^b, α^{b+1}, ..., $\alpha^{b+\delta-2}$, for an arbitrary b and α a primitive n^{th} root of unity. This gives a BCH code of designed distance δ, the minimum distance of these of codes is at least δ. If b = 1 the narrow-sense BCH codes are obtained and when $n = q^m - 1$ the code is said to be primitive.

A subclass of the BCH codes are the Reed-Solomon codes. These codes are primitive BCH codes of length $n = q-1$ over $\mathbb{F}_q$. The generator for this code is $g(z) = \prod_{i=1}^{d-1}(z-\alpha^i)$, with α primitive in $\mathbb{F}_q$. This is an (n, n−d+1, d) linear code.

The parity check matrix for these codes is given by the following

$$H = \begin{bmatrix} 1 & \alpha^{b} & \alpha^{2b} & \cdots & \alpha^{(n-1)b} \\ 1 & \alpha^{b+1} & \alpha^{2(b+1)} & \cdots & \alpha^{(n-1)(b+1)} \\ \vdots & & \vdots & \ddots & \vdots \\ 1 & \alpha^{b+\delta-2} & \cdots & & \alpha^{(n-1)(b+\delta-2)} \end{bmatrix}$$

To decode a (n, k, d) BCH code with generator g(X) and design distance d, let $\mathbf{r} = (r_0, r_1, \ldots, r_{n-1})$; $r_i \in \mathbb{F}_q$ be the received vector for a sent message vector **f**. Then **e** $= \mathbf{r} - \mathbf{f}$ is the error pattern, then $r(X) = f(X) + e(X)$, where r(X), e(X) and f(X) are the polynomials associated with **r**, e and f respectively. Associate with e(X) the symmetric functions $S_1, S_2, \ldots$ where $S_i = e(\alpha^i)$. Since g(X) divides f(X) it then follows that $S_i = r(\alpha^i)$, $i = b, b+1, \ldots, b+d-2$, and the set of the $d-1$ consecutive S_i can be formed by the receiver. The decoding problem is given the set of S_i, find the error polynomial e(X). Let e have Hamming weight t, and the j^{th} non-zero component of e be e_k, $X_i = \alpha^k \in \mathbb{F}_{q^m}$ the locator of the error and $Y_i = e_i \in \mathbb{F}_q$ the error magnitude. Thus $S_i = \sum_{j=1}^{t} Y_j X_j^i$, for $i = 1, 2, 3, \ldots$. The essential problem is the determination of the error locations. To do this determine the connection polynomial C(D) for the feedback shift register of least length that will generate the sequence $S_b, S_{b+1}, \cdots, S_{b+d-2}$, (see §3.1 for the Berlekamp-Massey algorithm to accomplish this task). The t roots of C(D) are the recipricals of the error locators, thus decoding is possible if $2t \leq d-1$.

II.1.2 Goppa Codes

If G(z) is a polynomial of degree t over $\mathbb{F}_{q^m}$ and $L = \{\gamma_0, \gamma_1, \ldots, \gamma_{n-1}\}$ is a subset of $\mathbb{F}_{q^m}$ with $G(\gamma_i) \neq 0$ for $i = 0, 1, \ldots, n-1$, then the Goppa code, $\Gamma(L, G)$, with Goppa polynomial G(z) is the set of codewords $\mathbf{c} = (c_0, c_1, \ldots, c_{n-1})$ over $\mathbb{F}_q$ satisfying $\sum_{i=0}^{n-1} \frac{c_i}{z-\gamma_i} \equiv 0(\mathrm{mod}(G(z)))$. This class of codes is linear, have dimension $k \geq n-mt$ and minimum distance at least $t+1$.

The parity check matrix for the Goppa codes is given by the following,

$$H = \begin{bmatrix} h_0 & h_1 & \cdots & h_{n-1} \\ h_0\gamma_0 & h_1\gamma_1 & \cdots & h_{n-1}\gamma_{n-1} \\ \vdots & \vdots & & \vdots \\ h_0\gamma_0^{t-1} & h_1\gamma_1^{t-1} & \cdots & h_{n-1}\gamma_{n-1}^{t-1} \end{bmatrix}$$

where $h_j = G(\gamma_j)^{-1}$.

Example: Let L = GF(3) and $G(z) = z^2+z+2 =(z-\alpha)(z-\alpha^3)$, where α is primitive in GF(3^2) and $\alpha^2+\alpha+2 = 0$. The parity check matrix is

$$H = \begin{bmatrix} 2 & 1 & 2 \\ 0 & 1 & 1 \end{bmatrix}.$$

This is a (3, 1) linear code over GF(3) that consists of the codewords {(0, 0, 0), (1, 2, 1), (2, 1, 2)}.

To decode, make use of Euclid's algorithm, a recursive technique for finding the greatest common divisor d(z) of two polynomials a(z) and b(z), namely producing the equation

$$s(z)a(z) + t(z)b(z) = d(z)$$

which expresses d(z) as a linear combination of a(z) and b(z).

Let $r = (r_0, r_1, \ldots, r_{n-1})$ be the received vector and $e = r-c$, where c is the transmitted codeword, the error vector. The syndrome s(z) of r is the unique polynomial of degree less than t such that

$$s(z) \equiv \sum_{i=0}^{n-1} \frac{r_i}{z-\alpha_i} (\mathrm{mod}(G(z)) \equiv \sum_{i=0}^{n-1} \frac{e_i}{z-\alpha_i} (\mathrm{mod}(G(z)); \; \alpha_i \in L.$$

Let $B = \{\alpha_i | \; e_i \neq 0\}$ be the set of error locations with $e_\beta = e_i$, where $\beta = \alpha_i$ is the value of the error at location e_β. Decoding is the process of determining the error locations and values. Rewrite the syndrome as

$$s(z) \equiv \sum_{\beta \varepsilon B} \frac{e_\beta}{z-\beta} \, \mathrm{mod}(G(z))$$

and define, the error-locator and error-evaluator polynomials as

$$\sigma(z) = \prod_{\beta \varepsilon B} (z-\beta), \text{ and}$$

$$\omega(z) = \sum_{\beta \varepsilon B} e_\beta \cdot \prod_{\substack{\gamma \varepsilon B \\ \gamma \neq \beta}} (z-\gamma), \text{ respectively.}$$

If B has e elements then $\deg(\sigma) = e$, $\deg(\omega) < e$, $\gcd(\sigma, \omega) = 1$, $e_\beta = \omega(\beta)/\sigma'(\beta)$, and $\sigma(z)s(z) \equiv \omega(z) \; (\mathrm{mod}(G(z))$.

If $\sigma(z)$ and w(z) are the error locator and evaluator polynomials respectively, and the error pattern has weight $\leq \left(\frac{t}{2}\right)$ then $\sigma(z) = \lambda t_j(z)$ and $w(z) = \lambda r_j(z)$, where $r_j(z)$ and $t_j(z)$ are obtained from Euclid's algorithm with a(z) =sG(z) and b(z) = S(z) and j is the least integer such that $\deg(r_j) < \left(\frac{t}{2}\right)$. The scalar $\lambda \in \mathbb{F}_{q^m}$ is chosen so that $\lambda t_j(z)$ is monic.

To continue, find the solutions to $\sigma(z) = 0$ in $\mathbb{F}_{q^m}$, these are the values of the error locations. The error values are found by finding $e_\beta = \omega(\beta)/\sigma'(\beta)$, where $\sigma'(z)$ is the formal derivative of $\sigma(z)$. If q = 2 this last step is not necessary since the error values will always be one.

II.1.3 Non-Linear Codes

Linear codes have the property that the sum of any two codewords is also a codeword. However, there exist codes that do not have this property, non-linear codes. An (n, M, d) non-linear code is a set of M vectors of lerngth n such that any two have at least d positions in which they do not agree.

For an example of a specific non-linear code consider the class of non-linear codes described by Preparata[21]. Let $q = 2^{m-1}$ for m greater than 2, $a(x) \in \mathbb{F}_2[x]/(x^{q-1}+ 1)$, and $B = \{b(x)\}$ a single error-correcting Reed-Solomon code of length $q-1$ generated by $g_1(x)$ with α primitive in $\mathbb{F}_q$ as a root. Let $C = \{c(x)\}$ be the BCH code whose generator polytnomial has roots α, α^3, and 1. Finally let $u(x) = (x^{q-1}+ 1)/(x+1)$. The code V consists of vectors of the form

$$v = [b(x), i, b(x) + \{b(1) + i\}u(x) + c(x)]$$

where i is a binary parameter. The code is a $(2^m-1, 2^m-3m+1)$ linear code of distance six.

If $z(x) = (x^{q-1}+ 1)/g_1(x)$, then there exists an s such that $x^s z(x) = (x^s z(x))^2$. Let $f(x) = x^s z(x)$. Then the set K, of vectors **w** such that

$$\mathbf{w}=[b(x) + p(x), i, b(x) + p(x)f(x) + \{b(1) + i\}u(x) + c(x)\}$$

with p(x) a monomial such that $\deg(p) \leq q-2$, and B(x), c(x), i and p(x) are independently chosen is a $(2^m-1, 2^m-2m, 5)$ non-linear code.

To decode assume that $\mathbf{r} = [r_0(x), r, r_1(x)] = \mathbf{w} + [e_0(x), e, e_1(x)]$ is received. Given the following:

$H_1 = [\alpha^{q-2}, \alpha^{q-3}, \ldots, \alpha, 1]$

$H_3 = [(\alpha^3)^{q-2}, (\alpha^3)^{q-3}, \ldots, (\alpha^3), 1]$

$$U = [1, 1, \ldots, 1, 1]$$

then the syndrome is calculated as follows.

$$\sigma_0 = r_0(x)H_1^T = a\alpha^S + e_0(\alpha)$$

$$\sigma_1 = r_1(x)H_1^T = a\alpha^S + e_1(\alpha)$$

$$\sigma = (r_0(x) + r_1(x))H_3^T$$

$$d = r + r_1(x)U^T$$

where $p(x) = ax^S$ is the monomial used in the codeword. Then $\sum = (\sigma_0, \sigma_1, \sigma, d)$ is the syndrome for **r**.

Let $\rho = \sigma + (\sigma_0 + \sigma_1)^3$. If $\rho = \sigma_j$ $(j = 1, 2)$ and $d = 0$ then **r** is a member of the non-linear code. If the above condition is not met then let $\mathbf{c} = [c_0(x), c, c_1(x)]$ be such that **c**+**r** is a codeword. It remains to find **c**, which is possible according to the following rules, taking j modulo 2;

Rule 1: If $\rho = \sigma_j$ and $\rho \neq \sigma_{j+1}$ then $c_{j+1}(x) = x^h$ where $\alpha^h = \sigma_0 + \sigma_1$ and

$$c = d + c_1(1).$$

In the following $\rho \neq \sigma_j^3$ for $j = 1, 2$.

Rule 2: If $d = 1$ then $c = 0$ and $c_j(x) = x^{k_j}$, where

$$\alpha^{k_j} = \sigma_{j+1} + 3\sqrt{\sigma + \sigma_0\sigma_1(\sigma_0 + \sigma_1)}.$$

Rule 3: If $d = 0$ and $\sigma_0 + \sigma_1 \neq 0$ then $c = 0$, $c_j(x) = 0$ and $c_{j+}(x) = x^{k_1} + x^{k_2}$,

where α^{k_1} and α^{k_2} are the solutions to

$$z^2 + (\sigma_0 + \sigma_1)z + \frac{\rho + \sigma_j^3}{\sigma_0 + \sigma_1} = 0.$$

Rule 4: If $d = 0$ and $\sigma_0 + \sigma_1 = 0$ then **r** is at distance at least three from any codeword.

II.2 OVERVIEW OF CRYPTOLOGY

In order to provide secure communications it is necessary to disguise the information sent. The system which does this is referred to as a cipher system, the original message as plaintext and the enciphered message as ciphertext. The cipher system can be thought of as a mapping from the plaintext to the ciphertext. The set of all possible plaintexts is called the message space, denoted by M,and the set of all possible ciphertexts is the cryptogram space, denoted by C. The mapping from M into C should be injective, that is each message uniquely determines a cipher, this will guarantee that the cryptogram is decipherable back to the original message. There will be more than one such mapping E_k: M → C, index these mappings with a key k, the set of possible indexes is called the keyspace. By abuse of notation the specific mapping used is sometimes referred to as the key.

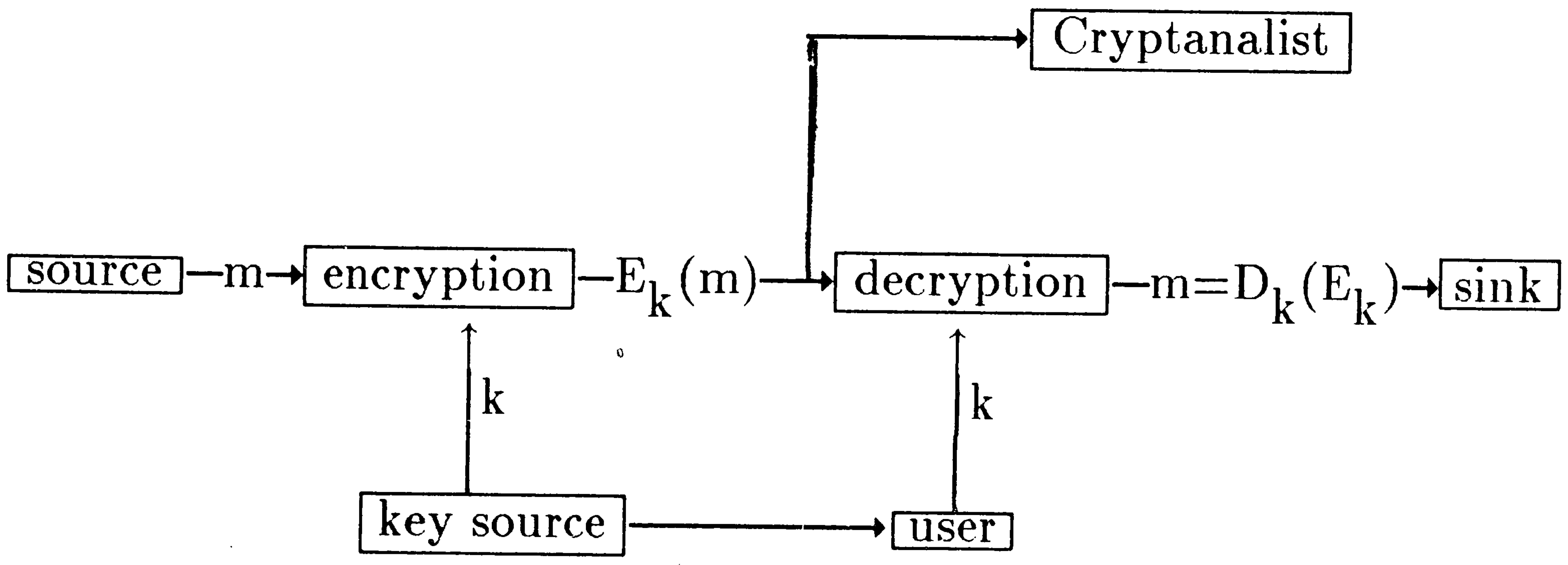

Figure 2.2 Private-Key Cryptosystem

We will only concern ourselves with block cipher systems, in which the message to be sent is first broken into blocks of n characters for some appropriate positive integer n.

Conventional cryptosystems in which the key is kept private are referred to as private-key cryptosystems (see figure 2.2). In these systems it is necessary to distribute the key to be used in that cryptographic session prior to the session in such a way as they are only known to the recipient, such as by courier.

II.2.1 Public-Key Cryptography

As an alternative to private-key systems Diffie and Hellman [7] propsed using publically distributed keys (see figure 2.3). To implement, all members who wish to communicate together decide on a common encryption algorithm, then each user determines and publishes an encryption key. For user A to send a message to user B, first A looks up B's key, then encrypts the message under that key. Upon receipt B then decrypts the message. In order for systems of this type to work it is necessary that the decryption algorithm not use the same key as the encryption algorithm. It must still be necessary for B to easily decrypt A's message. What is needed are encryption algorithms that are hard to invert, but with the introduction of additional information inversion is possible. Such functions are called one-way trap door functions, and in the determination of the public keys, users must keep the trap door information that is needed to invert the encryption key private.

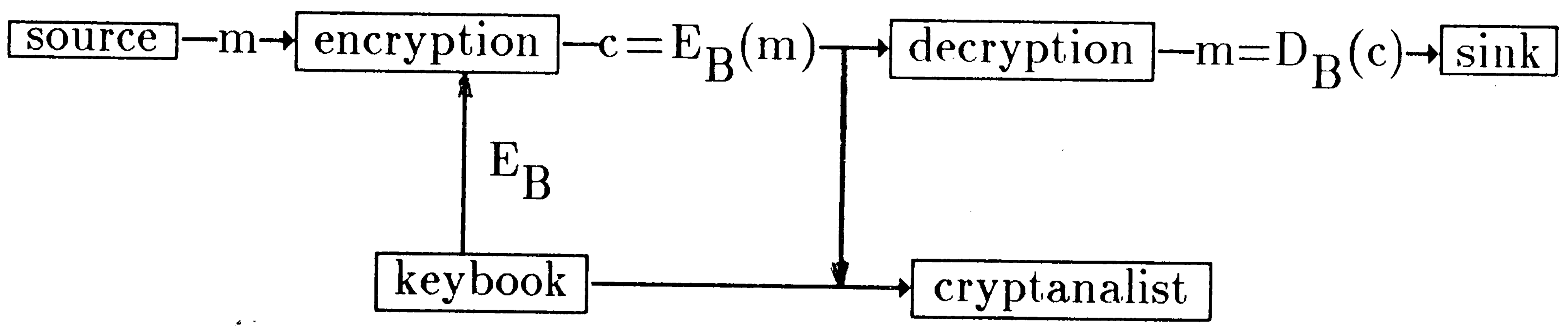

Figure 2.3 Public-Key Cryptosystem

The rules for public-key cryptography are summarized below.

(P1) If c = E(m) then m = D(c) and D(E(m)) = m for each m $\in$ M, where E (D) is the encryption (decryption) algorithm.

(P2) E and D must be quickly and easily applied.

(P3) E can be made public without revealing D, that is it is not computationally feasible to derive D from E.

II.2.2 Cryptanalysis

In the design of any cryptographic system, the question of how secure is that system must be answered. It is in the answering of this question that cryptanalysis of the system must be done. The cryptanalist is th e person whom is connected to the system in an attempt to either intercept and decipher messages, passive cryptanalysis or eavesdropping; or corrupt the message traffic itself in an attempt to make it undecipherable by any legitinmate recipient, active cryptanalysis or tampering. It is primarily with passive cryptanalysis that is the concern here.

There are three levels of cryptanalysis, concerned with how much information they have about the system and message traffic is needed in order to break the system. It is always assumed in determining how secure the system is that any cryptanalist knows complete details of the encryption algorithm, this is especially important, and true, when discussing public-key cryptosystems. A system is considered broken if a cryptanalist given any ciphertext then produce the corresponding plaintext.

The highest level is ciphertext only. In this the cryptanalist only knows certain encrypted messages. It may also be assumed that they also know context, but not content of the message. The second level is known plaintext. In this the cryptanalist knows certain plaintext as well as its corresponding ciphertext. If a system is insecure against this type of attack then it is important to destroy any deciphered messages. The third level is chosen plaintext. In this the cryptanalist knows plaintext-ciphertext pairs of their choosing. It is impoprtant that all public-key cryptosystems be secure against this type of attack since the encryption algorithm and keys are made public any cryptanalist has this information at their disposal.

II.2.3 Authentication

For any communication system it is also necessary for users to know who they are talking to. This problem is known as the authentication problem. Authentication can be obtained in a public-key system if we add the property

P(4) Every element in the cryptogram space C can be decrypted, that is there is an m such that $m = D(c)$ for every $c \in C$. This is equivalent to saying that $E(D(c) = c$ for every c. That is in addition to the mapping being injective it is also surjective.

With the adition of this property user A signs his message to B by forming a message dependent signature $s = D_a(c)$, and then computes $c' = E_b(s)$. User B upon receipt of c' applies D_b to obtain s, and then using the public encryption key for a finds $E_a(s) = E_a(D_a(c)) = c$. B is then satisfied that A sent the message for only A should know D_a and $E_a(D_d(c)) \neq c$ if $d \neq a$.

II.2.4 Joint Encryption Error-Correction

In some enviroments it may be desirable to have both error-correction and encryption capabilities. Certainly these can be implemented seperately, but with the use of algebraically coded encryption it is possible to implement both in one step. By incorporating both steps into one step speed and efficiancy are both increased, since they can be implemented on a single chip. The trade-off encountered is that the security provided by JOEEC is inferior to that provided by singular encryption as is the error-correcting capability.

II.3 FEEDBACK SHIFT REGISTERS

Let k be a positve integer and a, $a_0, a_1, \ldots, a_{k-1}$ be fixed elements in a finite field $\mathbb{F}_q$. A sequence (s_i) of elements from $\mathbb{F}_q$ satisfying

$$s_{n+k} = a_{k-1}s_{n+k-1} + a_{k-2}s_{n+k-2} + \cdots + a_0 s_n + a \quad \text{for } n = 0,1,\ldots$$

is called a linear recurring sequence, with initial values $s_0, s_1, \ldots, s_{k-1}$ in $\mathbb{F}_q$. If $a = 0$ then the sequence is said to be homogeneous. The polynomial

$$f(x) = x^k - a_{k-1}x^{k-1} - a_{k-2}x^{k-2} - \cdots - a_1 x - a_0; \ a_i \in \mathbb{F}_q[x]$$

is called the characteristic polynomial of the sequence. The characteristic polynomial of a linear recurring sequence of least possible order that generates the sequence is said to be the minimal polynomial of the sequence.

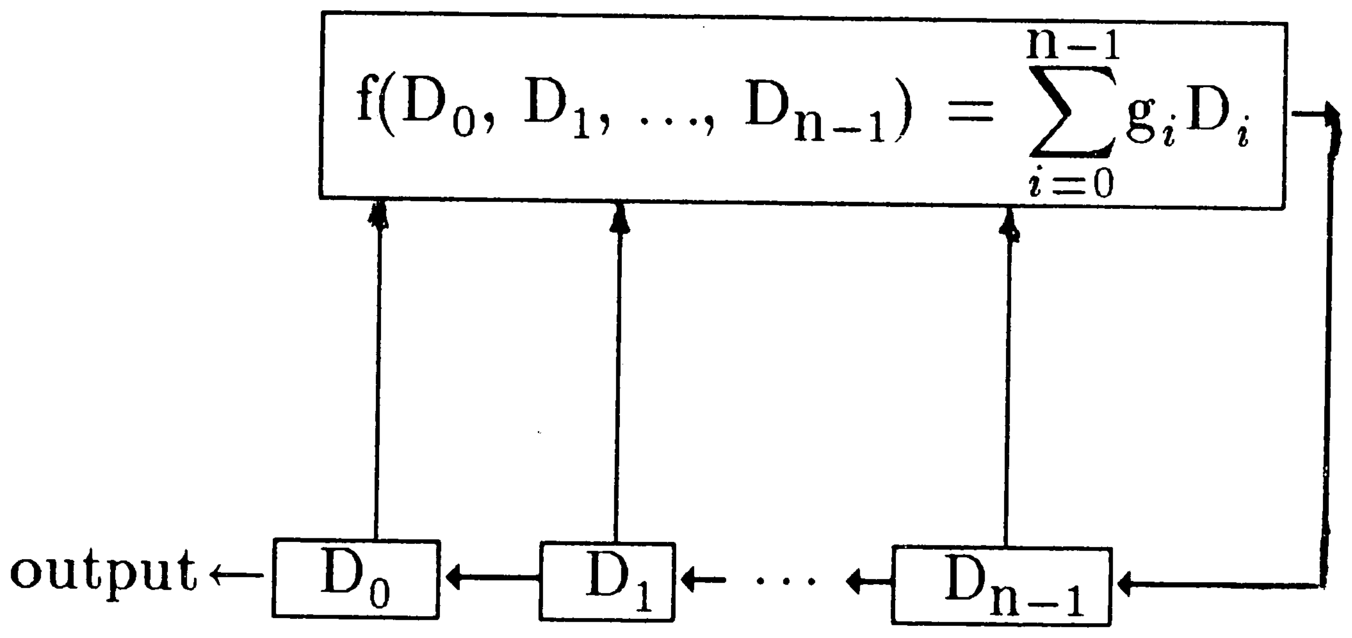

Figure 2.4 Feedback Shift Register System

The sequence is said to be ultimately periodic with period r if there exists positive integers r and n_0 such that $s_{n+r} = s_n$ for every $n \geq n_0$. The smallest number of all possible periods of an ultimately periodic sequence is called the least period of the sequence. An ultimately periodic sequence with least period r is called periodic if $s_{n+r} = s_n$ holds for all integers $n \geq 0$. It turns out that a sufficient condition for a sequnce with characteristic polynomial f(x) to be periodic is that $f(0) \neq 0$. The least period for a sequence with minimal polynomial m(x) is equal to the order of m(x).

The implementation of these sequences is done on a feedback shift register (FSR), for this reason these sequences are also known as FSR sequences. The characteristic polynomial is also known as the connection polynomial for the register (see figure 2.4).

We define the decimation of (s_i) by a factor of k as the sequence (s_{ik}), that is take every k^{th} term of the original sequence starting at s_0. To see that all the decimated sequences are n-stage FSR sequences over $\mathbb{F}_q$ with minimal polynomials of degree n we refer to the following lemmas.

Lemma 2: If the characteristic polynomial g(x) of an FSR sequence (s_i) in $\mathbb{F}_q$ has factorization $g(x) = \prod_{j=1}^{n} (x-\beta_j)$ in its splitting field over $\mathbb{F}_q$ then the decimated sequence (s_{ik}) has the characteristic polynomial $g_k(x) = \prod_{j=1}^{n} (x-\beta_j^k)$.

Lemma 3: If (s_i) is an FSR sequence over $\mathbb{F}_q$ with minimal polynomial and $\gcd(k, M) = 1$ and x^2 does not divide g(x) then (s_{ik}) has the minimal polynomial $g_k(x)$.

For feedback register sequences sometimes it is ideal to be able to efficiently determine the k^{th} element without having to calculate the entire sequence up to that point. Fiduccia [9] has given an algorithm that will do that. Given a shift register sequence of length n, with characteristic polynomial g(z), and initial elements s_i for $i = 0, 1, \ldots, n-1$, if we want to calculate the value of s_k, determine the value of z^k modulo g(z). This will be a polynomial $\Gamma(z) = \sum_{i=0}^{n-1} \gamma_i z^i$. The value of the k^{th} element of the sequence is then determined by $s_k = \sum_{i=0}^{n-1} \gamma_i s_i$. In this manner the k^{th} element of a sequence can be determined in $\mathcal{O}(\mu(k) \cdot \log n)$ arithmetic operations, where $\mu(k)$ is the number of arithmetic operations to multiply two length $k-1$ polynomials. This can be done in $\mathcal{O}(k \cdot \log k \cdot \log n)$ operations over a field that supports fast Fourier transforms.

II.3.1 FEEDBACK SHIFT REGISTER SYNTHESIS

Given a sequence (s_i) it is often desirable to know the shift register of minimum length that will generate that sequence. The Berlekamp-Massey algoriithm [17] is a recursive algorithm that will do so without knowledge of a characteristic polynomial for the sequence. The only knowledge needed is an upper bound for the degree of the polynomial.

Let (s_i) be sequence over $\mathbb{F}_q$, and let $G(z) = \sum_{i=0}^{\infty} s_i z^i$. Define polynomials $g_j(z)$ and $h_j(z)$ over $\mathbb{F}_q$ and integers $m_j \in \mathbb{F}_q$ recursively. Initially let $g_0(z) = 0$, $h_0(z) = z$ and $m_0 = 0$. Define b_j to be the coefficient of $g_j(z)$ $G(Z)$. Proceeding let

$$g_{j+1}(z) = g_j(z) - b_j\, h_j(z),$$

$$h_{j+1}(z) = \begin{cases} b_j^{-1} z\, g_j(z) & b_j \neq 0 \text{ and } m_j \geq 0 \\ z\, h_j(z) & \text{otherwise} \end{cases}$$

$$m_{j+1} = \begin{cases} -m_j & b_j \neq 0 \text{ and } m_j \geq 0 \\ m_j + 1 & \text{otherwise} \end{cases}$$

If the sequence has a minimal polynomial of degree k , then $g_{2k}(z)$ is the reciprical minimal polynomial. If instead the minimal polynomial is of degree $\leq k$, then let $r = \left\lfloor k + \frac{1}{2} - \frac{1}{2} m_{2k} \right\rfloor$, and the minimal polynomial $m(z) = z^r g_{2k}(1/z)$. In either case $m(z)$ depends only on the first 2k terms of the sequence (s_i) so $G(z)$ can be replaced by $G(z) = \sum_{i=0}^{2k-1} s_i z^i$.

Given m sequences $\left(s_i^{(h)}\right)$ we are interested in finding the feedback shift register of minimal length that will generate them. The following algorithm proposed by Feng and Tzeng [8] from their generalized Euclidean algorithm will do that.

Step 1. Let $r_0 = \sum_{h=0}^{m-1} z^{m-1-h} \sum_{i=0}^{n-1} s_i^{(h)} z^{m(n-1-i)}$, $U_0(z) = 1$,

$$b_0^{(h)}(z) = z^{mn+h} \text{ , } V_0^{(h)}(z) = 0, \text{ for all } h = 0, 1, \ldots, m-1; \text{ and } j = 0.$$

Step 2. $j = j+1$.

Step 3. Calculate $r_j(z)$, $p_j(z^m)$ and $q_j^{(h)}(z^m)$ from $r_{j-1}(z)$ and $b_{j-1}^{(h)}(z)$ so that

$$r_j(z) = p_j(z^m)\, r_{j-1}(z) + \sum_{h=0}^{m-1} q_j^{(h)}(z^m)\, b_{j-1}^{(h)}.$$

Let $v_{j-1} = \deg r_{j-1}(z) \bmod m$,

$$b_j^{(v_{j-1})}(z) = r_{j-1}(z) \text{ and } b_j^{(h)}(z) = b_{j-1}^{(h)}(z) \text{ for all } h \neq v_{j-1}.$$

Step 4. Determine $U_j(z)$ from U_{j-1} and $V_{j-1}^{(h)}(z)$ so that

$$U_j(z) = p_j(z)\, U_{j-1}(z) + \sum_{h=0}^{m-1} q_j^{(h)}(z)\, V_{j-1}^{(h)}(z).$$

Let $V_j^{(v_{j-1})}(z) = U_{j-1}(z)$ and $V_j^{(h)}(z) = V_{j-1}^{(h)}(z)$ for all $h \neq v_{j-1}$.

Step 5. If $\deg r_j(z) \geq \deg U_j(z^m)$ then go to step 2, otherwise go to step 6.

Step 6. Let $k = j$, and $\delta U_k(z)$ is the connection polynomial for the shortest length feedback shift register generating the multiple sequences. Where δ is the field element making $U_k(z)$ monic.

If $\deg r_{j-1}(z) \geq \deg U_{j-1}(z^m)$ for all $1 \leq j \leq k$ and $\text{degr}_k(z) < \deg U_k(z^m)$ and $\deg U_k(z^m) \leq g_k^{(h)}$ for all $h = 0, 1, \ldots, m-1$ where $m \cdot g_k^{(h)} = (\deg b_k^{(h)}(z)) - h$ then $\delta U_k(z)$ is the unique shortest length feedback shift register generating the sequences. If however $\deg r_{j-1}(z) \geq \deg U_{j-1}(z^m)$ for all $1 \leq j \leq k$ and $\deg r_k(z) < \deg U_k(z^m)$, $\deg U_k(z^m) = g_k^{(h)} + d^{(h)}$ for all $0 \leq h \leq m-1$, then $U_k(z) = \sum_{h=0}^{m-1} W^{(h)}(z)\, V_k^{(h)}(z)$, where for each h $W^{(h)}(z)$ is any polynomial of degree less than $d^{(h)}$ if $d^{(h)} > 0$, and $W^{(h)}(z) = 0$ otherwise are all the shortest length feedback shift registers that will generate the multiple sequences.

2.3.2 FAMILIES OF SEQUENCES

Let $f(z) \in \mathbb{F}_q[z]$ be monic of positive degree. Let $S(f)$ denote the set of linear recurring sequences in $\mathbb{F}_q$ having $f(z)$ as tehir characteristic polynomial. If $\deg(f) = k$ then $|S(f)| = q^k$, as there are that many choices for the initial states of the sequences generated by f. This set can also be viewed as a vector space of dimension k over $\mathbb{F}_q$ if operations are defined termwise.

If we define $f(x) = x^k - a_{k-1}x^{k-1} - a_{k-2}x^{k-2} - \cdots - a_1x - a_0$, and add the further requirement that $f(0) \neq 0$, then every sequence that has $f(z)$ as its characteristic polynomial will be periodic. If we let $\left(s_i^{(h)}\right)$ for $h = 0, 1, \ldots, q^k - 1$ be the sequences in $S(f)$, then we claim the following

Lemma: If $\left(s_i^{(h)}\right)$ for $h = 0, 1, \ldots, N-1$ are N linearly independent sequences over a finite field $\mathbb{F}_q$ of period M_h and $M = \operatorname{lcm}_h \{m_h\}$ then the decimated sequences $\left(s_{ik}^{(h)}\right)$ for k relatively prime to M are also linearly independent.

Proof: Suppose the decimated sequences were linearly dependent. Since the decimation factor was relatively prime to the period of each of the sequences the decimation ammounts to reordering of the elements of each sequence. Thus if the decimated sequences were linearly dependent, then the originals must have been also. Since this is a contradiction of our original assumption, the decimated sequences must be linearly independent.

Chapter III - Feedback Shift Register Cryptosystems

In this chapter encryption using feedback shift register sequences will be discussed. The system originally proposed by Niederreiter [21] will be presented, as well as a new system resulting from the work of Feng and Tzeng [8] in multiple sequences that is a generalization of the original system.

While these sequences are not directly related to error-correcting codes, their use in the decoding of certain types of codes, namely the Goppa codes makes them relevent for discussion here.

Section one will deal with the system proposed by Nieddereiter. Section two will be a discussion of our proposed generalization of that system using multiple sequences. Section three will be a discussion of the cryptanalysis of this system.

III.1 Niederreiter FSR Cryptosystem

Let q be a power of a prime and g(z) be a monic polyinomial of positive degree n over $\mathbb{F}_q$ such that $g(0) \neq 0$. Let (s_i) be a sequence with characteristic polynomial g(z), period M and initial terms $s_0 = \cdots = s_{n-2} = 0$, and $s_{n-1} = 1$. In this case g(z) will be the minimal polynomial of (s_i).

For two users of the system, A and B to correspond they each need to generate their own public key. This is done as follows: each user picks an integer h such that $1 \leq h < M$ and $\gcd(h, M) = 1$, then decimates the sequence (s_i) by that factor and publishes as their public key the second $2n-1$ terms of the decimated sequence (s_{ih}), namely $s_h, s_{2h}, \ldots, s_{(2n-1)h}$; it is not necessary to publish the first term as it will always be 0.

For user B to send a length n, non-zero message vector $(a_0, \ldots, a_{n-1})$ over $\mathbb{F}_q$ to A, B chooses an integer k such that $1 \leq k < M$ and $(k, M) = 1$. From A's public key, B determines the minimal polynomial of $(s_{ih}) = (t_i)$ and decimates this new sequence by the factor k to obtain a new sequence $(s_{kih}) = (u_i)$. Then B constructs the matrix

$$U = \begin{bmatrix} u_0 & u_1 & \cdots & u_{n-1} \\ u_1 & u_2 & \cdots & u_n \\ \vdots & & \ddots & \vdots \\ u_{n-1} & u_n & \cdots & u_{2n-2} \end{bmatrix}$$

and sends to A the ciphertext pair $s_k, s_{2k}, \cdots, s_{(2n-1)k}, (a_0, a_1, \cdots, a_{n-1})U$.

In order for A to decrypt B's message, A must first determine the minimal polynomial of the sequence $s_k, s_{2k}, \cdots, s_{(2n-1)k}$, and decimate this sequence by the factor h to recover the sequence (u_i). A then generates the matrix U. Since the matrix U is invertible, A inverts U and recovers the original message vector by postmultiplying $(a_0, a_1, \cdots, a_{n-1})U$ with U^{-1}.

Transmission requirements can be further reduced if instead of selecting a different decimation factor k and transmitting along with the ciphertext the initial elements of his sequence, B instead uses the same decimation factor k as the its private key. A then refers to the public key list to find the sequence (s_{ik}). This assumption will be made throughout the remainder of this thesis.

When $n = 1$, the cryptosystem reduces one based on discrete exponentiation, where the security is derived from the difficulty of given x and y in a field finding α such that $x^{\alpha} = y$.

III.2 Multiple Sequence Cryptosystem

As Niederreiter pointed out his system is susceptible to attack by knowledge of certain plaintext-ciphertext pairs. Notably two message vectors $(1, 0, \cdots, 0)$, $(0, 0, \cdots, 0, 1)$ and their ciphertext pairs are sufficient to completely determine the matrix U. While Niederreiter proposes to improve upon the situation by first encrypting the message by using either the knapsack method, two forms of which will be discussed in appendix A, or by using error correcting codes, we propose a method that will improve on the minimum number of plaintext-ciphertext pairs that are necessary to determine the matrix U. This will be done by using more than one sequence generated by a given characteristic polynomial to develop the encryption matrix U. It should be pointed out that after n linearly independent messages have been encrypted with the matrix U, they together with their corresponding ciphertexts will allow for the complete determination of U.

Let $g(z) = \sum_{i=0}^{n} g_i z^i$, with $g_n = 1$, be a publically known polynomial over $\mathbb{F}_q$ with $g(0) \neq 0$. Select a basis β for the vector space S(g), and a subset β' of β, let $X = |\beta'|$. For each element (s^h) of β' find its period M_h, then determine $M = \underset{h}{\text{l.c.m.}}\{M_h\}$. This may either be done privately or publically. To implement as a public-key cryptosystem make the characteristic polynomial g(z) together with the set β' public. As a private key system, keep knowledge of both private. Let $\left(s_i^{(h)}\right)$, $h = 0, 1, \ldots, X-1$, be the sequences of β'.

Each user A selects an integer k_A such that $1 \leq k_A < M$ and $(k_A, M) = 1$, and publishes as their public key the first 2n terms of the sequences $(t_i^{(h)}) = (s_{ik_A}^{(h)})$ for each h. It is important that the decimation factors k_A be kept private.

For user A to send a message to user B, first find the minimal polynomial $g'(z)$ that will generate B's public key; this is best done by using the Feng-Tzeng algorithm [8].

Once the minimal polynomial is obtained, user A then generates the family of sequences $(u_i^{(h)}) = (t_{ik_A}^{(h)})$.

If n = X then B constructs an n × n matrix V, where $v_{i,j} = \left(u_{j \bmod M}^{i}\right)$, for i, j = 0, 1, ..., n−1. Length n, non-zero message vectors **a** are encrypted by computing **a**V. Since the decimated sequences are linearly independent the matrix V is non-singular, and so from B's public key A is able to reconstruct the sequences $\left(u_i^{(h)}\right)$ and thus the matrix V. After finding V^{-1} A can readily recover **a**.

If n < X, let $\rho = \left\lceil \frac{n}{X} \right\rceil$ and then construct X $\rho \times n$ matrices V_p, where the pth block, p = 0, 1, ..., X−1 is given by $v_{i,j} = \left(u_{iX+j(\mathrm{mod} M)}^{(p)}\right)$; $i = 0, 1, \ldots, \rho-1$ and j = 0, 1, ..., n−1. If X∤n then for in V_p, p = 0, 1, ..., $n-\rho X$ let $i = 0, 1, \ldots, \rho-2$, that is let V_p have one fewer row in those matrices.

Length n, non-zero message vectors **a** are encrypted by the following process: first construct the matrix $V = \begin{bmatrix} V_0 & V_1 & \cdots V_{X-1} \end{bmatrix}^T$. If V is non-singular then the encrypted message is **c** = **a**V. If V is singular then let $\mathbf{a}_p = (a_{p\rho}, a_{p\rho+1}, \ldots, a_{p\rho+\rho-1})$, again if X∤n then for p = 0, 1, ..., $n-\rho X$, let $\mathbf{a}_p = (a_{p\rho}, a_{p\rho+1}, \ldots, a_{p\rho+\rho-2})$, and calculate $\mathbf{c}_p = \mathbf{a}_p V_p$. These vectors are then sent as the encrypted message. If X = 1 then the cryptosystem describes the original Niederreiter FSR cryptosystem, and the matrix $V = V_0$ is always non-singular.

In order to decrypt, the recipient looks up the senders public key, generates the minimal polynomial for the multiple sequence, and decimates by their own private factor to generate the multiple sequence $\left(u_i^{(h)}\right)$. From this the recipient then reconstructs the matrix V, if it is invertible, inverts it and recovers the message vector **a**. If V is not invertible, since the rank of the matrices V_p is ρ, each matrix has a right inverse V_p^{-r}, and $\mathbf{a}_p = \mathbf{c}_p V_p^{-r}$, and **a** is completely recoverable.

The major drawback to this generalization is that excpet for the case when $X = 1$ or $X = n$, the matrix V is not in general non-singular and so there is a increase in the data expaansion by a factor of X.

III.2.1 EXAMPLE OF GFSR CRYPTOSYSTEM

Let $g(z) = z^6+z^5+z^4+z^2+z+1$, over $\mathbb{F}_2$. Any sequence (s_i) with characteristic polynomial g(z) then satisfies $s_{n+6} = s_{n+5}+s_{n+4}+s_{n+2}+s_{n+1}+s_n$. Choose as β' the three sequences $\left(s_i^{(h)}\right)$ generated by the following vectors (0, 0, 0, 0, 0, 1), (0, 1, 0, 1, 0, 1) and (1, 1, 0, 0, 1, 0). The sequences are:

$$\left(s_i^{(0)}\right) = 0\ 0\ 0\ 0\ 0\ 1\ 1\ 0\ 1\ 0\ 1\ 1\ \cdots$$

$$\left(s_i^{(1)}\right) = 0\ 1\ 0\ 1\ 0\ 1\ 0\ 1\ 0\ 1\ 0\ 1\ \cdots$$

$$\left(s_i^{(2)}\right) = 1\ 1\ 0\ 0\ 1\ 0\ 1\ 0\ 0\ 1\ 1\ 1\ \cdots$$

here $M_0 = 12$, $M_1 = 6$, $M_2 = 12$, $M = 12$, $X = 3$ and $\rho = 2$.

Choosing $k_A = 5$, A constructs the public key:

$$\left(s_{5i}^{(0)}\right) = 0\ 1\ 1\ 0\ 1\ 0\ 1\ 1\ 0\ 0\ 0\ 0\ \cdots$$

$$\left(s_{5i}^{(1)}\right) = 0\ 1\ 0\ 1\ 0\ 1\ 0\ 1\ 0\ 1\ 0\ 1\ \cdots$$

$$\left(s_{5i}^{(2)}\right) = 1\ 0\ 1\ 0\ 0\ 1\ 1\ 1\ 1\ 1\ 0\ 0\ \cdots$$

After determining the characteristic polynomial for these three sequences, B decimates them by their factor k_B, here taken to be 7 to obtain:

$$\left(s_{35i}^{(0)}\right) = 0\ 1\ 1\ 0\ 1\ 0\ 1\ 1\ 0\ 0\ 0\ 0\ \cdots$$

$$\left(s_{35i}^{(1)}\right) = 0\ 1\ 0\ 1\ 0\ 1\ 0\ 1\ 0\ 1\ 0\ 1\ \cdots$$

$$\left(s_{35i}^{(2)}\right) = 1\ 1\ 1\ 1\ 0\ 0\ 1\ 0\ 1\ 0\ 1\ 0\ \cdots$$

and constructs the three matrices:

$$V_0 = \begin{bmatrix} 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 \end{bmatrix},\ V_1 = \begin{bmatrix} 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 \end{bmatrix},\ V_2 = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 \end{bmatrix}.$$

It is noted that row 1 of V_2 together with the two rows of V_1 are linearly dependent, so the matrix V would be singular. Thus any message vector **a** would have to be broken up into three length two vectors. Take as a sample message **a** = (1, 0, 0, 1, 1, 1). This would be encrypted into:

(0, 1, 1, 0, 1, 0), (1, 0, 1, 0, 1, 0), (0, 1, 1, 1, 0, 0, 1).

Using $V_0^{-r} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}$, $V_1^{-r} = \begin{bmatrix} 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 \end{bmatrix}$, $V_2^{-r} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$, the original message is recoverable. There are other choices for the right inverses.

III.3 Crypatanalysis

There are two ways to attack the FSR cryptosystem. The first is by attempting to derive the decimation factors h and k, and in this way deriving the matrix U. The second is by attempting to directly determine what the elements of the matrix U are.

There are three steps in determining k from the knowledge of both $g_k(z)$ and g(z). The first is to factor the characteristic polynomial g(z) of the original sequence. This needs only to be done once, when the the system is initialized, as it will remain constant. One also needs to factor the characteristic polynomial $g_k(z)$ of the decimated sequence (s_{ik}). The second step is to pair off the roots of $g_k(z)$ with the roots of g(z). There is one correct matching for this, and it can take up to n! trials to obtain it. This step is not necessary if g(z) is irreducible. The final step is inferring the factor k from the pairing of the roots, this is the same as solving discrete logarithms in various extensions of $\mathbb{F}_q$.

In both the original single sequence and the genralized multiple-sequence system the above steps must be performed. Thus in both cases g(z) should be reducible. It is also desirable that as large a value for the period of the sequences be obtained as possible, in order to have as many different decimation factors as possible. A large power of 2 for the period is ideal. In the multiple sequence case the complexity is not adjusted, with the exception that the period will increase as it is the least common multiple of the periods of the basis, as there is only one characteristic polynomial for all the sequences.

To determine the matrix V used for the encryption of the messages directly, as far as the multiple sequence system is concerned, we will present a lower bound on the number of linearly independent message vectors that will be necessary to determine U, it is to be noted that the upper bound on the number of messages needed to determine U is still n, that is any set of n linearly independent messages along with their respective ciphertexts is sufficient to derive the entries for U.

In U there are $\theta = [(\rho-2)X+n][n-\rho X]+[(\rho-1)X+n][X-(n-\rho X)] = X^2[2\rho-1]$ unknowns. Since each plaintext-ciphertext pair generates n equations in the θ unknowns, at least $\frac{\theta}{n}$, linearly independent plaintexts and their corresponding ciphertexts are needed in order to completely determine the entries in the matrix V. To see that in general $\frac{\theta}{n}$ is only necessary but not sufficient consider the 3×3 case with one sequence to determine V. The two vectors (1, 0, 0) and (1, 1, 0) while linearly independent cannot give the entry for u_4 whereas (1, 0, 1) and (1, 1, 0) will. In the case when $X = n$, $\theta = n^2$ and so the optimal security is obtained as n linearly independent messages and their ciphertexts are both necessary and sufficient.

Thus it can be seen that it is possible that in the original system two linearly independent vectors will be able to break the system, but as the number of sequences used to to describe the system increases the minimum number of known plaintext-ciphertext pairs necessary to break the system also increases.

Chapter IV - McEliece-Type Cryptosystems

In this chapter encryption using algebraic codes is presented. Discussion in this chapter will be restricted to those that are related to the scheme originally proposed by McEliece [18]. In addition to McEliece's system the variant of that system proposed by Jordan [11] is discussed.

Attention is then turned to the private key scheme proposed by Rao-Nam [27]. After a detalied ananlysis of these systems is presented the variations of this scheme is presented, as well as the resulting variations that have arisen from these analysis. Following these variations a new characterization formula that can be used to describe all codes in this class (both the McEliece and its variants as well as Rao-Nam and its variants) will be presented. Resulting from this generalization another variant will be discussed.

This is followed with a discussion of the applicability of these codes to joint encryption and error control. An observation on the insuitability of these codes to the authentication problem will be presented.

IV.1 McEliece Cryptosystem

In 1978 McEliece [18] proposed using algebraic error-correcting codes, specifically Goppa codes, to perform public key encryption. The user generates a t-error-correcting Goppa code of length $n = 2^m$, and dimension $k \geq n - tm$, then produces the associated k×n generator matrix G. Also used as part of the private key are two matrices, a k×k non-singular matrix S and an n×n permutation matrix P. The user then constructs and publishes their public encryption key $G' = SGP$.

A message is sent by breaking it into k-bit blocks. For each block m calculate and send the n-bit vector

$$c = mG' + z$$

where z is a randomly chosen vector of length n with weight no more than t, and G' is the public key for the receiver.

To recover m calculate $cP^{-1} = mSG + zP^{-1}$. Since P is a permutation matrix w(z) = $w(zP^{-1})$, so by decoding for the designed Goppa code obtain mS then postmultiply by S^{-1} to recover the original message m.

IV.1.1 Cryptanalysis of McEliece

It remains to analyze the security of the system as presented. An obvious attack would be to randomly choose k of the n cooridinates of c and denote this k-bit vector by c_k. Let G'_k and z_k denote the corresponding k columns from G' and z respectively. We now have $c_k = mG'_k + z_k$, or alternatively if G'_k is invertible we have $(c_k + z_k)(G'_k)^{-1} = m$. If the k components of z_k are zero then $c_k \times (G'_k)^{-1} = m$ and one can recover the message without decoding. The work factor for this attack is calculated as follows. If the error vector has t non-zero components (the maximum possible) the probability of choosing k non-zero cooridinates is $p = \binom{n-t}{k} / \binom{n}{k}$, and on average 1/p choices must be made before succesfully picking k zero cooridinates. The k × k submatrix G'_k must be inverted for each choice of k cooridinates, assuming matrix inversion requires between k^2 and k^3 steps the expected total work factor is as follows:

$$W = \binom{n}{k} / \binom{n-t}{k} \text{ steps.}$$

McEliece suggested using codes with n = 1024 and k = 50, but Adams and Meijer [1] have shown that for n = 1024, t = 37 gives the highest work factor, approximately $2^{84.1}$ (as opposed to $2^{80.7}$ for t = 50). They also point out that with the decreased value of t the value for k increases from 524 to 654 thus reducing the data expansion.

IV.1.2 Jordan Variation

Jordan [11] has proposed a variant of this system. Construct a Goppa polynomial with no linear or repeated factors over the base field, a bijection (not necessarily linear) S on the set of data vectors, and a permutation matrix P, which are then distributed to the intended receiver. The message **m** is encrypted by **c** = S(**m**)GP+**z**; w(**z**) $\leq$ t, where t is the error correcting capability of the Goppa code.

The recipient who knows S, G, and P then recovers **m** by calculating **c**P, decoding for G then c applying S^{-1} to the result.

By using as a private-key system it is possible to reduce the required weight of the error vectors, Jordan suggests as low as t=10, and thus further reduce the data expansion of the system. In analyzing the system, consider the case where S is linear. The proposed work factor to find the k^2 elements of SGP is $k^6/(1-t/n\ )^{k^2}$, where $(1-t/n)^{k^2}$is the probability of finding k^2 equations with no errors in them. If one first attempts to guess the Goppa polynomial, and use that to eliminate **z** then the work factor involved is at least $k^6(2^{mt}/t)$, where $(2^{mt}/t)$ arises as the number of irreducible polynomials with degree t over $\mathbb{F}_{2^m}$. In both cases the factor k^6 arises as the estimated work factor in solving k^2 equations in k^2 unknowns.. For n = 2^8, t = 5, k = 216 the work factor is 10^{25}.

IV.2 Rao-Nam Cryptosystem

In 1986 another variant on the McEliece system was proposed by Rao and Nam [27]. They proposed using low distance linear (n, k) codes with generator matrix G and error vectors **z** that have been specifically chosen with Hamming weight approximately $\frac{n}{2}$. Again let S be an k×k non-singular matrix and P be an n×n permutation matrix. We let $G' = SG$ and encrypt a message **m** by computing **c** = (**m**G$'$+**z**)P. The vector **z** is chosen by one of two methods.

Method 1: **z** is an ATE, a vector of length n with $t(\leq \frac{n}{2})$ adjacent ones, and the remaining $n-t$ cooridinates zero. It is important that the ATE chosen is not a codeword. For a non-cyclic code there are $n-t+1$ ATE's, for a cyclic code there are n ATE's.

Method 2: Use a predetermined set of vectors such as the syndrome-error table. Choose one vector for each possible error pattern with weight as close to $\frac{n}{2}$ as possible. Each error pattern will have a distinct syndrome and there are 2^{n-k} possible error patterns.

Regardless of the method chosen the keys S, G and the choices for **z** in method 2 are kept secret, as this system is implemented as a private-key cryptosystem.

In order to decrypt a message **c** the receiver must first calculate $cP^{-1} = mG' + z$ then by using the parity check matrix for the code calculate post mulitiply by H^T to obtain the syndrome zH^T and identify the error pattern. Recover **mS** by correcting for the appropriate error pattern, and postmultiply by S^{-1} to obtain the original message **m** .

Rao and Nam show that for method 1 there are at least $(n - \lfloor \frac{n}{t} \rfloor - 1)!$ possible permutation matrices P that transform the ATE's into non-ATE's where $2 \leq t \leq \frac{n}{2}$, where n is the length of the ATE, and t is the number of adjacent ones. Thus for $t=\lfloor \frac{n}{2} \rfloor$ there are at least $(n-3)!$ choices for P. Over GF(2) there are $N_S = \prod_{i=0}^{k-1}(2^k - 2^i) > 2^{k^2-k}$ possible non-singular matrices S. Due to the large number of matrices involved an attack by trying all possible matrices S, G, and P is not likely to work.

IV.2.1 Cryptanalysis of Rao-Nam System

The attack presented for the McEliece system will not be benificial since the value for t is approximately $\frac{n}{2}$. Instead the following attack is proposed.

Encipher a message $\mathbf{m}$ twice, that is, let $\mathbf{c}_j = \mathbf{m}G''+\mathbf{z}_jP$ and let $\mathbf{c}_k = \mathbf{m}G''+\mathbf{z}_kP$ then we have $\mathbf{c}_j-\mathbf{c}_k = (\mathbf{z}_j-\mathbf{z}_k)P$. Repeat for all pairs of $\mathbf{z}$'s, of which there are $\binom{N}{2}$ where $N = \frac{n}{2}$ for method 1, and $N \geq n$ for method 2. If we denote the i^{th} row of G'' by $[g'']_i$, then we see that $[g''] = c_1-c_2-(z_1-z_2)P$. This must be done for each row i, then the solution must be verified for correctness. Since the rows cannot be verified independently all rows must be calculated first, involving a work factor of $W \geq \frac{1}{2}\left(\frac{N^2}{2}\right)^k$.

Struik and van Tilburg [29] proposed an attack on method 2 of encipherment. An error pattern $\mathbf{z}$ is selected from $\mathcal{Z} = \{\mathbf{z}^{(j)}\}$ the set of N distinct error patterns. Denote by $\mathcal{Z}^P$ the set of these error patterns postmultiplied by the permutation matrix P. Let $\mathcal{Z}_\Delta = \{\mathbf{z}^{(i,j)} = \mathbf{z}^{(i)}-\mathbf{z}^{(j)}\}$, and $\mathcal{Z}_\Delta^P$ as the elements of $\mathcal{Z}_\Delta$ permuted by P. Denote by $\hat{\mathbf{z}}$ a guessed error pattern, and let $\tilde{\mathbf{z}}^{(i,j)} = \hat{\mathbf{z}}^{(i)}-\mathbf{z}^{(j)}$. For any message $\mathbf{m}$ there are N possible encipherments, $\mathbf{c}^{(j)} = \mathbf{m}SG + \mathbf{z}^{(j)}P$. Denote the set of posssible encipherments by $\mathcal{E}$. Let $\mathbf{u}_i$ be a unit vector and let $\mathbf{m}_i = \mathbf{m} + \mathbf{u}_i$. Denote the possible encipherments of $\mathbf{m}_i$ by $\mathcal{E}_i$.

To attack the Rao-Nam scheme first choose an arbitrary message m and obtain the N possible cryptograms. With these cryptograms construct the directed graph $\Gamma = (\mathcal{E}, \mathcal{Z}_\Delta^P)$. The vertices being the $\mathbf{c}^{(i)}$ and the edges the $\mathbf{z}^{(i,j)}$, being derived from the fact that

$$\mathbf{c}^{(i)}- \mathbf{c}^{(j)} = (\mathbf{m}SG+\mathbf{z}^{(i)}P)-(\mathbf{m}SG+\mathbf{z}^{(j)}P) = \mathbf{z}^{(i,j)}P.$$

Then construct the automorphism group Aut(Γ), the permutations on $\mathcal{E}$ which leave the labels for the edges invariant.

Now, for each $1 \leq i \leq k$ repeat for $\mathbf{m}_i = \mathbf{m} + \mathbf{u}_i$, enciphering for all of the N possible cryptograms, and constructing the graph $\Gamma_i = (\mathcal{E}_i, \mathcal{Z}_\Delta^P)$. Select an arbitrary $\Phi \in$ Aut(Γ). This mapping induces a mapping on Γ_i which will syncronize the elements of $\mathcal{E}_i$ with those of $\mathcal{E}$. From this syncronization calculate $\mathbf{c}_i^{(1)}-\mathbf{c}^{(1)}= \mathbf{e}_i + \widetilde{\mathbf{z}}^{(1,1)}P$, where $\mathbf{e}_i$ is the i^{th} row of the matrix SG. The probablity of this being the correct row is $|\text{Aut}(\Gamma)|^{-1}$ since there is exactly one automorphism Φ such that $\tilde{\mathbf{z}}^{(1,1)}= \mathbf{0}$. Since the correctness of each row cannot be independently verified from the other rows the cryptanalyst will have to construct on average $|\mathrm{Aut}(\Gamma)|^k$ encipher matrices before correctly approximating SG. After approximating SG the analyst must also find a matrix D = HP and then construct the syndrome error table. If the solution is not correct then a new Φ is chosen and new matrix D is generated.

Struik and van Tilburg have also shown that if method 1 is used- then $|\mathrm{Aut}(\ \Gamma)| = 1$, namely $\Phi(z) = z$, for any non-cyclic code. For any cyclic code $|\mathrm{Aut}(\Gamma)| = 2$, the mappings being $\Phi(z) = z$, and $\Phi(z) = z + 1$. Thus method 1 of choosing the error vector z is not appropriate for encryption as it is insecure against a chosen-plaintext attack.

Rao has since observed [26] that while the system is insecure against chosen-plaintext attack for practical codes the number of encipherments necessary to generate the set $\mathcal{E}$ is large. For example he proposes the (72, 64, 4) Hamming Code encrypted with method 2; here there are 2^8 possible error vectors, since the work factor projected to obtain all distinct c_i is $N*\ln(N)$ and is to be repeated for k unit vectors. Thus $\mathcal{O}(kN*\ln(N))$ plaintext-ciphertext pairs are needed, and for realistic codes the attack time is not feasible. While the scheme is succeptible to chosen-plaintext attacks it is still relatively secure due to the time required to break the system. Encryption under method 1, though is still easily broken and thus should not be consisdered further.

IV.2.2 Variations of the Rao-Nam Scheme

Rising from the Struik attack, adaptations have been made in the Rao-Nam scheme. Struik and van Tillburg [29] proposed to modify the scheme by replacing the k×k non-singular matrix S by a non-linear injective function operating on both the message m and on the error vector z, namely having $c = f(m,z)GP + z$, where f is chosen so that $\forall z$,

$\forall$m $f^{-1}(f(m, z), z) = m$. The decoding algorithm is the same as previously with the exception that instead of taking S^{-1} one must now take f^{-1}. This scheme will defeat the attack of Struik if one chooses a function f that will not allow the unit vectors to be able to estimate the rows of the matrix GP.

Denny and Rao [6] have also propsed an adaptation of the Rao-Nam ~~s~~cheme by using non-linear codes instead of a linear code as the basic block of the system. Let **m** be an k-bit message vector, and S an k×k non-singular matrix, and let $\mathbf{m}_1 = \mathbf{m}S$. The encryption, denoted by G(**m**), using Preperata's class of non-linear codes is done as follows. Use the first 2^{m-1} bits of $\mathbf{m}_1$ to be encoded using B. The next bit is used for the binary index *i*, followed by $2^{m-1}-2m$ bits encoded under C. The last $m-1$ bits will give the polynomial p(x), by $p(x) = 0$ if all the bits are one, and $p(x) = x^v$ where v is the decimal equivalent of the bits if the digits are not all ones. Compute **w** as described in chapter two to give $m_2 = G(m_1)$.

Choose a random error vector z of the form $\mathbf{z} = \mathbf{v} + \mathbf{y}$ where v is an error in the linear code and $w(y) \leq 2$, and compute $m_3 = (m_2+z)P$ for P a permutation matrix.

To decrypt first find $\mathbf{m}_3 P^T = \mathbf{m}_2+z$ and determine the syndrome described in chapter two. After finding the error positions use a look-up table to discover z, and add to reveal m_2. Recalculate σ_0 to give p(x). Adding p(x) to the first 2^{m-1} bits yields the encoded vector in B. This value and the knowledge of *i* allows for the recovery of the encoded vector in C. The decoding of these vectors will give **m**S and the message vector **m** $= (mS)S^{-1}$.

The work factors involved in this scheme are as follows. For encryption, two matrix multiplications are needed, of orders k and n. Also needed are the encoding of two BCH codes of order 2^{m-1}, as well as assorted vector additions. To decrypt again two matrix multiplications are needed, syndromes must be calculated as well as the corresponding error positions for the non-linear code, and σ_0 must be calculated a second time. Finally two 2^{m-1} BCH codes must be decoded. Thus the encryption by this method is more complicated than the encryption by any of the previously mentioned methods; however this encryption method stops the Struik attack at the point of using unit vectors to find the rows of SNP, where here N stands for the encryption using a suitable non-linear code. The encryptions from the unit vectors will not give proper approximations to the matrix elements due to the non-linearity of the code.

IV.3 Characterization of McEliece Type Cryptosystems

It has been observed previously that the Rao-Nam scheme is a derivative of the McEliece scheme. What will now be presented is a characterization equation for encryption that can be used to describe both general types of systems as well as their variants depending upon the context in which the code is implemented.

Let G be a generator matrix for a t-error correcting linear (n,k) code $\mathbb{C}$. Let S_1 be an $k \times k$ non-singular matrix, P_1, and P_2 be $n \times n$ permutation matrices, and S_2 be an $n \times n$ non-singular matrix. To encrypt an k-bit block m calculate

$$c = (mS_1GP_1+z)P_2S_2.$$

The multiplication of the error-vector z by P_2S_2 will alter the weight of z thus careful attention must be paid to both S_2 and the error-vectors z that are used.

If the system is to be implemented as a public-key cryptosystem then the public key becomes $[S_1GP, P_2S_2]$, where $P = P_1P_2$ which is also a permutation matrix. However, by making P_2S_2 public any cryptoanalyst will also know $(P_2S_2)^{-1}$ and thus can calculate

$$cS_2^{-1}P_2^{-1} = mS_1GP_1P_2^{-1} + z = mS_1GP + z.$$

When G describes a Goppa code the system reduces to the McEliece public-key cryptosystem.

By changing S_1 to be an automorphism of length k vectors and adding the requirement that G describe a Goppa code with no repeated factors the encryption process becomes that of the Jordan variant.

To implement as the Rao-Nam private-key cryptosystem, take $P_1 = P_2 = I_n$, S_2 to be a permutation matrix, and select z from the syndrome table. By additionally making S_1 a non-linear operator we get the Struik and van Tilburg variant. By having G describe the encryption process for a non-linear code then we are able to describe the Rao-Denny variant.

It remains to discuss the approximate key size for this type of encryption system. Using the (72, 64, 4) Hamming Code as mentioned by Rao [26] we need one 64×64 matrix, one 72×72 matrix and one 64×72 matrix for a total of approximately $2^{4.6}$K bits. It should be noted that for the variants mentioned above the key size will increase due to the added complexity needed to describe the non-linear operations, as well as thew non-linear codes. There is still a marked decrease in the key-size for the private-key variations as oppposed to the original scheme described by McEliece.

IV.4 JOINT ENCRYPTION ERROR CORRECTION

As was previously mentioned algebraic codes lend themselves to the notion of joint encryption and error correction (JOEEC). If we are communicating over a noisy channel, we want to also provide the ability to correct errors, and JOEEC is a fast method of doing this as well as supplying some security.

The idea of doing this with algebraic codes originates with Rao [25]. He suggests that for public key encryption, using McElice's scheme that using Reed-Solomon codes over $\mathbb{F}_{2^b}$ with distance $d \geq 6$ be used instead of Goppa codes, and instead of introducing a random error vector z introduce an error vector that is a single or double byte error, this will allow detection/correction of most errors due to noise.

If implementing as a private key system, codes simpler than the Reed-Solomon codes needed for the public key situation may be used. Rao suggests that even Hamming codes with distance three or four can be used.

In either case the presence of noise on the communication channel may help provide additional security, proivided the error vector originally used, when taken in conjunction with the errors that will be provided by the channel do not overload the error correcting capability of the code in question. On a particularly noisy channel it may be wise to use the scheme originally describe by McEliece, and instead of adding error vectors of weight t, add error vectors of weight $\frac{t}{2}$.

Recently Park and Tzeng [22] have developed a conctonated scheme that does not necessitate the addition of noise to provide security, thus allowing the full error-correcting capabilities of the codes used for error correction purposes. A scheme such as this would be highly desirable if JOEEC capability is needed.

IV.5 AUTHENTICATION

In the above mentioned encryption schemes the encryption process is an injective function E: $\mathbb{F}_q^k \rightarrow \mathcal{C} \subseteq \mathbb{F}_q^n$, and in most cases $\mathcal{C} \neq \mathbb{F}_q^n$. In order to implement authentication it must be that for any $\mathbf{x} \in \mathbb{F}_q^n$, $E(D(\mathbf{x})) = \mathbf{x}$. But, unless for every vector $\mathbf{x}$, $w(\mathbf{x}-\mathbf{c}) \leq t$ for some codeword $\mathbf{c}$, then the decryption algorithm will produce an error message and decryption process will not supply the message sent. Even if we do have the above desired property then when the encryption algorithm is applied, if $\mathbf{z} \neq \mathbf{x}-\mathbf{c}$, then the resulting encryption will produce a result different from $\mathbf{x}$. Thus for algebraic coded encryption schemes of this type authentication is not possible.

CHAPTER V - NIEDERREITER KNAPSACK SYSTEM

The general knapsack problem may be described as follows: let S be a set of distinct positive integers, given a sum s, find a subset $T \subseteq S$ such that $\sum_{t \in T} t = s$. It may be that no such subset exists, or there may be more than one such solution. As this problem is NP-complete, there have been many attempts to develop cryptosystems based on the knapsack problem, for example the systems of Merkle-Hellman [19], and Chor-Rivest [5], most of which have since been broken. In this chapter the knapsack system proposed by Niederreiter [20] will be discussed. Section one is a description of the knapsack cryptosystem, section two is the cryptanalysis of the same system.

V.1 Niederreiter Knapsack System

The following knapsack encryption method was proposed by Niederreiter [20] in 1986. Let C be a t-error correcting linear (n, k) code over $\mathbb{F}_q$, with parity-check matrix H, an $(n-k)\times n$ matrix over $\mathbb{F}q$ with rank $n-k$. Note that C consists exactly of those length n vectors $\mathbf{c}$ over $\mathbb{F}_q$ such that $H\mathbf{c}^T = \mathbf{0}$. Consider the following

Lemma. If $H\mathbf{u}^T = H\mathbf{v}^T$ for some $\mathbf{u}, \mathbf{v} \in \mathbb{F}_q^n$ and $w(\mathbf{u}), w(\mathbf{v}) \leq t$ then $\mathbf{u} = \mathbf{v}$.

Let $X = \{\mathbf{m} | \, w(\mathbf{m}) \leq t\}$, and $f\colon X \subseteq \mathbb{F}_q^n \to \mathbb{F}_q^{n-k}$ be given by $f(\mathbf{m}) = H\mathbf{m}^T$ then by the above lemma, f is an injection from X into $\mathbb{F}_q^{n-k}$.

A private-key cryptosystem can be obtained from this mapping by using the parity-check matrices as the keys, and vectors of weight no more than t as messages. To encipher $\mathbf{m}$ let $\mathbf{c} = H\mathbf{m}^T$. To recover the message use the syndrome $\mathbf{c}$ to decode according to the code C. Note that $\mathbf{m}$ is an error vector for the code word $\mathbf{0}$.

To use as a public-key cryptosystem we need to scramble H. Following the example presented by the McEliece type cryptosystems let S be an $(n-k)\times(n-k)$ non-singular matrix, and P be an n×n permuted diagonal matrix, both over $\mathbb{F}_q$. Let $K = SHP$ be the public-key, keeping S, H, and P private. Encrypting a message $\mathbf{m} \in \mathbb{F}_q^n$ with $w(m) \leq t$ is done by computing $\mathbf{c} = Km^T$. To decrypt the received vector first compute $M^{-1}\mathbf{c} = HPm^T = H(mP^T)^T$. Since $w(mP^T) \leq t$ obtain mP^T from the sysndrome as with the private-key cryptosystem. From mP^T recover $\mathbf{m}$ by postmultiplying it by P.

For the case where $q = 2$ this is a variant of the classical knapsack system, since the ciphertext is the sum of at most t columns of K and determining m is the problem of deciding which columns they were. For the general q, the ciphertext is a weighted sum of at most t columns from K and decrypting is equivalent to determining which columns were chosen and with what weight.

To determine the best types of codes to implement this system under it is desirable to have a large error-correcting capability since we have only $\sum_{i=1}^{t}\binom{n}{i}$ possible message vectors. As second property that is desirable is to have a code C that can be efficiently decoded so that decryption will run faster. It is important not to have $n-k$ be to small for if so the ciphertexts will be of short length, and thus the system will be easier to break. Niederrieter gives examples of two codes, the first a binary concatenated code (104, 24) code capable of correcting $t = 15$ errors, with a public keysize of 8320 bits. The second is an (30, 12, 19) Reed-Solomon code over $\mathbb{F}_{31}$ capable of correcting up to $t = 9$ errors, with a keysize of 2700 bits.

V.2 Cryptanalysis

To be able to defeat this system there are two possible attacks. In the first given by Brickell and Odlyzko, given ciphertext $\mathbf{c}$ pick a submatrix K' consisting of $n-k$ columns from K. This matrix is non-singular since H was of rank $n-k$ and both S and P were. Now compute $\mathbf{c}' = (\mathrm{K}')^{-1}\mathbf{c}$. If the t columns added to form **c** are in K$'$ then $\mathbf{c}'$ is the encrypted message, and $\mathrm{K}\mathbf{c}' = \mathbf{c}$. To find the message **m**: when a column is not included in K$'$ the corresponding cooridinate is zero. If a column is included in K$'$ then the corresponding cooridinate is the value of the same cooridinate from $\mathbf{c}'$. The probability of this occuring is $\rho = \binom{\text{n-k}}{\text{t}}/\binom{\text{n}}{\text{t}}$. We will have to repeat this procedure, and calculate the inverse of an $(\mathrm{n}-\mathrm{k})\times(\mathrm{n}-\mathrm{k})$ matrix on average $1/\rho$ times before success. For the first code suggested $1/\rho = 72$, and for the second $1/\rho = 295$.

For the second attack we will use the previous result that the matrix H invokes an injective mapping on vectors of weight t or less, thus there is a left inverse for this mapping, that is there exists an $\mathrm{n}\times(\mathrm{n}-\mathrm{k})$ matrix H^{-l} such that for all vectors $\mathbf{m} \in \mathbb{F}_{\mathrm{q}}^{\mathrm{m}}$ with $\mathrm{w}(\mathbf{m}) \leq \mathrm{t}$ $\mathrm{H}^{-l}\mathrm{H}\mathbf{m}^{\mathrm{T}} = \mathbf{m}^{\mathrm{T}}$. Since we know that the matrix H has a left inverse and that S and P are non-singular there a left inverse for the public-key K, K^{-l} so that if $\mathrm{K}\mathrm{m}^{\mathrm{T}} = \mathrm{c}$ then $\mathrm{m}^{\mathrm{T}} = \mathrm{K}^{-l}\mathrm{c}$, if $\mathrm{w}(\mathbf{m}) \leq \mathrm{t}$. In order to deteremine K^{-l} we need to solve $\mathbf{u}_i = \mathrm{K}^{-l}\mathbf{k}_i$, where $\mathbf{u}_i$ is the unit vector of length $\mathrm{n}-\mathrm{k}$ with the one in the i^{th} position, and $\mathbf{k}_i$ is the i^{th} column of the public matrix K. Each equation will give n equations in the $\mathrm{n}(\mathrm{n}-\mathrm{k})$ unknowns of K^{-l} thus solving $\mathrm{n}-\mathrm{k}$ of these sets of equations, for any $1 \leq \mathrm{i} \leq \mathrm{n}$ will give $\mathrm{n}(\mathrm{n}-\mathrm{k})$ equations in $\mathrm{n}(\mathrm{n}-\mathrm{k})$ unknowns, and thus we will be able to solve for K^{-l}.

CHAPTER VI - SUMMARY

Three types of cryptosystems that require concepts taken from algebraic coding for their implementation have been discussed.

The first making use of feedback shift registers was originally suggested by Niederreiter using a single FSR sequence. For this system a generalization has been proposed making use of multiple sequences generated by a single register. When only one sequence is used the generalization describes the original system.

The second type are variations of the system originally suggested by McEliece for public key encryption, for which a private key variant was recommended by Rao-Nam. These systems are shown to be variants of the original.

The third type is a knapsack system proposed by Niederreiter. This system has been shown to be insecure and so should not be considered for implementation.

The strengths of the McEliec type cryptosystems are their ability to be used in enviroments where JOEEC is desirable. Their major drawback however is the large key-size, on the order of thousands of bits per user, needed to implement these systems.

The GFSR cryptosystems alone are unable to support JOEEC. One major drawback to this class of cryptosystems is the need for changing the public key, or private decimation factors after a relatively small numbner of messages have been sent. This is due to the linear nature of the encryption process and its vulnerability to linear algebraic type atttacks. A second major drawback is that not all matrices generated are non-singular and thus require a significant data expansion factor.

REFERENCES

[1] Adams, C.M. and Meijer, H. "Security Related Comments Regarding McEliece's Public- Key Cryptosystem," Advances in Cryptology-Crypto'87, pp. 224-228.

[2] Beker, H.J. and Piper, F.C. "Communications Security - A Survey of Cryptography," IEE Proceedings vol. 129, pt A, no. 6 Aug 1982, pp. 357-376.

[3] Berlekamp, E., McEliece, R.J. and van Tilborg, H.C.A. "On the Inherent Intractability of Certain Coding Problems," IEEE Transactions on Information Theory, vol. IT-24 no. 3 May 1978, pp. 384-386.

[4] Brickell, E.F. and Odlyzko, A.M."Cryptanalysis: A Survey of Recent Results," Proceedings IEEE vol. 76 no.5 May 1988, pp. 578-593.

[5] Chor, B. and Rivest, R.L. "A Knapsack-Type Cryptosystem Based on Arithmetic in Finite Fields," IEEE Transactions on Information Theory, vol. IT-34 no, 5 September 1988, pp. 901-909.

[6] Denny, W.F. and Rao T.R.N. "Algebraic Encryptions Using Non-Linear Codes," 1988 International Symposium on Information Theory, Kobe City, Japan, June 1988.

[7] Diffie, W. and Hellman, M.E. "New Directions in Cryptography," IEEE Transactions on Information Theory vol. IT-22 no.6 November 1966, pp. 644-654.

[8] Feng, G.L. and Tzeng, K.K. "A Generalized Euclidean Algorithm for Multisequence Shift- Register Synthesis," IEEE Transactions on Information Theory, vol. IT-35, May 1989.

[9] Fiduccia, C.M. "An Efficient Formula for Linear Recurrences," SIAM J. Computation, vol. 14, no.1, Feb. 1985, pp. 106-112.

[10] Golomb, S.W. *Shift Register Sequences.* Holden-Day, San Francisco, California; 1967.

[11] Jordan, J.P. "A Variant of A public-Key Cryptosystem Based on Goppa Codes," Sigact News vol. 15 no.1 1983, pp. 61-66.

[12] Lidl, R. and Niederreiter, H. *Finite Fields, Encyclopedia of Mathematics and its*

Applications vol. 20. Addison-Wesley, Reading, Massachusettes; 1983.

[13] Lidl, R. and Niederreiter, H. *Introduction To Finite Fields and Their Applications* Cambridge University Press, New York, New York; 1986.

[14] Lin, S. and Costello, D. *Error-Control Coding: Fundamentals and Applications.* Prentice- Hall, Englewood Cliffs, New Jersey; 1983.

[15] MacWilliams, F.J. and Sloane, N.J.A. *Theory of Error-Correcting Codes.* North-Holland, New York, New York; 1977.

[16] Massey, J.L. "Shift Register Synthesis and BCH Decoding," IEEE Trans. on Information Theory, IT-15, no.1, Jan 1969, pp. 122-127.

[17] McEliece, R.J. *Theory of Information and Coding, Encyclopedia of Mathematics and its Applications vol. 3.* Addison-Wesley, Reading Massachusettes; 1977.

[18] McEliece, R.J. "A Public-Key Cryptosystem Based on Algebraic Coding Theory," DSN Progress Report 42-44 January-Feburary 1978, pp. 114-116.

[19] Merkle, R.C. and Hellman, M.E. "Hiding Information and Signatures in Trapdoor Knapsacks," IEEE Transactions on Information Theory, IT-24, no. 5, September 1978, pp. 525-530.

[20] Niederreiter, H. "Knapsack-Type Cryptosystems and Algebraic Coding Theory," Problems of Control and Information Theory vol. 15 no. 2, 1986, pp. 159-166.

[21] Niederreiter, H. "A Public-Key Cryptosystem Based on Shift Register Sequences," Advances in Cryptology-Eurocrypt '85, pp. 35-39.

[22] Park, C.S. and Tzeng, K.K. "Error-Control Private-Key Cryptosystem Based on a Concatanated Coding Scheme," private communication 1989.

[23] Patterson, W. *Mathematical Cryptology for Computer Scientists and Mathematicians.* Rowan and Littlefield, Totowa, New Jersey; 1987.

[24] Preparata, F.P. "A Class of Optimum Nonlinear Double Error Correcting Codes," Inf. and Control, vol. 13, pp. 378-400.

[25] Rao, T. R. N. "Joint Encryption and Error Correction Schemes," Proceedings 11^{th} International Symposium on Computer Architecture, Ann-Arbor Michigan, May 1984, pp. 240-241.

[26] Rao, T.R.N. "On Struik-van Tilburg Cryptanalysis of Rao-Nam Scheme," Advances in Cryptology-Crypto '87, pp. 458-460.

[27] Rao, T.R.N. and Nam, K. "Private-Key Algebraic-Coded Cryptosystems," Advances in Cryptology-Crypto '86, pp. 35-48.

[28] Smeets, B. "A Comment on Niederreiter's Public Key Cryptosystem," Advances in Cryptology-Eurocrypt '85, pp. 40-41.

[29] Struik, R. and van Tilburg, J. "The Rao-Nam Scheme is Insecure against a Chosen Plaintext Attack," ptt dr neher laboratories publication 87 DNL/56 September 1987.

[30] van Lint, J.H. *Introduction to Coding Theory.* Springer-Verlag, New York, New York; 1982.

[31] van Tilborg, H.C.A. *An Introduction to Cryptology.* Kluwer Academic Publishers, Norwell, Massachusettes; 1988.

VITA

Andrew C. Duke was born on November 16, 1964 in Batavia, New York. From 1982 until 1986 he attended Union College in Schenectady, New York where he received his Bachelor of Science with honors in mathematics in June 1986. Since 1986 he has been a graduate student at Lehigh University in Bethlehem, Pennsylvania, first in the Department of Mathematics and later in the Department of Computer Science and Elictrical Engineering where he has been working on his Master of Science degree in electrical engineering under Professor K.K. Tzeng. He has served as a research assistant for a project supported by the National Science Foundation as well as a teaching assistant. His primary research areas are algebraic coding theory and cryptology.