# A GENERALIZED LINEAR COMPLEXITY

by

MARK STEVEN STAMP, B.S., M.S.

A DISSERTATION

IN

MATHEMATICS

Submitted to the Graduate Faculty
of Texas Tech University in
Partial Fulfillment of
the Requirements for
the Degree of

DOCTOR OF PHILOSOPHY

Approved

May, 1992

## ACKNOWLEDGMENTS

During my years as a graduate student at Tech, I have been privileged to work on many interesting projects with my advisor, Clyde F. Martin. Dr. Martin never ceases to amaze me with his view of the mathematical "big picture."

The other members of my committee, Frits H. Ruymgaart, Linda J. S. Allen, Lance D. Drager, and Minerva Cordero-Vourtsanis were extremely helpful in the preparation of this dissertation. I would especially like to thank Drs. Drager and Allen for many suggestions which significantly improved the exposition. My thanks also goes to Dr. A. H. Chan whose well-timed comments on the paper [77] (which became a major part of Chapter III and the heart of Chapter IV) were critical to the success of this project.

My lovely wife, 杜 倩 倩, deserves special thanks for cheerfully tolerating the long years required to attain this goal. My parents, W. Fred Stamp and Marilyn J. Stamp, deserve credit for any successes I have had in the past or will have in the future.

This dissertation is dedicated to my father, W. Fred Stamp, who recently lost his battle with cancer. His humor and his uncommon "horse sense" will be missed by all who knew him.

CONTENTS

## ABSTRACT

Certain cryptographic applications require pseudo-random sequences which are "unpredictable," in the sense that recovering the sequence from a short captured segment is computationally infeasible. Such sequences are said to be cryptographically strong. Due to the Berlekamp-Massey algorithm, a cryptographically strong sequence must have a high linear complexity, where the linear complexity of a sequence $s$ is the minimum number of stages in a linear feedback shift register capable of generating $s$. However, trivial examples exist which show that a high linear complexity does not insure that a sequence is cryptographically strong.

In this thesis a generalized linear complexity—the $k$-complexity—is proposed and analyzed. The $k$-complexity of $s$ is defined to be the smallest linear complexity that can be obtained by altering any $k$ or fewer elements of $s$. The $k$-complexity can be interpreted as a "strong" measure of the complexity of a sequence, or as a worst-case measure of the linear complexity when $k$ or fewer errors occur.

It is shown that the $k$-complexity gives more information on the cryptographic strength of a sequence than other previously suggested methods. An efficient algorithm for finding the $k$-complexity in the special case where $s$ is a periodic binary sequence with period length $2^n$ is given. This algorithm generalizes a linear complexity algorithm of Games and Chan. The computational complexity of the general case is also considered.

The $k$-complexities of a particular class of binary sequences—the de Bruijn sequences—are analyzed and several computational results are given. In addition, a new class of binary sequences which appear to have good $k$-complexity properties is presented.

# TABLES

# FIGURES

## ALGORITHMS

# CHAPTER I
# INTRODUCTION

The linear complexity of a sequence $s$ is defined to be the minimum number of stages in a linear feedback shift register capable of generating $s$. If $s$ is periodic, its linear complexity is simply the "length" of the shortest linear recurrence that generates $s$. In a particular cryptographic application, pseudo-random sequences are required which are "unpredictable," in the sense that recovering more of the sequence from a short captured segment must be computationally infeasible. Such sequences are said to be cryptographically strong.

The Berlekamp-Massey algorithm gives an efficient method for computing the linear complexity of a periodic sequence. It follows that a cryptographically strong sequence must have a high linear complexity. However, trivial examples exist which show that a high linear complexity does not insure cryptographic strength. Various additional tests for determining cryptographically strong pseudo-random sequences have been suggested. In this paper we propose and analyze a new measure of the complexity of periodic sequences which has application to the security of a particular type of cryptosystem.

The next chapter contains a brief introduction to the field of cryptology—including several historical anecdotes—along with motivation for the problem under consideration. Chapter III gives the necessary background on the theory of linear feedback shift registers, including a discussion of linear complexity and the so-called linear complexity profile. We then define the $k$-complexity of a sequence $s$ to be the smallest linear complexity that can be obtained when any $k$ or fewer elements of $s$ are altered. The $k$-complexity can be interpreted as a "strong" measure of the complexity of a periodic sequence or as a worst-case measure of the linear complexity when $k$ or fewer errors occur. It is shown that in certain cases the $k$-complexity gives more information on the cryptographic strength of a sequence than the linear complexity profile.

In Chapter IV, we discuss algorithms for computing the linear complexity and we develop an efficient algorithm for computing the $k$-complexity in a particular special case. The standard algorithm for finding the linear complexity is the Berlekamp-Massey algorithm, but in the case where $s$ is a binary sequence

with period length $2^n$, the Chan-Games algorithm is more efficient. We give an entirely new proof of the validity of this latter algorithm. More significantly, we generalize the Chan-Games algorithm to efficiently compute the $k$-complexity of binary sequences with period $2^n$. Since no efficient algorithm for the general case has been discovered, we discuss the computational complexity of the underlying problem.

Our efficient $k$-complexity algorithm is applied in Chapter V to compute the $k$-complexities of a particular class of binary sequences—the de Bruijn sequences. De Bruijn sequences were chosen for two reasons. First, they have been extensively analyzed and a great deal is known about their linear complexities, and second, de Bruijn sequences have period $2^n$ and hence our efficient $k$-complexity algorithm can be applied. A new class of binary sequences which appear to have good $k$-complexity properties is also analyzed. Computational results are given for both this new class and the de Bruijn sequences.

Chapter VI summarizes the results obtained. The final chapter also includes several open problems related to the $k$-complexity.

# CHAPTER II
# CRYPTOLOGY

## 2.1 Introduction

*Cryptology*—the art of making and breaking "secret codes"—has a terminology all its own. The entire field is generally split into the two broad areas of *cryptography* (designing cryptosystems) and *cryptanalysis* (breaking or exposing the weaknesses of such systems). The process of converting the original message into its "secret" form is *encryption* or *enciphering* and the resulting secret message is a *cryptogram* or the *ciphertext*. The un-encrypted message is known as the *plaintext*. *Decrypting* or *deciphering* is the process of recovering the plaintext from the cryptogram.

Cryptography is not to be confused with coding theory. A *code* is simply a rule for replacing one piece of information with another—for example, the well-known ASCII code replaces alphanumeric characters with bit strings. Error correcting codes are designed to locate and correct errors which occur during transmission, making the original message easy to recover. In this sense, cryptography and coding theory are opposites.

Cryptosystems can be classified as *transposition ciphers*, *substitution ciphers*, or *product ciphers*. A transposition cipher rearranges the plaintext symbols without otherwise changing them, while a substitution cipher replaces plaintext characters with other symbols, without changing their order. Many sophisticated systems "cascade" some combination of these two basic operations. Such cascaded systems are referred to as product ciphers.

Some cryptosystems operate on single characters, while others encrypt blocks of a fixed size. The former are *stream ciphers* and the latter are *block ciphers*.

Every cryptosystem employs a key which determines the encryption and decryption process. In a *secret-key* system, the key must be transmitted to other users via a "secure channel." In *public-key* (or two-key) cryptography, each user has two keys, $K_E$ for encryption and $K_D$ for decryption. The encryption key $K_E$ is made public. In an ideal public-key system, any user could encrypt a message using the publicly available encryption key $K_E$, but only the corresponding $K_D$ (in the sole possession of the intended recipient) could decrypt the message. Note

that the recipient of a message cannot be certain which user actually sent the message, which illustrates the authentication problem. However, if the decryption key $K_D$ is made public and encryption key $K_E$ remains private, a secure authentication scheme results. Note that public-key cryptosystems avoid the key distribution problem inherent in private-key systems.

The goal of cryptanalysis—for both secret-key and public-key systems—is to recover the secret key. It is generally assumed that an enemy cryptanalyst knows the encryption process being used, but does not have direct access to the key. Various methods of "attacking" a cryptosystem may be available to a cryptanalyst, the most likely being the ***known-plaintext attack*** in which plaintext-cryptogram pairs formed using the actual secret key are available. Other possible methods of attack are a ***chosen-plaintext attack*** in which plaintext messages may be submitted and the resulting cryptograms obtained, and a ***chosen-ciphertext attack*** in which supposed cryptograms may be submitted and the resulting "plaintext" received.

## 2.2 History

Cryptology has a long and fascinating history which is described in overwhelming detail by Kahn [33]. A few of the many highlights are mentioned below.

- Credit for the first significant use of cryptography goes to Julius Caesar. The "Caesar cipher" is a simple transposition system in which each letter is replaced by the letter 3 positions ahead in the alphabet (*a* becomes *d*, *b* becomes *e*, and so on). This system apparently served the Caesars well, but today it could easily be broken using elementary techniques.

- The 1843 publication of Edgar Allen Poe's "The Gold-Bug" popularized cryptology in this country. In this short story Poe's eccentric friend Legrand finds a cryptogram on the beach. Legrand deduces that a simple substitution cipher is involved and he uses a system of frequency counts to decrypt the message. For example, since *e* is the most common letter in English, Legrand assumes that the most common character in the cryptogram represents *e* and so on. The resulting plaintext message is vague and mysterious,

but Legrand is once again able to determine the true meaning. His reward is a pirate's buried treasure.

- In the presidential election of 1876, the Republican, Rutherford B. Hayes, lost the popular vote by 250,000 to the Democrat Samuel J. Tilden. However, the 22 electoral votes of Florida, Louisiana, South Carolina, and Oregon were disputed. Congress created a commission to determine the the final disposition of the disputed votes. This commission split 8 to 7 along straight party lines and awarded all 22 votes to Hayes who then carried the electoral college by a margin of 1 vote. Not surprisingly, the Democrats claimed foul and even began insinuating that "Rutherfraud" Hayes was guilty of vote buying. Since no evidence of impropriety was forthcoming, Hayes was inaugurated on March 4, 1877.

  Congress was not satisfied to let the matter rest. Ironically, the only significant evidence was a collection of some 400 encrypted Western Union telegrams (i.e., cryptograms) which had been sent between Tilden's New York headquarters and officials in the 4 disputed states just prior to the electoral vote. In the summer of 1878 the *New York Daily Tribune* published several of these cryptograms, casting doubt on Tilden's claim of operating "in the keen bright sunlight of publicity" [23].

  Shortly thereafter the *Detroit Post* broke some of the ciphers. (Several systems had been used. For example, one method involved substituting the word in the *Household English Dictionary* which was four pages ahead of—and in the same position on the page as—the actual word.) The deciphered messages revealed that Democratic party bosses—not the Republican Hayes—had attempted to buy electoral votes in each of the 4 disputed states. The resulting scandal ruined Tilden and lead to a decisive Republican victory in the 1878 midterm election. The residual effects were still being felt in the presidential election of 1880 when Republican James A. Garfield narrowly defeated Democrat Winfield S. Hancock.

- In January 1917, World War I was entering its third bloody year. The carnage was incredible—over 315,000 French were lost to "the hell of Verdun"; British losses in a single day at the Somme exceeded 57,000 [19]. The

Germans had suffered comparable losses and yet the entrenched lines in the West remained intact. The British blockade was hurting Germany, but the Russian ally was collapsing in the East—the war was at a stalemate. Against this backdrop, the German generals were pushing hard for "unrestricted submarine warfare," even though such action would likely bring the United States into the war.

The German foreign minister, Arthur Zimmermann, had a plan. To counter the inevitable American reaction, he proposed to offer the government of Mexico a deal. In the event of war with the U.S., Mexico was to ally herself with Germany and in return, Mexico would "regain by conquest her lost territory in Texas, Arizona, and New Mexico" [80]. Coming on the heals of Pershing's expedition into Mexico, Zimmermann had some reason for optimism.

On January 16, 1917, the most famous cryptogram in history was sent from Berlin to the German ambassador in Washington who was to relay it to Mexico City. To ensure that this important message reached Washington, it was sent by two different routes. Since the German transatlantic cables had been cut on the first day of the war, the routes employed were unusual. The first was known by British cryptanalysts as the "Swedish roundabout"—from Berlin to Stockholm to Buenos Aires to Washington. Sweden transmitted the message from Stockholm to Buenos Aires, in violation of her proclaimed neutrality. The second route was shorter—from Berlin to Copenhagen to London to Washington—but even more bizarre, considering the content of the message. Since late 1916, the American government had allowed Germany to transmit cryptograms from London to Washington under American diplomatic auspices. The rationale was that improved communication might lead to a negotiated settlement.

Both of these cryptograms were intercepted by "Room 40," Britian's secret wartime cryptanalytic bureau. The code used by the German's was a difficult substitution cipher which had only been partially broken. Nevertheless, as early as January 17 enough had been decrypted to see that the Zimmermann telegram was an Allied propaganda windfall.

On February 1 unrestricted submarine warfare began; on February 3 the U.S. broke diplomatic relations with Germany. War between the U.S. and Germany appeared imminent, so the British—not wanting to inform the German's that their cables had been intercepted and deciphered—held the telegram. Room 40 had reason to suspect that the telegram which was sent from Washington to Mexico City had been encrypted using an older, easier system. Miraculously, a British agent (known only as "T") obtained the Mexico City cryptogram. Room 40's suspicions had been correct. As a result, the British government was able to get a precise reading of the telegram and to set up a clever ruse.

On February 22 the Zimmermann telegram was revealed to the American ambassador in London along with the true but misleading story that it had been obtained in Mexico City. The American public was outraged, but the U.S. Senate was skeptical. Zimmermann himself removed all doubt when he stated, "I cannot deny it. It is true" [80]. The U.S. declared war on April 2, with President Wilson citing the Zimmermann telegram as a key factor.

The German government investigated the matter and concluded that the Mexico City cryptogram had been the source of their troubles. As a result, Germany continued to employ the "Swedish roundabout," giving Room 40 access to much valuable intelligence.

- Whereas in World War I cryptanalysis played a minor role in the conduct of the war, in World War II it was often pivotal. Cryptanalysis was fundamental in at least four major events—the battle of Midway, the assassination of Admiral Yamamoto, the rapid cutting of Japan's lifeline, and the defeat of the German U-boats [33]. After the war some knowledgeable American officials estimated that cryptanalysis had shortened the war by one year. In any case, cryptanalysis had become the primary source of military intelligence.

- During the Cold War cryptology continued to play a key role. The post-war history of American cryptology is inextricably linked to the National Security Agency (NSA).

In 1952 the NSA was established as the primary cryptologic organization of the United States. The NSA is also the most secretive agency in this country—the government did not publicly acknowledge its existence until 1957. By the mid 1960's, the NSA was estimated to consume about 2% of the total defense budget [33].

The NSA has suffered a few highly publicized setbacks. Perhaps the most notable occurred in 1960 when two young technical employees, William H. Martin and Bernon F. Mitchell, defected to the Soviet Union via Cuba [33]. In Moscow, the two defectors held a press conference where they described details of NSA operations. As a result, many countries—including most U.S. allies—overhauled their cryptologic methods. The losses to the NSA must have been substantial, but ironically the losses to Soviet intelligence were probably even greater.

- In the 1970's the NSA found itself embroiled in controversy. In 1973 and again in 1974 the National Bureau of Standards (NBS) invited proposals for a data encryption standard (DES). The stated objective was to find a method of encryption that would be widely available, cheap, and secure. In response, IBM proposed a slight variant of their Lucifer system, a product block cipher which had been developed in 1970. The NBS subsequently selected the IBM system as a potential DES. The proposal was published in 1975 and comments were invited.

The DES met trouble from its inception. The security of the DES depends primarily on the key length—the longer the key, the more difficult is an exhaustive cryptanalysis. In the original IBM system, the key length had been 128 bits, but this was reduced in the proposal submitted to NBS. The published NBS proposal stated that the key length was 64 bits. However, a careful reading of the algorithm showed that 8 bits of the key were simply discarded and hence the effective key length was only 56 bits. The proposal also implied that cryptanalysis was impossible [55].

W. Diffie and M. E. Hellman led the attack on the DES. They suggested that a special purpose computer costing approximately $20,000,000 could be constructed which would allow an exhaustive cryptanalysis [17]. Diffie

and Hellman's results were disputed, but there was general agreement that the 56 bit key offered no margin of safety. It was felt that in about ten years advances in hardware might make effective cryptanalysis possible.

What was the role of the NSA in the adoption of the DES? The critics charged that NSA was behind the reduction in key length. The controversy reached a crescendo with the publication of "Computer encryption and the National Security Agency connection" [36] in *Science* in July 1977, even though the DES had been officially accepted as of early 1977.

A Senate investigation followed which showed that the NSA had been involved in the reduction of the key length—despite statements to the contrary by NBS and IBM. The NSA had also "indirectly assisted in the development of the S-box structure" [75]—a crucial component of the DES—and had classified the design.

In spite of these machinations, the DES is generally considered a secure cryptosystem. In the words of Morris [55]: "With proper use, the DES represents a large advance in the security available to the nonmilitary user."

- In 1976 Diffie and Hellman's paper "New directions in cryptography" [16] opened the field of public-key cryptology[1]—although National Security Agency officials have claimed that public-key cryptography was investigated by the NSA as early as the mid 1960's [73].

  Recall that the attraction of public-key systems is that they eliminate the key distribution problem of private-key cryptosystems. However, public-key systems are much slower and, more fundamentally, the underlying theory is based on computational complexity theory which presents some difficulties. This latter point is vividly illustrated by the case of the Merkle-Hellman public-key cryptosystem.

  In 1978 Merkle and Hellman proposed a public-key system based on the knapsack problem—a well-known NP-complete problem [53]. Shamir later

[1] Massey [49] notes that Diffie and Hellman's paper [16] and Merkle's paper [52] were submitted at about the same time, even though Merkle's paper—which contains many of the same ideas—appeared much later.

published "A polynomial-time algorithm for breaking the basic Merkle-Hellman cryptosystem" [70] in which it was shown that Merkle and Hellman were not using the full generality of the the knapsack problem. Nevertheless, there are public-key systems which have withstood the test of time, the best known being the RSA cryptosystem of Rivest, Shamir, and Adleman [63], which is based on the presumed intractability of factoring large composite integers.

Military and diplomatic needs for secure communication apparently still dominate the field of cryptology. However, businesses which handle large amounts of private information (banks, for example) are increasingly interested in inexpensive, fast, and secure cryptosystems. In addition—and for similar reasons—secure authentication schemes are of practical interest. In any area where privacy or security is a major concern, cryptology has a potential role to play. It has even been suggested that cryptosystems could be designed "to make big brother obsolete" [11].

## 2.3 The scientific era

Claude Shannon's 1948 paper "A mathematical theory of communication" [71] laid the foundation on which modern information theory is built. The following year Shannon published "Communication theory of secrecy systems" [72] which was based on his earlier paper. Massey [49] points out that this latter paper initiated the "era of scientific secret-key cryptology." Prior to 1949, most results were based on practical experience or intuition.

"A mathematical theory of communication" was revolutionary in its impact. "Communication theory of secrecy systems" was critically important in the development of cryptology, but it did not have the same impact as Shannon's earlier paper. This was due to the fact that the latter paper did not suggest new types of cryptosystems or new cryptanalytic methods, but instead it tended to confirm long-held beliefs. "New directions in cryptography" were not pursued until the mid 1970's when public-key cryptography burst onto the scene. Such public-key systems were only hinted at by Shannon.

## 2.4 A provably secure cryptosystem

A fundamental contribution of Shannon's paper [72] was a proof that the "random Vernam cipher" is secure—a fact which cryptologists had long suspected. The cryptosystem in question is a type of stream cipher which was developed by G. S. Vernam in 1926. In this system the message is converted into binary and a preselected "random" sequence—which acts as the key—is added (modulo 2) to this binary message. To recover the message, the receiving party simply adds the same random sequence. The random sequence is then discarded. For this reason, such a system is also referred to as a "one-time pad." Figure 2.1 gives a schematic for a one-time pad.

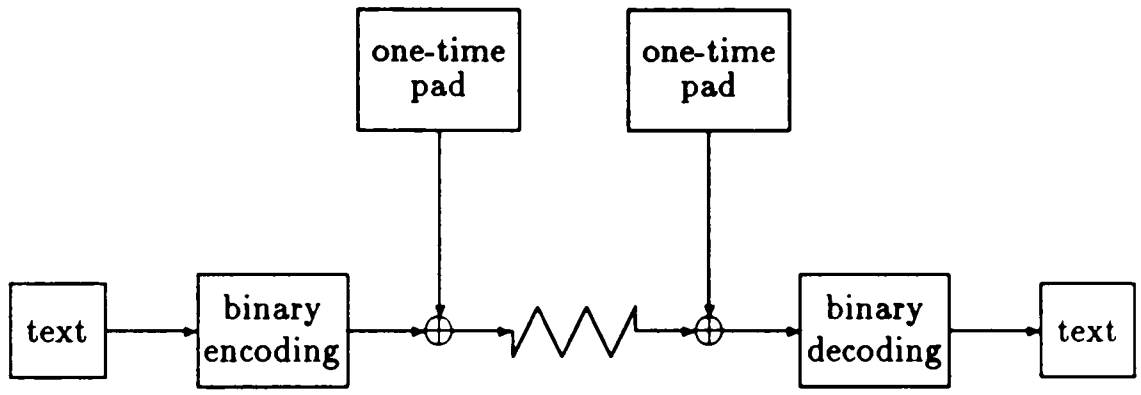


Figure 2.1: The one-time pad

Since the random Vernam cipher is a secret-key cryptosystem, the key (i.e., the one-time pad) must be securely transferred prior to the use of the system. However, the major drawback of such a system is that the key length is the same as the message length. In spite of this serious drawback, the provable security offered by the one-time pad has led to its use in certain critical situations. For example, the Moscow–Washington hotline employs a one-time pad [33].

The stream ciphers discussed in the next section are a variant of the one-time pad. In a stream cipher, the random sequence of the one-time pad is replaced by a pseudo-random sequence generated from a short secret key.

# CHAPTER III
# STREAM CIPHERS

## 3.1 Introduction

This chapter contains background material on stream ciphers, linear feedback shift registers, and linear complexity. We also define a new measure of the complexity of periodic sequences and we show that this new measure has potential cryptographic significance.

The cryptosystem illustrated in Figure 3.1 is generally referred to as a "stream cipher." In such a system, the text is first converted into a binary sequence which is then encoded to protect against errors in transmission. The encryption process is accomplished by adding (modulo 2) a pseudo-random sequence of bits. The resulting bit string is then transmitted. By simply reversing the process, the receiving party recovers the message. The book by Rueppel [65] contains a wealth of information on the theory of stream ciphers.

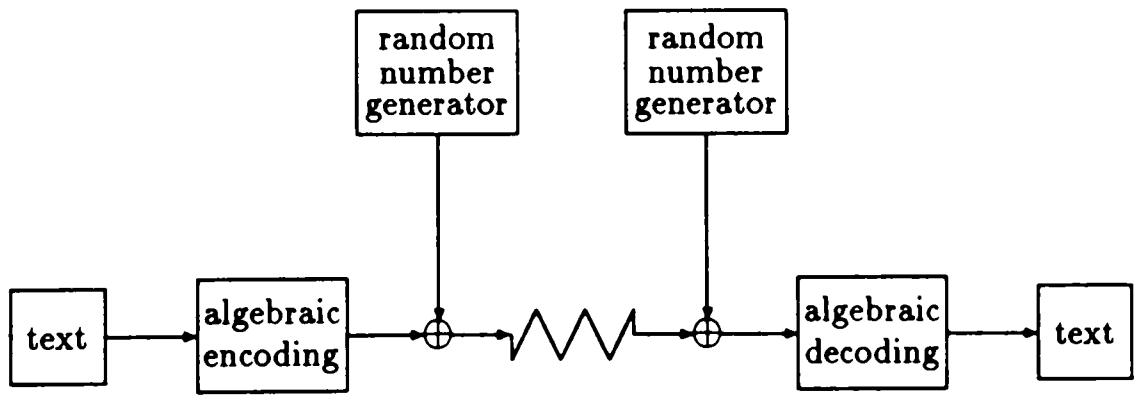


Figure 3.1: A stream cipher

The similarity between the stream cipher of Figure 3.1 and the one-time pad illustrated in Figure 2.1 is not accidental. We have simply replaced the random sequence of the one-time pad with a pseudo-random sequence generated from a short secret key. Thus the amount of secret key which must be securely transferred has been dramatically reduced. And since the one-time pad is provably secure, it is generally believed that if the pseudo-random sequence of the

stream cipher is "sufficiently random," cryptanalysis should be virtually impossible. However, such statements are difficult to formulate precisely and—as of today—impossible to prove.

It is assumed that a persistent eavesdropper will obtain some segment of the text and thereby obtain a segment of the pseudo-random sequence. For a stream cipher to be secure against such a known-plaintext attack, the pseudo-random sequence must be "unpredictable" in the sense that it must be computationally infeasible to recover more of the sequence from a captured segment.

There are at least two distinct approaches to this concept of unpredictability. Authors such as Shamir [69] and Blum and Micali [5] define a sequence to be *cryptographically strong* if determining the next element based on the previous elements is provably as difficult as inverting a one-way function, while authors such as Groth [29], Key [34], and Rueppel [65] emphasize linear complexity as the primary measure of unpredictability. Our approach follows this second line of attack.

Below we make frequent use of the term "cryptographically strong." This phrase is to be taken in an intuitive sense, not in the technical sense of [5] and [69].

## 3.2 Linear feedback shift registers

The pseudo-random number generator in Figure 3.1 must be capable of rapidly producing bits and it must be analyzable. Linear feedback shift registers (LFSR's) satisfy these requirements.

Figure 3.2 illustrates an LFSR which consists of 4 storage units, or *stages*, and a linear feedback function. Such an LFSR is controlled by a clock and at each clock pulse the element in $s_i$ is shifted into $s_{i-1}$, with the element $s_0$ taken as the output. The element inserted into the left-most stage ($s_3$ in Figure 3.2) is calculated from a linear recurrence.

It is easily verified that the LFSR in Figure 3.2 realizes the linear recurrence $s_i = -(s_{i-1} + s_{i-4})$, for all $i \geq 4$. The elements $s_i$ produced by any linear recurrence must satisfy a relation of the form

$$s_i = -\sum_{k=1}^{r} c_k s_{i-k} \quad \text{for all} \quad i \geq r. \tag{3.1}$$

Letting $E$ be the shift operator, i.e., $Es_i = s_{i+1}$, (3.1) takes the form

$$\left(E^r + \sum_{k=1}^{r} c_k E^{r-k}\right) s_i = 0.$$

The polynomial $C(E)$ of smallest degree such that $C(E)s_i = 0$ for all $i \geq 0$ is the *connection polynomial* (or feedback polynomial) of $s$. The linear recurrence in Figure 3.2 can be written as

$$(E^4 + E^3 + 1)s_i = 0,$$

and in this case the connection polynomial is indeed

$$C(E) = E^4 + E^3 + 1.$$

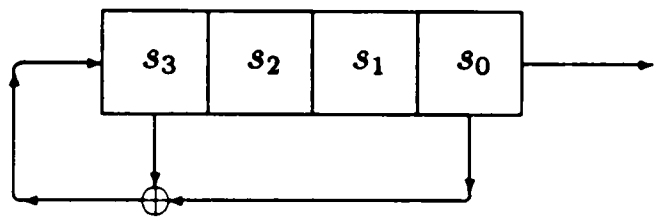


Figure 3.2: A 4-stage LFSR

The *characteristic polynomial* $c(x)$ is the reciprocal of the connection polynomial in the sense that $c(x) = x^n C(x^{-1})$, where $n$ is the degree of $C(E)$. For example, the characteristic polynomial of the LFSR in Figure 3.2 is

$$c(x) = x^4 + x + 1.$$

If an LFSR produces a periodic output sequence, the number of stages in the LFSR is equal to the degree of the corresponding characteristic polynomial. If the output sequence is not periodic, the LFSR is said to be singular and the degree of $c(x)$ is strictly less than the number of stages in the LFSR.

Taking $s_0 = 1$, $s_1 = 0$, $s_2 = 1$, and $s_3 = 0$ as initial values, the LFSR in Figure 3.2 generates the periodic output sequence

$$0,1,0,1,1,0,0,1,0,0,0,1,1,1,1,\underline{0,1,0,1}\ldots \tag{3.2}$$

which has period 15. Any other choice of initial values—with the obvious exception of $(0,0,0,0)$—produces a cyclic shift of (3.2).

The output from any LFSR is eventually periodic and the maximum possible period for an $n$-stage LFSR is $2^n - 1$. The output from an LFSR which attains this upper bound is referred to as an $m$-sequence (or pseudo-noise sequence).

Let $GF(q)$ be the *Galois field* with $q$ elements, i.e., $GF(q)$ is the $q$ element finite field. In this paper we are primarily concerned with $GF(2)$.

**Definition 3.1** *A primitive polynomial (modulo 2) is an irreducible polynomial of degree $n$ which is the minimal polynomial of a primitive root in $GF(2^n)$ [51].*

It is well known that an LFSR produces an $m$-sequence if and only if its characteristic polynomial is primitive. This fact was apparently first noted in an obscure work by Mantel [43] in 1895 and was rediscovered by Rees [62] in 1946.

We employ the notation $s$ for a finite sequence and $(s)$ for the periodic sequence obtained by appending copies of $s$. Given a periodic sequence $(s) = (s_0, s_1, \ldots, s_{n-1})$, with each $s_i \in \{-1, 1\}$, Golomb [24] has proposed the following "randomness postulates:"

**R–1** In each period the disparity between $+1$'s and $-1$'s does not exceed one.

**R–2** In each period, half of the runs have length one, one-fourth have length two, one-eighth have length three, and so on, provided that the number of such runs exceeds one. In addition, for each of these lengths, there are equally many runs of $+1$'s and $-1$'s.

**R–3** The auto-correlation function $C(\tau)$ assumes two values. In particular,

$$C(\tau) = \frac{1}{n}\sum_{i=0}^{n-1} s_i s_{i+\tau} = \begin{cases} 1 & \text{if } \tau = 0, \\ -1/n & \text{if } 0 < \tau < n. \end{cases}$$

The auto-correlation function in **R–3** measures the similarity between the sequence and its "phase shifts." The value of $C(\tau)$ is always highest for $\tau = 0$ and if a sequence is "random," $C(\tau)$ should be small for other values of $\tau$ [24].

Rueppel [65] notes that Golomb's "randomness postulates" almost exclusively describe $m$-sequences. These properties make $m$-sequences useful in many applications where pseudo-random bits are required. However, using the Berlekamp-Massey algorithm, only $2n$ consecutive bits are required in order to determine the

connection polynomial $C(E)$ and hence recover the entire $m$-sequence of length $2^n - 1$. It follows that $m$-sequences are cryptographically "weak." (However, certain nonlinear combinations of $m$-sequences do have desirable cryptographic properties [14,29,34,65].) This example also illustrates that statistical "randomness" does not insure cryptographic strength.

We have avoided the difficult but fascinating questions that arise whenever "random" sequences—and, in particular, finite random sequences—are discussed. The interested reader is referred to Knuth [35] and Martin-Löf [47] for guidance. We also note that various generalizations of LFSR's appear in the literature. For example, nonlinear feedback functions and nonlinear feedforward operations are frequently considered; see [4,24,29,34]. Martin and Stamp [44,45] (among others) consider the general case, where arbitrary feedback and feedforward functions are allowed. Nevertheless, LFSR's remain the cornerstone of most proposed pseudo-random sequence generators—undoubtedly due to their simplicity and analyzability.

## 3.3 Linear complexity

Linear complexity is of fundamental importance in the search for cryptographically strong pseudo-random sequences.

**Definition 3.2** *The linear complexity of a sequence is the minimum number of stages of an LFSR that generates the sequence.*

The linear complexity of a periodic sequence $(s)$ is the "length" of the shortest linear recurrence which generates $(s)$, i.e., the degree of the corresponding characteristic polynomial. The linear complexity of $(s) = (s_0, s_1, \ldots, s_{n-1})$ is bounded above by $n$ since the circulating shift register in Figure 3.3 will produce $(s)$. Note that the sequence (3.2) has linear complexity four, since the LFSR in Figure 3.2 generates $(s)$ and no LFSR with fewer than four stages can generate a sequence with period 15.

There is an elegant and efficient method for determining the linear complexity and the associated connection polynomial of any finite bit string or periodic sequence. This procedure is the well known Berlekamp-Massey LFSR synthesis algorithm [48]. In the special case where $(s)$ is a binary sequence with period $2^n$,

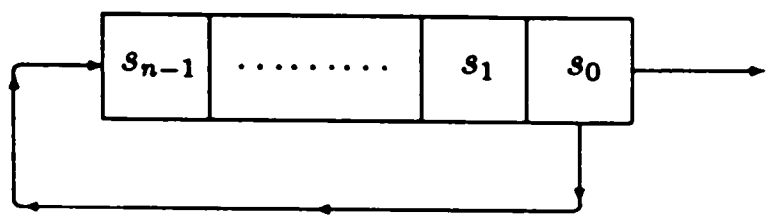


Figure 3.3: Circulating shift register

the lesser-known Chan-Games algorithm [21] gives an extremely efficient method for determining the linear complexity.

Due to the Berlekamp-Massey algorithm, it is necessary that a cryptographically strong sequence have a high linear complexity. However, this is not sufficient since, for example, the sequence

$$(s) = (\underbrace{0, 0, \ldots, 0, 1}_{n \text{ elements}}) \tag{3.3}$$

has linear complexity $n$—which is the highest possible—but $(s)$ is extremely "non-random." A more subtle example exhibiting this type of behavior is given by Safavi-Naini and Seberry [66].

The *linear complexity profile* of $(s)$ is obtained by plotting the linear complexity of $s_0 s_1 \cdots s_m$ against $m$ for $m = 0, 1, 2, \ldots$. The required linear complexities are obtained when the Berlekamp-Massey algorithm is applied to $(s)$. Rueppel [65] suggests the linear complexity profile as a means of detecting sequences with high linear complexities, but which are not cryptographically strong, such as the sequence (3.3). Rueppel concludes that a cryptographically strong sequence must have a linear complexity near the period length—which is the highest possible—and a linear complexity profile which follows the $m/2$ line "closely but irregularly." Figure 3.4 contains an example of a typical (good) linear complexity profile and Figure 3.5 gives an example of a very poor linear complexity profile.

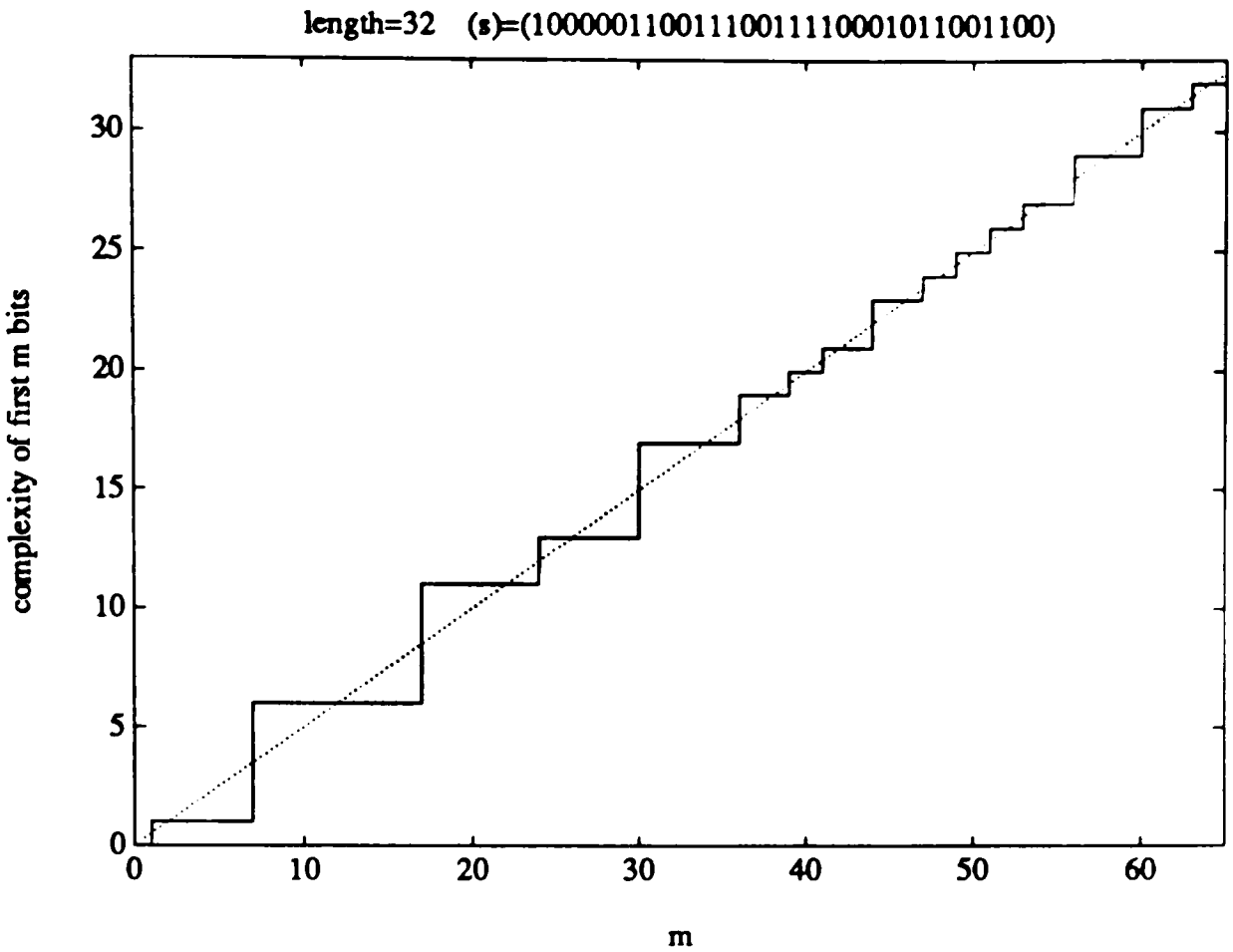


Figure 3.4: A typical linear complexity profile

## 3.4 The $k$-complexity

We now propose a new measure of the complexity of a periodic sequence. Let the $k$-complexity of a periodic sequence $(s)$ be the smallest linear complexity that can be obtained by altering any $k$ or fewer elements of $s$. More formally,

**Definition 3.3** *Let $(s)$ be a periodic sequence with*

$$s = s_0 s_1 s_2 \cdots s_{n-1}.$$

*Then for $k \in \{0, 1, \ldots, n\}$, the $k$-complexity of $(s)$, denoted $c_k(s)$, is the minimum number of stages of an LFSR that generates a sequence $(t)$, where $t$ is of length $n$ and $t_j = s_j$ for at least $n - k$ of the indices.*

For example, the 0-complexity of any sequence is just the usual linear complexity. Also, the 1-complexity of the sequence (3.3) is zero, since changing one bit produces the all-zero sequence, which—by definition—has linear complexity

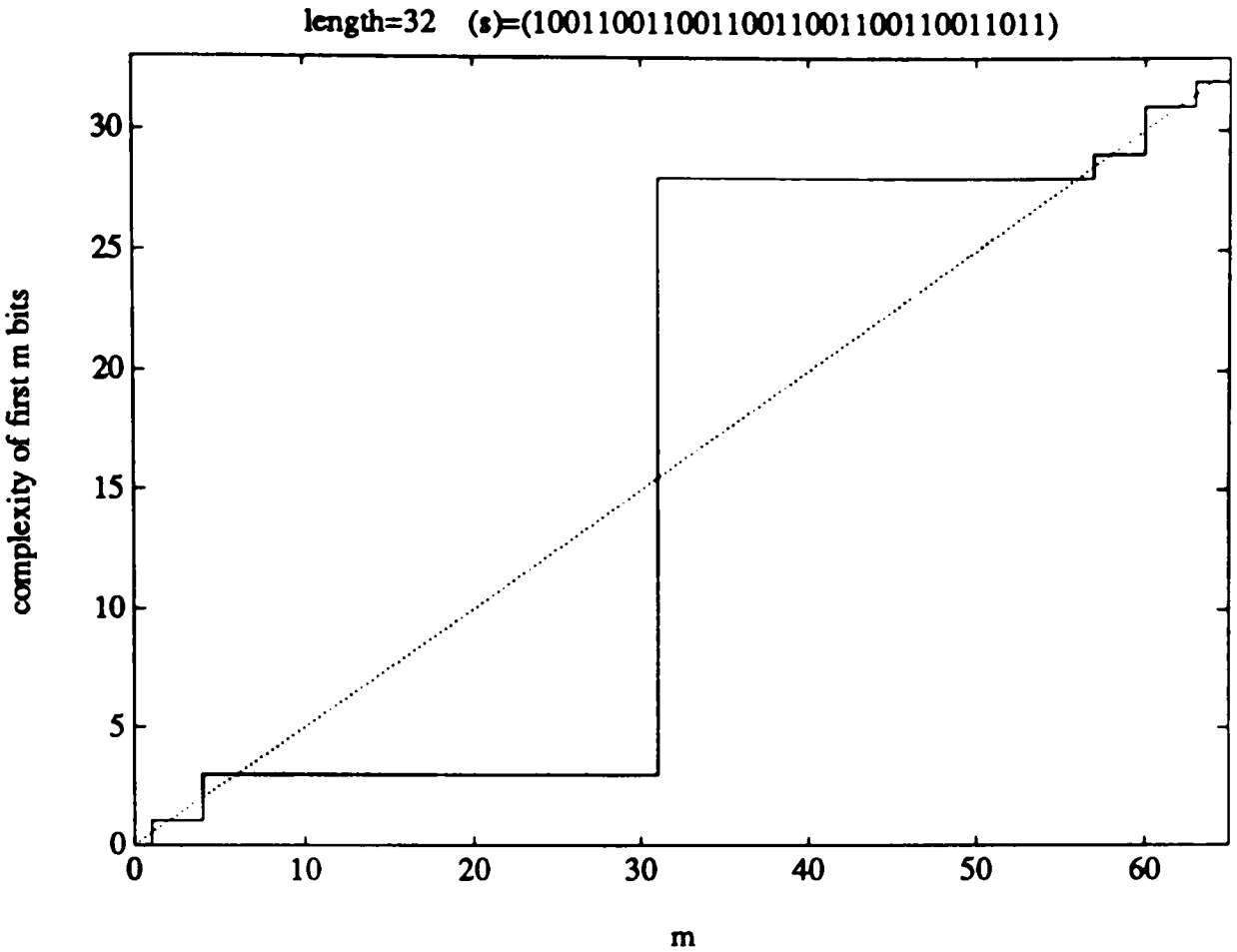


Figure 3.5: A poor linear complexity profile

zero. Note that for each $k \geq \lfloor n/2 \rfloor$, where $\lfloor x \rfloor$ is the greatest integer in $x$, the $k$-complexity is at most one.

One reason for considering the $k$-complexity is similar to the reason for examining the linear complexity profile, namely, we would like to identify those sequences which have a high linear complexity but which fail to be cryptographically strong. The $k$-complexity of a sequence can also be interpreted as a worst-case measure of the linear complexity when $k$ or fewer errors occur.

The concept of $k$-complexity is intuitively appealing. If a relatively small LFSR "almost" generates $(s)$, then $(s)$ should not be considered cryptographically strong. The sequence in (3.3) is an extreme example of this phenomenon.

For $s = s_0 s_1 \cdots s_{n-1}$, we refer to the sequence of $k$-complexities, for $k = 0, 1, 2, \ldots, \lfloor n/2 \rfloor$, as the *k-complexity profile* of $(s)$. Figure 3.6 contains an example of a typical $k$-complexity profile, while Figure 3.7 shows a very poor $k$-complexity profile.

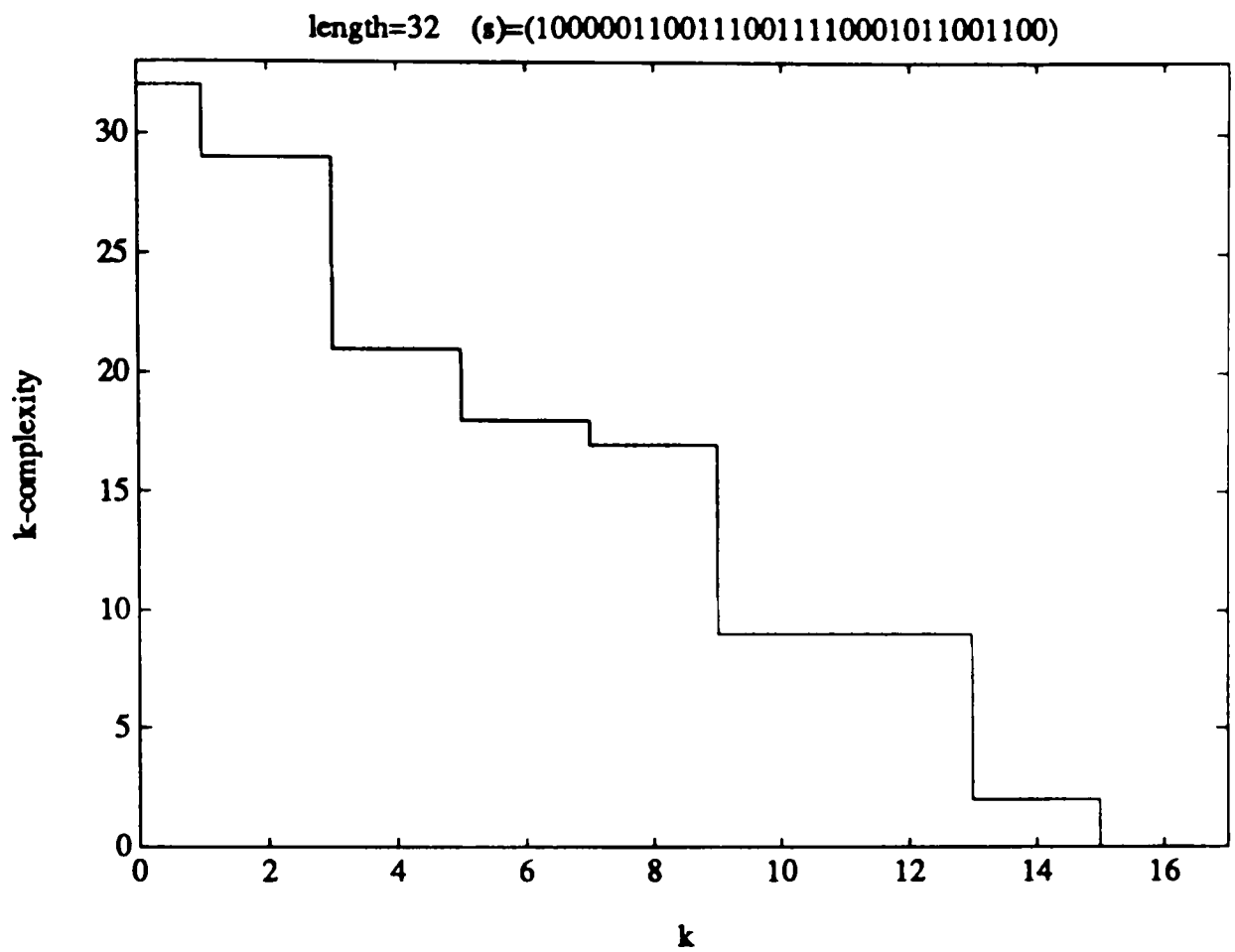


Figure 3.6: A typical $k$-complexity profile

The next section contains an example of a sequence with a high linear complexity and a typical linear complexity profile, but which has a poor $k$-complexity profile—i.e., the $k$-complexity drops sharply for small values of $k$. It follows that the $k$-complexity profile actually contains more information about the cryptographic strength of a sequence than the linear complexity profile. Unfortunately, there is no obvious analogue to the Berlekamp-Massey algorithm for computing the $k$-complexity. Nevertheless, we have found an efficient algorithm—which generalizes the Chan-Games algorithm [21]—for computing the $k$-complexity of binary sequences with period $2^n$ [77]. The general case appears to be more difficult and is discussed in the next chapter.

## 3.5 An example

Consider the sequence

$$(s) = (00011010100110101000101010011010)$$

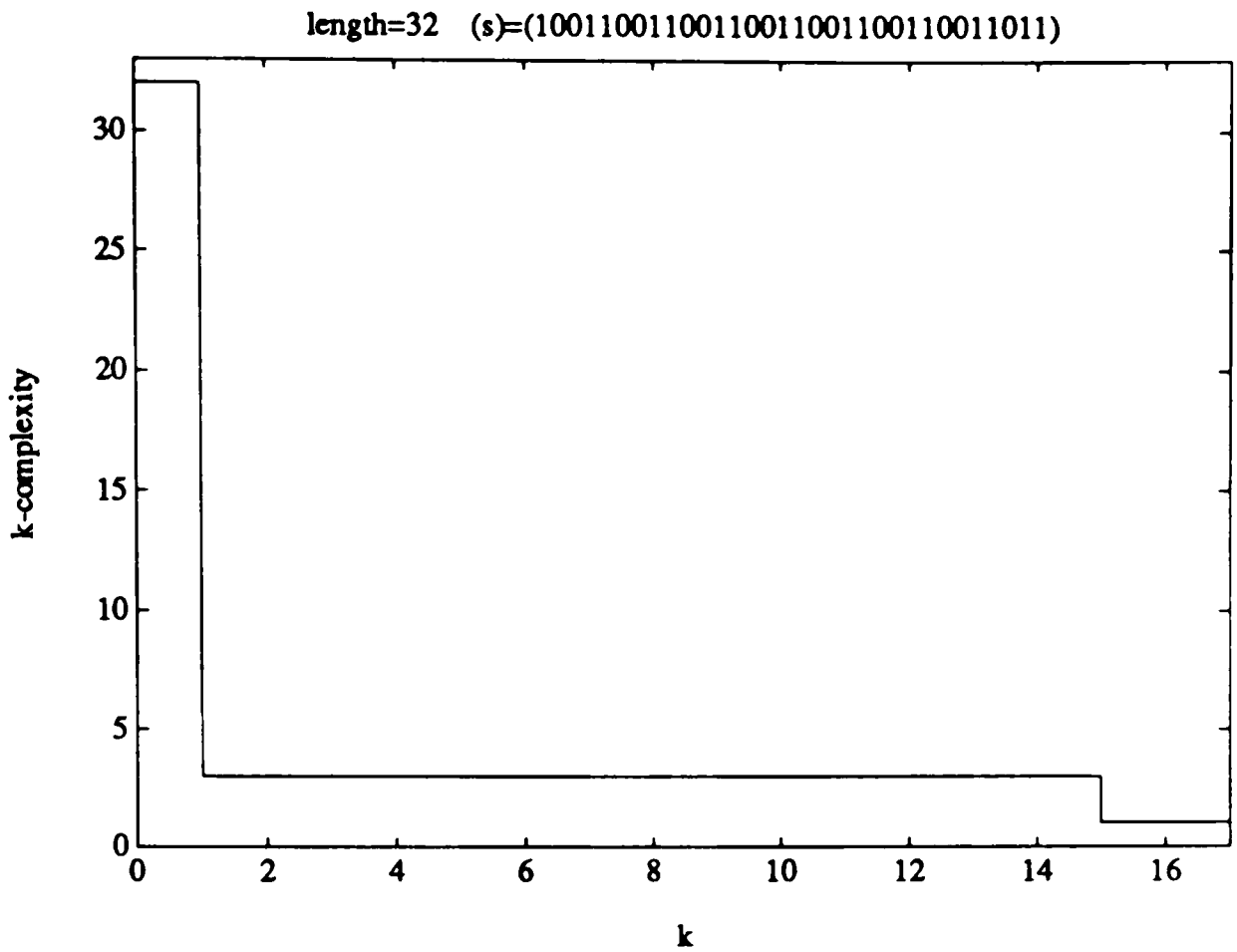


Figure 3.7: A poor $k$-complexity profile

which has period length 32. This sequence appears to satisfy the two criteria given by Rueppel [65] for being cryptographically strong, namely, it has a linear complexity of 31 (near the maximum possible 32) and its linear complexity profile—which appears in Figure 3.8—follows the $m/2$ line closely and does not exhibit any obvious regularity. Yet the $k$-complexity profile, which appears in Figure 3.9, is poor, since an LFSR with just 7 stages will correctly produce 30 of every 32 bits of $(s)$. Even more dramatic examples exist for larger values of $n$.

In this chapter we have provided the requisite background material on stream ciphers, linear feedback shift registers, and linear complexity. The $k$-complexity of periodic sequences was also defined and it was shown that in certain cases, the $k$-complexity provides more information on the cryptographic strength of a sequence than the linear complexity profile.

The remainder of the paper deals with problems related to the $k$-complexity. Known algorithms for computing the linear complexity are discussed in the next

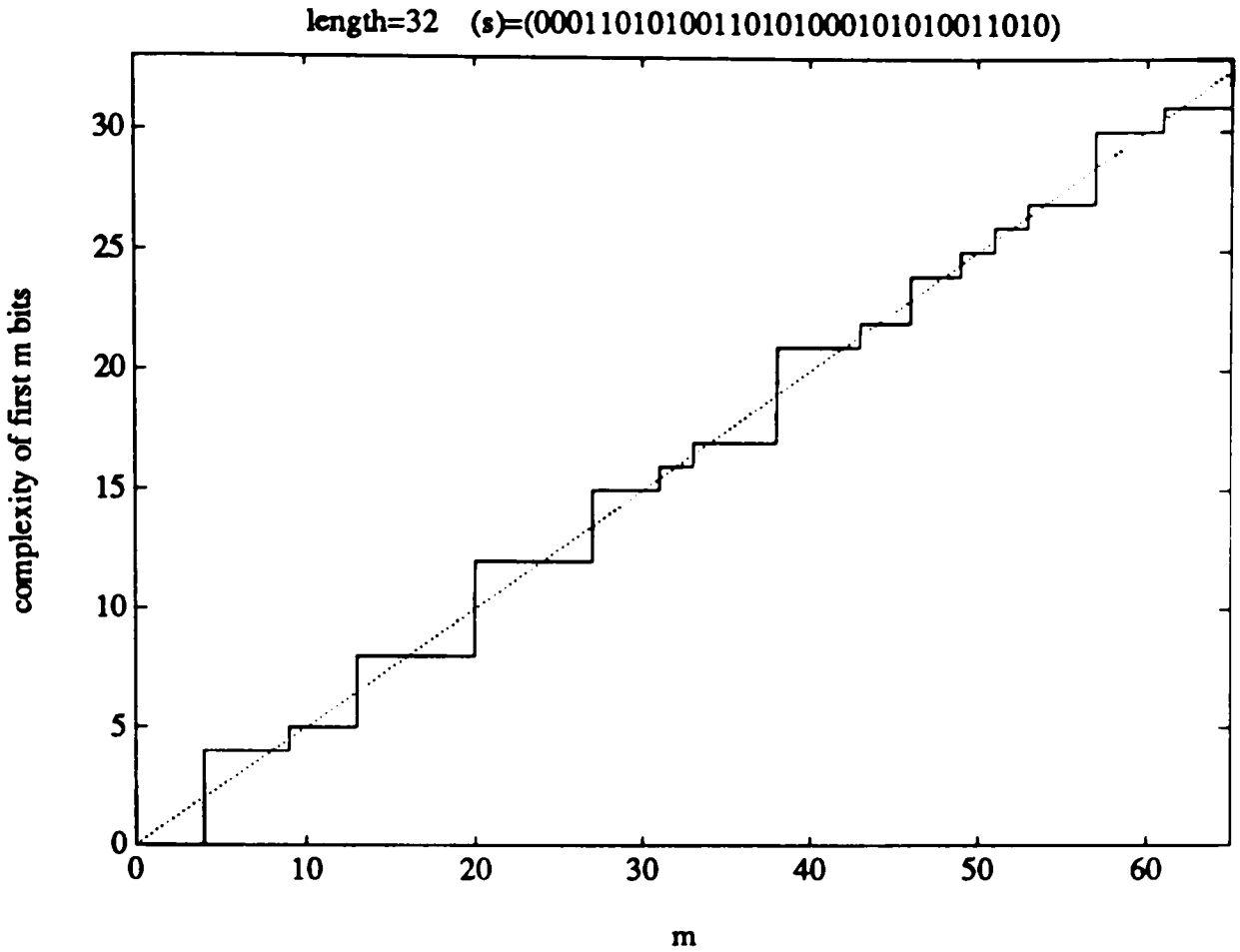


Figure 3.8: A linear complexity profile

chapter along with a new algorithm for finding the $k$-complexity of binary sequences with period $2^n$. The standard algorithm for computing the linear complexity is the Berlekamp-Massey algorithm. Since this algorithm is closely related to the continued fraction algorithm, continued fractions are mentioned. The problem of finding an efficient algorithm for the $k$-complexity also leads to a brief consideration of computational complexity theory.

An analysis of the $k$-complexities of the de Bruijn sequences is the focus of Chapter V. A new class of binary sequences is also considered and several computational results are given.

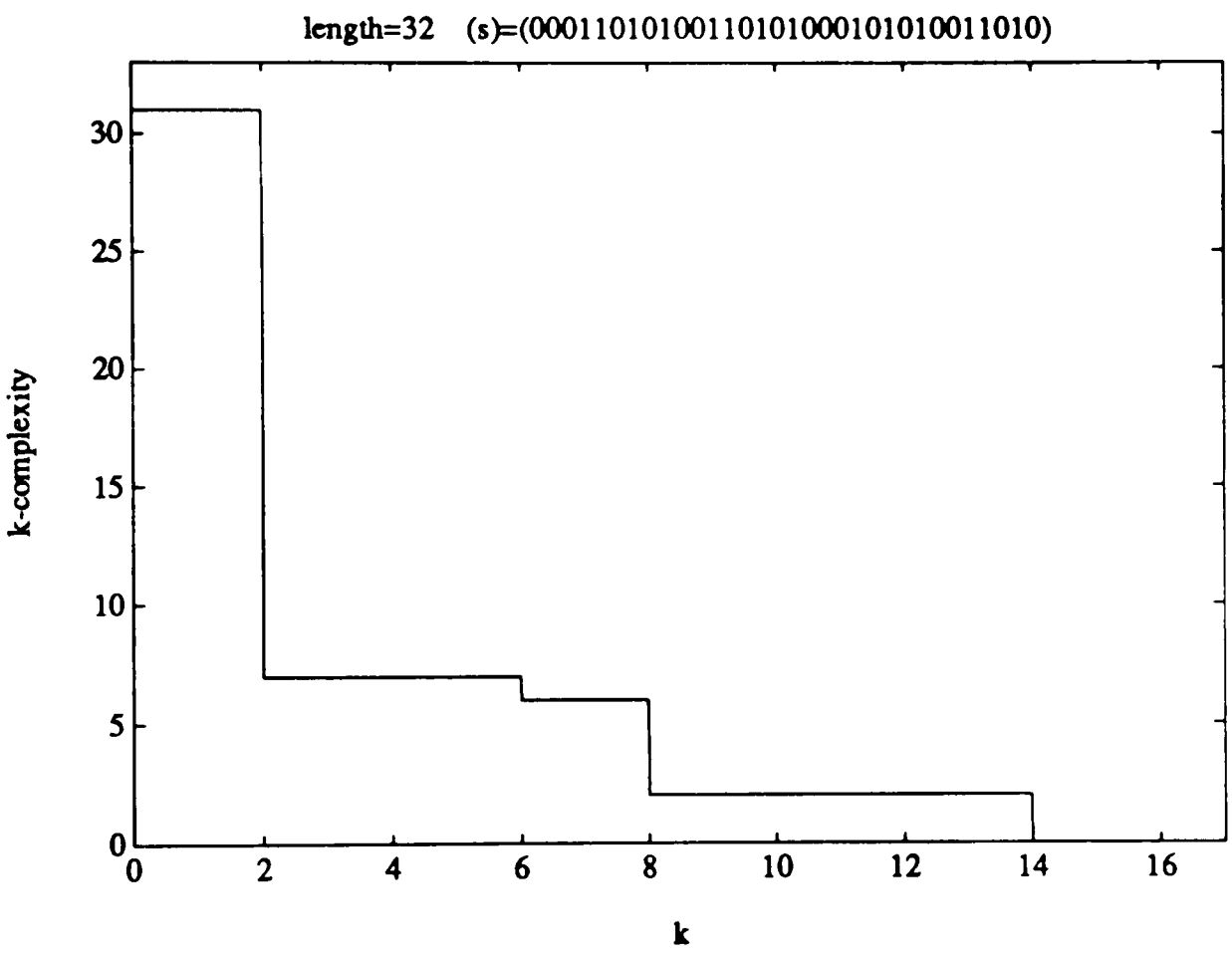


Figure 3.9: A $k$-complexity profile

# CHAPTER IV
# A $k$-COMPLEXITY ALGORITHM

## 4.1 Introduction

In this chapter, we first discuss the Berlekamp-Massey algorithm—including its connection with the continued fraction algorithm. Then the Chan-Games algorithm is presented and a completely new—and slightly more general—proof of its validity its given. Next, a new and efficient algorithm for computing the $k$-complexity of periodic sequences with period $2^n$ is proved and an example of its application is given. Since no efficient algorithm for the general case has been found, we conclude the chapter with a discussion of the computational complexity of the general problem of computing the $k$-complexity.

## 4.2 The Berlekamp-Massey algorithm

Berlekamp's book [2] contains an efficient algorithm for decoding BCH codes. Massey has shown that Berlekamp's algorithm is most naturally interpreted as an algorithm for "synthesizing the shortest linear feedback shift register capable of generating a prescribed finite sequence of digits" [48]. The resulting Berlekamp-Massey algorithm appears here as Algorithm 4.1, where $L$ is the linear complexity,

$$C(E) = 1 + c_1E + c_2E^2 + \cdots + c_LE^L$$

is the associated connection polynomial, and $s$ is a finite sequence of length $n$. Note that the algorithm accepts $s$ sequentially and at step $N$ produces the minimal LFSR for the first $N$ elements of $s$. It has been shown by Gustavson [30] that the algorithm requires $O(n^2)$ multiplications in the field of definition, except in the binary case where $O(n^2)$ binary additions are required.

The Berlekamp-Massey algorithm is important in the theory of stream ciphers since it gives an effective method for finding an LFSR which produces a given pseudo-random sequence. In fact, if an $n$-stage LFSR produces a sequence $(s)$, the Berlekamp-Massey algorithm requires only $2n$ consecutive elements of $(s)$ in order to completely determine the linear recurrence. It follows that a high linear complexity is necessary for a pseudo-random sequence to be considered

$C(E) = 1; \quad B(E) = 1; \quad L = 0;$
$b = 1; \quad x = 1; \quad N = 0;$
**while** $N < n$ **do**
    $d = s_N + \sum_{i=1}^{L} c_i s_{N-i};$
    **if** $d = 0$ **then**
        $x = x + 1;$
    **else**
        **if** $2L > N$ **then**
            $C(E) = C(E) - db^{-1}E^x B(E);$
            $x = x + 1;$
        **else**
            $T(E) = C(E);$
            $C(E) = C(E) - db^{-1}E^x B(E);$
            $L = N + 1 - L;$
            $B(E) = T(E);$
            $b = d;$
            $x = 1;$
        **end if**
    **end if**
    $N = N + 1;$
**end while**

Algorithm 4.1: Berlekamp-Massey algorithm

cryptographically strong, but the sequence in (3.3) illustrates that a high linear complexity is not sufficient to ensure cryptographic strength.

The Berlekamp-Massey algorithm has been intensively studied in the literature. For example, Reeds and Sloane [61] have extended the algorithm to the case where the elements of $s$ are integers (modulo $m$); the algorithm has been adapted by Mandelbaum [42] to produce convergents of continued fractions for binary numbers (without division); a simplified derivation has been given by Imamura and Yoshida [32]; connections with the continued fraction algorithm have been noted in [12,13,56,82] and elsewhere. In the next section, the connection between the Berlekamp-Massey algorithm and the continued fraction algorithm is made clear.

## 4.3 Continued fractions

The continued fraction algorithm is of fundamental importance in modern cryptology. Lidl [39] surveys some of the many applications of continued fraction techniques to cryptology. Here we are concerned with the relation between the continued fraction algorithm and the Berlekamp-Massey algorithm.

Mills [54] and van der Poorten [59] have noted the relation between linear recurrences and the continued fraction algorithm. Since the Berlekamp-Massey algorithm computes the connection polynomial of a finite sequence, it is not surprising that there is a connection between the Berlekamp-Massey algorithm and the continued fraction algorithm. This connection was apparently first noted by Welch and Scholtz [82].

Each real number $a \in [0, 1)$ has a continued fraction expansion of the form

$$a = \cfrac{1}{a_1 + \cfrac{1}{a_2 + \cfrac{1}{a_3 + \cdots}}}$$

which terminates if and only if $a$ is rational. The same technique can be applied more generally and is of significance, for example, in the problem of rational interpolation and in the partial realization problem of linear systems theory. Gragg [27] discusses the connections between the partial realization problem and the continued fraction algorithm.

Welch and Scholtz [82] consider the continued fraction algorithm in the following general setting. Let $\mathcal{R}$ be a commutative ring with multiplicative identity $I$ and suppose that $\mathcal{R}$ is contained in a field $\mathcal{F}$ and, further, suppose that there is a specified method of uniquely associating an element $[a]$ of $\mathcal{R}$ with a given field element $a$. In the standard case $\mathcal{F}$ is the set of real numbers and $\mathcal{R}$ is the ring of integers. We therefore refer to $[a]$ as the integer part and $\{a\} = a - [a]$ as the fractional part of $a$.

In this general setting we have

$$\begin{aligned} a &= a_0 + r_0 \\ &= a_0 + \cfrac{I}{a_1 + r_1} \end{aligned}$$

$$= a_0 + \cfrac{I}{a_1 + \cfrac{I}{a_2 + r_2}}$$
$$\vdots$$

where

$$a_0 = [a], \quad a_n = \left[\frac{I}{r_{n-1}}\right], \quad r_{n-1} \neq \mathcal{O},$$
$$r_0 = \{a\}, \quad r_n = \left\{\frac{I}{r_{n-1}}\right\}, \quad r_{n-1} \neq \mathcal{O},$$

and $\mathcal{O}$ is the additive identity in $\mathcal{F}$. A rational approximation is obtained by taking

$$S_n = \frac{P_n}{Q_n}$$

where $P_n$ and $Q_n$ are determined by the recurrences

$$P_{-1} = I, \quad P_0 = a_0, \quad \text{and} \quad P_{n+1} = a_{n+1}P_n + P_{n-1} \quad \text{for} \quad n = 0, 1, 2, \ldots$$

and

$$Q_{-1} = \mathcal{O}, \quad Q_0 = I, \quad \text{and} \quad Q_{n+1} = a_{n+1}Q_n + Q_{n-1} \quad \text{for} \quad n = 0, 1, 2, \ldots. \quad (4.1)$$

We are interested in the case where $\mathcal{F}$ is the collection of formal Laurent series over a finite field and $\mathcal{R}$ is the ring of polynomials. If the underlying field is the two element field $GF(2)$, then a typical element of $\mathcal{F}$ is of the form

$$S(z) = \sum_{j=0}^{\infty} s_j z^{d-j}, \quad s_j \in \{0, 1\}, \quad s_0 \neq 0,$$

where $d$ is an integer. The fractional part of $S(z)$ consists of the negative powers of $z$.

The continued fraction algorithm can be applied to produce rational approximations to $S(z)$. It is shown in [82] that the continued fraction algorithm is closely related to the Berlekamp-Massey algorithm, although the following theorem of Niederreiter [56] makes the connection much more explicit.

**Theorem 4.1** *[56, Theorem 1] Let $s = s_1 s_2 s_3 \ldots$ be a sequence of elements of a field $\mathcal{F}$ and define*

$$S(z) = \sum_{i=1}^{\infty} s_i z^{-i}.$$

*Then for any $n \geq 1$ the linear complexity of $s_1 s_2 \ldots s_n$ is given by $deg(Q_j)$, where $j$ is uniquely determined by the condition*

$$deg(Q_{j-1}) + deg(Q_j) \leq n < deg(Q_j) + deg(Q_{j+1}),$$

*and where the $Q_i$'s are given by (4.1). Furthermore, if $\mathcal{F} = GF(2)$, the corresponding connection polynomial is given uniquely by $Q_j$.*

Welch and Scholtz [82] also discuss the problem of measuring the error in the rational approximation $P_j/Q_j$. They show that this error is in some sense better than the corresponding result for real numbers. Their analysis involves non-Archimedian norms and other subjects found in the book by Lightstone and Robinson [40]. These results may be the key to finding an efficient general algorithm—or an efficient approximate algorithm—for computing the $k$-complexity.

## 4.4 The Chan-Games algorithm

Algorithm 4.2—the Chan-Games algorithm—was introduced by Games and Chan in [21]. This algorithm finds the linear complexity $c$ of a binary sequence with period $n = 2^m$. Note that only $m$ iterations are required and the $k$th iteration requires only $2^{m-k}$ binary additions. Hence the algorithm is—in terms of binary additions—$O(n)$. Unfortunately, the entire period must be stored, making the algorithm unsuitable for sequences with extremely long periods.

The Chan-Games algorithm is easy to understand and to apply. For example, in Figure 4.1 the algorithm is used to compute the linear complexity of the sequence

$$(s) = (1101010001011100).$$

The proof of the Chan-Games algorithm given in [21] relies heavily on properties of the basis sequences

$$\binom{i}{j} = \left( \binom{0}{j}, \binom{1}{j}, \ldots \right),$$

```
a = s;  c = 0;  ℓ = 2^n;
while ℓ > 1 do
    ℓ = ℓ/2;
    L = a_0 a_1 ··· a_{ℓ-1};
    R = a_ℓ a_{ℓ+1} ··· a_{2ℓ-1};
    b = L + R;
    if b = 0⃗ then
        a = L;
    else
        c = c + ℓ;
        a = b;
    end if
end while
if a_0 = 1 then
    c = c + 1;
end if
```

Algorithm 4.2: Chan-Games algorithm

where the binomial coefficients are reduced modulo 2. The use of the symbol $\binom{i}{j}$ for both the binomial coefficient and a basis sequence is somewhat confusing. Below, we prove a slightly more general result using completely different methods.

For $p$ prime, let $\mathcal{S}_p(n)$ be the collection of finite sequences of length $n$ over the finite field $GF(p)$. For $s \in \mathcal{S}_p(p^n)$ define

$$B_i(s) = s_{ip^{n-1}} s_{ip^{n-1}+1} \cdots s_{(i+1)p^{n-1}-1} \tag{4.2}$$

for $i = 0, 1, 2, \ldots, p-1$. Then $(s) = (B_0(s)|B_1(s)|\cdots|B_{p-1}(s))$, where $(a|b)$ denotes the concatenation of $a$ and $b$.

We now prove a general result which enable us to establish the Chan-Games algorithm as a special case. Our method of proof is distinct from—and more general than—the original proof given by Games and Chan [21].

**Theorem 4.2** *Let $p$ be prime, $s \in \mathcal{S}_p(p^n)$, and let*

$$b = B_0(s) + B_1(s) + \cdots + B_{p-1}(s),$$

*where the the $B_i(s)$'s are as in (4.2) and the additions are defined elementwise*

$$(s) = (1101010001011100)$$

| | | | | |
|---|---|---|---|---|
| Step 1) | $\ell$ | = | 8 | |
| | $L$ | = | 11010100 | |
| | $R$ | = | $\underline{01011100}$ | |
| | $b$ | = | 10001000 | $c = 8$ |
| | | | | $a = b$ |
| Step 2) | $\ell$ | = | 4 | |
| | $L$ | = | 1000 | |
| | $R$ | = | $\underline{1000}$ | |
| | $b$ | = | 0000 | $c = 8$ |
| | | | | $a = L$ |
| Step 3) | $\ell$ | = | 2 | |
| | $L$ | = | 10 | |
| | $R$ | = | $\underline{00}$ | |
| | $b$ | = | 10 | $c = 10$ |
| | | | | $a = b$ |
| Step 4) | $\ell$ | = | 1 | |
| | $L$ | = | 1 | |
| | $R$ | = | $\underline{0}$ | |
| | $b$ | = | 1 | $c = 11$ |
| | | | | $a = b$ |
| | $a_0$ | = | 1 | $c = 12$ |

Figure 4.1: Applying the Chan-Games algorithm

*modulo p. If* $b \neq \vec{0}$, *the linear complexity of* $(s)$ *satisfies*

$$c_0(s) = (p-1)p^{n-1} + c_0(b).$$

PROOF: Let $r = c_0(b)$ and let $f(E)$ be the connection polynomial of $b$, where $b = b_0 b_1 \ldots b_{p^{n-1}-1}$. Then

$$f(E)b_{i-r} = 0 \text{ for all } i \geq r, \tag{4.3}$$

where $\deg(f) = r$, $E$ is the shift operator, and $r$ is the smallest number such that a relation of the form (4.3) holds. From (4.3) and the definition of $b$ it follows that

$$f(E)(s_{i-r} + s_{i-r+p^{n-1}} + \cdots + s_{i-r+(p-1)p^{n-1}}) = 0$$

and hence

$$g(E)s_{i-r+(p-1)p^{n-1}} = 0 \text{ for all } i \geq r + (p-1)p^{n-1}$$

for some $g$ with $\deg(g) = r + (p-1)p^{n-1}$. We have thus established that $c_0(s) \leq (p-1)p^{n-1} + c_0(b)$.

Since $s \in \mathcal{S}_p(p^n)$, we have

$$(E^{p^n} - 1)s_i = (E-1)^{p^n} s_i = 0 \text{ for all } i \geq 0.$$

It follows that the connection polynomial of $s$, denoted $C(E)$, must divide

$$E^{p^n} - 1 = (E-1)^{p^n}.$$

Letting $c = c_0(s)$, we thus have $C(E) = (E-1)^c$.

Suppose $c \leq (p-1)p^{n-1}$. Then

$$(E-1)^{(p-1)p^{n-1}} s_i = (E-1)^{(p-1)p^{n-1}-c}(E-1)^c s_i = 0.$$

Since $(E-1)^{(p-1)p^{n-1}} = (E^{p^{n-1}} - 1)^{p-1}$ we have

$$\begin{aligned}(E-1)^{(p-1)p^{n-1}} &= E^{(p-1)p^{n-1}} - \binom{p-1}{1} E^{(p-2)p^{n-1}} + \cdots - \binom{p-1}{p-2} E^{p^{n-1}} + 1. \quad (4.4)\end{aligned}$$

It is an elementary result from number theory that [8]

$$(-1)^k \binom{p-1}{k} \equiv 1 \pmod{p} \text{ for } k = 1, 2, \ldots, p-2. \quad (4.5)$$

Combining (4.4) and (4.5) we have

$$\begin{aligned}(E-1)^{(p-1)p^{n-1}} s_i &= (E^{(p-1)p^{n-1}} + E^{(p-2)p^{n-1}} + \cdots + E^{p^{n-1}} + 1)s_i \quad (4.6)\\ &= 0.\end{aligned}$$

But the right-hand side of (4.6) is just $b_i$ and we have contradicted the assumption that $b \neq \vec{0}$. This contradiction shows that $c_0(s) \geq (p-1)p^{n-1} + 1$ and we have established that

$$(p-1)p^{n-1} + 1 \leq c_0(s) \leq (p-1)p^{n-1} + c_0(b).$$

If $r = c_0(b) = 1$ then $c = (p-1)p^{n-1}+1$ and we are finished. If $r > 1$, we must show $c = (p-1)p^{n-1}+r$. Suppose $c < (p-1)p^{n-1}+r$. Then $c = (p-1)p^{n-1}+r-\ell$, where $1 \leq r-\ell < r$. Once again we apply (4.5) and we find that

$$\begin{aligned}(E-1)^c s_i &= (E-1)^{r-\ell}(E-1)^{(p-1)p^{n-1}} s_i \\ &= (E-1)^{r-\ell}(E^{p^{n-1}}-1)^{p-1} s_i \\ &= (E-1)^{r-\ell} b_i = 0.\end{aligned}$$

But then $c_0(b) \leq r-\ell < r$, contradicting the fact that $r = c_0(b)$. Hence $c_0(s) = (p-1)p^{n-1} + c_0(b)$. ∎

**Corollary 4.1** *Let $s \in \mathcal{S}_2(2^n)$ and let $b = L(s) + R(s)$, where*

$$L(s) = s_0 s_1 \ldots s_{2^{n-1}-1} \quad \textit{and} \quad R(s) = s_{2^{n-1}} s_{2^{n-1}+1} \ldots s_{2^n-1}.$$

*Then if $b \neq \vec{0}$, we have $c_0(s) = 2^{n-1} + c_0(b)$*

Theorem 4.3 follows immediately from Corollary 4.1 and the observation that $b = \vec{0}$ implies $L(s) = R(s)$, in which case the linear complexity of $(s)$ is just the linear complexity of $L(s)$.

**Theorem 4.3** *[21, Theorem 6] Let $s \in \mathcal{S}_2(2^n)$ and let $b = L(s) + R(s)$. Then*

$$c_0(s) = \begin{cases} c_0(L(s)) & \textit{if } b = \vec{0} \\ 2^{n-1} + c_0(b) & \textit{if } b \neq \vec{0}. \end{cases}$$

The Chan-Games algorithm is obtained by applying Theorem 4.3 recursively. It is apparent from Theorem 4.2 that no simple analog of the Chan-Games algorithm exists for $s \in \mathcal{S}_p(p^n)$ for $p$ prime, $p > 2$, since in this case $b = \vec{0}$ does not necessarily imply that $B_0 = B_1 = \cdots = B_{p-1}$. However, if $b \neq \vec{0}$ at each step, then exact results would be obtained using the procedure in Theorem 4.2, but if $b = \vec{0}$ and $B_i \neq B_j$ at some stage, we can only obtain approximate results. Upper bounds are not particularly useful and, unfortunately, the lower bounds obtained may be poor.

## 4.5 An efficient $k$-complexity algorithm

The $k$-complexity of any sequence could be obtained by repeated application of the Berlekamp-Massey algorithm [48] or, if appropriate, the Chan-Games algorithm [21], but this would be extremely inefficient. However, in the special case where $s$ is a binary sequence of length $2^n$, we have found an algorithm which computes the $k$-complexity of $(s)$ in an efficient manner. Our efficient $k$-complexity algorithm is Algorithm 4.3. When $k = 0$, this algorithm reduces to the Chan-Games algorithm and hence Algorithm 4.3 may be considered as a generalization of the Chan-Games algorithm.

Before proving the validity of Algorithm 4.3, we use the algorithm to find the 2-complexity of the sequence

$$(s) = (00011010100110101000101010011010).$$

The results are summarized in Figure 4.2. Note that the subscripts represent the cost[$i$]'s of Algorithm 4.3. The value of cost[$i$] is the minimum number of bit changes in $s$ needed to change $a_i$ to $\tilde{a}_i$, where

$$\tilde{a}_i = a_i + 1 \pmod 2.$$

Let $s \in \mathcal{S}_2(2^n)$. By the notation $a_i \mapsto \tilde{a}_i$ we mean the bit changes in $s$ which will transform $a_i$ into $\tilde{a}_i$. For example, if $s = s_0s_1s_2s_3$ and $a_0 = s_0 + s_2$, then $a_0 \mapsto \tilde{a}_0$ occurs if $s_0 \mapsto \tilde{s}_0$ or $s_2 \mapsto \tilde{s}_2$, i.e., if $s_0$ is replaced with $s_0 + 1$ or $s_2$ is replaced with $s_2 + 1$ (but not both).

The following lemma is obvious.

**Lemma 4.1** *Let $s \in \mathcal{S}_2(2^n)$, $L = L(s)$, $R = R(s)$, and $b = L + R$.*

*(i) If $a_i = R_i$ then $a_i \mapsto \tilde{a}_i$ implies $s_{i+2^n} \mapsto \tilde{s}_{i+2^n}$.*

*(ii) If $a_i = L_i$ then $a_i \mapsto \tilde{a}_i$ implies $s_i \mapsto \tilde{s}_i$.*

*(iii) If $a_i = b_i$ then $a_i \mapsto \tilde{a}_i$ implies either $s_i \mapsto \tilde{s}_i$ or $s_{i+2^n} \mapsto \tilde{s}_{i+2^n}$, but not both.*

Lemma 4.2 is only slightly more difficult than Lemma 4.1.

**Lemma 4.2** *Let $a^0 = s \in \mathcal{S}_2(2^n)$, $L = L(a^0)$, $R = R(a^0)$, and $b = L + R$. Let $a^1 = a^1_0a^1_1 \ldots a^1_{2^{n-1}-1}$, where each $a^1_i$ is either $b_i$, $L_i$, or $R_i$. Next, form*

```
a = s;  c = 0;  ℓ = 2^n;
cost[i] = 1, for i = 0, 1, ..., ℓ − 1;
while ℓ > 1 do
    ℓ = ℓ/2;  L = a_0 a_1 ··· a_{ℓ−1};  R = a_ℓ a_{ℓ+1} ··· a_{2ℓ−1};
    b = L + R;  t = Σ_{i=0}^{ℓ−1} b_i min(cost[i], cost[i + ℓ]);
    if t ≤ k then
        k = k − t;
        for i = 0, 1, ..., ℓ − 1 do
            if b_i = 1 then
                if cost[i] < cost[i + ℓ] then
                    L_i = R_i;  cost[i] = cost[i + ℓ] − cost[i];
                else
                    cost[i] = cost[i] − cost[i + ℓ];
                end if
            else
                cost[i] = cost[i] + cost[i + ℓ];
            end if
        a = L;
    else
        c = c + ℓ;
        a = b;
        cost[i] = min(cost[i], cost[i + ℓ]),  for i = 0, 1, ..., ℓ − 1;
    end if
end while
if a_0 = 1 and cost[0] > k then
    c = c + 1;
end if
```

Algorithm 4.3: An efficient $k$-complexity algorithm

*$a^2 \in S_2(2^{n-2})$ from $a^1$ in a similar manner, and so on. Then $a_i^j \mapsto \tilde{a}_i^j$ can only affect elements of the set*

$$B_i^j = \{s_\ell \in s \,|\, \ell \equiv i \pmod{2^{n-j}}\},$$

*where $i \in \{0, 1, \ldots, 2^{n-j} - 1\}$.*

PROOF: If $a_i^1 \mapsto \tilde{a}_i^1$, either $s_i \mapsto \tilde{s}_i$ or $s_{i+2^{n-1}} \mapsto \tilde{s}_{i+2^{n-1}}$ and hence the lemma holds in this case. Suppose the lemma holds for some $j \geq 1$. Then $a_i^{j+1} \mapsto \tilde{a}_i^{j+1}$

$(s) = (00011010100110101000101010011010),\ k = 2$

| | | | | |
|---|---|---|---|---|
| Step 1) | $\ell$ | = | $16,\ T = 2,\ k = 2$ | |
| | $L$ | = | $0_1 0_1 0_1 1_1 1_1 0_1 1_1 0_1 1_1 0_1 0_1 1_1 1_1 0_1 1_1 0_1$ | |
| | $R$ | = | $\underline{1_1 0_1 0_1 0_1 1_1 0_1 1_1 0_1 1_1 0_1 0_1 1_1 1_1 0_1 1_1 0_1}$ | |
| | $b$ | = | $1_1 0_1 0_1 1_1 0_1 0_1 0_1 0_1 0_1 0_1 0_1 0_1 0_1 0_1 0_1 0_1$ | $c = 0$ |
| | | | | $a = L$ |
| Step 2) | $\ell$ | = | $8,\ T = 0,\ k = 0$ | |
| | $L$ | = | $1_0 0_2 0_2 0_0 1_2 0_2 1_2 0_2$ | |
| | $R$ | = | $\underline{1_2 0_2 0_2 1_2 1_2 0_2 1_2 0_2}$ | |
| | $b$ | = | $0_0 0_2 0_2 1_0 0_2 0_2 0_2 0_2$ | $c = 0$ |
| | | | | $a = L$ |
| Step 3) | $\ell$ | = | $4,\ T = 6,\ k = 0$ | |
| | $L$ | = | $1_2 0_4 0_4 1_2$ | |
| | $R$ | = | $\underline{1_4 0_4 1_4 0_4}$ | |
| | $b$ | = | $0_2 0_4 1_4 1_2$ | $c = 4$ |
| | | | | $a = b$ |
| Step 4) | $\ell$ | = | $2,\ T = 6,\ k = 0$ | |
| | $L$ | = | $0_2 0_4$ | |
| | $R$ | = | $\underline{1_4 1_2}$ | |
| | $b$ | = | $1_2 1_4$ | $c = 6$ |
| | | | | $a = b$ |
| Step 5) | $\ell$ | = | $1,\ T = 0,\ k = 0$ | |
| | $L$ | = | $1_2$ | |
| | $R$ | = | $\underline{1_2}$ | |
| | $b$ | = | $0_2$ | $c = 6$ |
| | | | | $a = L$ |
| | $a_0$ | = | $1,\ \mathrm{cost}[0] = 4,\ k = 0$ | $c = 7$ |

Figure 4.2: Computing the 2-complexity

implies either $a_i^j \mapsto \tilde{a}_i^j$ or $a_{i+2^{n-j-1}}^j \mapsto \tilde{a}_{i+2^{n-j-1}}^j$. By the induction hypothesis, the affected bits of $s$ are contained in the set $B_i^j \cup B_{i+2^{n-j-1}}^j$. But $\ell \equiv i \pmod{2^{n-j}}$ implies $\ell - i = (2k)2^{n-j-1}$. In addition, $\ell \equiv i + 2^{n-j-1} \pmod{2^{n-j}}$ implies $\ell - i = (2k+1)2^{n-j-1}$ and hence

$$\begin{aligned} B_i^j \cup B_{i+2^{n-j-1}}^j &= \{s_\ell \in s \,|\, \ell \equiv i \pmod{2^{n-j}} \text{ or } \ell \equiv i + 2^{n-j-1} \pmod{2^{n-j}}\} \\ &= \{s_\ell \in s \,|\, \ell \equiv i \pmod{2^{n-j-1}}\} \end{aligned}$$

and the lemma follows by induction. ∎

We can now verify that our new algorithm—Algorithm 4.3—computes the $k$-complexity of periodic binary sequences with period $2^n$.

**Theorem 4.4** *Let $s \in S_2(2^n)$. Then for $k \in \{0, 1, \ldots, 2^n\}$, Algorithm 4.3 determines the $k$-complexity of $(s)$ in $n$ steps.*

PROOF: For $k = 0$, Algorithm 4.3 reduces to the Chan-Games algorithm. If $k > 0$, we are allowed to make $k$ (or fewer) bit changes in $s$ in order to reduce the linear complexity of $(s)$ as much as possible. But the complexity only increases when $b \neq \vec{0}$. If we can force $b = \vec{0}$ at the $j$th step we should do so, since the total of all remaining additions is at most $2^{n-j}$. This is the basic logic of the algorithm—if $b \neq \vec{0}$ and we can "afford" to make $b = \vec{0}$, we do so. The complication arises from calculating the net "cost" of changing a particular bit several steps into the algorithm.

If $b_i = 1$, changing either $L_i$ or $R_i$ will force $b_i = \vec{0}$ and hence the "cost" associated with changing $b_i$ is the minimum of the costs of changing $L_i$ and $R_i$. The problem thus reduces to showing:

1. The value cost[$i$] in Algorithm 4.3 gives the minimum number of bit changes in $s$ required in order that $a_i \mapsto \tilde{a}_i$ and

2. The values of the cost[$i$]'s are additive, i.e.,

$$\text{cost}[i_1] + \text{cost}[i_2] + \cdots + \text{cost}[i_m]$$

must give the number of changes in $s$ required in order that $a_{i_j} \mapsto \tilde{a}_{i_j}$ for $j = 1, 2, \ldots, m$.

The first condition can be verified by applying Lemma 4.1 recursively and comparing with the **for** loop in Algorithm 4.3.

In the $j$th step of Algorithm 4.3 the sets $\{B_i^j\}_{i=0}^{2^{n-j}-1}$ of Lemma 4.2 are disjoint. Hence, the second condition above is satisfied and the $k$-complexity algorithm is verified. ∎

## 4.6 Computational complexity

Computational complexity theory is of interest since we are attempting to find efficient algorithms for computing the $k$-complexity. An excellent reference on the subject is the book by Garey and Johnson [22].

Let $s$ be a bit-string of length $n$. As previously noted, the $k$-complexity of $s$ could be computed by repeatedly applying the Berlekamp-Massey algorithm. However, this approach would require

$$\binom{n}{0} + \binom{n}{1} + \cdots + \binom{n}{k}$$

applications of the basic algorithm. If $n$ is even and $k = n/2$, finding the $k$-complexity by this method would require $2^{n-1}$ applications of an $O(n^2)$ algorithm. It is thus conceivable that the general problem of finding the $k$-complexity is NP-complete. If such is the case, we must restrict our attention to approximate algorithms, which may exist in light of some results on continued fractions in [82]. Certain procedures in coding theory are known to be NP-complete; see the article by Berlekamp et al. [3].

The problem of finding the $k$-complexity can be stated in the form of a *decision problem* [22] as follows:

**$k$-COMPLEXITY**

INSTANCE: $s = s_0 s_1 \ldots s_{n-1} \in \mathcal{S}_2(n)$, integers $k$ and $B$ such that $0 \leq k \leq n$ and $0 \leq B \leq n$.

QUESTION: Is $c_k(s) < B$?

It is easily verified that this problem is in the class NP. However, we have thus far been unable to show that the $k$-complexity problem is NP-complete and we have also been unable to find an efficient algorithm for arbitrary $n$.

The following theorem may help to illuminate the underlying structure of the $k$-complexity problem.

**Theorem 4.5** *Let $\mathcal{F}$ be a field and $(s) = (s_0, s_1, \ldots, s_{n-1})$, with each $s_i \in \mathcal{F}$.*

*Let $H_s$ be the $n \times n$ Hankel matrix*

$$H_s = \begin{pmatrix} s_0 & s_1 & \cdots & s_{n-2} & s_{n-1} \\ s_1 & s_2 & \cdots & s_{n-1} & s_0 \\ \vdots & \vdots & & \vdots & \vdots \\ s_{n-1} & s_0 & \cdots & s_{n-3} & s_{n-2} \end{pmatrix}. \tag{4.7}$$

*Then $c_0(s) = \text{rank}(H_s)$, i.e., the linear complexity of $(s)$ is given by the rank of the matrix $H_s$.*

PROOF: Let $k = \text{rank}(H_s)$. If $k = n$ then the circulating shift register in Figure 3.3 produces $(s)$. If a smaller shift-register produced $(s)$, there would exist coefficients $a_i$ such that

$$a = (a_1, a_2, \cdots, a_j, 1, 0, \cdots, 0)^T$$

and $H_s a = 0$, contradicting the invertibility of $H_s$. Hence, in this case, $c_0(s) = n$.

If $k < n$ there exists an invertible $T$ such that

$$H_s T = \begin{pmatrix} x_{11} & \cdots & x_{1k} & 0 & \cdots & 0 \\ x_{21} & \cdots & x_{2k} & 0 & \cdots & 0 \\ \vdots & & \vdots & \vdots & & \vdots \\ x_{n1} & \cdots & x_{nk} & 0 & \cdots & 0 \end{pmatrix}.$$

Furthermore, the matrix $T$ is of the form

$$T = \begin{pmatrix} 1 & * & \cdots & \cdots & * \\ & \ddots & & & \vdots \\ & & 1 & * & \cdots & * \\ & & & 1 & \cdots & 0 \\ & & & & \ddots & \vdots \\ & & & & & 1 \end{pmatrix} = \begin{pmatrix} A & B \\ O & I \end{pmatrix},$$

where $A$ is $k \times k$ upper triangular, $B$ is $k \times (n-k)$, $O$ is the $(n-k) \times k$ zero matrix, and $I$ is the $(n-k) \times (n-k)$ identity matrix. Let $e_{k+1}$ be the $(k+1)$st standard basis vector and note that

$$H_s T e_{k+1} = H_s (T e_{k+1}) = \vec{0},$$

where

$$Te_{k+1} = (a_1, a_2, \cdots, a_k, 1, 0, \cdots, 0)^T.$$

for some $a_i$'s. It follows that

$$-(a_1 s_\ell + a_2 s_{\ell+1} + \cdots + a_k s_{\ell+k-1}) = s_{\ell+k} \quad \text{for all} \quad \ell \geq 0$$

and hence $c_0(s) \leq k$.

Suppose $c_0(s) < k$. Then there exists

$$b = (b_1, \cdots, b_j, 1, 0, \cdots, 0)^T,$$

with $j < k$ for which $H_s b = \vec{0}$. Letting $c = T^{-1}b$ we have

$$c = (c_1, \cdots, c_j, 1, 0, \cdots, 0)^T,$$

for some $c_i$'s. But then $H_s b = H_s T c = \vec{0}$ and hence

$$c_1 \begin{pmatrix} x_{11} \\ \vdots \\ x_{n1} \end{pmatrix} + \cdots + c_j \begin{pmatrix} x_{1j} \\ \vdots \\ x_{nj} \end{pmatrix} + \begin{pmatrix} x_{1,j+1} \\ \vdots \\ x_{n,j+1} \end{pmatrix} = \vec{0},$$

i.e., the columns of $H_s T$ are linearly dependent, contradicting the fact that $\text{rank}(H_s) = k$. Thus $c_0(s) = k$. ∎

Let $s = s_0 s_1 \cdots s_{n-1} \in \mathcal{S}_2(n)$, let $H_s$ be the Hankel matrix in (4.7), and for each $t \in \mathcal{S}_2(n)$ let

$$d(s,t) = \sum_{i=0}^{n-1} |s_i - t_i|.$$

For each $k \in \{0, 1, \ldots, n\}$ let

$$A_k(s) = \{t \in \{0,1\}^n \mid d(s,t) \leq k\}.$$

It follows from Theorem 4.5 that the $k$-complexity of $s$, denoted $c_k(s)$, is given by

$$c_k(s) = \min_{t \in A_k(s)} (\text{rank}(H_t)).$$

In this chapter we considered the Berlekamp-Massey algorithm and its connection to the continued fraction algorithm. The Chan-Games algorithm was

also discussed and a new proof of its validity given. The primary result of this chapter is an efficient new $k$-complexity algorithm for the special case where the sequence is a periodic binary sequence with period $2^n$. This new algorithm was proved and an example of its application given. The computational complexity of the general case was also considered.

# CHAPTER V
# BINARY SEQUENCES WITH PERIOD $2^n$

## 5.1 Introduction

In this chapter, we analyze the $k$-complexities of particular classes of binary sequences with period $2^n$. In this case an efficient algorithm—namely, Algorithm 4.3—exists for computing the $k$-complexity. Of particular interest are de Bruijn sequences. De Bruijn sequences have been intensively studied in the literature and various results on their linear complexities are known. In addition, we consider a new class of binary sequences with period $2^n$. Computational results indicate that this new class of sequences has good $k$-complexity properties.

## 5.2 de Bruijn sequences

Suppose $n$ terminals are connected to a digital computer and that each terminal is capable of transmitting "0" or "1." Can a switch be constructed so that each of the $2^n$ possible binary $n$-tuples occurs? One particularly elegant solution is given—for the case $n = 4$—in Figure 5.1. As the dial in Figure 5.1 is rotated, each binary 4-tuple appears exactly once in a complete rotation. Such a switch could be used, for example, to set a washing machine to any of 16 distinct settings. This construction is the "Eulerian washing machine" discussed by Street in [79]. It will become clear below why the epithet "Eulerian" is attached to this device.

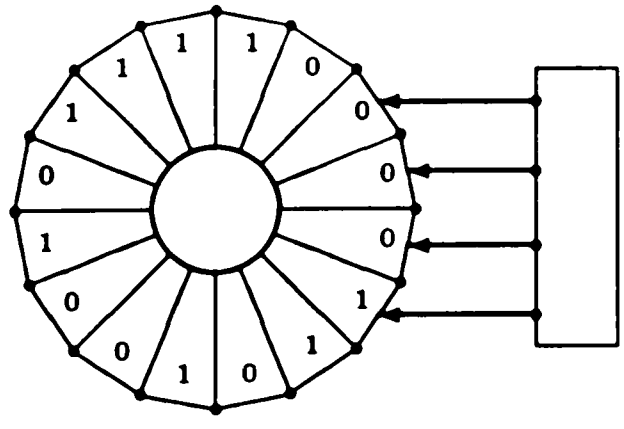


Figure 5.1: An Eulerian washing machine

A sequence of length $2^n$ in which all distinct $n$-tuples appear—provided the elements are taken cyclically—is referred to as a *de Bruijn sequence* of span $n$. The binary sequence in Figure 5.1, namely,

$$(0,0,0,0,1,1,0,1,0,0,1,0,1,1,1,1)$$

is a de Bruijn sequence of span 4.

According to Stein [78], the Sanskrit "memory word"

*yamátárájabhánasalagám*

has been employed for nearly 1000 years as a mnemonic device for remembering certain rhythms. By letting "0" stand for an unaccented syllable and "1" for an accented syllable, the sequence

$$0,1,1,1,0,1,0,0,0,1 \tag{5.1}$$

is obtained, which is a de Bruijn sequence of span three with the first two digits repeated at the end (in this case we do not take the digits cyclically).

The first modern account of de Bruijn sequences was apparently due to Flye Sainte-Marie [67]—in response to a problem posed by de Rivière [64]—and appeared in 1894, although Stein [78] mentions Émile Baudot with a date of 1882 but gives no citation. In 1895 Mantel [43] showed that "full cycles" (i.e., de Bruijn sequences) can be obtained using primitive polynomials—a result which was later rediscovered by Rees [62]. In his 1934 paper [46], M. H. Martin solved the following problem:

> Given $n$ distinct symbols and $r$ a positive integer, does there exist a sequence of these $n$ symbols such that each of the $n^r$ permutations (repeats allowed) occurs exactly once as a subsequence of $r$ consecutive symbols?

An algorithm was given for constructing such a sequence for any $n$ and $r$. The algorithm is essentially the "prefer zeros" algorithm [20]. The same year, Popper [60, pp. 162–163, p. 292] considered de Bruijn sequences in the context of probability theory.

In 1946, Good [25] and de Bruijn [6] independently published results proving the existence of binary de Bruijn sequences of span $n$ for every $n$. De Bruijn

also showed that there are $2^{2^{n-1}-n}$ such sequences. Flye Sainte-Marie [67] had derived this formula in his 1894 article but Sainte-Marie's work was unknown to de Bruijn in 1946 [7]. The problem of constructing de Bruijn sequences was rediscovered yet again in 1950 when it appeared in the problem section of the *American Mathematical Monthly* [38,81]. Recent publications tend to trace their origin to de Bruijn or Good, and hence the nomenclature "de Bruijn sequence."

Good [25] gives the following proof of the existence of a binary de Bruijn sequence of span $n$. First, let the de Bruijn graph (or Good diagram) $G_n$ be a directed graph on $2^n$ vertices, where each vertex is labeled with a distinct binary $n$-tuple. A directed edge is drawn from vertex $(a_1, a_2, \ldots, a_n)$ to vertex $(b_1, b_2, \ldots, b_n)$ if and only if $b_k = a_{k+1}$ for $k = 1, 2, \ldots, n-1$. The de Bruijn graphs $G_2$ and $G_3$ are illustrated in Figures 5.2 and 5.3, respectively.

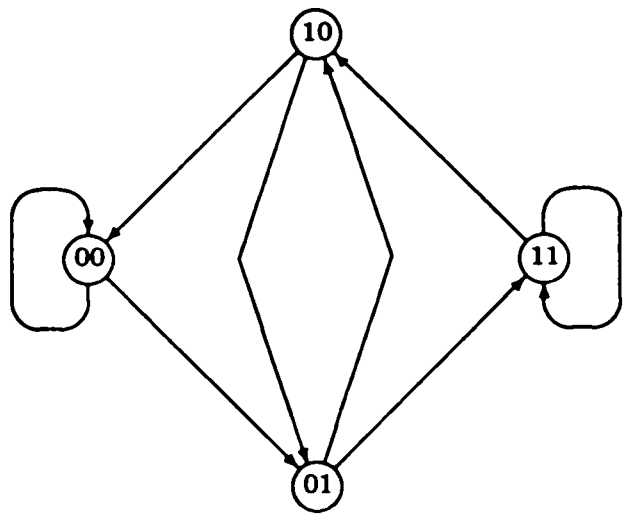


Figure 5.2: The de Bruijn graph $G_2$

Now to obtain a de Bruijn sequence of span $n$, construct the de Bruijn graph $G_{n-1}$ and label the edge from $(x_1 x_2 \cdots x_{n-1})$ to $(x_2 x_3 \cdots x_{n-1} a)$ with the binary $n$-tuple $(x_1 x_2 \cdots x_{n-1} a)$. The resulting graph has $2^n$ edges labeled with the $2^n$ distinct $n$-tuples. Denote the *in-degree*—i.e., the number of directed edges entering a node—by $d_{\text{in}}(\vec{x})$ and the *out-degree* (defined analogously) by $d_{\text{out}}(\vec{x})$. The de Bruijn graph evidently satisfies $d_{\text{in}}(\vec{x}) = d_{\text{out}}(\vec{x}) = 2$ for each node $\vec{x}$, and an elementary result in graph theory implies that $G_n$ has an Euler circuit [15]—i.e., a closed path which traverses each edge exactly once. Since consecutive edges of such an Euler circuit are of the form $x_1 x_2 \cdots x_n$ and $x_2 x_3 \cdots x_{n+1}$, it is apparent

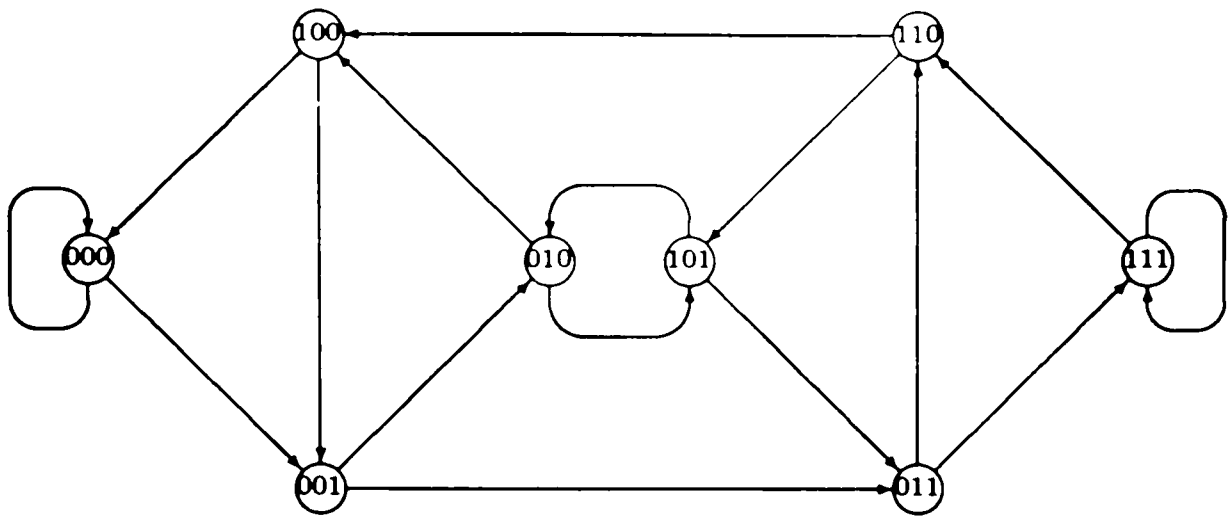


Figure 5.3: The de Bruijn graph $G_3$

that there exists a de Bruijn sequence of span $n$. Furthermore, each distinct Euler circuit gives a distinct de Bruijn sequence. For example, the graph in Figure 5.4 contains the Euler circuit

$$011, 111, 110, 101, 010, 100, 000, 001.$$

The resulting de Bruijn sequence, 01110100, appears in (5.1).

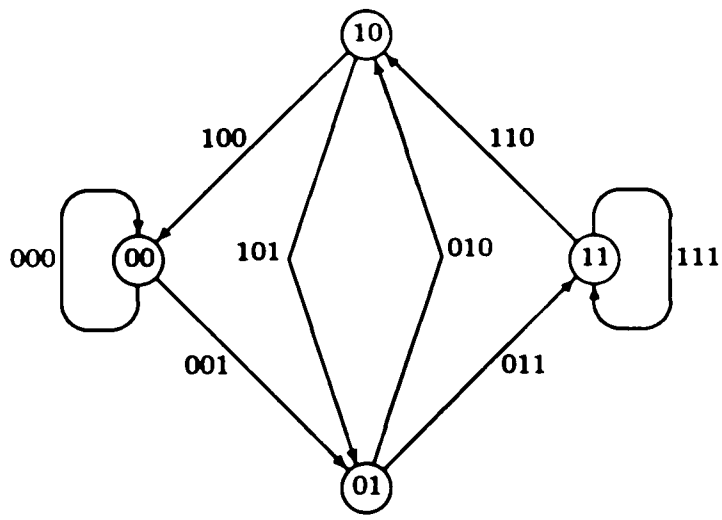


Figure 5.4: Edge-labeled de Bruijn graph $G_2$

Any de Bruijn sequence of span $n$ can be produced by a nonlinear recurrence in $n$ variables [20]. However, the linear complexities of de Bruijn sequences are high. Many interesting results concerning the linear complexities of these

sequences are given by Chan et al. [10]. An analysis of the $k$-complexities of de Bruijn sequences is considered in Section 5.3.

Van Aardenne-Ehrenfest and de Bruijn [1] generalized de Bruijn's earlier results to the case where the underlying set contains $k$ elements. A very readable exposition of the general case is given by van Lint [41]. In addition, there are many algorithms for obtaining some subset of the de Bruijn sequences of span $n$. Fredricksen [20] gives an encyclopedic collection of such algorithms along with the relevant theory and the papers by Etzion and Lempel [18] and Siu and Tong [74] contains some additional algorithms, while Lempel [37] presents a useful algebraic theory. De Bruijn sequences continue to be of considerable research interest; see, for example, the recent articles [31] and [50].

There are other classes of sequences whose linear complexities are known, such as GMW sequences [68] and the more general No sequences [58]. Determining the $k$-complexities of these various classes of sequences would be interesting. Unfortunately, GMW and No sequences have period $2^n-1$ and hence our modified version of the Chan-Games algorithm is not applicable.

## 5.3 The $k$-complexities of de Bruijn sequences

For $s \in \mathcal{S}_2(n)$ define $\omega(s)$ to be the number of nonzero elements of $s$. Recall that $c_k(s)$ is the smallest linear complexity that can be obtained by altering any $k$ or fewer elements of $s$. Let $\hat{c}_k(s)$ to be the minimum linear complexity that can be obtained when exactly $k$ elements of $s$ are changed. Lemma 5.1 follows immediately from the definitions.

**Lemma 5.1** *Let $s \in \mathcal{S}_2(n)$. Then*

*(i)* $c_k(s) \leq n$ *for* $k = 0, 1, \ldots, n$.

*(ii)* $c_{k+1}(s) \leq c_k(s)$ *for* $k = 0, 1, \ldots, n$.

*(iii)* $c_k(s) \leq 1$ *for* $k \geq \lfloor n/2 \rfloor$.

*(iv)* $c_k(s) = 0$ *for* $k = \omega(s)$.

Note that only *(i)* and *(iv)* of Lemma 5.1 hold if $c_k(s)$ is replaced by $\hat{c}_k(s)$.

Let $D(n)$ be the collection of binary de Bruijn sequence of span $n$. For $s \in D(n)$ it is well known that $\omega(s) = 2^{n-1}$. In Chan et al. [10, Theorem 3] it

is shown that for $s \in \mathcal{S}_2(2^n)$ we have $c_0(s) = 2^n$ if and only if $\omega(s)$ is odd. The next lemma is a trivial consequences of this fact.

**Lemma 5.2** *For $s \in D(n)$ we have $\hat{c}_{2k+1}(s) = 2^n$ for $k = 0, 1, \ldots, 2^{n-1} - 1$.*

The following theorem—which follows immediately from Lemma 5.2—states that when computing the $k$-complexity profile of a de Bruijn sequence, we only need to compute

$$c_0(s), c_2(s), c_4(s), \ldots, c_{n/2-2}(s).$$

**Theorem 5.1** *Let $s \in D(n)$ for some $n \geq 2$ and suppose*

$$\begin{aligned} c_0(s) = c_1(s) = \cdots = c_{m_1-1}(s) > c_{m_1}(s) = \cdots = c_{m_2-1}(s) \\ > \cdots > c_{m_{j-1}}(s) = \cdots = c_{m_j-1}(s) > c_{m_j}(s) = 0, \end{aligned}$$

*where $m_j = 2^{n-1}$. Then $m_i \equiv 0 \pmod 2$ for $i = 1, 2, \ldots, j$.*

For $s = s_0 s_1 \ldots s_{n-1} \in \mathcal{S}_2(n)$ let

$$r(s) = s_{n-1} s_{n-2} \ldots s_0$$

and

$$\tilde{s} = \tilde{s}_0 \tilde{s}_1 \ldots \tilde{s}_{n-1},$$

where

$$\tilde{s}_i = s_i + 1 \pmod 2.$$

Recall that a de Bruijn sequence $s$ of span $n$ can be generated by a nonlinear recurrence in $n$ variables. Since such a function could be used in a nonlinear feedback shift register to generate $s$, we refer to this function as the *feedback function* of $s$.

**Theorem 5.2** *The feedback function of a de Bruijn sequence $s$ of span $n \geq 3$ has the form*

$$f_s(x_1, x_2, \ldots, x_n) = 1 + x_1 + h(x_2, x_3, \ldots, x_n) + x_2 x_3 \cdots x_n \qquad (5.2)$$

*where* $h(\vec{0}) = 0$, *and if* $h$ *is written in the form*

$$\begin{aligned} h(x_2, \ldots, x_n) = {} & a_2x_2 + a_3x_3 \cdots + a_nx_n \\ & + a_{2,3}x_2x_3 + \cdots + a_{n-1,n}x_{n-1}x_n \\ & \vdots \\ & + a_{2,3,\ldots,n}x_2x_3 \cdots x_n, \end{aligned}$$

*then* $a_{2,3,\ldots,n} = 0$ *and* $h$ *consists of an odd number of nonzero terms. Also, the feedback function for* $r(s)$ *is*

$$f_{r(s)}(x_1, x_2, \ldots, x_n) = 1 + x_1 + h(x_n, x_{n-1}, \ldots, x_2) + x_2x_3 \cdots x_n \tag{5.3}$$

*and for* $\tilde{s}$ *is*

$$f_{\tilde{s}}(x_1, x_2, \ldots, x_n) = 1 + x_1 + h(\tilde{x}_2, \tilde{x}_3, \ldots, \tilde{x}_n) + \tilde{x}_2\tilde{x}_3 \cdots \tilde{x}_n. \tag{5.4}$$

PROOF: The formula (5.2) is given as a theorem by Fredricksen [20, Section 4]. By (5.2) we have

$$f_s(x_1, x_2, \ldots, x_n) = 1 + x_1 + h(x_2, \ldots, x_n) + x_2x_3 \cdots x_n$$

and

$$f_{r(s)}(x_1, x_2, \ldots, x_n) = 1 + x_1 + g(x_2, \ldots, x_n) + x_2x_3 \cdots x_n$$

for some $h$ and $g$ satisfying the conditions of the theorem. Now if

$$f_s(x_1, x_2, \ldots, x_n) = y$$

then we must have

$$f_{r(s)}(y, x_n, \ldots, x_2) = x_1,$$

and hence

$$1 + x_1 + h(x_2, \ldots, x_n) = y \quad \text{and} \quad 1 + y + g(x_n, \ldots, x_2) = x_1.$$

It follows that $h(x_n, \ldots, x_2) = g(x_2, \ldots, x_n)$ and (5.3) is established.

A similar argument can be used to verify (5.4). ∎

Theorem 5.2 gives a method for finding all de Bruijn sequences of span $n$ for small values of $n$. Note, however, that there are $2^{2^{n-1}-3}$ functions of the form (5.2), but only $2^{2^{n-1}-n}$ distinct de Bruijn sequences of span $n$.

The $k$-complexities of de Bruijn sequences of span $n = 1, 2, 3, 4, 5$ appear in Tables 5.2 through 5.6 (pp. 53–55), respectively. The de Bruijn sequences were found using the method implied by Theorem 5.2, i.e., all $2^{2^{n-1}-3}$ nonlinear feedback functions of the form (5.2) were generated and the resulting output sequences were tested for the de Bruijn property.

The computational results in Tables 5.2 through 5.6 suggest the following conjectures.

**Conjecture 5.1** *Let $s \in D(n)$ for some $n$. Then the $k$-complexity profiles of $s$, $\tilde{s}$, $r(s)$, and $r(\tilde{s})$ are identical.*

**Conjecture 5.2** *Let $s$ be a de Bruijn sequence of span $n \geq 3$ and let $a_k = c_k(s)$, for $k = 0, 2, 4, \ldots, n/2$. Then the number of de Bruijn sequences of span $n$ with $k$-complexity profile $\{a_0, a_2, a_4, \ldots, a_{n/2}\}$ is even.*

The de Bruijn sequences are generally considered "complex" and their linear complexities have been studied in detail [10]. The $k$-complexity profiles of these sequences more closely follow the "stair step" pattern in Figure 3.6 than the poor $k$-complexity profile in Figure 3.7. This indicates that there exists a large class of sequences with good $k$-complexity properties. The results in this section also establish a baseline for comparing the $k$-complexities of various other classes of sequences.

## 5.4 A class of binary sequences

The binomial coefficient $\binom{n}{k}$ can be interpreted as the number of distinct binary sequences of length $n$ which contain exactly $k$ ones. Let $\left\langle {n \atop k} \right\rangle$ be the number of "shift-distinct" binary sequences of length $n$ with $k$ ones, i.e., the number of sequences which are distinct modulo circular rotations. For example, any four-bit sequence can be obtained via a circular shift of 0000, 1000, 1100, 1010, 1110 or 1111 and hence there are only 6 shift-distinct sequences of length four. In particular, $\left\langle {4 \atop 2} \right\rangle = 2$. It is also apparent that $\left\langle {n \atop 0} \right\rangle = \left\langle {n \atop 1} \right\rangle = \left\langle {n \atop n-1} \right\rangle = \left\langle {n \atop n} \right\rangle = 1$

for all $n$. Computing $\left\langle {n \atop k} \right\rangle$ for $n = 1, 2, \ldots, 12$ and $k = 0, 1, \ldots, n$ results in the "triangle" of Table 5.1.

Table 5.1: A variant of Pascal's triangle

| $n$ | $\left\langle {n \atop 0} \right\rangle$ | $\left\langle {n \atop 1} \right\rangle$ | $\left\langle {n \atop 2} \right\rangle$ | $\left\langle {n \atop 3} \right\rangle$ | $\left\langle {n \atop 4} \right\rangle$ | $\left\langle {n \atop 5} \right\rangle$ | $\left\langle {n \atop 6} \right\rangle$ | $\left\langle {n \atop 7} \right\rangle$ | $\left\langle {n \atop 8} \right\rangle$ | $\left\langle {n \atop 9} \right\rangle$ | $\left\langle {n \atop 10} \right\rangle$ | $\left\langle {n \atop 11} \right\rangle$ | $\left\langle {n \atop 12} \right\rangle$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | | | | | | | | | | | |
| 2 | 1 | 1 | 1 | | | | | | | | | | |
| 3 | 1 | 1 | 1 | 1 | | | | | | | | | |
| 4 | 1 | 1 | 2 | 1 | 1 | | | | | | | | |
| 5 | 1 | 1 | 2 | 2 | 1 | 1 | | | | | | | |
| 6 | 1 | 1 | 3 | 4 | 3 | 1 | 1 | | | | | | |
| 7 | 1 | 1 | 3 | 5 | 5 | 3 | 1 | 1 | | | | | |
| 8 | 1 | 1 | 4 | 7 | 10 | 7 | 4 | 1 | 1 | | | | |
| 9 | 1 | 1 | 4 | 10 | 14 | 14 | 10 | 4 | 1 | 1 | | | |
| 10 | 1 | 1 | 5 | 12 | 22 | 26 | 22 | 12 | 5 | 1 | 1 | | |
| 11 | 1 | 1 | 5 | 15 | 30 | 42 | 42 | 30 | 15 | 5 | 1 | 1 | |
| 12 | 1 | 1 | 6 | 19 | 43 | 66 | 80 | 66 | 43 | 19 | 6 | 1 | 1 |

Parts (c) and (d) of the following problem are considered by Stamp in [76]:

(a) Find a formula for $\left\langle {n \atop k} \right\rangle$ in terms of $\left\langle {m \atop i} \right\rangle$'s with $m < n$ and/or other known quantities.

(b) Find a "sharp" lower bound for $\left\langle {n \atop k} \right\rangle$.

(c) Give a formula for $\left\langle {2^n \atop 2^{n-1}} \right\rangle$ in terms of binomial coefficients.

(d) Prove or disprove that $\left\langle {n \atop k} \right\rangle \leq \binom{n}{k}/(n-1)$ for $n > 1$ and $0 < k < n$.

The solutions to parts (a) through (c) are straightforward.

(a) If the sequence $s = s_0 s_1 \cdots s_{n-1}$ has minimum period length $\ell \leq n$ then there are $\ell$ distinct circular shifts of $s$. For example, if $s$ has $k$ ones and

$\gcd(n,k) = 1$ then $s$ can have no subperiods and hence

$$\left\langle {n \atop k} \right\rangle = \frac{1}{n}\binom{n}{k}. \tag{5.5}$$

On the other hand, if $n$ and $k$ are not relatively prime we must determine the number of sequences with periods which divide $\gcd(n,k)$. This suggests the following approach.

As noted above, $\left\langle {n \atop 0} \right\rangle = \left\langle {n \atop n} \right\rangle = 1$. To find $\left\langle {n \atop k} \right\rangle$ for $0 < k < n$ we first compute the greatest common divisor of $n$ and $k$. If $\gcd(n,k) = 1$ then (5.5) applies and we are finished. If not, let $\{c_i\}_{i=1}^{m}$ be the distinct, non-unity factors of $\gcd(n,k)$ and let $n_i = n/c_i$ and $k_i = k/c_i$ for $i = 1, 2, \ldots, m$. Then the recursive formula

$$\begin{aligned}\left\langle {n \atop k} \right\rangle &= \frac{1}{n}\left\{\binom{n}{k} - \sum_{i=1}^{m} n_i T(n_i,k_i)\right\} + \sum_{i=1}^{m} T(n_i,k_i) \\ &= \frac{1}{n}\left\{\binom{n}{k} + \sum_{i=1}^{m}(n-n_i)T(n_i,k_i)\right\}\end{aligned} \tag{5.6}$$

results, where, for $n$ and $k$ as above,

$$T(n,k) = \left\langle {n \atop k} \right\rangle - \sum_{i=1}^{m} T(n_i,k_i).$$

Note that if $\gcd(n,k) = 1$ then $T(n,k) = \left\langle {n \atop k} \right\rangle = \frac{1}{n}\binom{n}{k}$ and hence the recursion is well-defined. Note also that if $n$ and $k$ are relatively prime, (5.6) reduces to (5.5). Thus, for example,

$$\left\langle {9 \atop 4} \right\rangle = \frac{1}{9}\binom{9}{4} = 14.$$

However, if $\gcd(n,k) \neq 1$, the formula depends only on $\binom{n}{k}$ and $\left\langle {m \atop i} \right\rangle$'s with $m < n$. For example,

$$\begin{aligned}\left\langle {12 \atop 6} \right\rangle &= \frac{1}{12}\left\{\binom{12}{6} + 6T(6,3) + 8T(4,2) + 10T(2,1)\right\} \\ &= \frac{1}{12}\left\{\binom{12}{6} + 6\left(\left\langle {6 \atop 3} \right\rangle - \left\langle {2 \atop 1} \right\rangle\right) + 8\left(\left\langle {4 \atop 2} \right\rangle - \left\langle {2 \atop 1} \right\rangle\right) + 10\left\langle {2 \atop 1} \right\rangle\right\} \\ &= 80.\end{aligned}$$

(b) A simple lower bound for $\left\langle {n \atop k} \right\rangle$ is obtained from (5.6) by noting that each term $(n - n_i)T(n_i, k_i) \geq 0$, and hence

$$\left\langle {n \atop k} \right\rangle \geq \frac{1}{n}\binom{n}{k}.$$

This lower bound is "sharp" in the sense that for each value of $n$ there exists a $k$ for which the bound is attained.

(c) Applying (5.6) to $\left\langle {2 \atop 1} \right\rangle$, $\left\langle {4 \atop 2} \right\rangle$, $\left\langle {8 \atop 4} \right\rangle$, etc., a pattern quickly emerges. It is easily verified that

$$\left\langle {2^n \atop 2^{n-1}} \right\rangle = \frac{1}{2^n}\left\{\binom{2^n}{2^{n-1}} + \sum_{k=1}^{n-1} 2^{k-1}\binom{2^{n-k}}{2^{n-k-1}}\right\}. \tag{5.7}$$

The sum in (5.7) does not appear in [26] and we have thus far been unable to find a closed form expression using any of the various summation techniques discussed in [28].

Part (d) is more challenging. We do not have a proof, but the inequality has been verified for all $n \leq 16$.

It follows easily from (5.7) that

$$\left\langle {2^n \atop 2^{n-1}} \right\rangle - \left\langle {2^{n-1} \atop 2^{n-2}} \right\rangle = \frac{1}{2^n}\left\{\binom{2^n}{2^{n-1}} - \binom{2^{n-1}}{2^{n-2}}\right\}. \tag{5.8}$$

The quantity in (5.8) can be interpreted as the number of shift-distinct binary sequences of length $2^n$ with $2^{n-1}$ ones and no subperiods. We refer to such a sequence as an *S-sequence* of span $n$. The $S$-sequences are perhaps the largest class of sequences of length $2^n$ which have balanced statistics (i.e., $2^{n-1}$ ones and $2^{n-1}$ zeros) and high linear complexities. Note also that the de Bruijn sequences of span $n$ are a subset of the set of $S$-sequences of span $n$ and, furthermore, the $S$-sequences of span 1 and 2 are identical to the de Bruijn sequences of span 1 and 2. In Tables 5.7 and 5.8 (pp. 55 and 56), we give the $k$-complexities of $S$-sequences for $n = 3$ and $n = 4$, respectively. We list the linear complexities of $S$-sequences of span $n = 5$ in Table 5.9 (p. 57).

Recall that $(a|b)$ denotes the concatenation of $a$ and $b$. Theorem 5.3 is due to Chan, Games, and Key [10].

**Theorem 5.3** *[10, Theorem 2] Let $s \in \mathcal{S}_2(2^n)$. Then $c_0(s) = 2^{n-1} + 1$ if and only if $(s) = (r|\tilde{r})$, for some $r \in \mathcal{S}_2(2^{n-1})$.*

We prove the following result as a corollary to Theorem 5.3.

**Corollary 5.1** *The linear complexity of an $S$-sequence of span $n$ must satisfy $c_0(s) \geq 2^{n-1} + 1$. In addition, $2^{2^{n-1}-n}$ $S$-sequences of span $n$ satisfy $c_0(s) = 2^{n-1} + 1$.*

PROOF: Suppose $s \in \mathcal{S}_2(2^n)$ and $c = c_0(s) \leq 2^{n-1}$. Then

$$(E^{2^{n-1}} - 1)s_i = (E-1)^{2^{n-1}-c}(E-1)^c s_i = 0$$

which implies $s$ has period $2^{n-1}$ and hence $s$ is not an $S$-sequence of span $n$. The first statement in the corollary is thus proved.

If $s \in \mathcal{S}_2(2^n)$ satisfies $c_0(s) = 2^{n-1} + 1$, then by Theorem 5.3 we have $(s) = (r|\tilde{r})$. There are $2^{n-1}$ such sequences and each is half 0's and half 1's. It is apparent that such a sequence can have no subperiods. Also, given such a sequence $s$ it follows that each of its $2^n$ cyclic shifts is another such sequence. Hence, modulo cyclic shifts there are $2^{2^{n-1}}/2^n = 2^{2^{n-1}-n}$ such sequences and the result is proved. ∎

It is a curious fact that the number de Bruijn sequences of span $n$ is the same as the number of $S$-sequences of span $n$ with minimal linear complexity.

Tables 5.7 and 5.8 suggest the following conjecture.

**Conjecture 5.3** *Let $(s)$ be an $S$-sequence of span $n \geq 3$ and let $a_k = c_k(s)$ for $k = 0, 2, 4, \cdots, n/2$. Then the number of $S$-sequences of span $n$ with $k$-complexity profile $\{a_0, a_2, a_4, \ldots, a_{n/2}\}$ is $2^m$ for some $m \geq 1$.*

The computational results on $S$-sequences indicate that this class of sequences has $k$-complexity properties similar to the de Bruijn sequences, i.e., the $k$-complexity does not drop precipitously for relatively small values of $k$. However, the linear complexities of $S$-sequences are somewhat lower than those of de Bruijn sequences.

In this chapter, various results related to the $k$-complexity of binary de Bruijn sequences were given. A new class of binary sequences—the $S$-sequences—which have period $2^n$ were also discussed and several conjectures and various computational results were presented.

Table 5.2: $k$-complexities of de Bruijn sequences: $n = 1$

| sequence | 0-complexity | 1-complexity |
|---|---|---|
| 10 | 2 | 0 |

Table 5.3: $k$-complexities of de Bruijn sequences: $n = 2$

| sequence | 0-complexity | 2-complexity |
|---|---|---|
| 1100 | 3 | 0 |

Table 5.4: $k$-complexities of de Bruijn sequences: $n = 3$

| | $k$-complexity | | |
|---|---|---|---|
| sequence | $k = 0$ | $k = 2$ | $k = 4$ |
| 11101000 | 7 | 2 | 0 |
| 11100010 | 7 | 2 | 0 |

Table 5.5: $k$-complexities of de Bruijn sequences: $n = 4$

| | $k$-complexity | | | | |
|---|---|---|---|---|---|
| sequence | $k = 0$ | $k = 2$ | $k = 4$ | $k = 6$ | $k = 8$ |
| 1111000010011010 | 15 | 10 | 5 | 2 | 0 |
| 1111000010100110 | 15 | 10 | 5 | 2 | 0 |
| 1111011001010000 | 15 | 10 | 5 | 2 | 0 |
| 1111010110010000 | 15 | 10 | 5 | 2 | 0 |
| 1111010110000100 | 15 | 10 | 5 | 2 | 0 |
| 1111011000010100 | 15 | 10 | 5 | 2 | 0 |
| 1111001010000110 | 15 | 10 | 5 | 2 | 0 |
| 1111001000011010 | 15 | 10 | 5 | 2 | 0 |
| 1111001101000010 | 14 | 11 | 5 | 3 | 0 |
| 1111010011000010 | 14 | 11 | 5 | 3 | 0 |
| 1111010000101100 | 14 | 11 | 5 | 3 | 0 |
| 1111010000110010 | 14 | 11 | 5 | 3 | 0 |
| 1111010010110000 | 12 | 5 | 5 | 5 | 0 |
| 1111000011010010 | 12 | 5 | 5 | 5 | 0 |
| 1111000010110100 | 12 | 5 | 5 | 5 | 0 |
| 1111001011010000 | 12 | 5 | 5 | 5 | 0 |

Table 5.6: $k$-complexities of de Bruijn sequences: $n = 5$

| $c_0(s)$ | number | distinct $k$-complexity profiles |
|---|---|---|
| 31 | 1024 | 104 |
| 30 | 448 | 47 |
| 29 | 224 | 18 |
| 28 | 180 | 27 |
| 27 | 64 | 12 |
| 26 | 36 | 5 |
| 25 | 32 | 1 |
| 24 | 20 | 3 |
| 23 | 12 | 2 |
| 22 | 0 | 0 |
| 21 | 8 | 1 |
| total: | 2048 | 220 |

Table 5.7: $k$-complexities of $S$-sequences: $n = 3$

| | $k$-complexity | | |
|---|---|---|---|
| sequence | $k = 0$ | $k = 2$ | $k = 4$ |
| 11101000 | 7 | 2 | 0 |
| 11100010 | 7 | 2 | 0 |
| 11010100 | 7 | 2 | 0 |
| 11001010 | 7 | 2 | 0 |
| 11011000 | 6 | 3 | 0 |
| 11100100 | 6 | 3 | 0 |
| 11010010 | 5 | 5 | 0 |
| 11110000 | 5 | 5 | 0 |

Table 5.8: Distinct $k$-complexity profiles of $S$-sequences: $n = 4$

| $k$-complexity | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| $k = 0$ | $k = 2$ | $k = 4$ | $k = 6$ | $k = 8$ | $k = 10$ | $k = 12$ | $k = 14$ | number |
| 15 | 15 | 9 | 9 | 9 | 9 | 2 | 2 | 128 |
| 15 | 15 | 10 | 10 | 5 | 5 | 2 | 2 | 128 |
| 15 | 15 | 10 | 10 | 3 | 3 | 2 | 2 | 64 |
| 15 | 15 | 5 | 5 | 5 | 5 | 2 | 2 | 32 |
| 15 | 15 | 6 | 6 | 3 | 3 | 2 | 2 | 32 |
| 15 | 15 | 3 | 3 | 3 | 3 | 2 | 2 | 8 |
| 15 | 15 | 2 | 2 | 2 | 2 | 2 | 2 | 8 |
| 14 | 14 | 9 | 9 | 9 | 9 | 3 | 3 | 64 |
| 14 | 14 | 11 | 11 | 5 | 5 | 3 | 3 | 64 |
| 14 | 14 | 11 | 11 | 2 | 2 | 2 | 2 | 32 |
| 14 | 14 | 5 | 5 | 5 | 5 | 3 | 3 | 16 |
| 14 | 14 | 7 | 7 | 2 | 2 | 2 | 2 | 16 |
| 14 | 14 | 3 | 3 | 3 | 3 | 3 | 3 | 8 |
| 13 | 13 | 13 | 13 | 3 | 3 | 3 | 3 | 64 |
| 13 | 13 | 13 | 13 | 2 | 2 | 2 | 2 | 32 |
| 12 | 12 | 9 | 9 | 9 | 9 | 5 | 5 | 32 |
| 12 | 12 | 5 | 5 | 5 | 5 | 5 | 5 | 8 |
| 12 | 12 | 7 | 7 | 2 | 2 | 2 | 2 | 8 |
| 12 | 12 | 6 | 6 | 3 | 3 | 3 | 3 | 4 |
| 11 | 11 | 11 | 11 | 5 | 5 | 5 | 5 | 16 |
| 11 | 11 | 11 | 11 | 2 | 2 | 2 | 2 | 8 |
| 10 | 10 | 10 | 10 | 5 | 5 | 5 | 5 | 8 |
| 10 | 10 | 10 | 10 | 3 | 3 | 3 | 3 | 4 |
| 9 | 9 | 9 | 9 | 9 | 9 | 9 | 9 | 16 |
| | | | | | | | Total: | 800 |

Table 5.9: Linear complexities of $S$-sequences: $n = 5$

| $c_0(s)$ | number |
|---|---|
| 31 | 9,391,680 |
| 30 | 4,695,840 |
| 29 | 2,346,496 |
| 28 | 1,174,672 |
| 27 | 586,624 |
| 26 | 293,312 |
| 25 | 143,360 |
| 24 | 75,688 |
| 23 | 37,600 |
| 22 | 18,800 |
| 21 | 8960 |
| 20 | 4920 |
| 19 | 2240 |
| 18 | 1120 |
| 17 | 2048 |
| total: | 18,783,360 |

# CHAPTER VI
# CONCLUSIONS

Chapter II contained the terminology and background from cryptology needed in the remainder of the paper. We also mentioned several anecdotes from the fascinating history of cryptology and attempted to provide motivation for the problems considered in Chapters III through V.

Linear feedback shift registers, linear complexity, the linear complexity profile, and cryptographically strong pseudo-random sequences were discussed in Chapter III. We then defined the $k$-complexity—a generalized linear complexity for periodic sequences—to be the minimum linear complexity that can be obtained when any $k$ or fewer elements of the original sequence are changed. It was shown that in certain cases, the $k$-complexity provides more information on the cryptographic strength of a sequence than the linear complexity profile.

The search for efficient algorithms for computing the $k$-complexity was the focus of Chapter IV. The Berlekamp-Massey algorithm and the Chan-Games algorithm—two standard algorithms for computing the linear complexity—were presented. We also proved a new result from which the Chan-Games algorithm follows as a special case. We then introduced and proved a new and efficient $k$-complexity algorithm. This $k$-complexity algorithm is only valid in the special case where the sequences $s$ is a periodic binary sequence with period $2^n$, but it is surprisingly efficient. The general case was also considered, but no satisfactory algorithm has yet been discovered. The computational complexity of the underlying problem was also briefly considered.

In Chapter V our $k$-complexity algorithm of the previous chapter was applied to binary sequences with period $2^n$. Particular emphasis was given to the class of sequences known as binary de Bruijn sequences. The de Bruijn sequences were chosen because they have period $2^n$ and because they have been intensively studied in the literature. In particular, the linear complexities of de Bruijn sequences are known. We also mentioned a new class of binary sequences which appear to have good $k$-complexity properties. Various computational results were given for both the de Bruijn sequences and this new class of sequences.

There are many significant problems remaining, the most fundamental of which is to determine an efficient algorithm for the general case, or to show that the underlying problem is NP-complete. It would also be interesting to generalize known results on linear complexity to the case of the $k$-complexity. For example, the linear complexities of classes of binary sequences such as GMW sequences [68], No sequences [58], and modified de Bruijn sequences [50] are known, but the corresponding $k$-complexities are not.

Rueppel [65] analyzed the statistical properties of the linear complexity profile and Niederreiter [57] extended Rueppel's results to produce a detailed probabilistic theory of linear complexity. Analogous results for the $k$-complexity would be significant.

Chan and Games [9] give an algorithm for finding the *quadratic complexity* of a bit-string $s$, i.e., the shortest recurrence involving linear and quadratic terms which will produce $s$. Algorithms for quadratic and higher order $k$-complexities would also be of interest.

## REFERENCES


[1] T. van Aardenne-Ehrenfest and N. G. de Bruijn, "Circuits and trees in oriented linear graphs," *Simon Stevin*, Vol. 28, pp. 203–217, 1951.

[2] E. R. Berlekamp, *Algebraic Coding Theory*, McGraw-Hill, New York, 1968.

[3] E. R. Berlekamp, R. J. McEliece and H. C. A. van Tilborg, "On the inherent intractability of certain coding problems," *IEEE Transactions on Information Theory*, Vol. IT–24, No. 3, pp. 480–484, May 1978.

[4] J. Bernasconi and C. G. Günther, "Analysis of a nonlinear feedforward logic for binary sequence generators," *Advances in Cryptology—EUROCRYPT '85*, Lecture Notes in Computer Science 219, F. Pichler, ed., Springer-Verlag, pp. 161-166, 1986.

[5] M. Blum and S. Micali, "How to generate cryptographically strong pseudo-random sequences," *SIAM Journal on Computing*, Vol. 13, No. 4, pp. 850–864, November 1984.

[6] N. G. de Bruijn, "A combinatorial problem," *Koninklijke Nederlandse Akademie van Wetenschappen, Indagationes Mathematicae*, Series A, Vol. 49, pp. 758–764, 1946.

[7] N. G. de Bruijn, "Acknowledgement of priority to C. Flye Sainte-Marie on the counting of $2^n$ zeros and ones that show each $n$-letter word exactly once," *Technical Report TH 75-WSK-06*, Technische Hogeschool Eindhoven, 1975.

[8] D. M. Burton, *Elementary Number Theory*, second edition, Wm. C. Brown, Dubuque, Iowa, 1989.

[9] A. H. Chan and R. A. Games, "On the quadratic span of de Bruijn sequences," *IEEE Transactions on Information Theory*, Vol. 36, No. 4, pp. 822–829, July 1990.

[10] A. H. Chan, R. A. Games and E. L. Key, "On the complexities of de Bruijn sequences," *Journal of Combinatorial Theory*, Series A, Vol. 33, pp. 233–246, 1982.

[11] D. Chaum, "Security without identification: Transaction systems to make big brother obsolete," *Communications of the ACM*, Vol. 28, pp. 1030–1044, October 1985.

[12] U. Cheng, "On the continued fraction and Berlekamp's algorithm," *IEEE Transactions on Information Theory*, Vol. IT–30, No. 3, pp. 541–544, May 1984.

[13] Z. Dai and K. Zeng, "Continued fractions and the Berlekamp-Massey algorithm," in *Advances in Cryptology—AUSCRYPT '90*, Lecture Notes in Computer Science 453, J. Seberry and J. Pieprzyk, eds., Springer-Verlag, pp. 24–31, 1990.

[14] E. Dawson, "Linear feedback shift registers and stream ciphers," in *Number Theory and Cryptography*, London Mathematical Society Lecture Note Series 154, J. H. Loxton, ed., Cambridge University Press, pp. 106–119, 1990.

[15] N. Deo, *Graph Theory with Applications to Engineering and Computer Science*, Prentice-Hall, Englewood Cliffs, New Jersey, 1974.

[16] W. Diffie and M. E. Hellman, "New directions in cryptography," *IEEE Transactions on Information Theory*, Vol. IT–22, pp. 644–654, November 1976.

[17] W. Diffie and M. E. Hellman, "Exhaustive cryptanalysis of the NBS Data Encryption Standard," *Computer*, Vol. 10, pp. 74–84, June 1977.

[18] T. Etzion and A. Lempel, "Algorithms for the generation of full-length shift-register sequences," *IEEE Transactions on Information Theory*, Vol. IT–30, No. 3, pp. 480–484, May 1984.

[19] C. Falls, *The Great War*, G. P. Putnam's Sons, New York, 1959.

[20] H. Fredricksen, "A survey of full length nonlinear shift register cycle algorithms," *SIAM Review*, Vol. 24, No. 2, pp. 195–221, April 1982.

[21] R. A. Games and A. H. Chan, "A fast algorithm for determining the complexity of a pseudo-random sequence with period $2^n$," *IEEE Transactions on Information Theory*, Vol. IT–29, No. 1, pp. 144–146, January 1983.

[22] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W. H. Freeman and Company, New York, 1979.

[23] D. B. Glover, *Secret Ciphers of the 1876 Presidential Election*, Aegean Park Press, Laguna Hills, California, 1991.

[24] S. W. Golomb, *Shift Register Sequences*, Aegean Park Press, Laguna Hills, California, 1982.

[25] I. J. Good, "Normal recurring decimals," *Journal of the London Mathematical Society*, Vol. 21, pp. 167–169, 1946.

[26] H. W. Gould, *Combinatorial Identities*, West Virginia University, Morgantown, West Virginia, 1972.

[27] W. B. Gragg, "Matrix interpretations and applications of the continued fraction algorithm," *Rocky Mountain Journal of Mathematics*, Vol. 4, No. 2, pp. 213–225, Spring 1974.

[28] R. L. Graham, D. E. Knuth and O. Patashnik, *Concrete Mathematics*, Addison-Wesley Publishing Co., Reading, Massachusetts, 1989.

[29] E. J. Groth, "Generation of binary sequences with controllable complexity," *IEEE Transactions on Information Theory*, Vol. IT–17, pp. 288–296, May 1971.

[30] F. G. Gustavson, "Analysis of the Berlekamp-Massey linear feedback shift-register synthesis algorithm," *IBM Journal of Research and Development*, Vol. 20, pp. 204–212, May 1976.

[31] T. Helleseth and T. Kløve, "The number of cross-join pairs in maximum length linear sequences," *IEEE Transactions on Information Theory*, Vol. IT–37, No. 6, November, 1991, pp. 1731–1733.

[32] K. Imamura and W. Yoshida, "A simple derivation of the Berlekamp-Massey algorithm and some applications," *IEEE Transactions on Information Theory*, Vol. IT–33, No. 1, pp. 146–150, January 1987.

[33] D. Kahn, *The Codebreakers: The Story of Secret Writing*, MacMillan Co., New York, 1967.

[34] E. L. Key, "An analysis of the structure and complexity of nonlinear binary sequence generators," *IEEE Transactions on Information Theory*, Vol. IT–22, pp. 732–736, November 1976.

[35] D. E. Knuth, *The Art of Computer Programming*, Vol. 2, Seminumerical Algorithms, 2nd ed., Addison-Wesley Publishing Co., Reading, Massachusetts, 1981.

[36] G. B. Kolata, "Computer encryption and the National Security Agency connection," *Science*, Vol. 197, pp. 438–440, July 29, 1977.

[37] A. Lempel, "On a homomorphism of the de Bruijn graph and its applications to the design of feedback shift registers," *IEEE Transactions on Computers*, Vol. C–19, No. 12, pp. 1204–1209, December 1970.

[38] R. Lessard, "Cycles of $n$-digit binary numbers," *American Mathematical Monthly*, Vol. 58, No. 8, pp. 573–575, October 1951.

[39] R. Lidl, "Some mathematical aspects of recent advances in cryptology," in *Number Theory and Cryptography*, London Mathematical Society Lecture Note Series 154, ed. J. H. Loxton, Cambridge University Press, 1990.

[40] A. H. Lightstone and A. Robinson, *Nonarchimedean Fields and Asymptotic Expansions*, American Elsevier, New York, 1975.

[41] J. H. van Lint, *Combinatorial Theory Seminar Eidenhoven University of Technology*, Lecture Notes in Mathematics 382, Springer-Verlag, Berlin, 1974.

[42] D. M. Mandelbaum, "An approach to an arithmetic analog of Berlekamp's algorithm," *IEEE Transactions on Information Theory*, Vol. IT–30, No. 5, pp. 758–762, September 1984.

[43] W. Mantel, "Resten van wederkerige reeksen," *Nieuw Archief voor Wiskunde*, Serie 2, Vol. 1, pp. 172–184, 1897.

[44] C. F. Martin and M. Stamp, "Constructing polynomials over finite fields," in *Computation and Control: Proceedings of the Bozeman Conference* (Progress in Systems and Control Theory, Vol. 1), eds. K. Bowers and J. Lund, Birkhäuser, Boston, pp. 233–252, 1989.

[45] C. F. Martin and M. Stamp, "Classification and realization of pseudo-random number generators," *System and Control Letters*, Vol. 14, pp. 169–175, 1990.

[46] M. H. Martin, "A problem in arrangements," *Bulletin of the American Mathematical Society*, Vol. 40, No. 12, pp. 859–864, December 1934.

[47] P. Martin-Löf, "The definition of random sequences," *Information and Control*, Vol. 9, pp. 602–619, 1966.

[48] J. L. Massey, "Shift-register synthesis and BCH decoding," *IEEE Transactions on Information Theory*, Vol. IT–15, No. 1, pp. 122–127, January 1969.

[49] J. L. Massey, "An introduction to contemporary cryptology," *Proceedings of the IEEE*, Vol. 76, No. 5, pp. 533–549, May 1988,

[50] G. L. Mayhew and S. W. Golomb, "Linear spans of modified de Bruijn sequences," *IEEE Transactions on Information Theory*, Vol. 36, No. 5, pp. 1166–1167, September 1990.

[51] R. J. McEliece, *Finite Fields for Computer Scientists and Engineers*, Kluwer Academic Publishers, Norwell, MA, 1987.

[52] R. C. Merkle, "Secure communications over insecure channels," *Communications of the ACM*, Vol. 21, pp. 294–299, April 1978.

[53] R. C. Merkle and M. E. Hellman, "Hiding information and signatures in trapdoor knapsacks," *IEEE Transactions on Information Theory*, Vol. IT–24, pp. 525–530, September 1978.

[54] W. H. Mills, "Continued fractions and linear recurrences," *Mathematics of Computation*, Vol. 29, No. 129, pp. 173–180, January 1975.

[55] R. Morris, "The data encryption standard—retrospective and prospects," *IEEE Communications Society Magazine*, Vol. 16, pp. 11–14, November 1978.

[56] H. Niederreiter, "Sequences with almost perfect linear complexity profile," in *Advances in Cryptology—EUROCRYPT '87*, Lecture Notes in Computer Science 304, D. Chaum and W. L. Price, eds., Springer-Verlag, pp. 37–51, 1988.

[57] H. Niederreiter, "The probabilistic theory of linear complexity," in *Advances in Cryptology—EUROCRYPT '88*, Lecture Notes in Computer Science 330, C. G. Günther, ed., Springer-Verlag, pp. 191–209, 1988.

[58] J.-S. No and P. V. Kumar, "A new family of binary pseudorandom sequences having optimal periodic correlation properties and large linear spans," *IEEE Transactions on Information Theory*, Vol. 35, No. 2, pp. 371–379, March 1989.

[59] A. J. van der Poorten, "Notes on continued fractions and recurrence sequences," in *Number Theory and Cryptography*, London Mathematical Society Lecture Note Series 154, J. H. Loxton, ed., Cambridge University Press, pp. 86–97, 1990.

[60] K. R. Popper, *The Logic of Scientific Discovery*, Basic Books, New York, 1959.

[61] J. A. Reeds and N. J. A. Sloane, "Shift-register synthesis (modulo $m$)," *SIAM Journal on Computing*, Vol. 14, No. 3, pp. 505–513, August 1985.

[62] D. Rees, "A note on a paper by I. J. Good," *Journal of the London Mathematical Society*, Vol. 21, pp. 169–172, 1946.

[63] R. L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Communications of the ACM*, Vol. 21, pp. 120–126, February 1978.

[64] A. de Rivière, "Question no. 48," *l'Intermédiare des Mathématiciens*, Vol. 1, No. 2, February 1894, pp. 19–20.

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

[66] R. S. Safavi-Naini and J. R. Seberry, "Pseudo-random sequence generators using structured noise," in *Number Theory and Cryptography*, London Mathematical Society Lecture Note Series 154, J. H. Loxton, ed., Cambridge University Press, pp. 129–136, 1990.

[67] C. Flye Saint-Marie, "Solution to question no. 48," *l'Intermédiare des Mathématiciens*, Vol. 1, No. 6, pp. 107–110, June 1894.

[68] R. A. Scholtz and L. R. Welch, "GMW sequences," *IEEE Transactions on Information Theory*, Vol. IT–30, No. 3, pp. 548–553, May 1984.

[69] A. Shamir, "On the generation of cryptographically strong pseudo-random sequences," in *8th International Colloquium on Automata, Languages, and Programming*, Lecture Notes in Computer Science 62, Springer-Verlag, New York, pp. 544–550, 1981.

[70] A. Shamir, "A polynomial-time algorithm for breaking the basic Merkle-Hellman cryptosystem," *Proceedings of the 23rd IEEE Symposium on Foundations of Computer Science*, pp. 145–151, 1982; and in *IEEE Transactions on Information Theory*, Vol. IT–30, pp. 699–704, September 1984.

[71] C. E. Shannon, "A mathematical theory of communication," *Bell System Technical Journal*, Vol. 27, pp. 379–423, July 1948, and pp. 623–656, October 1948.

[72] C. E. Shannon, "Communication theory of secrecy systems," *Bell System Technical Journal*, Vol. 28, pp. 656–715, October 1949.

[73] G. J. Simmons, "Cryptology," in *The New Encyclopædia Britannica*, Vol. 16, Macropædia, pp. 860–873, Encyclopædia Britannica, Inc., Chicago, 1989.

[74] M.-K. Siu and P. Tong, "Generation of some de Bruijn sequences," *Discrete Mathematics*, Vol. 31, pp. 97–100, 1980.

[75] Staff Report of the Senate Select Committee on Intelligence, "Unclassified summary: Involvement of NSA in the development of the data encryption standard," April 1978.

[76] M. Stamp, "Circular binary sequences," to appear as Problem 92–12, *SIAM Review*, Vol. 34, No. 3, September 1992.

[77] M. Stamp and C. F. Martin, "An algorithm for computing the $k$-complexity of binary sequences with period $2^n$," submitted to *IEEE Transactions on Information Theory*.

[78] S. K. Stein, *Mathematics: The Man-Made Universe*, W. H. Freeman and Company, San Francisco, 1963.

[79] A. P. Street, "Eulerian washing machine," in *Combinatorial Mathematics, Proceedings of the Second Australian Conference*, Lecture Notes in Mathematics 403, A. Dold and B. Eckmann, eds., Springer-Verlag, Berlin, 1974.

[80] B. W. Tuchman, *The Zimmermann Telegram*, Dell Publishing Co., Inc., New York, 1963.

[81] P. Unger, "Problem 4385," *American Mathematical Monthly*, Vol. 57, No. 3, p. 188, March 1950.

[82] L. R. Welch and R. A. Scholtz, "Continued fractions and Berlekamp's algorithm," *IEEE Transactions on Information Theory*, Vol. IT–25, No. 1, pp. 19–27, January 1979.