How to Break Gifford's Cipher
(extended abstract)

Thomas R. Cain* and Alan T. Sherman†
Computer Science Department
University of Maryland Baltimore County
Baltimore, Maryland 21228-5398
cain@cs.umbc.edu, sherman@cs.umbc.edu

May 31, 1994
(revised August 18, 1994)

Abstract

We present and implement a ciphertext-only algorithm to break Gifford's cipher, a stream cipher designed in 1984 by David Gifford of MIT and used to encrypt New York Times and Associated Press wire reports. Applying linear algebra over finite fields, we exploit a time-space tradeoff to separately determine key segments derived from a decomposition of the feedback function. This work, the first proposed attack on Gifford's cipher, illustrates a powerful attack on stream ciphers and shows that Gifford's cipher is ill-suited for encrypting broadcast data in the MIT-based *Boston Community Information System (BCIS)*.

Gifford's cipher is a *filter generator*—a linear feedback shift register with nonlinear output. Our cryptanalytic problem is to determine the secret 64-bit initial fill, which is changed for each news article. Representing the feedback function as a binary matrix F, we decompose the vector space of register states into a direct sum of four F-invariant subspaces determined from the primary rational canonical form of F. The attack separately computes segments of the key corresponding to these invariant subspaces, which have dimensions 24, 5, 6, and 29, respectively. Because the dimension-24 subspace corresponds to a nilpotent transformation, Gifford's cipher effectively uses only 40 bits of key. With a novel hashing technique, we search these 40 bits in only 2^{27} steps. From the decomposition of F, we also compute the exact probability distribution of the leader and cycle lengths of all state sequences generated by Gifford's cipher.

Our attack runs in 2^{27} steps and 2^{18} bytes of memory, which is a significant shortcut over the 2^{64} steps required for a straightforward exhaustive search of all initial fills. Given ciphertext only from one encrypted article, our prototype implementation running on a loosely-coupled network of eight Sparcstations finds the article key within approximately four hours on average. Exploiting a key-management flaw of the BCIS, we also compute at no additional cost the corresponding master key, used for one month to encrypt all article keys in the same news section.

Keywords. Algorithms over finite fields, Boston Community Information System (BCIS), correlation attack, cryptanalysis, cryptography, cryptology, filter generators, Gifford's cipher, linear algebra over $GF(2)$, linear feedback shift registers (LFSRs), matrix decompositions, primary rational canonical form, similar matrices, similarity transformations, stream ciphers.

---

*Support for this research was provided in part by the University of Maryland Graduate School, Baltimore, through a 1991–92 Graduate Merit Fellowship.

†Part of this work was carried out while Sherman was a member of the Institute for Advanced Computer Studies, University of Maryland College Park.

1 Introduction

In 1982–84, David K. Gifford [12, 13, 14] and his research group at MIT designed and implemented a prototype system for transmitting up-to-the-minute New York Times and Associated Press wire reports to test subscribers in the Boston metropolitan area. Known as the *Boston Community Information System (BCIS)*, Gifford's system broadcast information streams on a subcarrier of MIT's FM radio station WMBR.[1] Each subscriber received and processed the streams using an IBM personal computer equipped with special-purpose receiver hardware. To protect against unauthorized access to the streams, and to be able to deny service to nonpaying customers, Gifford encrypted each stream. For this application, he devised and used a new stream cipher, which we shall call *Gifford's cipher.* The BCIS operated on an experimental basis from April 1984 through January 1988, providing a model for future community information systems. In this paper we analyze the security of Gifford's cipher, which had remained unbroken for almost a decade.

Gifford's cipher is a *filter generator.* As shown in Figure 1, this commonly-used type of cipher comprises a shift register, a linear feedback function, and a nonlinear output function. At each iteration, the feedback function is applied to the contents of the shift register to compute a feedback byte, which is shifted into the register. The output function is applied to four bytes of the shift register to produce a keystream byte. To encrypt a stream of plaintext bytes, each plaintext byte is exclusive-ORed (XORed) with a corresponding keystream byte to yield a ciphertext byte. The secret key is the 64-bit initial fill of the register. We present a new algorithm for computing the initial fill from ciphertext alone.

Several factors motivate us to study Gifford's cipher. First, since Gifford proposed his cipher for use in broadcast communications, it is important to know if this cipher might compromise valuable data. Second, we would like to further the understanding of filter generators so that system engineers can make prudent decisions regarding their implementation and appropriate use. Filter generators are interesting in part because they provide fast bulk encryption and because they can be easily implemented with limited resources. Third, Gifford's cipher provides a practical context in which to explore the general theme of exploiting algebraic decompositions in cryptanalysis.

Our goal is to evaluate the overall effectiveness of Gifford's cipher in protecting broadcast data in the BCIS, and more generally, to study the security of filter generators. Exploiting a decomposition of the feedback matrix F, we point out several ways to break Gifford's cipher. Our main result is the design and implementation of one of these methods, which computes the initial fill given ciphertext alone from one encrypted news article. This method applies a time-space tradeoff and runs in 2^{27} steps using 2^{18} bytes of memory; it does not require any statistical weaknesses of the output function. By contrast, our related statistical attack [6] on filter generators uses less space but assumes a slight statistical weakness in the output function; this alternate attack generalizes Siegenthaler's [35] correlation attack and runs in 2^{29} steps, or more generally 2^d steps, where d is the dimension of the largest subspace in any decomposition of the space of register states into a direct sum of F-invariant subspaces. Combining these two ideas achieves even faster attacks.

This paper explains in detail how to break a real cipher. The ingenuity and novelty of this work lies in its effective application of algorithmic and mathematical concepts—especially linear algebra over the finite field $GF(2)$—in a practical cryptanalytic context. Although we focus on Gifford's cipher, our methods are general in nature.

Previous Work

Although much is known about shift registers, we are aware of only five references to filter generators: Rueppel [34, pp. 83–93] outlines an application of Siegenthaler's [35] correlation attack to filter generators. Siegenthaler's attack is useful, but Rueppel's application of it appears not to be useful against Gifford's cipher. Rueppel [33, Ch. 5] also presents a framework in which to reason about filter generators using the algebraic normal form of the nonlinear output function. In their introduc-

[1] BCIS information was represented as an FM signal, superimposed over the primary WMBR signal. Receiver hardware separated the signals.

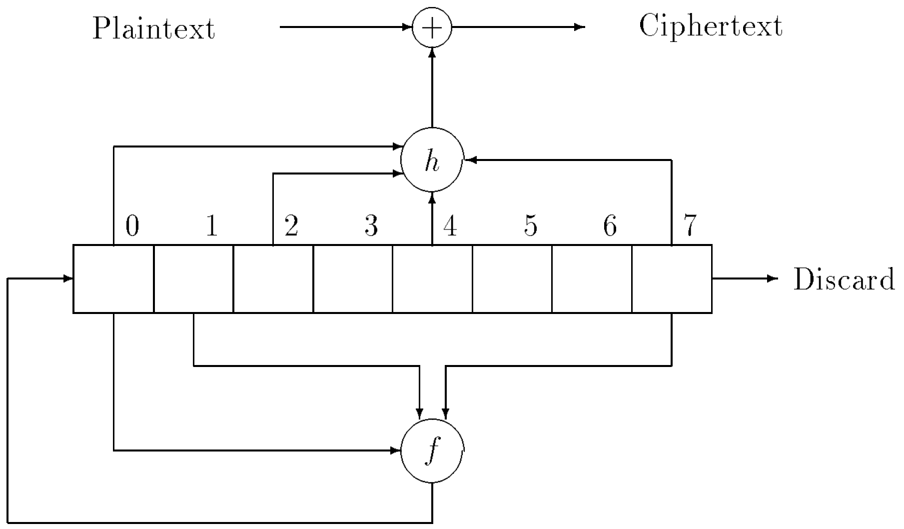

Figure 1: Gifford's stream cipher comprises an 8-byte shift register, a linear feedback function $f : \mathbb{Z}_2^{64} \rightarrow \mathbb{Z}_2^{64}$, and a nonlinear output function h. At each iteration, f computes a new register state as follows: A feedback byte is computed and shifted into the register from the left. Byte B_7 is discarded, and bytes B_0 through B_6 are shifted one byte to the right. The output function h computes an 8-bit keybyte from register bytes B_0, B_2, B_4, and B_7. The secret key is the initial fill of the register. Gifford's cipher generates a keybyte stream, which is XORed with the plaintext stream to produce a ciphertext stream. In this figure, $+$ denotes XOR.

tory survey on stream ciphers, Zeng, Yang, Wei, and Rao [37] briefly review how the linear syndrome attack can be applied to filter generators; and Dawson [7] states a few basic properties of filter generators. In addition, Key [23] sketches by example a method for analyzing the periodic properties of the keystream of certain filter generators.

For the classical theory of shift register cryptosystems, there are expositions by Beker and Piper [1, Ch. 5], Gill [15], Golomb [18], Rhee [31, Ch. 4], Ronse [32], and Rueppel [33]. A variety of attacks on stream ciphers have been published, including statistical attacks by Dawson and Clark [8], Golić and Mihaljević [16], Gollmann and Chambers [17], Klapper [24], Meier and Stafelbach [29], and Siegenthaler [35]. Many classical results are proven in Peterson and Weldon [30] and Berlekamp [2]. For a general survey of cryptanalytic techniques, see Brickell and Odlyzko [3].

These references, however, do not adequately address the algorithmic aspects of efficiently applying linear algebra (including matrix decompositions) to cryptanalysis. Moreover, we found no previous work that describes in complete practical detail how to break any stream cipher.

2 Gifford's Cipher

Gifford's cipher encrypts each news article under a separately chosen 64-bit article key s_0. Each article is a sequence of 8-bit bytes $P_0, P_1, \ldots, P_{N-1}$; typically, $N \approx 10,000$. As shown in Figure 1, Gifford's cipher encrypts each article byte-by-byte, XORing each byte of plaintext with a corresponding keystream byte. Our first two steps in analyzing the cipher were to determine its exact operation and to choose a suitable mathematical model in which to reason about its properties. Since Gifford's [13, pp. 464–465] published description of his cipher is incomplete, we started with source code from the BCIS.

The keystream bytes are computed by applying a nonlinear output function h to the contents of a 64-bit shift register, which for efficiency is implemented as a sequence of eight bytes. Specifically, for each $0 \leq t < N$, the tth byte of ciphertext is $C_t = P_t \oplus K_t$, where $s_t = f^t(s_0)$ is the tth state of the shift register; $K_t = h(s_t)$ is the tth keystream byte; and $\oplus$ denotes XOR. Here, $\mathbb{Z}_2 = \{0, 1\}$; $f : \mathbb{Z}_2^{64} \rightarrow \mathbb{Z}_2^{64}$ is the feedback function which is linear over the two-element Galois field $GF(2)$; and $h : \mathbb{Z}_2^{64} \rightarrow \mathbb{Z}_2^{8}$ is the output function which is nonlinear over $GF(2)$.

Each article key was encrypted by XOR with the master key. Therefore, compromise of any article key also compromised the corresponding master key, which remained valid for one month.

2.1 The Linear Feedback Function f

For any register state $s_t = (B_0, B_1, \ldots, B_7)$, the function f computes the next state of the register as $f(s_t) = (f_{new}(B_0, B_1, B_7), B_0, B_1, \ldots, B_6)$, where the new feedback byte is computed by the function $f_{new} : \mathbb{Z}_2^{24} \to \mathbb{Z}_2^8$. The feedback byte is the XOR of bytes B_0, B_1, and B_7, with byte B_1 "sticky"-shifted one bit to the right, and with byte B_7 zero-fill-shifted one bit to the left. Thus,

$$f_{new}(B_0, B_1, B_7) = B_0 \oplus (\gg_1^\star (B_1)) \oplus (\ll_1 (B_7)), \quad (1)$$

where $\gg_1^\star$ and $\ll_1$ denote, respectively, the sticky right-shift and zero-fill left-shift operations. Specifically, for any byte $B = (x_0, x_1, \ldots, x_7)$, $\gg_1^\star (B) = (x_0, x_0, x_1, x_2, \ldots, x_6)$ and $\ll_1 (B) = (x_1, x_2, \ldots, x_6, x_7, 0)$. Concerning Gifford's decision to tap bytes B_0, B_1, and B_7, see Section 6.2.

The bit-shifting of bytes B_1 and B_7 complicates the feedback function in two respects: First, this bit-shifting causes the function f_{new} to be nonlinear over bytes—*i.e.* over $GF(2^8)$. Second, the bit-shifting causes f to be (slightly) noninvertible. Specifically, f loses one bit—the high-order (leftmost) bit of byte B_7.

2.2 The Nonlinear Output Function h

The nonlinear output function h extracts an 8-bit byte from the product of two 16-bit integers derived from shift register bytes B_0, B_2, B_4, and B_7. For any register state $s = (B_0, B_1, \ldots, B_7)$,

$$\begin{aligned} h(s) &= \text{Extract\_Byte}\left((B_0 \| B_2) * (B_4 \| B_7)\right) \\ &= (B_2 B_4 + B_0 B_7 + \lfloor B_2 B_7 / 256 \rfloor) \bmod 256, \quad (2) \end{aligned}$$

where $\|$ denotes concatenation; $*$ denotes integer multiplication; and the function Extract_Byte : $\mathbb{Z}_2^{32} \to \mathbb{Z}_2^8$ extracts the third byte from the left of any 32-bit number. It is easy to verify that h is nonlinear, both over $GF(2)$ and over $GF(2^8)$.[2]

[2] Gifford's inspiration for h came from a well-known 1946 idea of John von Neumann (see Knuth [25, pp. 3–4]).

3 Decomposition of the Feedback Function

Our attack exploits a decomposition of the feedback function f to search segments of the key separately. To begin, we view the state space $\mathcal{S} = \mathbb{Z}_2^{64}$ as a vector space over $GF(2)$, and we view f as a linear transformation of $\mathcal{S}$. We represent the feedback function f as a binary matrix F and work with its primary rational canonical form R. The binary matrix R is a block diagonal matrix, similar to F. This decomposition of F induces a decomposition of the state space into a direct sum of F-invariant subspaces. In a one-time precomputation, we find the primary rational canonical form R of F and an invertible binary similarity matrix P such that $F = P^{-1} R P$.

3.1 The Feedback Matrix F

From Equation 1, we represent the feedback function f as a 64×64 binary matrix

$$F = \begin{pmatrix} F_0 & F_1 & 0 & F_7 \\ & I_{56,56} & & 0 \end{pmatrix}, \quad (3)$$

where F_0, F_1, F_7 are certain 8×8 blocks and $I_{56,56}$ is a 56×56 identity matrix. The blocks F_0, F_1, F_7 calculate the feedback byte; $I_{56,56}$ describes the shifting of bytes B_0 through B_6. Because byte B_1 is sticky-shifted, the upper-left bit of block F_1 is one (see Section 6.1).

3.2 The Characteristic and Minimal Polynomials of F and Their Factors

To compute R, it is helpful to know the characteristic and minimal polynomials of F and their irreducible factors. The *characteristic polynomial* of F is the degree 64 polynomial $p_F(x) = \det(F - xI)$, where I is the 64×64 identity matrix. The *minimal polynomial* of F, denoted by $m_F(x)$, is the polynomial of smallest degree over $\mathbb{Z}_2$ such that $m_F(F) = 0$.[3]

[3] For a review of linear algebra, see Hoffman and Kunze [20], Hungerford [21], and Jacob [22].

Using the numerical math package *Matlab*, we computed

$$p_F(x) = x^{64} + x^{62} + x^{61} + x^{60} + x^{59} + x^{58} + x^{57} + x^{55} + x^{54} + x^{52} + x^{50} + x^{48} + x^{44} + x^{40} + x^{24} \quad (4)$$

by working over $\mathbb{R}$ and accepting the modulo 2 values of the coefficients.[4] Intuitively, $p_F(x)$ encodes all of the information of F in a convenient algebraic form. Using *Macsyma* [27], we factored $p_F(x)$ into a product of irreducible polynomials $p_F(x) = p_0^{24}(x)\, p_1(x)\, p_2(x)\, p_3(x)$, where

$$p_0(x) = x, \quad (5)$$
$$p_1(x) = x^5 + x^3 + x^2 + x + 1, \quad (6)$$
$$p_2(x) = x^6 + x^5 + x^4 + x^2 + 1, \quad (7)$$
$$p_3(x) = x^{29} + x^{28} + x^{26} + x^{22} + x^{20} + x^{19} + x^{18} + x^{16} + x^{14} + x^{13} + x^{10} + x^9 + x^7 + x^5 + x^4 + x^3 + x^2 + x + 1. \quad (8)$$

For Gifford's cipher, $m_F(x) = p_F(x)$, which fact simplifies some of the theory. Let $m_F(x) = m_0(x)\, m_1(x)\, m_2(x)\, m_3(x) = p_F(x)$, where $m_i(x) = p_i^{t_i}(x)$, for $0 \le i \le 3$, and $t_0 = 24$ and $t_1 = t_2 = t_3 = 1$. The prime-power polynomials $m_i(x)$ are called the *elementary divisors* of F.

3.3 An Invariant Decomposition of the State Space

The elementary divisors of F decompose the state space into a direct sum of four F-invariant subspaces $\mathcal{S} = V_0 \oplus V_1 \oplus V_2 \oplus V_3$. By the Invariant Subspace Decomposition Theorem [22, p. 390], $m_i(x)$ is the minimal polynomial of $F|V_i$, and $V_i = \ker(m_i(F))$, for $0 \le i \le 3$. Furthermore, since $m_F(x) = p_F(x)$, the Cyclic Subspace Decomposition Theorem [21, p. 356] implies that each V_i is F-cyclic.[5]

[4] We also tried using the symbolic math packages *Mathematica* and *Macsyma*. We found *Mathematica* poorly suited for doing arithmetic over $\mathbb{Z}_2$, and *Macsyma* ran too slowly.

[5] We say that $v \in V$ is a *T-cyclic vector* for V if and only if the set $\{v, T(v), \ldots, T^{n-1}(v)\}$ forms a basis of V. We say that V is a *T-cyclic vector space* if and only if V has a cyclic vector.

3.4 The Primary Rational Canonical Form R of F

The *primary rational canonical form (RCF)* for F, denoted $RCF(F)$, is a block-diagonal matrix corresponding to the F-cyclic decomposition of the state space given in Section 3.3.

For each $0 \le i \le 3$, let $m_i(x) = p_i^{t_i}(x) = x^{n_i} + \sum_{j=0}^{n_i-1} \alpha_{ij} x^j$ be the minimal polynomial of $F|V_i$, as defined in Section 3.2, where n_i is the degree of $m_i(x)$, and α_{ij} is the jth (binary) coefficient of $m_i(x)$. Thus, $n_0 = 24$, $n_1 = 5$, $n_2 = 6$, and $n_3 = 29$.

By definition, the RCF for F is the 64×64 block-diagonal matrix

$$R = \begin{pmatrix} R_0 & & & \\ & R_1 & & \\ & & R_2 & \\ & & & R_3 \end{pmatrix}, \quad (9)$$

where for each $0 \le i \le 3$, block R_i is the *companion matrix* [20, p. 230]

$$R_i = \begin{pmatrix} 0 & 0 & \ldots & 0 & \alpha_{i0} \\ 1 & 0 & \ldots & 0 & \alpha_{i1} \\ 0 & 1 & \ldots & 0 & \alpha_{i2} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \ldots & 1 & \alpha_{i(n_i-1)} \end{pmatrix}. \quad (10)$$

Each companion matrix R_i is a lower-diagonal matrix, whose last column consists of the coefficients in the associated minimal polynomial $m_i(x)$. The matrix R is a canonical representation of the equivalence class of all matrices similar to F; it is unique up to the order of the blocks.

Block R_0 plays a special role because it represents a nilpotent transformation with $R_0^{24} = 0$. To see that R_0 is nilpotent, observe that the only ones in R_0 are along the lower diagonal, which happens whenever the corresponding elementary divisor is a power of x. Since the nilpotent block determines the leader length of any state sequence, the maximum leader length of any state sequence generated by Gifford's cipher is 24 states.

3.5 A Similarity Transformation P from F to R

A *similarity matrix* from F to R is any invertible binary matrix P such that $F = P^{-1}RP$. Our attack uses such a matrix to move between the original and decomposed state spaces, as needed to check any candidate key segments. We computed a similarity matrix using our own 59^3-step algorithm [5, 6], which runs faster than the 64^6-step algorithm suggested by Gill [15]. Our algorithm constructs the columns of P^{-1} as powers of cyclic vectors for each of the blocks.

4 The Probability Distribution of Leader and Cycle Lengths

For each initial fill $s_0 \in \mathcal{S}$, Gifford's cipher computes an eventually periodic sequence of keybytes $\{K_t\}_{t=0}^{\infty}$. Each sequence consists of a leader and a cycle, where the *leader* is the initial nonperiodic part, and the *cycle* is the periodic part. Although long periods do not guarantee high security, short periods create serious weaknesses. Therefore, it is important to know the probability distribution of the leader and cycle lengths of the keystream. Similarly, it is important to understand the related periodic properties of the underlying sequence of register states $\{s_t\}_{t=0}^{\infty}$.

From the exponents of the elementary divisors of F, we compute the exact probability distribution of the leader and cycle lengths of the state stream. Excluding the degenerate $s_0 = 0$, leader lengths range from 0 to 24 states, and cycle lengths range from 21 to $349,502,963,061 \approx 3.5 \times 10^{11}$ states.

The leader and cycle lengths of the state stream are upper bounds on the leader and cycle lengths of the keystream. It is possible, however, that the keystream repeats before the state stream repeats—though if this happens, the length of the keystream cycle must properly divide the length of the corresponding state stream cycle. A one-day computer search found no initial fill whose keystream repeats before the state stream repeats.

4.1 Leaders, Cycles, and Maximum Periods

For any initial fill $s_0 \in \mathcal{S}$, let $\lambda_f(s_0)$ and $\pi_f(s_0)$ denote, respectively, the *leader length* and *cycle length* of the eventually periodic sequence $\{s_t\}_{t=0}^{\infty}$. Thus, $\pi_f(s_0) = \min\{p \in \mathbb{N} : s_{t+p} = s_t \text{ for all sufficiently large } t \in \mathbb{N}\}$. If $\lambda_f(s_0) = 0$, then we say that the sequence is *strictly periodic*.

Also, let λ_f^* and π_f^* denote, respectively, the maximum leader and cycle lengths over all initial fills. Thus, $\pi_f^* = \max\{\pi_f(s) : s \in \mathcal{S}\}$ and $\lambda_f^* = \max\{\lambda_f(s) : s \in \mathcal{S}\}$. Note that $\pi_f(0) = 1$ and $\pi_f^* \leq 2^{64} - 1$.

4.2 Periodic Properties of F

For any $s_0 \in \mathcal{S}$, $\pi_f(s_0)$ can be computed in terms of the exponents of the elementary divisors of F that generate the subspace to which s_0 belongs. Let $f(x) \in \mathbb{Z}_2[x]$. The *exponent of* f, denoted $\exp(f)$, is the least positive integer r such that $f(x) \mid x^r - 1$. If there is no such integer r, we say that the exponent is 0. Theorem 1 states a relationship between exponents and periods.

Theorem 1. *Let $\mathcal{S} = V_0 \oplus V_1 \oplus V_2 \oplus V_3$ be the direct sum decomposition of the state space of Gifford's cipher given in Section 3.3. For $0 \leq i \leq 3$, let $m_i(x) = p_i^{t_i}(x)$ be the elementary divisors of F given in Section 3.2, and let $e_i = \exp(m_i)$. Also, for each $1 \leq i \leq 3$, define $\beta_i = \min\{2^j : j \in \mathbb{N} \text{ and } 2^j \geq t_i\}$, and let $\beta = \max\{\beta_1, \beta_2, \beta_3\}$. Let $v = (v_0, v_1, v_2, v_3) \in \mathcal{S}$. For each $1 \leq i \leq 3$, it is true that: (1) If $v_i \neq 0$, then $\pi_f(v_i) = \beta_i e_i = e_i$. (2) $\pi_f(v) = \beta' \operatorname{lcm}\{e_i : v_i \neq 0 \text{ and } 1 \leq i \leq 3\}$, where $\beta' = \max\{\beta_i : v_i \neq 0 \text{ and } 1 \leq i \leq 3\} = 1$. (3) $\pi_f^* = \beta \operatorname{lcm}(e_1, e_2, e_3) = \operatorname{lcm}(e_1, e_2, e_3)$. (4) $F^{t_0}(v) = F^{24}(v) \in V_1 \oplus V_2 \oplus V_3$, and (5) $\lambda_f^* = t_0 = 24$.*

Proof. First, observe that $t_0 = 24$, $t_1 = t_2 = t_3 = 1$, and $\beta = \beta_1 = \beta_2 = \beta_3 = 1$. The theorem follows from Section 3.3 and from Berlekamp [2, pp. 150–151]. ■

4.3 Exponents of the Elementary Divisors of F

To interpret Theorem 1 numerically, we need to compute the exponents of the elementary divisors of F. Any irreducible polynomial $f(x)$ of degree n is said to be a *primitive polynomial* if and only if $\exp(f) = 2^n - 1$. Any n-stage shift register achieves the maximum period of $2^n - 1$ states if and only if its characteristic polynomial is primitive [1, Ch. 5].

Proposition 1. *For each $0 \leq i \leq 3$, let $e_i = \exp(m_i)$ be the exponent of the elementary divisor*

$m_i(x)$ of F defined in Section 3.2. It is true that $e_0 = 0$, $e_1 = 31$, $e_2 = 21$, and $e_3 = 2^{29} - 1$. Consequently, $m_1(x)$ and $m_3(x)$ are primitive polynomials, but $m_0(x)$ and $m_2(x)$ are not primitive.

Proof. See Appendix A. This calculation requires some work. ■

4.4 An Exact Characterization of Leaders and Cycles

We apply Theorem 1 and Proposition 1 to characterize exactly the set of eventually periodic state sequences that can be generated by Gifford's cipher. Corollary 1 computes the maximum period, which depends only on subspaces V_1, V_2, and V_3. Subspace V_0 affects only the leader. The maximum leader length is $\lambda_f^* = 24$ states (see Section 3.4).

Corollary 1. $\pi_f^* = 349,502,963,061$.

Proof. By Theorem 1 and Proposition 1, $\pi_f^* = \mathrm{lcm}(31, 21, 2^{29} - 1) = 31 \cdot 21(2^{29} - 1) = 349,502,963,061$, since the exponents 31, 21, $2^{29} - 1$ are relatively prime. ■

Each initial fill s_0 determines four subfills $Ps_0 = (d_0, d_1, d_2, d_3)$ in our invariant decomposition of $\mathcal{S}$. These subfills belong to the invariant subspaces V_0, V_1, V_2, V_3 of dimensions 24, 5, 6, 29, respectively. There are eight possible periods $0, 21, 31, 2^{29}-1, 21\cdot 31, 21\cdot 2^{29}-1, 31\cdot 2^{29}-1, 21\cdot 31\cdot 2^{29}-1$, corresponding to the eight possible ways in which up to three of the subfills d_1, d_2, and d_3 can be zero. In addition, cycles achieving these periods can occur with or without a leader, depending on whether subfill d_0 is zero. Thus, there are 16 equivalence classes of initial fills. For example, a 31-state cycle is created whenever $d_1 \neq 0$ and $d_2 = d_3 = 0$. Similarly, a maximum-length cycle occurs whenever $d_1 d_2 d_3 \neq 0$.

The probability of generating any one of these 16 possible equivalence classes of sequence lengths can be computed from the dimension of the subspace U that generates the given sequence length: the probability that a randomly chosen initial fill lands in U is $|U|/|\mathcal{S}|$. For example, the probability of generating a length 31 cycle (with nonzero leader) is $|V_0^+ \oplus V_1^+|/|\mathcal{S}| = (2^{24} - 1)(2^5 - 1)/2^{64} \approx 2^{-(64-29)} = 2^{-35}$, where V_0^+ and V_1^+ denote the set of nonzero elements in V_0 and V_1, respectively.

The shortest cycle, however, comprises 21 and not 31 states. For dimension-6 subspace V_2, elementary divisor $m_2(x)$ is not primitive and generates one of three submaximal-length cycles of length 21. By contrast, because elementary divisors $m_1(x)$ and $m_3(x)$ are primitive, they always generate maximal-length cycles of lengths $2^5 - 1 = 31$ and $2^{29} - 1$, respectively.

With very high probability (> 0.9998) the cycle will contain at least $2^{29} - 1 \approx 5.4 \cdot 10^8$ states, and the maximum period $\pi_f^* \approx 3.5 \cdot 10^{11}$ occurs with probability approximately 0.9536. Yet with non-negligible probability of $2^{-11} \approx 0.0005 = 0.05\%$, the cycle length is only $2^{29} - 1 < 10^9$. This fact partially contradicts Gifford's [13, p. 465] experimental finding that "the period has consistently been found to exceed 10^9."[6]

5 Attacks

The decomposition of the state space into a direct sum of invariant subspaces makes possible a variety of cryptanalytic attacks on filter generators that search segments of the key corresponding to this decomposition. In this section, we outline four such ciphertext-only attacks applied to Gifford's cipher: (1) a simple 2^{40}-step attack based on exhaustive search, (2) our novel time-space tradeoff attack, which uses 2^{27} steps and 2^{18} bytes of memory, (3) a 2^{29}-step correlation attack that adapts a correlation procedure of Siegenthaler [35], and (4) an application of Hellman's [19] time-space tradeoff, which requires a short chosen-plaintext and a 2^{40}-step precomputation.

These attacks have differing advantages and requirements. Attack (3) requires a slight statistical weakness in the output function; the other attacks require no such weakness. Attacks (2) and (3) require ciphertext from one news article (a few thousand bytes); attack (1) requires only seven bytes of ciphertext from ASCII-encoded English; and attack (4) requires only approximately one dozen such bytes of ciphertext. We implement our time-space tradeoff to demonstrate one effective method for breaking Gifford's cipher.

[6] Gifford [13, p. 465] did not specify whether his experiments looked for cycles in the keystream or in the state stream.

Figure 2: A decomposition of Gifford's cipher. Our attacks search subfills corresponding to the invariant subspaces of dimensions 24, 5, 6, and 29 induced by the primary rational canonical decomposition R of the feedback matrix F. A similarity transformation P satisfying $F = P^{-1}RP$ maps each register fill s from the original world into four subfills $Ps = (d_0, d_1, d_2, d_3)$ in the decomposed world.

Combining attacks (2) and (3) yields an even faster attack: for example, the cryptanalyst could first search subspaces V_1 and V_2 with attack (3), and then search subspace V_3 with attack (2). This combined attack would require only approximately 2^{16} steps on average. For our implementation, we estimate this attack would run in less than one minute using eight Sparcstations.

5.1 Overview of Attacks

As shown in Figure 2, the matrix R decomposes the 64-bit shift register into four subregisters $\mathcal{R}_0$, $\mathcal{R}_1$, $\mathcal{R}_2$, $\mathcal{R}_3$, of lengths 24, 5, 6, and 29 bits, respectively. The similarity transformation P maps each register state into a corresponding sequence of four subregister states; thus, the key s_0 can be attacked in the four segments $Ps_0 = (d_0, d_1, d_2, d_3)$. Once any segment is known at any time, it is known for all future time: for each $0 \leq t < N$, it is true that $Ps_t = Pf^t(s_0) = (R_0^t d_0, R_1^t d_1, R_2^t d_2, R_3^t d_3)$, where R_0, R_1, R_2, R_3 are the four blocks of R. Furthermore, because R_0 is nilpotent, for all $t \geq 24$, $R_0^t d_0 = 0$. Therefore, for all practical purposes, Gifford's cipher uses only $5 + 6 + 29 = 40$ bits of key. Moreover, since R_1, R_2, and R_3 are nonsingular, knowing the state of $\mathcal{R}_i$ for any $1 \leq i \leq 3$ determines all previous states of $\mathcal{R}_i$.

Our attacks check candidate subfills in different ways. The exhaustive search attack maps an entire vector of candidate subfills back to the main register and checks if the resulting candidate plaintext appears valid [9, 10]; our time-space tradeoff optimizes this idea by hashing into a table derived from the ciphertext. Our adaptation of Siegenthaler's correlation attack separately checks each subfill by correlating its state sequence with the ciphertext stream. Finally, our application of Hellman's time-space tradeoff checks candidate subfills using precomputed tables based on a sequence of chosen-plaintext (we suggest using the seven-character string ".␣␣The␣", which appeared in every news report that we examined). For more details, see [5, 6].

5.2 Our Time-Space Tradeoff

Given $N = 2^n$ bytes of ciphertext from one news article, we determine the key s_0 by searching over the initial fills of subregisters $\mathcal{R}_1$, $\mathcal{R}_2$, and $\mathcal{R}_3$. Our attack exploits two tricks to carry out this search in only 2^{27} steps. First, we recover the high-order bit of each keystream byte because Gifford represented each plaintext byte in extended 8-bit ASCII, with the leading bit 0. We check each candidate key against these bits. Second, instead of searching over all 2^{29} initial states of $\mathcal{R}_3$, we search over only 2^{29-N} states of $\mathcal{R}_3$ by checking every Nth state of $\mathcal{R}_3$. Using a hashing technique, we check each candidate fill in expected constant time against all possible N positions of the ciphertext. This method works because every nondegenerate state sequence for $\mathcal{R}_3$ traverses the same cycle of $2^{29} - 1$ states, from different initial states. As an optimization, we precompute a table T_3 of every Nth state in this cycle. With $N = 2^{13} = 8192$, the total expected time for this search is $2^{5+6}2^{29-13} = 2^{27}$ steps.

The space usage is controlled by a parameter l, which also affects the collision and false-alarm rates. The hash table has 2^l slots (each of approximately $\lceil 2^{n-l} \rceil$ items), and the precomputed table T_3 requires 2^{29-n} entries (each of four bytes). Each address σ of the hash table points to a list of all positions in the ciphertext whose l consecutive high-order bits match the bit string σ. We select $n = 13$ and $l = 16$, for which our ciphertext-only attack runs in $2^{40-n} = 2^{27}$ steps on average and uses $1 \cdot 2^l + 4 \cdot 2^{29-n} \approx 2^{18}$ bytes of memory.

If the initial subfills for $\mathcal{R}_1$ and $\mathcal{R}_2$ are already known (say, by using a correlation attack), then our algorithm would run 2^{11} times faster.

5.3 Experimental Results

We implemented our time-space tradeoff attack on a loosely-coupled network of eight Sparcstations. On average, it takes approximately four hours to recover an initial fill from ciphertext alone. Including our library of linear algebra operations over $GF(2)$, our cryptanalytic engine comprises approximately 2,500 lines of C code.

6 Discussion

The feedback function bit-shifts bytes B_1 and B_7, using a sticky right-shift and a zero-fill left-shift, as explained in Section 2.1. We now analyze the effect of these operations on the period of F.

6.1 Sticky versus Non-Sticky Bit-Shifting

If the sticky shift of byte B_1 were replaced by a zero-fill shift, the feedback matrix would differ from F in exactly one bit: the upper-left bit of block F_1 would be zero rather than one (see Section 3.1). Let F' denote this modified matrix. Using methods described in Section 3.2, we computed the characteristic polynomial of F' to be $p_{F'}(x) = x^{64} + x^{56} + x^{54} + x^{52} + x^{50} + x^{48} + x^{44} + x^{40} + x^{24} = [x^{12}(x^2 + x + 1)(x^3 + x + 1)(x^6 + x^3 + 1)(x^9 + x^8 + x^7 + x^5 + x^4 + x^3 + 1)]^2$.

The maximum period of the state sequences generated by this variation of Gifford's cipher is the exponent of $p_{F'}$, which we shall now compute. The only non-primitive irreducible factor in $p_{F'}(x)$ is $x^6 + x^3 + 1$, whose exponent is 9. As for the other polynomials, $\exp(x^2+x+1) = 3$, $\exp(x^3+x+1) = 7$ and $\exp(x^9 + x^8 + x^7 + x^5 + x^4 + x^3 + 1) = 511$. By Theorem 1, $\exp(p_{F'}) = 2 \cdot \mathrm{lcm}(3, 7, 9, 511) =$ 9,198. Remarkably, changing one bit in the feedback matrix reduces the maximum period of Gifford's cipher from 349,502,963,061 states to only 9,198 states. This calculation is instructive because, originally, Gifford left to the compiler the decision whether to use a zero-fill right-shift versus a sticky right-shift.

6.2 Byte-Shifting Only

Regarding his choice for f to depend solely on bytes B_0, B_1, and B_7, Gifford [13, p. 465] explained that "the tap positions were chosen to yield the longest period that could be obtained" if the new byte were computed as $B_0 \oplus B_1 \oplus B_7$. We prove that Gifford's choice of taps does not achieve this objective.

Without any bit-shifting, the feedback function would be linear over $GF(2^8)$, and its characteristic polynomial (acting on bytes) would be $g(x) = x^8 + x^7 + x^6 + 1 = (x+1)(x^7 + x^5 + x^4 + x^3 + x^2 + x + 1)$. Since $\exp(x^7 + x^5 + x^4 + x^3 + x^2 + x + 1) = 127$, the longest state sequence produced by this simplified feedback function would be only 127 states (not $2^8 - 1 = 255$ states).

7 Conclusion

We have concretely demonstrated one effective method for breaking Gifford's cipher: given ciphertext from one news article, within approximately four hours on average, our implementation recovers the secret article key and corresponding master key used for one month. Thus, Gifford's cipher is not suitable for its intended use in broadcast encryption. Moreover, our work introduces a new powerful attack on filter generators, illustrates the power of algebraic decompositions, and provides an instructive detailed example of how to apply linear algebra over $GF(2)$ in cryptanalysis.

To improve Gifford's cipher, it would be desirable to use a longer register and to use more carefully chosen tap positions than did Gifford. In addition, it would be helpful to add more complexity to the encryption process. For example, some cryptographers incorporate nonlinear feedback functions into their designs. But these simple modifications do not guarantee security. As we show, what appears to be an intractable cryptanalytic problem can be computationally feasible when attacked with appropriate mathematical machinery.

Acknowledgments

We thank Robert W. Baldwin, Xuejia Lai, Rainer A. Rueppel, and Richard Stein for editorial comments. All computer work was carried out on workstations at the University of Maryland Baltimore County.

References

[1] Beker, Henry; and Fred Piper, *Cipher Systems: The Protection of Communications*, John Wiley (New York, 1982).

[2] Berlekamp, Elwyn R., *Algebraic Coding Theory*, Aegean Park Press (Laguna Hills, CA, 1984).

[3] Brickell, Ernest F.; and Andrew M. Odlyzko, "Cryptanalysis: A survey of recent results" in [36], Chapter 10 (1992), 501–540.

[4] Cain, Thomas R., "How to break Gifford's cipher," CMSC-693 Project, Computer Science Department, University of Maryland Baltimore County (May 28, 1993). 57 pages.

[5] Cain, Thomas R.; and Alan T. Sherman, "How to break Gifford's cipher" (June 2, 1994), submitted to *Cryptologia*. Available as Technical Report CS TR-94-07, University of Maryland Baltimore County. 49 pages.

[6] Cain, Thomas R.; and Alan T. Sherman, "Cryptanalysis of filter generators using the rational canonical decomposition of the feedback function" (1994), in preparation.

[7] Dawson, Ed, "Linear feedback shift registers and stream ciphers" in [26], Chapter 8 (1990), 106–119.

[8] Dawson, Ed; and Andrew Clark, "Divide and conquer attacks on certain classes of stream ciphers," *Cryptologia*, **XVIII**: 1 (January 1994), 25–40.

[9] Ganesan, Ravi; and Alan T. Sherman, "Statistical techniques for language recognition: An introduction and guide for cryptanalysts," *Cryptologia*, **XVII**:4 (October 1993), 321–366.

[10] Ganesan, Ravi; and Alan T. Sherman, "Statistical techniques for language recognition: An empirical study using real and simulated English" (September 27, 1993), *Cryptologia*, to appear.

[11] Giesbrecht, Mark, "Fast algorithms for matrix normal forms" in *Proceedings of the 33rd Annual Symposium on Foundations of Computer Science*, ACM Press (1992), 121–130.

[12] Gifford, David K.; Dawn Heitmann; David A. Segal; Robert G. Cote; Kendra Tanacea; and David E. Burmaster, "Boston Community Information System 1986 experimental test results," technical report MIT/LCS/TR-397, MIT Laboratory for Computer Science (August 1987).

[13] Gifford, David K.; John M. Lucassen; and Stephen T. Berlin, "The application of digital broadcast communication to large scale information systems," *IEEE Journal on Selected Areas in Communications*, **SAC-3**:3 (May 1985), 457–467.

[14] Gifford, David K.; and David Andrew Segal, "Boston Community Information System 1987–1988 experimental test results," technical report MIT/LCS/TR-422, MIT Laboratory for Computer Science (May 1989).

[15] Gill, Arthur, *Linear Sequential Circuits: Analysis, Synthesis, and Applications*, McGraw-Hill (New York, 1966).

[16] Golić, Jovan Dj.; and Miodrag J. Mihaljević, "A generalized correlation attack on a class of stream ciphers based on the Levenshtein distance," *Journal of Cryptology*, **3**:3 (1991), 201–212.

[17] Gollmann, Dieter; and William G. Chambers, "Clock-controlled shift registers: A review," *IEEE Journal on Selected Areas in Communications*, **7**:4 (May 1989), 525–533.

[18] Golomb, Solomon, *Shift Register Sequences*, Aegean Park Press (Laguna Hills, CA, 1982).

[19] Hellman, Martin E., "A cryptanalytic time-memory trade-off," *IEEE Transactions on Information Theory*, **IT-26**:4 (1980), 401–406.

[20] Hoffman, Kenneth; and Ray Kunze, *Linear Algebra*, second edition, Prentice-Hall (1971).

[21] Hungerford, Thomas W., *Algebra*, Springer-Verlag (New York, 1974).

[22] Jacob, Bill, *Linear Algebra*, W. H. Freeman and Company (New York, 1990).

[23] Key, Edwin L., "An analysis of the structure and complexity of nonlinear binary sequence generators," *IEEE Transactions on Information Theory*, **IT-22**:6 (November 1976), 732–736.

[24] Klapper, Andrew, "The vulnerability of geometric sequences based on fields of odd characteristic," *Journal of Cryptology*, **7**:1 (winter 1994), 33–51.

[25] Knuth, Donald E., *Seminumerical Algorithms* in *The Art of Computer Programming*, Vol. 2, second edition, Addison-Wesley (Reading, MA, 1981).

[26] Loxton, J. H., ed., *Number Theory and Cryptography*, Cambridge University Press, London Mathematical Society Lecture Note Series, No. 154 (Cambridge, Great Britain, 1990).

[27] *Macsyma Reference Manual*, version ten, Mathlab Group, MIT Laboratory for Computer Science (January 1983).

[28] Marsh, R. W., *Table of Irreducible Polynomials Over GF*(2) *Through Degree* 19, National Security Agency (Washington D.C., 1957).

[29] Meier, Willi; and Othmar Stafelbach, "Fast correlation attacks on certain stream ciphers," *Journal of Cryptology*, **1**:3 (1989), 159–176.

[30] Peterson, W. Wesley; and E. J. Weldon, *Error-Correcting Codes*, MIT Press (Cambridge, Mass. 1972).

[31] Rhee, Man Young, *Cryptography and Secure Communications*, McGraw-Hill (Singapore, 1994).

[32] Ronse, Christian, *Feedback Shift Registers*, Lecture Notes in Computer Science 169, G. Goos and J. Hartmanis, eds., Springer-Verlag (Berlin, 1984).

[33] Rueppel, Rainer A., *Analysis and Design of Stream Ciphers*, Springer-Verlag (New York, 1986).

[34] Rueppel, Rainer A., "Stream ciphers" in [36], Chapter 2 (1992), 65–134.

[35] Siegenthaler, T., "Decrypting a class of stream ciphers using ciphertext only," *IEEE Transactions on Computers*, **C-34**:1 (January 1985), 81–85.

[36] Simmons, Gustavus J., editor, *Contemporary Cryptology: The Science of Information Integrity*, IEEE Press (Piscataway, NJ, 1992).

[37] Zeng, Kencheng; Chung-Huang Yang; Dah-Yea Wei; and T. R. N. Rao, "Pseudorandom bit generators in stream-cipher cryptography," *Computer*, **24**:2 (February 1991), 8–17.

Appendix A: Proof of Proposition 1

Proposition 1 computes the exponents of the elementary divisors of F. To prove this proposition, it is useful to review some basic concepts from finite field theory, including the notion of a cyclotomic polynomial.

Let n be any positive integer. The *nth roots of unity* are the roots of the polynomial $x^n - 1$. These roots form a multiplicative cyclic group. If ζ is an nth root of unity that generates this group, then ζ is said to be a *primitive nth root of unity*. The *nth cyclotomic polynomial* $C_n(x)$ is the monic polynomial $C_n(x) = \prod_{i=1}^{r}(x - \zeta_i)$, where $\zeta_1, \zeta_2, \ldots, \zeta_r$ are the distinct primitive n^{th} roots of unity.

Cyclotomic polynomials are useful in computing exponents. Let $f(x) \in \mathbb{Z}_2[x]$ be any irreducible polynomial. Since all roots of $f(x)$ over $\mathbb{Z}_2$ have the same order in any extension field of $\mathbb{Z}_2$, it is true that $\exp(f)$ is the order of its roots in that extension field. Therefore, if $f(x)$ divides $C_k(x)$ for some cyclotomic polynomial $C_k(x)$, it follows that $\exp(f) = k$.

To prove Proposition 1, we also apply the following two well-known lemmas.

Lemma 1. *Let n be any positive integer. If $S = \{f(x) : f(x)$ is an irreducible polynomial over $\mathbb{Z}_2$ of degree dividing $n\}$, then $x^{2^n} + x = \prod_{f \in S} f(x)$.*
Proof. See Berlekamp [2, p. 103]. ■

Lemma 2. *Let $n \in \mathbb{Z}^+$ and let $C_d(x)$ be the dth cyclotomic polynomial over $\mathbb{Z}_2$. If $2 \not| \, n$, then $x^n + 1 = \prod_{d|n} C_d(x)$.*
Proof. See Berlekamp [2, p. 91]. ■

Proposition 1. *For each $0 \leq i \leq 3$, let $e_i = \exp(m_i)$ be the exponent of the elementary divisor $m_i(x)$ of F defined in Section 3.2. It is true that $e_0 = 0$, $e_1 = 31$, $e_2 = 21$, and $e_3 = 2^{29} - 1$. Consequently, $m_1(x)$ and $m_3(x)$ are primitive polynomials, but $m_0(x)$ and $m_2(x)$ are not primitive.*
Proof. We compute the exponent of each elementary divisor separately.

$m_0(x)$ is the polynomial x^{24}. Since x does not divide $x^n - 1$ for any positive integer n, it follows that $e_0 = 0$ and $m_0(x)$ is not primitive.

$m_1(x)$ is a degree 5 irreducible polynomial. We find the unique cyclotomic polynomial $C_k(x)$ such that m_1 divides $C_k(x)$ to establish that $e_1 = k$. By Lemma 1, m_1 divides $x^{2^5} + x = x(x^{31} + 1)$; and by Lemma 2, $x^{31} + 1 = C_1(x)\,C_{31}(x)$. Because $C_1(x) = x + 1$, it follows that $m_1(x)$ divides $C_{31}(x)$. Therefore, $e_1 = 31$ and $m_1(x)$ is primitive.

$m_2(x)$ is a degree 6 irreducible polynomial; therefore, $e_2 \leq 2^6 - 1 = 63$. By Lemma 2, $x^{63} + 1 = C_1(x)\,C_3(x)\,C_7(x)\,C_9(x)\,C_{21}(x)\,C_{63}(x)$. It is easy to verify that $m_2(x)$ divides $C_{21}(x)$, but $m(x)$ does not divide $C_{63}(x)$, $C_9(x)$, or $C_7(x)$. Hence, $e_2 = 21$ and $m_2(x)$ is not primitive.

$m_3(x)$ is a degree $d = 2^{29} - 1$ polynomial; therefore, $e_3 | d$. We will prove that $e_3 = d$ by showing that $m_3(x)$ is primitive. Let α be any root of $m_3(x)$. To prove that $m_3(x)$ is primitive, it suffices to verify that $\alpha^r \neq 1$ for all $r < d$ such that $r|d$. Since $2^{29} - 1 = 233 \cdot 1103 \cdot 2089$ is the prime factorization of d, only six r must be checked. To carry out this verification, we implemented and ran an algorithm in Appendix C of Peterson [30]. Thus, $m_3(x)$ is irreducible and primitive and $e_3 = 2^{29} - 1$. ■

As a partial check of our calculations, note that Marsh [28] also lists the exponents of $m_1(x)$ and $m_2(x)$ as 31 and 21, respectively. We could not find any table that lists $m_3(x)$.