J. Cryptology (1988) 1: 3–36

Journal of Cryptology

Is the Data Encryption Standard a Group? (Results of Cycling Experiments on DES)[1]

Burton S. Kaliski Jr., Ronald L. Rivest, and Alan T. Sherman[2]

MIT Laboratory for Computer Science, 545 Technology Square, Cambridge, MA 02139, U.S.A.

Abstract. The Data Encryption Standard (DES) defines an indexed set of permutations acting on the message space $\mathcal{M} = \{0, 1\}^{64}$. If this set of permutations were closed under functional composition, then the two most popular proposals for strengthening DES through multiple encryption would be equivalent to single encryption. Moreover, DES would be vulnerable to a known-plaintext attack that runs in 2^{28} steps on the average. It is unknown in the open literature whether or not DES has this weakness.

Two statistical tests are presented for determining if an indexed set of permutations acting on a finite message space forms a group under functional composition. The first test is a "meet-in-the-middle" algorithm which uses $O(\sqrt{K})$ time and space, where K is the size of the key space. The second test, a novel cycling algorithm, uses the same amount of time but only a small constant amount of space. Each test yields a known-plaintext attack against any finite, deterministic cryptosystem that generates a small group.

The cycling closure test takes a pseudorandom walk in the message space until a cycle is detected. For each step of the pseudorandom walk, the previous ciphertext is encrypted under a key chosen by a pseudorandom function of the previous ciphertext. Results of the test are asymmetrical: long cycles are overwhelming evidence that the set of permutations is not a group; short cycles are strong evidence that the set of permutations has a structure different from that expected from a set of randomly chosen permutations.

Using a combination of software and special-purpose hardware, the cycling closure test was applied to DES. Experiments show, with overwhelming confidence, that DES is not a group. Additional tests confirm that DES is free of certain other gross algebraic weaknesses. But one experiment discovered fixed points of the so-called "weak-key" transformations, thereby revealing a previously unpublished additional weakness of the weak keys.

Key words. Birthday Paradox, Closed cipher, Cryptanalysis, Cryptology, Cryptography, Cycle-detection algorithm, Data Encryption Standard (DES), Finite permutation group, Idempotent cryptosystem, Multiple encryption, Pure cipher, Weak keys.

[1] Support for this research was provided in part by the National Science Foundation under contract number MCS-8006938 and by the International Business Machines Corporation.

[2] Address: Department of Computer Science, Tufts University, Medford, MA 02155, USA.

1. Introduction

On November 23, 1976, the United States National Bureau of Standards adopted the Data Encryption Standard (DES) as a federal standard for the cryptographic protection of computer data [12], [61].[3] Although the National Security Agency has withdrawn its support of DES, many banks and other organizations continue to use DES to protect unclassified data [34]. Despite its widespread use, numerous fundamental questions about the standard remain unanswered in the open literature. In this paper we address one such important question: "Is the set of DES transformations closed under functional composition?"

DES defines an indexed set of permutations acting on the message $\mathcal{M} = \{0, 1\}^{64}$. There are $M = 2^{64}$ messages and $K = 2^{56}$ keys. Each key k represents a transformation T_k, with inverse T_k^{-1}. Let $\mathcal{K} = \{1, 0\}^{56}$ denote the set of keys.

It is important to know whether or not DES is closed since, if DES were closed, it would have the following two weaknesses. First, both sequential multiple encryption and Tuchman's multiple encryption scheme—the two most popular proposals for strengthening DES through using multiple encryption—would be equivalent to single encryption.[4] That is, if DES were closed, then for every three keys i, j, k there would exist keys r, s such that $T_i T_j(x) = T_r(x)$ and $T_i T_j^{-1} T_k(x) = T_s(x)$ for all messages x. Even worse, DES would be vulnerable to a known-plaintext attack that runs in 2^{28} steps, on the average. Each weakness follows from the fact that the set of cryptographic transformations of any closed cipher forms a group under functional composition. For similar reasons, it is important to know if DES is pure.[5] Although most researchers believe DES is neither closed nor pure, no one has proven this conjecture in the open literature.

To determine whether DES is a group, we developed two statistical tests. The first test is based on a "meet-in-the-middle" strategy and takes $O(\sqrt{K})$ time and space. The second test follows a pseudorandom walk in the message space until a cycle is detected, using $O(\sqrt{K})$ time and constant space. Each test yields a known-plaintext attack against any group cipher. Although we focus on DES, the methods presented here work for any finite, deterministic cryptosystem.

Using a combination of software and special-purpose hardware, we applied the cycling closure test and other related tests to DES. None of our experiments involving randomly chosen DES transformations detected any algebraic weaknesses. In particular, our data show with extremely high confidence that DES is neither closed nor pure. However, one experiment unexpectedly discovered fixed points for two of the so-called "weak-key" transformations [9], thereby revealing a previously unpublished additional weakness of the weak keys.

The body of this paper is organized in five sections. Section 2 presents a brief

---

[3] We expect the reader to be familiar with the fundamentals of cryptology (as presented in [15] or [1], for example), as well as with the basics of DES (as described in [12] or [38], for example).

[4] To encrypt a message x using *sequential multiple encryption* is to compute $T_i T_j(x)$, where the keys i and j are chosen independently. Similarly, to encrypt a message x under *Tuchman's scheme* is to compute $T_i T_j^{-1} T_k(x)$, where the keys i, j, and k are independently chosen [52], [38], [37].

[5] DES is *pure* if and only if, for every three keys i, j, k, there exists some key l such that $T_i T_j^{-1} T_k = T_l$ [48]. See Section 3.1.

overview of the cycling closure test. Section 3 presents some background information useful to understanding our results. Section 4 describes the cycling closure test and other algebraic tests in detail. Section 5 explains how the meet-in-the-middle and cycling closure tests can be modified into known-plaintext attacks against group ciphers. Section 6 summarizes and interprets our experimental results. An appendix, which gives additional detailed descriptions of our experiments, is also included.

2. An Overview of the Cycling Closure Test

This section summarizes how we applied the cycling closure test to DES to determine if DES is closed.

Let x_0 be any message and consider the set S_{x_0} recursively defined as follows: x_0 is an element of S_{x_0}, and, for any key k and any message $x \in S_{x_0}$, $T_k(x)$ is also an element of S_{x_0}. Thus, S_{x_0} is the set of messages that can be reached through multiple encrypting x_0 zero or more times with arbitrary keys.

If DES acted like a set of randomly chosen permutations, then we would expect $S_{x_0} = \mathcal{M}$ and thus $|S_{x_0}| = M = 2^{64}$. However, if DES were closed, then $|S_{x_0}| \leq K = 2^{56}$, since sequential multiple encryption would be equivalent to single encryption and there are at most K distinct encryption transformations. The cycling closure test computes a statistic based on the size of S_{x_0}.

The cycling closure test picks an initial message x_0 at random and then takes a pseudorandom walk in S_{x_0}, beginning at x_0. For each step of the pseudorandom walk, the previous ciphertext is encrypted under a key chosen by a pseudorandom function of the previous ciphertext. The walk continues until a cycle is detected. By the "Birthday Paradox" (see Section 3.4), the walk is expected to cycle after approximately $|S_{x_0}|^{1/2}$ steps.

More specifically, the test computes a sequence of messages $x_0, x_1, \ldots$. For each $i \geq 0$, the next message x_{i+1} is computed by

$$x_{i+1} = f_\rho(x_i), \tag{1}$$

where the function $f_\rho: \mathcal{M} \to \mathcal{M}$ is defined by

$$f_\rho(x) = T_{\rho(x)}(x) \tag{2}$$

for all messages $x \in \mathcal{M}$. The walk is guided by a deterministic, pseudorandom function $\rho: \mathcal{M} \to \mathcal{K}$ that maps messages to keys. If ρ is "random," then f_ρ acts like a random function on S_{x_0}.

Since S_{x_0} is finite, the walk will eventually encounter the same message twice. Thereafter, the walk will remain periodic because f_ρ is deterministic. Let λ be the least integer such that $x_\lambda = x_i$ for some $0 \leq \lambda < i$, and let μ be the least positive integer such that $x_{\lambda+\mu} = x_\lambda$. The walk is completely determined by the *leader* $x_0, x_1, \ldots, x_{\lambda-1}$ and the *cycle* $x_\lambda, x_{\lambda+1}, \ldots, x_{\lambda+\mu}$. The integers λ and μ are called, respectively, the *leader length* and *cycle length* of the sequence $x_0, x_1, \ldots$. See Fig. 1. To detect cycles and to compute cycle and leader lengths, we used a variation of the Sedgewick–Szymanski cycle detection algorithm [27], [49].

Results of the test are asymmetrical. Walks significantly longer than $\sqrt{K} = 2^{28}$

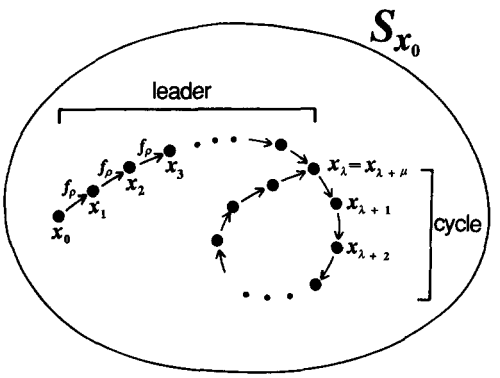

Fig. 1. The cycling closure test takes a pseudorandom walk in the message space.

are strong evidence that DES is not a group. Walks significantly shorter than $\sqrt{M} = 2^{32}$ are strong evidence that DES has a structure different from that expected from a set of randomly chosen permutations.

Our experiments detected cycles after approximately 2^{33} steps, giving overwhelming evidence that DES is not a group.

3. Background

This section presents background material helpful in understanding the rest of this paper. Section 3.1 introduces the notion of a finite, deterministic cryptosystem and explains some terminology used throughout the paper. Section 3.2 discusses the *a priori* chance that DES is a group. Section 3.3 summarizes several important differences between closed ciphers and ciphers that consist of randomly chosen permutations. Section 3.4 reviews the so-called "Birthday Paradox," and Section 3.5 surveys previous work on DES relevant to this paper.

3.1. *Definitions and Notation*

A (*finite, deterministic*) *cryptosystem* is an ordered 4-tuple $(\mathscr{K}, \mathscr{M}, \mathscr{C}, T)$, where $\mathscr{K}$, $\mathscr{M}$, and $\mathscr{C}$ are finite sets called the *key space*, *message space*, and *ciphertext space*, and $T: \mathscr{K} \times \mathscr{M} \to \mathscr{C}$ is a transformation such that, for each $k \in \mathscr{K}$, the mapping $T_k = T(k, \cdot)$ is invertible.

The *order* of a cryptosystem is the number of distinct transformations; the *degree* of a cryptosystem is the size of the message space. A cryptosystem is *endomorphic* if and only if the message space and ciphertext space are the same set.

Thus, for any cryptosystem $(\mathscr{K}, \mathscr{M}, \mathscr{C}, T)$, each key $k \in \mathscr{K}$ represents a transformation $T_k: \mathscr{M} \to \mathscr{C}$. In an endomorphic cryptosystem, each key represents a permutation on $\mathscr{M}$. A cryptosystem is *faithful* if and only if every key represents a distinct transformation.

For any cryptosystem $\Pi = (\mathscr{K}, \mathscr{M}, \mathscr{C}, T)$, let $\mathscr{T}_\Pi = \bigcup \{T_k: k \in \mathscr{K}\}$ be the set of all encryption transformations, and, whenever Π is endomorphic, let $G_\Pi = \langle \mathscr{T}_\Pi \rangle$ be

the group generated by $\mathscr{T}_\Pi$ under functional composition. For any transformation $T_k \in \mathscr{T}_\Pi$, let T_k^{-1} denote the inverse of T_k. In addition, let $K = |\mathscr{K}|$ be the size of the key space; let $M = |\mathscr{M}|$ be the degree of Π; and let $m = |\mathscr{T}_\Pi|$ be the order of Π. Whenever the meaning is clear, we will omit the subscript Π.

Let I be the identity permutation on $\mathscr{M}$, and let $\mathscr{A}_\mathscr{M}$ and $\mathscr{S}_\mathscr{M}$ be, respectively, the *alternating group* and *symmetric group* on $\mathscr{M}$. For any permutations g, h on M we denote the composition of g and h by $gh = g[h(\cdot)]$. For any permutations $g_1, g_2, \ldots, g_n$, let $\langle g_1, g_2, \ldots, g_n \rangle$ denote the group generated by $g_1, g_2, \ldots, g_n$ under functional composition.

To analyze the closure tests it is useful to introduce the following standard terminology from permutation group theory [5], [46], [53]. Let G be any subgroup of $\mathscr{S}_\mathscr{M}$, and let x be any message in $\mathscr{M}$. The *order of* G is the number of elements in G; the *degree of* G is the cardinality of $\mathscr{M}$. For any $g \in \mathscr{S}_\mathscr{M}$, the *order of* g is the order of $\langle g \rangle$.

The *G-orbit of x* is the set $G\text{-orbit}(x) = \{g(x): g \in G\}$. For any permutation $g \in \mathscr{S}_\mathscr{M}$, we will write $g\text{-orbit}(x)$ to denote the $\langle g \rangle$-orbit of x. If f is any function (not necessarily a permutation) and if $x \in \text{Domain}(f)$, we define the *f-closure of x* to be the set $f\text{-closure}(x) = \{f^i(x): i \geq 0\}$.

The *G-stabilizer of x* is the set $H_x = \{g \in G: g(x) = x\}$, which forms a subgroup of G.

For any subset of permutations $S \subseteq \mathscr{S}_\mathscr{M}$ and for any subset of messages $X \subseteq \mathscr{M}$, we say *S acts transitively on X* if and only if, for every pair of messages $x, y \in X$, there exists some transformation $g \in S$ such that $g(x) = y$.

Let $\Pi = (\mathscr{K}, \mathscr{M}, \mathscr{C}, T)$ be any finite deterministic cryptosystem. Π is *closed* if and only if its set of encryption transformations is closed under functional composition, i.e., if and only if for every two keys $i, j \in \mathscr{K}$ there exists a key $k \in \mathscr{K}$ such that $T_i T_j = T_k$.[6] Since every finite cancellation semigroup is a group [46], Π is closed if and only if $\mathscr{T}_\Pi$ forms a group under functional composition.

Shannon's notion of a pure cipher generalizes the idea of closure to nonendomorphic cryptosystems [48]. Π is *pure* if and only if, for every three keys $i, j, k \in \mathscr{K}$, there exists a key $l \in \mathscr{K}$ such that $T_i T_j^{-1} T_k = T_l$.[7]

Thus, Π is pure if and only if for every $T_0 \in \mathscr{T}_\Pi$ the set $T_0^{-1}\mathscr{T}_\Pi$ is closed. But for any $T_0 \in \mathscr{T}_\Pi$, $T_0^{-1}\mathscr{T}_\Pi$ is closed if and only if $T_0^{-1}\mathscr{T}_\Pi$ forms a group under functional composition. Hence, $T_0^{-1}\mathscr{T}_\Pi$ is closed *for every* $T_0 \in \mathscr{T}_\Pi$ if and only if $T_0^{-1}\mathscr{T}_\Pi$ is closed *for some* $T_0 \in \mathscr{T}_\Pi$. Every closed cryptosystem is pure, but not every endomorphic pure cryptosystem is closed.[8]

For any string $s \in \{0, 1\}^+$, let $\bar{s}$ denote the bitwise complement of s.

DES defines a particular endomorphic cryptosystem with $\mathscr{M} = \mathscr{C} = \{0, 1\}^{64}$ and $\mathscr{K} = \{0, 1\}^{56}$. Because DES has degree 2^{64}, but order at most 2^{56}, DES is

[6] Note that we are using the term *closed cipher* to refer to what Shannon called an *idempotent cipher* [48]. Shannon defined a closed cipher to be any cryptosystem with the property that each cryptographic transformation is surjective.

[7] Shannon also required each transformation of a pure cipher to be equally likely.

[8] The restriction of simple substitution [19] on the standard alphabet where the letter "A" is always mapped to "B" is an endomorphic system that is pure but not closed.

intransitive. It is unknown if DES is faithful, closed, or pure. It is also unknown if any DES transformation is the identity permutation.

3.2. *Is DES a Group?—A Priori Beliefs*

The question of whether or not DES is closed is a question about the order of the group generated by DES. Grossman and Coppersmith observed that $G_{\text{DES}} \subseteq \mathscr{A}_{\mathscr{M}}$ [8], but no one has disproved the possibility that $G_{\text{DES}} = \mathscr{T}_{\text{DES}}$.[9]

There are several reasons to suspect DES is not closed. First, Coppersmith and Grossman proved "DES-like" permutations generate the alternating group [8].[10] Second, if even just two permutations are chosen at random from $\mathscr{S}_{\mathscr{M}}$, then there is an overwhelming chance (greater than $1 - e^{-\sqrt{M}}$) that these permutations generate either $\mathscr{A}_{\mathscr{M}}$ or $\mathscr{S}_{\mathscr{M}}$ [3], [16]. Third, no one has announced finding any three keys $i, j, k \in \mathscr{K}$ such that $T_k = T_i T_j$. Finally, according to a 1977 unclassified summary of a report of the Senate Select Committee on Intelligence, the National Security Agency certified that "the final DES algorithm was, to the best of their knowledge, free of any statistical or mathematical weaknesses" [64].

On the other hand, DES is not a set of randomly chosen permutations, and Coppersmith and Grossman did not prove that DES generates $\mathscr{A}_{\mathscr{M}}$. Furthermore, DES is known to have the following three regularities [12], [38], [9], [24]:

1. *Complementation property.* For every key k and every message x, $T_{\bar{k}}(\bar{x}) = \overline{T_k(x)}$.
2. *Existence of weak keys.* There exist at least four distinct keys k such that $T_k^2 = I$.
3. *Existence of semiweak keys.* There exist at least six distinct pairs of keys $k_1 \neq k_2$ such that $T_{k_2} T_{k_1} = I$.

The last two properties, however, apparently involve only a small fraction of the total number of DES transformations. Although many people may have a strong belief that DES is not closed, there is a need for convincing objective evidence to answer this question.

3.3. *Algebraic Properties of Closed and Random Ciphers*

In this section we review several important differences between closed cryptosystems and cryptosystems that consist of randomly chosen permutations.[11] These differences form the basis of the statistical closure tests.[12]

Since every finite cancellation semigroup is a group, any endomorphic cryptosystem is closed if and only if its set of encryption transformations forms a group under functional composition. Thus, closed ciphers have a great deal of algebraic structure. By contrast, one expects a set of randomly chosen permutations to have virtually no algebraic structure, as the following lemmas makes precise.

---

[9] To see that $G_{\text{DES}} \subseteq \mathscr{A}_{\mathscr{M}}$, note that each round of DES is an even permutation.

[10] See [21] for an extension of this result.

[11] By "a set of randomly chosen permutations on $\mathscr{M}$," we mean a set of permutations each member of which is chosen independently, with uniform probability from $\mathscr{S}_{\mathscr{M}}$.

[12] This section draws heavily from basic results in permutation group theory and from Shannon's classic paper [48], [36].

Properties of cryptosystems can be studied both by examining abstractly the set of encryption transformations and by examining how the transformations act on the message space. Lemma 3.1 captures one important difference between closed and random ciphers by focusing on a property of the set of encryption transformations. This lemma says that if a cryptosystem is closed, then for every transformation T_k there are many pairs T_i, T_j such that $T_k = T_i T_j$; but, if a cryptosystem consists of randomly chosen permutations, then for every transformation T_k it is unlikely to find any pair T_i, T_j such that $T_k = T_i T_j$. This lemma provides the basis of the meet-in-the-middle closure test.

Lemma 3.1. *Let $\Pi = (\mathcal{K}, \mathcal{M}, \mathcal{M}, T)$ be any endomorphic cryptosystem of order m, and let $k \in \mathcal{K}$ be any key. If Π is closed, then there are exactly m pairs of keys $T_i, T_j \in \mathcal{T}_\Pi$ such that $T_i T_j = T_k$. If $\mathcal{T}_\Pi$ is selected at random from $\mathcal{S}_\mathcal{M}$, then the expected number of pairs of transformations $T_i, T_j \in \mathcal{T}_\Pi$ such that $T_i T_j = T_k$ is $m^2/M!$.*

Proof. Part 1: assume Π is closed. For every transformation $T_i \in \mathcal{T}_\Pi$, there is exactly one transformation $T_j \in \mathcal{T}_\Pi$ such that $T_i T_j = T_k$. Part 2: assume $\mathcal{T}_\Pi$ is chosen at random. There are m^2 pairs T_i, $T_j \in \mathcal{T}_\Pi$ and each pair has a $1/|\mathcal{S}_\mathcal{M}|$ chance of corresponding to T_k. Moreover, these probabilities are independent. □

For unfaithful cryptosystems, it is important to distinguish between drawing a transformation from the set of transformations and picking a representation of a transformation from the key space. Mathematically, it is usually more convenient to think about selecting a transformation from a set of transformations, but in practice, one must often select a transformation by choosing a key. Let $\mathcal{T}$ be the set of cryptographic transformations in any cryptosystem with key space $\mathcal{K}$. If T_k is selected from $\mathcal{T}$ at random, then the probability of picking any particular transformation in $\mathcal{T}$ is exactly $1/m$, where $m = |\mathcal{T}|$. However, if a key k is selected at random from $\mathcal{K}$, then the probability that k represents any particular transformation in $\mathcal{T}$ is between $1/m$ and $1/K$, where $K = |\mathcal{K}|$. If the cryptosystem is unfaithful, then $m < K$.

The next lemma describes the structure imposed on the message space by any closed cipher; specifically, Lemma 3.2 says that the orbits of any closed cipher partition the message space into transitive sets. This lemma provides the basis of the cycling closure test.

Lemma 3.2. *Let $\Pi = (\mathcal{K}, \mathcal{M}, \mathcal{M}, T)$ be any endomorphic cryptosystem of order m. If Π is closed, then, for some $1 \leq r \leq m$, the $\mathcal{T}_\Pi$-orbits of messages in $\mathcal{M}$ partition $\mathcal{M}$ into r mutually disjoint sets $\mathcal{M} = B_1 \cup \cdots \cup B_r$ such that, for each $1 \leq i \leq r$, the following two statements hold:*

1. *$\mathcal{T}_\Pi$ acts transitively on B_i.*
2. *$|B_i|$ divides m; in fact, for any $x \in B_i$, $|B_i| = m/|H_x|$, where H_x is the $\mathcal{T}_\Pi$-stabilizer of x.*

Proof. (Sketch) For each $x \in \mathcal{M}$, consider the left cosets of H_x in $\mathcal{T}$ [46]. □

Corollary 3.3. *If DES is closed, then DES partitions its message space into at least* 2^8 *mutually disjoint transitive sets, each of size at most* 2^{56}.

Proof. DES has degree 2^{64}, but order at most 2^{56}. □

The next lemma calculates the expected number of spurious decipherments of closed and random ciphers; this lemma is useful in the analysis of the tests.

Lemma 3.4. *Let* $\Pi = (\mathcal{K}, \mathcal{M}, \mathcal{M}, T)$ *be any endomorphic cryptosystem of order* m, *let* $p \in \mathcal{M}$ *be any message, let* $k \in \mathcal{K}$ *be any key, and let* $c = T_k(p)$. *If* Π *is closed, then the number of transformations that map* p *to* c *is* $m/|B_p| = |H_p|$, *where* B_p *is the* $\mathcal{T}_\Pi$*-orbit of* p, *and* H_p *is the* $\mathcal{T}_\Pi$*-stabilizer of* p. *If* $\mathcal{T}_\Pi$ *is chosen at random, then the expected number of transformations that map* p *to* c *is* m/M.

Proof. Part 1: (sketch) by Lemma 3.2 and the fact that, for any x, $y \in B_p$, $|\{T_i \in \mathcal{T}_\Pi\colon T_i(x) = y\}| = |\{T_i \in \mathcal{T}_\Pi\colon T_i(p) = c\}|$. Note that $|H_p| = |H_c|$. Part 2: each transformation in $\mathcal{T}_\Pi$ other than T_k maps p to c with probability $1/M$. □

3.4. *The Birthday Paradox*

This section briefly reviews the "Birthday Paradox" [18], which plays a dominant role in the analysis of the closure tests. The Birthday Paradox involves the question, "If r people are selected at random, what is the chance that no two people will have the same birthday?" Let p_r be this chance. If birthdays are independently and uniformly distributed between 1 and m, then

$$p_r = \frac{(m)_r}{m^r} = \frac{m!}{m^r(m-r)!} \approx e^{-r^2/(2m)}, \tag{3}$$

where $(m)_r = m(m-1)\cdots(m-r+1)$. (The approximation in equation (3), and other similar approximations, can be obtained from Stirling's formula [18], [43].) The "paradox" is that many students are surprised to learn that the probability p_r is so low: with only $r = \sqrt{m}$ people, the chance is approximately 0.5 that at least two people will have the same birthday. More precisely, for any constant $c > 0$, if $r = c\sqrt{m}$ and m is sufficiently large, then $p_r \approx e^{-c^2/2}$. Thus, by choosing $r = c\sqrt{m}$ with c sufficiently large, p_r can be made as small as desired.

The meet-in-the-middle test uses a variation of the Birthday Paradox in which two samples X and Y, each of size r, are drawn at random from a universe of m elements. If X and Y each are drawn without replacement, and if each element is drawn independently with probability $1/m$, then the chance that X and Y do not intersect is exactly $(m)_{2r}/((m)_r)^2$. If $r = c\sqrt{m}$, this chance is approximately e^{-3c^2}.

3.5. *Previous Cycling Studies on DES*

To the best of our knowledge, only three other cycling experiments on DES have been reported in the open literature. These experiments were performed by Gait,

Davies and Parkin, and Hellman and Reyneri. Each of these experiments differs from our cycling closure test, and none of these previous experiments determined if DES generates a small group.

The analysis of each of these previous experiments depends heavily on the following two facts [23], [44], [32, exercise 3.1.12]. Let $x_0 \in \mathcal{M}$ be any message. For a randomly selected *function* f on $\mathcal{M}$, the expected size of f-closure(x_0) is approximately $\sqrt{M}$. (This follows from the Birthday Paradox.) But for a randomly selected *permutation* g on $\mathcal{M}$, the expected size of g-orbit(x_0) is approximately $M/2$. (See Section 4.4.3.)

Gait [20] investigated the statistical properties of pseudorandom key streams produced by DES in output-feedback mode [62]. Provided the feedback width is exactly 64 bits, each such key stream describes the orbit of a DES transformation on some initial message. In a series of software experiments, Gait computed the key stream produced by DES in output-feedback mode to at most $10^6 \approx 2^{20}$ places. Gait found no cycles for nonweak keys.[13] Gait did not state what feedback width he used. Gait also proposed a new power-spectrum test for nonrandomness and applied it to each of the pseudorandom sequences he computed from nonweak keys. Gait observed that each of these sequences was considered random by his test.[14]

Provided a feedback width of 64 bits is used, the cycling study considered by Gait is equivalent to what we call the "orbit test," which can be viewed as a closure test (see Section 4.4.3). If DES were closed, then each of the orbits considered by Gait would have at most $K = 2^{56}$ messages (see Lemma 3.2). Hence, observing an orbit of length greater than 2^{56} would be direct proof that DES is not closed. Although we do not do so in this paper, it is also possible to interpret the orbit test as a statistical closure test. In contrast with Gait's experiments, we followed the orbit of a randomly chosen DES transformation for over 2^{36} steps (see Section 6).

Davies and Parkin [11], [10] and Jueneman [28] studied mathematically the cycle structure of the key stream produced in output-feedback mode. Each of these studies concluded that if DES is used in output-feedback mode with a feedback-width of less than 64 bits, then the resulting key stream will cycle in approximately 2^{32} steps, on the average (the exact expected cycle length depends slightly on the feedback width). If all 64 bits are fed back, then the expected cycle length is approximately 2^{63}. The point is that the state-transition function in output-feedback mode is a permutation if and only if all 64 bits are fed back. Although Davies and Parkin did not report performing any experiments on the full DES algorithm, Davies and Parkin did run a series of experiments on DES substitutes consisting of random permutations on $\{0, 1\}^8$. Their experimental results agreed with their theoretical predictions.

In an attempt to understand better how effectively the Hellman cryptanalytic time–space tradeoff [25] could be applied to DES, Hellman and Reyneri [26] examined the cycle structure of mappings induced by DES on the key space. Specifically, they considered mappings $F_x: \mathcal{K} \to \mathcal{K}$ defined by $F_x(k) = \rho(T_k(x))$,

[13] Since $T_k^2 = I$ for any weak key k, the key stream produced in output-feedback mode with feedback width 64 bits cycles after 128 bits whenever a weak key is used.

[14] See [17] for some recent observations on Gait's work.

where ρ: $\mathcal{M} \to \mathcal{K}$ is a projection[15] and $x \in \mathcal{M}$ is some fixed message. Their studies detected no significant statistical irregularities. Whether or not DES is closed, the expected cycle length of the Hellman–Reyneri experiment is about $\sqrt{K} = 2^{28}$.

Each of these previous cycling projects studied the behavior of the powers of some indexed function (i.e., $T_k^i(x_0)$ or $F_x^i(k_0)$ for $i = 1, 2, \ldots$) where the index of the function was held fixed throughout the experiment: Gait and Davies and Parkin held the key fixed; Hellman and Reyneri held the message fixed. By contrast, our cycling test computes the sequence $x_i = T_{k_i} T_{k_{i-1}} \cdots T_{k_1}(x_0)$ for $i = 1, 2, \ldots$ where at each step i the key k_i is chosen as a pseudorandom function of the previous ciphertext x_{i-1}.

4. Testing Cryptosystems for Algebraic Structure

This section describes and analyzes two statistical tests for determining if a cryptosystem is closed under functional composition. The first test is a meet-in-the-middle algorithm that uses $O(\sqrt{K})$ time and space. The second test is a novel cycling algorithm that uses $O(\sqrt{K})$ time, but only a small constant amount of space. Each test is based heavily on the Birthday Paradox (see Section 3.4). This section also describes several other related tests. Although our primary interest in these tests was to examine DES for closure, purity, and other extreme algebraic weaknesses, the tests are general in nature.

4.1. *Conducting and Interpreting the Algebraic Tests*

This section describes the nature of our tests, concentrating on the framework in which the tests operate and on how to interpret the test results.

4.1.1. *Testing Framework.* Input to each test is a finite, deterministic cryptosystem $\Pi = (\mathcal{K}, \mathcal{M}, \mathcal{C}, T)$, with the encryption transformation T presented as a "black box." Given any key $k \in \mathcal{K}$ and any message $x \in \mathcal{M}$, the box computes $T_k(x)$ and $T_k^{-1}(x)$. No additional information about T is provided. To ensure that messages and keys are easy to detect, generate, and compare, we assume that $\mathcal{M} = \{0, 1\}^u$, $\mathcal{K} = \{0, 1\}^v$, and $\mathcal{C} = \{0, 1\}^w$, for some u, v, w also provided to the test.

We assume that the sets $\mathcal{K}$, $\mathcal{M}$, and $\mathcal{C}$ are so large that they cannot be exhaustively searched; each test must proceed by examining a limited number of messages and keys. We do assume, however, that running times of $O(\sqrt{M})$, $O(\sqrt{K})$, and $O(\sqrt{C})$ are tractable.

4.1.2. *Interpreting the Results.* Each closure test computes a statistic, which can be used to calculate a measure of our relative degree of belief in the following two competing hypotheses:

- H_G = "$\mathcal{T}_\Pi$ is a group."
- H_R = "Each transformation T_k was chosen independently with uniform probability from the symmetric group on $\mathcal{M}$."

[15] Hellman and Reyneri used the projection that removes each of the eight parity bits.

To compute this measure we will apply the *theory of the weight of evidence*, as explained by Good [40], [22].

Let E be experimental evidence produced by one trial of one of the closure tests. From this evidence we can compute the conditional probabilities $P(E|H_G)$ and $P(E|H_R)$, as explained in the next two sections. Note, however, that neither closure test enables us to compute $P(E|\overline{H_G})$ or $P(E|\overline{H_R})$, where $\overline{H_G}$ and $\overline{H_R}$ are the complements of H_G and H_R, respectively. Thus, on the basis of experimental evidence, we would be able to conclude only that Π is *not* closed or that Π has a structure different from that expected from a set of randomly chosen permutations; we would not be able to conclude that Π is closed. In the worst case, Π could be closed, except for some isolated pair of keys a, b such that $T_b T_a$ is not in $\mathscr{T}_\Pi$, even though there exists some key k and some message x_0 such that $T_b T_a(x) = T_k(x)$ for all messages $x \in \mathscr{M}$, $x \neq x_0$.

Initially, each person may have some (subjective) degrees of belief $P(H_G)$ and $P(H_R)$ in hypotheses H_G and H_R, respectively. From these initial degrees of belief, each person can compute $O(H_G/H_R) = P(H_G)/P(H_R)$ as his or her initial *odds in favor of* H_G *over* H_R. After seeing any experimental evidence E, however, each rational person should update his or her own odds in favor of H_G over H_R.

Given any evidence E, a Bayesian would update his or her odds in favor of H_G over H_R as follows:

$$O(H_G/H_R|E) \leftarrow \frac{P(E|H_G)}{P(E|H_R)} O(H_G/H_R), \tag{4}$$

where $O(H_G/H_R|E)$ is the *odds in favor of* H_G *as opposed to* H_R *given* E.

We encourage the reader to update his or her own odds in favor of H_G over H_R in light of the evidence presented in Section 6.

4.2. *Meet-in-the-Middle Closure Test*

The meet-in-the-middle closure test (MCT) works as follows: given any endomorphic cryptosystem $\Pi = (\mathscr{K}, \mathscr{M}, \mathscr{M}, T)$, pick any key $k \in \mathscr{K}$ and search for keys $a, b \in \mathscr{K}$ such that $T_k = T_b T_a$. If Π is closed, then such a pair of keys a, b can be efficiently found, with high probability. If $\mathscr{T}_\Pi$ is selected at random, then it is unlikely to find any such pair.

To search for a pair of keys $a, b \in \mathscr{K}$ such that $T_k = T_b T_a$, we use a standard "meet-in-the-middle" algorithm similar to that described in [37], for example. Specifically, choose $2r$ keys $a_1, a_2, \ldots, a_r$ and $b_1, b_2, \ldots, b_r$ at random and look for a pair of keys a_i, b_j for some $1 \leq i, j \leq r$ such that $T_k = T_{b_j} T_{a_i}$. To find such a match, represent the cryptographic transformations by their images or preimages of some particular message. Specifically, pick any message $p \in \mathscr{M}$, calculate $c = T_k(p)$, and compute $x_i = T_{a_i}(p)$ and $y_i = T_{b_i}^{-1}(c)$ for $1 \leq i \leq r$. Then look for matches $x_i = y_j$ by sorting the triples $(x_i, a_i, \text{"}A\text{"})$ and $(y_j, b_j, \text{"}B\text{"})$ for $1 \leq i, j \leq r$ on their first components. Screen out false matches by testing if $T_k(p_h) = T_{b_j} T_{a_i}(p_h)$, for all $1 \leq h \leq l$, for a small number of additional messages $p_1, p_2, \ldots, p_l \in \mathscr{M}$. (A *false match* is a pair of keys $a', b' \in \mathscr{K}$ such that $T_k(p) = T_{b'} T_{a'}(p)$ even though $T_k \neq T_{b'} T_{a'}$.) Figure 2 summarizes this process.

input: An endomorphic cryptosystem $\Pi = (\mathcal{K}, \mathcal{M}, \mathcal{M}, T)$ and integer control parameters r, l.

begin

1. Pick $k \in \mathcal{K}$ and $p_1, \ldots, p_l \in \mathcal{M}$ at random. For $i = 1$ to l, compute $c_i = T_k(p_i)$. Let $p = p_1$ and $c = c_1$.
2. For $i = 1$ to r, select $a_i, b_i \in \mathcal{K}$ at random and compute $x_i = T_{a_i}(p)$ and $y_i = T_{b_i}^{-1}(c)$.
3. Sort the triples $(x_i, a_i, \text{"}A\text{"})$ and $(y_i, b_i, \text{"}B\text{"})$ for $1 \le i \le r$ on their first components.
4. For each "match" $x_i = y_j$ with $1 \le i, j \le r$, if $T_k = T_{b_j} T_{a_i}$, then **return**("Match found"). To test if $T_k = T_{b_j} T_{a_i}$, statistically verify that $c_h = T_{b_j} T_{a_i}(p_h)$ for all $1 \le h \le l$.
5. **return**("No match found")

end

Fig. 2. Meet-in-the-middle closure test (MCT).

Proposition 4.1 summarizes the main properties of MCT. Informally, Proposition 4.1 says that MCT is likely to find a match if Π is closed, but MCT is unlikely to find a match if Π is chosen at random. These facts follow from Lemma 3.1 and the Birthday Paradox.

Proposition 4.1. *If Π is closed, then MCT finds a match with probability at least $1 - e^{-3r^2/K}$. If $\mathcal{T}_\Pi$ is chosen at random, then we expect MCT to find a match with probability at most $K^2/M!$.*

Proof. If Π is closed, then for each $1 \le j \le r$, $T_{b_j}^{-1} T_k \in \mathcal{T}_\Pi$. In this case the situation is a variation of the Birthday paradox in which we are drawing two samples X and Y, each of size r, from an urn containing m elements, where m is the order of Π. The first sample consists of the transformations $T_{a_1}, \ldots, T_{a_r}$; the second consists of the transformations $T_{b_1}^{-1} T_k, \ldots, T_{b_r}^{-1} T_k$. If Π is faithful, each element is drawn with probability exactly $1/K$; otherwise, each element is drawn with probability at least $1/K$. Thus, the worst case is when Π is faithful. We are interested in the probability that the samples overlap.

If $\mathcal{T}_\Pi$ is chosen at random, then by Lemma 3.1, for any $T_k \in \mathcal{T}_\Pi$, we expect $\mathcal{T}_\Pi$ to contain a pair T_a, $T_b \in \mathcal{T}_\Pi$ such that $T_k = T_b T_a$ with probability at most $K^2/M!$. □

Thus, by choosing $r = c\sqrt{m}$ with c sufficiently large, we can make the probability $q_r \approx 1 - e^{-3c^2}$ of finding a match if Π is closed as large as desired.

The analysis in Proposition 4.1 assumes that each sequence of keys $a_1, \ldots, a_r$ and $b_1, \ldots, b_r$ was drawn without replacement. If these sequences are drawn with replacement, then the expected number of samples required to obtain r distinct keys is $K \log((K + 0.5)/(K - r + 0.5))$. This situation is a variation of the "collector's problem" [18].

To carry out MCT efficiently, it is important that the expected number of false matches be small. As shown by Lemma 3.4, if Π is closed, then at most $(K - 1)/|B_p|$ keys other than k map p to c, where B_p is the $\mathcal{T}_\Pi$-orbit of p. If $\mathcal{T}_\Pi$ is chosen at random, then we expect at most $(m - 1)/M$ keys other than k to map p to c. Thus, provided K is not too much larger than M, the expected number of false matches is small.

MCT requires $O(r)$ steps and $O(r)$ words of memory. The two most time-consuming operations are generating and sorting the lists $x_1, x_2, \ldots, x_r$ and $y_1, y_2, \ldots, y_r$. The required number of encryptions is $2r$ plus the number of additional evaluations used to screen out false matches. If sorting is performed in main memory using radix sort, then sorting will take $O(r)$ machine operations; otherwise, $O(r \log r)$ external memory operations would be needed. The main difficulty with carrying out this test on DES is the high space requirement.

Given the high space requirement of MCT, in practice it may be helpful to use variations of this test that involve time–space tradeoffs. For example, the test could be repeated several times with small values of r. Alternately, the test could build a small hash table for the x_i's and then lookup each y_i in the table without saving the y_i's. If encryption is relatively fast in comparison to the other required operations, then it might be advantageous to save only those x_i's that fall into some subset of the message space. Parallel variations of MCT are also possible.

4.3. *Cycling Closure Test*

Given any endomorphic cryptosystem $\Pi = (\mathcal{K}, \mathcal{M}, \mathcal{M}, T)$, the cycling closure test (CCT) takes a pseudorandom walk in $\mathcal{M}^l$ for some small l. The walk continues for a specified number of steps or until a cycle is encountered. Long walks are strong evidence that Π is not closed; short walks are strong evidence that Π has a structure different from that expected from a set of randomly chosen permutations.

Specifically, CCT picks an initial vector of messages $\hat{x}_0 \in \mathcal{M}^l$ at random and computes the leader length and cycle length of a sequence $\hat{x}_0, \hat{x}_1, \ldots$. For each $i \geq 0$, the next element in this sequence is computed by

$$\hat{x}_{i+1} = f_\rho(\hat{x}_i), \tag{5}$$

where the function $f_\rho: \mathcal{M}^l \to \mathcal{M}^l$ is defined by

$$f_\rho(\hat{x}) = T_{\rho(\hat{x})}(\hat{x}) \tag{6}$$

for all $\hat{x} \in \mathcal{M}^l$. The sequence is guided by a deterministic pseudorandom function $\rho: \mathcal{M}^l \to \mathcal{K}$ which maps message vectors to keys. Figure 3 summarizes this process. (In equation (6) and throughout this section we use the convenient notation $T_k(\hat{x})$

input: An endomorphic cryptosystem $\Pi = (\mathcal{K}, \mathcal{M}, \mathcal{M}, T)$,
integer control parameters $r_{\max}$, l, and
a deterministic pseudorandom function $\rho: \mathcal{M}^l \to \mathcal{K}$.

begin

1. Pick $\hat{x}_0 \in \mathcal{M}^l$ at random.
2. Compute the leader length and cycle length of the sequence $\hat{x}_0, \hat{x}_1, \ldots$ defined by $\hat{x}_{i+1} = f_\rho(\hat{x}_i) = T_{\rho(\hat{x}_i)}(\hat{x}_i)$ for all $i > 0$. If no cycle is detected after $r_{\max}$ steps, then **return**("No cycle detected").
3. **return**(ω), where $\omega = \lambda + \mu$; λ is the leader length; and μ is the cycle length computed in step 2.

end

Fig. 3. Cycling closure test (CCT).

to denote $(T_k(x_1), \ldots, T_k(x_l))$, for any key $k \in \mathcal{K}$ and any message vector $\hat{x} = (x_1, \ldots, x_l) \in \mathcal{M}^l$.)

To detect cycles and to compute leader lengths and cycle lengths, use the efficient algorithms described by Sedgewick and Szymanski [49] that generalize the well-known "two-finger" algorithm due to Floyd [32].

The entire cycling closure test requires $O(\omega)$ time and a constant amount of space, where $\omega = \lambda + \mu$ and λ and μ are, respectively, the leader length and cycle length computed by the test.

The cycling closure test is similar in spirit to Pollard's ρ-factoring method [41], [4]. It is also similar to the algorithm described by Sattler and Schnorr for determining the order of any element in any finite group, but Sattler and Schnorr's algorithm requires the group to have an efficient multiplication procedure [47]. The cycling test differs from the cycling experiments performed by Gait [20] and Hellman and Reyneri [26], who held either the key or message fixed (see Section 3.5).

Proposition 4.2 states the main properties of CCT, which follow from the Birthday Paradox. In summary, CCT takes a pseudorandom walk in $S_{\hat{x}_0}$ where $S_{\hat{x}_0} = G_\Pi$-orbit$(\hat{x}_0)$. This walk is defined by the f_ρ-closure of $\hat{x}_0$. Provided ρ is "random," f_ρ acts like a random function on $S_{\hat{x}_0}$. If Π is closed, then $|S_{\hat{x}_0}| \leq K$. If Π is chosen at random, then we expect $|S_{\hat{x}_0}| = M^l$. Hence, if Π is closed we expect the pseudorandom walk to cycle in approximately $\sqrt{K}$ steps, but if Π is chosen at random we expect the walk to cycle in approximately $M^{l/2}$ steps. The constant l should be selected so that $M^{l/2}$ is sufficiently larger than $\sqrt{K}$ (to apply the test to DES, $l = 1$ suffices).

Proposition 4.2. *Let ω be the statistic computed by CCT and let r be any positive integer. If Π is closed, then $P(\omega > r) \leq e^{-r^2/(2K)}$. If Π is chosen at random, then $P(\omega > r) \approx e^{-r^2/(2M^l)}$.*

Proof. CCT computes a sequence of message vectors $\hat{x}_0, \hat{x}_1, \ldots$ in $\mathcal{M}^l$. More specifically, for each $i \geq 0$, it is true that $\hat{x}_i \in B$, where B is the set $B = \{\hat{y}: \hat{y} = g(\hat{x}_0)$, for some $g \in G_\Pi\}$ and G_Π is the group generated by $\mathcal{T}_\Pi$.

If Π is closed, then $|B| \leq K$. In this case we can model the pseudorandom walk $\hat{x}_0, \hat{x}_1, \ldots$ as a discrete finite Markov Process with a $|B| \times |B|$ transition matrix A. For each $1 \leq i, j \leq |B|$, the (i, j)th entry a_{ij} of A denotes the probability of selecting $\hat{x}_i$ next, given that $\hat{x}_j$ was the last selected message vector. Each pseudorandom selection depends only on the immediately preceding message vector. Moreover, for any message vectors $\hat{x}, \hat{y} \in B$, there is some key k such that $T_k(\hat{x}) = \hat{y}$. Therefore, each entry of A is at least $1/K$. Thus, for any positive integer r, the probability of a pseudorandom walk not cycling within r steps is at most $(K)_r/K^r$.

If $\mathcal{T}_\Pi$ is chosen at random, then the walk in G_Π induces a pseudorandom walk in $\mathcal{M}^l$. □

We now explain how to interpret CCT experiments in light of the two hypotheses H_G and H_R. The evidence obtained from CCT is the value of the statistic ω. From this evidence, a Bayesian would update his or her odds in favor of H_G over H_R by

a factor of p_G/p_R, where $p_G = P(\omega = r|H_G)$ and $p_R = P(\omega = r|H_R)$ and r is the observed value of ω.

We now estimate the density functions p_G and p_R. By Proposition 4.2 we know that, for any r,

$$P(\omega > r|H_G) \leq e^{-c_1^2/2} \tag{7}$$

and

$$P(\omega > r|H_R) \approx e^{-c_2^2/2}, \tag{8}$$

where $c_1 > 0$ and $c_2 > 0$ are defined by $r = c_1\sqrt{K} = c_2 M^{l/2}$. Hence, $P(\omega \leq r|H_G) \leq 1 - e^{-c_1^2/2}$ and $P(\omega \leq r|H_R) \approx 1 - e^{-c_2^2/2}$.

The density function p_R can be obtained by differentiating the distribution function obtained from equation (8) with respect to r. Thus,

$$p_R \approx \frac{d(1 - e^{-c_2^2/2})}{dr} = c_2 \cdot e^{-c_2^2/2} \cdot \frac{dc}{dr} = \frac{c_2}{M^{l/2}} \cdot e^{-c_2^2/2} = \frac{r}{M^l} \cdot e^{-c_2^2/2}. \tag{9}$$

Since equation (7) gives only an upper bound (and not an approximation) of $P(\omega > r|H_G)$, computing p_G is more involved than computing p_R. For simplicity, we use the coarse bound

$$p_G \leq P(\omega \geq r|H_G) \approx P(\omega > r|H_G) \leq e^{-c_1^2/2}, \tag{10}$$

which is sufficiently powerful to interpret results of CCT. We leave as an open problem how to compute p_G more exactly.

Thus, the ratio p_G/p_R is bounded by

$$\frac{p_G}{p_R} \leq \frac{e^{-c_1^2/2}}{e^{-c_2^2/2}} \cdot \frac{M^l}{r}. \tag{11}$$

The validity of the cycling test depends in part on the extent to which the pseudorandom walk behaves like a truly random walk. To increase our confidence that the pseudorandom function does not interact with the cryptosystem in a way that would invalidate the statistical analysis, we recommend that each trial of the experiment be repeated with different types of pseudorandom functions.[16] (See Appendix A for a description of the particular pseudorandom functions used in our experiments.)

It is possible for the random walk to cycle prematurely if certain special sequences of transformations are chosen during the walk. For example, the cycle will close if the identity transformation is selected, or if a transformation and its inverse are selected one after the other. In particular, the latter condition happens if a pair of weak DES keys is selected one after the other. In general, every closing of the cycle reveals an algebraic identity of the cryptosystem. Any short cycle is evidence that $\mathcal{T}_\Pi$ has a structure different from that expected from a set of randomly chosen permutations.

[16] For example, the pseudorandom function might be table look-up into a table of randomly generated values, modification of table look-up in which each input into the table is first XOR'd with the previous output from the table, or DES under a randomly chosen fixed key.

4.4. *Additional Algebraic Tests*

Although the emphasis of our experiments centered around the cycling closure test, we also carried out five additional cycling tests on DES. This section briefly describes these additional tests, which we call the *purity test*, *orbit test*, *small subgroup test*, *extended message space closure test*, and *reduced message space test*. We carried out each of these additional tests using the same special-purpose hardware.

4.4.1. *Overview and Motivation.* It is important to know if DES is pure for essentially the same reasons that it is important to know if DES is closed. If DES were pure, then Tuchman's multiple encryption scheme would be equivalent to single encryption, and DES would be vulnerable to a known-plaintext attack that runs in 2^{28} steps on the average. It is possible that DES is pure, but not closed. Although there is no particular reason to suspect that DES is pure, it is unknown in the open literature if DES has this weakness.

Since any set of DES transformations that generates a small group would suffer the weaknesses of closed ciphers, is natural to ask: "What is the order of the group generated by n given DES transformations?" We call this question the *small subgroup question*. Any set of transformations that generates a small group would be vulnerable to our known-plaintext attacks against closed ciphers. In addition, multiple encryption (using either sequential multiple encryption or Tuchman's scheme) involving only transformations from such a set would be equivalent to single encryption from the set. Finally, when used in output-feedback mode with feedback width 64 [62], any transformation from such a set would be at greater risk to produce a key stream with short period. Two of our tests address the small subgroup question for $n = 1, 2$.

To test DES for purity and other algebraic weaknesses, we examined the orbits of subsets of DES transformations on particular messages. Our method was to compute the orbits of single DES transformations and to apply the cycling closure test to subsets of two or more DES transformations. We applied the tests both to randomly chosen transformations and to transformations with special properties (e.g., transformations represented by weak keys). The dominant theme of our tests was to determine if DES has algebraic properties different from those expected from a set of randomly selected permutations.

When applied to any subset $S \subseteq \mathscr{T}_\Pi$ of two or more transformations, the cycling closure test computes the f_ρ-closure of some message x_0, where $\rho: \mathscr{M} \to H$ and $H \subseteq \mathscr{K}$ is a set of keys that represents S and f_ρ is the function defined by equation (6).

Since there is an overwhelming chance that even two randomly selected permutations will generate either the alternating group or the symmetric group [3], [16], we did not expect to detect any pairs of DES transformations that generate small groups.

4.4.2. *Purity Test.* Pick any transformation $T_0 \in \mathscr{T}_\Pi$ and apply the cycling closure test to the set $T_0^{-1}\mathscr{T}_\Pi$. As explained in Section 3.1, $\mathscr{T}_\Pi$ is pure if and only if $T_0^{-1}\mathscr{T}_\Pi$ is closed.

4.4.3. *Orbit Test.* Given any key k and any message x_0, compute $x_i = T_k^i(x_0)$, $i = 1, 2, \ldots$, for a specified number of steps or until a cycle is detected.

The period of this sequence is the length of T_k-orbit(x_0). In other words, if we consider the permutation T_k as a product of disjoint cycles, then the period of the sequence is simply the length of the cycle that contains x_0. If this test is run for r steps without detecting a cycle, then r is a lower bound on order(T_k) and hence on order($\langle \mathscr{T}_\Pi \rangle$). Unlike the function f_ρ from the cycling closure test, T_k is a permutation.

For a randomly chosen permutation on $\mathscr{M}$, for each $1 \leq l \leq M$, the probability that x_0 lies in a cycle of length exactly l is $1/M$, independent of l [23], [44], [32, exercise 3.1.12]. Hence, the expected cycle length of the longest cycle of a randomly chosen permutation on n letters is about $0.624n$ [50] (for DES, this is about 2^{63}). For a randomly chosen permutation on $\mathscr{M}$, the chance that we fall into a cycle of length 2^{36} or less is about $2^{-(63-36)} = 2^{-27}$.

It is possible to interpret results of the orbit test to obtain statistical lower bounds on the order of the group generated by DES. Such analysis depends on the structure of the group. For example, the orbit test behaves differently on cyclic groups than on symmetric groups.[17] Consequently, it is useful to combine the orbit test with other algebraic tests, including tests for faithfulness, commutativity, solvability at various levels, and nilpotence at various classes. We leave such analysis as future research.

4.4.4. *Small Subgroup Test.* Given two distinct keys $i, j \in \mathscr{K}$ and any message x_0, apply the cycling closure test to the set $\{T_i, T_j\}$ to obtain a statistical lower bound on the length of the $\langle T_i, T_j \rangle$-orbit of x_0.

To increase our likelihood of finding algebraic structure, we carried out the small subgroup test on a highly structured pair of DES transformations—the pair of weak keys consisting of all zeros and all ones. Since each of the weak keys is self-inverse, we implemented this experiment as an orbit test of the composition of these two transformations. Our experiment unexpectedly encountered a short orbit (see Section 6).

4.4.5. *Extended Message Space Closure Tests.* For any experiment that uses the cycling closure test, perform the cycling closure test with an extended message space $\mathscr{M}^l$ with $l > 1$.

For $l = 1$, the closure test works by computing a statistical lower bound on the length of $\langle \mathscr{T}_\Pi \rangle$-orbit($x_0$), which, in turn, yields a lower bound on the order of $\langle \mathscr{T}_\Pi \rangle$. Limits on the lower bounds achievable by this test are imposed both by the number of steps the test is carried out for and by the relative sizes of the message space and key space. For all $1 \leq r \leq \sqrt{M}$, if the test is run for r steps without detecting a cycle, then with high probability order($\langle \mathscr{T}_\Pi \rangle$) $\geq r^2$. To use the cycling closure test to obtain statistical lower bounds on order($\langle \mathscr{T}_{\text{DES}} \rangle$) greater than 2^{64}, it is necessary to perform an extended message test with $l > 1$.

4.4.6. *Reduced Message Space Tests.* Perform each of the above tests on a modified version of DES in which the message space is reduced in size. Specifically, consider DES-derived functions $\varphi_k: \mathscr{M}_r \to \mathscr{M}_r$ on the reduced message space $\mathscr{M}_r = \{0, 1\}^r$, where r is some small integer (say, $r = 8$) and φ_k is defined as follows. For each

[17] See [2] for an approximate probability distribution of the order of elements in $\mathscr{S}_\mathscr{M}$.

key $k \in \mathcal{K}$, define φ_k by $\varphi_k = \pi_2 T_k \pi_1$, where $\pi_1 : \mathcal{M}_r \to \mathcal{M}$ is an injection and $\pi_2 : \mathcal{M} \to \mathcal{M}_r$ is a projection. (For example, π_1 might fix the first 56 DES input bits to 0, and π_2 might take only the last 8 DES output bits.)

Studying reduced message space versions of DES is useful for two reasons. First, it is one way to look for structures that may be present on subsets of the message space. Second, by sufficiently restricting the message space, it is possible to write down a complete description of the action of particular transformations on the reduced message space.

5. Attacks Against Group Ciphers

Each of the closure tests can be used with only slight modifications as a known-plaintext attack against any closed cipher. The input to each attack is a short sequence $(p_1, c_1), (p_2, c_2), \ldots, (p_l, c_l)$ of matched plaintext/ciphertext pairs derived from the same secret key k. With high probability each attack finds a representation of T_k as a product of two or more transformations. The cryptanalyst can use this representation of T_k to decrypt additional ciphertexts also encrypted under k. The meet-in-the-middle attack requires more space than the cycling attack, but finds a shorter representation of T_k. Neither attack finds k.

5.1. *Meet-in-the-Middle Known-Plaintext Attack*

The meet-in-the-middle test first picks any message p and any key k at random and then computes the ciphertext $c = T_k(p)$. Next, the test searches for a pair of keys a, b such that $T_k = T_b T_a$. Alternately, a cryptanalyst could begin with any matched plaintext/ciphertext pair (p, c) that was encrypted using some unknown key k, and then search for a representation of the secret transformation T_k as a product $T_b T_a$. This attack requires $O(\sqrt{K})$ time and space on the average. See Fig. 4.

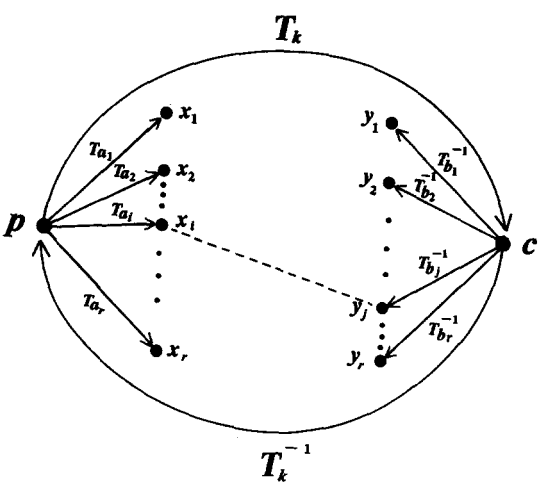

Fig. 4. Meet-in-the-middle attack against group ciphers.

5.2. *Cycling Known-Plaintext Attack*

The cycling closure test also yields a known-plaintext attack. Given a matched plain-text/ciphertext pair (p, c) that was encrypted under some secret key k, the cryptanalyst computes two pseudorandom walks of the type used in the cycling test, starting from messages p and c. The same pseudorandom function is used for each of the walks. If the attacked cryptosystem is closed, then, since p and c lie in the same orbit, with very high probability the two pseudorandom walks will intersect within about $\sqrt{K}$ steps. Since the same deterministic pseudorandom function is used for each of the walks, once the two walks intersect, they will forever follow exactly the same path and will therefore drain into the same cycle.

By running the Sedgewick–Szymanski [49] cycle-detection algorithm for each of the pseudorandom walks, and by sharing the same memory for both algorithms, a specific point at which the walks intersect can be found, provided the walks intersect. The two walks can be computed sequentially or simultaneously.

Thus, the cycling test gives a way of generating two sequences of keys $a_1, a_2, \ldots, a_i$ and $b_1, b_2, \ldots, b_j$ such that $g(p) = h(c) = hT_k(p)$, where $g = T_{a_i} T_{a_{i-1}} \cdots T_{a_1}$ and $h = T_{b_j} T_{b_{j-1}} \cdots T_{b_1}$. With high probability, $T_k = h^{-1}g$, which can be statistically verified by applying $h^{-1}g$ to additional matched plaintext/ciphertext pairs. If $T_k \neq h^{-1}g$, then the entire procedure can be repeated on the next plaintext/ciphertext pair. See Fig. 5.

To decrypt each additional ciphertext y, the cryptanalyst computes $T_k^{-1}(y) = g^{-1}h(y)$. To compute h in constant space is easy—simply generate the sequence of keys $b_1, b_2, \ldots, b_j$ by retracing the pseudorandom walk starting from c. The difficulty is to compute g^{-1} in a time- and space-efficient manner. The problem is that each

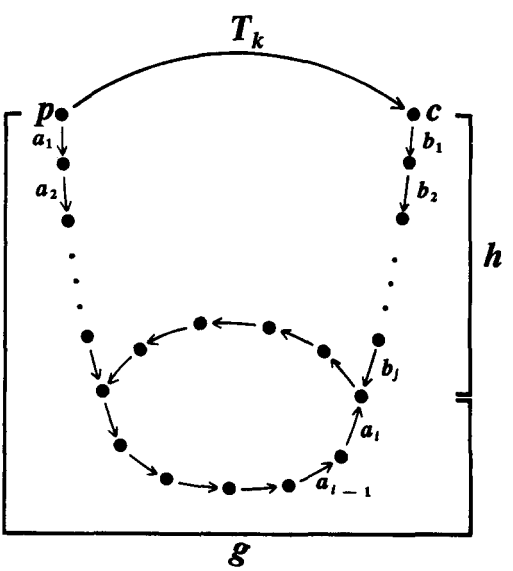

Fig. 5. Cycling known-plaintext attack.

pseudorandom walk is a "one-way walk" in the sense that reversing any step of the walk requires inverting the encryption function.

One could save each of the keys $a_1, a_2, \ldots, a_i$, but that would require $O(i)$ space, where i is the length of the walk starting at p. If the attacked cryptosystem is closed, then i will be about $\sqrt{K}$, on the average. On the other hand, one could reverse any step of the walk in constant space by retracing the walk from the beginning, but this procedure would yield an $O(i^2)$-time algorithm for computing g^{-1}. Chandra shows that a range of time–space tradeoffs can be used to solve this type of problem. In particular, for any $\varepsilon > 0$, it is possible to compute g^{-1} in $O(1/\varepsilon)$ space and time $O(i^{1+\varepsilon})$ [6]. Therefore, if the attacked cryptosystem is closed, then, for any $0 < \varepsilon < 1$, the cycling known-plaintext attack can be carried out in space $O(1/\varepsilon)$ and time $O(K^{(1+\varepsilon)/2})$, on the average.

5.3. *Application of Attacks*

Since these attacks can be launched against the group generated by DES, these attacks do not require that DES be closed; they require only that DES generate a small group.

Since DES's relatively small key space of 2^{56} keys allows no margin of safety even for 1977 technology [14], these attacks would be a devastating weakness for DES, if DES were a group. In particular, if DES were closed, our IBM personal computer equipped with special-purpose hardware could decrypt DES ciphertexts under the cycling known-plaintext attack in less than 2 hours, on the average (see Section 6).

6. Cycling Experiments on DES

Using a combination of software and special-purpose hardware, we applied the cycling closure test and other algebraic tests to DES. This section describes our experimental work. Section 6.1 summarizes our results. Section 6.2 explains two structural findings. Section 6.3 describes our special-purpose hardware. For more detailed descriptions of our results, see Appendix A.

6.1. *Summary of Experimental Results*

On April 4, 1985, we completed our first trial of the cycling closure test [30]. This single experiment gives strong evidence that DES is not closed. During May–August, 1985, we performed additional experiments, including two more closure tests, one extended message space closure test, two purity tests, and two orbit tests [31]. Results from seven of these experiments were consistent with the hypothesis that DES acts like a set of randomly chosen permutations. In particular, these experiments gathered overwhelming evidence that DES is neither closed nor pure. But one orbit experiment involving the composition of two weak keys unexpectedly encountered a small cycle, which was the result of hitting fixed points for each of the weak keys.

Table 1 summarizes our experimental findings. For each experiment, the table lists the approximate leader length and cycle length encountered. The sums of these

Table 1. Summary of DES experiments, May–August, 1985.*

No.	Experiment	|Leader|	|Cycle|	p_G	p_R	P_G	P_R
1	Closure	$\approx 2^{25}$	$\approx 2^{33}$	$\leq 10^{-193}$	$\approx 10^{-10}$	$\leq 10^{-193}$	≈ 0.17
2	Closure	$\approx 2^{30}$	$\approx 2^{33}$	$\leq 10^{-264}$	$\approx 10^{-10}$	$\leq 10^{-264}$	≈ 0.09
3	Closure	$\approx 2^{31}$	$\approx 2^{30.5}$	$\leq 10^{-41}$	$\approx 10^{-10}$	$\leq 10^{-41}$	≈ 0.69
4	Extended closure		(no cycle in 2^{34} steps)			$\leq 10^{-889}$	$\approx 1 - 10^{-18}$
5	Purity	$\approx 2^{31.5}$	$\approx 2^{30}$	$\leq 10^{-61}$	$\approx 10^{-10}$	$\leq 10^{-61}$	≈ 0.57
6	Purity	$\approx 2^{30}$	$\approx 2^{32}$	$\leq 10^{-94}$	$\approx 10^{-10}$	$\leq 10^{-94}$	≈ 0.43
7	Weak key orbit	0	$\approx 2^{33}$	†	$\approx 10^{-19}$	†	$\approx 10^{-9}$
8	Orbit		(no cycle in 2^{36} steps)			†	$\approx 1 - 10^{-8}$

* The numbers p_G, p_R, P_G, and P_R are the conditional probabilities of the experimental evidence under the hypotheses "DES is closed (pure)" and "each DES transformation was drawn at random from the symmetric group on $\mathcal{M}$," respectively. The numbers p_G and p_R indicate the chance of encountering a cycle after exactly r steps, where r is the sum of the observed leader and cycle lengths. The numbers P_G and P_R indicate the chance of not encountering a cycle within r steps.

† Depends on hypothesized group structure.

lengths form the values of the statistic ω computed by the tests. The table also lists the conditional probabilities p_G, p_R, P_G, and P_R of the experimental evidence under the hypotheses H_G and H_R, respectively. The numbers p_G and p_R are based on probability *density* calculations and indicate the chance of encountering a cycle after exactly r steps, where r is the observed value of ω. The numbers P_G and P_R are based on probability *distribution* calculations and indicate the chance of not encountering a cycle within r steps.

For experiments 1, 2, 3, 5, and 6, p_G and p_R were computed from equations (10) and (9), respectively. For these same experiments, as well as for experiment 4, P_G and P_R were computed from equations (7) and (8), respectively. For experiments 7 and 8, the values of p_R and P_R were computed as explained in Section 4.4.3. As explained in Section 4.3, for simplicity, we coarsely bound p_G from above by P_G.

In the first cycling closure experiment—which ran for about 2 days—we found a cycle of length exactly $\mu = 7{,}985{,}051{,}916$ with a leader of length $\lambda = 34{,}293{,}589$. Let E denote the evidence from our experiment. Since $\mu + \lambda \approx 2^{33} = 2\sqrt{M} = 32\sqrt{K}$, it follows from equation (11) that $P(E|H_G)/P(E|H_R) \leq (e^{-32^2/2}/e^{-2^2/2}) \cdot (2^{64}/2^{33}) \leq e^{-510+22} = e^{-488}$. On the basis of this experiment alone, each reader should decrease his or her odds in favor of H_G over H_R by a factor of about e^{-488}. Results of the other closure and purity experiments can be interpreted in a similar fashion.

The second closure experiment produced even stronger evidence that DES is not closed. Moreover, the pseudorandom walks from the first two experiments drained into the same cycles (see Section 6.2.1).

Using 128-bit messages, the extended closure test did not cycle after 2^{34} steps, showing that the group generated by DES probably has at least 2^{68} elements.

In experiment 7 we computed the orbit of the composition of the two weak keys that consist, respectively, of all zeros and all ones. This experiment produced a short cycle of approximately 2^{33} steps, which would be unusual (the probability of encountering a cycle of length at most 2^{33} is less than 10^{-9}) if the tested permutation were chosen at random from $\mathcal{S}_{\mathcal{M}}$ (see Section 6.2.2).

In experiment 8 we computed the orbit of a randomly chosen transformation for 2 weeks. No cycle was detected after 2^{36} steps. This experiment provided no evidence of any algebraic weakness.

In addition, we ran one reduced message space test for which we observed no algebraic weaknesses.

As one test of correctness, we ran a software implementation of the cycling closure test for 30,000 steps. The software and hardware implementations agreed on all values. As a second test of correctness, we repeated each experiment and obtained identical results. We invite the interested reader to verify our results using the detailed experimental data found in Appendix A. Additional analysis by Kaliski further strengthens the reliability of our experimental results [29].

6.2. *Two Structural Findings*

Although most of our experimental results are consistent with the hypothesis that DES acts like a set of randomly chosen permutations, three experiments did yield interesting regularities. One regularity is a result of the well-known complementation property; the other involves a newly discovered property of the weak keys. We will now explain these structural findings.

6.2.1. *Complementation and Drainage Properties.* In the first two experiments, we performed two independent trials of the cycling closure test. Each of these experiments used the "identity" next-key function—the function ρ: $\mathcal{M} \to \mathcal{K}$ that removes each of the eight parity bits. These two experiments produced two interesting findings. First, each of the pseudorandom walks drained into the same cycle. Second, each point on the cycle was the bitwise complement of the corresponding point exactly halfway around the cycle. Figure 6 illustrates these findings.

The first finding is explained by the fact that, for the graph of a randomly chosen function, most points on the graph will probably drain into the same cycle. (See [26] for one analysis of this phenomenon.)

The second finding is a consequence of DES's complementation property and the fact that the identity next key function also has a complementation property: for all messages x, $\rho(\bar{x}) = \overline{\rho(x)}$. The cycling closure test computes a pseudorandom walk $x_0, x_1, \ldots$, where $x_{i+1} = T_{\rho(x_i)}(x_i)$ for $i \geq 1$. If $x_i = \bar{x}_j$ for any $i > j$, then it would

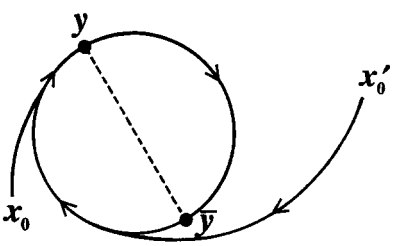

Fig. 6. Results of experiments 1 and 2. Starting at different initial messages, both pseudorandom walks entered the same cycle. Every message on the cycle is the bitwise complement of the corresponding message halfway around the cycle.

follow that

$$x_{i+1} = T_{\rho(x_i)}(x_i) = T_{\rho(\overline{x_j})}(\overline{x_j}) = T_{\overline{\rho(x_j)}}(\overline{x_j}) = \overline{T_{\rho(x_j)}(x_j)} = \overline{x_{j+1}}. \tag{12}$$

Therefore, by induction, $x_{i+h} = \overline{x_{j+h}}$ for all $h \geq 0$. This situation arises whenever some $x_i = \overline{x_j}$ before any $x_i = x_j$ with $i > j$, which will happen for about half of all initial messages.

6.2.2. *Fixed Points of the Weak Keys.* In experiment 7 we computed the orbit of a message under the composition of the two weak keys that consist, respectively, of all zeros and all ones. Let these two keys be denoted by w_0 and w_1, respectively. Although each weak key transformation is self-inverse, we did not expect the composition $T_{w_1} T_{w_0}$ to produce short orbits. Much to our surprise we detected a cycle of length less than 2^{33}. We presented this finding at the Crypto 85 conference and sought a simple explanation.

After some thought, Don Coppersmith suggested that we had encountered fixed points of the weak keys, i.e., messages x, y for which $T_{w_0}(x) = x$ and $T_{w_1}(y) = y$. Figure 7 illustrates the effect of the fixed points on experiment 7 and explains why a cycle resulted. Experiment 7 computes the ψ-orbit of an initial message x_0, where $\psi = T_{w_1} T_{w_0}$. Let $x_i = \psi^i(x_0)$ for all $i \geq 1$. In Fig. 7 solid circles denote the messages $x_0, x_1, \ldots$ in the ψ-orbit of x_0. Open circles denote intermediate messages $y_i = T_{w_0}(x_i)$ for all $i \geq 0$. After encountering a fixed point for T_{w_0} on the jth step ($j \approx 2^{32}$), the walk began to retrace its steps "out of phase" in the sense that $x_{j+i} = y_{j-i}$ for all $i \geq 0$. Continuing in this fashion, the walk passed over the initial message x_0 in a "hidden crossing" $y_{2j} = x_0$, unnoticed during the experiment since the intermediate values y_i were not examined. After approximately 2^{32} steps past the hidden crossing, the walk encountered a fixed point for T_{w_1}. Again, the walk retraced its steps, but this time "in phase," finally returning to the initial message x_0.

As we will show, for each weak key, a fixed point results whenever the L and R registers of DES agree after eight rounds.[18] Assuming that the distribution of values taken on by the 32-bit L and R registers is random after eight rounds, the L

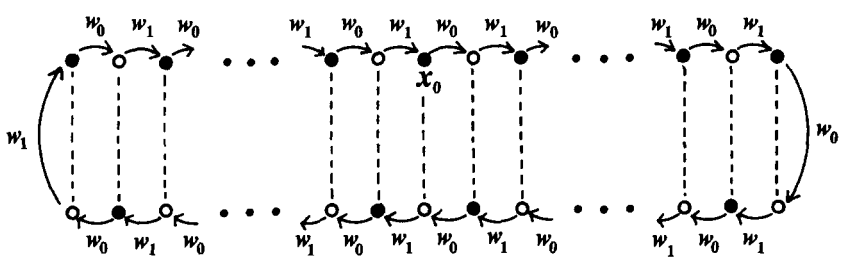

Fig. 7. Experiment 7 discovered fixed points of the weak keys. Let w_1 and w_0 denote, respectively, the weak keys that consist of all ones and all zeros. Solid circles denote the messages x_i on the $T_{w_1} T_{w_0}$-orbit of an initial message x_0. Open circles denote intermediate values $T_{w_0}(x_i)$. Dashed lines link identical messages.

[18] The L and R registers are commonly used in the definition of DES. See [61] or [12], for example.

and R registers will agree after eight rounds with probability $1/2^{32}$. Hence, since there are 2^{64} messages, we expect there to be approximately $2^{64} \cdot 2^{-32} = 2^{32}$ fixed points for each weak key.[19]

To understand why a fixed point results for each weak key whenever the L and R registers agree after eight rounds, it is helpful to describe DES as a product of permutations

$$T_k = P^{-1}\pi(\pi h_{k_{16}}) \cdots (\pi h_{k_1})P, \tag{13}$$

where k is the 56-bit key, P is the initial permutation, and $k_1, \ldots, k_{16}$ are the sixteen 48-bit round keys derived from k. If k is weak, then $k_1 = k_2 = \cdots = k_{16}$. For all $1 \leq i \leq 16$, the ith round consists of the permutation πh_{k_i} where π, h_{k_i}: $\mathcal{M} \to \mathcal{M}$. It is especially convenient to define π and h_{k_i} in terms of their effects on the L and R registers. For any $r, s \in \{0, 1\}^{32}$, π is the "swap" function

$$\pi(r, s) = (s, r) \tag{14}$$

and h_{k_i} is the function

$$h_{k_i}(r, s) = (r \oplus f_{k_i}(s), s), \tag{15}$$

where f is DES's nonlinear function defined in [61]. Note that, for all round keys k_i, both π and h_{k_i} are self-inverse.

Let x be any message and let k be any weak key. If, during the computation of $T_k(x)$, the L and R registers agree after eight rounds, then the effect of rounds eight through nine on the computation of $T_k(x)$ is

$$(\pi h_{k_9})(\pi h_{k_8}) = (\pi h_{k_9})\pi(\pi h_{k_8}) = \pi h_{k_9} h_{k_8} = \pi h_{k_8} h_{k_8} = \pi. \tag{16}$$

By similar argument, it then follows that the effect of rounds seven through ten is also π. By induction, it follows that the effect of rounds one through sixteen is π. Hence, $T_k(x) = (P^{-1}\pi(\pi)P)(x) = x$. Note that fixed points arise not only when the round keys are equal, but also when they are "palindromic" in the sense that $k_i = k_{17-i}$ for all $1 \leq i \leq 8$.

After the conference we found the fixed points and thus confirmed Coppersmith's hypothesis (see Appendix A). To the best of our knowledge, these fixed points are the first published in the open literature. These fixed points further illustrate the deficiencies of the weak keys.

Coppersmith also suggested that the algebraic structure detected in experiment 7 can be used to prove strong lower bounds on the size of the group generated by DES. Experiment 7 computed the length, l, of the ψ-orbit of x_0, where $\psi = T_{w_1} T_{w_0}$ and x_0 is the initial message. Since l divides the order of ψ, it follows that l divides the order of the group generated by DES. Therefore, if experiment 7 were repeated r times with different initial messages, and if these experiments yielded orbit lengths $l_1, l_2, \ldots, l_r$, then $\text{lcm}(l_1, l_2, l_r)$ would be a lower bound on the order of the group generated by DES. We have not extended our results in this direction.

Motivated by our findings, Moore and Simmons carried out additional experiments to investigate the cycle structure of the weak and semiweak keys [39].

[19] Moore and Simmons have proven a stronger result: for each "palindromic" and "antipalindromic" key, there are *exactly* 2^{32} fixed points [39].

6.3. *Cycling Hardware*

We carried out each experiment on a IBM Personal Computer equipped with special-purpose hardware. Our hardware can compute a sequence of 2^{32} DES encryptions per day, where at each step the previous ciphertext is encrypted under a key that depends on the previous ciphertext. This section gives a summary description of our cycling hardware.[20]

Our goal was to implement the cycling closure test in the simplest way that would enable us to carry out each trial of the experiment within a few days. For each experiment we needed to compute about 2^{32} encryptions, changing the key at each step. For this application, software implementations of DES are too slow.[21] Moreover, commercially available DES boards are not suited for our purposes: to compute and load a new key for each encryption would require interaction by the host computer, introducing tremendous overhead. Therefore, we built our own hardware.

Our cycling hardware is a custom wire-wrap board for an IBM personal computer.[22] Our board contains a microprogrammed 7.1 MHz 32-bit finite-state controller and a single 3.6 MHz AMD AmZ8068 DES chip [54]. Data paths 8 bits wide connect the finite-state controller, the DES chip, a 16-byte buffer, a PROM computing the next-key function, an 8-bit counter, and the PC Bus interface to the host computer. To increase the board's flexibility, the controller's microprogram is stored in RAM accessible to the host computer. Figure 8 shows a simplified block diagram of our special-purpose hardware.

Each algebraic test is programmed in microcode for the board's finite-state controller. The next-key function is computed in a byte-by-byte fashion using a PROM, which can be easily replaced to implement different next-key functions. A read-write counter indicates the number of consecutive messages to compute. By periodically reading the board's counter, the host computer detects completion of the board's activity. Our board also supports all approved modes of operation for DES.

We performed cycle detection in two passes: data acquisition and analysis. During data acquisition, the host computer stored every 2^{20}th message on a floppy disk. During analysis, these messages were loaded into main memory, and up to 2^{20} consecutive messages were computed and compared with those already present. In effect we performed the Sedgewick–Szymanski algorithm [49] with a fixed estimate of the cycle length. We used an open-addressing, double-hashing scheme for stores and lookups [33]. All data acquisition and analysis routines were written in the C programming language.[23]

---

[20] See [29] for a complete description and schematic diagrams of our hardware.

[21] Software implementations of the DES for the IBM PC run at about 200–300 encryptions/second. According to Davio, by using a space-intensive implementation of DES, it is possible to perform about 2.5K encryptions/second on the VAX 11/780 [13]. Thus, it would take the IBM PC about 10–16 days to compute 2^{28} DES encryptions; a VAX 11/780 would require about a day and a half. Running the test for 2^{32} steps would take at least 16 times longer.

[22] We chose to use an IBM PC because an IBM PC was available to us, and because it is easy to attach special-purpose hardware to an IBM PC [63].

[23] See [27] for a complete description and listing of all supporting software.

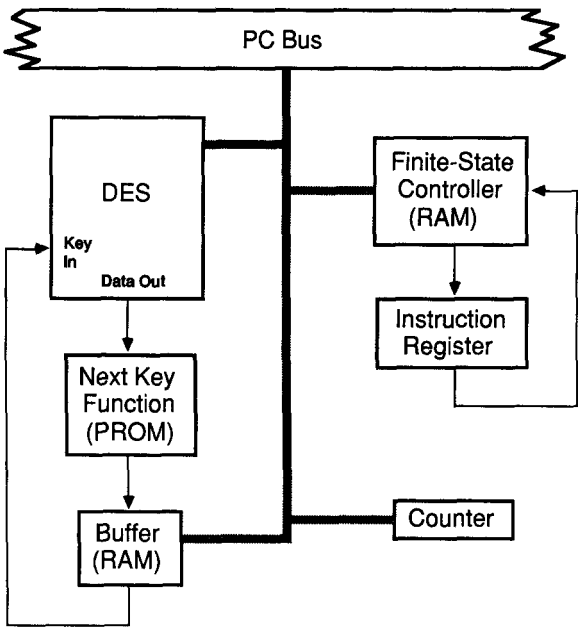

Fig. 8. Block diagram of special-purpose hardware.

Using cipher-block chaining in direct control mode [54], the AmZ8068 chip is capable of performing approximately 200K encryptions/second. But, for simplicity, we ran our experiments using electronic-codebook mode. We clocked our control loop at 140 ns and the inner encryption loop at 280 ns. Including all overhead for computing and loading a new key for each encryption, our board performed approximately 43K encryptions/second, or about 2^{32} per day. This enabled us to carry out each trial of the experiment within a few days.

7. Conclusion and Open Problems

We have developed two statistical tests for determining if a cryptosystem is closed under functional composition, and we have shown how each of these tests yields a known-plaintext attack against any group cipher. Moreover, by applying the cycling closure test to DES we have shown, with overwhelming confidence, that DES is neither closed nor pure. In the process we discovered fixed points for the weak keys. Although we had not expected to find fixed points for the weak keys, our experiments otherwise confirmed our expectations: we showed that DES is free of certain gross algebraic weaknesses, and we uncovered an additional weakness of a set of DES transformations already known to be weak.

Our tests are sufficient for determining whether or not DES is closed, but we do not know if the cycling closure test is the most efficient closure test either for DES or for cryptosystems in general. In particular, we do not know if the cycling closure test is an optimal solution to the *Group Detection Problem*, as formulated by

Sherman [51]. To settle this question, it would be interesting to analyze the orbit test further.

Although our known-plaintext attacks can be applied against any group cipher, and although it would be a devastating weakness for DES to be a group, it would be wrong to infer that all group ciphers are insecure. For example, the *RSA cryptosystem* [45] is also vulnerable to our attacks, since the cryptanalyst can proceed as if he were attacking the common modulus RSA cryptosystem, which is a group cipher.[24] However, such attacks are less efficient at breaking RSA than are attacks based on known factoring techniques [42], [35]. We view the RSA cryptosystem as evidence that, provided the key space is large enough to withstand our attacks, group ciphers are not necessarily insecure.

Similarly, it would be wrong to conclude that all algebraic properties of cryptosystems are necessarily bad. Our attacks show that, for any group cipher, there is a short-cut solution over exhaustive search of the key space, under a known-plaintext attack. Even though algebraic properties can give the cryptanalyst something to exploit, algebraic properties can also endow a cryptosystem with desirable capabilities or security properties. For example, the rich algebraic structure in the RSA cryptosystem makes it possible to perform public-key cryptography, to compute digital signatures, and to prove a double-edged "bit-security" property [7]. Although there are general relationships between algebraic and security properties of cryptosystems, how algebraic structure affects a cryptosystem depends in part on the particular cryptosystem.

We conclude by listing several open questions about the algebraic structure of DES:

- Does DES generate $\mathscr{A}_M$? What is the order of the group generated by DES? What is the group generated by DES? For how many keys i, j, k is it true that $T_k = T_i T_j$?
- Is DES faithful? How many distinct transformations are represented by the DES keys?
- What subsets of DES transformations generate small groups? (Note that each weak key represents a transformation that generates the cyclic group of order 2).
- Is DES *homogeneous* in the sense that for every $k \in \mathscr{K}$ it is true that $T_k^{-1} \in \mathscr{T}_{\text{DES}}$? For how many $k \in \mathscr{K}$ is it true that $T_k^{-1} \in \mathscr{T}_{\text{DES}}$?
- Is $I \in \mathscr{T}_{\text{DES}}$?

Our results show that the composition of every pair of weak keys will likely have a short orbit for every message. It would also be interesting to know if other special pairs of DES transformations have similar properties. For example, it would be interesting to explore semiweak keys, *light keys* (keys with a low density of ones), *heavy keys* (keys with a high density of ones), and pairs of related keys (e.g., keys that differ in exactly one bit and keys that are complements of each other).

Knowing whether or not $I \in \mathscr{T}_{\text{DES}}$ is interesting—not because this property would necessarily be a weakness in DES—but because this question would answer several

[24] The *common modulus RSA cryptosystem* is a variation of the RSA cryptosystem in which the same modulus is used for every key.

other questions about DES. By the complementation property, for any key k, $T_k = I$ implies $T_{\bar{k}} = I$. Hence, if $I \in \mathcal{T}_{\text{DES}}$, then DES is not faithful. In particular, if DES is closed, then DES is not faithful. Conversely, if $I \notin \mathcal{T}_{\text{DES}}$, then DES is not closed.

Each of the known-plaintext attacks finds a representation of the secret transformation T_k as a product of two or more transformations. In practice, it would suffice to find an approximate representation of T_k. To this end we could say that two permutations $T_1, T_2 \in \mathcal{T}_{\text{DES}}$ are *q-approximately equal on* $X \subseteq \mathcal{M}$ if and only if, for all $x \in X$, $T_1(x)$ and $T_2(x)$ always agree on at least q bits.

- For each $1 \leq q \leq 64$, for how many keys i, j, k is it true that T_k is *q-approximately equal to* $T_i T_j$ on $\mathcal{M}$?
- What other notions of "approximately equal" transformations would be useful in finding approximate representations?

Since our closure tests do not depend on the detailed definition of DES, it is natural to ask:

- What can be proven from the detailed definition of DES about the order of the group generated by DES?
- Are there more powerful statistical closure tests than the two tests presented in this paper that are based on the detailed definition of DES?

Our research also raises questions involving the design of cryptosystems:

- It is possible to build a secure, practical cryptosystem for which it can be proven that the cryptosystem generates either $\mathcal{A}_{\mathcal{M}}$ or $\mathcal{S}_{\mathcal{M}}$? (see [8] for one suggestion.)
- Is it possible to hide a trapdoor in a cryptosystem by concealing a secret set of generators for a small group? (Note that it does not work simply to have a large subset of the transformations generate a small group, since the enemy could guess a small number of transformations in the subset and apply the cycling closure test to the guessed transformations.)

We presented two known-plaintest attacks against closed ciphers, but other attacks may also exist.

- What attacks are possible against closed ciphers? How can knowledge of the specific group help?

Finally, it would be interesting to apply the closure tests to variations of DES that exaggerate certain types of possible weaknesses in the standard.

- What is the order of "crippled" DES transformations formed by reducing the number of rounds or by replacing one or more of the S-boxes with linear mappings?

Acknowledgments

We would like to thank several people who contributed to this research. Leon Roisenberg assisted Burt with the design and construction of the cycling hardware, and John Hinsdale wrote some of the supporting software. The cycling hardware was built in the hardware laboratory of the Functional Languages and Architecture Research Group at the MIT Laboratory for Computer Science. All experiments were carried out on an IBM Personal Computer, which was one of several denoted

by the International Business Machines Corporation to the MIT Laboratory for Computer Science. In addition, we are grateful to László Babai, Don Coppersmith, and Gary Miller for helpful comments.

Appendix A: Detailed Descriptions of Experiments

This appendix presents detailed descriptions of the eight cycling experiments we carried out during April–August, 1985. The appendix begins with an explanation of the next-key functions used in our experiments; the rest of the appendix consists of nine tables that thoroughly document our experiments.

Notation

In Section 1 we defined the key space of DES to be the set $\mathcal{K} = \{0, 1\}^{56}$. Most DES implementations, however, nominally treat each key as a string of 64 bits, where every eighth key bit is a *parity bit* which is ignored. In this appendix we too shall specify keys and messages as 64-bit strings, described in hexadecimal notation. To do this, it is convenient to introduce the DES function $\hat{T}: \hat{\mathcal{K}} \times \mathcal{M} \to \mathcal{M}$ that operates on the nominal key space $\hat{\mathcal{K}} = \{0, 1\}^{64}$.

Next-Key Functions

The cycling closure test depends on a function $\rho: \mathcal{M} \to \mathcal{K}$ to compute the next-key from the current message. We will now describe the two particular *next-key functions* that we used in our experiments. We will define each next-key function in terms of its related function $\hat{\rho}: \mathcal{M} \to \hat{\mathcal{K}}$.

Each next-key function operated in a byte-by-byte fashion using a byte substitution table (1 byte = 8 bits). For any $0 \leq i \leq 7$ and any $x \in \mathcal{M}$, let $x^{(i)}$ denote the ith byte of x. For each $0 \leq i \leq 7$, we computed $\hat{\rho}(x)^{(i)} = S(x^{(i)})$, for some byte substitution table S: $\{0, 1\}^8 \to \{0, 1\}^8$.

In experiments 1 and 2 we chose S to be the identity function. In the other cycling closure experiments we used the byte substitution table given by Table 2.[25] This table was designed so that each entry has odd parity and such that each entry appears exactly twice. The table was generated using the random number generator in the C library on our IBM PC.

For the experiments that used the extended message space $\mathcal{M}^2$, we computed $\hat{\rho}(x)^{(i)} = S(x^{(2i)})$ using the substitution table given in Table 2.

Detailed Experimental Results

Tables 2–10 give thorough descriptions and results of our cycling experiments. Table 2 defines the pseudorandom next-key function used in several of the experiments. The remaining eight tables—one for each experiment—list all relevant experimental parameters together with important checkpoints encountered during the experiments. Initial messages and keys were chosen in a variety of *ad hoc* ways.

[25] The substitution table is used as follows. To substitute any byte B, consider the representation of B as two hexadecimal digits. Select the table entry whose row is given by the first digit and whose column is given by the second digit.

Table 2. Byte substitution table for pseudorandom next-key function.

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00	3E	46	B6	26	AE	F8	2A	AE	CE	57	E6	98	07	5D	92	2C
10	FE	58	EF	CD	F7	76	2F	91	8F	2F	0E	D0	07	B0	73	51
20	20	5E	76	B3	86	9D	16	01	31	EF	D3	8F	D6	40	2A	F8
30	01	C7	C7	19	F7	31	A2	62	9E	B9	DA	D9	34	85	19	D9
40	61	A8	3D	B0	0E	79	C2	BC	52	04	37	FD	6E	85	FB	BA
50	DF	C8	6D	13	43	1C	0B	4A	89	83	E3	20	4F	A7	BA	3B
60	80	D0	67	EA	7F	A8	C8	43	79	6D	1A	4C	A7	CB	86	23
70	5B	02	C2	4C	58	38	FE	CE	B9	1C	15	A4	25	29	1A	15
80	C1	98	7F	4A	64	57	97	32	26	F2	E5	91	D6	E9	6B	F4
90	4F	80	67	DF	F1	BF	B3	B5	3E	E5	7A	EC	A1	B5	92	29
A0	10	DC	97	46	94	CB	49	6B	10	45	3B	F2	E6	FD	B6	BC
B0	40	0D	1F	AD	52	BF	62	23	61	49	E0	0D	08	CD	E3	C4
C0	68	1F	9E	E9	FB	7C	13	75	8A	89	04	5D	6E	DC	54	D5
D0	EA	F1	9D	F4	94	75	D3	70	8C	54	AB	2C	D5	02	98	7A
E0	3D	5B	25	8A	A1	38	8C	EC	70	9B	A4	45	64	51	AB	7C
F0	C1	AD	34	C4	E0	A2	68	83	16	08	DA	32	73	37	0B	5E

Table 3. Closure experiment with identity next-key function.
Experiment 1: $x_{i+1} = \hat{T}_{x_i}(x_i)$.*

i	x_i	Note
0	0123456789ABCDEF	
34,293,588	B0FDED3BD0DD918C	End of leader
34,293,589	AE5530A0E971B5E8	Start of cycle
2,030,556,568	12B67D3796106D30	Quarter cycle
4,026,819,547	51AACF5F168E4A17	Half cycle
6,023,082,526	ED4982C869EF92CF	Three-quarters cycle
8,019,345,504	A032CE0D3F436EFE	End of cycle
8,019,345,505	AE5530A0E971B5E8	Restart of cycle

* Cycle length 7,985,051,916 $\approx 2^{33}$; leader length 34,293,589 $\approx 2^{25}$.

Table 4. Closure experiment with identity next-key function.
Experiment 2: $x_{i+1} = \hat{T}_{x_i}(x_i)$.*

i	x_i	Note
0	121502850B020664	
1,389,523,413	48BB5C9F85CD285A	End of leader
1,389,523,414	AFF50E97653421BF	Start of cycle
5,152,082,299	AE5530A0E971B5E8	Experiment 1 intersection
9,374,575,329	FB0A1398E92D1473	End of cycle
9,374,575,330	AFF50E97653421BF	Restart of cycle

* Cycle length 7,985,051,916 $\approx 2^{33}$; leader length 1,389,523,414 $\approx 2^{30}$.

Table 5. Closure experiment with pseudorandom next-key function. Experiment 3: $x_{i+1} = \hat{T}_{\hat{p}(x_i)}(x_i)$.*

i	x_i	Note
0	6036222982B03104	
2,138,241,978	68955F4BF000A6E0	End of leader
2,138,241,979	C9DB8E7169CCF272	Start of cycle
3,706,679,992	433B74E2CB18DDFD	end of cycle
3,706,679,993	C9DB8E7169CCF272	Restart of cycle

* Cycle length 1,568,438,014 $\approx 2^{30.5}$; leader length 2,138,241,979 $\approx 2^{31}$.

Table 6. Extended closure experiment with pseudorandom next-key function. Experiment 4: $x_{i+1} = \hat{T}_{\hat{p}(x_i)}(x_i)$, $x_i \in \mathcal{M}^2$.*

i	x_i		Note
0	4C957F303AC4D08B	63E15C9C7A398042	
4,294,967,296	2C173869EAF8804B	767469BB19B26D8A	2^{32} iterations
8,589,934,592	4349368A49700D3B	55FC02F8848BC64F	2^{33} iterations
12,884,901,888	55D1292F5D99B268	C30AB80FF3B03D08	3×2^{32} iterations
17,179,869,184	4A224C65B8A48DEB	00C7D0CA64C4B240	2^{34} iterations

* No cycle detected in 2^{34} steps.

Table 7. Purity experiment with pseudorandom next-key function. Experiment 5: $x_{i+1} = \hat{T}_{\hat{k}}^{-1}\hat{T}_{\hat{p}(x_i)}(x_i)$.*

i	x_i	Note
0	0123456789ABCDEF	
3,233,340,362	0EC45F7157BD8749	End of leader
3,233,340,363	EFE7B7112233DD88	Start of cycle
4,531,729,424	C09DFA478C3849BE	End of cycle
4,531,729,425	EFE7B7112233DD88	Restart of cycle

* Cycle length 1,298,389,062 $\approx 2^{30}$; leader length 3,233,340,363 $\approx 2^{31.5}$. Key $\hat{k}$ = 97778E1BC3FD8E07.

Table 8. Purity experiments with pseudorandom next-key function. Experiment 6: $x_{i+1} = \hat{T}_{\hat{k}}^{-1}\hat{T}_{\hat{p}(x_i)}(x_i)$.*

i	x_i	Note
0	121502850B020664	
1,366,287,307	E43D6EF9351DDB4A	End of leader
1,366,287,308	75C6C23C21EA50DA	Start of cycle
5,584,675,814	FDBE1ECDF38BF3E5	End of cycle
5,585,675,815	75C6C23C21EA50DA	Restart of cycle

* Cycle length 4,218,388,507 $\approx 2^{32}$; leader length 1,366,287,308 $\approx 2^{30}$. Key $\hat{k}$ = 4D3FD0FED9A4FA9B.

Table 9. Orbit experiment using composition of weak keys. Experiment 7: $x_{i+1} = \hat{T}_{1\cdots1}(\hat{T}_{0\cdots0}(x_i))$.*

i	x_i	Note
0	0123456789ABCDEF	Start of cycle
2,227,161,945	654B672D3DBC73AB	$0\cdots0$ fixed point
4,454,323,890	293FD4F2C13DD94F	"Hidden crossing"
5,890,012,565	3CC5B06ADEFD30A0	$1\cdots1$ fixed point
7,325,701,239	0123456789ABCDEF	Restart of cycle

* Cycle length 7,325,701,239 $\approx 2^{33}$; leader length 0.

Table 10. Orbit experiment. Experiment 8: $x_{i+1} = \hat{T}_{\hat{k}}(x_i)$.*

i	x_i	Note
0	41184DCAB17324C8	
17,179,869,184	B98C3A67CD6F8267	2^{34} iterations
34,359,738,368	632509BC9F57DF8A	2^{35} iterations
51,539,607,552	ED4B06ABBF5515FB	3×2^{34} iterations
68,719,476,736	2C84263510AEAD34	2^{36} iterations

* No cycle detected in 2^{36} steps. Key $\hat{k}$ = 116E0B8278AEC431.

References

[1] Beker, H., and F. Piper, *Cipher Systems: The Protection of Communications*, Wiley, New York, 1982.

[2] Bovey, J. D., An approximate probability distribution for the order of elements of the symmetric group, *Bulletin of the London Mathematical Society*, **12** (1980), 41–46.

[3] Bovey, J., and A. Williamson, The probability of generating the symmetric group, *Bulletin of the London Mathematical Society*, **10** (1978), 91–96.

[4] Brent, R. P., Analysis of some new cycle-finding and factorization algorithms, Technical Report, Department of Computer Science, Australian National University (1979).

[5] Carmichael, R. D., *Introduction to the Theory of Groups of Finite Order*, Dover, New York, 1956.

[6] Chandra, A. K., Efficient compilation of linear recursive programs, Technical Report STAN-CS-72-282, Computer Science Department, Stanford University (April 1972).

[7] Chor, B.-Z., *Two Issues in Public-Key Cryptography: RSA Bit Security and a New Knapsack Type Cryptosystem*, MIT Press, Cambridge, MA, 1985.

[8] Coppersmith, D., and E. Grossman, Generators for certain alternating groups with applications to cryptology, *Siam Journal on Applied Mathematics*, **29** (1975), 624–627.

[9] Davies, D. W., Some regular properties of the DES, in [55], 89–96.

[10] Davies, D. W., and G. I. P. Parkin, The average size of the key stream in output feedback encipherment, in [59], 263–279.

[11] Davies, D. W., and G. I. P. Parkin, The average size of the key stream in output feedback mode, in [55], 97–98.

[12] Davies, D. W., and W. L. Price, *Security for Computer Networks: An Introduction to Data Security in Teleprocessing and Electronic Funds Transfer*, Wiley, Chichester, 1984.

[13] Davio, M. Y. Desmedt, J. Goubert, F. Hoornaert, and J.-J. Quisquater, Efficient hardware and software implementations for the DES, in [56], 144–146.

[14] Diffie, W. and M. E. Hellman, Exhaustive cryptanalysis of the NBS Data Encryption Standard, *Computer*, **10** (1977), 74–84.

[15] Diffie, W., and M. E. Hellman, Privacy and authentication: an introduction to cryptography, *Proceedings of the IEEE*, **67** (1979), 397–427.
[16] Dixon, J. D., The probability of generating the symmetric group, *Math Zentrum*, **110** (1969), 199–205.
[17] Feldman, F., A new spectral test for nonrandomness and the DES, *IEEE Transactions on Software Engineering*, to appear.
[18] Feller, W., *An Introduction to Probability Theory and Its Applications*, vol. I, Wiley, New York, 1968.
[19] Gaines, H. F., *Cryptanalysis: A Study of Ciphers and Their Solution*, Dover, New York, 1956.
[20] Gait, J., A new nonlinear pseudorandom number generator, *IEEE Transactions on Software Engineering*, **3** (1977), 359–363.
[21] Goldreich, O., DES-like functions can generate the alternating group, *IEEE Transactions on Information Theory*, **29** (1983), 863–865.
[22] Good, I. J., *The Estimation of Probabilities: An Essay on Modern Bayesian Methods*, MIT Press, Cambridge, MA, 1965.
[23] Harris, B., Probability distributions related to random mappings, *Annals of Mathematical Statistics*, **31** (1959), 1045–1062.
[24] Hellman, M. E., R. Merkle, R. Schroeppel, L. Washington, W. Diffie, S. Pohlig, and P. Schweitzer, Results of an initial attempt to cryptanalyze the NBS Data Encryption Standard, Technical Report SEL 76–042, Information Systems Laboratory, Stanford University (November 1976).
[25] Hellman, M. E., A cryptanalytic time–memory tradeoff, *IEEE Transactions on Information Theory*, **26** (1980), 401–406.
[26] Hellman, M. E., and J. M. Reyneri, Distribution of drainage in the DES, in [55], 129–131.
[27] Hinsdale, J. K., Implementing the Sedgewick–Szymanski cycle detection algorithm, B.Sc. thesis, Department of EECS, MIT (February 1985).
[28] Jueneman, R. R., Analysis of certain aspects of output-feedback mode, in [55], 99–127.
[29] Kaliski, B. S., Jr., Design and reliability of custom hardware for DES cycling experiments, M.Sc. thesis, Department of EECS, MIT (January 1987).
[30] Kaliski, B. S., Jr., R. L. Rivest, and A. T. Sherman, Is the Data Encryption Standard a group?, in [60], 81–95.
[31] Kaliski, B. S., R. L. Rivest, and A. T. Sherman, Is the Data Encryption Standard a pure cipher? (Results of more cycling experiments on DES), in [57], 212–226.
[32] Knuth, D. E., *The Art of Computer Programming*, vol. II: *Seminumerical algorithms*, Addison-Wesley, Reading, MA, 1981.
[33] Knuth, D. E., *The Art of Computer Programming*, vol. III: *Sorting and searching*, Addison-Wesley, Reading, MA, 1973.
[34] Kolata, G., Codes go public, *Boston Globe* (September 30, 1985), 44.
[35] Lenstra, H. W., Jr., Factoring integers with elliptic curves, *Annals of Mathematics*, to appear.
[36] Longo, G., ed., *Secure Digital Communications*, Springer-Verlag, Vienna, 1983.
[37] Merkle, R. C., and M. E. Hellman, On the security of multiple encryption, *Communications of the Association for Computing Machinery*, **24** (July 1981), 465–467.
[38] Meyer, C. H., and S. M. Matyas, *Cryptology: A New Dimension in Computer Data Security*, Wiley, New York, 1982.
[39] Moore, J. H., and G. J. Simmons, Cycle structure of the DES with weak and semi-weak keys, in [58], 3–32.
[40] Osteyee, D. B., and I. J. Good, *Information, Weight of Evidence, the Singularity Between Probability Measures and Signal Detection*, Springer-Verlag, Berlin, 1974.
[41] Pollard J. M., A Monte Carlo method for factorization, *Bit*, **15** (1975), 331–334.
[42] Pomerance, C., Analysis and comparison of some integer factoring algorithms, in *Computational Methods in Number Theory*, H. W. Lenstra, Jr., and R. Tijdeman, eds., Math. Centrum Tract 154, Amsterdam, 1982, 89–139.
[43] Purdom, P. W., Jr., and C. A. Brown, *The Analysis of Algorithms*, Holt, Rinehart, and Winston, New York, 1985.
[44] Purdom, P. W., and J. H. Williams, Cycle length in a random function, *Transactions of the American Mathematical Society*, **133** (1968), 547–551.

[45] Rivest, R., A. Shamir, and L. Adleman, On digital signatures and public-key cryptosystems, *Communications of the Association of Computing Machinery*, **21** (1978), 120–126.
[46] Rotman, J. J., *The Theory of Groups: An Introduction*, Allyn and Bacon, Boston, 1978.
[47] Sattler, J., and C. P. Schnorr, Generating random walks in groups, unpublished manuscript (October 1983).
[48] Shannon, C. E., Communication theory of secrecy systems, *Bell System Technical Journal*, **28** (1949), 656–715.
[49] Sedgewick, R. T. G. Szymanski, and A. C. Yao, The complexity of finding cycles in periodic functions, *Siam Journal on Computing*, **11** (1982), 376–390.
[50] Shepp, L. A., and S. P. Lloyd, Ordered cycle lengths in a random permutation, *Transactions of the American Mathematical Society*, **121** (1966), 340–357.
[51] Sherman, A. T., Cryptology and VLSI (a two-part dissertation): I. Detecting and exploiting algebraic weaknesses in cryptosystems. II. Algorithms for placing modules on a custom VLSI chip, Technical Report TR–381, MIT Laboratory for Computer Science (October 1986).
[52] Tuchman, W. L., talk presented at the National Computer Conference (June 1978).
[53] Wielandt, H., *Finite Permutation Groups*, Academic Press, New York 1964.
[54] *Data Ciphering Processors Am*9518, *Am*9568, *AmZ*8068 *Technical Manual*, Advanced Micro Device Inc., Sunnyvale, CA (1984).
[55] Chaum, D., R. L. Rivest, and A. T. Sherman, eds., *Advances in Cryptology: Proceedings of Crypto* 82, Plenum, New York, 1983.
[56] Blakley, G. R., and D. Chaum, eds., *Advances in Cryptology: Proceedings of Crypto* 84, Springer-Verlag, New York, 1985.
[57] Williams, H. C., ed., *Advances in Cryptology: Proceedings of Crypto* 85, Springer-Verlag, New York, 1986.
[58] Odlyzko, A., ed., *Advances in Cryptology: Proceedings of Crypto* 86, Springer-Verlag, New York, 1987.
[59] Beth, T., ed., *Cryptography, Proceedings of the Workshop on Cryptography, Burg Feuerstein, Germany, March* 29–*April* 2, 1982, Springer-Verlag, Berlin, 1983.
[60] Pichler, F., ed., *Advances in Cryptology: Proceedings of Eurocrypt* 85, Springer-Verlag, Berlin, 1986.
[61] *Data Encryption Standard*, Federal Information Processing Standards Publications 46, National Bureau of Standards, U.S. Department of Commerce, Washington, DC (January 15, 1977).
[62] *DES Modes of Operation*, Federal Information Processing Standards Publication 81, National Bureau of Standards, U.S. Department of Commerce, Washington, DC (December 1980).
[63] *IBM Personal Computer Technical Reference*, Bocaraton, FL (July 1982).
[64] Unclassified summary: involvement of NSA in the development of the Data Encryption Standard, Staff Report of the Senate Select Committee on Intelligence, United States Senate (April 1978).