Is the Data Encryption Standard a Group?[1] (Preliminary Abstract)[2]

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

MIT Laboratory for Computer Science
545 Technology Square
Cambridge, MA 02139

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 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.

We describe two statistical tests 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 test takes a pseudo-random walk in the message space until a cycle is detected. For each step of the pseudo-random walk, the previous ciphertext is encrypted under a key chosen by a pseudo-random 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, we applied the cycling test to DES. Our experiments show, with a high degree of confidence, that DES is not a group.

Key Words and Phrases

Birthday Paradox, closed cipher, cryptanalysis, cycle-detection algorithm, Data Encryption Standard (DES), finite permutation group, idempotent cryptosystem, multiple encryption, pure cipher.

[1]This research was supported by NSF grant MCS-8006938 and IBM.

[2]A revised and more detailed version of this paper will be available from the authors sometime in the future. In August 1985, the authors reported results of additional cycling experiments on DES at the Crypto 85 conference at the University of California, Santa Barbara [41].

F. Pichler (Ed.): Advances in Cryptology - EUROCRYPT '85, LNCS 219, pp. 81-95, 1986.

1 Introduction

On November 23, 1976, the United States National Bureau of Standards (NBS) adopted the Data Encryption Standard (DES) as a federal standard for the cryptographic protection of computer data [2] [28].[3] Although a few studies on DES have been openly published [4] [30] [35] [38],[4] to date, 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?"

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.[5] 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. Although most researchers believe DES is not closed, no one has proven this conjecture in the open literature.

In this paper we present two statistical tests for determining if a finite, deterministic cryptosystem is a closed under functional composition. The first test is based on a "meet-in-the-middle" strategy and takes $O(\sqrt{K})$ time and space, where K is the size of the key space. The second test follows a pseudo-random walk in the message space until a cycle is detected, using $O(\sqrt{K})$ time and constant space. Although we focus on DES, the methods presented here are general in nature.

Using a combination of software and special-purpose hardware, we applied the cycling test to DES. Our initial experiments revealed no algebraic weaknesses with DES.

The body of this paper is organized in six sections. Section 2 discusses the contrasting properties of closed and random ciphers. Section 3 presents two statistical closure tests. Section 4 describes how each test can be modified into a known-plaintext attack against closed ciphers. Section 5 lists our initial experimental results and explains how to interpret them. Section 6 poses several open problems, and section 7 summarizes our conclusions. An appendix, which briefly describes our implementation of the cycling test, is also included.

1.1 Definitions and Notations

A *(finite, deterministic) cryptosystem* is an ordered 4-tuple $(\mathcal{K}, \mathcal{M}, \mathcal{C}, T)$, where $\mathcal{K}$, $\mathcal{M}$, and $\mathcal{C}$ are finite sets called the *key space*, *message space*, and *ciphertext space*, and $T : \mathcal{K} \times \mathcal{M} \rightarrow \mathcal{C}$ is a transformation such that, for each $k \in \mathcal{K}$, 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* iff the message space and ciphertext space are the same set.

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

We shall use the following notations throughout the paper. For any cryptosystem $\Pi = (\mathcal{K}, \mathcal{M}, \mathcal{C}, T)$, let $\mathcal{T}_\Pi = \bigcup\{T_k : k \in \mathcal{K}\}$ be the set of all encryption transformations, and let $G_\Pi = \langle \mathcal{T}_\Pi \rangle$ be the group generated by Π. For any transformation $T_k \in \mathcal{T}_\Pi$, let T_k^{-1} denote the inverse of T_k. In addition, let $K = |\mathcal{K}|$ be the size of the key space; let $M = |\mathcal{M}|$ be the degree of Π; and let $m = |\mathcal{T}_\Pi|$ be the order of Π. Whenever the meaning is clear, we will omit the subscript Π.

Let I be the identity permutation on $\mathcal{M}$, and let $\mathcal{A}_\mathcal{M}$ and $\mathcal{S}_\mathcal{M}$ be, respectively, the *alternating group* and *symmetric group* on $\mathcal{M}$ [13]. For any permutations g, h we will denote the composition of g and h by

[3] We expect the reader to be familiar with the fundamentals of cryptology (as presented in [3] or [1], for example), as well as with the basics of DES (as described in [2] or [4], for example).

[4] See bibliography for a list of additional technical works on DES.

[5] 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 [44] [4] [42].

$gh = g \circ h = g[h(\cdot)]$.

An endomorphic cryptosystem is *closed* iff its set of encryption transformations is closed under functional composition.[6] Shannon's notion of a pure cipher generalizes the idea of closure to non-endomorphic cryptosystems [57]. A cryptosystem $\Pi = (K, \mathcal{M}, \mathcal{C}, T)$ is *pure* iff, for every $T_0 \in T_\Pi$, the set $T_0^{-1} T_\Pi$ is closed.[7] Every closed cryptosystem is pure, but not every endomorphic pure cryptosystem is closed (see section 2.2).

To analyze the cycling test, it is useful to introduce the following standard terminology from permutation group theory [13] [15] [16]. For any subgroup $G \subseteq S_\mathcal{M}$, for any $x \in \mathcal{M}$, the *G-orbit of x* is the set $G\text{-orbit}(x) = \{g(x) : g \in G\}$ and the *G-stabilizer of x* is the set $G\text{-stabilizer}(x) = \{g \in G : g(x) = x\}$. If f is any function (not necessarily a permutation) and if $x \in \text{Domain}(f)$, the *f-closure of x* is the set $f\text{-closure}(x) = \{f^i(x) : i \geq 0\}$. For any permutation $g \in S_\mathcal{M}$, we will sometimes write $g\text{-orbit}(x)$ to denote the $\langle g \rangle$-orbit of x. For any subgroup $G \subseteq S_\mathcal{M}$, the *order of G* is the number of elements in G; for any $g \in S_\mathcal{M}$, the *order of g* is the order of $\langle g \rangle$.

Whenever $T \subseteq S_\mathcal{M}$, we say T *acts transitively on* $\mathcal{M}$ iff, for every pair of messages $x, y \in \mathcal{M}$, there exists some transformation $T_k \in T$ such that $T_k(x) = y$.

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

The Data Encryption Standard defines a particular endomorphic cryptosystem with $\mathcal{M} = \mathcal{C} = \{0,1\}^{64}$ and $K = \{0,1\}^{56}$. Because DES has degree 2^{64}, but order at most 2^{56}, DES is intransitive. It is unknown if DES is faithful, closed, or pure. It is also unknown whether or not any DES transformation is the identity permutation. See NBS FIPS publication 46 [28] or most any cryptography survey work (*e.g.* [2] or [4]) for a detailed definition of the DES encryption function.

1.2 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 $G_{DES} \subseteq A_\mathcal{M}$ [48], but no one has disproved the possibility that $G_{DES} = T_{DES}$.[8]

There are several reasons to suspect DES is not closed. First, Coppersmith and Grossman proved "DES-like" permutations generate the alternating group [48].[9] Second, if even just two permutations are chosen at random from $S_\mathcal{M}$, then there is an overwhelming chance (greater than $1 - e^{-\sqrt{M}}$) that these permutations generate either $A_\mathcal{M}$ or $S_\mathcal{M}$ [12] [14]. Third, no one has announced finding any three keys $i, j, k \in K$ 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" [58].

On the other hand, DES is not a set of randomly chosen permutations, and Coppersmith and Grossman did not prove that DES generates $A_\mathcal{M}$. Furthermore, DES is known to have the following three regularities [2] [4] [30] [38].

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

[6] Note that we are using the term *closed cipher* to refer to what Shannon calls an *idempotent cipher* [57]. Shannon defines a closed cipher to be any cryptosystem with the property that each cryptographic transformation is surjective.

[7] Shannon defines purity in a different but essentially equivalent way. Shannon also requires each transformation of a pure cipher to be equally likely.

[8] To see that $G_{DES} \subseteq A_\mathcal{M}$, note that each round of DES is an even permutation.

[9] See Goldreich's paper [37] for a minor extension of this result.

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

1.3 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 answered the question, "Does DES generate a small group?"

The analysis of each of these previous experiments depends heavily on the following two facts [8] [10] ([20], 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 about $\sqrt{M}$. (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 about $M/2$. (This is true because, for any $1 \leq l \leq M$, the probability that the cycle containing x_0 has length exactly l is $1/M$.)

Gait [36] investigated the statistical properties of pseudo-random key streams produced by DES in output-feedback mode [29]. 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 $10^6 \cong 2^{20}$ places. Gait found no cycles for nonweak keys.[10] Unfortunately, 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 pseudo-sequences he computed from non-weak keys. Gait observed that each of these sequences was considered random by his test.

Provided a feedback width of 64 bits is used, the cycling study considered by Gait can be viewed as a closure test. If DES were closed, then each of the orbits considered by Gait would have at most $K = 2^{56}$ messages (see lemma 2.2). Hence, observing an orbit of length greater than 2^{56} would be direct proof that DES is not closed. Although we will not do so in this preliminary abstract, it is also possible to interpret Gait's orbit test as a statistical closure test. Viewed as a statistical closure test, the orbit test can be strengthened by combining the test with tests for other algebraic properties.

Davies and Parkin [31] [32] and Jueneman [40] 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 about 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 about 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 better understand how effectively the Hellman time-space tradeoff [53] could be applied to DES, Hellman and Reyneri [39] examined the cycle structure of mappings induced by DES on the keyspace. Specifically, they considered mappings $F_x : \mathcal{K} \rightarrow \mathcal{K}$ defined by $F_x(k) = \rho(T_k(x))$, where $\rho : \mathcal{M} \rightarrow \mathcal{K}$ is a projection[11] 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 $x_i = T_{k_i} T_{k_{i-1}} \ldots T_{k_1}(x_0)$ for $i = 1, 2, \ldots$ where at each

[10] 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.

[11] Hellman and Reyneri used the projection that removes each of the 8 parity bits.

step i the key k_i is chosen as a pseudo-random function of the previous ciphertext x_{i-1}.

2 Closed Ciphers versus Random Ciphers

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

2.1 Algebraic Properties of Closed and Random Ciphers

Since every finite cancellation semigroup is a group [15], any endomorphic cryptosystem is closed iff 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.

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 2.1 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 $T_k = T_iT_j$; 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 $T_k = T_iT_j$. This lemma provides the basis of the meet-in-the-middle closure test.

Lemma 2.1 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 $T_i, T_j \in \mathcal{T}$ such that $T_iT_j = T_k$. If $\mathcal{T}$ is selected at random from $S_{\mathcal{M}}$, then the expected number of pairs of transformations $T_i, T_j \in \mathcal{T}$ such that $T_iT_j = T_k$ is $m^2/M!$.

Proof. Part 1: Assume Π is closed. For every transformation $T_i \in \mathcal{T}$, there is exactly one transformation $T_j \in \mathcal{T}_\Pi$ such that $T_iT_j = T_k$. 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 $1/|S_{\mathcal{M}}|$ 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 keyspace. 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 keyspace $\mathcal{K}$. 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 underlying cryptosystem is unfaithful, then $m < K$.

The next lemma describes the structure imposed on the message space by any closed cipher; specifically, lemma 2.2 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. (See section 1.1 for a review of some basic definitions from permutation group theory.)

Lemma 2.2 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 $\mathcal{T}$-orbits of $\mathcal{M}$ 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. $\mathcal{T}$ 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 $\mathcal{T}$-stabilizer of x.

Proof. (Sketch) For each $x \in \mathcal{M}$, consider the left cosets of H_x in $\mathcal{T}$ [15]. ∎

[12] This section draws heavily from basic results in permutation group theory and from Shannon's classic paper [57] [55].

Corollary 2.3 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}. ∎

To implement the cycling test, it is especially convenient that order(DES) < degree(DES). Note, however, that for any cryptosystem one can create a similar situation by considering the action of the set of transformations on the Cartesian product M^l, for a sufficiently large integer $l \geq 1$.

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 2.4 Let $\Pi = (K, M, M, T)$ be any endomorphic cryptosystem of order m, let $p \in M$ be any message, let $k \in K$ 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 T-orbit of p, and H_p is the T-stabilizer of p. If T_Π is chosen at random, then the expected number of transformations that map p to c is m/M.
Proof. Part 1: (Sketch) By lemma 2.2 and the fact that, for any $x, y \in B_p$, $|\{T_i \in T_\Pi : T_i(x) = y\}| = |\{T_i \in T_\Pi : T_i(p) = c\}|$. Note that $|H_p| = |H_c|$. Part 2: Each transformation in T other than T_k maps p to c with probability $1/M$. ∎

2.2 Closed Ciphers: Two Examples

One interesting example of a closed cipher is a single-key variation of the RSA cryptosystem [56] in which the same modulus is used for every key. Only the encryption exponent varies. In this cryptosystem, the modulus n is chosen to be the product of two large primes p, q. The message space is the multiplicative group modulo n, and the key space is the set of all integers $1 < e < \phi(n)$ such that e has a multiplicative inverse modulo $\phi(n)$, where $\phi(n) = (p-1)(q-1)$ is the *totient* function. The encryption function is defined by $T_{e,n}(x) = x^e \bmod n$. It is easy to verify that this cryptosystem is closed.

Although this variation of RSA is vulnerable to the known-plaintext attacks described in this paper, these attacks are less efficient at breaking the cryptosystem than are known factoring techniques [23]. We view this example as evidence that, provided the key space is large enough to withstand an $O(\sqrt{K})$ time and space attack, closed ciphers are not necessarily insecure. Of course, the security of this variation of RSA remains to be further evaluated [49].

Simple substitution [50] is also a closed cipher. Note that the restriction of simple substitution where the letter 'A' is always mapped to 'B' is an endomorphic system that is pure but not closed.

3 Statistical Closure Tests

In this section we describe two statistical tests for determining if an indexed set of permutations T generates a small group. Each test tries to distinguish between the two competing hypotheses: "T is closed" and "T was selected at random." Both tests are based heavily on the Birthday Paradox.

3.1 The Birthday Paradox

The Birthday Paradox [6] involves the question, "If r people are selected at random, what is the chance that no two people will have the same birthday?" If birthdays are independently and uniformly distributed between 1 and m, then the answer to this question is about $p_r = 1 - \frac{1}{m}\binom{r}{2}$, since there are $\binom{r}{2}$ pairs of people and each pair has a $1/m$ chance of having the same birthday. This approximate analysis, however, ignores the possibility that more than two people might have the same birthday. 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 about .5 that at least two people will have the same birthday.

More exactly,

$$p_r = \frac{(m)_r}{m^r} = \frac{m!}{m^r(m-r)!} \tag{1}$$

where $(m)_r = m(m-1)\cdots(m-r+1)$. Using Stirling's formula [6] [24], it can be shown that, for any constant $c > 0$, if $r = c\sqrt{m}$ then for sufficiently large m

$$p_r \approx e^{-c^2/2}. \tag{2}$$

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 $r = c\sqrt{m}$, then this chance is about e^{-3c^2}.

3.2 Meet-in-the-Middle Closure Test

The meet-in-the-middle closure test is based on lemma 2.1 and the Birthday Paradox: given any endomorphic cryptosystem $\Pi = (K, \mathcal{M}, \mathcal{M}, T)$, pick any key $k \in K$ and search for keys $a, b \in K$ such that $T_k = T_b T_a$. If Π is closed, then such a pair of keys a, b can be efficiently found, on the average. If $\mathcal{T}$ were selected at random, then it is unlikely to find any such pair.

To search for a pair of keys $a, b \in K$ such that $T_k = T_b T_a$, we use a standard "meet-in-the-middle" attack similar to that described in [42], for example. To wit, choose $2r$ keys $a_1, a_2, \ldots, a_r$ and $b_1, b_2, \ldots, b_r$ at random[13] and look for a pair of keys a_i, b_j for some $1 \le i, j \le r$ 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 $p \in \mathcal{M}$, calculate $c = T_k^{-1}(p)$, and compute $x_i = T_{a_i}(p)$ and $y_i = T_{b_i}^{-1}(c)$, for $1 \le i \le r$. Then, look for matches $x_i = y_j$ by sorting the triples $(x_i, a_i, \text{"A"})$ and $(y_j, b_j, \text{"B"})$ for $1 \le i, j \le r$ on their first components. Screen out false matches by testing if $T_k(p_i) = T_{b_j} T_{a_i}(p_i)$, for all $1 \le i \le l$, for a small number of additional messages $p_1, p_2, \ldots, p_l \in \mathcal{M}$. (A false match is a pair of keys $a', b' \in K$ such that $T_k(p) = T_{b'} T_{a'}(p)$ even though $T_k \ne T_{b'} T_{a'}$.)

If Π is closed, this procedure will find a match $T_k = T_b T_a$ with probability $q_r \ge 1 - e^{3r^2/K}$. 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. We are interested in the probability that the samples overlap. If Π is faithful, each element is drawn with probability exactly $1/K$; otherwise, each element is drawn with probability at least $1/K$. If $\mathcal{T}$ was chosen at random, then, for any $T_k \in \mathcal{T}$, we would expect $\mathcal{T}$ to contain a pair $T_a, T_b \in \mathcal{T}$ such that $T_k = T_b T_a$ with probability at most $K^2/M! \cong 0$. By choosing $r = c\sqrt{m}$ with c sufficiently large, we can make the probability $q_r \cong 1 - e^{-3c^2}$ of finding a match as large as desired.

The expected number of false matches is very small, as shown by lemma 2.4. If Π is closed, then at most $(K-1)/|B_p|$ keys other than k map p to c, where B_p is the $\mathcal{T}$-orbit of p. Conversely, if $\mathcal{T}$ was chosen at random, then we would expect at most $(m-1)/M \le 2^{-8}$ keys other than k to map p to c.

This statistical test 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 problem with carrying out this test on DES is the high space requirement, but even today using 2^{28} words of external tape storage is not totally unreasonable. Most steps of this test can be performed in parallel.

3.3 Cycling Closure Test

Given any endomorphic cryptosystem $\Pi = (K, \mathcal{M}, \mathcal{M}, T)$, the cycling test takes a pseudo-random walk in G_Π, the group generated by Π. By the Birthday Paradox, the expected cycle length of such a walk is about $\sqrt{\hat{m}}$, where $\hat{m} = |G_\Pi|$. If Π is closed, then $\hat{m} = m$, where $m = order(\Pi)$. But if $\mathcal{T}_\Pi$ is chosen at random, then with extremely high probability $\mathcal{T}_\Pi = \mathcal{A}_\mathcal{M}$ or $\mathcal{T}_\Pi = \mathcal{S}_\mathcal{M}$ and hence $\hat{m} \ge (M!)/2$.

[13] Actually, it suffices to choose $b_i = a_i$, for $1 \le i \le r$.

The walk $\hat{g}_1, \hat{g}_2, \ldots$ in G_Π is computed from a pseudo-random sequence of transformations $g_0, g_1, \ldots \in \mathcal{T}_\Pi$ by letting $\hat{g}_0 = I$ and $\hat{g}_i = g_i \hat{g}_{i-1}$, for $i \geq 1$. Each g_i is chosen by selecting a key k_i and letting $g_i = T_{k_i}$.

To implement this cycling test efficiently, represent the walk $\hat{g}_1, \hat{g}_2, \ldots$ in G_Π by an induced walk $\hat{x}_1, \hat{x}_2, \ldots$ in $\mathcal{M}^l$, for some l. Specifically, select some message sequence $\hat{x}_0 \in \mathcal{M}^l$ at random and represent each $\hat{g}_i$ by its image $\hat{x}_i = \hat{g}_i(\hat{x}_0)$. To prevent the induced walk in $\mathcal{M}^l$ from cycling before the main walk in G_Π cycles, the integer l must be chosen sufficiently large. For DES, $l = 1$ suffices, since DES has many more messages than keys.

To enable the cycle length of the walk to be computed efficiently and exactly, take a deterministic pseudo-random walk rather than a truly random walk. In particular, for $i = 1, 2, \ldots$, choose the key k_i as a pseudo-random function of $\hat{x}_{i-1}$. For $i = 1, 2, \ldots$, let $\hat{x}_i = T_{k_i}(\hat{x}_0)$, where $k_i = \rho(\hat{x}_{i-1})$ for some deterministic pseudo-random function $\rho : \mathcal{M}^l \to K$. Finally, to detect cycles and to compute the lengths of cycles and their leaders, use the efficient algorithms described by Sedgewick and Szymanski [27] that generalize the well-known "two-finger" algorithm due to Floyd [20].

The validity of the cycling test depends in part on the extent to which the pseudo-random walk behaves like a truly random walk. To increase one's confidence that the pseudo-random 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 several different types of pseudo-random functions.[14] (See section 5.2 and Appendix A for a description of the particular pseudo-random functions used in our experiments.)

In other words, the cycling closure test picks an initial message x_0 at random and computes the ψ_ρ-closure of x_0, where the function $\psi_\rho : \mathcal{M} \to \mathcal{M}$ is defined by $\psi_\rho(x) = T_{\rho(x)}(x)$ whenever $x \in \mathcal{M}$, and $\rho : \mathcal{M} \to K$ is a deterministic pseudo-random function. If ρ is "random," then ψ_ρ acts like a random function on the $\langle \mathcal{T} \rangle$-orbit of x_0. The expected length of the ψ_ρ-closure computed by the test is about the square root of the length of the $\langle \mathcal{T} \rangle$-orbit of x_0. If DES acts like a set of randomly chosen permutations, then we would expect $\langle \mathcal{T} \rangle\text{-orbit}(x_0) = \mathcal{M}$, in which case we would expect $|\psi_\rho\text{-closure}(x_0)| \cong \sqrt{M} = 2^{32}$. However, if DES were closed, then $|\langle \mathcal{T} \rangle\text{-orbit}(x_0)| \leq K$, in which case we would expect $|\psi_\rho\text{-closure}(x_0)| \leq \sqrt{K} = 2^{28}$.

The second test is similar in spirit to Pollard's ρ-factoring method [22] [18]. It is also similar to but different from the algorithm discovered by Sattler and Schnorr for determining the order of any element in any finite group that has an efficient multiplication procedure [25]. The cycling test differs from the cycling experiments performed by Gait [36] and Hellman and Reyneri [39], who held either the key or message fixed (see section 1.3).

If $\mathcal{T}_\Pi$ is chosen at random, then the walk in G_Π induces a pseudo-random walk in $\mathcal{M}^l$. If $r = cM^{l/2}$ for some constant $c > 0$, then the chance that the induced walk in $\mathcal{M}^l$ cycles within r steps is only about $e^{-c^2/2}$.

For the case that Π is closed, it helpful to model the pseudo-random walk $\hat{g}_1, \hat{g}_2, \ldots$ in G_Π as a discrete finite Markov Process with a $K \times K$ transition matrix A. For each $1 \leq i, j \leq K$, the (i,j)th entry a_{ij} of A denotes the probability of selecting $\hat{g}_i$ next, given that $\hat{g}_j$ was the last selected transformation. Each pseudo-random selection depends only on the immediately preceding state. If Π is faithful, then each entry of A is exactly $1/K$; otherwise, each entry of A is at least $1/K$. In either case, the probability of a pseudo-random walk not cycling within r steps is at most $(K)_r / K^r$.

The second test computes a statistic $w = \lambda + \mu$, where λ and μ are respectively the leader length and cycle length of a particular pseudo-random walk in $\mathcal{M}^l$, starting at some randomly selected point $\hat{x}_0$. The value of this statistic depends on the size of the G_Π-orbit of x_0. If Π is closed, then by lemma 2.2 this orbit contains at most K messages. However, if $\mathcal{T}_\Pi$ is chosen at random, then with very high confidence the G_Π-orbit of x_0 is $\mathcal{M}^l$. Therefore, if Π is closed, the expected value of w is at most approximately $\sqrt{K}$; but, if $\mathcal{T}_\Pi$ is chosen at random, then the expected value of w is approximately $M^{l/2}$. For DES with $l = 1$, the expected value of w is about 2^{28} if DES is closed and about 2^{32} if $\mathcal{T}_{DES}$ is chosen at random.

It is possible for the random walk to cycle prematurely if certain special keys are chosen during the walk. For example, the cycle will close if a pair of semi-weak keys are chosen one after the other, or, if

[14] For example, the pseudo-random 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.

the identity permutation is selected. Such events would be interesting, but are unlikely to happen. In any case, such events would not contradict any of our analysis, since short cycles are evidence that $\mathcal{T}$ is not a random set of permutations.

This test requires $O(w)$ time and a constant amount of space, where w is the statistic computed by the test. The cycle detection and cycle length computations use a small constant amount of space and require about w encryptions [27].

By picking any $T_0 \in \mathcal{T}$ and by applying the test to $T_0^{-1}\mathcal{T}$, the cycling test can be used to test for purity as well.

4 Known-Plaintext Attacks against Closed 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 the same key k. This attack does not find k.

4.1 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.

4.2 Cycling Known-Plaintext Attack

The cycling test also yields a known-plaintext attack. Given a matched plaintext/ciphertext pair (p, c) that was encrypted under some secret key k, the cryptanalyst computes two pseudo-random walks of the type used in the cycling test, starting from messages p and c. The same pseudo-random 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 pseudo-random walks will intersect within about $\sqrt{K}$ steps. Since the same deterministic pseudo-random 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 [27] cycle-detection algorithm for each of the pseudo-random walks, and by sharing the same memory for both algorithms, it is easy to find a specific point at which the walks intersect, provided the walks intersect. The two walks can be computed sequentially or simultaneously.

Thus, the cycling test gives a way to generate 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.

To decrypt each additional ciphertext c_0, the cryptanalyst computes $T_k^{-1}(c_0) = g^{-1}h(c_0)$. To compute h in constant space is easy—simply generate the sequence of keys $b_1, b_2, \ldots, b_j$ by retracing the pseudo-random walk starting from c. The difficulty is to compute g^{-1} in a time- and space-efficient manner. The problem is that each pseudo-random 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 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 $\epsilon > 0$, it is possible to compute g^{-1} in constant space and time $i^{1+\epsilon}$ [19]. Therefore, if the attacked cryptosystem is closed, then, for any $\epsilon > 0$, the cycling known-plaintext attack can be carried out in constant space and time $O\left(K^{(1+\epsilon)/2}\right)$, on the average.

4.3 Application of Attacks to DES

Each of the known-plaintext attacks can be applied to any finite, deterministic cryptosystem by launching the attack against the group generated by the cryptosystem. For this reason, it is very important to know the order of the group generated by DES.

Since DES's relatively small key space of 2^{56} keys allows no margin of safety even for 1977 technology [35], these attacks would be a devastating weakness for DES, if DES generated a small group. In particular, if DES were closed, a personal computer equipped with special-purpose hardware could decrypt DES ciphertexts under a known-plaintext attack in less than two hours, on the average (See appendix A).

5 Experimental Results

This section explains how to interpret the results of the statistical closure tests and summarizes the initial results we obtained by applying the cycling test to DES.

5.1 Interpreting the Experimental Results

Each statistical test gives a method for collecting evidence that can be used to compute a measure of our relative degree of belief in the following two competing hypotheses:

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

To compute this measure, we will apply the *theory of the weight of evidence*, as explained by Good [9] [7].

Each test is asymmetrical in the sense that it allows us to compute the conditional probabilities $P(E \mid H_G)$ and $P(E \mid H_R)$, but not $P(E \mid \overline{H_G})$ nor $P(E \mid \overline{H_R})$, where E is experimental evidence and $\overline{H_G}$ and $\overline{H_R}$ are the complements of H_G and H_R respectively. This means that, on the basis of experimental evidence, we would be able to conclude only that DES is *not* closed or that DES has a structure different from that expected from a set of randomly chosen permutations; we would not be able to conclude that DES is closed. In the worst case, DES could be closed, except for some isolated pair of keys a, b such that $T_b T_a$ is not in $\mathcal{T}$, 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 $x \in \mathcal{M}$, $x \neq x_0$.

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, each believer in the theory of the weight of evidence should update his or her odds in favor of H_G over H_R as follows:

$$O(H_G/H_R \mid E) \longleftarrow \frac{P(E \mid H_G)}{P(E \mid H_R)} O(H_G/H_R). \tag{3}$$

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

In light of the our experimental evidence, we encourage each reader to update his or her own odds in favor of H_G over H_R.

5.2 Summary of Experimental Results

On April 4, 1985, we completed the first trial of the cycling test, detecting a cycle of length nearly 2^{33}. For this test, we chose the pseudo-random function to be the "identity" projection.[15] Starting with the initial message x_0 = 0123 4567 89AB CDEF (in hexadecimal notation), we found a cycle of length exactly $\mu = 7,985,051,916$ with a leader of length $\lambda = 34,293,589$. As one test of the correctness of our computations, we ran a software implementation of the cycling test for 30,000 steps. The software and hardware implementations of the cycling test agreed on all values. As a second test of correctness, we repeated the initial experiment and obtained identical results.

This single experiment gives strong evidence that DES is not closed. Let E denote the evidence from our experiment. Since $\mu + \lambda \approx 2^{33} = 2\sqrt{M} = 32\sqrt{K}$, it follows that $P(E \mid H_G)/P(E \mid H_R) \approx e^{-32^2/2}/e^{-2^2/2} = e^{-510}$. Therefore, each reader should decrease his or her odds in favor of H_G over H_R by a factor of about e^{-510}.

During May through August 1985, we performed additional trials of the cycling closure test as well as other cycling experiments on DES. Results of these experiments were described at the Crypto 85 conference [41]. All additional trials of the cycling closure test supported our initial findings.

6 Open Problems

Although our experiments give strong statistical evidence that DES is not closed, numerous interesting questions remain unanswered. We begin with several questions about the algebraic structure of DES.

- Does DES generate $A_{\mathcal{M}}$? 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? What is the order of DES?
- 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 $k \in K$ it is true that $T_k^{-1} \in \mathcal{T}$? For how many $k \in K$ is it true that $T_k^{-1} \in \mathcal{T}$?
- Is $I \in \mathcal{T}$?

Knowing whether or not $I \in \mathcal{T}_{DES}$ is interesting—not because this property would necessarily be a weakness in DES—but because this question would answer several other questions about DES. By the complementation property, for any key k, $T_k = I$ implies $T_{\overline{k}}$=I. Hence, if $I \in \mathcal{T}_{DES}$, then DES is not faithful. In particular, if DES is closed, then DES is not faithful. Conversely, if $I \notin \mathcal{T}_{DES}$, 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 $T_1, T_2 \in \mathcal{T}$ are *q-approximately equal on* $X \subseteq \mathcal{M}$ iff, 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 the closure tests do not depend on the detailed definition of DES, it is natural to ask:

[15] More specifically, we used the projection that removes each of the eight parity bits.

- 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.

- Is it possible to build a secure, practical cryptosystem for which it can be proven that the cryptosystem generates either A_M or S_M? (See [48] 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-plaintext 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?

7 Summary

We have presented two statistical tests for determining whether or not any finite, deterministic cryptosystem generates a small group. Each test yields a known-plaintext attack against closed cryptosystems.

Using a combination of software and special-purpose hardware, we applied the cycling test to DES. Our experiments show, with a high degree of confidence, that DES does not generate a small group. These results should increase our confidence in the security of using DES with multiple encryption. However, since cryptosystems that generate large groups are not necessarily secure, our experiments say only that DES does not fail in one extreme way.

This work leaves open the possibility of proving that DES is not closed directly from the detailed definition of DES.

8 Acknowledgments

We would like to thank four people who contributed to this paper. Leon Roisenberg helped out with the design and construction of our special-purpose hardware. John Hinsdale wrote the C software used by our host IBM personal computer to control our special-purpose hardware and to carry out the cycle-detection algorithm. Gary Miller answered several of our questions about permutation group theory, and Oded Goldreich participated in a conversation that led to the meet-in-the-middle closure test. Finally, we would like to thank the Functional Languages and Architectures (FLA) research group of the MIT Laboratory for Computer Science (LCS) for use of their new state-of-the-art hardware laboratory during the construction and testing of our special-purpose hardware.

A A Fast Implementation of the Cycling Closure Test

To test the DES for closure, we designed and built special-purpose hardware for an IBM PC. Our experiment required special-purpose hardware for two reasons: we needed to compute about 2^{32} encryptions[16] and we needed to change the key at each step.[17]

The special-purpose hardware is a custom wire-wrap board for an IBM personal computer,[18] containing a microprogrammed finite-state controller and an AMD AmZ8068 DES chip [52]. Data paths connect the DES chip, a 16-byte ciphertext buffer, a PROM computing the next-key function, and the host computer (see figure 1). The next-key function is computed byte-by-byte. A read-write counter indicates the number of consecutive messages to compute. To increase the board's flexibility, the microprogram is stored in RAM accessible to the host computer. The PROM can be easily replaced to implement different next-key functions.

We perform cycle detection in two passes: data acquisition and analysis. During data acquisition, the host computer stores every 2^{20}th message on a floppy disk. During analysis, these messages are loaded into main memory, and up to 2^{20} consecutive messages are computed and compared to those already present. In effect, we perform the Sedgewick-Szymanski [26] algorithm with a fixed estimate of the cycle length. We use an open-addressing, double-hashing scheme for stores and lookups [21]. We wrote all data acquisition and analysis routines in C.

Including all overhead for computing and loading a new key for each encryption, our board performs about 45K encryptions/second, or almost 2^{32} per day. This enables us to carry out each trial of the experiment within a few days. Our board also supports all approved modes of operation for DES.

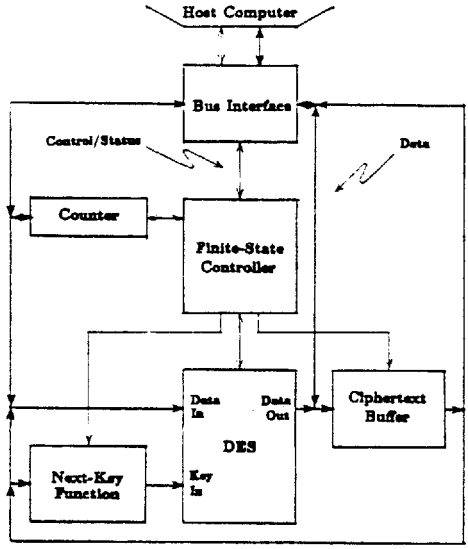

Figure 1: Block diagram of special-purpose hardware

[16] Software implementations of the DES for the IBM PC run at about 200-300 encryptions/second. According to Davio, by using an efficient space-intensive implementation of the DES, it is possible to perform about 2.5K encryptions/second on the VAX 11/780 [33]. Thus, it would take the IBM PC about 10 to 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.

[17] 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.

[18] 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 [54].

References

[] Survey Works on Cryptology

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

[2] Davies, Donald W.; and W. L. Price, *Security for Computer Networks: An Introduction to Data Security in Teleprocessing and Electronic Funds Transfer*, John Wiley (Chichester, England, 1984).

[3] Diffie, Whitfield; and Martin E. Hellman, "Privacy and authentication: An introduction to cryptography," *Proceedings of the IEEE*, 67 (March 1979), 397–427.

[4] Meyer, Carl H.; and Stephen M. Matyas, *Cryptology: A New Dimension in Computer Data Security*, John Wiley (New York, 1982).

See also [50] [55].

Works on Probability and Statistics

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

[6] Feller, W., *An Introduction to Probability Theory and its Applications*, vol. I, John Wiley (New York, 1971).

[7] Good, Irving John, *The Estimation of Probabilities: An Essay on Modern Bayesian Methods*, MIT Press (1965).

[8] Harris, Bernard, "Probability distributions related to random mappings," *Annals of Math. Statistics*, 31 (1959), 1045–1062.

[9] Osteyee, David Bridston; and Irving John Good, *Information, Weight of Evidence, the Singularity between Probability Measures and Signal Detection*, Springer (Berlin, 1974).

[10] Purdom, Paul W.; and J. H. Williams, "Cycle length in a random function," *Transactions of the American Mathematics Society*, 133 (1968), 547–551.

[11] Shepp, L. A.; and S. P. Lloyd, "Ordered cycle lengths in a random permutation," *Transactions of the American Mathematics Society*, (February 1966), 340–357.

See also [12] [14] [25].

Works on Algebra

[12] Bovey, John; and Alan Williamson, "The probability of generating the symmetric group," *Bull. London Math Society*, 10 (1978), 91–96.

[13] Carmichael, Robert D., *Introduction to the Theory of Groups of Finite Order*, Dover (New York, 1956).

[14] Dixon, John D., "The probability of generating the symmetric group," *Math Zentrum*, 110 (1969), 199–205.

[15] Rotman, Joseph J., *The Theory of Groups: An Introduction*, Allyn and Bacon (Boston, 1978).

[16] Wielandt, Helmut, *Finite Permutation Groups*, Academic Press (New York, 1964).

See also [5] [8] [10] [25] [11].

Works on Algorithms and Complexity Theory

[17] Allender, Eric; and Maria Klawa, "Improved Lower Bounds for the Cycle Detection Problem," working paper.

[18] Brent, Richard P., "Analysis of some new cycle-finding and factorization algorithms," technical report, Department of Computer Science, Australian National University (1979).

[19] Chandra, Ashok K., "Efficient compilation of linear recursive programs," technical report no. STAN–CS–72–282, Computer Science Dept., Stanford Univ (April 1972).

[20] Knuth, Donald E., *Seminumerical Algorithms* in *The Art of Computer Programming*, vol. 2, Addison-Wesley (1969).

[21] Knuth, Donald E., *Sorting and Searching* in *The Art of Computer Programming*, vol. 3, Addison-Wesley (1973).

[22] Pollard, J. M., "A Monte Carlo method for factorization," *Bit*, 15 (1975), 331–334.

[23] Pomerance, Carl, "Analysis and comparison of some integer factoring algorithms," technical report, Math Dept., Univ. of Georgia.

[24] Purdom, Paul W. Jr.; and Cynthia A. Brown, *The Analysis of Algorithms*, Holt, Rinehart, and Winston (New York, 1985).

[25] Sattler, J.; and C. P. Schnorr, "Generating random walks in groups," unpublished manuscript (October 1983).

[26] Sedgewick, Robert; and Thomas G. Szymanski, "The complexity of finding periods," *Proceedings of the 11th Annual STOC Conference* (1979), 74–80.

[27] Sedgewick, Robert; Thomas G. Szymanski; and Andrew C. Yao, "The complexity of finding cycles in periodic functions," *Siam Journal on Computing*, 11 (1982), 376–390.

Selected Federal Standards Involving DES

[28] "Data Encryption Standard," National Bureau of Standards, Federal Information Processing Standards Publications No. 46 (January 15, 1977).

[29] "DES modes of operations," Federal Information Standards Publication No. 81 (December 1980).

Selected Technical Works on DES

[30] Davies, Donald W., "Some regular properties of the DES," in [46], 89–96.

[31] Davies, Donald W.; and G. I. P. Parkin, "The average size of the key stream in output feedback mode," in [46], 97–98.

[32] Davies, Donald W.; and G. I. P. Parkin, "The average size of the key stream in output feedback encipherment," in [45], 263–279.

[33] Davio, Mark; Yvo Desmedt; Josef Goubert; Frank Hoornaert; and Jean-Jacques Quisquater, "Efficient hardware and software implementations for the DES," *Proceedings of Crypto 84*, Springer (1985).

[34] Desmedt, Yvo, "Analysis of the security and new algorithms for modern industrial cryptography," dissertation, Department Elektrotechniek, Katholieke Universiteit Leuven (October 1984).

[35] Diffie, Whitfield; and Martin E. Hellman, "Exhaustive cryptanalysis of the NBS Data Encryption Standard," *Computer*, 10 (March 6, 1980), 74–84.

[36] Gait, Jason, "A new nonlinear pseudorandom number generator," *IEEE Transactions on Software Engineering*, SE-3 (September 1977), 359–363.

[37] Goldreich, Oded, "DES-like functions can generate the alternating group," *IEEE Transactions on Information Theory*, IT-29 (1983), 863–865.

[38] Hellman, Martin E., *et al.*, "Results of an initial attempt to cryptanalyze the NBS Data Encryption Standard," technical report SEL 76–042, Information Systems Laboratory, Stanford Univ. (November 1976).

[39] Hellman, Martin E.; and Justin M. Reyneri, "Distribution of Drainage in the DES," in [46] (1982), 129–131.

[40] Jueneman, Robert R., "Analysis of certain aspects of output-feedback mode," in [46] (1982), 99–127.

[41] Kaliski, Burton S., Jr.; Ronald L. Rivest; and Alan T. Sherman, "Is DES a pure cipher? (Results of more cycling experiments on DES)," *Proceedings of Crypto 85*, to appear.

[42] Merkle, Ralph C.; and Martin E. Hellman, "On the security of multiple encryption," *CACM*, 24 (July 1981), 465–467.

[43] Reeds, J. A.; and J. L. Manferdell, "DES has no per round linear factors," *Proceedings of Crypto 84*, Springer (1985).

[44] Tuchman, W. L., talk presented at the National Computer Conference, (June 1978).

See also [2] [4] [48] [51] [53].

Other Works

[45] Beth, Thomas, *ed.*, *Cryptography, Proceedings of the Workshop on Cryptography, Burg Feuerstein, Germany, March 29–April 2, 1982*, Springer (Berlin, 1983).

[46] Chaum, David; Ronald L. Rivest; and Alan T. Sherman, *eds.*, *Advances in Cryptology: Proceedings of Crypto 82*, Plenum Press (New York, 1983).

[47] Chaum, David, *ed.*, *Advances in Cryptology: Proceedings of Crypto 83*, Plenum Press (New York, 1984).

[48] Coppersmith, Don; and Edna Grossman, "Generators for certain alternating groups with applications to cryptology," *Siam Journal on Applied Mathematics*, 29 (December 1975), 624–627.

[49] DeLaurentis, John M., "A further weakness in the common modulus protocol for the RSA cryptosystem," *Cryptologia*, 8 (July 1984), 253–259.

[50] Gaines, Helen Fouché, *Cryptanalysis: A Study of Ciphers and Their Solution*, Dover (1956).

[51] Grossman, Edna; and Bryant Tuckerman, "Analysis of a Feistel-like cipher weakened by having no rotating key," IBM research report RC 6375 (#27489), (January 31, 1977).

[52] *Data Ciphering Processors Am9518, Am9568, AmZ8068 Technical Manual*, Advanced Micro Devices, Inc. (1984).

[53] Hellman, Martin E., "A cryptanalytic time-memory tradeoff," technical report, Stanford Univ. (1978).

[54] *IBM Personal Computer Technical Reference* (July 1982).

[55] Longo, G., *ed.*, *Secure Digital Communications*, Springer (Vienna 1983).

[56] Rivest, Ronald; Adi Shamir; and Leonard Adleman, "On digital signatures and public-key cryptosystems," *CACM*, 21 (February 1978), 120–126.

[57] Shannon, Claude E., "Communication theory of secrecy systems," *Bell System Technical Journal*, 28 (October 1949), 656–715.

[58] "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).