Complexity Theoretic Issues Concerning Block Ciphers Related to D.E.S.

*Richard Cleve**
Department of Computer Science
University of Calgary
Calgary, Canada T2N 1N4

Abstract

The D.E.S. cipher is naturally viewed as a composition of sixteen invertible transformations on 64-bit strings (where the transformations depend of the value of a 56-bit key). Each of the transformations has a special form and satisfies the particular property that each of its output bits is determined by a "small" number of its input bits. We investigate the computational power of block ciphers on n-bit strings that can be expressed as polynomial-length (with respect to n) compositions of invertible transformations that have a form similar to those of D.E.S. In particular, we require that the basic transformations have the property that each of their output bits depends on the value of a small number of their input bits (where "small" is somewhere in the range between $O(1)$ and $O(\log n)$). We present some sufficient conditions for ciphers of this type to be "pseudorandom function generators" and, thus, to yield private key cryptosystems that are secure against adaptive chosen plaintext attacks.

1 Introduction

The Data Encryption Standard (D.E.S.) was developed at IBM in the seventies to be used as a private key cryptosystem (i.e. a system that enables two parties, who share

*Research partially conducted while the author was at the University of Toronto, partially supported by an NSERC postgraduate scholarship, and at the International Computer Science Institute in Berkeley, CA.

A.J. Menezes and S.A. Vanstone (Eds.): Advances in Cryptology - CRYPTO '90, LNCS 537, pp. 530-544, 1991.

quantity is one that is bounded below by a quantity that is larger than the inverse of a polynomially bounded quantity (i.e. $(\frac{1}{n})^{O(1)}$). A cipher (or "permutation generator") is any function that maps an $m(n)$-bit string (called the key) and an n-bit string (called the plaintext) to an n-bit string (called the ciphertext). It is reasonable to require that a cipher be feasibly computable (i.e. computable in time polynomial with respect to n). A cipher that passes the black box test is called a "pseudorandom permutation generator" and, when used as a private key cryptosystem, is secure against adaptive chosen plaintext attacks [7].

It is not clear how to naturally scale up D.E.S. for arbitrarily large block sizes; however, certain of its structural features do scale up naturally. The D.E.S. function is a composition of functions of the following forms. Let $F : \{0,1\}^n \to \{0,1\}^n$. Define the mapping $T_F : \{0,1\}^{2n} \to \{0,1\}^{2n}$ as $T_F(x,y) = (x, y \oplus F(x))$ and define the mapping $S : \{0,1\}^{2n} \to \{0,1\}^{2n}$ as $S(x,y) = (y,x)$. Let $\circ$ denote the composition operator. The D.E.S. cipher, for each value of its key, is of the form

$$T_{F_1} \circ S \circ T_{F_2} \circ S \circ \cdots \circ S \circ T_{F_{16}} ,$$

where n is set to 64, and where $F_1, ..., F_{16}$ also depend on the value of the key. Moreover, the functions F_i ($i \in \{1, ..., 16\}$) all satisfy the following restrictive property: each output bit of F_i depends on at most six of its input bits. These structural features scale up naturally with larger values of n to compositions of the form

$$T_{F_1} \circ S \circ T_{F_2} \circ S \circ \cdots \circ S \circ T_{F_{r(n)}} ,$$

where some bound may be placed on $r(n)$, the length of the composition, and a bound may be placed on the number of input bits that each output bit of F_i depends on.

We investigate the class of permutations that can be expressed as

$$T_{F_1} \circ S \circ T_{F_2} \circ S \circ \cdots \circ S \circ T_{F_{r(n)}} ,$$

where $r(n)$ is polynomial in n, and, for each $i \in \{1, ..., r(n)\}$, each output bit of F_i depends on a number of input bits that is bounded somewhere from $O(1)$ to $O(\log n)$. (For technical reasons, we consider it a realistic reflection of the design of D.E.S. to disallow F_i's whose output bits depend on more than $O(\log n)$ input bits. This is because, in the finer structure of the D.E.S. cipher, there are functions, called "S-boxes," that are expressed in tabular form. In asymptotic versions of D.E.S., in order for these tables to be of polynomial size, the number of inputs to the S-boxes must be logarithmically bounded. We do not elaborate further on this aspect of D.E.S. in this report.)

Our first observation is that, by a construction of Coppersmith and Grossman [4], and Even and Goldreich [6], the above class of permutations is robust in that the class obtained is the same when the bound on the number of input bits that the output bits depend on is $O(1)$ or $O(\log n)$. Also, applying a result of Luby and Rackoff [7], if using our basic permutations, we could feasibly simulate permutations of the form T_F, where F is an arbitrary polynomial-time computable function (that also depends on a key) then there would be a pseudorandom permutation generator in the above class (assuming a one-way function exists). Our main result shows how to simulate, in terms of our basic permutations, permutations of the form T_F where $F \in NC^1$, and we then extend this to permutations of the form T_F where F is computed in nonuniform logarithmic space. As a consequence, if there exists a pseudorandom function generator in nonuniform logarithmic space then there exists a "secure" cipher that is in our class, and therefore one that has a form similar to D.E.S.

2 Overview of Related Work

Coppersmith and Grossman [4] investigate certain special permutations on sets of strings that are different from—but related to—those permutations described above. They show that, by composing sufficiently many of their special permutations, any permutation of even parity can be constructed. Even and Goldreich [6] make the connection between the work of Coppersmith and Grossman, and D.E.S. more explicit. The resulting theorem is that any of the $\frac{1}{2}2^{2n}!$ permutations of even parity on $\{0,1\}^{2n}$ can be expressed as

$$T_{F_1} \circ S \circ T_{F_2} \circ S \circ \cdots \circ S \circ T_{F_{r(n)}} ,$$

where $F_1, ..., F_{r(n)}$ have the property that each of their output bits depends on at most a constant number (in this case two is sufficient) of their input bits, and $r(n)$ is *exponentially* (in n) bounded. This implies that, allowing exponentially long compositions of the basic permutations, and allowing the functions $F_1, ..., F_{r(n)}$ to also depend on a key (which in this case would also have to be exponentially long), ciphers that pass the black box test can be constructed. This result is of some support to the security of D.E.S. in that if all compositions of the above form were in a sufficiently restricted class then this would expose an insecurity of D.E.S. For example, if we consider compositions of the above form where the functions $F_1, ..., F_{r(n)}$ have the property that each of their output bits depends on at most *one* input bit then the resulting permutations are all affine linear [4,6], and such transformations are easily defeated by the black box test. On the other hand, exponentially long compositions of permutations seem too long to realistically reflect the design of D.E.S. We consider it more realistic to investigate the permutations that can be generated by compositions of the above form where $r(n)$ is polynomial in n.

Luby and Rackoff [7] show that much shorter compositions of permutations yield ciphers that are in some sense pseudorandom, provided that one allows the basic permutations to be powerful enough. What they show is that permutations of the form

$$T_{F_1} \circ S \circ T_{F_2} \circ S \circ T_{F_3}$$

are pseudorandom, provided that F_1, F_2, F_3 are independently generated pseudorandom functions. Using other results and assuming that a one-way function exists, there exist functions F_1, F_2, F_3 (that also depend on keys) that are of polynomial time complexity and are pseudorandom. This is an interesting result because, prior to this work, there were no known constructions of pseudorandom permutation generators in terms of pseudorandom function generators. This result does not, however, explain a mechanism that works in D.E.S. In D.E.S., the functions $F_1, ..., F_{16}$ are definitely *not* pseudorandom. Any function that, like $F_1, ..., F_{16}$, has the property that each of its output bits does not depend on all of its input bits can be very easily distinguished from a random function. In fact, in the case of D.E.S., the individual functions F_i can each be broken in a much stronger sense: they can be *completely determined* in a simple manner by evaluating them at only 64 different inputs. Also, Biham and Shamir [8] show that if one varies D.E.S. by reducing the number of rounds then it quickly becomes insecure as the number of rounds decreases from 16. Thus, it appears necessary to use all 16 rounds in D.E.S.

The design principle in D.E.S. seems to be to employ simpler functions than those considered in [7] (more along the lines of those considered in [4,6]) and allow the length of the composition to be more than constant (yet shorter than the exponential lengths considered [4,6]).

3 Results

The main results are Theorem 5 and Theorem 6.

3.1 Function Generators

Definition 1: A *function generator* G is a function of the from $G = \bigcup_{n=1}^{\infty} G^n$, where, for each n, $G^n : \{0,1\}^{m(n)} \times \{0,1\}^n \to \{0,1\}^n$. We call the first input to G^n (the

$m(n)$-bit string) the *key*. Each value of the key determines a function from $\{0,1\}^n$ to $\{0,1\}^n$. When convenient, we will sometimes interchange the notation G and G^n when no ambiguity results.

We are interested in classes of function generators that have certain properties, such as having polynomial (in n) time complexity.

Definition 2: P is the class of function generators computed in polynomial time. More formally, a function generator G^n is in P (for polynomial-time) if, for each n, each output bit of G^n is computed by a Boolean circuit of size polynomial in n.

We are also interested in other classes of function generators. Each of the three classes defined below have the following properties. Each of the function generators G^n are expressible as $G^n(y,x) = F^n(H^n(y),x)$, where $H^n : \{0,1\}^{m(n)} \rightarrow \{0,1\}^{p(n)}$ and $F^n : \{0,1\}^{p(n)} \times \{0,1\}^n \rightarrow \{0,1\}^n$, and H is polynomial-time computable, and F satisfies a condition that depends on which of the three specific classes that G is in. H is called the *key preprocessing* phase of the computation. For convenience, we may regard F^n as a function from $\{0,1\}^{p(n)+n}$ to $\{0,1\}^n$.

Definition 3: *FAN-IN*$[k(n)]$ denotes the class of all function generators that, after polynomial-time preprocessing of key bits, are computed by a function with *fan-in* bounded above by $k(n)$, where the *fan-in* of a function is, informally, the maximum number of input bits that any output bit depends on. More formally, a function generator G^n has *fan-in* $k(n)$ if it is expressible as $G(y,x) = F(H(y),x)$, where $H : \{0,1\}^{m(n)} \rightarrow \{0,1\}^{p(n)}$ is polynomial-time computable, and $F : \{0,1\}^{p(n)+n} \rightarrow \{0,1\}^n$ has the following property. Each output bit of $F(z_1,...,z_{p(n)+n})$ is determined by the values of at most $k(n)$ of the values $z_1,...,z_{p(n)+n}$. That is, for all $i \in \{1,...,n\}$, there exists $j_1,...,j_{k(n)} \in \{1,...,n\}$ and $f : \{0,1\}^{k(n)} \rightarrow \{0,1\}$ such that, for all $z_1,...,z_{p(n)+n} \in \{0,1\}$, $[F(z_1,...,z_{p(n)+n})]_i = f(z_{j_1},...,z_{j_{k(n)}})$.

Definition 4: NC^1 is the class of function generators that, after polynomial-time preprocessing of key bits, are computed by logarithmic-depth Boolean circuits. More formally, a function generator G^n is in NC^1 if it is expressible as $G(y,x) = F(H(y),x)$, where $H : \{0,1\}^{m(n)} \rightarrow \{0,1\}^{p(n)}$ is polynomial-time computable, and $F : \{0,1\}^{p(n)+n} \rightarrow \{0,1\}^n$ has the following property. Each output bit of $F(z_1,...,z_{p(n)+n})$ is computed by a Boolean circuit of depth $O(\log n)$.

Definition 5: *SPACE*$[w(n)]$ is the class of all function generators that, after polynomial-time preprocessing of key bits, are computed by $w(n)$-space computations (which are defined formally below). Informally, when $w(n)$-space computations are analogous to non-uniform Turing machine computations that use $w(n)$-space and run in polynomial-time. Formally, a function generator G^n is in $SPACE[w(n)]$ if it is expressible as $G(y,x) = F(H(y),x)$, where $H : \{0,1\}^{m(n)} \to \{0,1\}^{p(n)}$ is polynomial-time computable, and $F : \{0,1\}^{p(n)+n} \to \{0,1\}^n$ has the following property. The computation of F on the inputs $(z_1, ..., z_{p(n)+n}) = (H(y), x)$ is defined in terms of a sequence

$$\Pi_1, ..., \Pi_{t(n)} : \{0,1\} \times \{0,1\}^{w(n)} \to \{0,1\}^{w(n)}$$

of *transition functions*, and a sequence

$$\alpha_1, ..., \alpha_{t(n)} : \{0,1\}^{w(n)} \to \{1, ..., p(n)+n\}$$

of *address functions*, where $t(n) \in n^{O(1)}$. The resulting computation is then a sequence $s_0, ..., s_{t(n)} \in \{0,1\}^{w(n)}$ of *configurations*, where the initial configuration s_0 is 0...0, and, for $i \in \{1, ..., t(n)\}$,

$$s_i = \Pi_i(z_{\alpha_i(s_{i-1})}, s_{i-1}) \ .$$

This latter part means that, from each configuration, the following configuration is determined by the current configuration and the value of one input (which may depend on the current configuration). The *output* of the computation is 1 if the final configuration $s_{t(n)}$ is, say, 0...0, and 0 otherwise.

We consider some relationships between the above classes of function generators. Clearly, $FAN\text{-}IN[O(1)] \subset FAN\text{-}IN[O(\log n)]$ and this containment is *proper*. Also, the following theorem is elementary to prove.

Theorem 1: $FAN\text{-}IN[O(\log n)] \subset NC^1$ and the containment is proper.

From Barrington's work [1], we can obtain the following.

Theorem 2 (Barrington [1]): $SPACE[4] = SPACE[O(1)] = NC^1$.

Finally, it is clear that $NC^1 \subset SPACE[O(\log n)] \subset P$ and it is not known whether any of these containments are proper (though they are widely believed to be so).

3.2 Permutation Generators

Definition 6: A *permutation generator* is a function of the form $A = \bigcup_{n=1}^{\infty} A^n$, where $A^n : \{0,1\}^{m(n)} \times \{0,1\}^{2n} \to \{0,1\}^{2n}$, and, for each $z \in \{0,1\}^{m(n)}$ (called a *key*), $A^n(z) : \{0,1\}^{2n} \to \{0,1\}^{2n}$ is a permutation (i.e. an invertible mapping). We call the first input G^n (the $m(n)$-bit string) the *key*, and the second input (the $2n$-bit string) the *plaintext*. Also, we call the $2n$-bit output string the *ciphertext*. (For technical reasons, that will soon become apparent, the sizes of the plaintext and ciphertext are $2n$ rather than n.) For convenience, we will sometimes interchange the notation A and A^n when no ambiguity results. The *composition* of two permutation generators A and B is $A \circ B : \{0,1\}^{m(n)} \times \{0,1\}^{2n} \rightarrow \{0,1\}^{2n}$, where the permutations are taken relative to the same key (i.e. $(A \circ B)(z) = A(z) \circ B(z)$, for each $z \in \{0,1\}^{m(n)}$).

We are interested in permutation generators that have a structure similar to that of D.E.S. To this end, we define the following.

Definition 7: For any function generator $G : \{0,1\}^{m(n)} \times \{0,1\}^n \rightarrow \{0,1\}^n$, define the associated permutation generator $T_G : \{0,1\}^{m(n)} \times \{0,1\}^{2n} \rightarrow \{0,1\}^{2n}$ as

$$T_G(z)(x,y) = (x, y \oplus G(z,x))$$

for all $z \in \{0,1\}^{m(n)}$ and $x, y \in \{0,1\}^n$ (where the $\oplus$ is taken bitwise). Note that T_G is clearly a permutation generator since $T_G \circ T_G$ is the identity permutation for each value of the key. Also, define the permutation $S : \{0,1\}^{2n} \rightarrow \{0,1\}^{2n}$ as

$$S(x,y) = (y,x)$$

for all $x, y \in \{0,1\}^n$.

Definition 8: For any class of function generators B (e.g. *FAN-IN*$[k(n)]$, NC^1, *SPACE*$[w(n)]$, or P), define $DES[B]$ as the class of permutation generators of the form

$$T_{G_1} \circ S \circ T_{G_2} \circ S \circ \cdots \circ S \circ T_{G_{r(n)}} ,$$

where $r(n)$ is polynomial in n and $G_1, G_2, ..., G_{r(n)}$ are all in B.

It should be noted that the D.E.S. cipher is expressible in the form of the above definition with $n = 64$, $m(n) = 56$, $r(n) = 16$, and $B =$ *FAN-IN*$[12]$ (6 key bits and 6 plaintext bits contribute to the fan-in).

We also note that the form of the permutations in Definition 8 need not strictly alternate between T_G permutations and S permutations, since T_ϕ and $S \circ T_\phi \circ S$ are both the identity permutation if ϕ is the zero function generator (i.e. always zero).

We consider any class that contains a realistic asymptotic extension of D.E.S. to be in *DES*[*FAN-IN*$[k(n)]$], where $k(n)$ is somewhere between $O(1)$ and $O(\log n)$.

Luby and Rackoff [7] show the following (we do not formally define pseudorandom permutation generators or one-way functions; the reader is referred to [7] for more details about this).

Theorem 3 (Luby and Rackoff [7]): *If there exists a one-way function then there exists a pseudorandom permutation generator in* $DES[P]$.

As noted, we consider $DES[P]$ to be too powerful to reflect the design principles of D.E.S. In this report, we are primarily interested in $DES[FAN\text{-}IN[O(1)]]$ and $DES[FAN\text{-}IN[O(\log n)]]$. If we could show that one of these classes is equivalent to $DES[P]$ then, by the result of Luby and Rackoff, it would follow that there exists a pseudorandom permutation generator in $DES[FAN\text{-}IN[O(1)]]$ or $DES[FAN\text{-}IN[O(\log n)$ (provided a one-way function exists).

If one could, for any $G \in P$, construct a permutation of the form T_G in terms of polynomially many permutations of the form S and T_H, where $H \in FAN\text{-}IN[O(\log n)]$, then we would have $DES[FAN\text{-}IN[O(\log n)]] = DES[P]$ and a pseudorandom permutation generator would exist in $DES[FAN\text{-}IN[O(\log n)]]$.

Coppersmith and Grossman [4], and Even and Goldreich [6] considered a class similar to $DES[FAN\text{-}IN[2]]$, but without the polynomial bound on the length of compositions of the permutations. Nevertheless, by analyzing their constructions, we find that permutations of the form T_F where $F \in FAN\text{-}IN[O(\log n)]$ can be expressed as polynomial-length compositions of permutations of the form S, and T_G, where $G \in FAN\text{-}IN[2]$. Thus, we have the following theorem (which is also a consequence of Theorem 5 below).

Theorem 4 (Coppersmith and Grossman [4]; Even and Goldreich [6]): $DES[FAN\text{-}IN[2]] = DES[FAN\text{-}IN[O(\log n)]]$.

Thus, we obtain the same complexity class if the fan-in is anywhere in the between $O(1)$ and $O(\log n)$. Note that this is in spite of the fact that $FAN\text{-}IN[2]$ is a proper subset of $FAN\text{-}IN[O(\log n)]$.

We do not know whether $DES[FAN\text{-}IN[2]] = DES[P]$, but we can show that, for some interesting complexity classes B that are much more powerful than $FAN\text{-}IN[O(\log$ $DES[FAN\text{-}IN[2]] = DES[B]$. Our first result along these lines is the following (bear in mind that $FAN\text{-}IN[O(\log n)]$ is a proper subset of NC^1).

Theorem 5: $DES[FAN\text{-}IN[2]] = DES[NC^1]$.

Proof: It is sufficient to show that, for every function generator G that is in NC^1, the permutation generator T_G is in $DES[FAN\text{-}IN[2]]$. Some parts of our constructions are related to those used by Ben-Or and Cleve in [2].

For $i, j, k \in \{1, ..., n\}$, define $\Phi^k_{i,j} : \{0,1\}^{2n} \to \{0,1\}^{2n}$ as, for all $x, y \in \{0,1\}^n$, $\Phi^k_{i,j}(x, y) = (x, z)$, where

$$z_r = \begin{cases} y_k \oplus (y_i \wedge y_j) & \text{if } r = k \\ y_r & \text{if } r \neq k \ . \end{cases}$$

We first show that, for any $i, j, k \in \{1, ..., n\}$ for which with i and j are both distinct from k, $\Phi^k_{i,j} \in DES[FAN\text{-}IN[2]]$. To do this, define $\Delta^k_{i,j} : \{0,1\}^n \to \{0,1\}^n$ ($i, j, k \in \{1, ..., n\}$) as

$$[\Delta^k_{i,j}(x)]_r = \begin{cases} (x_i \wedge x_j) & \text{if } r = k \\ 0 & \text{if } r \neq k \end{cases}$$

for all $x \in \{0,1\}^n$. Note that $\Delta^k_{i,j} \in FAN\text{-}IN[2]$. Also, it is straightforward to verify that, for all $i, j, k \in \{1, ..., n\}$ with $i, j \neq k$,

$$\Phi^k_{i,j} = S \circ T_{\Delta^1_{i,j}} \circ S \circ T_{\Delta^k_{1,1}} \circ S \circ T_{\Delta^1_{i,j}} \circ S \circ T_{\Delta^k_{1,1}} \ .$$

Therefore, $\Phi^k_{i,j} \in DES[FAN\text{-}IN[2]]$, as claimed.

Now, for $g : \{0,1\}^{m(n)} \times \{0,1\}^n \to \{0,1\}$ and $i \in \{1, ..., n\}$, define the permutation generator $\Gamma_{g,i} : \{0,1\}^{m(n)} \times \{0,1\}^{2n} \to \{0,1\}^{2n}$ as $\Gamma_{g,i} = T_A$, where $A : \{0,1\}^{m(n)} \times \{0,1\}^n \to \{0,1\}^n$ is defined as

$$[A(y, x)]_r = \begin{cases} g(y, x) & \text{if } r = i \\ 0 & \text{if } r \neq i \ , \end{cases}$$

for all $y \in \{0,1\}^{m(n)}$ and $x \in \{0,1\}^n$.

Now, let $G \in NC^1$ be given and let $F : \{0,1\}^{p(n)} \times \{0,1\}^n \to \{0,1\}^n$ and $H : \{0,1\}^{m(n)} \to \{0,1\}^{p(n)}$ be as in Definition 4. In particular, the depth complexity of F over the basis $\{\wedge, \oplus, 1\}$ is $O(\log n)$. Let $(z_1, ..., z_{p(n)+n}) = (H(y), x)$.

We shall show that, for each $i \in \{1, ..., n\}$, $\Gamma_{[G]_i,i} \in DES[FAN\text{-}IN[2]]$. First, note that, for each $i \in \{1, ..., n\}$ and $j \in \{1, ..., p(n) + n\}$,

$$\Gamma_{1,i} \in DES[FAN\text{-}IN[0]] \subset DES[FAN\text{-}IN[2]]$$

and

$$\Gamma_{z_j,i} \in DES[FAN\text{-}IN[1]] \subset DES[FAN\text{-}IN[2]] \ .$$

Therefore, if the depth complexity of $[F]_i$ is 0 then $\Gamma_{[G]_i,i} \in DES[FAN\text{-}IN[2]]$. Also, for any two functions f_1 and f_2, it is straightforward to verify that the identities

$$\Gamma_{(f_1 \oplus f_2),i} = \Gamma_{f_1,i} \circ \Gamma_{f_2,i}$$

and

$$\Gamma_{(f_1 \wedge f_2),k} = \Phi^k_{i,j} \circ \Gamma_{f_1,i} \circ \Phi^k_{i,j} \circ \Gamma_{f_2,j} \circ \Phi^k_{i,j} \circ \Gamma_{f_1,i} \circ \Phi^k_{i,j} \circ \Gamma_{f_2,j} \ ,$$

for all $i, j, k \in \{1, ..., n\}$ with $i, j \neq k$, hold. Therefore, if the depth complexity of $[F]_i$ is d then, by recursively applying the above identities, $\Gamma_{[G]_i,i}$ is expressible as a composition of length $O(4^d)$ of elements of $DES[FAN\text{-}IN[2]]$. Therefore, since, for each $i \in \{1, ..., n\}$, the depth complexity of $[F]_i$ is $O(\log n)$, the composition is of polynomial length, so $\Gamma_{[G]_i,i} \in DES[FAN\text{-}IN[2]]$, as claimed.

Finally, since $\Gamma_{[G]_1,1}, \Gamma_{[G]_2,2}, ..., \Gamma_{[G]_n,n} \in DES[FAN\text{-}IN[2]]$, it follows that

$$T_G = \Gamma_{[G]_1,1} \circ \Gamma_{[G]_2,2} \circ \cdots \circ \Gamma_{[G]_n,n} \in DES[FAN\text{-}IN[2]] \ .$$

□

In is conceivable that NC^1 is equivalent to $SPACE[O(\log n)]$ —or even P—but, given our current knowledge, such equivalences are considered unlikely. Thus, the following theorem extends Theorem 5.

Theorem 6: $DES[FAN\text{-}IN[2]] = DES[SPACE[\frac{1}{2}\log(\frac{n}{3})]]$.

Prior to proving Theorem 6, we prove the following lemma.

Lemma 7: *If a function generator G^n is in $SPACE[w(n)]$ then it is expressible as $G(y,x) = F(H(y),x)$, where $H : \{0,1\}^{m(n)} \rightarrow \{0,1\}^{p(n)}$ is polynomial-time computable, and $F : \{0,1\}^{p(n)+n} \rightarrow \{0,1\}^n$ has the following property. For each $i \in \{1, ..., n\}$, there exists a sequence of $2^{w(n)} \times 2^{w(n)}$ matrices over $\{0, 1, z_1, ..., z_{p(n)+n}, \neg z_1, ..., \neg z_{p(n)+n}\}$, namely*

$$M_1, M_2, ..., M_{t(n)}$$

(where $t(n) \in n^{O(1)}$), such that

$$[F(z_1, ..., z_{p(n)+n})]_i = [M_1 \cdot M_2 \cdot ... \cdot M_{t(n)}]_{1,1} \ ,$$

where the iterated matrix product on the right is defined relative to modulo 2 arithmetic.

Proof: The idea of the proof is to associate each configuration $s \in \{0,1\}^{w(n)}$ with a unique vector of length $2^{w(n)}$ that consists of a 1 in one position and 0s in all other positions. Then the transition functions for any computation are easily expressible in terms of the desired matrices.

More formally, for the function generator G, let $H : \{0,1\}^{m(n)} \to \{0,1\}^{p(n)}$, and $F : \{0,1\}^{p(n)+n} \to \{0,1\}^n$ be as in Definition 5. Fix $i \in \{1,...,n\}$ and let

$$\Pi_1, ..., \Pi_{t(n)} : \{0,1\} \times \{0,1\}^{w(n)} \to \{0,1\}^{w(n)} ,$$

and

$$\alpha_1, ..., \alpha_{t(n)} : \{0,1\}^{w(n)} \to \{1,...,p(n)+n\}$$

be as in Definition 5 for the i-th output bit of F.

For $s \in \{0,1\}^{w(n)}$, associate the vector $\mu(s) \in \{0,1\}^{2^{w(n)}}$ as follows. For $r \in \{1,...,2^{w(n)}\}$,

$$[\mu(s)]_r = \begin{cases} 1 & \text{if } r = 1 + [s]_1 + 2[s]_2 + \cdots + 2^{w(n)-1}[s]_{w(n)} \\ 0 & \text{otherwise .} \end{cases}$$

Clearly, μ is a one-to-one mapping.

Now, for each $j \in \{1,...,t(n)\}$, we define M_j so as to simulate the effect that Π_j and α_j have on each $s \in \{0,1\}^{w(n)}$. For each $k \in \{1,...,2^{w(n)}\}$, the k-th row of M_j has the following form. If $\Pi_j(0,\mu^{-1}(k)) = \Pi_j(1,\mu^{-1}(k))$ then the row has a 1 in position $\mu\left(\Pi_j(0,\mu^{-1}(k))\right)$, and 0s in all other positions. Otherwise (if $\Pi_j(0,\mu^{-1}(k)) \neq \Pi_j(1,\mu^{-1}(k))$), the row has $z_{\alpha_j(\mu^{-1}(k))}$ in position $\mu\left(\Pi_j(1,\mu^{-1}(k))\right)$, and $\neg z_{\alpha_j(\mu^{-1}(k))}$, in position $\mu\left(\Pi_j(0,\mu^{-1}(k))\right)$, and 0s in all other positions. It is then straightforward to verify that, for all $s \in \{0,1\}^{w(n)}$,

$$\mu\left(\Pi_j(z_{\alpha_j(s)}, s)\right) = \mu(S) \cdot M_j .$$

Therefore, the final configuration of the computation is

$$s_{t(n)} = \mu^{-1}\left(\mu(0...0) \cdot M_1 \cdot M_2 \cdot ... \cdot M_{t(n)}\right) ,$$

and $s_{t(n)} = 0...0$ if and only if

$$1 = [[1\ 0\ ...\ 0] \cdot M_1 \cdot M_2 \cdot ... \cdot M_{t(n)}]_1 = [M_1 \cdot M_2 \cdot ... \cdot M_{t(n)}]_{1,1} ,$$

as required. □

Proof of Theorem 6: It is sufficient to show that, for every function generator G that is in $SPACE[\frac{1}{2}\log(\frac{n}{3})]$, the permutation generator T_G is in $DES[FAN\text{-}IN[2]]$. The construction used here can be viewed as an extension of the construction of Theorem 5.

Let $G \in SPACE[\frac{1}{2}\log(\frac{n}{3})]$ be given, and let $H : \{0,1\}^{m(n)} \to \{0,1\}^{p(n)}$, and $F : \{0,1\}^{p(n)+n} \to \{0,1\}^n$ be as in Lemma 7. Let $(z_1, ..., z_{p(n)+n})) = (H(y), x)$.

For any $\sqrt{\frac{n}{3}} \times \sqrt{\frac{n}{3}}$ matrix M whose entries are polynomials over $(z_1, ..., z_{p(n)+n})$, define the permutation generators $\Lambda_M^1, \Lambda_M^2, \Lambda_M^3 : \{0,1\}^{m(n)} \times \{0,1\}^{2n} \to \{0,1\}^{2n}$ as follows. For $e \in \{1,2,3\}$, let $\Lambda_M^e = T_D$, where $D : \{0,1\}^{m(n)} \times \{0,1\}^n \to \{0,1\}^n$ is defined as

$$[D(y,x)]_r = \begin{cases} M_{\beta((r \operatorname{mod} \frac{n}{3})+1)} & \text{if } 1 + (e-1)(\frac{n}{3}) \leq r \leq e(\frac{n}{3}) \\ 0 & \text{otherwise ,} \end{cases}$$

for all $y \in \{0,1\}^{m(n)}$ and $x \in \{0,1\}^n$. Here,

$$\beta : \{1, ..., \tfrac{n}{3}\} \to \{1, ..., \sqrt{\tfrac{n}{3}}\} \times \{1, ..., \sqrt{\tfrac{n}{3}}\}$$

is the natural bijection defined as

$$\beta(r) = (((r-1) \bmod \sqrt{\tfrac{n}{3}}) + 1, ((r-1)\operatorname{div}\sqrt{\tfrac{n}{3}}) + 1) .$$

As in the proof of Theorem 5, it is sufficient to show that, for any $i \in \{1, ..., n\}$, $\Gamma_{[G]_{i},i} \in DES[FAN\text{-}IN[2]]$. Let $i \in \{1, ..., n\}$ be given, and let $M_1, M_2, ..., M_{t(n)}$ be the corresponding $\sqrt{\frac{n}{3}} \times \sqrt{\frac{n}{3}}$ matrices over $\{0, 1, z_1, ..., z_{p(n)+n}, \neg z_1, ..., \neg z_{p(n)+n}\}$ that exist from the application of Lemma 7.

We shall show that, for each $e \in \{1,2,3\}$, $\Lambda^e_{M_1 \cdot M_2 \cdot ... \cdot M_{t(n)}} \in DES[FAN\text{-}IN[2]]$. First note that, for each $j \in \{1, ..., t(n)\}$, since the entries of M_j are in $\{0, 1, z_1, ..., z_{p(n)+n}, \neg z_1, ..., \neg z_{p(n)+n}\}$, it follows that $\Lambda^e_{M_j} \in DES[FAN\text{-}IN[1]] \subset DES[FAN\text{-}IN[2]]$ ($e \in \{1,2,3\}$).

At this point, we define the permutations $\Theta_1, \Theta_2, \Theta_3 : \{0,1\}^{2n} \to \{0,1\}^{2n}$. First, we introduce the following permutations in $DES[FAN\text{-}IN[2]]$. Let $\Phi^k_{i,j} : \{0,1\}^{2n} \to \{0,1\}^{2n}$ (for $i,j,k \in \{1, ..., n\}$) be as in the proof of Theorem 5. Also, define $\Upsilon_1, \Upsilon_2, \Upsilon_3 : \{0,1\}^{2n} \to \{0,1\}^{2n}$ as, for all $x \in \{0,1\}^n$, and $u, v, w \in \{0,1\}^{\frac{n}{3}}$,

$$\Upsilon_1(x,u,v,w) = (x,u,v,w)$$

$$\Upsilon_2(x,u,v,w) = (x,w,u,v)$$

$$\Upsilon_3(x,u,v,w) = (x,v,w,u) .$$

It is easily shown that $\Upsilon_1, \Upsilon_2, \Upsilon_3 \in DES[FAN\text{-}IN[2]]$. Finally, for $e \in \{1,2,3\}$, define $\Theta_e : \{0,1\}^{2n} \to \{0,1\}^{2n}$ as the composition

$$\textbf{for } a = 1 \textbf{ to } \sqrt{\tfrac{n}{3}} \left(\textbf{for } b = 1 \textbf{ to } \sqrt{\tfrac{n}{3}} \left(\Upsilon_e \circ \Phi^{\beta(a,b)+2\frac{n}{3}}_{\beta(a,1),\beta(1,b)+\frac{n}{3}} \circ \cdots \circ \Phi^{\beta(a,b)+2\frac{n}{3}}_{\beta(a,\sqrt{\frac{n}{3}}),\beta(\sqrt{\frac{n}{3}},b)+\frac{n}{3}} \circ \Upsilon_e^{-1} \right) \right.$$

Since $\Theta_1, \Theta_2, \Theta_3$ are polynomial-length compositions of elements of $DES[FAN\text{-}IN[2]]$, $\Theta_1, \Theta_2, \Theta_3 \in DES[FAN\text{-}IN[2]]$.

Now, the key part of our construction is the identity, for $e \in \{1, 2, 3\}$,

$$\Lambda^{(e+2)\mathrm{mod}3}_{M \cdot M'} = \Theta_e \circ \Lambda^{(e+1)\mathrm{mod}3}_{M'} \circ \Theta_e \circ \Lambda^{e}_{M} \circ \Theta_e \circ \Lambda^{(e+1)\mathrm{mod}3}_{M'} \circ \Theta_e \circ \Lambda^{e}_{M} \ .$$

By applying this identity recursively, and using the fact that, for each $j \in \{1, ..., t(n)\}$, $\Lambda^{e}_{M_j} \in DES[FAN\text{-}IN[2]]$ ($e \in \{1, 2, 3\}$), it follows that $\Lambda^{e}_{M_1 \cdot M_2 \cdot ... \cdot M_{t(n)}} \in DES[FAN\text{-}IN[2]]$ ($e \in \{1, 2, 3\}$).

Since

$$[F(z_1, ..., z_{p(n)+n})]_i = [M_1 \cdot M_2 \cdot ... \cdot M_{t(n)}]_{1,1} \ ,$$

it follows that

$$T_{[G]_i, i} = \Phi^{i}_{1+(e-1)(\frac{n}{3}), 1+(e-1)(\frac{n}{3})} \circ \Lambda^{e}_{M_1 \cdot M_2 \cdot ... \cdot M_{t(n)}} \circ \Phi^{i}_{1+(e-1)(\frac{n}{3}), 1+(e-1)(\frac{n}{3})} \ ,$$

whenever $i \notin \{1 + (e-1)(\frac{n}{3}), ..., e(\frac{n}{3})\}$ and, therefore, $T_{[G]_i, i} \in DES[FAN\text{-}IN[2]]$.

Finally, since $\Gamma_{[G]_1, 1}, \Gamma_{[G]_2, 2}, ..., \Gamma_{[G]_n, n} \in DES[FAN\text{-}IN[2]]$, it follows that

$$T_G = \Gamma_{[G]_1, 1} \circ \Gamma_{[G]_2, 2} \circ \cdots \circ \Gamma_{[G]_n, n} \in DES[FAN\text{-}IN[2]] \ .$$

□

It would be interesting to extend this work to showing that $DES[FAN\text{-}IN[2]] = DES[B]$ for more powerful complexity classes B. In particular, the following problem is of interest.

Open Problem: *Determine whether or not* $DES[FAN\text{-}IN[2]] = DES[P]$.

4 Further Work

We can extend our results to function generators G in $SPACE[c \cdot \log n]$, for arbitrary $c > 0$, and for function generators in a nonuniform version of *nondeterministic* logarithmic space. We cannot show that for such function generators G, $T_G \in DES[FAN\text{-}IN[2]]$, but we can show that, given a pseudorandom function generator in one of these classes, we can nevertheless construct a pseudorandom permutation generator in $DES[FAN\text{-}IN[2]]$. Some of these results are explained in [3] and will appear in the final paper.

5 Acknowledgement

Thanks to Charles Rackoff for suggesting this direction of research.

References

[1] D. A. Barrington, "Bounded-Width Polynomial-Size Branching Programs Recognize Exactly Those Languages in NC^1," *J. Computer System Sci.* Vol. 38, pp. 150–164, 1989.

[2] M. Ben-Or, and R. Cleve, "Computing Algebraic Formulas Using a Constant Number of Registers," *Proc. 20th Ann. ACM Symp. on Theory of Computing*, pp. 254–257, 1988.

[3] R. Cleve, *Methodologies for Designing Block Ciphers and Cryptographic Protocols (Part I)*, Ph.D. Thesis, University of Toronto, 1989.

[4] D. Coppersmith, and E. Grossman, "Generators for Certain Alternating Groups with Applications to Cryptography," *SIAM J. Appl. Math.*, pp. 624-627, 1975.

[5] D. Coppersmith, "Cryptography," *IBM J. Res. Develop.*, Vol. 31, No. 2, pp. 244–248, 1987.

[6] S. Even, and O. Goldreich, "DES-Like Functions Can Generate the Alternating Group," *IEEE Trans. on Information Theory*, pp. 863–865, 1983.

[7] Luby, M., and C. Rackoff, "How to Construct Pseudorandom Permutations From Pseudorandom Functions," *SIAM J. Comput.*, Vol. 17, No. 2, pp. 373–386, 1988.

[8] E. Biham, and A. Shamir, "Differential Cryptanalysis of DES-like Cryptosystems," these proceedings, 1990.