Foiling Birthday Attacks in Length-Doubling Transformations

Benes: a non-reversible alternative to Feistel

William Aiello and Ramarathnam Venkatesan

Math and Cryptography Research Group, Bell Communications Research, 445 South Street, Morristown NJ 07960. {venkie,aiello}@bellcore.com

Abstract. For many cryptographic primitives, e.g., hashing and pseudorandom functions & generators, doubling the output length is useful even if the doubling transformation is not reversible. For these cases, we present a non-reversible construction based on a Benes network, as an alternative to the traditional Feistel construction (which is the basis of DES).

Assuming that a given primitive behaves like an n-bit to n-bit random function, we present a length-doubling scheme that yields a $2n$-bit to $2n$-bit function that provably requires $\Omega(2^n)$ queries to distinguish with $\Theta(1)$ probability from a truly random function of that length. This is true even if the adversary is of unlimited computing power and is allowed to query the function adaptively. Our construction is minimal in the sense that omitting any operation makes the resulting network susceptible to birthday attacks using $O(2^{n/2})$ queries.

Feistel networks also use truly random n-bit functions to achieve $2n$-bit functions. Luby and Rackoff [16] showed that 3 and 4 round Feistel networks require $\Omega(2^{n/2})$ queries to distinguish with $\Theta(1)$ probability from truly random. We show that these bounds are tight by showing that these networks are susceptible various types of birthday attacks using $O(2^{n/2})$ queries.

1 Introduction

Many cryptographic primitives in practice are believed to behave like "random functions" or at least this turns out be a good empirical approximation. Using the random function model, several works have made a rigorous analysis of some standard and new cryptographic schemes [1, 2, 20, 22].

In fact, most designs of primitives (e.g., MD5,SHA) involve *rounds* of "mixing" the input, creating an "avalanche effect," and making the result look "random." In the case of DES, one starts with a 32-bit valued primitive which is iterated 16 rounds to get a 64-bit valued "random permutation." In the case of, say, 128-bit cryptographic hash function constructions based on 64-bit block cyphers, meticulous care must be exerted to ensure that the resultant function behaves like a 128-bit random function. Otherwise, it may yield to 2^{32} step birthday attacks as a 64-bit random function would. In fact our work originated in the process of

U. Maurer (Ed.): Advances in Cryptology - EUROCRYPT '96, LNCS 1070, pp. 307-320, 1996.

looking for a practical scheme that avoids these problems and admits an analysis in a random function model. Recent attacks on some of these cryptographic primitives are practical [10] or bordering on feasibility [21]. Hence constructions using old smaller-length primitives to get new longer-length primitives—with a provable increase in security—would be desirable.

The Feistel transformation (See Fig. 1) is an obvious candidate for such a construction, and indeed, its security properties have been well studied. Luby and Rackoff [16] study three and four rounds of the Feistel transformation. They show that using pseudo-random functions (introduced in [11]) as n-bit primitives yields a $2n$-bit pseudo-random permutation. In a crucial step in their proof, they analyze the case when the primitives are n-bit truly random functions. They show that the three-round Feistel network requires $\Omega(2^{n/2})$ queries to distinguish with $\Theta(1)$ probability from a truly random $2n$ bit function. (Later, Maurer [17] simplified their proof.) However, they show that the same network can be distinguished in a constant number of queries if the adversary is allowed to ask for inverses of points chosen in the range. They then show that the four-round Feistel network requires $\Omega(2^{n/2})$ queries to distinguish with $\Theta(1)$ probability from a truly random $2n$ bit function even when inverse queries are allowed.

Luby and Rackoff were concerned with the polynomial time invariant model where the difference between the security of 2^n and $2^{n/2}$ is not important. However, in practice this difference can be crucial. For example, using a construction with security $2^{n/2}$ to design a 64-bit function from 32-bit random primitives would yield a function which could be distinguished in approximately 2^{16} queries whereas using a construction with 2^n security would require approximately 2^{32} queries. Hence, an important goal is to resolve the query security of the three-and-four-round Feistel constructions. In this paper we show that the Luby-Rackoff bounds are tight by showing that these networks are susceptible to various types of birthday attacks using $O(2^{n/2})$ queries.

Given the limited security provided by using just a few rounds of Feistel it is natural to ask if there is an alternative construction which admits analysis and provides greater security. In this paper we provide such an alternative which we call the Benes network construction (two back-to-back butterflies, see Fig. 1). The Benes construction has the following properties. First, it is formally provable that our functions cannot be distinguished with $\Theta(1)$ probability, in an information theroretic sense, from truly random ones, unless it is queried $\Omega(2^n)$ times. Second, it is minimal in the sense that deleting any one of the operations reduces the security. Third, it yields constructions and variations that are efficient.

We introduce a formal notion of (query) security in Section 2. Generalizing birthday and other statistical attacks, we introduce the notion of a *dependency graph* on a (query) sample of size q. The nodes of this graph are pairs of strings which are internal variables of the construction that are not directly observable from the input-output relations. Two types of dependencies are identified for these variables which are signified by edges of two colors between corresponding nodes. We show that the output random variables are k-wise independent if and only if there are no alternatingly colored l-cycles in the graph, for $l \leq k \leq q$. We then show that with very high probability the dependency graph has no such q cycles.

Our construction has many applications including authentication, random number generation, stream or packet encyphement, and reducing correlations in key exchange protocols. We also suggest a way to double hash code lengths which admits some parallelism and eludes attacks like [21, 10]. (See Section 3.)

2 Definitions and Main Results

Let $\mathbf{F}_n$ be the set of all functions $f : \{0,1\}^n \rightarrow \{0,1\}^n$. We would like to construct a family of functions from $2n$ bits to $2n$ bits using functions from $\mathbf{F}_n$ such that the functions drawn randomly from this new family behave very much like randomly chosen functions from $\mathbf{F}_{2n}$. To formalize this we make the following definitions.

Adaptive Adversaries: Let A be an information theoretic adversary (i.e., of unbounded computing power) which can make queries to a function as an oracle. We limit our adversaries (only) by the number of query points. A can *adaptively* make queries and obtain function values at the query points. Then, over this sample he could perform *any* test. We say A distinguishes $\mathbf{G}_n \subset \mathbf{F}_n$ from $\mathbf{H}_n \subset \mathbf{F}_n$ with advantage

$$|\Pr[A^g(1^n)=1|g \in_R \mathbf{G}_n] - \Pr[A^g(1^n)=1|g \in_R \mathbf{H}_n]|.$$

When $\mathbf{G}_n$ and $\mathbf{H}_n$ are understood from context let $q_A(n)$ be the number of queries made by $A(1^n)$.

Definition 1 (Query Security) *A function family $\mathbf{G}_n \subset \mathbf{F}_n$ has query security at least $s(n)$ if there is an constant $\alpha > 0$ such that all information theoretic adversaries A can distinguish $\mathbf{G}_n$ from $\mathbf{F}_n$ with advantage at most $(q_A(n)/s(n))^\alpha$. For the purposes of this paper α will always be 2.*

For example, suppose that $\mathbf{G}_n$ can be distinguished from $\mathbf{F}_n$ in q queries with advantage at most $q^2/2^n$. Then the query security of $\mathbf{G}_n$ is at least $2^{n/2}$. When it is clear from context that we are using the information theoretic model we will simply denote query security by security.

The Feistel transformation is a well-known function family from $2n$ bits to $2n$ bits and it is natural to ask whether it achieves sufficient security. Let us first review the construction.

For any $f \in \mathbf{F}_n$ define the $2n$ bit to $2n$ bit Feistel transformation G_f as $G_f(l,r) = (r, l \oplus f(r))$. Given a collection of k functions $\mathcal{A} = (f_1, f_2, \ldots, f_k)$ from n bit to n bit define $G_{\mathcal{A}} \in \mathbf{F}_{2n}$ as the composition of $G_{f_1}, G_{f_2}, \ldots, G_{f_k}$. More explicity, if the input to $G_{\mathcal{A}}$ is denoted l_0, r_0 then the output l_k, r_k can be computed by the recurrence $l_i = r_{i-1}, \quad r_i = f_i(r_{i-1}) \oplus l_{i-1}$. We say that l_i, r_i are the output of the ith round. DES may be described by $G_{\mathcal{A}}$ where there are $k = 16$ functions (rounds) and the functions f_i are given by the DES key and the s-boxes. For now we will consider functions which are truly random functions.

Denote the set of all $G_{\mathcal{A}}$ (where each of the k functions in $\mathcal{A}$ range over all the functions in $\mathbf{F}_n$) as $\mathbf{G}_{2n}^k$

As is well known, the Feistel transformation is a permutation. For permutations the definition of security must be modified since a random permutation on n bits can be distiguished from a random function from n bits to n bits in $\theta(2^{n/2})$ queries (a random function will repeat with constant probability whereas a permutation obviously will not). When discussing a permutation family we implicitly assume that the adversary is trying to distinguish it from a random permutation, and the definition of security is adjusted accordingly.

Not only is the Feistel transformation a permutation, it is also invertible (even without inverting the underlying n bit to n bit random functions). Following [16], define super query security, or super security, identically to query security except that the adversary is allowed to query both the function and its inverse.

Luby and Rackoff [16] studied the security and super security of Feistel transformations (with random functions) with a small number of rounds. They broke two round Feistel in a constant number of function queries, and they broke three round Feistel in a constant number of function and inverse queries. To complement these results they derived lower bounds, summarized below, for the security and super-security of the three and four round Feistel transformations, respectively.

THEOREM 1 *(Luby, Rackoff) The security of* $\mathbf{G}_{2n}^3$ *is* $\Omega(2^{n/2})$ *and the super security of* $\mathbf{G}_{2n}^4$ *is also* $\Omega(2^{n/2})$.

The proof of this theorem, and the others cited in this section, will be given in subsequent sections.

For DES $n = 32$ and so the above suggests that three (four) rounds of DES may have (super) query security of 2^{16} or more. In practice, this is very little security and so a natural question is whether the lower bound of Luby and Rackoff can be improved. We show that it cannot by providing the following matching upper bound.

THEOREM 2 *The security of* $\mathbf{G}_{2n}^3$ *and* $\mathbf{G}_{2n}^4$ *is* $O(2^{n/2})$. *Moreover, the adversary need not make any inverse queries.*

In [27] three generalizations of the Feistel transformation are described. These transformations map kn bit to kn bits using functions from $\mathbf{F}_n$. Zheng, et. al. show that if they are run for sufficiently many rounds (i.e., $2k-1$, $2k$, and $k+1$ depending on the variation) then they have security $\Omega(2^{n/2})$. Using arguments similar to those for the Feistel transformation, we can show that this is tight. We omit the details for now.

Given the limited security of the three and four round Feistel transformations, a natural problem is to construct a $2n$-to-$2n$ bit transformation which achieves security $\Omega(2^n)$ using n-to-n bit random functions and just a few rounds. The emphasis here is on security and not on invertibility so that non-invertible transformations are acceptable.

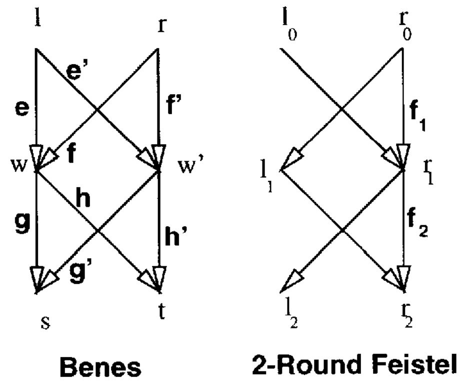

Fig. 1. Benes and Feistel (2-round) transformations from $2n$-bits to $2n$-bits. Edge labels denote random functions from n-bits to n-bits. Unlabelled edges are identities. Node values are bitwise xor of values of incoming edges.

In this paper we describe one solution to this problem. (See Fig. 1) Given four functions e, f, e', f' from n bits to n bits, we use them to define the butterfly transformation from $2n$ bits to $2n$ bits. On input l, r, the output w, w' is given by

$$w = e(l) \oplus f(r); \quad w' = e'(l) \oplus f'(r).$$

Let $\mathbf{B}_{2n}$ be the set of all butterfly transformations where e, f, e', f' range over $\mathbf{F}_n$. Given eight functions from n bits to n bits, e, f, e', f', g, g', h, h', define the Benes transformation (back-to-back butterfly) as the composition of two butterfly transformations. The first stage is as above. Then, the second transformation maps w, w' into s, t as follows:

$$s = g(w) \oplus g'(w'); \quad t = h(w) \oplus h'(w').$$

Let $\mathbf{B}^2_{2n}$ be the set of all Benes transformations where each of the eight functions range over $\mathbf{F}_n$.

Clearly, functions drawn randomly from $\mathbf{B}^2_{2n}$ are information theoretically distinguishable from functions drawn randomly from $\mathbf{F}_{2n}$. However, as we state formally below, even an information theoretic adversary requires a large number of queries to the randomly drawn functions in order to distinguish $\mathbf{B}^2_{2n}$ from $\mathbf{F}_{2n}$ with noticeable advantage.

THEOREM 3 *The Benes family $\mathbf{B}^2_{2n}$ has security $\Theta(2^n)$.*

Corollary 1 *The Benes transformation using n-bit pseudo-random functions in place of n-bit random functions yields a $2n$-bit pseudo-random function.*

The proof of this corollary follows the proof in [16] and is omitted.

In a natural sense the Benes construction is optimal. In fact, we next show that its security drops to $O(2^{n/2})$ queries, if any edge is *deleted*.

THEOREM 4 (Edge Optimality) *The Benes and modified Benes transformations are minimal: i.e., deletion of any of its edges results in a function which has security $O(2^{n/2})$.*

Simplifications: Depending on the application, some of the random functions can be replaced by simpler hash functions as described below.

THEOREM 5 *The Benes transformation using k-universal hash functions in the top butterfly produces outputs which cannot be distinguished from k-wise independent outputs in q queries with advantage better than $q^2/2^{2n}$.*

If the two cross edges in the first butterfly, e', and f, are replaced by the identity function we call the resulting transformation the modified Benes transformation. All of the results above hold for the modified Benes transformation as well.

Finally, we would like to mention another candidate method for producing a $2n$ bit to $2n$ function family from a n bit to n bit function family. Suppose that the functions in the function family $\mathbf{H}_n \subset \mathbf{F}_n$ can be indexed by a key of length n. Then a function family from $2n$ to $2n$ bits with a key of length $2n$ can be constructed as follows. Given k_1, k_2 as the $2n$ bit key and w, x as the $2n$-bit input, compute the output y, z as follows. First compute $k_1' = f_{k_1}(w)$ and $k_2' = f_{k_2}(w)$. Then compute $y = f_{k_1'}(x)$ and $z = f_{k_2'}(x)$. This construction is similar to the construction of Goldwasser, Goldreich, and Micali [11]. Using analysis similar to that in [11] it can be shown that if $\mathbf{H}_n$ is pseudo-random (as defined in [11]) then the new family is pseudo-random [24, 1]; however, a simple birthday attack on k_1' or k_2' upper bounds the query security by $O(2^{n/2})$ [24].

3 Applications to Hashing

Our construction gives a hueristic for doubling the output length of keyed hash functions (See for example [23]). Let H be a one-way hash function that behaves as a random function to n bits when keyed by a random string K. Define $\tilde{H}_K(x) \equiv H(K, x)$. MD5 is a candidate for such a hash function with $n = 128$. Let $\bar{K} = K_1, \ldots, K_8$ and let $B^2_{\bar{K}}$ be the modified Benes transformation where the random functions are instantiated by the keyed hash functions H_{K_i}, $i := 1, \ldots, 8$. Now to hash a document D, devide the text in some well-defined way into two equal parts D_0 and D_1, and compute $B^2_{\bar{K}}(D_0, D_1)$. The output length of $B^2_{\bar{K}}$ is twice that of $H(K, D)$. Surprisingly, the running times have approximately the same ratio. This is because the four hash functions computed in the top butterfly of the Benes transformation each operate on only half the document. The hash computations in the bottom butterfly are all on inputs of size n independent of the length of the document. Note that the Benes transformation admits some parallelism: two independent processes would run in approximately half the time of one sequential process.

To avoid attacks such as those in [5, 7, 10, 21], $\bar{K}$ may itself need to be seeded in such a way as to not allow the adversary to vary one of the eight keys while leaving the others fixed. In practice this can be instantiated by letting $K_1, \ldots, K_8$ be the output of a cryptographically secure random generator on a random seed [6, 26]. Such a transformation is one-way and almost surely one-to-one.

4 Security of Feistel Transformations

Proof of theorem 2

3-round case: The adversary evaluates the given function on $q+2$ inputs. The first q distinct inputs are $(l_0, r_0^{(1)}), \ldots, (l_0, r_0^{(q)})$. For notational simplicity let $r_1^{(m)} = \alpha_m$, and $r_2^{(m)} = \beta_m$, and $r_3^{(m)} = \gamma_m$, $m := 1, \ldots, q$. Using this notation, the corresponding outputs are labelled by $(\beta_1, \gamma_1), \ldots, (\beta_q, \gamma_q)$. (See Fig. 2)

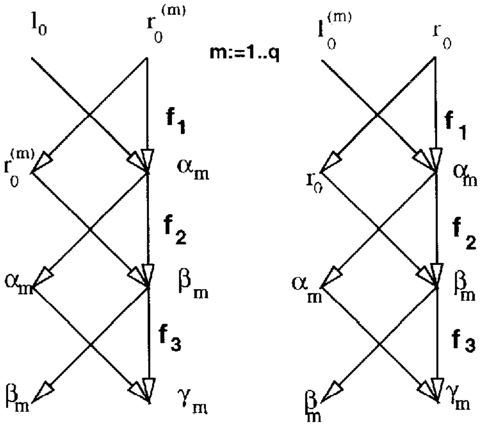

Fig. 2. Distinguishing Three Round Feistel Tranformations from Random Ones:Attacks matching the lower bounds on Luby-Rackoff pseudo-random functions

The adversary computes $\beta_m \bigoplus r_0^{(m)}$, $m := 1, \ldots, q$, and checks for collisions. If he finds a collision at $m = i$ and $m = j$ he queries at $l_0', r_0^{(m)}$, and obtains outputs β_m', γ_m', for $m := i, j$. If now $\beta_i' \bigoplus r_0^i = \beta_j' \bigoplus r_0^j$, the adversary rejects the function as non-random.

Now let us analyze how G_{f_1, f_2, f_3} fares under this adversary. With probability $\Omega(q^2/2^n)$, (for $q^2/2^n < \delta < 1$) there exist an i and a j such that $f_1(r_0^{(i)}) = f_1(r_0^{(j)})$. Since l_0 is held constant and $\alpha_m = l_0 \bigoplus f_1(r_0^{(m)})$, we have $\alpha_i = \alpha_j$ and $f_2(\alpha_i) = f_2(\alpha_j)$. Since $\beta_m = r_0^{(m)} \bigoplus f_2(\alpha_m)$ one gets the collision $\beta_i \bigoplus r_0^{(i)} = \beta_j \bigoplus r_0^{(j)}$. It is easy to check that $\beta_i' \bigoplus r_0^{(i)} = \beta_j' \bigoplus r_0^{(j)}$ as well. Hence with probability $\Omega(q^2/2^n)$ the adversary will reject $G_{\mathcal{A}}$. It is easy to show that the adversary will reject truly random permutation with probability $O(q^2/2^{2n})$. We omit the details here.

We have a second attack on 3-round Feistel but due to lack of space we omit it. We can also upper bound the security of the generalized Feistel transformations described by Zheng, et. al. [27]. The two $k+1$ round transformations are broken using attacks similar to the attack above and the $2k-1$ round transformation is broken using an attack similar to our second attack on the three rounds. The complete description of these attacks will be given in the full paper.

4-round Case: In addition to the notation above let $r_4^{(m)} = \delta_m$ Adversary evaluates the function on q distinct inputs $(l_0^{(m)}, r_0)$ and gets outputs γ_m, δ_m, and checks whether there exists i, j so that $l_i \bigoplus \gamma_i = l_j \bigoplus \gamma_j$. It can be checked that the probability this occurs for the 4-round Feistel is approximately twice the probability this occurs for a truly random permutation.

5 Analysis of Benes Transformation

Recall that we do not insist on invertible transformations. As a starting point for understanding the implications of this assumption we first analyze the K transformation which we now define. As we will see, the Benes transformation is simply the bitwise xor of two independent K transformations. While the K transformation has the same security as 3-round Feistel transformations, it is easier to analyze.

Given e, f, g, and h in $\mathbf{F}_n$ and input l, r, $K_{e,f,g,h}(l, r)$ is computed as follows. Let $w = e(l) \bigoplus f(r)$. Then the output s, t is simply $g(w), h(w)$. To visualize the K transformation simply delete all the edges incident to the intermediate node w' in the Benes transformation (See Figure 3).

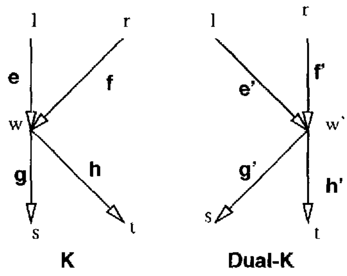

Fig. 3. The K and Dual-K transformations, which jointly yield a 2x2 double butterfly

Lemma 1 *If e, f, g, and h are truly random functions then $K_{e,f,g,h}$ has security $\Theta(2^{n/2})$.*

Proof: To show that the security is $\Omega(2^{n/2})$ note that if the $w_1, \ldots, w_q$ are all distinct then all the bits of the output s_i, t_i, $i = 1, \ldots, q$, are uniform and independent. Hence, following [16] and [17], the distinguishing advantage is at most the probability that the w_i are not all distinct. For any two distinct inputs, l_i, r_i and l_j, r_j, the probability that $w_i = w_j$ is $1/2^n$, since e and f are truly random. It follows that the distinguishing advantage is at most $q^2/2^{n+1}$.

This security bound is tight. In $\Theta(2^{n/2})$ queries of K there exists i, j such that $w_i = w_j$, and hence $s_i, t_i = s_j, t_j$, with constant probability, . But the probability that the entire output collides for a random function in this many queries is $\Theta(2^{-n})$.

Simplifications: If e is a 2-universal hash function and f is the identity function (or visa versa) one gets the same security. Call this the modified K transformation. To achieve a lower bound on security the analysis is the same as above except that there are two cases when analyzing the probability that $w_i = w_j$. If $l_i = l_j$ then r_i must be distinct from r_j (since queries are distinct), hence w_i is distinct from w_j and the probability is zero. If $l_i \neq l_j$ then the probability that $w_i = w_j$ is 2^{-n} since the hash function is 2-universal. We omit the simple upper bound of the security.

Having analyzed K, we define $B^2(l, r)$ as the bitwise exclusive-or of the outputs of $K_{e,f,g,h}(l, r)$ and $K_{e',f',g',h'}(l, r)$. We will show that this exclusive-or at least *squares* the security. For a quick review of our notation, see Figure 1. We will now prove Theorem 3.

Proof of Theorem 3:

Before analyzing the double butterfly transformation we will first show that a single butterfly produces random and independent output except when the inputs have a special form. Given a set of q pairs $(w_1, w_1'), \ldots, (w_q, w_q')$, the **dependency graph** on these pairs is defined as follows. Consider the pairs as nodes in a graph. If any two nodes in the graph are equal in the left-hand coordinate put an edge between them and color it with the label "left." Edges so colored will be called "left edges." If any two nodes in the graph are equal in the right-hand coordinate put a "right edge" between them.

Note that the edge relation is transitive. For example, if there is a right edge between node u and v and another right edge between v and w, then there is a right edge between u and w.

Define a *bichromatic path or cycle* to be a path or cycle, respectively, which contains both left and right edges if it is of length greater than one. (For notational convenience we will consider every path of length one to be bichromatic). Define an *alternating path or cycle* to be a path or cycle, respectively, in which the edges stricly alternate color as the path or cycle is traversed. Clearly, an alternating path (cycle) is a bichromatic path (cycle). The following is also true.

Claim 1 *If there is a bichromatic cycle then there is an alternating cycle.*

Proof: Omitted.

We can now state our main lemma.

Lemma 2 *The outputs of a butterfly transformation are random and independent iff the dependency graph of the inputs has no alternating cycles.*

Proof:

For reasons that will be clear below we will use the notation of the second butterfly in the double butterfly transformation. That is, the inputs and outputs are denoted $(w_1, w'_1), \ldots, (w_q, w'_q)$ and $(s_1, t_1), \ldots, (s_q, t_q)$, respectively, where $s_i = g(w_i) \oplus g'(w'_i)$ and $t_i = h(w_i) \oplus h'(w'_i)$.

To start let us describe a simple relationship for alternating paths. Without loss of generality let $(w_1, w'_1), (w_2, w'_2), \ldots, (w_{l+1}, w'_{l+1})$ be an alternating path with l edges. Let $(s_1, t_1), \ldots, (s_{l+1}, t_{l+1})$ be the associated outputs. The following claim can be easily verified by induction.

Claim 2 *If an alternating path begins with a right-edge and ends with a left-edge, then*

$$\bigoplus_{m=1}^{l+1} s_m = g(w_1) \bigoplus g'(w'_{l+1})$$

If the path begins with a left-edge rather than a right-edge, $g'(w'_1)$ is substituted for $g(w_1)$ in the above. If the path ends with a right-edge rather than a left-edge, $g(w_{l+1})$ is substituted for $g'(w'_{l+1})$ in the above.

The expressions for $\bigoplus_{m=1}^{l+1} t_m$ in terms of h and h' are analogous.

Demonstrating the correlation due to an alternating cycle is easy. An alternating cycle is simply an alternating path with an even number of edges where the node 1 and node $l+1$ are identified. (i.e., $w_{l+1} = w_1$ and $w'_{l+1} = w'_1$). Using the expression above for even length paths, the sum of the output values around the cycle is given by $\left(\bigoplus_{m=1}^{l} s_m\right) \bigoplus s_{l+1} = g(w_1) \bigoplus g'(w'_{l+1})$. Since $s_{l+1} = g(w_{l+1}) \bigoplus g'(w'_{l+1})$ and $g(w_{l+1}) = g(w_1)$, it follows that $\bigoplus_{m=1}^{l} s_m = 0$ for an alternating cycle.

Now suppose that there is no alternating cycle. We will show that all of the outputs are random and independent. We will first show that $s_1, \ldots, s_q$ are uniform and independent. A similar argument shows that the $t_1, \ldots, t_q$ are uniform and independent. That the two sequences are independent of each other follows from the fact that g and g' are indpendent of h and h'.

We will use the following claim which follows from the transitivity of the edges and the lack of alternating cycles.

Claim 3 *If there is a bichromatic path between two nodes, then there is a unique alternating path between those two nodes which is also the shortest path.*

Assume that the dependency graph is connected. (If not, repeat the analysis below on each component.) We start by finding a node which has only right edges or only left edges incident. It is easy to check that such a node must exist since the dependency graph has no alternating cycles. Without loss of generality, assume that this node has only right edges. Relabel this node to (w_1, w'_1).

Now create a breadth-first search tree with (w_1, w'_1) as the root and label the nodes according to the breadth-first order. As is well known, the tree paths are the shortest paths between the root and the other nodes. By Claim 3 all shortest paths are alternating paths. Hence, the color of the edges of the tree alternate with the level. For example, the first level edges (between the root and the depth 1 nodes) are colored "right," and the second level edges are colored "left," etc. We will use the following claim about this tree.

Claim 4 *(i) Suppose a node v has breadth-first order i and depth $d > 0$ which is even. Let the right coordinate of v be w'_i. No other node with breadth-first order less than i has right coordinate equal to w'_i. In fact, all other nodes with right coordinate w'_i are children of v.*

(ii) Suppose a node v has breadth-first order i and depth $d \geq 1$ which is odd. Let the left coordinate of v be w_i. No other node with breadth-first order less than i has left coordinate equal to w_i. In fact, all other nodes with left coordinate equal to w_i are children of v.

Proof: (i) Due to the fact that the levels of the tree alternate color, v is connected to its parent by a left edge and connected to all of its children by right edges. If v were connected by a right edge to another node in the discrepancy graph besides its children in the breadth-first search tree, then there would be a bichromatic cycle in the discrepancy graph. (ii) Similar. ■

For fixed but arbitrary values of $\sigma_1, \ldots, \sigma_q$, define $E(i)$ for $i > 0$ to be the event that $(s_1 = \sigma_1) \wedge \cdots \wedge (s_i = \sigma_i)$ and $E(0)$ to be the null event where, as before, $s_i = g(w_i) \oplus g'(w'_i)$. We must show that $\Pr[E(q)] = 2^{-qn}$. We start as follows:

$$\Pr[E(q)] = \sum_{\bar{g}_1} \Pr[g(w_1) = \bar{g}_1] \Pr[E(q) \mid g(w_1) = \bar{g}_1].$$

Since $\Pr[g(w_1) = \bar{g}_1] = 2^{-n}$ for all $\bar{g}_1 \in \{0,1\}$, it is enough to show that $\Pr[E(q) \mid g(w_1) = \bar{g}_1] = 2^{-qn}$ for all $\bar{g}_1$. Fix an arbitrary value of $\bar{g}_1$. As can be easily verified, it is enough to show that $\Pr[s_i = \sigma_i \mid E(i-1) \wedge g(w_1) = \bar{g}_1] = 2^{-n}$ for all $i := 1, \ldots, q$.

First consider $i := 1$. In this case, $\Pr[g(w_1) \oplus g'(w'_1) = \sigma_1 \mid g(w_1) = \bar{g}_1]$ which is clearly equal to 2^{-n}.

Now consider the case when $i \geq 1$. Let $1 < \alpha < \beta < \gamma < \cdots < i$ be the breadth-first order of the nodes on the path from the root to (w_i, w'_i). That is, this path is given by $(w_1, w'_1), (w_\alpha, w'_\alpha), (w_\beta, w'_\beta), \ldots, (w_i, w'_i)$. Suppose (w_i, w'_i) has odd depth. Using the path formula in Claim 2 the following holds,

$$\Pr[g(w_i) \oplus g'(w'_i) = \sigma_i \mid E(i-1) \wedge g(w_1) = \bar{g}_1]$$
$$= \Pr[g(w_i) \oplus \bar{g}_1 = \sigma_1 \oplus \sigma_\alpha \oplus \sigma_\beta \oplus \cdots \oplus \sigma_i \mid E(i-1) \wedge g(w_1) = \bar{g}_1].$$

By Claim 4, w_i is not a left coordinate for any node with breadth-first order less than i. Hence, the event $g(w_i) \oplus \bar{g}_1 = \sigma_1 \oplus \sigma_\alpha \oplus \cdots \oplus \sigma_i$ is independent of the event $E(i-1) \wedge g(w_1) = \bar{g}_1$ and thus the above probability is 2^{-n}.

The analysis is similar in the case that (w_i, w'_i) has even depth.

■

Before completing the main theorem, let us state the following corollary of Lemma 2.

Corollary 2 *The outputs of a butterfly transformation are k-wise independent iff the dependency graph of the inputs contains no alternating cycles of length k.*

Proof: If there is an alternating cycle of length k in the inputs then the corresponding outputs sum to zero as before. To show the converse consider any k outputs and their corresponding inputs. Apply Lemma 2 to dependency graph induced by these input nodes. There is no alternating cycle in this dependency graph by hypothesis, and hence all k outputs are independent. ■

Let us now return to the proof of the main theorem. By Lemma 2, if the dependency graph of the outputs of the first butterfly $(w_1, w_1'), \ldots, (w_q, w_q')$ does not contain an alternating cycle, then the outputs of the second butterfly are uniform and independent. Hence, the advantage of an adversary is bounded from above by the probability that the dependency graph of $(w_1, w_1'), \ldots, (w_q, w_q')$ contains an alternating cycle.

The number of possible alternating cycles of length $2j$ on q nodes is $q!/(q-2j)!j$. Hence the probability that there is an alternating cycle is

$$\leq \sum_j \frac{q!}{(q-2j)!j} 2^{-2nj} \leq \sum_j \left(\frac{q^2}{2^{2n}}\right)^j \leq \frac{q^2}{2^{2n}}.$$

assuming $q^2/2^{2n} < 1$.

This concludes the proof that the security of the Benes transformation is $\Omega(2^n)$. The proof that the security is $O(2^n)$ is easy and omitted due to lack of space.

Lemma 3 *The modified Benes transformation has security $\Omega(2^n)$.*

Proof Sketch: It can be shown that the probability that the top butterfly produces an alternating cycle of length $2j$ is $\leq 2^{-2jn}$. But we omit the details. ■

Finally, let us sketich the proof of the minimality of the modified Benes transformation.

Proof Sketch of Theorem 4. When one of the edges of the top butterfly is deleted it is easy to create a sequence of q inputs which yields an alternating cycle of length two in the intermediate variables with probability $\Theta(q^2/2^n)$.

When one of the edges of the bottom butterfly is deleted it is easy to show that either the left or right hand output repeats with probability approximately twice as large as for a random function. ■

6 Open Questions

A natural open problem is bounding the query security for the five-round Feistel transformation. A related problem is determining the number of rounds needed to achieve query security 2^n.

Unfortunately, n-bit truly random primitives are not available for values of n of interest. In such a case the actual security of the Benes or Feistel construction depends a great deal on the quality of the underlying primitive. For example, with $n = 32$, the four-round Feistel transformation with truly random functions requires approximately 2^{16} queries to distinguish from a truly random 64-bit function. However, when the truly random functions are instantiated by S-boxes, differential cryptanalysis [4] needs only 16 chosen plaintext queries (or 2^{33} known plaintext queries) for the independent key case. (See also, [13, 18].) However, additional rounds appear to add increasing amounts of security. Shedding light on how additional rounds of the Feistel or Benes transformation may improve security in the complexity-theoretic setting is an important open problem.

References

1. M. Bellare, R. Canetti, and H. Krawcyk, "Keying MD5 – Message Authentication via Iterated Pseudorandomness," manuscript.
2. M. Bellare, J. Kilian, and P. Rogaway, "The Security of Cipher Block Chaining," *Advances in Cryptology–Crypto '94*, Springer Verlag (1994).
3. E. Biham and A. Shamir, *Differential Cryptanalysis of the Data Encryption Standard,* Springer Verlag (1993).
4. E. Biham and A. Shamir, "Differential Cryptanalysis of Snefru, Khafre, REDOC II, LOKI, Lucifer," *Advances in Cryptology–Crypto '91,* Springer Verlag (1992).
5. E. Biham and A. Shamir, "Differential Cryptanalysis of Feal and N-hash", *Advances in Cryptology–Eurocrypt '91,* Springer Verlag (1991).
6. M. Blum and S. Micali, "How to Generate Cryptographically Strong Sequences of Pseudorandom Bits," *SIAM Journal on Computing,* **13** pp. 850-864 (1984).
7. D. Coppersmith, "Another Birthday Attack," *Advances in Cryptology–Crypto '85,* Springer Verlag (1986).
8. I. Damgard, "A Design Principle for Hash Functions," *Advances in Cryptology–Crypto '89,* Springer Verlag (1989).
9. D. Davies and W. Price, *Security for Computer Networks (2e),* John Wiley (1989).
10. H. Dobbertin, "Cryptanalysis of MD4," To appear at the Fast Software Encryption Workshop, February, 1996.
11. O. Goldreich, S. Goldwasser, and S. Micali, "How To Construct Random Functions," *JACM,* **33**, 4, pp. 792-807 (1986).
12. J. Hastad, R. Impagliazzo, L. Levin, and M. Luby, "Pseudorandom Generation From One-Way Functions," *Proc. ACM Symp. on Theory of Computing,* (1989); "Pseudorandom Generators Under Uniform Assumptions," *Proc. of the ACM Symp. on Theory of Computing* (1990)
13. S. Langford and M. Hellman, "Differential-Linear Cryptanalysis," *Advances in Cryptology–Crypto '94,* Springer Verlag (1994).

14. L. Levin, "One-Way Functions and Pseudo-Random Generators," *Proc. of the ACM Symp. on Theory of Computing* (1985).
15. M. Luby, *Pseudorandomness And Its Cryptographic Applications*, Princeton Univ. Press, to appear.
16. M. Luby and C. Rackoff, "How to Construct Pseudorandom Permutations from Pseudorandom Functions," *SIAM Journal on Computing*, **17**, 373-386 (1988).
17. U. Maurer, "A Simplified and Generalized Treatment of Luby-Rackoff Pseudo-Random Permutation Generators," *Advances in Cryptology–Eurocrypt '92*, Springer Verlag (1992).
18. M. Matsui, "The First Experimental Cryptanalysis of the Data Encryption Standard," *Advances in Cryptology–Crypto '94*, Springer Verlag (1994).
19. R. Merkle, "A Fast in Software One-Way Hash Function," *Journal of Cryptology*, **3**, 1, pp. 43-58 (1990).
20. R. Merkle, "One-Way Hash Functions and DES," *Advances in Cryptology–Crypto '89*, Springer Verlag (1989).
21. P. van Oorschot and M. Wiener, "Parallel Collision Search with Application to Hash Functions and Discrete Logarithms," *Proc. of the 2nd ACM Conf. on Computer and Communications Security*, (1994).
22. B. Preneel, *Analysis and Design of Cryptographic Hash Functions*, Ph.D Thesis, Katholieke Universiteit Leuven (1993).
23. B. Preneel, and P. van Oorschot, "MDx-MAC and Building Fast MACs from Hash Functions," *Advances in Cryptology–Crypto '95*, Springer Verlag (1995).
24. V. Shoup, personal communication (1995).
25. A. Wyner, "The Wire-Tap Channel," *Bell System Technical Journal*, **54** (1975).
26. A. Yao, "Theory and Applications of Trapdoor Functions," *Proc. of the IEEE Symp. on Foundations of Computer Science*, (1982).
27. Y. Zheng, T. Matsumoto, and H. Imai, "On the Construction of Block Ciphers Provably Secure and Not Relying on Any Unproved Hypotheses," *Advances in Cryptology–Crypto '89* (1989).