Generalized Birthday Attacks on Unbalanced Feistel Networks

Charanjit S. Jutla

IBM T. J. Watson Research Center,
Yorktown Heights, NY 10598, USA

Abstract. Unbalanced Feistel networks F_k which are used to construct invertible pseudo-random permutations from kn bits to kn bits using d pseudo-random functions from n bits to $(k-1)n$ bits, $k \geq 2$ are studied. We show a new generalized birthday attack on F_k with $d \leq 3k-3$. With $2^{(k-1)n}$ chosen plaintexts an adversary can distinguish F_k (with $d = 3k-3$) from a random permutation with high probability. If $d < (3k-3)$ then fewer plaintexts are required. We also show that for any F_k (with $d = 2k$), any adversary with m chosen plaintext oracle queries, has probability $O(m^k/2^{(k-1)n})$ of distinguishing F_k from a random permutation.

Keywords: Block ciphers, Feistel networks, pseudo-random permutations, second moment method, birthday attacks.

1 Introduction

We study the security of unbalanced Feistel networks [12]. In particular, we demonstrate a new class of attacks based on generalizations of the birthday paradox. Feistel networks are used to construct pseudo-random permutations ($2n$ bits to $2n$ bits) from pseudo-random functions (n bits to n bits). *Unbalanced Feistel networks* are also used to construct pseudo-random permutations, but from pseudo-random functions in which the range and domain of the functions may not be of the same size.

Unbalanced Feistel networks in which the size of the domain of the pseudo-random functions is larger than that of the range will be called *contracting unbalanced Feistel networks.* The pseudo-random functions used in the construction will be called *contracting substitution boxes.* Similarly, networks in which the size of the domain of the pseudo-random functions is smaller than that of the range will be called *expanding unbalanced Feistel networks.* The pseudo-random functions used in the construction will be called *expanding substitution boxes.* Such Feistel networks are also called *complete target heavy unbalanced Feistel networks* [12]. BEAR and LION [11] are two block ciphers which employ both expanding and contracting unbalanced Feistel networks.

In this paper we will be concerned with expanding unbalanced Feistel networks. From a practical point of view, expanding unbalanced Feistel networks

are easier to devise. For if the substitution boxes were to be given explicitly (i.e. by giving the value of the function explicitly for each input) the expanding boxes require much less memory. More precisely, a function from n bits to kn bits requires $2^n kn$ bits of memory, whereas a function from kn bits to n bits requires $2^{kn} n$ bits of memory. A similar information-theoretic argument can be made if the substitution boxes were not given explicitly, but were themselves constructed using smaller boxes or functions.

Naor and Reingold [13] have studied the security of contracting unbalanced Feistel networks. They show much better security (lower) bounds for such networks compared to the bounds proved for usual Feistel networks. Proving comparable bounds for usual Feistel networks is much more difficult. This disparity is apparently due to the information-theoretic distinction mentioned in the previous paragraph. Proving security (lower) bounds for expanding Feistel networks turns out to be even more difficult.

If L and R are bit strings, then let $L||R$ denote their concatenation. For $k \geq 2$, an expanding Feistel network is a permutation $F_k : \{0,1\}^{kn} \rightarrow \{0,1\}^{kn}$, given by composition of several subrounds of the following transformation:

$(L_1||L_2||...||L_k) \rightarrow ((L_1||(f(L1) \oplus (L_2||...||L_k))) <<< n)$.

Here, L_i is a n bit string, and f a random function from n bits to $(k-1)n$ bits. The functions used in different subrounds may be independent. $X <<< n$ denotes X rotated left by n bits. An F_k^r $(r \geq 1)$ expanding Feistel network has r subrounds. For example, DES [3] is a F_2^{16} Feistel network.

We show that for any F_k^{2k} expanding Feistel network F, with independent random functions for different subrounds, any adversary with m chosen plaintext oracle queries (i.e. values of $F(x)$ for m chosen values of x), has probability $O(m^k/2^{(k-1)n})$ of distinguishing F from a random permutation $\{0,1\}^{kn} \rightarrow \{0,1\}^{kn}$. For, $k = 2$ (i.e. for the usual Feistel networks) the result was already known [9]. In fact, the bound in [9] held for just 3 sub-rounds. Recently, Patarin [10] has shown that for F_2^6, the adversary has distinguishing probability at most $O(m^4/2^{3n} + m^2/2^{2n})$.

We also show that as k increases more and more subrounds of F_k can be broken by chosen plaintext attacks. We show that F_k^{3k-3} can be distinguished from a random permutation with about $2^{(k-1)n}$ chosen plaintexts. For example, F_4^9 can be distinguished from a random permutation with 2^{3n} chosen plaintexts. Various other such attacks can be obtained; the number of plaintexts required increasing with the number of subrounds. Some attacks lead to complete recovery of the key (or the substitution boxes, in case the substitution boxes were key dependent).

These attacks are based on a new technique employing certain generalizations of the birthday paradox. Usual birthday-like attacks (see e.g. [1],[8],[4],[6]) are based on requiring two random variables involved to be same. Usually, two such "coincidence" events are pairwise independent. Generalizations to more than one coincidence have been studied in [6],[5],[7]. In this paper we study coincidences which are much more dependent than previously considered. We employ the second moment method for our analysis.

We also note that if the exclusive-or operation above in the subround definition is replaced by an addition (modulo 2^n) operation, then these attacks do not work. However, if only some of the subrounds use the addition operation, the attacks are still possible.

2 Definitions

Let $\{0,1\}^n$ denote all n bit strings. If $x \in \{0,1\}^n$ then let x_i denote the ith bit of x. Let $x <<< t$ denote the string obtained by rotating x left by t bits. If x and y are two bit strings then $x||y$ will denote their concatenation. Thus, if x and y are n bit strings, then $(x||y) <<< n$ is $(y||x)$.

Let $\mathcal{F}^{n,m}$ denote the class of all functions $F : \{0,1\}^n \rightarrow \{0,1\}^m$. A *random function* $F^{n,m}$ is a function chosen uniformly from $\mathcal{F}^{n,m}$. If the function is one-to-one and $n = m$, then we call such a function a *permutation*. When n, m are clear from context we drop the superscripts.

An *adversary* S is a function from bit strings to bit strings. Note that we are not defining complexity theoretic adversaries, as we will not concern ourselves to pseudo-randomness due to lack of space. Whenever the results can be generalized to pseudo-random functions, we will mention so.

An *oracle adversary* is a function $S \in \mathcal{F}^{r,t}$ which takes as parameter another adversary $A \in \mathcal{F}^{n1,n2}$ (A is called the *oracle*). However, S may not have full access to A, and may only be allowed certain invocations of A (called oracle queries). Thus, if the oracle adversary is allowed to make m oracle queries, then $S(x)$ depends on $(P_1, A(P_1), ..., P_i, A(P_i), ..., P_m, A(P_m))$. Here $P_i \in \{0,1\}^{n1}$ and $A(P_i) \in \{0,1\}^{n2}$. P_1 is determined by x, and P_{i+1} is determined by $P_1, A(P_1),..., A(P_i)$. Also, when A is the oracle, we will refer to the oracle adversary as S^A.

For ($k \geq 2$), we define the *operator* $\mathcal{H}_k$ (or just $\mathcal{H}$ if k is understood) applied to a function $f_1 \in \mathcal{F}^{n,(k-1)n}$ to be a permutation $\mathcal{H}_k[f_1] \in \mathcal{F}^{kn,kn}$ as follows:
For all $z = z_1||...||z_k$, $z_i \in \{0,1\}^n$,
$\mathcal{H}_k[f_1](z) = (z_1||(f_1(z_1) \oplus (z_2||...||z_k))) <<< n$

Sometimes it is convenient to view f_1 as $(k-1)$ functions $f_{11}, f_{12}, ...f_{1(k-1)}$, where $f_{11}(z)$ is defined as the restriction of $f_1(z)$ to the first n bits and so on (see Fig 1). Then, the above definition of $\mathcal{H}_k[f_1]$ (in the case k=4) can be rewritten as:

$\mathcal{H}_4[f_1](z) = (f_{11}(z_1) \oplus z_2||f_{12}(z_1) \oplus z_3||f_{13}(z_1) \oplus z_4||z_1)$

The above transformation $\mathcal{H}_k[f_1]$ is called one subround of the *expanding unbalanced Feistel network*. The function f_1 is called the *expanding substitution box*. In a block cipher, this function can either be a fixed function xored with a secret key (as is the case in DES), or a function generated from a secret key.

It is not difficult to see that $\mathcal{H}_k[f_1]$ is a permutation. For $d \geq 2$, the d-subround transformation is defined recursively:
$\mathcal{H}[f_1, f_2, ...f_d](z) = \mathcal{H}[f_d](\mathcal{H}[f_1, ..f_{d-1}](z))$.

A typical permutation $\mathcal{H}_k[f_1, ...f_d]$ will be denoted F_k^d. Figure 1 shows a nine subround Feistel permutation F_4^9.

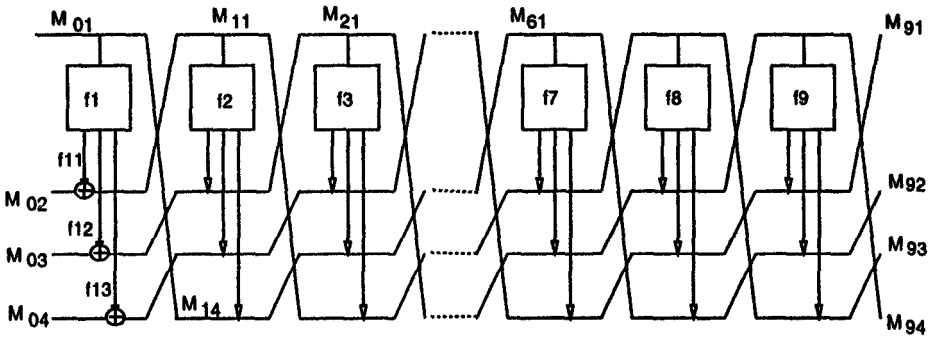

Fig. 1. An unbalanced Feistel network F_4^9

Let S be an oracle adversary which outputs one bit and makes at most m oracle queries. The *distinguishing probability* of S for the operator $\mathcal{H}_k$ composed d times is
$|Pr_{f_1,f_2,...f_d}[S^{\mathcal{H}[f_1,f_2,...f_d]} = 1] - Pr_F[S^F = 1]|$,
where $f_1, f_2, ...f_d$ are uniformly chosen from $\mathcal{F}^{n,(k-1)n}$, and F is uniformly chosen from all permutations in $\mathcal{F}^{kn,kn}$.

Probability Theory Basics

For a random variable X its variance is defined as
$var[x] = E[(x - E[x])^2]$

If $X = X_1 + ... + X_i + ... + X_m$, where each X_i is a indicator random variable for certain event A_i (i.e. $X_i = 1$ if event A_i holds and $X_i = 0$ otherwise), then $var[X] \leq E[X] + \Sigma_{i \neq j} cov[X_i, X_j]$, where $cov[Y, Z] = E[YZ] - E[Y]E[Z]$.

For indices i, j, write $i \sim j$ if $i \neq j$ and the events A_i, A_j are not independent. Denote by Δ the following sum

$$\Delta = \sum_{i \sim j} Pr[A_i \wedge A_j]$$

Using Chebyschev's inequality the following lemma can be proved (see e.g. [2])
Lemma 1: If $E[X] \to \infty$ and $var[X] = o(E[X]^2)$, then $X > 0$ almost always. It follows that if $E[X] \to \infty$ and $\Delta = o(E[X]^2)$, then $X > 0$ almost always.

This is also called the *second moment method.*

The *birthday paradox* refers to the fact that if r objects are selected with replacement from a set of N objects, then two of the objects chosen will be same with probability tending to $1 - exp(-r^2/(2N))$.

3 Generalized Birthday Attacks

Let F_k^d be a d subround unbalanced Feistel network employing random subround functions $f_1, ...f_d$ (i.e. $F_k^d = \mathcal{H}[f_1, ...f_d]$). We show that for $d = 3k-3$ subrounds, with about $2^{(k-1)n}$ chosen plaintexts one can distinguish F_k^d from a random permutation in $\mathcal{F}^{kn,kn}$ with probability close to one.

The total time required is $O(n2^{(k-1)n})$, and memory required is $O(2^{(k-1)n})$.

For simplicity, we demonstrate the attack for F_4^9 (see Fig 1). From now on we will refer to the Feistel permutation as F.

For input $M = (M_{01}||M_{02}||M_{03}||M_{04})$, we use the following notation to denote intermediate values after each subround of F. After round j the intermediate value will be denoted $(M_{j1}||M_{j2}||M_{j3}||M_{j4})$. Recall that

$(M_{(j+1)1}||M_{(j+1)2}||M_{(j+1)3}||M_{(j+1)4}) = (f_{(j+1)1}(M_{j1}) \oplus M_{j2}||\ f_{(j+1)2}(M_{j1}) \oplus M_{j3}||\ f_{(j+1)3}(M_{j1}) \oplus M_{j4}||M_{j1})$.

Of particular interest are the intermediate values M_{j1}, as it is to this value that the next subround function is applied.

We start with a *pack* of s chosen plaintexts with the same first n bits, i.e. keeping M_{01} constant. We will have a total of t different such packs (i.e. each pack having its distinguishing M_{01} value). Let M^1 and M^3 be two plaintexts in two different packs (wlog pack1 and pack2 respectively) such that

$$M_{11}^1 = M_{11}^3 \tag{1}$$

If M^2 and M^4 are plaintexts in pack1 and pack2 respectively such that

$$M_{02}^1 \oplus M_{02}^2 = M_{02}^3 \oplus M_{02}^4 \tag{2}$$

then it follows that

$$M_{11}^2 = M_{11}^4 \tag{3}$$

If we also require that

$$M_{03}^1 \oplus M_{03}^2 = M_{03}^3 \oplus M_{03}^4, \text{ and } M_{04}^1 \oplus M_{04}^2 = M_{04}^3 \oplus M_{04}^4 \tag{4}$$

then more good properties will follow. If,

$$M_{21}^1 = M_{21}^2 \tag{5}$$

then using the earlier equations it follows that

$$M_{21}^3 = M_{21}^4 \tag{6}$$

To see (6), note that $M_{21} = f_{21}(M_{11}) \oplus f_{12}(M_{01}) \oplus M_{03}$. If we take the xor-sum of four of these equations (corresponding to the four texts involved), we get

$$M_{21}^1 \oplus M_{21}^2 \oplus M_{21}^3 \oplus M_{21}^4 = 0 \tag{7}$$

The RHS sums to zero because of (1), (3), (4), and $M_{01}^1 = M_{01}^2$, $M_{01}^3 = M_{01}^4$. Then, (6) follows from (5) and (7).

Since the underlying subround functions are random, equation (1) holds with probability 2^{-n}. Similarly, equation (5) holds with probability 2^{-n}. It is not difficult to prove that (5) is independent of (1) and hence the combined probability of (1),(3),(5) and (6) is 2^{-2n}.

Figure 2 illustrates these and many more equalities which the adversary would require. The columns represent a particular plaintext. The rows represent the intermediate values. Two points are joined by an edge if the two values are

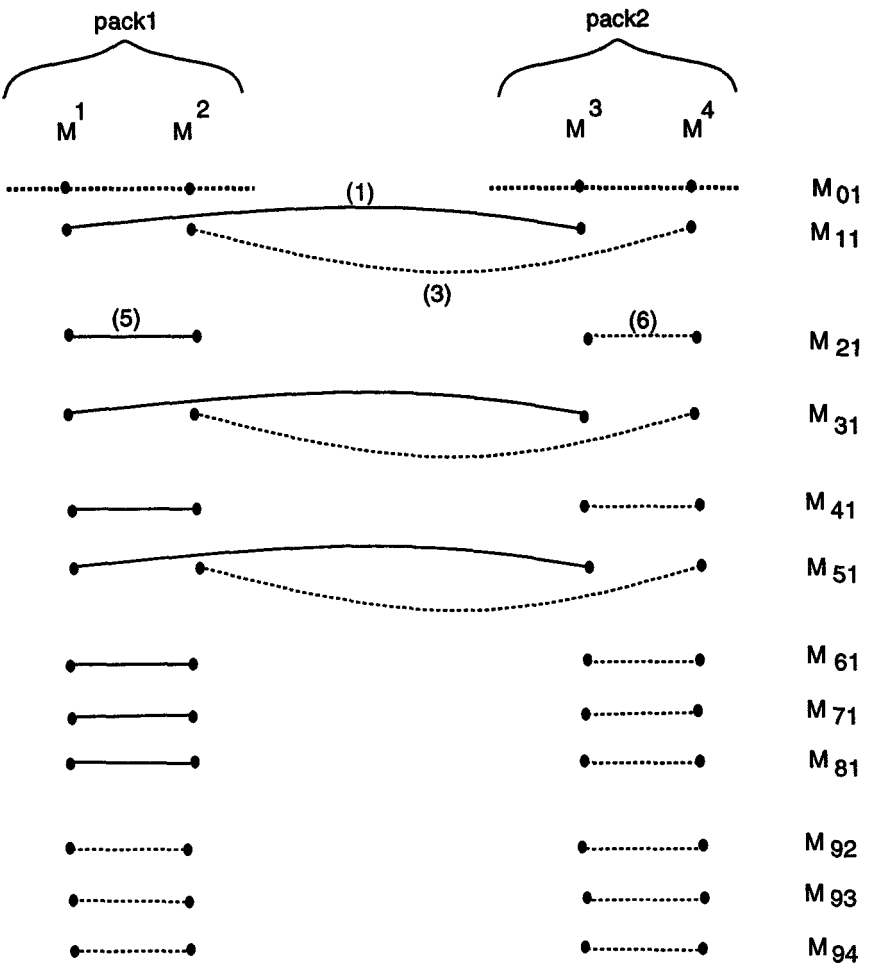

Fig. 2. Attack on F_4^9

equal. It is a solid edge if the equality happens with probability 2^{-n}. It is a dotted edge if it follows conditionally with probability one (or if it is required by the adversary as part of the plaintext conditions). Edges between different packs like the ones between M_{11}^1 and M_{11}^3 will be called *cross edges.*

Let us restrict our attention to just the plaintexts M^1 and M^2. The combined probability required by the equations in Figure 2 is 2^{-5n} (it is not difficult to see that the events are independent, as the subround functions are independent). Let X_i be the indicator random variable of such an event. Let X be the sum of all $\Theta(s^2t)$ such random variables, as there are t packs, and each pack has s texts. The expected number of such events ($E[X]$) is $\Theta(s^2t2^{-5n})$. We next show that $var[X] = o(E[X]^2)$. We just show that $\Delta = o(E[X]^2)$.

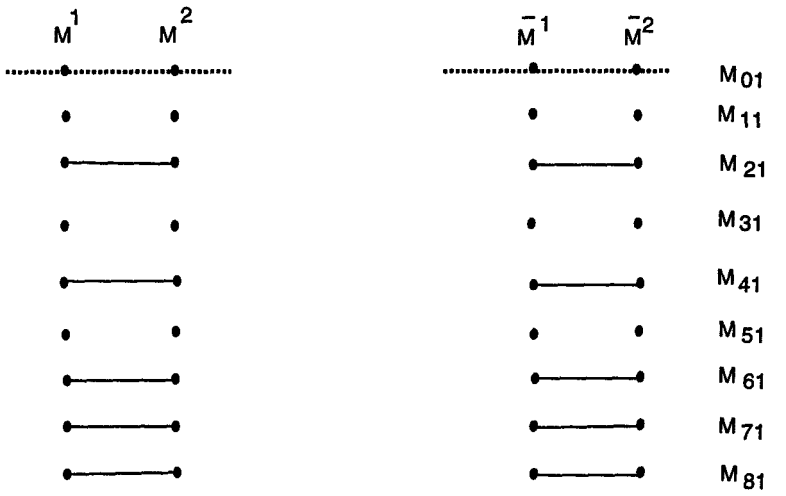

Fig. 3. Two events

Let X_i and X_j be the indicator variables of two such events. Let us denote the plaintexts involved in the event $(X_i = 1)$ by M^1 and M^2, and the plaintexts involved in the event $(X_j = 1)$ by $\bar{M}^1$ and $\bar{M}^2$. The subevents in the two events are shown in Figure 3 by solid edges. Here, the two packs could be the same pack or different packs. If there are no cross edges then the combined probability of the two events is clearly 2^{-10n}. Also, if $\{M^1, M^2\} \cap \{\bar{M}^1, \bar{M}^2\} \neq \phi$, then the combined probability of the two events remains 2^{-10n}. It can be shown that some of the solid edges can be made dotted edges by requiring two edges in each of the previous few subrounds (as in fig 2). For example, the edge between $\bar{M}^1_{81}$ and $\bar{M}^2_{81}$ can be made a dotted edge (i.e. conditional probability one) if there are two edges in each of the rows M_{71}, M_{61}, and M_{51}, and $M^1_{41} \oplus M^2_{41} \oplus \bar{M}^1_{41} \oplus \bar{M}^2_{41} = 0$. Moreover, this is the only way to make the edges dependent on other edges.

Thus the probability of the event depicted in Figure 3 may be higher if conditions as in Figure 2 hold (i.e. cross edges in the rows M_{51}, M_{31}, M_{11}, and additional requirements on the plaintexts; or some such similar set of conditions). With additional requirements on the plaintexts, the combined probability can indeed be higher. There are a few cases:

1. $\bar{M}^1 \oplus \bar{M}^2 = M^1 \oplus M^2$: For $\Theta(s^3t^2)$ such pair of events, the combined probability is at most $3 * 2^{-8n}$. This follows because by requiring a solid cross edge in each of the rows M_{11}, M_{31} and M_{51}, five solid edges among the columns $\bar{M}^1$ and $\bar{M}^2$ can be made dotted edges. The factor three comes from the different ways in which the cross-edges are organised.
2. $\bar{M}^1_{02} \oplus \bar{M}^2_{02} = M^1_{02} \oplus M^2_{02}$, and a similar relation for M_{04}: For $\Theta(s^3 2^n t^2)$ such pair of events, the combined probabilty remains 2^{-10n}. It may seem that by requiring a solid cross edge in each of the rows M_{11}, M_{31} and M_{51}, four solid

edges among the columns $\bar{M}^1$ and $\bar{M}^2$ can be made dotted edges (in the rows M_{41}, M_{61}, M_{71}, M_{81}). But under such conditions, $\bar{M}^1_{21} \neq \bar{M}^2_{21}$ since $\bar{M}^1_{03} \oplus \bar{M}^2_{03} \neq M^1_{03} \oplus M^2_{03}$.

3. As in the previous case, with relations only among M_{02} and M_{03}, the two events remain independent.

Thus, as long as $s \approx 2^{3n}$, it follows that $\Delta = o(E[X]^2)$. Thus, if $s = 2^{3n}$, $X > 0$ with high probability (see Lemma 1).

We now turn our attention to the whole event in Figure 2. Again, let X_i be the indicator variable of such an event. Let X be the sum of all $\Theta(s^3t^2)$ such random variables. Note that we only have s^3t^2 and not s^4t^4, as (2) and (4) have completely determined M^4. Once again, $E[X] = \Theta(s^3t^2 2^{-8n})$. Once again we have to show that $\Delta = o(E[X]^2)$. This is proved in a fashion similar to one mentioned in the previous two paragraphs. Thus, if $s = 2^{3n}$, such an event will happen with high probability.

How does the adversary detect such an event? The four ciphertexts satisfy seven equations. Six of these are depicted by dotted edges in rows corresponding to M_{92}, M_{93}, M_{94}. Further, the following relation holds

$$M^1_{91} \oplus M^2_{91} \oplus M^3_{91} \oplus M^4_{91} = 0 \tag{8}$$

Thus the expected number of such events (noise) occurring in a random permutation over the plaintexts as supplied by the adversary is $\Theta(s^3t^2 2^{-7n})$. Thus, there is more noise than signal. However, this situation can be improved by adding more packs to Figure 2 (see Figure 4). Note that addition of each pack adds a factor of $\Theta(st2^{-3n})$ to the expected number of events the adversary is interested in (essentially the cross edges are the only edges which will appear as additional solid edges). Thus, with $s = 2^{3n}$ the probability of occurrence of such an event remains close to one. The factor added in the expected value of the noise is $O(st2^{-4n})$ (2^{-3n} for the three edges in M_{92}, M_{93}, and M_{94}, and another 2^{-n} for the relation among M_{91} similar to (8)). Thus, if there are a total of five packs involved, as shown in figure 4, the expected value of noise becomes $O(s^6t^5 2^{-19n})$. If the adversary keeps s close to 2^{3n} and t a small constant, noise becomes insignificant. Hence the adversary will be able to distinguish F_4^9 from a random permutation with high probability.

Note that the adversary is also able to infer many relations involving the subround functions, the plaintexts and ciphertexts. If the number of subrounds are decreased further, the adversary can actually obtain almost all of the subround functions.

We next look at the computational efficiency of the attack. By sorting each pack by its ciphertext values M_{92}, M_{93}, M_{94} one can determine in each pack about 2^{3n} pairs with the same M_{92}, M_{93}, M_{94} values. For each such pair i, j, compute $M^i_{91} \oplus M^j_{91}, M^i_{02} \oplus M^j_{02}, M^i_{03} \oplus M^j_{03}, M^i_{04} \oplus M^j_{04}$, and again sort them by these values. Next we look for equality of these values across five different packs out of t. The total time required is $O(n2^{3n})$, and memory required is $O(2^{3n})$.

Of course, these attacks become much more effective in terms of time, mem-

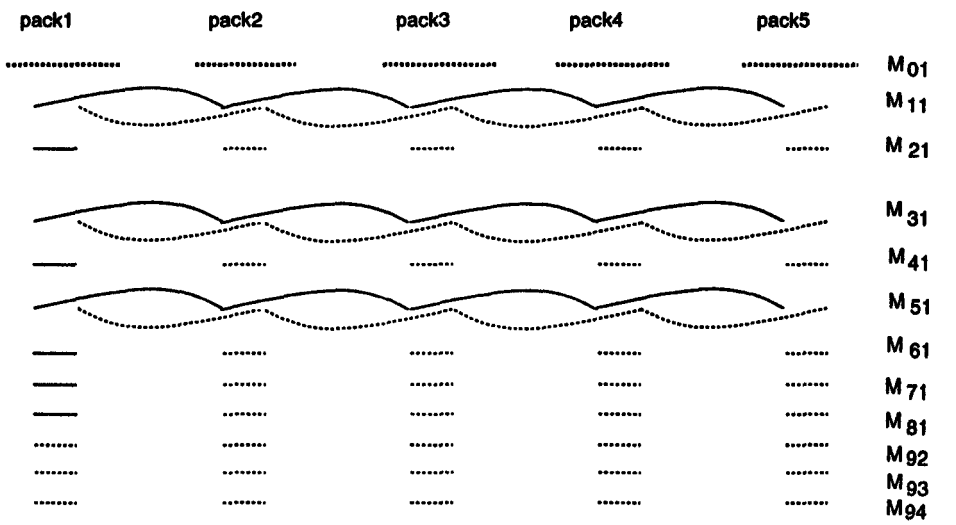

Fig. 4. A five pack attack on F_9^4

ory and number of plaintexts required when the number of subrounds are fewer.

It should be noted that if the subround functions are permutations (i.e. f_{11} etc. are permutations) then the equality in M_{81} is not possible. However, the adversary could achieve the same result by requiring in Figure 2 that the cross edges be in row M_{41} instead of row M_{31} (call this the *modified* attack). It should also be noted that if the xors are replaced by addition modulo 2^n this attack does not seem to work, as it depended on obtaining equations like (7). However e.g., if $M_{61} = f_{61}(M_{51}) \oplus (f_{52}(M_{41}) \oplus (f_{43}(M_{31}) + M_{21}))$ and all other operations remain xor, the *modified* attack still works. To see this, note that $f_{43}(M_{31}^1) + M_{21}^1 = f_{43}(M_{31}^2) + M_{21}^2$, and hence one still obtains

$M_{61}^1 \oplus M_{61}^2 \oplus M_{61}^3 \oplus M_{61}^4 = 0$

As a general rule in the case of F_k^{3k-3} (with $2^{(k-1)n}$ plaintexts), we note that the birthday paradox allows for $2(k-1)$ edges among intermediate values of M^1 and M^2. However, because of the Feistel structure, the second moment method only allows for $2(k-1)-1$ edges. The generalization presented here allows for additional $(k-1)$ cross edges. Also, the initial conditions take care of one subround, yielding an attack on $(3k-3)$ subrounds.

4 Lower bound on security of F_k^{2k}

Theorem: Let $f_1, f_2, ..., f_{2k}$ be functions chosen randomly and uniformly from $\mathcal{F}^{n,(k-1)n}$, $k \geq 2$. Let $F_k^{2k} = \mathcal{H}_k[f_1, ...f_{2k}]$ be the unbalanced Feistel permutation. Then any oracle adversary S which makes m oracle queries has probability at most $O(m^k/2^{(k-1)n})$ of distinguishing F_k^{2k} from a permutation randomly and

uniformly chosen from $\mathcal{F}^{kn,kn}$ (distinguishing probability is defined in section 2).

When $k = 2$, this theorem is related to the well known Luby-Rackoff Theorem [9]; both give a distinguishing probability bound of $O(m^2/2^n)$, but we require four rounds whereas Luby-Rackoff requires three.

We give a sketch of the proof for F_3^6, which is a generalization of the Luby-Rackoff proof (see [8]). Note that an obvious generalization of Luby-Rackoff would only yield a bound of $O(m^2/2^n)$.

Proof: We will write F for the permutation F_3^6. It suffices to show the result assuming that f_{62} is a constant function (say, $\forall i : f_{62}(i) = 0$).

Let $M_{61}^i, M_{62}^i, M_{63}^i$ represent the random variables corresponding to the output of F on oracle query i, i.e.

$(M_{61}^i || M_{62}^i || M_{63}^i) = F(M_{01}^i || M_{02}^i || M_{03}^i)$

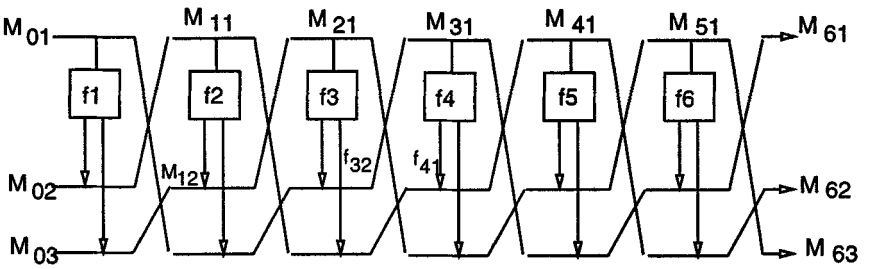

Fig. 5. An unbalanced Feistel network F_3^6

We describe two algorithms, B and C, for computing the answers to the oracle queries of S, both taking $f_1, f_2, f_3, f_4, f_5, f_6$ as input. We will denote $f_1, f_2, f_3, f_4, f_5, f_6$ by f. Recall that f is a sequence of functions chosen randomly from $\mathcal{F}^{n,(k-1)n}$, and these functions are used in different rounds of F by the operator $\mathcal{H}$. The algorithm B will define new random functions, namely $\phi_{32}, \phi_{41}, \phi_{42}, \phi_{51}, \phi_{52}, \phi_{61}$, and use them instead of $f_{32}, ..., f_{61}$ respectively. We will show that these new functions are themselves uniformly and independently distributed.

The answer of B on the i^{th} oracle call will be denoted $B^i(\mathbf{f})$ (similarly for C). The description of the algorithm B is given in Figure 6. The algorithm $C^i(\mathbf{f})$ behaves exactly like $B^i(\mathbf{f})$ except that its output is $(f_{61}(i)||f_{41}(i)||f_{51}(i))$.

Thus,

$$Pr_F[S^F = 1] = Pr_{\mathbf{f}}[S^{C(\mathbf{f})} = 1]$$

where F is uniformly chosen from all permutations in $\mathcal{F}^{kn,kn}$.

Oracle computation $B^i(\mathbf{f})$ on input $(M^i_{01}||M^i_{02}||M^i_{03})$

Let $a^i(\mathbf{f}) = min\{j \leq i : M^i_{01} = M^j_{01}\}$

$M^i_{11}(\mathbf{f}) = M^i_{02}(\mathbf{f}) \oplus f_{11}(a^i(\mathbf{f}))$, $\quad M^i_{12}(\mathbf{f}) = M^i_{03}(\mathbf{f}) \oplus f_{12}(a^i(\mathbf{f}))$

Let $b^i(\mathbf{f}) = min\{j \leq i : M^i_{11} = M^j_{11}\}$

$M^i_{21}(\mathbf{f}) = M^i_{12}(\mathbf{f}) \oplus f_{21}(b^i(\mathbf{f}))$, $\quad M^i_{22}(\mathbf{f}) = M^i_{01}(\mathbf{f}) \oplus f_{22}(b^i(\mathbf{f}))$

Let $c^i(\mathbf{f}) = min\{j \leq i : M^i_{21} = M^j_{21}\}$

$M^i_{31}(\mathbf{f}) = M^i_{22}(\mathbf{f}) \oplus f_{31}(c^i(\mathbf{f}))$

Let $d^i(\mathbf{f}) = min\{j \leq i : M^i_{31} = M^j_{31}\}$

$$\phi_{32}(\mathbf{f})(i) = \begin{cases} M^i_{11}(\mathbf{f}) \oplus f_{32}(i) & \text{if } (d^i = i) \\ M^i_{11}(\mathbf{f}) \oplus f_{32}(d^i) \oplus f_{41}(d^i) \oplus f_{41}(i) & \text{otherwise} \end{cases}$$

$M^i_{32}(\mathbf{f}) = M^i_{11}(\mathbf{f}) \oplus \phi_{32}(\mathbf{f})(c^i(\mathbf{f}))$

$$\phi_{41}(\mathbf{f})(i) = \begin{cases} M^i_{32}(\mathbf{f}) \oplus f_{41}(i) & \text{if } (d^i = i) \\ M^i_{32}(\mathbf{f}) \oplus f_{32}(i) & \text{otherwise} \end{cases}$$

$M^i_{41}(\mathbf{f}) = M^i_{32}(\mathbf{f}) \oplus \phi_{41}(\mathbf{f})(d^i(\mathbf{f}))$

Let $e^i(\mathbf{f}) = min\{j \leq i : M^i_{41} = M^j_{41}\}$

$$\phi_{42}(\mathbf{f})(i) = \begin{cases} M^i_{21}(\mathbf{f}) \oplus f_{42}(i) & \text{if } (e^i = i) \\ M^i_{21}(\mathbf{f}) \oplus f_{42}(e^i) \oplus f_{51}(e^i) \oplus f_{51}(i) & \text{otherwise} \end{cases}$$

$M^i_{42}(\mathbf{f}) = M^i_{21}(\mathbf{f}) \oplus \phi_{42}(\mathbf{f})(d^i(\mathbf{f}))$

$$\phi_{51}(\mathbf{f})(i) = \begin{cases} M^i_{42}(\mathbf{f}) \oplus f_{51}(i) & \text{if } (e^i = i) \\ M^i_{42}(\mathbf{f}) \oplus f_{42}(i) & \text{otherwise} \end{cases}$$

$M^i_{51}(\mathbf{f}) = M^i_{42}(\mathbf{f}) \oplus \phi_{51}(\mathbf{f})(e^i(\mathbf{f}))$

Let $g^i(\mathbf{f}) = min\{j \leq i : M^i_{51} = M^j_{51}\}$

$$\phi_{52}(\mathbf{f})(i) = \begin{cases} M^i_{31}(\mathbf{f}) \oplus f_{52}(i) & \text{if } (g^i = i) \\ M^i_{31}(\mathbf{f}) \oplus f_{52}(g^i) \oplus f_{61}(g^i) \oplus f_{61}(i) & \text{otherwise} \end{cases}$$

$M^i_{52}(\mathbf{f}) = M^i_{31}(\mathbf{f}) \oplus \phi_{52}(\mathbf{f})(e^i(\mathbf{f}))$

$$\phi_{61}(\mathbf{f})(i) = \begin{cases} M^i_{52}(\mathbf{f}) \oplus f_{61}(i) & \text{if } (g^i = i) \\ M^i_{52}(\mathbf{f}) \oplus f_{52}(i) & \text{otherwise} \end{cases}$$

$M^i_{61}(\mathbf{f}) = M^i_{52}(\mathbf{f}) \oplus \phi_{61}(\mathbf{f})(g^i(\mathbf{f}))$

$M^i_{62}(\mathbf{f}) = M^i_{41}(\mathbf{f})$ - since we assumed $f_{62} = 0$

Output of $B^i(\mathbf{f})$: $(M^i_{61}(\mathbf{f})||M^i_{62}(\mathbf{f})||M^i_{51}(\mathbf{f}))$

Fig. 6. Definition of algorithm B

Proposition 1:

$$Pr_{\mathbf{f}}[S^{\mathcal{H}[\mathbf{f}]} = 1] = Pr_{\mathbf{f}}[S^{B(\mathbf{f})} = 1]$$

Proof: We first show that $f_{11}, f_{12}, ..., f_{31}, \phi_{32}(\mathbf{f}), ..., \phi_{61}(\mathbf{f})$ are uniformly and independently distributed, as far as the first m inputs are concerned. By induction it can be shown that (e.g.) in the definition of $\phi_{61}(\mathbf{f})(i)$, the random variable $f_{61}(i)$ (random variable $f_{52}(i)$) has never been used before in the definition of

any ϕ, if $g^i = i$ (resp. $g^i \neq i$). Even though $\phi_{61}(\mathbf{f})(i)$ is never used by B (while computing M_{61}) if $g^i \neq i$, it is important to define it so that ϕ_{61} turns out to be uniformly and independently distributed.

Then the above claim follows as the role of these new functions in B is the same as that of the old functions in $\mathcal{H}$. □

We next show that

$$Pr_{\mathbf{f}}[S^{B(\mathbf{f})} \neq S^{C(\mathbf{f})}] \leq m^3/2^{2n}$$

from which the theorem follows.

We say that $\mathbf{f}$ is *preserving* if with adversary using C as oracle, for all $i \in [1..m]$ the following three conditions hold (from now on we will drop $\mathbf{f}$ whenever it is clear from context)

1. $(g^i = i)$ or $((e^i = i)$ and $(e^{g^i} = g^i))$
2. $(e^i = i)$ or $((d^i = i)$ and $(d^{e^i} = e^i))$
3. $(d^i = i)$ or $((c^i = i)$ and $(c^{d^i} = d^i))$

Proposition 2: If $\mathbf{f}$ is preserving then $S^{B(\mathbf{f})} = S^{C(\mathbf{f})}$.

Proof: If $\mathbf{f}$ is preserving, we show that $M^i_{61} = f_{61}(i)$.

If $g^i = i$, then $M^i_{61} = f_{61}(i)$. Otherwise, $M^i_{61} = M^i_{52} \oplus \phi_{61}(g^i)$. Now, since $g^{g^i} = g^i$, $\phi_{61}(g^i) = M^{g^i}_{52} \oplus f_{61}(g^i)$. Moreover $e^i = i$, and $e^{g^i} = g^i$ because $\mathbf{f}$ is preserving. Thus,

$$\begin{aligned} M^i_{52} \oplus M^{g^i}_{52} &= (M^i_{31} \oplus \phi_{52}(i)) \oplus (M^{g^i}_{31} \oplus \phi_{52}(g^i)) \\ &= (f_{52}(g^i) \oplus f_{61}(g^i) \oplus f_{61}(i)) \oplus (f_{52}(g^i)) \\ &= f_{61}(i) \oplus f_{61}(g^i) \end{aligned}$$

It follows that $M^i_{61} = f_{61}(i)$.

Similarly, it can be shown that $M^i_{51} = f_{51}(i)$, and $M^i_{41} = f_{41}(i)$.

The proposition then follows by a simple induction, as the internal computations of B and C are the same. □

It remains to bound the probability of $\mathbf{f}$ not being preserving. Since the output of C is independent of $f_{11}, f_{12}, f_{21}, f_{22}, f_{31}, f_{32}$, under the use of oracle C, all query-inputs used by S are also independent of $f_{11}, f_{12}, f_{21}, f_{22}, f_{31}, f_{32}$. (4)

We first bound the probability ($p1$) of $\exists i \in [1..m]$ not $((d^i = i)$ or $((c^i = i)$ and $(c^{d^i} = d^i)))$. Without loss of generality, assume that no two oracle query-inputs are same.

We calculate the probability of
(i) $(d^i = j)$ and $(c^i = k)$, $i > j$, $i > k$,
(ii) $(d^i = j)$ and $(c^j = k)$, $i > j > k$.

Now, $c^i = k$ implies $M^i_{21} = M^k_{21}$, which is same as $M^i_{03} \oplus f_{12}(a^i) \oplus f_{21}(b^i) = M^k_{03} \oplus f_{12}(a^k) \oplus f_{21}(b^k)$. If $a^i = a^k$, and $b^i = b^k$ then $M^i_{03} = M^k_{03}$ is not possible, and hence $c^i \neq k$. If either $a^i \neq a^k$, or $b^i \neq b^k$, then by (4) the equality holds with probability 2^{-n}.

Similarly, $d^i = j$ implies $M^i_{01} \oplus f_{22}(b^i) \oplus f_{31}(c^i) = M^j_{01} \oplus f_{22}(b^j) \oplus f_{31}(c^j)$. Again, if either $b^i \neq b^j$, or $c^i \neq c^j$, then by (4) the equality holds with probability 2^{-n}. By (4) this event is also independent of $c^i = k$.

Thus the probability of (i) is at most 2^{-2n}. Similarly, the probability of (ii) is at most 2^{-2n}. Thus, $p1 = O(m^3/2^{2n})$.

Next we bound the probability ($p2$) of $\exists i \in [1..m]$ not $((e^i = i)$ or $((d^i = i)$ and $(d^{e^i} = e^i)))$.

Again, we calculate the probability of
(i) $(e^i = j)$ and $(d^i = k)$, $i > j$, $i > k$,
(ii) $(e^i = j)$ and $(d^j = k)$, $i > j > k$.

We first assume that for all $i \in [1..m]$ condition (3) holds. Then, just as in the proof of proposition 2, it can be shown that $M^i_{41} = f_{41}(i)$. Thus, the probability of $e^i = j$ is 2^{-n}. We already know the bound on probability of $d^i = k$, from the previous step. Moreover, $e^i = j$ is also independent of $d^i = k$. Thus, given condition (3) for all i, $p2$ is bounded by $O(m^3/2^{2n})$.

The probability ($p3$) of $\exists i \in [1..m]$ not $((g^i = i)$ or $((e^i = i)$ and $(e^{g^i} = g^i)))$ (given that (2) and (3) hold for all i) is again bounded by $(O(m^3/2^{2n})$. □

5 Conclusion

In this paper we have initiated the study of expanding unbalanced Feistel networks. However, further research is required to better our understanding of these and other such networks.

In particular, there seems to be scope for further improvement in the security lower bounds for the expanding Feistel networks. We conjecture that any adversary which distinguishes F^{2k}_k from a random permutation using chosen plaintext attacks requires $\Omega(2^{(k-1)n/2})$ chosen plaintexts. Since the attacks shown on unbalanced Feistel networks F_k work only for $3k - 3$ and fewer subrounds, the natural question arises as to the applicability of these or similar approaches to more subrounds.

Another interesting problem is to use differential characteristics in these attacks, especially if the characteristics are uniform in nature. In a similar vein, networks in which the xor operations are replaced by modular addition, or other invertible operations (e.g. data dependent rotation) need to be studied.

6 Acknowledgments

The author would like to thank Don Coppersmith for carefully reading the paper, and for several helpful suggestions. The author would also like to thank Pankaj Rohatgi for helpful discussions.

References

1. W. Aiello, R. Venkatesan, *Foiling birthday attacks in length-doubling transformations*, Eurocrypt 1996, LNCS 1070.
2. N. Alon, J.H. Spencer, *The probabilistic method*, John Wiley and Sons, 1992.
3. FIPS 46, *Data Encryption Standard*, Federal Information Processing Standards Publication 46, U.S. Department of Commerce/N.I.S.T., National Technical Information Service, Springfield, Virginia, 1977.
4. D. Coppersmith, *Another Birthday attack*, Advances in Cryptology, Crypto 1985.
5. D. Coppersmith, *Luby-Rackoff: Four rounds is not enough*, IBM Research Report, RC20674, Dec. 96.
6. M. Girault, R. Cohen, M. Campana, *A Generalized birthday attack*, Eurocrypt 1988, LNCS 330.
7. L. Knudsen, X. Lai, B. Preneel, *Attacks on fast double block length hash functions*, J. of Cryptology, 1998, 11:59-72.
8. M. Luby, *Pseudorandomness and cryptographic applications*, Princeton University Press, 1996.
9. M. Luby and C.Rackoff, *How to construct pseudorandom permutations from pseudorandom functions*, SIAM J.of Comp., 17, pp.373–386, 1988.
10. J. Patarin, *About Feistel Schemes with Six (or More) Rounds*, Proc. Fast Software Encryption, March 1998.
11. R. Anderson, E. Biham, *Two Practical and Provably Secure Block Ciphers: BEAR and LION*, 1996 Workshop on Fast Software Encryption.
12. B. Schneier, J. Kelsey, *Unbalanced Feistel Networks and Block-Cipher Design*, Fast Software Encryption, Third International Workshop Proceedings (February 1996), Springer-Verlag, 1996, pp. 121-144.
13. Moni Naor, O. Reingold, *On the Construction of Pseudo-Random Permutations: Luby-Rackoff Revisited*, Proc. STOC 97