Unbalanced Feistel Networks and Block Cipher Design

Bruce Schneier John Kelsey,
schneier@counterpane.com jmkelsey@delphi.com
Counterpane Systems
101 East Minnehaha Parkway
Minneapolis, MN 55419
(612) 823-1098

Abstract. We examine a generalization of the concept of Feistel networks, which we call Unbalanced Feistel Networks (UFNs). Like conventional Feistel networks, UFNs consist of a series of rounds in which one part of the block operates on the rest of the block. However, in a UFN the two parts need not be of equal size. Removing this limitation on Feistel networks has interesting implications for designing ciphers secure against linear and differential attacks. We describe UFNs and a terminology for discussing their properties, present and analyze some UFN constructions, and make some initial observations about their security.

It is notable that almost all the proposed ciphers that are based on Feistel networks follow the same design construction: half the bits operate on the other half. There is no inherent reason that this should be so; as we will demonstrate, it is possible to design Feistel networks across a much wider, richer design space. In this paper, we examine the nature of the structure of Feistel-based ciphers. In particular, we examine the consequences of "unbalanced" structures in which different numbers of bits are used as input and output to the F-function in each round.

This paper is organized as follows. Section 2 reviews Feistel networks. Section 3 provides a taxonomy of Feistel networks, and places some previous Feistel-based designs within this taxonomy. Section 3 gives some general analysis of unbalanced Feistel networks in relation to linear and differential cryptanalysis. Section 4 suggests some open problems and areas for future study. An appendix shows analysis of two specific block-cipher designs in a variety of unbalanced forms.

1 Feistel Networks

A Feistel network is a general method of transforming any function (usually called an F-function) into a permutation. It was invented by Horst Feistel in his design of Lucifer [Fei73], and has been used in many block cipher designs since then: DES [NBS77], FEAL [SM88], GOST [GOST89], Khufu and Khafre [Mer91], LOKI [BPS93], CAST [AT93], Blowfish [Sch94b], and RC5 [Riv95].

The fundamental building block of a Feistel network is the F-function: a key-dependent mapping of an input string onto an output string. An F-function is always nonlinear and almost always irreversible.

Definition 1. The F-function of a conventional Feistel network can be expressed as:

$$F : \{0,1\}^{n/2} \times \{0,1\}^k \rightarrow \{0,1\}^{n/2}$$

In this definition, n is the size of the block. F is a function taking $n/2$ bits and a key, k, as input, and producing an output of length $n/2$.

Definition 2. One round of a conventional Feistel network (also called a *balanced* Feistel network in this paper) is:

$$X_{i+1} = (F_{k_i}(msb_{n/2}(X_i)) \bigoplus lsb_{n/2}(X_i)) || msb_{n/2}(X_i)$$

Here, X_i is the input to the round, X_{i+1} is the output of the round, k_i is the key, n is the block length, $\mathrm{lsb}_u(x)$ and $\mathrm{msb}_u(x)$ select the least significant and most significant u bits of x respectively, $\bigoplus$ indicates modulo-2 addition, and $||$ indicates concatenation. In each round, $\mathrm{msb}_{n/2}(X_i)$ operates, via a key-dependent non-linear F-function on $\mathrm{lsb}_{n/2}(X_i)$. This is often referred to as the "left half" operating on the "right half."

Definition 3. A balanced Feistel network consists of j rounds, where:

$$X_{i+1} = (F_{k_i}(msb_{n/2}(X_i)) \bigoplus lsb_{n/2}(X_i))||msb_{n/2}(X_i)$$

The keys k_i are the round keys, which typically are output from a key schedule algorithm on input a key K.

The security of a Feistel network is based on the repeated iteration of the F-function. The number of rounds required for resistance to a given attack is dependent on the properties of the function. While there has been considerable research in determining what sorts of F-functions yield secure Feistel networks [Nyb91], [Nyb93], [OCo94a], [OCo94b], [OCo94c], [Knu94a], [Knu94b], [Nyb94], [DGV94], [NK95], little has been written about the underlying Feistel structure. The aim of our research is to generalize Feistel networks and show the implications of different structures for block-cipher design.

2 A Taxonomy of Feistel Networks

One of the reasons for a lack of research in the underlying structure of Feistel networks is a lack of language to describe them. Here we present a taxonomy of generalized Feistel networks.

2.1 Unbalanced Feistel Networks

An Unbalanced Feistel Network (UFN) is a Feistel network where the "left half" and the "right half" are not of equal size.

Definition 4. One round of a s-on-t, or s:t, UFN is:[1]

$$X_{i+1} = (F(msb_s(X_i), k_i) \bigoplus lsb_t(X_i))||msb_s(X_i)$$

Definition 5. The $\mathrm{msb}_s(X_i)$ is called the *source block.*

Definition 6. The $\mathrm{lsb}_t(X_i)$ is called the *target block.*

UFNs where $s > t$ are called *source heavy,* and UFNs where $s < t$ are called *target heavy.* A *sub-block*, b, is the gcd of s, t, and n. The F-function is a collection of 2^k mappings of the s-bit numbers onto the t-bit numbers, or s/b sub-blocks onto t/b sub-blocks.

[1] This definition becomes more complicated for incomplete UFN constructions, as is discussed below.

2.2 Homogenous and Heterogenous UFNs

Although in most Feistel ciphers, the F-function is altered only by the round keys from round to round, there is no reason why this must be the case. In Merkle's Khufu[Mer91], for example, the F-function changes (the S-boxes change) once per "octet." In the hash functions of the MD4 family, the bitwise combining operation used in the F-function is changed several times during the operation of the compression function[Riv91].

Definition 7. A UFN is *homogenous* when the F-function function is identical in each round, except for the round keys.

Definition 8. A UFN is *heterogenous* when the F-function for different rounds is not always identical except for the round keys.

The advantage of a heterogenous UFN is that, since its internal properties change from round to round, it may be much more difficult to find any kind of characteristic that propogates well through all of the different kinds of round that appear in the cipher. However, implementing heterogenous UFNs and analyzing them is often much more complicated. Hardware implementations are generally to be cheaper, and software implementations smaller and easier to correctly code, with a homogenous UFN.

Except where otherwise stated, all formulae and discussion in the remainder of this paper refer only to homogenous UFNs.

2.3 Incomplete and Inconsistent UFNs

It is possible that not all bits in a block are used at all in every round of a UFN.

Definition 9. A UFN is *complete* when $s + t = n$, that is, when in each round every bit in the block is either part of the source block or part of the target block.

Definition 10. A UFN is *incomplete* when $s + t < n$.

Definition 11. In an incomplete UFN, the $n - s - t = z$ bits that are not part of the source block or part of the target block are called the *null block*.

It is also possible that the UFN structure changes from round to round or from cycle to cycle.

Definition 12. A UFN is *consistent* when s, t, z, and n remain constant for the entire cipher.

Definition 13. A UFN in which the sizes of the source and target block change during encryption is called *inconsistent*.

Note that an inconsistent UFN is always heterogenous, but it is possible for a heterogenous UFN to be consistent.

Unless otherwise noted, discussion and formulae in this paper refer only to complete, consistent, homogenous UFN structures. Note that many of the concepts discussed below (such as rates of confusion and diffusion, cycle length, etc.) are made significantly more complex by the use of inconsistent structures.

2.4 Generalized UFNs

The most general case of a Feistel network simply requires that one part of the block being encrypted controls the encryption of another part of the block. For example, given some n-bit keyed reversible function, $E_k(X)$, we can define a block cipher whose rounds look like:

$$X_{i+1} = E_{k_i \bigoplus msb_s(X_i)}(lsb_t(X_i))||msb_s(X_i)$$

Further, we need not leave the source block alone in each round–we may perform some unkeyed reversible function on it, as well. Additionally, we can apply some nonreversible function on the source block to derive the key for the keyed reversible function.

For example, the hash funtions in the MD4 family use addition modulo 2^{32} as their keyed reversible function. A DES variant replaces the XOR operation with Latin squares [CDN95]. The only restriction is that the combining function must be reversible to allow decryption. Also, in SHA, 32 bits undergo an internal circular shift during the overall block shift. This is a simple reversible function.

Definition 14. One round of a $s{:}t$ n-bit Generalized Unbalanced Feistel Network (GUFN) is defined as:

$$X_{i+1} = R(E_{F(k_i, msb_s(X_i))}(lsb_t(X_i)), msb_s(X_i)).$$

Unless otherwise stated, the discussion and formulae in this paper refer to conventional UFNs, rather than the more complicated GUFNs.

2.5 Cycles and Rotations

A *cycle* is the number of rounds necessary for each bit in the block to have been part of both the source and target blocks at least once. A *rotation* is the number of rounds needed for each bit in the block to return to its starting position.

Definition 15. A cycle, C, of a $s{:}t$ UFN is

$$C = \lceil \frac{n}{\min(s,t)} \rceil.$$

Definition 16. A rotation, G, of an $s{:}t$ UFN is

$$G = \frac{n}{\gcd(s,t)}.$$

A UFN is called *even* if $C = G$; otherwise it is called *odd.* A UFN is *prime* when $G = n$; i.e. when s, t, and n are relatively prime. Most of the analyses in this paper focus on even UFNs. Note that a balanced Feistel cipher such as DES or Blowfish can be seen as a special case of even complete UFN—one with $G = C = 2$.

2.6 The Rate of Confusion

An observer who knows some information about X_i, but not round key k_i, should wind up knowing less about X_{i+1}. The process by which this observer loses knowledge of the sequence of X_i values is called *confusion* [Sha49].

Given information about X_i, there should be some X_{i+t} about which the observer who knows nothing about the round keys has no knowledge—such values should be indistinguishable to this observer from uniformly distributed, random n-bit strings.

In a Feistel network, bit j of the block can be obscured only when bit j appears in the target block of a given round. This means that the number of times that bit j can be obscured per cycle can be no larger than the number of times per cycle that bit j appears in the source block. This is called the *rate of confusion*, denoted as R_c.

Definition 17. The *rate of confusion* of a consistent UFN[2] is the minimum number of times per cycle that any bit can occur in the target block.

Lemma 18. *For a s:t UFN, the rate of confusion,*

$$R_c <= \frac{t}{n}.$$

Note that it can never take fewer than $\frac{1}{R_c}$ rounds for an observer to lose all information about the block in a UFN.

The rate of confusion and linear cryptanalysis An increase in R_c tends to increase resistance to linear cryptanalysis, when all other variables are held constant. While we're not yet able to prove this for the general case, we have a result for even complete UFNs [Wag95].

Lemma 19. *A complete UFN is even if its structure is b:ub or ub:b, with some j, b where $b \mid n$. This is a simple restatement of our definition of an even complete UFN.*

Definition 20. An *active round* in a linear attack is a round in which there is some linear approximation in the target block.[3]

[2] Note that the rate of confusion of an inconsistent UFN may not be the same for different cycles.

[3] This discussion may be complicated somewhat by multiple linear approximations. In this section, we are discussing single linear approximations only.

Theorem 21. *If an even complete UFN has C rounds per cycle, then the fraction of rounds per cycle that are active in a linear attack is at least $R_c = t/n$.*

This follows from the fact that R_c measures the minimum number of times per cycle that any bit or subset of bits may have appeared in the target block.□

Theorem 22. *Let p be the bias of the best possible approximation for an active round under a linear attack. Then the output bias of any nontrivial linear characteristic propagating through C rounds is at most $(2^{(CR_c-1)}p^{CR_c})$.*

From our previous assertions, we can see that in each C rounds, there are at least CR_c active rounds under a linear attack. The best possible bias of a C-round approximation follows from this, and from the rules for concatenating linear characteristics [Bih95].□

The rate of confusion also appears to provide resistance to the extended class of linear attacks described in [HKM95].

Intuitively, anytime a bit string with any uncertainty at all is XORed into the bits of an approximation, that approximation's bias decreases. Assuming that there is no perfect approximation of any subset of the F-function's outputs, this implies that a higher rate of confusion leads to a smaller bias for a linear approximation sent through a full cycle. It is important to note, however, that a high rate of confusion is not enough to guarantee resistance to linear cryptanalysis.

2.7 The Rate of Diffusion

Any change in X_i should have some chance to change every bit of X_{i+t}, for some t. The process by which one bit in the block has the chance to affect other bits in the block is called *diffusion* [Sha49].

In a Feistel cipher, the only time bit j can affect other bits in the block is when bit j is in the source block. This leads naturally to the idea of the *rate of diffusion*.

Definition 23. The *rate of diffusion* is the smallest number of times per cycle that a given bit can have the chance to affect other bits in the block.

Lemma 24. *For a s:t UFN, the rate of diffusion is*

$$R_d, <= \frac{s}{n}.$$

As a single bit progresses through a source-heavy even complete UFN it is an input into the F-function $C-1$ times; after that it is encrypted itself. Each time it is used differently; that is, each input into the F-function is unique. Each bit is used with $n+s-1$ other bits, some more than others. After one cycle, each bit is diffused $C-1$ times through the block, using $C-1$ different applications of the F-function.

The rate of diffusion is the measure of how many times per cycle each bit is used to encrypt other bits. It is upwardly bounded by the proportion of the block used as input to the F-function. DES has a much slower actual diffusion rate, due to the particular characteristics of its F-function. Both Blowfish and CAST have the maximum diffusion rate for a conventional Feistel network: $\frac{1}{2}$.

Note that bits in the same subblock cannot directly affect each other. These bits can affect each other only by affecting other bits not in the same subblock, which then affect bits in this subblock.

The rate of diffusion and differential cryptanalysis An increase in R_d tends to increase resistance to differential cryptanalysis, when all other variables are held constant. While we are not yet able to prove this for the general case, we have a result for even complete UFN constructions [Wag95].

Definition 25. An *active round* under a differential attack is a round in which there is a nonzero input difference into the F-function.

Theorem 26. *If an even complete UFN has C rounds per cycle, then the fraction of rounds per cycle that are active in a differential attack is at least $R_d = s/n$.*

This follows from the definition of an active round under differential attack, and from the definition of the rate of diffusion.□

Theorem 27. *Let p be the greatest possible probability for a nontrivial characteristic to propagate through a round. Then, the probability of any nontrivial characteristic propagating through C consecutive rounds is at most $p^{(CR_d)}$.*

A differential characteristic has its probability computed by multiplying all of its constituent characteristics' probabilities. No C round characteristic can have fewer than CR_d active one round characteristics, and inactive round characteristics have probability of one. Since no one round active characteristic can have probability of more than p, it's clearly impossible for any C round characteristic to have greater than p^{CR_d} probability.□

Note that this is related to a result in [NK95].

Intuitively, each time an input difference appears in the source block, unless the desired output difference happens with probability 1, the differential characteristic becomes less likely to successfully pass through a cycle. However, it is important to note that a high rate of diffusion alone is not sufficient to ensure resistance to differential attack–the differential properties of the cipher's F-function must also be taken into account. Additionally, differential attacks on source-heavy UFNs (those with realtively high rates of diffusion) are complicated by the fact that inputs into successive rounds are closely related. This is discussed below.

2.8 Examples of our Formulation

While the great majority of block ciphers designs are conventional Feistel networks, there have been examples of UFNs in the literature.

MacGuffin MacGuffin is a 48:16 UFN ($b = 16$ and $n = 64$) designed to introduce the concept of UFNs [BS95]. The F-function closely mirrors DES; it consists of a permutation, an XOR of 48 key bits, an S-box substitution (using the high-order two bit output of the eight DES S-boxes), and another permutation. The S-box size, 6-by-2, and contents left it susceptible to a differential attack [PR95].

BEAR and LION BEAR and LION are block cipher constructions designed by Ross Anderson and Eli Biham, which can be used to build a three-round inconsistent heterogenous UFN out of any keyed hash function with an n-bit output, and any stream cipher with an n-bit key [AB96]. BEAR uses the keyed hash function as the F-function for the first round, which is generally source heavy, and then uses the stream cipher for the second round, which is target heavy. The third round is another source heavy application of the keyed hash function. LION uses the stream cipher for the first and third (target-heavy) rounds, and the keyed hash function for the second round. Both apparently have the property that if the hash function and stream cipher are secure (for reasonable definitions of "secure"), the resulting block cipher is also secure.

The MD4 Family of Hash Functions The underlying structure of all these hash functions is a block algorithm, with a Davies-Meyer feedforward function to convert it into a one-way hash function [Win84]. In SHA [NIST93], for example, the block algorithm is an 80-round, 128:32 even complete UFN. There are some additional complications making SHA a GUFN: the output of the F-function is combined with the target block using addition modulo 2^{32} instead of XOR, and there is an additional cyclic shift of part of the source block in each round.

Other hash functions in this family include MD4 [Riv91], MD5 [Riv92], RIPE-MD [RIPE92], and Haval [ZPS93]. All use the same basic UFN structure: MD4 is a 48 round 96:32 UFN, MD5 is a 64 round 96:32 UFN, and Haval is a 224:32 UFN with a variable number of rounds. RIPE-MD runs two almost identical copies of MD4 in parallel, allows some interaction between the two copies' internal values during operations, and combines the outputs together to get the hash value. Its internal UFN structure is the same as that of MD4.

Note that these hash functions' compression functions are source-heavy heterogenous UFNs.

GDES GDES is a variant of DES where the output of the F-function is concatenated with itself multiple times and XORed with a much larger target block[Sch83]. It was defined for a variety of parameters, and is a 32:32q UFN. However, the number of rounds recommended was too small and the cipher was vulnerable to differential cryptanalysis [BS93].

Khufu/Khafre Khufu and Khafre [Mer91] are both even incomplete heterogenous target-heavy UFNs: $s = 8$, $t = 32$, $b = 8$, and $n = 64$. One cycle equals an *octet* in Merkle's terminology. Note that in Khufu, the sub-blocks are shifted around a little differently than in our notation, and also that each round's source block is taken from the previous round's target block.

REDOC III REDOC III [Sch94a] is a target-heavy UFN with $n = 80$, $s = 8$, and $t = 72$. The notation in the REDOC III documentation is somewhat different than what appears here, because REDOC III isn't designed to shift around the source and target blocks.

Non-Linear Feedback Shift Registers The most extreme case of a source-heavy UFN — $(n-1)$:1 — encrypts a one-bit target using the rest of the block's bits as source bits. This is essentially a nonlinear feedback shift register (NLFSR). The most extreme case of a target-heavy UFN — 1:$(n-1)$ — encrypts all but one bit of the block, using a value determined by a single source bit. This can be viewed as a NLFSR in Galois format. Note that this kind of construction is always affine, and thus can be easily broken [Wag95].

3 Comparisons of Generic Constructions

This section examines differential and linear attacks against a variety of even complete UFN constructions. We try to make as few assumptions about the F-function as possible. They are assumed only to be mappings as defined in Definition X. We will assume specific linear and differential characteristics in order to see how the they propagate through different UFN constructions.

3.1 Differential Cryptanalysis

Consider an F-function where an input difference a produces a 0 output difference with probability p. Used in a b:b UFN, this can be used to create a 2-round differential characteristic with probability p. (In the following examples, a, b, ?, and 0 are all b-bit numbers. In this section, $\delta_s \rightarrow \delta_t$ is used to show the actual differential relationship which is being exploited.)

Round	Input Difference	$\delta_s \rightarrow \delta_t$	Output Difference	Prob.
1	$(0, a)$	$0 \rightarrow 0$	$(a, 0)$	1
2	$(a, 0)$	$a \rightarrow 0$	$(0, a)$	p

If we assume a b:$3b$ UFN, and that an input difference a produces a zero difference, this can be used to create a 4-round characteristic with probability p:

Round	Input Difference	$\delta_s \rightarrow \delta_t$	Output Difference	Prob.
1	$(0, 0, 0, a)$	$0 \rightarrow (0, 0, 0)$	$(a, 0, 0, 0)$	1
2	$(a, 0, 0, 0)$	$a \rightarrow (0, 0, 0)$	$(0, a, 0, 0)$	p
3	$(0, a, 0, 0)$	$0 \rightarrow (0, 0, 0)$	$(0, 0, a, 0)$	1
4	$(0, 0, a, 0)$	$0 \rightarrow (0, 0, 0)$	$(0, 0, 0, a)$	1

A similar differential can be used in a b:ub UFN to create an $u+1$-round characteristic with probability p.

It is more difficult to exploit this differential with a $3b$:b UFN.[4] In order to get anywhere, we need to add the assumption that a differential of a in any of the three sub-blocks produces an output differential of 0 with probability p. This is a 4-round characteristic with probability $= p^3$:

Round	Input Difference	$\delta_s \to \delta_t$	Output Difference	prob.
1	$(0,0,0,a)$	$(0,0,0) \to 0$	$(a,0,0,0)$	1
2	$(a,0,0,0)$	$(a,0,0) \to 0$	$(0,a,0,0)$	p
3	$(0,a,0,0)$	$(0,a,0) \to 0$	$(0,0,a,0)$	p
4	$(0,0,a,0)$	$(0,0,a) \to 0$	$(0,0,0,a)$	p

A similar differential can be used in a ub:b UFN to create an $u+1$-round characteristic with probability p^u. In general, the differential can be used in a ub:vb UFN to create a $(u+v)$-round iterative characteristic with probability p^u. This is another illustration: a $3b$:$2b$ UFN:

Round	Input Difference	$\delta_s \to \delta_t$	Output Difference	prob.
1	$(0,0,0,a,0)$	$(0,0,0) \to (0,0)$	$(a,0,0,0,0)$	1
2	$(a,0,0,0,0)$	$(a,0,0) \to (0,0)$	$(0,0,a,0,0)$	p
3	$(0,0,a,0,0)$	$(0,0,a) \to (0,0)$	$(0,0,0,0,a)$	p
4	$(0,0,0,0,a)$	$(0,0,0) \to (0,0)$	$(0,a,0,0,0)$	1
5	$(0,a,0,0,0)$	$(0,a,0) \to (0,0)$	$(0,0,0,a,0)$	p

While at first glance, this seems to imply that UFNs with a larger R_d are less likely to have good multi-round differential characteristics, this is not always the case. As b gets smaller, the assumptions required to produce this characteristic become more implausible. At the extreme, if $b=1$ we would have to assume that any time we have a single bit difference somewhere in the input, it always leads to the same output difference with some usably high probability. However, as t gets smaller, it appears that high-probability differentials become easier to find again [OCo94b]. Note that this may lead to situations in which a multi-round characteristic is extremely difficult to find, but relatively easy to exploit.

A second type of differential works with any even construction: a difference of a in every sub-block of s produces a difference of 0 in every sub-block of t with probability p [Knu95]. This results in a 1-round characteristic with probability p.

Now consider a different type of differential: an F-function where an input difference a produces an output difference b with probability p_1, and where an input difference b produces an output difference a with probability p_2. Used in a (balanced) b:b UFN, this can be used to create a 3-round characteristic with probability $p_1 p_2$.

[4] For some F-functions, differential attacks on source-heavy UFNs have additional complications, because inputs to successive rounds' F-functions are closely related. This is discussed below.

Round	Input Difference	$\delta_s \to \delta_t$	Output Difference	prob
1	$(0, a)$	$0 \to 0$	$(a, 0)$	1
2	$(a, 0)$	$a \to b$	(b, a)	p_1
3	(b, a)	$b \to a$	$(0, b)$	p_2

Note that this is not an iterative characteristic, because the differences of the inputs and outputs are not equal. However, this is one half of a 6-round iterative characteristic: concatenated with itself with rounds 2 and 3 exchanged [Knu93], with probability ${p_1}^2{p_2}^2$.

Assume a b:$3b$ UFN, and that an input difference a produces an output difference $(0, b, 0)$ with probability p_1, and an input difference b produces an output difference $(0, a, 0)$ with probability p_2. These differentials can be used to create a 6-round characteristic with probability p_1p_2 — one that can be concatenated with another similar characteristic to create a 12-round iterated characteristic with probability ${p_1}^2{p_2}^2$:

Round	Input Difference	$\delta_s \to \delta_t$	Output Difference	prob.
1	$(0, 0, 0, a)$	$0 \to (0, 0, 0)$	$(0, 0, a, 0)$	1
2	$(0, 0, a, 0)$	$0 \to (0, 0, 0)$	$(0, a, 0, 0)$	1
3	$(0, a, 0, 0)$	$0 \to (0, 0, 0)$	$(a, 0, 0, 0)$	1
4	$(a, 0, 0, 0)$	$a \to (0, b, 0)$	$(0, b, 0, a)$	p_1
5	$(0, b, 0, a)$	$0 \to (0, 0, 0)$	$(b, 0, a, 0)$	1
6	$(b, 0, a, 0)$	$b \to (0, a, 0)$	$(0, 0, 0, b)$	p_2

This kind of differential cannot be exploited in a ub:b UFN, even with the added assumptions that a differential of a in any of the three sub-blocks produces an output differential of b with probability p_1, and that a differential of b in any of the three sub-blocks produces an output differential of a with probability p_2. This example shows a $3b$:b UFN:

Round	Input Difference	$\delta_s \to \delta_t$	Output Difference	Prob.
1	$(0, 0, 0, a)$	$(0, 0, 0) \to 0$	$(a, 0, 0, 0)$	1
2	$(a, 0, 0, 0)$	$(a, 0, 0) \to b$	$(b, a, 0, 0)$	p_1
3	$(b, a, 0, 0)$	$(b, a, 0) \to ?$	$(?, b, a, 0)$	?

Complications of Differential Attacks on UFNs

Related inputs to successive rounds' F-functions Source-heavy UFNs have the property that successive rounds have some portions of their inputs in common. For example, consider a $3b$:b UFN:

Round	Operation
1	$A = A \bigoplus f_{k1}(B, C, D)$
2	$B = B \bigoplus f_{k2}(C, D, A)$
3	$C = C \bigoplus f_{k3}(D, A, B)$
4	$D = D \bigoplus f_{k4}(A, B, C)$

It's clear that each round's F-function shares two of its three input sub-blocks with the previous round, shifted over one sub-block. This is significantly different than the situation in a balanced Feistel network, in which the the entire input to the F-function was XORed with the output of the previous round's F-function, and thus may be assumed to be somewhat random.

Key-dependent characteristics The notion of key-dependent characteristics is discussed in [BB93]. A key-dependent differential characteristic is a differential characteristic whose probability is partially a function of the specific round keys chosen. In some cases, such as the attacks on RDES and Lucifer in [BB93], this provides a generally useful attack on the cipher. In other cases, such as in the weak keys of IDEA [Dae95], a rare weak key condition (consisting of an interaction between the cipher key, the key schedule, and the round function) can lead to relatively high probability differential characteristics.

It is natural to imagine designing a UFN in which the round key material is XORed with the bits in the source block before being fed into the F-function. In this case, if a given differential for the F-function is based on a small number of specific pairs of inputs, then we will wind up with C round differential characteristics that are strongly dependent upon relationships between the subkeys for subsequent rounds. Depending on the specific probabilities involved, this can lead to a significant vulnerability of the cipher to differential attack, a rare weak key condition, or no additional vulnerability. Note that this occurs because the inputs to the F-function for successive active rounds are related to one another, based on the round keys.

Another natural way to design a UFN might be to XOR the round key into the output of the F-function for that round. In this case, the inputs to the F-function for successive rounds are very simply related. For some F-functions, this can lead to C round differential characteristics with probability p, where p is the probability of a single-round differential.

Thus, we can see that some issues that are apparently not very important in balanced Feistel network design can become important in UFN design.

Using multiple key-dependent differential characteristics to learn key information When there are many possible key-dependent differential characteristics, we may be able to the fact that some characteristics pass through the cipher, while others do not, as a way to immediately learn quite a bit of internal key information.

Treating different sub-blocks of the F-function's input differently Good differential characteristics seem less likely to exist when different sub-blocks of the F-function's input are treated differently enough that they do not have the same differential properties. For example, even if an input XOR of α leads to an output XOR of 0 in the first sub-block of the F-function's input, if the same input XOR has no such effect when it appears in the second sub-block, then a differential characteristic based on α cannot pass through a full cycle. This is easy to see in the discussion that appears in the appendix.

Conclusions From the discussion in this section, a couple of things should be clear:

1. As a general rule, differential cryptanalysis becomes more difficult as the rate of diffusion becomes higher—that is, more source-heavy UFNs tend to have more inherent resistance to differential attacks.
2. Determining the susceptibility to differential attack of a UFN is somewhat more complex than is suggested by the first section of our discussion, above. In particular, source heavy UFNs have related inputs going into successive rounds' F-functions, and in some circumstances, this can lead to new vulnerabilities to differential cryptanalysis. In other circumstances, such UFNs can actually be strengthened against some kinds of differential attack by this property.
3. F-functions which treat different sub-blocks of input differently are much more likely to be resistant to differential cryptanalysis than F-functions which treat different sub-blocks identically.

3.2 Linear Cryptanalysis

Consider an F-function with a linear approximation of some output bits, a, based on some linear combination of the key bits only. Used in a b:b UFN, this can be used to create a 2-round linear characteristic with probability $(\frac{1}{2}+p_1)$. Note that, in the tables in this section, $A_s \rightarrow A_t$ is used to denote the actual linear approximation being exploited in a given round. This should be read "the parity of $A_s \bigoplus$ the parity of $A_t = 1$.

Round	Input Block	$A_s \rightarrow A_t$	Output Block	Prob.
1	$(a,?)$	$a \rightarrow ?$	$(?,a)$	1
2	$(?,a)$	$? \rightarrow a$	$(a,?)$	$(\frac{1}{2}+p_1)$

In these examples, ? indicates no approximation.

If we assume a $3b$:b UFN, this linear approximation gives rise to a 4-round characteristic with probability $(\frac{1}{2}+p_1)$:

Round	Input Block	$A_s \rightarrow A_t$	Output Block	Prob.
1	$(?,?,?,a)$	$(?,?,?) \rightarrow a$	$(a,?,?,?)$	$(\frac{1}{2}+p_1)$
2	$(a,?,?,?)$	$(a,?,?) \rightarrow ?$	$(?,a,?,?)$	1
3	$(?,a,?,?)$	$(?,a,?) \rightarrow ?$	$(?,?,a,?)$	1
4	$(?,?,a,?)$	$(?,?,a) \rightarrow ?$	$(?,?,?,a)$	1

A similar linear approximation can be used in a ub:b UFN to create an $u+1$-round linear characteristic with probability $(\frac{1}{2}+p)$.

It is much more difficult to exploit this linear approximation in a b:$3b$ UFN. To even begin, we need the additional assumption that the approximation occurs with some useful bias in all of the three output sub-blocks of the F-function. Here is a 4-round characteristic with probability $(\frac{1}{2}+4p^3)$:

Round	Input Block	$A_s \to A_t$	Output Block	Prob.
1	$(?,?,?,a)$	$? \to (?,?,a)$	$(?,?,a,?)$	$(\frac{1}{2}+p)$
2	$(?,?,a,?)$	$? \to (?,a,?)$	$(?,a,?,?)$	$(\frac{1}{2}+p)$
3	$(?,a,?,?)$	$? \to (a,?,?)$	$(a,?,?,?)$	$(\frac{1}{2}+p)$
4	$(a,?,?,?)$	$a \to (?,?,?)$	$(?,?,?,a)$	1

A similar approximation can be used in a b:ub UFN to create a $u+1$-round characteristic with probability $(\frac{1}{2}+2^{u-1}p^u)$. In general, the linear approximation can be used in a ub:vb UFN to create a $(u+v)$-round iterative characteristic with probability $(\frac{1}{2}+2^{v-1}p^v)$.

While at first glance, this seems to imply that UFNs with a larger R_c are less likely to have good multi-round linear characteristics, this is not always the case: as t gets larger with respect to s, we are more likely to have linear relationships [Bih95]. There would appear to be an optimal s:t ratio for resistance to linear cryptanalysis, but we are not yet able to determine that ratio.

Now consider a different linear approximation: an F-function with a two useful linear approximations: First, there is a linear relationship between the input bits in a and the output bits in b with bias p_1. Second, there is a linear relationship between the input bits in b and the output bits in a with bias p_2.

Used in a b:b UFN, this can be used to create a 3-round characteristic with probability $(\frac{1}{2}+2p_1p_2)$.

Round	Input Block	$A_s \to A_t$	Output Block	Prob.
1	$(a,?)$	$a \to ?$	(b,a)	1
2	(b,a)	$b \to a$	(a,b)	$(\frac{1}{2}+p_2)$
3	(a,b)	$a \to b$	$(b,?)$	$(\frac{1}{2}+p_1)$

Note that this is not an iterative characteristic, because the inputs and outputs are not equal. However, this is one half of a 6-round iterative characteristic: concatenated with itself with rounds 2 and 3 exchanged [Knu94a].

Assume a $3b$:b UFN. There are two linear approximations: a linear combination of input bits $(0,a,0)$ produces a linear combination of output bits b with probability $(\frac{1}{2}+p_1)$, and where an linear combination of input bits $(0,b,0)$ produces a linear combination of output bits a with probability $(\frac{1}{2}+p_2)$. This can be used to create a 5-round characteristic with probability $(\frac{1}{2}+2p_1p_2)$:

Round	Input Block	$A_s \to A_t$	Output Block	Prob.
1	$(?,a,?,?)$	$(?,a,?) \to ?$	$(b,?,a,?)$	1
2	$(b,?,a,?)$	$(b,?,a) \to ?$	$(?,b,?,a)$	1
3	$(?,b,?,a)$	$(?,b,?) \to a$	$(a,?,b,?)$	$(\frac{1}{2}+p_2)$
4	$(a,?,b,?)$	$(a,?,b) \to ?$	$(?,a,?,b)$	1
5	$(?,a,?,b)$	$(?,a,?) \to b$	$(?,b,?,?)$	$(\frac{1}{2}+p_1)$

This characteristic be concatenated with another similar characteristic to create a 12-round iterated characteristic.

This kind of characteristic is more complicated in a b:$3b$ UFN. More assumptions about the linear characteristic are required, and the probabilities drop much faster.

Another type of linear relation works with any construction: a linear combination of input bits a in every sub-block produces a linear combination of output bits a in every sub-block of t with probability $(\frac{1}{2}+p)$ [Knu95]. This results in a 1-round iterative characteristic with probability $(\frac{1}{2}+p)$.

Complications of Linear Cryptanalysis in UFNs There are some complications that need to be dealt with in linear cryptanalysis of UFNs. First, an n-round b:$3b$ UFN has an F-function output which has three times as many output bits as input bits. This appears to make linear relationships between the input and output bits, or simply between output bits, more likely to occur. Second, in a source-heavy UFN, the inputs to successive rounds' F-functions are closely related—depending upon the definition of the F-function, this may further complicate linear cryptanalysis of this kind of UFN.

Conclusions While this analysis is by no means exhausting, it strongly suggests that resistance to linear cryptanalysis is tied directly to R_c: as the size of t grows with respect to s, linear attacks become more difficult. However, the larger the target block, the more likely it is to have some useful linear approximations. Extremely target-heavy UFNs (such as an 8:1024 UFN) are certain to have some strong linear approximations in the outputs of their F-functions [Bih95, Bih95].

4 Conclusions and Open Problems

Our research suggests that the structure of the Feistel network affects the security of the block cipher. This is surprising, given how sensitive Feistel networks are to the choice of F-function. Modifying the ratio of s and t can systematically affect the security of the cipher, when all other variables are held constant. Additionally, such issues as how the round keys are combined into the cipher can become important in UFN design—much more so than in balanced Feistel cipher design.

The goal of block cipher design is not so much to create a strong algorithm, but to create a strong algorithm that runs in a reasonable time on currently available hardware, that is reasonably easy to implement, etc. As such, UFN designs may allow us to design faster and better block ciphers than we could if we limited ourselves to balanced Feistel structures.

Most of this paper consisted of laying the foundations for future discussion and making some initial observations. Much work remains to be done in this area. Some of the open problems that we feel are important are:

- It seems that source-heavy UFNs have some inherent resistance to differential attacks. Assuming some basic number of cycles, perhaps four, is there an optimal s:t ratio for resistance to differential attacks?

- A target-heavy UFN seems to have some inherent resistance to linear attacks. Clearly, a UFN with t very large and s very small is going to have some linear characteristics that happen with certainty [Bih95], and any $1{:}(n-1)$ UFN will be affine [Wag95]. Assuming some basic number of cycles, again perhaps four, is there an optimal $s{:}t$ ratio for resistance to linear attacks?
- Is it possible to use the design of the F-function to provide some resistance against linear attacks, and the UFN structure to provide some resistance against differential attacks? For example, F-functions that XOR the results of their S-box lookups together to form their outputs make linear attacks more difficult, but XORing S-box entries together adds no difficulty to differential attacks. Would combining an F-function with natural resistance to linear attacks with a UFN structure with natural resistance to differential attacks give us a cipher resistant to both linear and differential attack?
- We have discussed UFN security primarily in the context of number of cycles rather than number of rounds. However, this may not be as useful as we would like, because extremely source- or target-heavy UFNs are likely to have very large numbers of rounds per cycle. For example, a 63:1 UFN would require 64 rounds per cycle, and probably at least four cycles (256 rounds) for good security. Most other useful ways to measure UFN security appear to be very dependent on the specific F-function (such as gate count and latency for hardware implementations, and number of sequential operations and memory accesses for software implementations). Is there a better way to compare security of generic UFN constructions?
- Is it reasonable to alternate several rounds of source-heavy UFN for resistance to differential attacks, followed by several rounds of target-heavy UFN for resistance to linear attacks? What would differential and linear attacks against such structures look like?
- Do incomplete UFNs have any advantages over complete UFNs? Khufu is similar to Blowfish in that an 8-bit sub-block becomes the input to an 8-by-32 S-box. While in Blowfish four such S-box outputs are combined together and XORed with t, in Khufu only one is XORed per round. In some software implementations four rounds of Khufu is no slower than one round of Blowfish: one cycle takes the same amount of processing time. Is one construction preferable to the other?
- The design of Khufu hints at a useful design principle for incomplete UFN structures—one which is naturally adhered to in even complete UFNs: The source block for round i should be taken from part or all of the target block for round $i-1$. What other useful design principles should be adhered to when designing incomplete UFNs?
- Do odd UFNs have important advantages over even UFNs? The interesting thing about odd UFNs is that one rotation (the time it takes for all bits in the block to return to their original positions) takes more than one cycle. For example, in a 40:24 UFN, one cycle takes three rounds, but one rotation takes eight rounds:

Block(showing bytes)	Round	Cycle	Rotation
(a, b, c, d, e, f, g, h)	1	1	1
(f, g, h, a, b, c, d, e)	2	1	1
(c, d, e, f, g, h, a, b)	3	1	1
(h, a, b, c, d, e, f, g)	4	2	1
(e, f, g, h, a, b, c, d)	5	2	1
(b, c, d, e, f, g, h, a)	6	2	1
(g, h, a, b, c, d, e, f)	7	3	1
(d, e, f, g, h, a, b, c)	8	3	1
(a, b, c, d, e, f, g, h)	9	3	2

- For prime UFNs, n rounds are required before the same bit positions enter the F-function in the same way. One rotation takes n rounds, i.e. 64 for a 64-bit block cipher. This construction seems to make it extremely difficult to find differential and linear characteristics in F-functions that use S-boxes. Differential characteristics keep getting split between S-box entries from round to round; similarly linear characteristics keep getting different S-box outputs XORed into them. While this intuitively suggests that prime UFNs with randomly generated S-boxes should have a very low probability of being susceptible to differential or linear attacks, we aren't yet able to prove or disprove this. (It is possible that prime UFN construction simply makes differential and linear characteristics harder to find. Also, it looks like this kind of design may add strength to UFNs with some kinds of F-function, but not to others.)
- UFNs with a $ub{:}b$ structure can be viewed as simple NLFSRs with a block size u and length n. This observation provides a theoretical link between block- and stream-cipher cryptanalysis. Average cycle length of this NLFSR tells us important things about the group properties of the underlying cipher. Worst-case cycle length tells us important things about weak keys or special input values that may demonstrate weaknesses in the cipher. Are there other stream-cipher techniques that could be applied to block ciphers of this type?

5 Acknowledgments

The authors would like to thank Matthew Blaze, Lars Knudsen, and David Wagner for their invaluable help with the paper, and Steve Bellovin for his invaluable help with LaTeX.

References

[AB96] R. Anderson and E. Biham, "Two Practical and Provably Secure Block Ciphers: BEAR and LION," *Proceedings of the Cambridge Algorithms Workshop*, 1996, to appear.

[AT93] C.M. Adams and S.E. Tavares, "Designing S-boxes for Ciphers Resistant to Differential Cryptanalysis," *Proceedings of the 3rd Symposium on State and Progress of Research in Cryptography*, Rome, Italy, 15-16 Feb 1993, pp. 181-190.

[BB93] I. Ben-Aroya and E. Biham, "Differential Cryptanalysis of Lucifer," *Advances in Cryptology—CRYPTO '93 Proceedings*, Spinger-Verlag, 1994.

[Bih95] E. Biham, "On Matsui's Linear Cryptanalysis," *Advances in Cryptology — EUROCRYPT '94 Proceedings*, Springer-Verlag, 1995, to appear.

[BJ77] G. Bhattacharyya and R. Johnson, *Statistical Concepts and Methods*, John Wiley and Sons, 1977.

[BS93] E. Biham and A. Shamir, *Differential Cryptanalysis of the Data Encryption Standard*, Springer-Verlag, 1993.

[BS95] M. Blaze and B. Schneier, "The MacGuffin Block Cipher Algorithm," *Fast Software Encryption, Second International Workshop Proceedings*, Springer-Verlag, 1995, pp. 97-110.

[BPS93] L. Brown, M. Kwan, J. Pieprzyk, and J. Seberry, "Improving Resistance to Differential Cryptanalysis and the Redesign of LOKI," *Advances in Cryptology — ASIACRYPT '91 Proceedings*, Springer-Verlag, 1993, pp. 36-50.

[CDN95] G. Carter, E. Dawson, and L. Nielsen, "DESV: A Latin Square Variation of DES," *Proceedings of the Workshop on Selected Areas in Cryptography*, Ottawa, Canada, 1995.

[DGV94] J. Daemen, R. Govaerts, and J. Vandewalle, "A New Approach to Block Cipher Design," *Fast Software Encryption, Cambridge Security Workshop Proceedings*, Springer-Verlag, 1994, pp. 18-32.

[Dae95] J. Daemen, "Cipher and Hash Function Design," Ph.D Thesis, Katholieke Universiteit Leuven, Mar 95.

[Fei73] H. Feistel, "Cryptography and Computer Privacy," *Scientific American*, v. 228, n. 5, May 1973, pp. 15-23.

[GOST89] GOST, Gosudarstvennyi Standard 28147-89, "Cryptographic Protection for Data Processing Systems," Government Committee of the USSR for Standards, 1989.

[HKM95] C. Harpes, G. Kramer, J. Massey, "A Generalization of Linear Cryptanalysis and the Applicability f Matsui's Piling-up Lemma," *Advances in Cryptology — EUROCRYPT '95 Proceedings*, Springer, 1995, pp. 24-38.

[Knu93] L.R. Knudsen, "Iterative Characteristics of DES and s^2 DES," *Advances in Cryptology — CRYPTO '92*, Springer-Verlag, 1993, pp. 497-511.

[Knu94a] L.R. Knudsen, "Block Ciphers — Analysis, Design, Applications," Ph.D. dissertation, Aarhus University, Nov 1994.

[Knu94b] L.R. Knudsen, "Practically Secure Feistel Ciphers," *Fast Software Encryption, Cambridge Security Workshop Proceedings*, Springer-Verlag, 1994, pp. 211-221.

[Knu95] L.R. Knudsen, personal communication.

[Mer91] R.C. Merkle, "Fast Software Encryption Functions," *Advances in Cryptology — CRYPTO '90 Proceedings*, Springer-Verlag, 1991, pp. 476-501.

[NBS77] National Bureau of Standards, NBS FIPS PUB 46, "Data Encryption Standard," National Bureau of Standards, U.S. Department of Commerce, Jan 1977.

[NIST93] National Institute of Standards and Technology, NIST FIPS PUB 180, "Secure Hash Standard," U.S. Department of Commerce, May 93.

[Nyb91] K. Nyberg, "Perfect Nonlinear S-boxes," *Advances in Cryptology — EUROCRYPT '91 Proceedings*, Springer-Verlag, 1991, pp. 378-386.

[Nyb93] K. Nyberg, "On the Construction of Highly Nonlinear Permutations," *Advances in Cryptology — EUROCRYPT '92 Proceedings*, Springer-Verlag, 1993, pp. 92-98.

[Nyb94] K. Nyberg, "Differentially Uniform Mappings for Cryptography," *Advances in Cryptology — EUROCRYPT '93 Proceedings*, Springer-Verlag, 1994, pp. 55-64.

[NK95] K. Nyberg and L.R. Knudsen, "Provable Security Against Differential Cryptanalysis," *Journal of Cryptology*, v. 8, n. 1, 1995, pp. 27-37.

[OCo94a] L. O'Connor, "Enumerating Nondegenerate Permutations," *Advances in Cryptology — EUROCRYPT '93 Proceedings*, Springer- Verlag, 1994, pp. 368-377.

[OCo94b] L. O'Connor, "On the Distribution of Characteristics in Bijective Mappings," *Advances in Cryptology — EUROCRYPT '93 Proceedings*, Springer-Verlag, 1994, pp. 360-370.

[OCo94c] L. O'Connor, "On the Distribution of Characteristics in Composite Permutations," *Advances in Cryptology — CRYPTO '93 Proceedings*, Springer-Verlag, 1994, pp. 403-412.

[PR95] B. Preneel and V. Rijmen, "Cryptanalysis of MacGuffin," *Fast Software Encryption, Second International Workshop Proceedings*, Springer-Verlag, 1995, pp. 353-358.

[RIPE92] Research and Development in Advanced Communication Technologies in Europe, *RIPE Integrity Primitives: Final Report of RACE Integrity Primitives Evaluation (R1040)*, RACE, June 1992.

[Riv91] R.L. Rivest, "The MD4 Message Digest Algorithm," *Advances in Cryptology — CRYPTO '90 Proceedings*, Springer- Verlag, 1991, pp. 303-311.

[Riv92] R.L. Rivest, "The MD5 Message Digest Algorithm," RFC 1321, Apr 1992.

[Riv95] R.L. Rivest, "The RC5 Encryption Algorithm," *Fast Software Encryption, Second International Workshop Proceedings*, Springer-Verlag, 1995, pp. 86-96.

[Sch83] I. Schaumuller-Bichl, "On the Design and Analysis of New Cipher Systems Related to the DES," Technical Report, Linz University, 1983.

[Sch94a] B. Schneier, *Applied Cryptography, Second Edition*, John Wiley & Sons, 1996.

[Sch94b] B. Schneier, "Description of a New Variable-Length Key, 64-Bit Block Cipher (Blowfish)," *Fast Software Encryption, Cambridge Security Workshop Proceedings*, Springer-Verlag, 1994, pp. 191-204.

[Sha49] C.E. Shannon, "Communication Theory of Secrecy Systems," *Bell Systems Technical Journal*, v. 27, n. 4, 1948, pp. 379-423.

[SM88] A. Shimizu and S. Miyaguchi, "Fast Data Encipherment Algorithm FEAL," *Advances in Cryptology — EUROCRYPT '87 Proceedings*, Springer-Verlag, 1988, pp. 267-278.

[Vau96] S. Vaudenay, "On the Weak Keys in Blowfish," *Proceedings of the Cambridge Algorithms Workshop*, 1996, to appear.

[Wag95] D. Wagner, personal communication.

[Win84] R.S. Winternitz, "Producing One-Way Hash Functions from DES," *Advances in Cryptology: Proceedings of Crypto 83*, Plenum Press, 1984, pp. 203-207.

[ZPS93] Y. Zheng, J. Pieprzyk, and J. Seberry, "HAVAL – A One-Way Hashing Algorithm with Variable Length of Output," *Advances in Cryptology — AUSCRYPT '92 Proceedings*, Springer-Verlag, 1993, pp. 83-104

A Preliminary Cryptanalysis of a Source-Heavy UFN Construction

A.1 Introduction

This appendix discusses a preliminary attempt at cryptanalysis of a conceptually simple 48:16 UFN. The discussion that follows is not meant to be an exhaustive discussion of the relative merits of these two ciphers—rather it is intended to demonstrate in a more concrete form the complications that can occur in attacking and designing UFNs.

A.2 The Construction: A Source-Heavy Blowfish Variation

Description of the Construction[5] This is a 48:16 even complete source-heavy UFN.

Definition 28. Let $X = (x_0, x_1, ..., x_5)$, and $K_r = (k_{r,0}, k_{r,1}, ..., k_{r,5})$. Then the rth round's F-function is

$$F(X, K_r) = \bigoplus_{i=0}^{s/8-1} S_i[x_i \bigoplus k_{r,i}].$$

Differential Cryptanalysis Serge Vaudenay pointed out a kind of differential characteristic that can occur in a Blowfish-like F-function with known, weak S-boxes [Vau96]. A very powerful one-round differential involves having two or more identical entries in the same S-box. For example, if there exists $q \neq r$ such that $S[q] = S[r]$ for an S-box, $d = q \bigoplus r$ can be used an the input difference to this S-box, yielding an output difference of zero with probability 2^{-7}.

For the 48:16 construction, the characteristic does not extend past two rounds unless there is a rare, special situation with three of the six S-boxes used. Otherwise, the progression of the differential attack based on this one-round characteristic looks like this:

Round	Input Difference	$\delta_s \rightarrow \delta_t$	Output Difference	Prob.
1	$(0, 0, 0, d)$	$(0, 0, 0) \rightarrow 0$	$(d, 0, 0, 0)$	1
2	$(d, 0, 0, 0)$	$(d, 0, 0) \rightarrow 0$	$(0, d, 0, 0)$	p
3	$(0, d, 0, 0)$	$(0, d, 0) \rightarrow ?$	$(?, 0, d, 0)$	?

[5] Note that this construction is also closely related to CAST. A real implementation of this kind of cipher with key-dependent S-boxes might alternate addition modulo 2^{16} and XORing to combine S-box outputs. This appears to have no effect on our differential analysis, but would probably make linear cryptanalysis less effective in most cases.

While the probability of existence of an internal collision in an 8:16 random S-box is greater than half (by the birthday paradox), the existence of three such collisions (necessary to extend this attack to a $3b$:b UFN) is considerably less likely, and it is still less likely that such collisions will all have the same $d = q \bigoplus r$ value. However, further analysis shows that even this highly unlikely condition is not sufficient to generally allow differential attacks on this structure.

Consider an input pair, X and X^*. Suppose that we have a 48:16 UFN with $S_0[0] = S_0[128]$, $S_1[1] = S_1[129]$, and $S_2[2] = S_2[130]$. To simplify this attack, assume that we XOR the round keys into the outputs of the S-boxes instead of the inputs. The values of X, X^*, and X' are shown as 8-tupples of bytes.

X	X^*	$X' = X \bigoplus X^*$	Comments
$(0, a, b, c, d, e, f, g)$	$(128, a, b, c, d, e, f, g)$	$(128, 0, 0, 0, 0, 0, 0, 0)$	We get our zero output delta.
$(h, i, 0, a, b, c, d, e)$	$(h, i, 0, a, b, c, d, e)$	$(0, 0, 128, 0, 0, 0, 0, 0)$	We cannot get a zero output delta.

In words: It isn't enough to have the same characteristic to attack three S-boxes. We must actually have the same S-box inputs in X and X^* (and thus the same q and r values) give us identical S-box outputs for all S-boxes involved in the attack. For a 48:16 UFN, there are three S-boxes involved.

If S-boxes with identical input pairs leading to identical output pairs are installed, and the F-function is modified to XOR the round keys *after* the S-box entries, a 4-round iterative characteristic exists with probability 2^{-7} for the 48:16 construction, and any such ub:b Blowfish variant has a $(u+1)$-round iterative differential characteristic with probability 2^{-7}. However, if the F-function's definition is left so that the round key is XORed into the input before being fed into the S-boxes, then this differential characteristic is not even *possible* except in very rare weak-key conditions. In the vast majority of the cases, the round keys make it impossible for this characteristic to make it through even a single cycle. This happens because, for the differential to make it through two consecutive active rounds, it must be possible for the input to the active S-box in X_i and X_{i+1} to be either q or r in both rounds. This means that the corresponding byte of K_i XORed with the corresponding byte of K_{i+1}must be either 0 or d. If the round keys are random, the probability of such a differential characteristic being possible for a given cycle is 2^{-21}.

Lessons from the analysis There seem to be a few design principles that allow us to design source-heavy UFNs which will be highly resistant to differential cryptanalysis.

1. Design the F-function so that different sub-blocks of input are treated differently. This ruins most possible multi-round differential characteristics immediately.
2. Design the UFN so that the round key's effect on the F-function output is nonlinear. For many constructions, this means XORing the round key into

the F-function input before passing it through the S-boxes or other nonlinear components.

3. Analyze the F-function to see what the effects of having successive inputs be related in the way that they are for source-heavy UFNs. For many UFNs, this seems likely to lead to differential characteristics that are strongly related to subkeys.
4. For such UFNs, analyze the key schedule to verify that it is either impossible or extremely unlikely for any key to provide round keys related in such a way that any such differential attack is possible. For UFNs in which such round keys can possibly occur, those cipher keys that lead to such round keys form a class of weak keys for the cipher. Getting a given characteristic through the cipher detects the weak key condition, as well as revealing some information about the last round key.

Example 1. We can use our 48:16 Blowfish variant as an example of these design principles. Assuming independent round keys XORed into the input of the F-function each round, the probability of getting the round-key relations necessary for making it through two consecutive active rounds is 2^{-7}. Assuming four round iterative characteristics with three active adjacent rounds each, we appear to have a probability of only 2^{-21} of having the special round key relationships needed to allow this differential attack against each cycle. The probability of the round key relationships needed to get a characteristic through three cycles is 2^{-63}. This implies that with randomly chosen round keys, the probability of getting a set of round keys that makes the four-cycle version cipher vulnerable to a differential attack is 2^{-84}, and such an attack can have a three-cycle characteristic with probability 2^{-21}. Because of the F-function construction, even this rare special case requires an extremely unlikely set of conditions to be true of the S-boxes.

Linear Cryptanalysis While the source-heavy UFN structure of this cipher makes linear cryptanalysis potentially more effective, the F-function's construction appears to offer some protection against high-bias linear approximations. In this F-function, as is discussed above, six randomly-generated S-boxes' outputs are XORed together. We generated and tested three sets of 8:16 S-boxes. From this limited sample, the best approximation had a bias of about 0.1758. As we expected, there was no apparent tendency for the S-boxes to share a similar high-probability approximation.

For a linear approximation to be useful with this F-function, it must occur in all six S-boxes with some useful bias. It appears to be extremely unlikely that there will be any linear approximation in all six S-boxes simultaneously with better than 2^{-3}. We can therefore consider this the worst case. If we XOR the linear approximations from all six S-box outputs together, we get a linear approximation with a bias of 2^{-13}. This leads to a four-round iterative linear characteristic for this cipher with bias 2^{-13}.

Round	Input Block	$\alpha_s \to \alpha_t$	Output Block	Prob.
1	$(?,?,?,\alpha)$	$(?,?,?) \to \alpha$		$(\frac{1}{2}+0.2)$
2	$(\alpha,?,?,?)$	$(\alpha,?,?) \to ?$		1
3	$(?,\alpha,?,?)$	$(?,\alpha,?) \to ?$		1
4	$(?,?,\alpha,?)$	$(?,?,\alpha) \to ?$		1
	$(?,?,?,\alpha)$			

Using Matsui's rule of thumb, we can determine how many known plaintexts we expect to need to carry out a linear attack based on this worst-case assumption.

Cycles	Bias	Known Plaintexts
2	2^{-25}	2^{53}
3	2^{-37}	2^{77}
4	2^{-49}	2^{95}

Note that a slightly different worst-case assumption can lead to much more risk of successful attack. For example, if we assume that the worst case is for all six S-boxes to have the same approximation with bias of 2^{-2}, then we get an F-function approximation with bias 2^{-7}, which leads to the following situation:

Cycles	Bias	Known Plaintexts
2	2^{-13}	2^{29}
3	2^{-19}	2^{41}
4	2^{-25}	2^{53}
5	2^{-31}	2^{65}
6	2^{-37}	2^{77}
7	2^{-43}	2^{89}

Lessons from the analysis

1. Even though a given UFN structure may be inherently susceptible to linear attacks, its F-function may still provide enough resistance to prevent successful linear cryptanalysis.
2. For this kind of F-function, it makes sense to try to find the worst-case one-cycle linear approximation based on an extremely unlikely set of S-boxes with the same linear approximation, all with reasonably high probability.

This article was processed using the LaTeX macro package with LLNCS style