Key-Schedule Cryptanalysis of IDEA, G-DES, GOST, SAFER, and Triple-DES

John Kelsey Bruce Schneier David Wagner

Counterpane Systems
101 E. Minnehaha Parkway
Minneapolis, MN 55419
{kelsey,schneier}@counterpane.com

U.C. Berkeley
C.S. Div., Soda Hall
Berkeley, CA 94720-1776
daw@cs.berkeley.edu

Abstract. We present new attacks on key schedules of block ciphers. These attacks are based on the principles of related-key differential cryptanalysis: attacks that allow both keys and plaintexts to be chosen with specific differences. We show how these attacks can be exploited in actual protocols and cryptanalyze the key schedules of a variety of algorithms, including three-key triple-DES.

1 Introduction

A key schedule is an algorithm that expands a relatively short master key (typically between 40 and 256 bits long) to a relatively large expanded key (typically several hundred or thousand bits) for later use in an encryption and decryption algorithm. Key schedules are used in several ways:

a. To specify the round keys of a product cipher. DES [NBS77] uses its key schedule in this way, as do many other product ciphers.
b. To initialize some fixed elements of a cryptographic transform. Khufu [Mer91], Blowfish [Sch94], and SEAL [RC94] use a key schedule this way.
c. To initialize the state of a stream cipher prior to generating keystream. RC4 [Sch96] uses a key schedule in this way.

Note that (b) and (c) are the only instances where synchronous stream ciphers can fall prey to any chosen-input attack.

In this paper, we present new attacks on key schedules. In Section 2, we summarize existing cryptanalysis of key schedules and extend some of those results. In Section 3, we introduce and motivate related-key cryptanalysis. Section 4 describes several new related-key attacks against a variety of algorithms. Finally in Section 5, we briefly discuss the design of good key schedules.

2 Attacks on Key Schedules

In this section, we present several previous attacks on key schedules. While many of these attacks cannot break the underlying algorithms in all forms, they show a "theoretical weakness" which may be exploited in certain circumstances.

2.1 Meet-in-the-Middle Attacks

Meet-in-the-middle attacks occur when the first part of a cipher depends upon a different set of key bits than does the second part. This allows an attacker to attack the two parts independently, and works against double-encryption with a block cipher and two different keys [MH81, OW91, OW95].

2.2 Linear Factors

A linear factor is a fixed set of key bits whose complementation leaves the XOR of a fixed set of ciphertext bits unchanged; this weakness can be used to speed up an exhaustive key search. Six-round DES has a linear factor [CE86].

2.3 Weak Keys

A weak key, K, is a key for which encryption is the same function as decryption. A pair of semi-weak keys, K and K', are keys for which encryption with K is the same as decryption with K' and vice versa. Both DES and LOKI89 have weak keys [Dav83, Cop86, MS87, Knu93a].[1] If the number of weak keys is relatively small, they may not compromise the cipher when used to assure confidentiality. However, several hash modes use block ciphers where an attacker can choose the key input in an attempt to find a collision; in these modes the block cipher should not have any weak or semi-weak keys.

2.4 Detectable Key Classes

One way to reduce the effective keyspace is to divide the keyspace into classes, and then find an attack that reveals to which class the key belongs. In some instances, the workload of identifying a key with a specific class is very small; these too are sometimes referred to as weak keys.

For example, certain keys in Blowfish [Sch94] result in identical S-box entries, and can be detected in reduced-round variants [Vau96]. IDEA [LMM91] has several classes of keys detectable with just two chosen-plaintext encryptions [DGV93]. The key schedule in LOKI91 allows two different keys to have several round keys in common; this reduces the effective keyspace by almost a factor of four using 2^{33} chosen plaintexts [Knu93b]. Due to the weak mixing in its key schedule, RC4 has a class of detectable keys [Roo95]. One out of 256 keys are detectable, and a detectable key has about a 13.8% chance of revealing 16 bits of the key in the first output byte. Lucifer has differential characteristics which are conditional on the key [BB93].

[1] It is interesting to note that the weak keys in DES were instrumental in proving that the cipher is not a group [CW93].

RC5 The following is a new detectable key class attack on RC5 [Riv95]. Suppose all but the first pair of RC5 subkeys has only 0 bits in their low 5 bit positions. This is a weak key, because it is possible (with 1024 tries) to get a plaintext value that never rotates. To check for a weak key of this kind, let $X[i]$ be one of the 1024 chosen 64-bit values which differ in the low five bits of each word. Let $D = (c, 0, 0, 0, d, 0, 0, 0)$, where c and d are random bytes. Send in the pairs

$$X[i], X[i] \oplus D$$

for $i = 0$ to 1023. If we have a weak key of this kind, we get some i whose output XOR is $(e, 0, 0, 0, f, 0, 0, 0)$ where e and f are random bytes. This has to be the case if we don't get any rotation, since nothing else in the cipher could propagate to the right. Any rotation, however small, anywhere, will destroy this property.

If the subkey expansion function were random, we would have a probability of choosing a weak key of 2^{-10R}, where R is the number of weak-subkey rounds. (Rivest defines one "round" of RC5 as two rounds, i.e. applications, of the Feistel function.) For $R = 4$, this is large enough to be of concern; for $R = 12$ the chances of getting a weak key at random are about 2^{-120}. However, we need to worry about this attack if we try to build hash functions out of RC5 with almost any R.

2.5 Simple Relations and Equivalent Keys

A simple relation occurs between two different keys, manifesting itself as a relationship between the resulting plaintexts and ciphertexts [Knu94]. This also allows the keyspace to be reduced in a search. DES and LOKI have a simple relation known as the complementation property: if K encrypts P to C, then the bitwise complement of K encrypts the bitwise complement of P to the bitwise complement of C. This reduces the effective keyspace by one bit. DES and LOKI have pairs of keys for which a simple relation exists, for at least a fraction of all plaintexts [Knu95a]. Kwan and Pieprzyk describe a technique for locating complementation properties in linear key schedules [KP93].

Two keys are equivalent if they transform all plaintexts identically. This can be considered a special kind of simple relation.

TEA TEA [WN95] is an easy-to-memorize Feistel block cipher. It has a 128-bit master key $K_{0..3}$, and the key schedule is simple: all odd rounds use K_0, K_1 as the round subkey, and all even rounds use K_2, K_3.[2] One cycle of TEA applied to the block Y_i, Z_i performs

$$Y_{i+1} \leftarrow Y_i + F_i(Z_i, K_0, K_1) \qquad\qquad Z_{i+1} \leftarrow Z_i + F_i(Y_{i+1}, K_2, K_3)$$

[2] The round function is perturbed slightly by a different constant for each round to avoid trivial attacks.

where the round function $\mathtt{F}$ is

$$F_i(z, k, k') = (\text{ShiftLeft}(z, 4) + k) \oplus (z + C_i) \oplus (\text{ShiftRight}(z, 5) + k')$$

Consider complementing the most significant bits of K_0 and K_1. Note that flipping the most significant bit propagates through both the addition and XOR operations, and flipping it twice cancels the modification. Therefore, modifying the 128-bit master key in this way does not effect the encryption process. We can also complement the most significant bits of K_2, K_3 without any effect. This means that each TEA key has 3 other equivalent keys. In particular, it is easy to construct collisions for TEA when used in a Davies-Meyer hashing mode [Win84].

2.6 Attacks on One-Wayness

A key schedule is one-way if, given several round subkeys, it is infeasible for an attacker to gain any new information about the master key or about other unknown round subkeys. For instance, recovering a few round subkeys allows one to recover most of the master key in the DES key schedule; Biham and Shamir exploited this to optimize their differential attack on DES [BS93b]. Furthermore, it may be easier to find weak keys and related keys for key schedules which are not one-way.

3 Introduction to Related-Key Cryptanalysis

A related-key attack is one where the attacker learns the encryption of certain plaintext not only under the original (unknown) key K, but also under some derived keys $K' = f(K)$. In a chosen-related-key attack, the attacker specifies how the key is to be changed; known-related-key attacks are those where the key difference is known but cannot be chosen by the attacker. We emphasize that the attacker knows or chooses the relationship between keys, but not the actual key values.

3.1 Overview of General Techniques

The simplest related-key attack treats the cipher as a black box. Winternitz and Hellman show that by obtaining the encryption of a single chosen plaintext under 2^n chosen keys, $n \leq k$, one may recover the key values with 2^{k-n} offline trial encryptions, if the cipher uses k-bit keys [WH87]. This attack easily extends to a probabilistic known-key attack with similar complexities, and shows that any cipher has strength of at most $2^{k/2}$ against the naive black box attack, if related-key queries are not much more expensive than chosen-plaintext queries.

Biham introduced a form of related-key cryptanalysis of product ciphers based on rotating the round subkeys [Bih94]. Grossman and Tuckerman have described an attack on Feistel ciphers with identical subkeys in each round [GT78].

With hindsight, the ideas used in their known-plaintext attack seem similar to the general approach of Biham's related-key attacks, though their paper predated Biham's by 16 years. Others have used similar techniques to exploit key schedule weaknesses in chosen-plaintext attacks [Knu93b].

Let SK_i be the ith round subkey generated from master key K under the key schedule. Biham's attack takes advantage of a related key-pair (K, K') such that $SK_i = SK'_{i+1}$ for nearly all i, by noting that the action of rounds 1 to $r-1$ with master key K is equivalent to the action of rounds 2 to r with master key K'. This method is successful against LOKI and Lucifer [Bih94].

One very useful cryptanalytic technique considers a differential attack in which the keys, as well as the plaintexts, are chosen with specific differences. Given a vulnerable key schedule, we can often insert a chosen difference into the middle of a cipher, rather than having to pass through all rounds of the cipher. This formulation appears to be novel, though some earlier work [Knu95b] implicitly used this type of method.

3.2 Motivation and Impact

Related-key cryptanalysis has commonly been considered a powerful, but strictly theoretical, attack. We believe that this view is inaccurate and that related-key cryptanalysis is of practical interest as well. Certain real-world cryptographic implementations may allow for related-key cryptanalysis. Secure communications protocols sometimes use K to encrypt in one direction and $\overline{K}$ in the reverse direction. At least one message encryption program uses K, $K+1$, $K+2$, etc., to encrypt a series of messages.[3] Implementations like these inadvertently allow for related-key attacks.

We admit that the most obvious method for performing related-key attacks is rather impractical. In the straightforward approach, an adversary (an insider, perhaps?) must somehow manage to change the key in a predictable manner; only then will it be possible to mount a related-key query. The adversary is assumed to have write-access to the key, but not read-access to the key or the protected plaintexts. Because of these strong (and strange) requirements, this naive scenario should not be a concern in most practical applications.

However, it is not safe to dismiss related-key cryptanalysis on the basis of this analysis alone. The following sections explain why we believe that related-key vulnerabilities can yield practical attacks on real cryptographic applications.

3.3 Attacks on Key Exchange Protocols

We present a new attack on key exchange protocols which may allow an adversary to mount related-key queries on the underlying cipher. Suppose Alice and Bob are attempting to exchange a session key for bulk encryption; after the protocol

[3] These implementations are proprietary, and no references are available.

run Alice receives a secret K_A and Bob receives a secret K_B, and in a successful run $K_A = K_B$. An adversary may attack this setup by tampering with the key-exchange protocol messages, so that K_A, K_B have a known or chosen relation (but $K_A \neq K_B$, in general). After this attack, when Alice encrypts plaintext to Bob, Bob will not receive the correct message but will find random garbage upon decryption. Under a known-plaintext assumption, the adversary can receive the decryption of C under both K_A and K_B, for many random values of C. This is not an unreasonable assumption; we can easily imagine that Bob may be willing to disclose his garbage (decrypted plaintext) for debugging purposes when the key exchange fails. Furthermore, if the adversary can modify ciphertext en route it is possible to mount differential attacks with control over the difference entering the key and the ciphertext ports. In the end, a key-exchange protocol is vulnerable to this attack if it does not provide integrity protection for the shared secret it exchanges. For example, a protocol which distributes a session key K_s between Alice and Bob, who share a long-term key K_{ab}, by sending

$$A \to B : E(K_{ab}) \oplus K_s$$

will be vulnerable since an adversary can flip bits in K_s at will.

To demonstrate this protocol attack, we examine the 2PKDP key distribution protocol [TH93]. The 2PKDP protocol is

$$\begin{aligned} &A \to B : A, \quad N \\ &B \to A : N', \ MAC_{K_{ab}}(N, N', B) \oplus K_s \end{aligned}$$

where MAC is a cryptographic message authentication code, N, N' are nonces, K_{ab} is a long-term symmetric shared key between A and B, and K_s is the negotiated session key. We note that an adversary can easily flip bits in the second message to flip corresponding bits in K_s and thereby mount one related-key query. It is interesting to note that the authors (incorrectly) believed that key integrity was not a necessary goal for a key distribution protocol.[4] Our attack shows that, when the cipher is vulnerable to related-key attacks (as most are), key integrity is vital.[5]

Though these examples have used symmetric-key protocols, one should not assume that a public-key protocol is inherently safer. For instance, if session keys are encrypted under an RSA key (e, n) without completely padding all the unused input bits, an adversary can shift the key over j bits by multiplying the RSA ciphertext by $2^{ej} \bmod n$; this shift may create exactly the sort of key difference needed for a key rotation attack.[6]

[4] Furthermore, their optional protocol extension for key integrity can fail when there are related keys K, K' such that $E_K(P) = E_{K'}(P)$ holds with high probability.

[5] Other researchers, e.g. [MB94], have also noted the need for message integrity protection in key distribution protocols.

[6] Standard public key schemes such as PKCS [PKCS] and Bellare-Rogaway [BR94] use padding and thus are not vulnerable to this sort of attack.

Key exchange protocols may also be vulnerable when they transmit some key data as salt in the clear (perhaps to avoid dictionary attacks), since this practice gives an adversary some control over the input to the bulk cipher key schedule.

As stated, the protocol attacks provide only one related-key query. To obtain multiple related-key queries, the adversary could try to attack several sessions; but all those failed key exchanges will probably be noticed, so it may be very difficult to obtain too many related-key queries. Alternatively, attacking n-party key distribution (such as secure conferencing and multicasting) protocols may yield up to $n-1$ related-key queries; again, this is not practical for large values of n. In this context, then, we can conclude that related-key queries should be considered very expensive (though not infeasible) to obtain. A block cipher which succumbs to an attack requiring just one related-key and 2^{32} known plaintexts should be very worrisome; an attack needing 2^{32} chosen keys seems impractical.

3.4 Attacks on Hash Functions

Probably the most relevant context for related-key cryptanalysis is the design of hash functions. Typically, hash functions are constructed from an underlying block cipher; often a Davies-Meyer type structure is used, so that the message input feeds into the block cipher's key. This setup is ideal for related-key cryptanalysts: a related-key query on the block cipher requires just one off-line chosen-message trial hash computation in this model, so a related-key weakness in the block cipher could easily weaken the hash function severely. In other words, for hash functions, related-key queries are cheap—usually cheaper than chosen-plaintext queries. Accordingly, hash functions must rely on the underlying block cipher to have an exceedingly strong key schedule.

4 New Related-Key Attacks

4.1 GOST

GOST [GOST89] is an excellent example of a cipher that was designed to resist rotating-subkey related-key attacks but not related-key differential cryptanalysis. Let $K_{0..7}$ be the eight 32-bit words of the key. GOST is a 32-round Feistel cipher; the key schedule generates round subkeys $sk_{0..31}$ according to

$$sk_i = \begin{cases} K_{i \bmod 8} & \text{if } i < 24 \\ K_{7-i \bmod 8} & \text{otherwise} \end{cases}$$

Note that a non-zero ΔK_0 difference gets introduced into only sk_0, sk_8, sk_{16}, and sk_{31}; the difference Δsk_0 can be handled by the standard trick of offsetting the first round key difference with an appropriately chosen plaintext difference. This approach allows us to bypass the first eight rounds for free.

GOST's F-function is based on the parallel application of eight 4-bit wide bijective S-boxes, followed by a left rotation of 11 bits. This means that it is possible to attack GOST with a triple of one-round characteristics $A \to B$ by F, $B \to C$, and $C \to A$ (here A, B, C are 32-bit differences), where each one-round characteristic has just one active S-box.[7] In more detail, let A be an input difference with just one active S-box and with only the high-order bit set in the input difference to that S-box. Choose B so that $A \to B$ and the low bit of active S-box's output difference is zero. The round function rotates the output difference left by 11 bits, which puts the three non-zero bits of B into one S-box's input in the second round. This second round is covered with $B \to C$ by a similar technique; the active S-box's output difference should be zero in the high three bits, so that we can cover the third round with $C \to A$.[8]

If the probability of all three of these relationships is p, and they are allowed to overlap (which is reasonable, as A, B, and C are each no more than four bits wide), then we can choose a key difference of A, get a 20-round characteristic with probability p^{32}, and carry out a 4R attack on 32-round GOST. This analysis will only be practical when the S-boxes have a very bad difference distribution, since when $p < 2^{-2}$, $p^{32} < 2^{-64}$. Against a 24-round variant of GOST, this attack becomes more practical. (A 24-round variant of GOST should have a key schedule just like full GOST, but with one of the intermediate eight-round sequences removed. The last eight subkeys should still be reversed from the first eight subkeys.) Suppose $A \to B$ with probability p_0, $B \to C$ with p_1, and $C \to A$ with p_2; then we get a 12-round characteristic with probability $p_0^6 p_1^6 p_2^7$, allowing us to mount a 4R attack on the 24-round GOST variant.

We examined the security of the "standard" set of GOST S-boxes [Sch96]: we estimate that the 12-round characteristic has probability about 2^{-68}, which is too low to make the attack practical. Randomly chosen S-boxes were much weaker. The average probability over 10000 random S-boxes was 2^{-54}; other values were in the range 2^{-43} to 2^{-80}, with a large variation between trials.

4.2 IDEA

IDEA [LMM91] is an eight-round block cipher with a 64-bit block and a 128-bit key. Each round key is 96 bits long; there is also a final output transformation which uses a 64 bit round key. IDEA's key schedule is very simple. To generate $6n + 4$ 16-bit subkey words for a n-round IDEA variant, a 128-bit register is filled with the key. The first 16 bits are taken for the first subkey word, the next for the second, and so on, until all eight 16-bit words in the register are taken. The register is then rotated left 25 bits, and the process is repeated until all the required subkey words have been taken.

Because of the simple key schedule, there is a chosen-key differential attack on 3-round IDEA. This attack can recover 32 bits of key using six chosen plaintexts—two under the first key, four under the second. Recovering another 64

[7] The analysis is complicated by the fact that the GOST S-boxes may vary in different implementations, so we will describe a general approach for cryptanalysis.

[8] Note that there are many other possible ways to push characteristics through GOST.

bits of the key requires another 2^{17} chosen plaintexts under the third key, leaving us with 32 bits to recover by exhaustive search.

There is also a chosen-key ciphertext-only timing attack on full eight-round IDEA. It requires 5×2^{17} related-key queries, each used to encrypt 2^{20} random, unknown plaintext blocks. We must be able to measure the time to perform each aggregate of 2^{20} encryptions to an accuracy of one second.

The Attack on 3-Round IDEA We first observe that the only subkeys used in IDEA-3 that are different for K and for $K \oplus 2^{103}$ are subkeys 1 and 15 (numbering from 0). This allows us to recover 32 bits of the key by encrypting A_0 and A_1 under K, and $A_0 + 2^9$, $A_0 - 2^9$, $A_1 + 2^9$, and $A_1 - 2^9$ under $K \oplus 2^{103}$. Depending on the initial value of key bit 103, one or the other adjustment to A_0 and A_1 will cancel out the change to subkey 1. For either the $+2^9$ or -2^9 values, there will usually be only one pair of values for subkeys 18 and 20 that are consistent with the output differences.[9]

Extending the attack requires encrypting about 2^{16} different quadruples under $K \oplus 2^{127}$, corresponding to the possible different values of subkey 1. For one of these quadruples, we will usually have only one pair of values for subkeys 19 and 21 consistent with the output differences. From this, not only have we learned some information about the value of subkey 1, we can also carry out a further attack to recover subkeys 16 and 17. With knowledge of the last four subkeys, we can look at the inputs to the MA-box in the last round. We can then verify, for each of the 2^{32} possible values for subkeys 16 and 17, whether these subkey values are consistent with the input and output differences, which we can see directly. We will usually find only one such value for these two subkeys.

Related-Key Differential Timing Attacks [Koc96] cryptanalyzes several important cryptosystems by timing their operation. We present a relatively simple chosen-key timing attack on IDEA. It exploits the fact that the total time to encrypt a large number of random blocks of data with IDEA is a function of the total number of zero multiplicative subkeys.[10]

On a 33 MHz 486SX, we have measured that increasing the number of zero multiplicative subkeys by one decreases the time required to carry out 1,000,000 block encryptions by an average of 3 seconds. This allows an attacker to determine whether a given change in a key has caused the total number of zero subkeys to go up, go down, or stay the same.

The IDEA attack takes advantage of this observation, as follows. First, request the encryption of about 2^{20} random plaintexts (such as we would expect to

[9] The simplest way to implement this attack is simply to try all 2^{32} candidate values for subkeys 18 and 20 together, and look for the values that have the same XOR difference between the MA-box outputs under K and under $K \oplus 2^{103}$.

[10] There is also an adaptive chosen plaintext timing attack on full eight-round IDEA which requires no related key queries, but it is considerably more complex and demands many chosen plaintexts.

get from CBC, OFB64, or CFB64 modes) under key K, and record the elapsed time to within one second[11] as T_0. Continue to record similar timings T_i for each key $K \oplus i$ for all 16-bit values of i. Next, fix a 128-bit value MASK which has all but the low 16 bits set, and request the timings T'_i for each key $K \oplus \mathrm{MASK} \oplus i$ for all 16-bit values of i. If the low 16 bits of K are denoted by k, then the timing T_k will be the smallest of the T_i's, and also T'_k will be smaller than any other T'_i. Therefore, by searching for a k with this property, we can recover one subkey word from K. We repeat the process to recover 80 bits of K and exhaustively search over the remaining 48 unknown bits.[12]

4.3 SAFER K-64

SAFER K-64 is a 6-round block cipher whose round function combines the input with a round subkey, applies eight parallel 8-bit permutation S-boxes, combines the result with another subkey, and ends with a diffusion layer [Mas94]. The key schedule rotates the ith master key byte and adds a constant to obtain the ith byte of each round subkey; therefore master key byte i affects only the input and output of S-box i in every round. Knudsen shows that this regularity in the SAFER K-64 key schedule causes serious weaknesses in the block cipher [Knu95b]. He finds a related-key attack which has a 1/73 probability of recovering 8 key bits given one chosen related-key query and 2^{30}–2^{36} chosen plaintexts.

We believe that Knudsen's differential attack can be optimized by using standard ideas from differential cryptanalysis [BS93a]. Adopt a 1R attack, so that the differential characteristic covers rounds 2–5; the first round can be bypassed by using structures of 2^8 plaintexts, as Knudsen suggested in [Knu95b]. This 4-round characteristic has probability approximately 2^{-23} to 2^{-28} (when the key difference is favorable, which happens about 1/73 of the time) and S/N ratio of at least 2^{11}. If just one chosen related-key query is available, we expect this to have about a 1/73 probability of recovering about 28 key bits after 2^{24}–2^{29} chosen plaintexts.[13] With a structure of 2^8 chosen related-key queries, we predict this will find about 28 key bits (with very high probability) after approximately 2^{24}–2^{29} total chosen plaintexts.

SAFER K-64 is weak against related-key cryptanalysis because its key schedule has little avalanche and generates all round subkeys in nearly the same way. SAFER SK is a variant with an updated key schedule, designed to increase its avalanche qualities; it is not vulnerable to the related-key attack described above [Knu95b].

[11] These numbers reflect the machine and implementation used for the experiment—other machines and implementations may change these significantly. For example, improved time resolution will decrease the number of ciphertexts needed.

[12] Alternatively, one could continue to recover multiplicative subkeys in this way until all 128 bits of K are known.

[13] The 1R attack allows one to extract more information about the last round subkey from a right pair than Knudsen's 0R attack does.

4.4 DES with Independent Round Subkeys

A 768-bit DES variant uses independent round subkeys [Ber83]. This variant will be much weaker in some situations: there is a very simple related-key attack needing just 15 related keys and 60 chosen plaintexts. Obtain the encryptions $E(k,p)$ and $E(k',p)$, where k' is obtained from k by flipping some bits in the last round subkey; this can be thought of as a differential 1R attack with a characteristic of probability 1. The last round subkey can be recovered with four chosen plaintexts, and then we can peel off the last round and repeat the attack on 15-round DES. This attack can also be optimized for the case when related key queries are very expensive to achieve a complexity of one related key and 2^{16} or so chosen plaintexts. For nearly any product block cipher, if it's possible to flip bits in a cipher's expanded key, it's possible to mount an XOR differential attack on the last round of the cipher. This may be useful in attacking some systems that leave expanded keys vulnerable to change.

4.5 G-DES

G-DES [PA90a, PA90b] is a DES variant with independent round subkeys. The authors foresee difficulty if it is used for hashing (such as a Davies-Meyer construction), so they specify two special modes for use as a one-way function. Both modes use 16 G-DES encryptions, with the 768-bit message block as the key; the first mode iterates encryptions serially to an initial constant plaintext block, while the second mode applies them in parallel to 16 initial constants. Both hashing modes can be broken by a trivial related-key differential attack which modifies part of the last round subkey: at most 2^{32} trial encryptions suffice to find collisions. The authors did not envision any difficulty using G-DES for confidentiality, but our analysis of DES with independent round subkeys show that G-DES is also vulnerable to related-key attacks in these applications.

4.6 Three-Key Triple-DES

Three-key triple-DES is a well-known method for strengthening DES with a 168-bit key; it is also susceptible to related-key attacks. This mode can be considered a 3-round cipher with independent 56-bit round subkeys, realizing that each round is very strong. Naively, one might use rotational related-key cryptanalysis; however, such an approach would require many known plaintexts.

There is a better related-key attack on triple-DES. Denote the triple-DES encryption of P under key $K = (k_a, k_b, k_c)$ by

$$C = E(k_c, E^{-1}(k_b, E(k_a, P))).$$

A related key-pair will be

$$K = (k_a, k_b, k_c) \qquad K' = (k_a \oplus \Delta, k_b, k_c),$$

where Δ is an arbitrary fixed known constant. Once we have a known-plaintext pair[14] P, C for the key K, we obtain the decryption of C under key K' to obtain $P' = E^{-1}(k_a \oplus \Delta, E(k_a, P))$; now exhaustive search will recover k_a in 2^{56} encryptions. After that, one can find k_b, k_c with a meet-in-the-middle attack on double-DES, which has complexity of approximately 2^{56} to 2^{72} [MH81, OW91, OW95]. In total, we need one chosen related-key query, one chosen-ciphertext query, and 2^{56}–2^{72} offline trial encryptions.

Note this attack does not work against two-key triple-DES, and is the first attack for which two-key triple-DES is stronger than three-key triple-DES.

4.7 ECB+OFB

Matt Blaze proposed a double-DES variant [Bla93], as follows:

$$M_{n+1} = E(K_1, M_n) \qquad M_0 = IV$$
$$C_n = E(K_2, M_n \oplus P_n)$$

It was intended as a stronger replacement for single DES for use in a cryptographic filesystem.

This scheme is also vulnerable to related key attack. Obtain a known plaintext pair P, C encrypted under key $K = (K_1, K_2)$, and obtain the chosen ciphertext decryption P' of C under the related key $K' = (K_1 \oplus \Delta, K_2)$, where Δ is a known constant. Since $P' \oplus P = \mathrm{OFB}(K_1, IV) \oplus \mathrm{OFB}(K_1 \oplus \Delta, IV)$, exhaustive search will recover K_1 with 2^{56} trial encryptions, and then K_2 follows with another exhaustive search. In total, the attack requires one related-key query, one chosen-ciphertext query, and 2^{57} trial encryptions.

Note that Blaze's encrypting filesystem [Bla94] actually implements a modification of this mode; the deployed mode is resistant to our attack.

5 Designing Strong Key Schedules

Two basic approaches prevent related-key attacks on a cryptosystem. First, one may provide protection in the high-level protocol. To this end, we propose the following explicit design principle: *key exchange protocols should guarantee the integrity (not just the confidentiality) of exchanged keys.*

In the second approach, one prevents related-key attacks against the cipher by immunizing the key schedule against related-key attacks. To that end, we recommend that designers *maximize avalanche in the subkeys and avoid linear key schedules.* Every key bit should affect nearly every round, if possible, but not in exactly the same way; also the key schedule should be designed to resist differential attacks. This type of approach was adopted by SEAL [RC94] and Blowfish

[14] An anonymous reviewer has pointed out that the known-plaintext attack can be converted to a ciphertext-only attack with knowledge of a faulty P', C pair.

[Sch94], and suggested in [Knu94], and resulted in strong key schedules. The key schedules of FEAL [SM88] and RC5 [Riv95] also resist related-key cryptanalysis. As an open question, we note that the DES key schedule is linear, and wonder why it appears to resist related-key attacks.

If you are already using a key schedule that may be vulnerable, we recommend the following design principle: *pass the key through a cryptographically strong hash function before sending it to the key schedule.* Of course, this technique will make the algorithm less attractive for use as a hash function.

Ideally, designers should adopt both the protocol-level and cipher-level approaches simultaneously, providing defense in depth.

6 Conclusions

Related-key cryptanalysis is well-known to be extremely powerful. We have extended this technique to include differential cryptanalysis, and have shown weaknesses in many published algorithms. Furthermore, we have shown how these weaknesses can be exploited in real-world situations. Even though these attacks are not generally applicable in situations where random keys and a secure key distribution system are used, we regard related-key cryptanalysis as an important tool for both the cryptanalyst's and the key-schedule designer's workbench.

References

[BR94] M. Bellare and P. Rogaway, "Optimal Asymmetric Encryption—How to Encrypt with RSA," *Advances in Cryptology—EUROCRYPT '94*, Springer-Verlag, 1995, pp. 92–111.

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

[Ber83] T.A. Berson, "Long Key Variants of DES," *Advances in Cryptology: CRYPTO '82*, Plenum Press, 1983, pp. 311–313.

[Bih94] E. Biham, "New Types of Cryptanalytic Attacks Using Related Keys," *Advances in Cryptology—EUROCRYPT '93*, Springer-Verlag, 1994, pp. 398–409.

[BS93a] E. Biham and A. Shamir, *Differential Cryptanalysis of the Data Encryption Standard*, Springer-Verlag, 1993, pp. 187-199.

[BS93b] E. Biham and A. Shamir, "Differential Cryptanalysis of the Full 16-round DES," *Advances in Cryptology—CRYPTO '92*, Springer-Verlag 1993, pp. 487–496.

[Bla93] M. Blaze, "A Cryptographic File System for UNIX," *1st ACM Conference on Computer and Communications Security,* ACM Press, 1993, pp. 9–16.

[Bla94] M. Blaze, "Key Management in an Encrypting File System," *Proceedings of the 1994 USENIX Summer Tech. Conference,* June 1994.

[CW93] K.W. Campbell and M.J. Wiener, "DES is Not a Group," *Advances in Cryptology—CRYPTO '92*, Springer-Verlag, 1993, pp. 512–520.

[CE86] D. Chaum and J.-H. Evertse, "Cryptanalysis of DES With a Reduced Number of Rounds," *Advances in Cryptology—CRYPTO '85*, Springer-Verlag, 1986, pp. 192–211.

[Cop86] D. Coppersmith, "The Real Reason for Rivest's Phenomenon," *Advances in Cryptology—CRYPTO '85*, Springer-Verlag, 1986, pp. 535–536.

[DGV93] J. Daemen, R. Govaerts, and J. Vanderwalle, "Block Ciphers Based on Modular Arithmetic," *Proceedings of the 3rd Symposium on State and Progress of Research in Cryptography*, 1993, pp. 80–89.

[Dav83] D.W. Davies, "Some Regular Properties of the DES," *Advances in Cryptology—CRYPTO '92*, Plenum Press, 1983, pp. 89–96.

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

[GT78] E.K. Grossman and B. Tuckerman, "Analysis of a Weakened Feistel-like Cipher," *1978 International Conference on Communications,* Alger Press Limited, 1978, pp. 46.3.1–46.3.5.

[Knu93a] L.R. Knudsen, "Cryptanalysis of LOKI," *Advances in Cryptology—ASIACRYPT '91*, Springer-Verlag, 1993, pp. 22–35.

[Knu93b] L.R. Knudsen, "Cryptanalysis of LOKI91," *Advances in Cryptology—AUSCRYPT '92*, Springer-Verlag, 1993, pp. 196–208.

[Knu94] L.R. Knudsen, "Practically Secure Feistel Ciphers," *Fast Software Encryption, Cambridge Security Workshop Proceedings*, Springer-Verlag, 1994, pp. 211–221.

[Knu95a] L.R. Knudsen, "New Potentially 'Weak' Keys for DES and LOKI," *Advances in Cryptology—EUROCRYPT '94*, Springer-Verlag, 1995, pp. 419–424.

[Knu95b] L.R. Knudsen, "A Key-schedule Weakness in SAFER K-64," *Advances in Cryptology—CRYPTO '95*, Springer-Verlag, 1995, pp. 274–286.

[Koc96] P.C. Kocher, "Timing Attack Cryptanalysis of Diffie-Hellman, RSA, and Other Systems," *Advances in Cryptology—CRYPTO '96*, Springer-Verlag, 1996, this volume.

[KP93] M. Kwan and J. Pieprzyk, "A General Purpose Technique for Locating Key Scheduling Weaknesses in DES-like Cryptosystems," *Advances in Cryptology—ASIACRYPT '91*, Springer-Verlag, 1993, pp. 237-246.

[LMM91] X. Lai, J. Massey, and S. Murphy, "Markov Ciphers and Differential Cryptanalysis," *Advances in Cryptology—CRYPTO '91*, Springer-Verlag, 1991, pp. 17–38.

[MB94] W. Mao and C. Boyd, "Development of Authentication Protocols: Some Misconceptions and a New Approach," *Computer Security Foundations Workshop VII*, IEEE Computer Society Press, 1994, p. 178–86.

[Mas94] J.L. Massey, "SAFER K-64: A Byte-Oriented Block-Ciphering Algorithm", *Fast Software Encryption, Cambridge Security Workshop Proceedings*, Springer-Verlag, 1994, pp. 1–17.

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

[MH81] R.C. Merkle and M. Hellman, "On the Security of Multiple Encryption," *Communications of the ACM*, v. 24, n. 7, Jul 1981 pp. 465–467.

[MS87] J.H. Moore and G.J. Simmons, "Cycle Structure of the DES with Weak and Semi-Weak Keys," *Advances in Cryptology—CRYPTO '86,* Springer-Verlag, 1987, pp. 3–32.

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

[OW91] P.C. van Oorschot and M.J. Wiener, "A Known-Plaintext Attack on Two-Key Triple Encryption," *Advances in Cryptology—CRYPTO '90*, Springer-Verlag, 1991, pp. 318–325.

[OW95] P.C. van Oorschot and M.J. Wiener, "Parallel Collision Search with Cryptanalytic Applications," *to appear*, 1995.

[PA90a] A. Pfitzmann and R. Abmann, "Efficient Software Implementations of (Generalized) DES," *Proc. SECURICOM '90*, Paris, 1990, pp. 139–158.

[PA90b] A. Pfitzmann and R. Abmann, "More Efficient Software Implementations of (Generalized) DES," Technical Report PfAb90, Interner Bericht 18/90, Fakultat fur Informatik, Universitat Karlsruhe, 1990.[15]

[PKCS] RSA Data Security, Inc., "Public-Key Cryptography Standard (PKCS) #1: RSA Encryption Standard," Version 1.5, Nov 1993.

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

[RC94] P. Rogaway and D. Coppersmith, "A Software-Optimized Encryption Algorithm," *Fast Software Encryption, Cambridge Security Workshop Proceedings*, Springer-Verlag, 1994, pp. 56–63.

[Roo95] A. Roos, "A Class of Weak Keys in the RC4 Stream Cipher," Vironix Software Laboratories, Westville, South Africa Sep 1995.[16]

[Sch94] 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.

[Sch96] B. Schneier, *Applied Cryptography, Second Edition*, John Wiley & Sons, 1996.

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

[TH93] G. Tsudik and E.V. Herreweghen, "On Simple and Secure Key Distribution," *1st ACM Conference on Computer and Communications Security*, Nov. 1993, pp. 49–57.

[Vau96] S. Vaudenay, "On the Weak Keys in Blowfish," *Fast Software Encryption, Third International Workshop Proceedings*, Springer-Verlag, 1996, pp. 27–32.

[WN95] D. Wheeler and R. Needham, "TEA, a Tiny Encryption Algorithm," *Fast Software Encryption, Second International Workshop Proceedings*, Springer-Verlag, 1995, pp. 97–110.

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

[WH87] R. Winternitz and M. Hellman, "Chosen-key Attacks on a Block Cipher," *Cryptologia*, v. 11, n. 1, Jan 1987, pp. 16–20.

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

[15] `http://www.informatik.uni-hildesheim.de/~sirene/lit/abstr90.html#PfAss_90`

[16] `http://www.itribe.net/CTRS/Papers/CTRS-0008.txt`