Related-Key and Slide Attacks: Analysis, Connections, and Improvements

—Extended Abstract—

Mathieu Ciet, Gilles Piret, and Jean-Jacques Quisquater

Université catholique de Louvain, Crypto Group
Place du Levant, 3 1348 Louvain-la-Neuve, Belgium
{ciet, piret, jjq}@dice.ucl.ac.be – http://www.dice.ucl.ac.be/~crypto

Abstract. In this paper we present the most important results developed in key schedule cryptanalysis during the last ten years. Namely, we deal with related-key attacks, differential related-key attacks, and slide attacks. The related-key attack is presented in a more general framework than in the paper of Biham [1]. We give two improvements to the slide attacks. Finally, the link between these attacks is studied.

1 Introduction

The most well known attacks on block ciphers are those targeting at the ciphering itself, for example the linear and differential cryptanalysis. In this paper we present attacks that are particular in the sense that they focus on the key scheduling. Firstly, the related key attack, introduced by Eli Biham in [1], secondly a variant with differential related key attack and finally the slide attack developed by Alex Biryukov and David Wagner [5, 6]. Biham's related-key attack is presented in a general framework. We also give two personal improvements to the slide attack. Finally, we emphasize on the strong link between these attacks.

2 Related-key Attacks

2.1 The chosen key attack

The basic related-key attack that we refer as "chosen key attack" relies on an unusual hypothesis: namely, encryption is performed using two unknown different keys that have a particular relationship known to the attacker. First we present the attack in a general framework; the first step is to describe the relationship that must be satisfied by the two keys K and K^*, in order to be vulnerable to Biham's attack. Then we analyze what can be improved if we are dealing with the particular case of a Feistel cipher.

The general case. We denote by n the block length, and by r the number of rounds of the cipher. We also write $K \rightarrow (K_1, K_2, ..., K_{r-1}, K_r)$ to signify that the key schedule derives the sequence of round keys $(K_1, K_2, ..., K_{r-1}, K_r)$ from the key K. Finally, we denote by $F(x, k)$ the round function applied to the data x with round key k.
To be vulnerable to Biham's attack, K and K^* must obey the following link: if key K gives rise to a certain sequence of subkeys, then key K^* must give rise to the same sequence, rotated by one round. Or more formally:

$$K \rightarrow (K_1, K_2, ..., K_{r-1}, K_r) \Leftrightarrow K^* \rightarrow (K_2, K_3, ..., K_r, K_1)$$

This condition is very restrictive, as for most of the block ciphers the sequence of subkeys $(K_2, K_3, ..., K_r, K_1)$ simply cannot be derived from a key! It is however met by the two LOKI ciphers attacked in [1]. The reason for which this relationship makes sense is the following: suppose that an attacker can identify two plaintexts P and P^* such that $P^* = F(P, K_1)$. Then the encryptions of P under key K and that of P^* under key K^* process the same way during $n-1$ rounds; as a consequence, the ciphertexts verify the same property as the plaintexts: $C^* = F(C, K_1)$. Two such plaintexts are called a **slid pair**. Identifying such a pair is not evident, as the round key K_1 is unknown to the attacker.

Suppose we know $2^{n/2}$ plaintext-ciphertext pairs (P, C) encrypted with the key K, and $2^{n/2}$ plaintext-ciphertext pairs (P^*, C^*) encrypted with the key K^*. For each pair $((P, C); (P^*, C^*))$ we try to solve the both equations: $F(P, K') = P^*$ and $F(C, K') = C^*$, until we indeed find a key K' that verifies the system. With a high probability this K' is the subkey K_1: most of the time the equations $F(P, K') = P^*$ and $F(C, K') = C^*$ cannot both hold if P and P^* do not form a slid pair, because of the structure of the round function attacked. Thus when we effectively find a solution to the system there is a high probability that we are dealing with a slid pair, and that the solution K' we found is actually K_1. The probability for a random pair to be a slid pair is 2^{-n}, so with $2 \cdot 2^{n/2}$ plaintext-ciphertext pairs we may expect to find one.
The function F must be weak enough to permit us to retrieve the key K_1. We call F a **weak** permutation if given the two equations $F(x_1, k) = y_1$ and $F(x_2, k) = y_2$ it is "easy" to extract the key k. The round function F must satisfy this definition in order to be vulnerable to the attack presented. Of course it is informal since the amount of *easiness* may vary from cipher to cipher. The complexity of this basic attack is $O(2^n)$ work, as there are $O(2^n)$ pairs to examine.

The particular case of Feistel ciphers: Known plaintext attack. In the case of Feistel ciphers, the round function $F(\langle L, R\rangle) = \langle R, L \oplus f(R)\rangle$ modifies only one half of its input. Therefore, the condition $F(x, k) = y$ can be recognized by simply comparing the right half of x against the left half of y, without knowing anything about the round key k. This filtering condition eliminates all but $2^{-n/2}$ of the pairs.
This property can be used to improve the efficiency of the attack: indeed, (P, C)

and (P^*, C^*) form a slid pair if and only if $F(P, K_1) = P^*$ and $F(C, K_1) = C^*$. We have a $n/2$-bit filtering condition for each of these equalities, and thus globally a n-bit filtering condition.
Therefore, potential slid pairs can be identified using two sorted lists with $2^{n/2}$ entries: we sort the texts (P_i, C_i) encrypted with key K based on the right half of P_i and C_i, and the texts (P_j^*, C_j^*) encrypted with key K^* based on the left half of P_j^* and C_j^*. It is then easy to look for a match between the left halves of P_j^* and C_j^* and the right halves of P_i and C_i (respectively).
With this filtering technique, we expect to find no more than one false match along with one good slid pair. The false match(-es) can be easily eliminated in a second phase. While we still need about $O(2 \cdot 2^{n/2})$ slid pairs, the time complexity is now reduced to $O(2^{n/2})$ offline work.

The particular case of Feistel ciphers: Chosen plaintext attack. In the previous section, the fact that the round function $F(\langle L, R\rangle) = \langle R, L \oplus f(R)\rangle$ modifies only one half of its input was used as a filtering condition. In the context of chosen plaintext attack, it can be used to select appropriate plaintexts.
Suppose that we choose a pool of $2^{n/4}$ plaintexts $P_i = \langle P_{Li}, P_R\rangle$ (with P_R constant) encrypted under the key K, and another pool of $2^{n/4}$ plaintexts $P_j^* = \langle P_R, P_{Rj}\rangle$ encrypted under the key K^*. Now a pair of plaintexts $(P_i; P_j^*)$ has probability $2^{-n/2}$ to be a slid pair, so we may hope to find one slid pair with only $2 \cdot 2^{n/4}$ chosen plaintexts rather than $2 \cdot 2^{n/2}$ known plaintexts; this slid pair can be recognized using the $n/2$-bits filtering condition on the ciphertexts.
Note that such a chosen text attack can only be performed when we know something about the round function, as the structure of the set of plaintexts we chose depends on the particular structure of the round function. However the two attacks we presented previously are not restricted to Feistel ciphers. It is often possible to exploit the particular structure of a given round function (especially if the cipher is more or less Feistel-like).

An attack on slightly more general key schedules, in the case of Feistel ciphers. It must be noted that the attack presented above works only if the key K^* is derived from key K by a perfect rotation of the subkeys :

$$K \rightarrow (K_1, K_2, ..., K_{r-1}, K_r) \Leftrightarrow K^* \rightarrow (K_2, K_3, ..., K_r, K_1)$$

We call this type of relationship between K and K^* the **strong assumption**.

Suppose now that the relationship between K and K^* is a bit weaker, i.e. that we only have:

$$K \rightarrow (K_1, K_2, ..., K_{r-1}, K_r) \Leftrightarrow K^* \rightarrow (K_2, K_3, ..., K_r, K_{r+1})$$

where K_{r+1} is not necessarily equal to K_1. This second type of relationship will be called **weak assumption**. For example, one can show that LOKI89 with 16 rounds satisfies the strong assumption, while if it is reduced to 11 rounds only the weak assumption is.
Under the weak assumption hypothesis, the attack we presented does not work

anymore in the general case: indeed, a slid pair $((P, C); (P^*, C^*))$ would satisfy the two equations: $F(P, K_1) = P^*$ and $F(C, K_{r+1}) = C^*$. For any pair $(x; y)$ the equation $F(x, k) = y$ has at least one solution. Therefore if K_1 and K_{r+1} share few or no bits, it is not possible to identify slid pairs by simply trying to solve this system. However we have seen that in particular cases, such as Feistel ciphers, it is possible to use filtering conditions. The following attack is thus applicable. Suppose we know $2^{n/2}$ plaintext-ciphertext pairs (P, C) encrypted with the key K, and $2^{n/2}$ plaintext-ciphertext pairs (P^*, C^*) encrypted with the key K^*. The texts (P_i, C_i) encrypted with key K, and the texts (P_j^*, C_j^*) encrypted with key K^* are sorted based respectively on the right half of P_i and C_i, and on the left half of P_j^* and C_j^*. Then we look for a match between the left halves of P_j^* and C_j^* and the right halves of P_i and C_i (respectively). Doing this, we filter the pairs $((P, C), (P^*, C^*))$ for which the distinguishing condition for $F(P) = P^*$ and $F(C) = C^*$ hold. For each of the pairs selected (we hope to find around one false alarm along with one slid pair), the two equations $F(P, K_1) = P^*$ and $F(C, K_{r+1}) = C^*$ are solved separately. Note that for certain round functions, we get many possibilities for K_1 and K_{r+1}. But if K_1 and K_{r+1} have several bits in common, the number of possibilities for (K_1, K_{r+1}) is not too big. Finally, for each suggestion for (K_1, K_{r+1}), we try to find the remaining key bits by exhaustive search, until we succeed.

2.2 The chosen plaintext attack

The hypothesis we made in the previous section looks very restrictive. Nevertheless this attack could be useful if combined with an attack on key-exchange protocols that do not guarantee key-integrity, or if bad key-update protocols are used. We are now going to use the related-key principle in a more classical context: namely, we have a certain number of chosen plaintext-ciphertext pairs encrypted under an unknown key K that we are searching for. First we show how to deal with the case of a Feistel cipher. Then we analyze if this principle can be used for the non-Feistel ciphers.
For a key K such that $K \rightarrow (K_1, ..., K_r)$, we denote K^* as being the key such that $K^* \rightarrow (K_2, K_3, ..., K_{r+1})$ (for some K_{r+1}) and K_* as being the key such that $K_* \rightarrow (K_0, K_1, ..., K_r)$ (for some K_0)[1]. Also by $P \xrightarrow{K} C$ we denote the fact that the encryption of P under key K gives ciphertext C.

The case of Feistel ciphers. The attack is as follows:

- The following plaintext-ciphertext pairs are chosen: $P = \langle P_L, P_R \rangle \xrightarrow{K} C$ for some plaintext P, $\forall a \in \{0, ..., 2^{n/2} - 1\} : P_a^* = \langle P_R, a \rangle \xrightarrow{K} C_a^*$ and $\forall a \in \{0, ..., 2^{n/2} - 1\} : P_{*a} = \langle a, P_L \rangle \xrightarrow{K} C_{*a}$

[1] We emphasize on the fact that only some particular algorithms are such that for a given K the keys K^* and K_* exist; thus the attack we present is only applicable to very specific algorithms.

- Let Φ be a set of keys such that $\Phi \cup \{k^* | k \in \Phi\} \cup \{k_* | k \in \Phi\}$ covers all the key space.
- For each key $K' \in \Phi$: Compute $P \stackrel{K'}{\rightarrow} C'$. If $C' = C$ we can guess that with a high probability $K = K'$ and stop. Else compute $F(P, K'_1)$, that equals $P^*_{a'}$ for some a' (in fact, we already did it when computing $P \stackrel{K'}{\rightarrow} C'$), and also $F^{-1}(P, K'_0) = P_{*a''}$. Finally compute one round backward and one round forward from the ciphertext C'. We obtain $C'^* = F(C', K'_{r+1})$ and $C'_* = F^{-1}(C', K'_r)$ (in fact, C'_* was already computed). Thus, if $C'^* = C^*_{a'}$, we can deduce that $K = K'^*$. If $C'_* = C_{*a''}$, we deduce that $K = K'_*$.

The underlying philosophy is the following: instead of proceeding through all the possible keys, we only deal with a part of the whole key space. For each trial encryption we make, we get two more encryptions as a bonus, by computing only two more rounds ($P_{*a''} = F^{-1}(P, K'_0)$ and $C'^* = F(C', K'_{r+1})$).

Remark 1. It is not necessarily easy to construct a set Φ, as small as possible, such that $\Phi \cup \{K^* | K \in \Phi\} \cup \{K_* | K \in \Phi\}$ covers all the key space, and such that the elements of Φ are easy to characterize. Ideally, the size of Φ would be one third of the size of the key space; in his attack on LOKI, Biham obtained a size of 3/8 of the key space.

Remark 2. This attack uses the same kind of approach that those based on complementation properties. Indeed, it is based on two equivalencies: $P \stackrel{K'}{\rightarrow} C' \Leftrightarrow F(P, K'_1) \stackrel{K'^*}{\rightarrow} F(C', K'_{r+1})$ and $P \stackrel{K'}{\rightarrow} C' \Leftrightarrow F^{-1}(P, K'_0) \stackrel{K'_*}{\rightarrow} F^{-1}(C', K'_r)$. Complementation attacks are also based on equivalencies. For example, the complementation property of DES may be written as: $P \stackrel{K'}{\rightarrow} \overline{C} \Leftrightarrow \overline{P} \stackrel{\overline{K'}}{\rightarrow} C$.
Furthermore, these attacks (related key and complementation) can be combined, as Biham did in the case of LOKI.
Note that the weak assumption is sufficient for this attack.

The general case. At first view, it might appear that the fact we are dealing with a Feistel cipher plays no role in the previous attack. Simply, the set $\{P^*_a\}_a$ is defined in a general framework as: $\{F(P, k)\}_{k \in RK}$, where P is a given plaintext and RK is the set of all possible round keys; and the set $\{P_{*a}\}_a$ is $\{F^{-1}(P, k)\}_{k \in RK}$.

However the attack is not useful if the sets of pairs $\{P^*_a\}_a$ and $\{P_{*a}\}_a$ that are needed for the attack cover all the codebook, because then it is not really interesting to discover the key, as we have a list of all plaintext-ciphertext pairs. It is generally the case with substitution-permutation structures. Moreover if the cardinality of $\{P^*_a\}_a \cup \{P_{*a}\}_a$ approaches the size of the key space, it is easier to perform an exhaustive key search.
Thus the attack is always possible, but makes sense only if the following two inequalities

$$\#[\{P^*_a\}_a \cup \{P_{*a}\}_a] < 2^n \quad \text{and} \quad \#[\{P^*_a\}_a \cup \{P_{*a}\}_a] \ll 2^{n_k}$$

are met, where n represents the block size, n_k represents the key size, and $\#[.]$ is the cardinality of a set.

2.3 Summary

The following table summarizes our discussions.

Algorithm	Hyp. on the Key Sch.	Chosen key attack	Classical attack
General Case	Strong	$\sim 2^{n/2+1}$ P/C pairs $\sim 2^n$ offline work	•$\#[\{P_a^*\}_a \cup \{P_{*a}\}_a]$ +1 P/C pairs • At best, $2^{n_k}/3$ keys to explore
	Weak	Impossible, except if K_1 and K_{r+1} share a sufficient number of key bits	
Feistel	Strong	$\sim 2^{n/2+1}$ P/C pairs $\sim n/2 \cdot 2^{n/2}$ offline work	•$\#[\{P_a^*\}_a \cup \{P_{*a}\}_a]$ +1 P/C pairs • At best, $2^{n_k}/3$ keys to explore
	Weak	$\sim 2^{n/2+1}$ P/C pairs $\sim n/2 \cdot 2^{n/2}$ offline work	

3 Differential Related-key Attacks

The basic idea of differential cryptanalysis is to perform the encryption of *pairs* of plaintexts, say $(P, P \oplus \Delta)$, with Δ a chosen difference. After a large number of rounds (typically $r-2$ or $r-3$), we hope to observe another given difference, say Δ', with a "high" probability.
The idea of differential related-key attack is not very different: simply we allow the two plaintexts P and $P \oplus \Delta$ to be encrypted with different keys, say K and $K \oplus \Delta K$: $C = F(P, K)$ and $C' = F(P \oplus \Delta, K \oplus \Delta K)$. ΔK must be chosen carefully, so as to obtain the desired difference in the subkeys.
This allows much more freedom for the attacker, as he can act on the key difference too. This way he can find attacks that are not possible otherwise. Note however that the basic hypothesis are not the same as those of differential cryptanalysis, as we assume we get an access to encryption under two (or more) related keys.

4 Slide Attacks

The slide attack was developed by Alex Biryukov and David Wagner in [5], [6]. In some sense it may be viewed as a particular case of related-key attack, but tends to be more powerful than this last, as it applies in a more classical context. However, it is applicable to a smaller range of algorithms.

4.1 Basic principle

Consider a block cipher that may be decomposed into r identical permutations $X_j = F(X_{j-1}, k)$, where X_j denotes the intermediate value of the block after j permutations. Note that F does not necessarily correspond to one single round of the cipher; it might include several rounds. For simplicity purpose, in the remaining of this paper we speak about one round to designate the function F. The idea of the attack is to "slide" a copy of the encryption process against another copy of the encryption process, so that both processes are one round out of phase. As in the case of related key attack, we obtain two equations $P' = F(P, k)$ and $C' = F(C, k)$. The only difference with related key attack is that the two encryptions have been computed under the same key. If the function F is weak enough, these two equations permit us to retrieve the key k. However identifying slid pairs is not trivial, as we cannot get the value $F(X_0, k)$. Nevertheless most of the time the equations $P' = F(P, k)$ and $C' = F(C, k)$ simply cannot hold both for some key if P and P' do not form a slid pair; this gives us a criterion to recognize slid pairs.
The attack proceeds as follows: we obtain $2^{n/2}$ known texts (P_i, C_i) (n denotes the block size), and we look for slid pairs. By the birthday paradox, we hope to find about one slid pair, thanks to which we can recover key material. In general, we expect one slid pair to disclose n key bits. If needed, we can search for a few more slid pairs or use exhaustive search to recover the rest of the key. The time complexity of the attack is $O(2^n)$, as there are $O(2^n)$ pairs to examine.

The condition for a block cipher to be vulnerable to the related key attack was that the sliding of all subkeys derived from a given key by one round (or eventually more) give rise to a sequence or subkeys that can be derived from another key. The slide attack may thus be viewed as a kind of "auto-related-key attack", as the algorithms vulnerable to it are those for which the sliding of the subkeys gives rise to the same sequence of subkeys!

4.2 Advanced slide techniques for Feistel ciphers

The following definition provides us with a good classification of the block ciphers with regard to their suitability for slide attacks: a **p-round-self-similar** cipher is a cipher whose key schedule is periodic with period p.
The basic slide attack generally deals with 1-round-self-similar ciphers (except if F includes several rounds). The goal of advanced attacks is to be able to attack p-round-self-similar ciphers, with $p > 1$. Various advanced techniques are proposed in [6], that work only in the case of Feistel ciphers. We present one of them, that we have slightly improved: the ***sliding with a twist***. This attack applies to 2-round-self-similar Feistel ciphers. We slide a decryption process against an encryption process by one round (see Fig.1). If the data after the first round of the encryption are the same as those after the decryption process, then the data after the encryption process are the same as those after the first round of the decryption process. The slid equations are thus: $\langle M', N' \rangle = \langle L \oplus f(K_0 \oplus R), R \rangle$ and $\langle L', R' \rangle = \langle M \oplus f(K_0 \oplus N), N \rangle$.

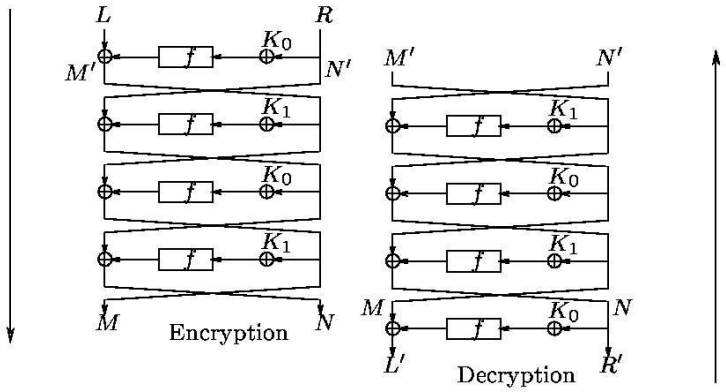

Fig. 1: Sliding with a Twist

These give us a n-bit filtering condition on slid pairs (namely $N' = R$ and $R' = N$). Thus given $2^{n/2}$ known texts, a slid pair can be found easily with a sorted list and $2^{n/2}$ work. The subkey K_0 may be easily deduced.

The method proposed in [6] in order to find the other subkey K_1 is a bit complicated: it relies on using a conventional sliding, and encrypt ciphertexts partially using K_0. The method we propose is both simpler and slightly faster: simply, we suggest to slide the decryption process in the other direction (see Fig.2)! The slid equations are now: $\langle L, R\rangle = \langle M', N' \oplus f(K_1 \oplus M')\rangle$ and $\langle M, N\rangle = \langle L', R' \oplus f(K_1 \oplus L')\rangle$. Thus the n-bit filtering condition becomes $L = M'$ and $L' = M$. This time it is the subkey K_1 that can be deduced.

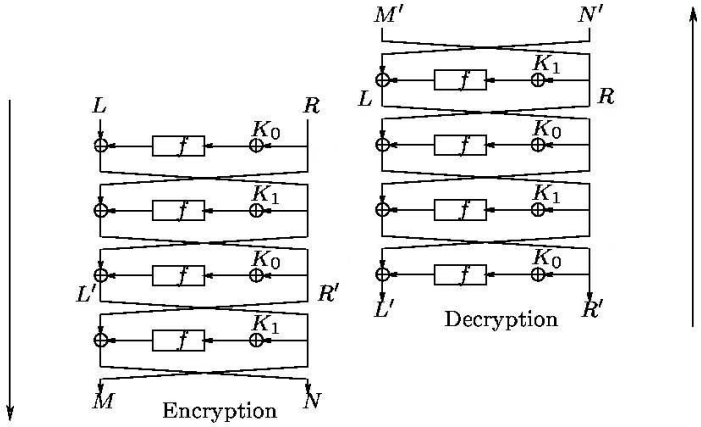

Fig. 2: Sliding with a Twist with other direction

4.3 How to deal with stronger functions?

In [6], methods to deal with functions F that need more than two pairs (x_1, y_1) and (x_2, y_2) with $F(x_1) = y_1$, $F(x_2) = y_2$ in order to be broken are presented. One approach was suggested that uses differential analysis. Suppose that the round function F has a non-trivial differential characteristic $\Delta X \rightarrow \Delta Y$ that holds with probability p. Then if we find a slid pair $(P; P')$ with $F(P) = P'$, the pair $(P \oplus \Delta X; P' \oplus \Delta Y)$ will be a slid pair too with probability p. They exploit this principle in a chosen plaintext attack, by generating a set of $3 \cdot 2^{n/2} p^{-1/2}$ chosen plaintexts such that for plaintext P in the chosen set the plaintexts $P \oplus \Delta X$ and $P \oplus \Delta Y$ are also in the set. Then it may be expected to see one pair P, P' satisfying both the slide and the differential patterns.
However the number of plaintexts needed can be improved by choosing structures of four plaintexts of the form $(P, P \oplus \Delta X, P \oplus \Delta Y, P \oplus \Delta X \oplus \Delta Y)$. It is easy to verify that for any plaintext P in the chosen set, $P \oplus \Delta X$ and $P \oplus \Delta Y$ are also in the set. In such a set with only $2^{n/2} p^{-1/2}$ plaintexts, we may hope to find p^{-1} slid pairs and one pair satisfying both the slide and the differential patterns.

If we are dealing with a set of *known* plaintexts, once a slid pair has been found, it is possible to develop strategies on the order we check the other pairs, based on the probability of the different characteristics $\Delta X \rightarrow \Delta Y$.

5 Link Between The Attacks Presented

Slide attack and differential cryptanalysis have in common that we try to find pairs of plaintext that have a certain relationship, so that there is some link between their encryption process.
The only difference between differential cryptanalysis and differential related-key attacks, is that in the second one, the pairs of plaintext are encrypted using two different keys. On the one side, it makes the attack more powerful, in the sense that it permits to find better characteristics on a larger number of rounds. On the other side, it must be noted that it relies on a more restrictive hypothesis.
Exactly in the same way, the slide attack and Biham's related-key attack distinguish by the fact that the first one deals with a single key, while the second one attacks a pair of keys. And once again, using related-key attack allows to attack a wider range of algorithms, as the key schedule do not longer need to be periodic in order to be vulnerable, but only to have a weaker property.
We could wonder if other known attacks can be turned into a related-key version. First, it must be noted that it is only applicable for attacks that deal with pairs (or bigger groups) of plaintexts. Thus linear cryptanalysis, for example, does not fit. On the contrary, there is a related-key counterpart to the differential-linear cryptanalysis. Moreover in [9], it was pointed out another example of this kind of counterpart: namely, a related-key version of the integral attack (although it is not realistic).

We have seen that Biham's related key principle can be used to reduce the complexity of an exhaustive key search. The question is: is it possible to reduce the complexity of exhaustive search using the differential related key principle?

For a given block cipher, suppose we can find differences ΔP, ΔC and ΔK such that for randomly chosen plaintext P and key K, $P \xrightarrow{K} C$ implies $P \oplus \Delta P \xrightarrow{K \oplus \Delta K} C \oplus \Delta C$ with a "high" probability p. An improved key search algorithm would be:

- Let (P, C) and $(P \oplus \Delta P, C^*)$ be two plaintext-ciphertext pairs obtained under the unknown key K.
- For each trial key K', compute $P \xrightarrow{K'} C'$. If $C' = C$, we conclude that $K = K'$ else if $C' = C^* \oplus \Delta C$, we conclude that with a probability p, $K = K' \oplus \Delta K$. Assuming $p > 2^{n_k}$ (n_k being the number of key bits), $K' \oplus \Delta K$ can be considered as a favorite key candidate. We check immediately if it is the right key by computing the encryption of P under $K' \oplus \Delta K$. If it is not the right one, we insert it in a sorted list of already tried keys.

At first glance, this algorithm looks fine, but consider the complexity analysis:

- If $P \xrightarrow{K \oplus \Delta K} C^* \oplus \Delta C$, the key search is reduced by about half of the encryptions (provided we try $K \oplus \Delta K$ before K). Knowing that $P \oplus \Delta P \xrightarrow{K} C^*$, it happens with probability p. Thus the average gain is $p \cdot 2^{n_k - 2}$ encryptions.
- The drawback is that for each trial key it is necessary to do one comparison (namely, check if $C' = C^* \oplus \Delta C$ during the first half of the key search; then for each key check if it has not been already tried [2]). The time needed to build the sorted list of already tried keys may be considered as negligible.

The average gain of our algorithm is thus $p \cdot 2^{n_k - 2}$ encryptions, while the drawback is 2^{n_k} comparisons. Thus for our algorithm to make sense, the ratio between the time for one comparison and the one for one encryption must be no more than $p/4$. But as p is most of the time very small, it is usually not the case.
The conclusion is that it is the fundamentally probabilistic nature of the differential related key attack that makes it (almost) impossible to use in order to improve exhaustive key search.

6 Conclusion

We discussed the conditions that a key schedule algorithm must satisfy in order to be vulnerable to Biham's related key attack, for a general algorithm as well as for a Feistel cipher. We then presented the slide attack, that is showed to have a strong relationship with Biham's attack. Two improvements on variants of it are presented.

We also presented another related-key attack, namely differential related-key, and discussed under which conditions a related-key counterpart could be found for other attacks.

Finally, we discussed whether these various related-key attacks can be used in reducing the complexity of exhaustive key search.

[2] We assume here that the key search begins with exploring one half of the key space, say Ψ_1 (with $\#\Psi_1 = 2^{n_k - 1}$), such that $\forall K' \in \Psi_1 : K' \oplus \Delta K \notin \Psi_1$.

7 Acknowledgements

The authors are grateful to Orr Dunkelman for valuable comments. Mathieu Ciet and Gilles Piret were supported by the European Commission through the IST Programme respectively under Contracts IST-1999-12324 and IST-1999-11159.

References

1. E. Biham. New Type of Cryptanalytic Attacks Using Related Key. In T. Helleseth, editor, *Advances in Cryptology - EUROCRYPT'93*, volume 765 of *Lecture Notes in Computer Science*, pages 229–246. Springer, 1994.
2. E. Biham and A. Shamir. Differential Cryptanalysis of DES-like Cryptosystem. *Journal of Cryptology*, 4(1):3–72, 1991.
3. E. Biham and A. Shamir. Differential Cryptanalysis of Snefru, Khafre, REDOC-II, LOKI and LUCIFER (extended abstract). In J. Feigenbaum, editor, *Advances in Cryptology - Proceedings of CRYPTO'91*, volume 576 of *Lecture Notes in Computer Science*, pages 156–171. Springer, 1991.
4. E. Biham and A. Shamir. *Differential Cryptanalysis of Data Encryption Standard*. Springer Verlag. 1993.
5. A. Biryukov and D. Wagner. Slide Attacks. In L.R. Knudsen, editor, *Advances in Cryptology - Proceedings of FSE'99*, volume 1636 of *Lecture Notes in Computer Science*, pages 245–259. Springer, 1999.
6. A. Biryukov and D. Wagner. Advanced Slide Attacks. In B. Preneel, editor, *Advances in Cryptology - Proceedings of EUROCRYPT 2000*, volume 1807 of *Lecture Notes in Computer Science*, pages 589–606. Springer, 2000.
7. L. Brown and J. Seberry. Key Scheduling in DES Type Cryptosystems. In J. Seberry and J. Pieprzyk, editors, *Advances in Cryptology - Proceedings of AUSCRYPT'90*, volume 453 of *Lecture Notes in Computer Science*, pages 221–228. Springer, 1990.
8. M. Ciet, G. Piret, and J.-J. Quisquater. A survey of key schedule cryptanalysis. Technical Report CG-2002/1, Université catholique de Louvain, Crypto Group, Available at: `http://www.dice.ucl.ac.be/crypto/techreports.html`, 2002.
9. J. Nakahara Jr, P. S.L.M. Barreto, B. Preneel, J. Vandewalle, and H.Y. Kim. SQUARE Attacks on Reduced-Round PES and IDEA Block Ciphers. Technical report. Available at `http://eprint.iacr.org/complete/`.
10. J. Kelsey, B. Schneier, and D. Wagner. Key-Schedule Cryptanalysis of IDEA, G-DES, GOST, SAFER, and Triple-DES. In N. Koblitz, editor, *Advances in Cryptology - Proceedings of CRYPTO'96*, volume 1109, pages 237–251. Springer, 1996. Lecture Notes in Computer Science.
11. J. Kelsey, B. Schneier, and D. Wagner. Related-Key Cryptanalysis of 3-WAY, Biham-DES, CAST, DES-X, NewDES, RC2, and TEA. In Y. Han, T. Okamoto, and S. Qing, editors, *Advances in Cryptology - ICICS'97*, volume 1334 of *Lecture Notes in Computer Science*, pages 233–246. Springer, 1997.
12. L. Knudsen. Cryptanalysis of LOKI. In H. Imai, R.L. Rivest, and T. Matsumoto, editors, *Advances in Cryptology - Proceedings of ASIACRYPT'92*, volume 739 of *Lecture Notes in Computer Science*, pages 22–35. Springer, 1992.
13. L. Knudsen. Cryptanalysis of LOKI91. In J. Seberry and Y. Zheng, editors, *Advances in Cryptology - ASIACRYPT'92*, volume 718 of *Lecture Notes in Computer Science*, pages 196–208. Springer, 1993.