

A Tutorial on Slide Attacks

By Chalermpong Worawannotai and Isabelle Stanton

Introduction

The slide attack is a cryptanalytic attack first created by David Wagner and Alex Biryukov in [1] in 1999. The differential attack was first introduced in 1990 and has been very successful in breaking what were once thought to be very strong block ciphers. However, it is a prevailing idea that even weak ciphers can become very strong by increasing the number of rounds and this is used to ward off a differential attack. For example, AES can be broken when it is only nine rounds, but the full strength cipher still stands unbroken.

The slide attack works in such a way as to make the number of rounds in a cipher irrelevant. Rather than looking at the data-randomizing aspects of the block cipher the slide attack works by analyzing the key schedule and exploiting weaknesses in it to break the cipher. The most common one is the keys repeating in a cyclic manner. The only requirements for a slide attack to work on a cipher is that it can be broken down into multiple rounds of an identical F function. This means it probably has a cyclic key schedule and that the F function is vulnerable to a known-plaintext attack. The slide attack is closely related to the related-key attack.

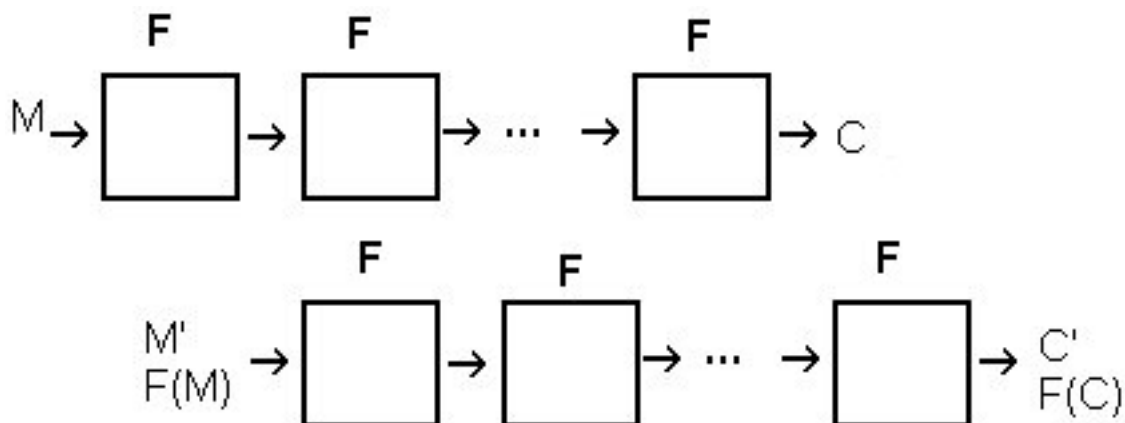
The Actual Attack

First, to introduce some notation. In this section we assume the cipher takes n bit blocks and has a key-schedule using $K_1 \cdots K_m$ as keys of any length.

The slide attack works by breaking the cipher up into identical permutation functions, F . This F function may consist of more than one round of the cipher; it is defined by the key-schedule. For example, if a cipher uses an alternating key schedule where it switches between a K_1 and K_2 for each round, the F function would consist of two rounds. Each of the K_i will appear at least once in F .

The next step is to collect $2^{n/2}$ plaintext-ciphertext pairs. Depending on the characteristics of the cipher we may need fewer, but, by the birthday paradox, we expect to need no more than $2^{n/2}$ from [1]. These pairs, which we will denote as (P, C) are then used to find a **slid pair** which we will denote (P', C') . A slid pair has the property that $P' = F(P)$ and that

$C' = F(C)$. Once we have identified a slid pair, the cipher is broken because of the vulnerability to known-plaintext attacks. The slid pair can be thought to be what happens to your message after one application of the function F ; it is 'slid' over one encryption round and this is where the attack gets its name.



The process of finding a slid pair is somewhat different for each cipher but follows the same basic scheme. One uses the fact that it is relatively easy to extract the key from just one iteration of F . You pick any pair of plaintext-ciphertext pairs and check to see what the keys corresponding to $P' = F(P)$ and $C' = F(C)$ are. If these keys match, you have found a slid pair, if not you should move on to the next pair. With $2^{n/2}$ plaintext-ciphertext pairs you should expect to find one slid pair and a small number of false-positives depending on the structure of the cipher. The false positives can be eliminated by using the keys on a different message-ciphertext pair and see if the encryption is correct. The probability that the wrong key will correctly encipher two or more messages is very low for a good cipher.

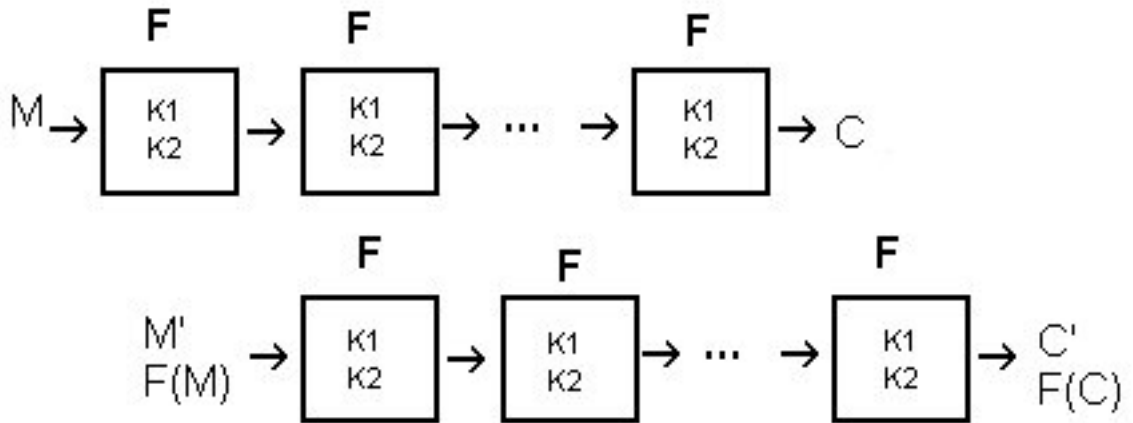
Sometimes the structure of the cipher greatly reduces the number of plaintext-ciphertext pairs needed, and thus also a large amount of the work. The clearest of these examples is the Feistel cipher using just one key. The reason for this is given a $P = (L_0, R_0)$ you must look for a $P' = (R_0, L_0 \oplus F(R_0, K))$. This reduces the possible paired messages from 2^n down to $2^{n/2}$ (since half the message is fixed) and so we will only need to collect at most $2^{n/4}$ plaintext-ciphertext pairs in order to find a slid-pair.

A Demonstration

In order to demonstrate the slide attack we have created a cipher vulnerable to it and created a software implementation. The code is attached as an index.

The cipher uses an alternating key schedule in a Feistel type scheme. These keys are called K_0 and K_1 . The *ROT* function simply rotates the bits 1 to the left. This is to prevent the keys from XORing themselves out every other round. For the purposes of ease of reading we have reduced the message size to 8 bits and each of the two keys is only 4 bits.

$$\begin{array}{ll}
 \text{Round 0} & L_0 \qquad R_0 \\
 \text{Round 1} & R_0 \qquad L_0 \oplus \text{ROT}(R_0 \oplus K_0) \\
 \text{Round 2} & L_0 \oplus \text{ROT}(R_0 \oplus K_0) \quad R_0 \oplus \text{ROT}(L_0 \oplus \text{ROT}(R_0 \oplus K_0) \oplus K_1)
 \end{array}$$



This forms the F function that a cipher vulnerable to slide attack must have. The only requirement on the number of rounds of this cipher is that it is even. If we do r rounds we apply F to M $r/2$ times. This r is independent and has absolutely no affect on the amount of work required to break the cipher and so we omit it.

In order to implement the slide attack we need pairs of message-ciphertext to extract the keys from. Assume we have (M, C) and (M', C') as a slid pair where $M = (R_0, L_0)$ and $M' = (R'_0, L'_0)$. We'd like to be able to solve the relation $M' = F(M)$ and $C' = F(C)$ for the keys.

$$(L'_0, R'_0) = (L_0 \oplus \text{ROT}(R_0 \oplus K_0), R_0 \oplus \text{ROT}(L_0 \oplus \text{ROT}(R_0 \oplus K_0) \oplus K_1))$$

We know L_0 and R_0 because $M = (L_0, R_0)$ so we need to solve to get the keys out of the cipher.

$$L'_0 = L_0 \oplus ROT(R_0 \oplus K_0) \text{ so } K_0 = ROT^{-1}(L'_0 \oplus L_0) \oplus R_0$$

Similarly

$$R'_0 = R_0 \oplus ROT(L_0 \oplus ROT(R_0 \oplus K_0) \oplus K_1) = R_0 \oplus ROT(L'_0 \oplus K_1) \text{ so } K_1 = ROT^{-1}(R'_0 \oplus R_0) \oplus L'_0$$

This provides us with an easy way to check for slid pairs because the keys are easy to extract from just one operation of the F function.

As the function takes 8 bit messages, we require $2^{8/2} = 16$ messages to find a slid pair. For this we had the computer generate and then find the slid pairs for us. Selected parts of our results as follows:

Reference	Plaintext	Ciphertext
1	00000010	00110010
2	00001011	11111101
3	00010100	01000101
4	00011101	10001010
5	00100110	10010110
6	00101111	01011001
7	00111000	10101011
8	01000001	10111111
9	01001010	01101010
10	01010011	11101101
11	01011100	00011101
12	01100101	00011011
13	01101110	11001110
14	01110111	01001001
15	10000000	00001100
16	10001001	11000011

To demonstrate the extracting of keys and testing for slid pairs, take the message-ciphertext pairs (00001011, 11111101) and (10001001, 11000011). From the messages $M = 00001011$ and $M' = 10001001$ the keys are

$$K_0 = ROT^{-1}(1000 \oplus 0000) \oplus 1011 = ROT^{-1}(1000) \oplus 1011 = 0100 \oplus 1011 = 1111$$

and

$$K_1 = ROT^{-1}(1011 \oplus 1001) \oplus 1000 = ROT^{-1}(0010) \oplus 1000 = 1001.$$

Meanwhile, for the ciphertext $C = 11111101$ and $C' = 11000011$ we have

$$K_0 = ROT^{-1}(1111 \oplus 1100) \oplus 1101 = 1001 \oplus 1101 = 0100$$

and

$$K_1 = ROT^{-1}(0011 \oplus 1101) \oplus 0011 = 0111 \oplus 0011 = 0100.$$

These values for K_0 and K_1 do not match so it is not a slid pair.

If we test the messages 00001011 and 00000010 for keys we get: $K_0 = 1011$ and $K_1 = 1100$ while their ciphertext 11111101 and 00110010 produce the keys: $K_0 = 1011$ and $K_1 = 1100$. This is our slid pair and we have now extracted the keys and broken the cipher.

Note that if we have multiple matches for slid pairs it is easy to check which ones are incorrect. You simply need to take another message and attempt to encrypt it with each set of keys from each possible slid pair. It is highly unlikely that you will get the correct encryption (since you know the corresponding correct ciphertext) from a wrong key. However, if you still have matches you can repeat this till you have narrowed it to one choice.

Real Ciphers

The following ciphers have all been broken by use of the slide attack:

2K-DES (DES with alternating keys)

DES with Brown-seberry Key Schedule

DESX

GOST

MISTY

TREYFER

WAKE-ROFB

Blowfish variants

As well as many more.

Ways to Improve Your Cipher Against the Slide Attack

There are two immediately obvious changes one can make to make a slide attack ineffective against a cipher. If we look at the requirements for the slide attack to work they are that the key schedule is weak and that it is easy to extract the key from the rounds. Thus, the obvious improvements would be to fix these problems.

As was noted in some of the papers referenced, the larger the gap between the keys (i.e. the more rounds included in F) the harder it is to extract the key. Obviously, if you increase the number of keys used and then vary the order in which they are used, the number of rounds in the F function will be increased to an unmanageable proportion. This is not enough to ensure security, as the focus in [2] is how to deal with larger gaps.

The other possibility is to reduce the vulnerability of the rounds to a known-plaintext attack. It is not as immediately obvious how to fix this problem as it is more dependant upon the actual structure of the cipher.

The best strategy would be a combination of the two techniques; strengthening the key schedule and also increasing the difficulty in extracting the key.

References

1. A. Biryukov, D. Wagner, "Slide Attacks" *Preproceedings of FSE6, Fast Software Encryption Workshop 1999* 1999
2. S. Furuya "Slide Attacks with a Known-Plaintext Cryptanalysis" *Information Security and Cryptology – ICISC 2001, 4th International Conference, Seoul, Korea, Dec. 2001, Proceedings*
3. E. Biham "New Types of Cryptanalytic Attacks Using Related Keys"
4. A. Biryukov, D. Wagner, "Advanced Slide Attacks"
5. M. Ciet, G. Piret, J. Quisquater, "Related-Key and Slide Attacks: Analysis, Connections and Improvements"