Session 6

Differential Fault Analysis

This session discusses Fault Attacks on symmetric ciphers, for the example of AES. Unlike the fault attack on RSA, usually several faults are needed to recover the full key. Often, a correct and a faulty output for the same input are needed, making these attacks less severe. At the same time, preventing these attacks can also be more difficult.

Contents

6.1 Differential Fault Analysis of AES

For DFA attacks on AES, usually an error is induced to the AES state towards the end of the algorithm (in one of the last rounds). Key information is recovered from the difference between the correct ciphertext c and the faulty ciphertext (faultytext) c', i.e. from the difference $\Delta = c \oplus c'$. That is why the attacks are called *differential* fault attacks. The same plaintext is used for both encryptions. Following descriptions will be for AES-128 with 10 rounds. The attacks work the same way on other versions of AES, just that the round number changes.

6.1.1 Known Fault in Round 10

We assume an adversary is able to induce a fault at the end of round 10, right before the `Add_Key`.

Toggling bits The adversary can toggle bits right before the `Add_Key` operation. Then the corresponding ciphertext bits change and the adversary learns nothing.

Setting bits The adversary can set bits (either to 1 or to 0, but to a *known* value) right before the `Add_Key` operation. Then the corresponding ciphertext bits give direct information about the key.

The latter fault model assumes a very strong adversary. We continue to get a working attack for the first fault model as well.

6.1.2 Known Fault in Round 9

We move the fault deeper into the cipher. Assume a fault is induced at the end of round 9, right before the `Add_Key`. The affected byte of the state is x, where x is the correct state and x' is the faulty version of x. The difference is

$$\Delta_{in} = x \oplus x'$$

The `Add_Key` of round 9 will add round key k_9 to x, yielding the state (byte) $y = x \oplus k_9$ (and $y' = x' \oplus k_9$) as the input of round 10. Round 10 does not have a `Mix_Columns` so that the error remains local to that byte in the faultytext c'. Note that it is easy to detect such an error, as only one byte (which we refer to as c') is affected by it. We define the output difference between the correct ciphertext byte c ant the faulty byte c' as

$$\Delta_{out} = c \oplus c'$$

and work our way into the cipher now. The `Add_Key` of round 10 will add round key k_10 to both c and c', meaning that the correct and faulty s-box output also differ by $\Delta_{out} = (c \oplus k_{10}) \oplus (c' \oplus k_{10})$.

Figure 6.1: Figure1 for DFA missing here

Now, if Δ_{in} is known, we can find all candidates for k_{10} for which

$$S^{-1}(c \oplus k_{10}) \oplus S^{-1}(c \oplus k_{10}) = \Delta_{in}$$

where c, c' and Δ_{in} are known and we simply have to check 256 candidates for k_{10}. Usually the attack will return 2 candidates for k_{10}. This can be seen when considering the equation $S(y) \oplus S(y \oplus \Delta_{in}) = \Delta_{out}$, as for one tuple oa input and output difference there are typically two (or 0) possible input states y (where $k_{10} = c \oplus S(y)$, obviously).

This attack works if:

Δ_{in} **is known:** The adversary can flip chosen bit(s). This is still a strong adversarial assumption. However, it is common for *Feistel* ciphers that output part of the internal state of the second-last round (e.g. DES).

y' **is known** The adversary can set bits (either to 1 or to 0, but to a *known* value). for this case we already had a working attack before. Very strong adversary.

This attack is relevant for *Feistel* ciphers and as a building block for more generic attacks on AES.

6.2 A realistic DFA of AES

By moving the fault deeper into the cipher we get a stronger DFA [PQ03]. Assume a single faulty byte in round 9 *before* `Mix_Columns` (any byte, any value, hence a simple fault model, more realistic attack).

Figure 6.2: Figure2 for DFA on AES missing here

the four input bytes to the `Mix_Columns` are labeled w_0, w_1, w_2, w_3. Let's assume the fault occurs for $w'_0 = w_0 + \Delta$. It follows:

$$\texttt{Mix\_Columns}(w_0 \oplus \Delta, w_1, w_2, w_3) = \texttt{Mix\_Columns}(w_0, w_1, w_2, w_3) \oplus \texttt{Mix\_Columns}(\Delta, 0, 0, 0)$$

where the first term is the fault free case and the second term is the influence of the fault, which is linear, since `Mix_Columns` is a linear operation in $\mathbb{F}_{2^8}$.

Now, since $\Delta \in \{0,1\}^8$, there are 256 possibilities for $\texttt{Mix\_Columns}(\Delta, 0, 0, 0)$, with

$$\texttt{Mix\_Columns}(\Delta, 0, 0, 0) = [\Delta_0, \Delta_1, \Delta_2, \Delta_3] = \Delta_{in}$$

The the correct/faulty state *after* `Mix_Columns` of round 9 is:

$$X = [x_0, x_1, x_2, x_3] \qquad X' = [x_0 \oplus \Delta_0, x_1 \oplus \Delta_1, x_2 \oplus \Delta_2, x_3 \oplus \Delta_3]$$

Furthermore $\Delta_{out} = [c_0 \oplus c'_0, c_1 \oplus c'_1, c_2 \oplus c'_2, c_3 \oplus c'_3]$ is known.

Now we test for each possible Δ until we get a correct Δ_{out} The attack works as follows:

- for all $\Delta \in \{0,1\}^8$ (if byte of error is unknown, test for each byte (complexity $\times 4$))

1. Compute $\Delta_{in} = \texttt{Mix\_Columns}(\Delta, 0, 0, 0)$
2. for all $k \in \{0,1\}^{32}$ (four bytes of last round key)

$$(if\ )\Delta_{in} = S^{-1}(c \oplus k) \oplus S^{-1}(c' \oplus k)$$

   add k to candidate list

Attack complexity:

- Straightforward implementation $\approx 2^{40}$
- Better: in step 2., check equation for the first byte, only if solvable, check for the second byte etc.

Attack properties

- usually 2 ciphertext-faultytext pairs are sufficient to recover 4 key bytes of last round key

- reasonably fast
- can be further optimized by moving single-bit fault before `Mix_Columns` of round 8:
  - will induce 4 faults in same column after `Mix_Columns` of round 8
  - will result in 4 faults in different columns after `Shift_Rows` of round 9
  - will result in 16 faults in the whole state after `Mix_Columns` of round 9, *all of them related!*
  - only 2 ciphertext-faultytext pairs are sufficient to recover full key
  - fault model quite generic (cf. [PQ03])

Faulting the Control Flow Note that faulting the control flow is equally devastating as the previously described attack. Assume a correct ciphertext and a faulty computation on the same input, where the last round is skipped (or executed twice). In either case, the input state to the last round is revealed, allowing to recover the last round key with a single ciphertext-faultytext pair.

Bibliography

[PQ03] Gilles Piret and Jean-Jacques Quisquater, *A differential fault attack technique against spn structures, with application to the aes and khazad*, Cryptographic Hardware and Embedded Systems — CHES 2003 (Colin D. Walter, Çetin K. Koç, and Christof Paar, eds.), Lecture Notes in Computer Science, vol. 2779, Springer Berlin Heidelberg, 2003, pp. 77–88 (English).