The next Stage of Differential Fault Analysis: How to break completely unknown cryptosystems

Eli Biham
Adi Shamir
Computer Science Dept.

Applied Math Dept.
Technion
The Weizmann Institute
Israel
Israel

October 30-th, 1996
(DRAFT)

The idea of using computational faults to break cryptosystems was first applied by Boneh Demillo and Lipton to public key cryptosystems, and then extended by Biham and Shamir to most types of secret key cryptosystems. In this new research announcement, we introduce a modified fault model which makes it possible to find the secret key stored in a tamperproof cryptographic device even when nothing is known about the structure and operation of the cryptosystem. A prime example of such a scenario is the Skipjack cryptosystem, which was developed by the NSA, has unknown design, and is embedded as a tamperproof chip inside the commercially available Fortezza PC cards.We have not tested this attack on Skipjack, but we believe that it is a realistic threat against some smart card applications which were not specifically designed to counter it.

The main assumption behind the new fault model is that the cryptographic key is stored in an asymmetric type of memory, in which induced faults are much more likely to change a 1 bit into a 0 than to change a 0 bit into a 1 (or the other way around). CMOS registers seem to be quite symmetric, but most types of nonvolatile memory exhibit some degree of asymmetry. For example, a 1 bit in an EEPROM cell is stored as a small charge on an electrically isolated gate. If the fault is induced by external radiation (e.g., ultraviolet light), then the charges are more likely to leak out of the gate than to be forced into the gate.

To make the analysis simpler, we assume that we can apply a low level physical stress to the tamperproof device when it is disconnected from power, whose only possible effect is to occasionally flip one of the 1 bits in the key register to a 0. The plausibility of this assumption depends on numerous physical and technical considerations, which are beyond the scope of this note.

We further assume that we are allowed to apply two types of cryptographic functions to the given tamperproof device: We can supply a cleartext m and use the current key k stored in the nonvolatile memory of the device to get a ciphertext c, or we can supply a new n-bit key k' which replaces k in the nonvolatile memory.

The cryptanalytic attack has two stages:

1. In the first stage of the attack, we keep the original unknown secret key k stored in the tamperproof device, and use it to repeatedly encrypt a fixed cleartext m_0. After each encryption, we disconnect the device from power and apply a gentle physical stress. The resultant stream of ciphertexts is likely to consist of several copies of c_0, followed by several copies of a different c_1, followed by several copies of yet another c_2, until the sequence stabilizes on c_f. Since each change is likely to be the result of one more key bit flipping from 1 to 0 (thus changing the current key k_i into a new variant k_i+1), and since there are about n/2 1 bits in the original unknown key k, we expect f to be about n/2,and c_f to be the result of encrypting m_0 under the all-zero key k_f.

2. In the second stage of the attack, we work our way backwards from the known all-zero key k_f to the unknown original key k_0. Assuming that we already know some intermediate key k_i+1, we assume that k_i differs from k_i+1 in a single bit position. If we knew the cryptographic algorithm involved, we could easily try all the possible single bit changes in a simple software simulation on a personal computer, and find the (almost certainly unique) change which would give rise to the observed ciphertext c_i. However, we dont need either a simulator or knowledge of the cryptographic algorithm, since we are given the real thing in the form of a

tamperproof device into which we can load any key we wish, to test out whether it produces the desired ciphertext c_i. We can thus proceed deterministically from the known k_f to the desired k_0 in O(n) stages, trying O(n) keys at each stage. The attack is guaranteed to succeed if the fault model is satisfied, and its total complexity is at most O(n^2) encryptions.

This seems to be the first cryptanalytic attack which makes it possible to find the secret key of a completely unknown cryptosystem in polynomial time (quadratic time in our case). It relies on a particular fault model which seems to be realistic, but requires further study. In the full version of this paper we'll discuss numerous extensions of the attack, including the analysis of more complicated fault models in which the sequence of corrupted keys forms a biased random walk in the space of 2^n possible keys.