Faults, Injection Methods, and Fault Attacks

Chong Hee Kim and Jean-Jacques Quisquater,
Université Catholique de Louvain

An active attacker can induce errors during the computation of the cryptographic algorithm and exploit the faulty results to extract information about the secret key in embedded systems. We call this kind of attack a *fault attack*. Fault attacks can break an unprotected system more quickly than any other kind of side-channel attack such as simple power analysis (SPA), differential power analysis (DPA), or electromagnetic analysis (EMA). For example, the attacker can break RSA-CRT (RSA with Chinese Remainder Theorem) with one faulty result, and Data Encryption Standard (DES) and Advanced Encryption Standard (AES) with two. Furthermore, the protection of fault attacks is more costly in terms of chip area. Here, we survey fault injection methods, types of faults, and fault attack models.

Fault injection methods

There are many ways to induce faults. We describe the most common fault injection techniques here.

Glitch attack

Variations in the supply voltage or in the external clock can impede a device's functionality. It can lead to the misinterpretation or omission of instructions. It can also cause data misreads. To succeed, the attacker must control the glitch's amplitude and duration. This attack is the most common method for breaking several cryptosystems. It's easy to apply because the attacker need not worry about localization. Conversely, the main drawback of this type of attack is that the attacker cannot focus on specific parts of the device. Nowadays, most smart cards have glitch detectors and DC filters to resist such attacks.

Temperature attack

At extreme temperatures, devices do not work properly. Random modification of RAM cells or read and write threshold mismatches in nonvolatile memories (NVMs) can occur. Most smart cards have temperature detectors, but a mismatch can occur between the memory cells' operating temperature and the detector's detecting range.

Light attack

Today, this is the most powerful attack. Unlike a glitch attack, in this case the attacker can choose the location of the attack in the device. Because all electric circuits are sensitive to light due to photoelectric effects, the attacker can use the current caused by photons to induce faults. To succeed, the attacker must control the light's energy, wavelength, location, and emission time. A simple camera flash can execute a light attack. The advantage of this simple method is that it is very cheap. But the penetration depth of a light attack depends on the light's wavelength, and camera flash provides only visible wavelengths (white light). Furthermore, a camera flash is difficult to control accurately. Therefore, a laser system may be more effective for a light attack. It allows using several discrete wavelengths and targeting a very small area of the device, making it difficult to protect against even though most smart cards have light detectors and metal shields. Furthermore, today it's possible to perform a laser attack to the back side of the chip, where usually no protecting mechanism is applied.

Magnetic attack

Another way to induce faults is by using an emission of a powerful magnetic pulse near the silicon. The magnetic field creates local currents on the component's surface to generate a fault. This attack can be performed with cheap materials—for example, a needle wound with wire—and allows attacking small parts of the chip. However, other than the low cost, it's more practical to use a laser system.

Types of faults

Two kinds of faults can be generated as an effect of the attack: permanent and transient.

Permanent faults

In a permanent fault, the value of a cell is definitely changed. Either data (EEPROM or RAM) or code (EEPROM) can be damaged. A permanent fault can be very powerful when a secret key is changed. However, it's very difficult to induce permanent faults on specific logic cells by using a memory-ciphering mechanism and scrambling the physical address of memory in modern smart cards.

Transient faults

Transient faults are provisional faults. The circuit recovers its original behavior after reset or when the fault's stimulus ceases. A transient fault can disturb code execution or a particular computation. Therefore,

0740-7475/07/$25.00 © 2007 IEEE

Copublished by the IEEE CS and the IEEE CASS

a call to a subroutine might be skipped, a test might be avoided, different executions might be executed, a wrong value could be fetched, or a program counter could be modified.

Models of fault attacks

For a successful fault attack, the attacker must first know which errors to introduce, then try to put those errors into practice. According to the error models, it's possible to distinguish whether the proposed attack is practical or not.

Bit versus byte errors

The attacker can assume it's possible to change a value of one bit or one byte. Usually, a byte error model is more practical, because a byte is the basic level for storing data in memory or transferring data on a bus. Therefore, it's easier to change the value of a byte than that of a bit. Using a bit error model, almost all cryptosystems can be broken. However, inducing a bit error is very difficult with current silicon technology.

Specific versus random value errors

The attacker can assume it's possible to change the value of data into a specific or random value. Typically, all 0s or 1s are used for the specific values. In general, a random value error is easier to induce.

Static versus computational errors

The attacker can assume it is possible to induce an error on memory itself or during the computation of some operations. For example, the attack on RSA-CRT needs a faulty computation on one of the exponentiations. However, a differential fault attack (DFA) on a digital-signature algorithm (DSA) would require flipping a bit of the secret key stored in memory. Usually, a computational error is easy to put into practice, whereas changing a value stored in memory is difficult.

Data versus control errors

A control error occurs when some iterations or operations are skipped because of faults. Normally, data errors (such as those on the secret key or intermediate values) are used to attack the cryptosystem. Although the induction of a control error is more difficult, it can be very powerful.

Chong Hee Kim *is a postdoctoral researcher in the Crypto Group at Université Catholique de Louvain, Belgium. Contact him at chong-hee.kim@uclouvain.be.*

Jean-Jacques Quisquater *is a professor of cryptography and multimedia security in the Department of Electrical Engineering at Université Catholique de Louvain. Contact him at quisquater@dice.ucl.ac.be.*