From Reliability to Safety

P.-Y. Liardet & Y. Teglia
STMicroelectronics
Crypto and Security Team
Hardware Development & Systems Engineering Dept.
Smartcard ICs Division
Z.I. Rousset, 13 106 Rousset Cedex,
France
{Pierre-Yvan.Liardet,Yannick.Teglia}@st.com

Abstract

Twenty years ago, the semiconductor industry was investing millions dollars to build tools to investigate technology defects, to modelize them and guarantee reliability. The technologic knowledge resulting from this investment has made possible the arrival of smartcards as low-cost portable security devices enabling several today applications. As a side effect, the same knowledge also provided the basis for new attack means allowing one to break security devices. Reliability has for a long time been considered as the final goal in electronic design. Despite internal and external perturbations, devices were conceived to eventually output a result, be it correctly computed or partially incorrect. What we expect today from perturbed devices is also to not compromise their secrets. We will see on some examples how, through the last decade, attacks were devised that take advantage of potential unreliable hardware behaviors to recover internally managed secret key material. Studying these examples, we will draw some perspectives for the security industry to win the faults battle.

1. Introduction

Since the beginning of the era of semiconductor industry, there has been a race to shrink the technologies, in order to increase the density of components available on a given area and then to put more and more chips on wafers. This has generated very important investments in building tools to investigate technology defects and to modelize them. As a matter of fact, as technology is shrinking, the silicon manufacturers engineering teams need sharper tools to physically investigate the defects while the design teams need to rely on adequate models for building their new architectures. These two branches serve the same objective, having reliable chips, in term of coherent construction as well as in term of complete technology testing.

2. The reliability race

In the meantime the aeronautic industry was integrating more and more electronic devices, embedded systems, etc ... The huge cost of an aeronautic operation leaded to put a reliability layer at system level which was added to the safe electronic components to resist an environmental stress like cosmic rays in the case of spatial systems.

The protection added ranged from redundancy to error detecting or correcting capabilities of the devices. The kind of redundancy used relied for example on building several instances of the same system, by independent teams and on different technologies and then compare their results to establish a vote in order to take the decision. Previous work, due to L. Lamport et al. known under the name of “the Byzantine Generals Problem” [3] reports an algorithm to solve the problem of taking decisions in an untrustworthy environment. But such a replicated approach has the considerable drawback to be very expensive for customer-oriented products like multimedia applications that moreover do not need to go this far in security concerns. Of course, redundancy can be addressed at a lower level by duplicating selected parts of the device, according to the knowledge of the error model. For instance only CPU registers can be replicated, since a single bit change in the condition code register can turn the expected behavior in its complete opposite.

Unfortunately, the cost of redundancy adding, at the bosom of cost-effective and customer oriented small devices, even partial and despite its efficiency to thwart errors, is often not compatible with mass-market products price requirements.

In the early age of computers, in order to improve storage reliability, error detecting and correcting capabilities were developed in a more mathematical and theoretical approach. Classical Error Correcting Code (ECC) such as Hamming and Reed Solomon Codes, are still widely used. The latter can be found in audio Compact-Disc systems to improve error tolerance and reduce alteration effects dues to media manipulations.

For checking the integrity of a memory area or in transmission of a burst of data, mechanisms like Cyclic Redundancy Code (CRC) and checksum have been standardized.

Furthermore CRC and ECC can easily be implemented at a reasonably low cost and guarantee most of the time an accurate level of integrity for a known fault model. Thus, this kind of protection was implemented to assert a degree of confidence, in Personnal Computer memories, but also in secure tokens like smartcards.

3. Where reliability meets safety

But secure tokens have other threats to face than cosmic rays. The idea came to malicious insiders to turn the resulting tools of the race to reliability into attack means allowing one to break security devices; these tools, first used to debug products, range from passive ones like memory scanner (SEM), to more invasive ones like the Focused Ion Beam (FIB) that can be seen as a high-end "cutter" to remove or partially deactivate physical parts of the device. Fortunately, handling these tools limits the number of possible malicious people by the required skill and financial resources. The latter is actually the key point because as technology has shrunk these tools have become more and more expensive. By the way, on current technologies these threats are not the biggest hazard to face. In the meantime the attacker have investigated a way to perform lower cost attacks.

Spurred in 96 by the first publication of fault attacks on cryptosystems by Boneh, Demillo and Lipton [2], the fault makers have tried very simple setup like those described in [4] on old fashion memories and were challenged to induce defaults on today's technology.

But breaching into the authentication process of a secure device like a smartcard was already experimented by hackers, using the well-known card tearing technique that can be viewed as a kind of fault attack. Up to these publications, secret keys of cryptographic systems were not identified as targets of fault attacks. Even if some times requiring very sophisticated and expensive material, faults have become a major field of investigation these last years.

Today the concept of fault attacks has been investigated to find ad-hoc fault models regarding the current technology, but some key information on models are always under investigation and will always be because of the continuous evolution of technologies.

4. Description of Fault Injection

The so-called fault injection, whatever it works with (rays, glitch, temperature) aims at disturbing the normal flow of execution of the running program, or at modifying the content of memories or registers. One generally distinguishes transient faults from permanent faults. The former implies that the disturbed device can recover its "normal" state, immediately after the fault has occurred, or after a reset for instance. The latter first described in [1] implies that the disturbed device will remain in its "faulty" state even after a power-off. Since classical programs and the value they handle can be modified by malicious external events, this attack was eventually transposed in the field of cryptography. Indeed, secure token intensively use cryptographic algorithms both for internal use and communication with the outside untrusted world. The first practical example of fault injection in the last round of the DES was demonstrated in the 2nd IBM Security Workshop that took place in April 2000 in Amsterdam. The principle is to combine by a simple bit-wise exclusive or, the equations of the last round were a fault occurs in the right part of data and the same equation without fault. Therefore we get an equation that holds almost only for the good sub-key. With few error samples the last round sub-key is then easily recovered. Against all expectation the famous randomization countermeasure applied to prevent from DPA does not always help in protecting from such previously described attack since the equation holds even with randomized states.

It is well known that any cryptosystem is likely to fall under the threat of a fault attack. Nowadays most of cryptosystems are built iterating a core function (generally called round for secret key ones) that involves few secret bits that can easily be guessed if the round number is reduced to one or a few by a fault in operation on the encryption. More precisely, for secret key cryptosystems the round itself is most of the time a weak sub-cipher, so that the result of a few rounds can directly be broken by standard cryptanalysis. For public key cryptosystems, like elliptic curve based schemes, the work is even easier since a few computations can reveal involved key bits. While reliability is still of concern in those devices, attacks to steal internal data is the recurrent issue the silicon manufacturers and other actors in this market have to handle. Safety is one of the main issues that nowadays-secure tokens have to deal with.

5. How to make faults exploitation difficult

The first element that plays a major role in the difficulty of faults exploitation is the technology evolution it-

self. The shrinking avalanche that makes technology tuning more and more difficult has the benefic effect to make unpractical most of the localized errors on dedicated parts of silicon. Automatic routing techniques under relaxed constraints, also participate to the complexity of focalized error.

Sensors mechanisms that act as regulators for external events like temperature or voltage, also make more difficult the day-to-day job of the hackers; nevertheless portable security devices have to be smart enough in order to distinguish a real attack from an erratic but honest behavior due to difficult conditions of use.

Still at the technology level, the fault maker often needs synchronization points in order to improve the probability of an effective and useful fault. Resorting to invasive techniques in order to get these triggers has become also hazardous with today technologies. A natural way for an attacker to look for synchronization points are side channels, and if no care is taken on the component leakage this can be successful. Obviously, countermeasure addressing the side-channel leakage are implemented in security devices and avoid timing references for the fault maker pulling to neglect the success probability for a subtly induced fault. With the same effects, randomization of execution flow (by induction of random delays) is an easy, low-cost and efficient to implement (both in hardware or software) countermeasures against fault attacks. Working on the effects of different kinds of faults, it was a very early identified matter of fact that some instructions, like conditional jumps were at risk. Obviously people have developed work-around sometimes going to simply never use such instructions.

Randomization techniques used to blind internal computations may also help, for instance in the attack that consists in reducing the number of rounds in a DES by modifying the round counter, since an attacker will recover blinded results and may have problems to deal with. But one has to be careful in the way the randomization is done, since the attack by fault injection in last round still works, as seen previously.

Obviously error control and corrections, from the simple double computation (serially or in parallel) to more sophisticated techniques based on coding theory, thus participates to the reliability and can be turned into efficient and well known countermeasures.

All of this must be completed by a fault strategy. At first the implementation must ensure that every potential threat is taken into account, and thwarted according to the appropriate risk model, and further, that detected or guessed faults are correctly treated, providing none or very few information to the outside. For example, one should not neglect that if a secure device detects a malicious fault induction and jumps into a blocking state, it then provides to the external world the information that the fault appeared. In some particular circumstances this can lead to secret material retrieval.

6. Conclusion

Faults attacks are far to be a new threat for secure-devices. Its impact is still bad understood by the layman, sometimes exaggerated, most of the time neglected. It must embody a major concern for the industry. Even if today, many techniques exist in order to counter the less costly ones they make hardware and software more and more complex increasing the price of the technology. What we can also learn from the history of attacks in general, is that side-channel, fault and classical cryptanalysis must not be only considered separately.

References

[1] E. Biham and A. Shamir. On the importance of checking cryptographic protocols for faults. *Advances in Cryptology – CRYPTO'97, LNCS Springer-Verlag*, 1294:513–525, August 1997.

[2] D. Boneh, H. Shacham, and B. Lynn. On the importance of checking cryptographic protocols for faults. In *Advances in Cryptology – Eurocrypt'97*, volume 1233 of *Lecture Notes in Computer Science*, pages 37–51. Springer-Verlag, 1997.

[3] R. S. L. Lamport and M. Pease. The byzantine generals problem. *In ACM Transactions on Programming Languages and Systems (TOPLAS) archive*, 4(3):382–401, July 1982.

[4] S. Skorobogatov and R. Anderson. Optical fault induction. In *In B.Kaliski, Ç. K. Koç and C. Paar – CHES'02: Cryptographic Hardware and Embedded Systems*, volume 2523 of *Lecture Notes in Computer Science*, pages 2–12. Springer-Verlag, June 2002.