A New Class of Low Cost Attacks on Smart Cards

Balakrishna Anumolu
School of Electrical Engineering & Computer Science,
Oregon State University, Corvallis, Oregon 97331 -USA.
E-mail: anumolu@cs.orst.edu

Abstract—
This paper describes a new type of attack on tamper-resistant smart cards. The attack uses a inexpensive microscope and a flash gun to extract secret information from smart cards. The intensity of the flash can either permanently remove charge from a memory cell, or cause a temporary, incorrect flip of a transistor switch. Memory cells often contain data that secures other data, so reprogramming them can open access to the secrets stored inside the card. The low cost of the equipment makes this kind of attack especially dangerous and easy to implement. Existing tampering techniques, including Microprobing, Software attacks, Eavesdropping and Fault genaration are discussed to supply the reader with the necessary background.

I. Introduction

Smartcards promise numerous security benefits. They can participate in cryptographic protocols, and unlike magnetic stripe cards, the stored data can be protected against unauthorized access. However, the strength of this protection seems to be frequently overestimated.

Section 2, shall review the most important hardware techniques for breaking into smartcards[1]. Which should give a realistic impression and an idea of how physical tampering works and what it costs. Section 3, shall describe in detail a new kind of physical attack and some countermeasures that would circumvent these type of attacks.

II. Tampering Techniques

We can categorize the attacks into two classes invasive and non-invasive.

Invasive Attacks: All microprobing attacks are invasive attacks. Microprobing techniques are usually used to access the chip surface directly, thus facilitating the observation and manipulation of the intergrated circuit of the smart card. These kind of attacks require hours or weeks in a specialized laboratory and in the process destroy the packaging of the card.

Non-Invasive Attacks: Here the attacked card is not physically harmed and the equipment used in the attack can usually be disguised as a normal smartcard reader. Examples of these kinds of attacks are software attacks, Eavesdropping attacks, Fualt generation attacks.

- **Software attacks** use the normal communication interface of the processor and exploit security vulnerabilities found in the protocols, cryptographic algorithms, or their implementation.

- **Fualt generation attacks** use abnormal environmental conditions to generate malfunctions in the processor that provide additional access.

- **Eavesdropping Attacks** monitor, with high time resolution, the analog characteristics of all supply and interface connections and any other electromagnetic radiation produced by the processor during normal operation. The statistics thus obtained are used to make some calculated guesses about the bits in the secret key stored in the smart card.

Non-invasive attacks are particularly dangerous in some applications for two reasons. Firstly, the owner of the compromised card might not notice that the secret keys have been stolen, therefore it is unlikely that the validity of the compromised keys will be revoked before they are abused. Secondly, non-invasive attacks often scale well, as the necessary equipment (e.g., a small DSP board with special software) can usually be reproduced and updated at low cost.

The design of most non-invasive attacks requires detailed knowledge of both the processor and software. On the other hand, invasive microprobing attacks require very little initial knowledge and usually work with a similar set of techniques on a wide range of products. Attacks therefore often start with invasive reverse engineering, the results of which then help to develop cheaper and faster non-invasive attacks.

A. Invasive Attacks

A.1 Depackaging:

Invasive attacks start with the removal of the chip package [2]. The card plastic is heated until it becomes flexible. This softens the glue and the chip module can then be removed easily by bending the card. The chip module is then covered with 20-50 ml of fuming nitric acid heated to around 60 C and wait for the black epoxy resin that encapsulates the silicon die to completely dissolve. The chip is then washed with acetone in an ultrasonic bath, followed optionally by a short bath in deionized water and isopropanol. The remaining bonding wires are removed with tweezers and the die is glued into a test package, and its pads are bonded manually to the pins.

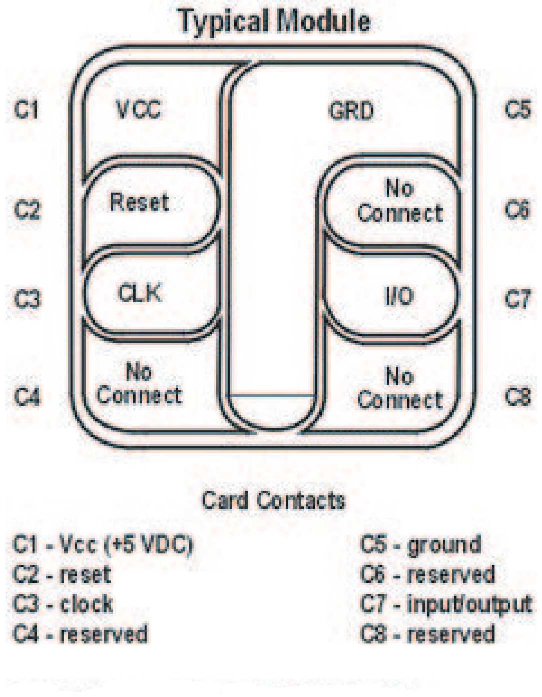

Figure 1: The ISO 7816-2 Standard smart chip.

A.2 Layout Reconstruction:

The next step in an invasive attack on a new processor is to reconstruct the layout of it. An optical microscope with a CCD camera can be used to produce several meter large mosaics of high-resolution photographs of the chip surface. Basic architectural structures, such as data and address bus lines, can be identified quite quickly by studying connectivity patterns and by tracing metal lines that cross clearly visible module boundaries (ROM, RAM, EEPROM, ALU, instruction decoder, etc.).

The chip surface is not transparent and therefore obscures the view of many structures below. Lower layers can still be recognized through the height variations that they cause in the covering layers. Deeper layers can only be recognized in a second series of photographs after the metal layers have been stripped (using hydrofluoric acid). Confocal microscopes are usually used in circuit reconstruction, these microscopes assign different colors to different focal planes, thus preserving depth information. The layout has to be reconstructed only until the necessary bus lines and modules needed to access all memory values have been identified. Recently, chip designers started to add prorietary cryptographic hardware, non-standard instruction sets and bus scrambling techniques.These techniques make the attackers task much harder, but are still suceptible to threats.

With semiautomatic image-processing methods, significant protions of the processor can be reverse engineered, then using the resulting polygon data tansistor and gate-level netlists can be automatically generated for circuit simulations. Information in the ROM is stored in the diffusion layer, which can retrieved easily be removing all the covering layers using HF acid. The stored bit pattern is easy to recognize from the rims of the diffusion regions. While the ROm usually does not contain any cryptographic key material, it does often contain enough I/O, access control, and cryptograhic routines to be of use in the design of a non-invasive attack.

A.3 Manual Microprobing:

The most important tool for invasive attacks is a microprobing workstation [3]. Its major component is a special optical microscope (e.g., Mitutoyo FS-60) with a working distance of at least 8mm between the chip surface and the objective lens. On a stable platform around a socket for the test package, several micropositioners are installed, which allow us to move a probe arm with submicrometer precision over a chip surface. On this arm, a "cat whisker" probe is installed (e.g., Picoprobe T-4-10). This is a metal shaft that holds a 10 micrometer diameter and 5mm long tungsten-hair, which has been sharpened at the end into a ¡ 0:1 micrometer tip. These elastic probe hairs are used to establish electrical contact with on chip bus lines without damaging them. These are then connected via an amplifier to a digital signal processor card that records or overrides processor signals and also provides the power, clock, reset, and I/O signals needed to operate the processor via the pins of the test package.

A.4 Memory Read-out Techniques:

It is usually not practical to read the information stored on a security processor directly out of each single memory cell, except for ROM. The stored data has to be accessed via the memory bus where all data is available at a single location. Microprobing is used to observe the entire bus and record the values in memory as they are accessed.In order to read out all memory cells without the help of the card software, a CPU component has to be abused as an address counter to access all memory cells for us. The program counter is already incremented automatically during every instruction cycle and used to read the next address, which makes it perfectly suited to serve as an address sequence generator. We only have to prevent the processor from executing jump, call, or return instructions, which would disturb the program counter in its normal read sequence. Tiny modifications of the instruction decoder or program counter circuit, which can easily be performed by opening the right metal interconnect with a laser, often have the desired effect.

B. Non-Invasive Attacks

Attacks usually start with invasive reverse engineering, whose results aid in the formulation of cheaper and faster non-invasive attacks. Any processor is essentially a set of a few hundred flipflops (registers, latches, and SRAM cells) that define its current state, plus combinatorial logic that calculates from the current state the next state during every clock cycle. Many analog effects in such a system can be used in non-invasive attacks. Smartcard processors are particularly vulnerable to non-invasive attacks [4], because the attacker has full control over the power and clock supply lines. Larger security modules can be equipped with backup batteries, electromagnetic shielding, low-pass filters, and autonomous clock signal generators to reduce many of the risks to which smartcard processors are particularly exposed.

B.1 Glitch Attacks:

In a glitch attack, a malfuntion is deliberately generated, which causes one or more flipflops to adopt the wrong state. The aim is usually to replace a single critical machine instruction with an almost arbitrary other one. Glitches can also aim to corrupt data values as they are transferred between registers and memory. There are currently three popular techniques for creating fairly reliable malfunctions that affect only a very small number of machine cycles in smartcard processors: clock signal transients, power supply transients, and external electrical field transients.

Particularly interesting instructions that an attacker might want to replace with glitches are conditional jumps or the test instructions preceding them. They create a window of vulnerability in the processing stages of many security applications that often allows us to bypass sophisticated cryptographic barriers by simply preventing the execution of the code that detects that an authentication attempt was unsuccessful. Instruction glitches can also be used to extend the runtime of loops, for instance in serial port output routines to see more of the memory after the output buffer, or also to reduce the run time of loops, for instance to transform an iterated cipher function into an easy to break single-round variant.

Clock-signal glitches are currently the simplest and most practical ones. They temporarily increase the clock frequency for one or more half cycles, such that some flipflops sample their input before the new state has reached them.In some designs, a clock-frequency sensor that is perfectly secure under normal operating voltage ignores clock glitches if they coincide with a carefully designed power fluctuation. There are published clock and power waveform combinations for some widely used processors that reliably increment the program counter by one without altering any other processor state. An arbitrary subsequence of the instructions found in the card can be executed by the attacker this way, which leaves very little opportunity for the program designer to implement effective countermeasures in software alone. Power fluctuations can shift the threshold voltages of gate inputs and anti-tampering sensors relative to the unchanged potential of connected capacitances, especially if this occurs close to the sampling time of the flipflops. Smartcard chips do not provide much space for large buffer capacitors, and voltage threshold sensors often do not react to very fast transients.

B.2 Eavesdropping Attacks:

These attacks take advantage of the analog characteristics of all supply and interface connections and any other electromagnetic radiation produced by the smartcard processor during normal operation. The popular kinds of attacks in this class are the Timing analysis attacks and Power analysis attacks.

Timing analysis exploits the execution time of operations on a smart card. If an attacker has access to the card and can make a series of measurements of the time required for partial operations, this data can be used to determine the key. Paul Kocher discovered timing analysis to the public in December 1995 [5]. An effective and efficient method of counteracting timing analysis is to use nonlinear key updating.

Simple Power Analysis, Differential Power Analysis and High Order Differential Power Analysis attacks exploit the power consumption characteristics of the smart card and can be used to expose the secret key and the protocols and algorithms used on it. A more detailed description of these techniques can be found in the publications of the inventor of the technique, Paul Kocher [6].

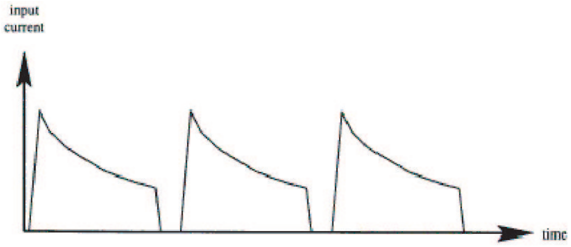

Figure 2: The current supplied to smart cards with detached power supplies.

Since power analysis attacks were discovered, not many practical solutions for preventing them have been proposed. And of the few methods that have been proposed for preventing power analysis attacks they have yet to be implemented in the majority of smart cards. The most promising method, was proposed by Shamir [7].

III. Optical Fault Induction Attack

Unfortunately for the attacker, many chipmakers have now implemented defenses against the most obvious non-invasive attacks. These defenses include randomized clocking to make power analysis harder, and circuits that react to glitches by resetting the processor. Meanwhile invasive attacks are becoming constantly more demanding and expensive, as feature sizes shrink and device complexity increases, Below is described a new class of attacks called semi-invasive attacks. This means that, like invasive attacks, they require depackaging the chip to get access to the chip surface. But the passivation layer of the chip remains intact - semi-invasive methods do not require making an electrical contact to the metal surface or doing mechanical damage to the silicon.

Semiconductor transistors nowadays are more sensitive to ionizing radiation whether caused by nuclear explosions, radioactive isotopes, X-rays or cosmic rays - than the thermionic valves used previously [8]. In the middle sixties, during experiments with pulsed lasers, it was found that coherent light causes some similar phenomena. Lasers started to be used to simulate the effects of ionizing radiation on semiconductors.

Laser radiation can ionize an ICs semiconductor regions if its photon energy exceeds the semiconductor band gap. The laser radiation with 1.06m wavelength and 1.17eV photon energy has a penetration depth of about 700m and provides good spatial ionization uniformity for silicon devices. So an it was found that when a intense light source is applied to a semiconductor chip, it was possible to change the state of a memory cell easily.

Manufacturer	Wavelength nm	Depth mm
Gemplus	3.8	2.1
ORGA	4.1	1.5
Certicom	1.6	2.3
IBM	3.6	2.2
Ankari	2.4	1.6
ActivCard	3.1	2.9

Table 1: Variations of laser wavelength and depth with respect to smart card manufacturer.

Standard smart card circuitry is extremely vulnerable to attack using optical probing. By exposing a transistor to a laser beam, or even the focused light from a flashlamp, it can be made to conduct. This gives rise to many effects that can be used by an attacker. For example,it can be used to load a short program that outputs sensitive data or to induce a fault in the integrated circuit, in any targeted transistor, and at precisely the clock cycle of choice.

The work reported above shows that optical probing attacks and fault indcution attacks are possible using low-cost equipment.In particular, this technique is effective at implementing the attack of Boneh et al on RSA signatures against at least one smartcard currently on the market. The Digital Millennium Copyright Act makes the the reporting of any further details imprudent until countermeasures have been implemented (more details and specifics about the attack shall be published at the coming FHES Confrence).

Further scientific work includes a fuller investigation of the potential for attacks by an opponent with a moderately resourced laboratory, which means a modern probing station with both a multiple wavelength laser and a motorized stage under program control. We hope to have such apparatus operational by the time of the conference, and intend to use it for testing a number of new attack ideas on different types of chip.

References

[1] Oliver Kommerling and Markus G. Kuhn, “Design principles for tamper-resistant smartcard processors,” in *Security Protocols - 5th International Workshop*, Ç. M Lomas, Ed., 1997, pp. 125–136.

[2] Steve H. Weingart, “Physical security devices for computer subsystems: A survey of attacks and defences,” in *Cryptographic Hardware and Embedded Systems - CHES 2000*, Ç. K. Koç and C. Paar, Eds., 2000, Lecture Notes in Computer Science No. 1965, pp. 302–317.

[3] P. Pallier H. Handschuh and J. Stern, “Probing attacks on tamper resistant devices,” in *Cryptographic Hardware and Embedded Systems - CHES 1999*, Ç. K. Koç and C. Paar, Eds., 1999, Lecture Notes in Computer Science No. 1717, pp. 303–316.

[4] E. von Faber, “Security evaluation schemas for the public and private market with a focus on smart card systems,” in *Cryptographic Hardware and Embedded Systems - CHES 1999*, Ç. K. Koç and C. Paar, Eds., 1999, Lecture Notes in Computer Science, No. 1717, pp. 187–204.

[5] Paul Kocher, “Timing attacks on implementation of diffie-hellman, rsa, dss and other systems,” 1995.

[6] B. Jun Paul Kocher, J. Jaffe, “Differential power analysis,” 1998.

[7] Adi Shamir, “Protecting smart cards from passive power analysis with detached power supplies,” in *Cryptographic Hardware and Embedded Systems*, Ç. K. Koç and C. Paar, Eds., 2000, Lecture Notes in Computer Science, No. 1965, pp. 71–77.

[8] Sergei Skorobogatov and Ross Anderson, “Optical fault induction attacks,” 2002.