Advances in Computer Science and Information Technology (ACSIT)
p-ISSN: 2393-9907; e-ISSN: 2393-9915; Volume 3, Issue 5; July-September, 2016, pp. 377-385

http://www.krishisanskriti.org/Publication.html

Review of Smartcard Attacks and Countermeasures

Sushma Bahuguna

BCIIT, Affiliated to GGSIPU
E-mail: sushmabahuguna@gmail.com

Abstract—*Smart card is not tamper proof but it is temper resistant and temper evident to a degree. Smart card is widely used for business transactions and multiple services in a wide range of industries worldwide to support access, identity, payments and other applications. This paper presents an overview of attacks against smart card implementations and possible countermeasures for attacks that can give background for the assessment of the tools to improve security system of cards.*

1. INTRODUCTION

Smart card (Fig. 1) is a piece of specialized cryptographic hardware that contains its own CPU, memory and OS.

Fig. 1: Smartcard

The self- containment of smart cards make them challenging to attacks as they are not dependent on potentially vulnerable external resources and are mostly used in applications needing strong security protection.

Security has been always big concern for smart card applications though smart card is highly restricted and is unable to interact with the world without outside peripherals. Involvement of other parties i.e. card holder, data owner, card issuer, card manufacturer, card wire manufacturer, terminal owner etc. [1] should not be essentially a threat to one another but further examination is needed in design and analysis of smart card authentication and identification protocol. Numerous intrusion techniques and temper resistant devices have been presented indicating need for effective intrusion and preventive technologies [2]. Multifactor and proximity authentication has been embedded in smart card to increase the security of the card. [3] proposed additional I/O channels such as buttons to alleviate shortcomings.

Threats against Smart card security are:

(a) **Confidentiality** i.e. unauthorized disclosure of information, Get access to keys stored on cards, to clone cards

(b) **Integrity** i.e. unauthorized modification of information, change data stored on card or change behavior of card

(c) **Authenticity** i.e. unauthorized use of service.

Information age has introduced an array of security and privacy issues that have called for advanced smart card security applications. There have been several types of smart card vulnerabilities that could have, or potentially be, exploited. Smart cards security threats are classified as Invasive attacks and Non-invasive attacks. The present paper summarizes Invasive attacks and Non-invasive attacks and possible countermeasures.

The organization of the paper is as follows:

Section 2 and section 3 highlights details of Invasive attacks and Non-invasive attacks respectively. Section 4 pertains to countermeasures for attacks and section 5 concludes the paper.

2. INVASIVE ATTACKS

Invasive attacks also known physical attack exploits use analysis or modification of smart card hardware. The card is physically tempered with using special equipments. The function encapsulated on the chip can be reversed engineered with the help of high end lab equipment [4]. All microprobing techniques are invasive attacks.[5]. Invasive attacks are described as follows.

2.1 DE-PACKAGING AND LAYOUT RECONSTRUCTION OF SMART CARD

Invasive attacks involve de-packaging (i.e. with the removal of the chip package) and applying different physical methods of tampering. In the De-packaging process, etching material dissolves the metal and silicon layers of the chip to de-layer and de-capsulate Smartcards.

Firstly with the help of a sharp knife or any such device cut away the plastic which encapsulates the chip module to make the epoxy resin visible. Next repeat the process five to ten times of settling a few drops of fuming nitric acid on the resin, wait a few minutes and then wash acid and resin away by shaking the card in acetone to fully expose the silicon surface[6].

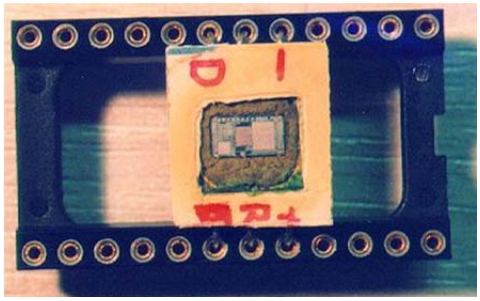

Fig. 2: Smartcard processor depackaged for microprobing experiments.

Next step in invasive attack on a smartcard processor is to create a map of it. The attacker use optical microscope with a camera to study connectivity patterns and tracing metal lines that cross clearly the visible module boundaries i.e. RAM, ROM, ALU, EEPROM, instruction decoder, etc. which is helpful to identify basic structure such as data and address bus lines[5]. Deeper layers can only be recognized in a second series of photographs after the metal layers have been stripped off [5]. Image processing system software reduces the initially fuzzy image to a clean polygon representation and identifies common chip features [7]. If the processor has a commonly accessible standard architecture, then the attacker has to reconstruct the layout only, until he has identified those bus lines and functional modules that he has to manipulate in order to access all memory values [5]. The information about the operations, layouts and functioning can be achieved by various techniques. Developed at Cavendish laboratory in Cambridge, Reverse engineering is the most expensive invasive attack could be used. It gives all the necessary information about chip functioning and layout of protection circuit. Operations of the chip can be observed by the technique developed by IBM in revealing the secret keys.

2.2 MICRO-PROBING WORKSTATION

Micro-probing workstation is the most important tool for invasive attacks with special optical microscope. The attacker installs a metal shaft probe, holding a sharp long tungsten-hair, allowing the attacker to establish electrical contact with on-chip bus lines without damaging them. The probe is 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 pins [5].

2.3 FOCUSED ION BEAM

Expensive tools of advanced beam technologies are being used for new card generations. Vacuum chamber with a particle, gun form focused ion beam workstation. From a liquid metal cathode, Gallium ions are accelerated and focused into a beam imaging samples from secondary particles [5, 7]. By injecting a gas like iodine, greater precision can be achieved and chip material is removed with high resolution by increasing the beam current [5]. To simplify manual probing of deep metal and poly silicon lines, a hole is drilled to the signal line of interest and is filled with platinum to bring the signal to the surface, where a large probing pad of several micrometers created to allow easy access [5]. Electron-beam testers are suitable tools if the clock frequency of the observed processor can be reduced below 100 kHz to allow real-time recording of all bus lines. EBTs are also used if the processor can be forced to generate periodic signals by continuously repeating the same transaction during the measurement [5]. A new technique invented at Sandia National Laboratories in which infrared laser of a particular wavelength is used to create transparent silicon substrate. The produced photocurrents allow probing of the device's operation and identification of logic states of individual transistors [7, 8].

2.4 MEMORY READING

2.4.1 READ ONLY MEMORY

To read ROM directly, optical reconstruction techniques are used. The Diffusion layers store ROM bit pattern and leaves hardly any optical indication of the data on the chip surface [5].In Some ROM technologies additional selective staining techniques have to be applied to make the bits visible as bits are not stored in the shape of the active area but modify transistor threshold voltages [5].

2.4.2 BUS PROBING

It is 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 where all data is available at a single location i.e. memory bus. The entire bus is examined and values are recorded in the memory as they are accessed by Micro-probing [5].

All significant memory locations may not be enough to make the processor access by just replaying transactions. Fortunately, sometimes bus observers may encounter a card where the programmer believed that by calculating and verifying some memory checksum, after every reset the tamper-resistance could somehow be increased. Certainly this process gives the attacker easy access to memory locations on the bus and significantly simplifies completing the Read-out operation [5].

The attacker has to abuse a CPU component to read out all memory cells without the help of the card software. During

every instruction cycle, the program counter is already incremented automatically to serve as an address sequence generator. The processor is prevented from executing jump, call, or return instructions, so that program counter in its normal read sequence is undisturbed. The desired result can be obtained by small modifications of the instruction decoder or program counter circuit that can simply be performed by opening the correct metal interconnect with a laser [5].

2.4.3 THE TEST-MODE

[9] has explained breaking smartcards by using two microprobe needles to link the fuse blown at the end of the card test cycle and using the re-enabled test usual to read out the memory contents.

2.5 KEY RETRIEVAL

2.5.1 ROM OVERWRITING

A laser cutter microscope can overwrite single bit in a ROM which permit the attacker to make code changes leading to disclosure of the key. In DES implementation, the attacker can find bit with the property, that by changing it, key can be extracted easily. The particulars depend on the exact implementation of DES and attackers can make a jump instruction unconditional to reduce the number of rounds [10]. DES S-boxes can be identified and a number of their bits are overwritten such that the encryption function becomes a linear transformation; using linear cryptanalysis techniques key from a single plaintext / cipher text pair can be extracted [10].

2.5.2 EEPROM OVERWRITING

Attacker can modify the contents of EEPROM memory to recover keys. If attacker knows the location of the DES key in EEPROM memory but cannot read it directly, may still derive the key by modifying EEPROM contents [11].

2.5.3 PARITY CHECKS

[2] introduced an EEPROM modification attack in which attacker is assumed to be able to write arbitrary values to locations where the secret key is stored but cannot read a value from the EEPROM as writing can be done with low cost equipment whereas reading require much more expensive equipment [12]. In the process, two micro-probing needles are used to set or clear target bits in order to infer those bits. If one bit of the secret key is set correctly, there would be no error in the output of the device [12].

2.5.4 GATE DESTRUCTION

Keys can be retrieved if the attacker has the talent to harm a gate in a register so that throughout the cryptographic process it is stuck on a constant value in DES. [13] noticed that in case least significant bit of register (that holds the output of round k) is stuck then least significant bit of the output of the round function is set to zero. Several bits of key can be recovered by comparing the least significant six bits of the left half and the right half. This attack works against ciphers such as DES when the plaintext is completely unknown [10].

2.5.5 MEMORY REMANENCE

Even after disconnected from power, RAM may physically keep portions of its contents for a while. The basis of memory remanence attacks is that the values leave magnetic traces when stored for a longer period of time in computer memory and these traces can be used to recover the values [11].The mechanisms that cause both static and dynamic RAM to “remember” values, stored for a longer period of time have explained by [14]. The reliable power-off memory time to minutes or even hours can be extended by cooling the whole circuit with liquid nitrogen or helium to disable the alarm system and reapply the power [7]. SRAM cells may adapt their “preferred” power-up state due to long-term exposure to a constant bit pattern and effect can remain for several days without any supply voltage [7].

2.5.6 PROBING SINGLE BUS BITS

An attacker could easily recover information on the secret key being used by locally observing the value of a few RAM or address bus bits during the execution of a cryptographic algorithm by the mean of a probing needle [15]. In this attack no statistical analysis is needed and is comparatively more powerful. The attacker just has to access a probe station, which for a moment is a kind of needle that during the execution of cryptographic algorithm allows monitoring the value of a single bit. These attacks are not necessarily destructive and attacker just observes a single bit during execution.

2.6 BUS PROBING

2.6.1 ASYMMETRIC ALGORITHMS IMPLEMENTATIONS

The method recovers the exponent of a typical Square-and - Multiply implementation and provides a tool for breaking RSA, DSA, and others [15]. It is assumed that at each execution of the internal loop of Square-Multiply, attacker is provided with the value of certain accumulator bits and collects bit-values just after the accumulator was squared or squared-then-multiplied. The required guess does not increase exponentially and the attack is feasible. Moreover, attacker can guess the position of the bits that he probes during the process itself.

2.6.2 DES IMPLEMENTATIONS

An attacker may retrieve the secret key of a DES implementation by giving one single bit of information at each round [15]. If an attacker uses an electronic station during the execution of DES and observe the value of a given bit,

assuming sufficient knowledge about of the device, any bit of one or the other register is enough to attack the sub keys of first and the last round.

2.6.3 RC5 IMPLEMENTATIONS

This attack is less intricate and requires less than the exhaustive search of a single 32-bit sub key. At each round, the knowledge of a single intermediate bit enables the attacker to derive the complete extended secret key to recover the initial secret key [15]

2.7 EEPROM ALTERATION

The unusual voltage and temperatures can affect write operations of EEPROM and all the key material information of a smart card stored in the electrically erasable, programmable, read only, memory (EEPROM), can be trapped by raising or dropping the supplied voltage to the micro-controller [6].

2.7.1 VOLTAGE CHANGING

By raising the voltage without erasing the memory, the security bit of the controller can be cleared by attack on the PIC16C84 micro-controller [6]. A short voltage drop can release the security lock without erasing the secret data by an attack on the DS5000.Low voltage can make possible other attacks not related to EEPROM. When the supply voltage is lowered, an analogue random generator used to create cryptographic keys will produce an output of almost all 1's [6].

A relatively high voltage is required to erase the charge stored in the floating gate of a memory cell and changes might not be written if the attacker can block this voltage from the card [7].

2.7.2 UV LIGHT

Under UV radiation, security block cell of the EEPROM is used to erase the lock bit to read data in the memory [16].

3. NON-INVASIVE ATTACKS

Non-invasive attacks are not card-specific and card is not harmed physically. The attacker attack for a specific processor type and software version and can reproduce another card of the same type in no time [5].The attacker requires detailed knowledge of the processor and software. The equipment used in the attack can be disguised as a normal smartcard reader. In these attacks attacker has full control over the power and clock supply lines. The attacks often scale well, as the necessary equipment can be reproduced and updated at low cost. To reduce risks, security modules can be equipped with electromagnetic shielding, backup batteries, and autonomous clock signal generators and low pass filters to which smartcard processors are particularly exposed [5]. Non- Invasive attacks are categorized as logical attacks, side channel attacks, Glitch attacks and other attacks.

3.1 LOGICAL ATTACKS

Logical Attacks use bugs in the software implementation effecting confidentiality of data and undesired data modifications. These attacks include Command scan, File system scan, Invalid/inappropriate request, and Crypt-analysis and protocol abuse. A number Hidden Commands are provided by smartcard operating systems in which commands that are active from execution of a previous application can be abused to retrieve data from or modify data. Unexpected results may be achieved from misinterpreted disallowances on the parameters of commands known Parameter Poisoning and Buffer Overflow. There are detailed permissions on files and directories of smartcard file systems. The security procedures to access a file are determined by command access permissions. File access permissions with complex interactions may lead to confusion. Smartcard security can be compromised with Malicious Applets. Communication protocol handles data flow control and error recovery. Cryptographic protocols should be designed cautiously to avoid fall backs with transactions, as consecutive cryptographic operations are handled by cryptographic protocols.

3.2 SIDE CHANNEL ATTACKS

Side channel attacks use physical phenomena to analyze or modify the smartcard behavior. Integrated circuits composed of switching semiconductors, sensitive to basic physical phenomena like electric power and radiation are used by Side Channel Attackers to manipulate the behavior of a smartcard chip. These include use of hidden signals like timing, power consumption, electromagnetic emission etc.

3.2.1 TIMING ANALYSIS ATTACKS

Time measurement of a unit to perform operation can lead to information about the secret keys. Attacker might find fixed Diffie-Hellman exponents, factor RSA keys and break other cryptosystems. In case of vulnerable unit, the attack requires only known cipher text. Performance characteristics usually depend on the encryption key and the input data. Cryptosystems take slightly different amounts of time to process different inputs due to branching and conditional statements, performance optimizations to bypass unnecessary operations, processor that run in non-fixed time, RAM cache hits etc.

Timing attack on the RC5 block encryption algorithm is described by [17]. The principle state that some implementations of RC5 could result in the data-dependent rotations, taking a time that is a function of the data. Assuming that encryption timing measurements enable the cryptanalyst to deduce the total amount of rotations carried out during an RC5 encryption, it is shown that, for the nominal version of

RC5, only a few thousand cipher texts are required to determine 5 bits of the last half-round sub key with high probability [18]. According to [19] statement RC5 is at some risk on platforms where rotations take a variable amount of time and suggests that one should be very careful when RC5 is implemented on such platforms.

3.2.2 POWER CONSUMPTION ATTACKS

We can measure with an analog/digital converter the fluctuations in the current consumed by the card using a resistor in the power supply. Drivers on the address and data bus consist of up to a dozen parallel inverters per bit, each driving a large capacitive load and Integrated circuits act as voltage controlled switches. When charge is applied to the gate, current flowing across the transistor substrate delivers charge to the gates of other transistors, interconnect wires and other circuit loads. The motion of electric charge consumes power and produced electromagnetic radiation is detectable [20]. The different levels of activity in the instruction decoder and arithmetic can clearly distinguish reconstruct parts of algorithms with the help of various instructions. Strongest signals are generated by SRAM write operations. The attacker may be able to identify even smaller signals that are not transmitted over the bus by averaging the current measurements of many repeated identical transactions [5].

Many cryptographic key scheduling algorithms use shift operations that single out individual key bit in the carry flag. They cause changes in the instruction sequencer or micro-code execution, even if the status-bit changes cannot be measured directly. This lead in a clear change in the power consumption [5].

3.2.3 SIMPLE POWER ANALYSIS

Simple Power Analysis (SPA) interprets power consumption measurements collected during cryptographic operations. It capitulate information about operation of operation as well as key material [18]. It can be used to break cryptographic implementations in which the execution path depends on the data being processed, as it can disclose the sequence of instructions executed. Attacker directly observes a system's power consumption and the amount of power consumed varies depending on the microprocessor instruction performed. The operations performed by the microprocessor vary significantly during different parts of these operations and large features such as DES rounds, RSA operations, etc. may be recognized. Individual instructions can be differentiated at higher magnification. The analysis can be used to break RSA implementations by informative differences between multiplication and squaring operations. Similarly, many DES implementations have evident differences within permutations and shifts [20].

***(a)* Simple power analysis on multipliers operation:**

A great deal of information about the data they process may be leaked by Modular multiplication circuits. The leakage functions are strongly correlated to operand values and hamming weights and depends on the multiplier design [21].

***(b)* Simple power analysis on DES key schedule:**

The DES key schedule computation involves rotating 28-bit key registers. A conditional branch is commonly used to check the bit shifted off the end so that "1" bits can be wrapped around. The resulting power consumption traces for a "1"bit and a "0" bit will contain different SPA features if the execution paths take different branches for each [18]. Noteworthy power consumption differences for "0" and "1" bits may result by conditional branching in software or micro-code [21].

***(c)* Simple power analysis on comparison operations:**

In case of mismatch, string or memory comparison operations usually perform a conditional branch causing large SPA characteristics [21].

***(d)* Simple power analysis on exponentiations operation:**

If squaring and multiplication operations take different amounts of time, have different power consumption characteristics or are separated by different code, the exponent can be compromised. Modular exponentiation functions may have more complex leakage functions when operating on two or more exponent bits at a time [21].

3.2.4 DIFFERENTIAL POWER ANALYSIS

SPA attacks mainly use visual inspection to identify relevant power fluctuations whereas to extract information correlated to secret keys, differential power analysis (DPA) attacks use statistical analysis and error correction techniques [20].Besides large-scale power variations, there are effects correlated to data values being manipulated due to the instruction sequence. These variations sometimes are overshadowed by measurement errors and other noise. In such cases, using statistical functions, it is still often possible to break the system modified to the target algorithm [21].

(a) Differential power analysis of Asymmetric Algorithms Implementations: Public key algorithms can be analyzed using DPA. Exponent bit guesses can be tested for modular exponentiation operations by predicting correlation of intermediate values to the actual computation. Defining Selection functions over the CRT reduction or recombination processes can help in analyzing Chinese Remainder Theorem RSA implementations [21]. Because of the relatively high computational complexity of multiplication operations signals, leakage during symmetric operations have a tendency to be less strong than those from asymmetric algorithms [17].

(b) High power differential power analysis: During application of DPA techniques, signals collected from multiple sources using different measuring techniques with different temporal offsets are combined in a high-order DPA attack. No authentic systems are known that are susceptible to High-Order DPA, still we must address high power differential power analysis attacks to be fully effective.

3.2.5 DIFFERENTIAL FAULT ANALYSIS

A crypto systems embodied in smart cards is susceptible to differential fault analysis. Cryptanalyst can compare correct and flawed outputs has a dangerous entry point to the processors internals, including keys, if device can make output under stress [22]. Secret key cryptosystems including DES, IDEA, FEAL and RC5 can be broken by Differential Fault Analysis (DFA). An inducing errors based attacks in instruction code are more informative and easier and leading program execution glitches is better than forcing errors in data [10].

(a) Differential fault analysis attack on DES implementation: [13] explained that DFA attack is applicable to almost any secret key. They described implementation of DFA in DSE case and confirmed extraction of full DES key from a sealed tamperproof DES machine by analyzing fewer than 200 cipher texts generated from unknown clear texts. Even if DES is replaced by triple DES, same attack can break it with the same number of given cipher texts [13]. The attack follows principle that one can induce a fault with rational probability at a random bit location in one of the registers at some random intermediate stage in the cryptographic computation by exposing smart card to certain physical effects. The round number and bit location in which error originated are unknown to attacker. The attack is used to find the last sub key and once sub key is known, the attacker can either use the fact that this sub key contains 48 out of the 56 key bits in order to guess the missing 8 bits in all the possible combinations or he can use the knowledge of the last sub key to skin up the last round and then preceding rounds with the same data, using the same attack.

(b) Differential fault analysis attack on public key algorithms implementation: One can induce faults by physical effects at random bit locations in a tamperproof device at some random intermediate stage in the cryptographic computation with a reasonable probability. Further attacker can repeat the experiment with the same private key by applying external physical effects to obtain outputs due to faults of the tamperproof device [23]. One bit fault at certain location and time can cause fatal leakage of the secret key [23]. This attack also works for multiple bit errors.

3.3 GLITCH ATTACKS

In this attack malfunction is generated deliberately by the attacker that causes one or more flip-flops to take up the wrong state with an objective to replace a single critical machine instruction with an almost arbitrary other one. Glitches are transferred between registers and memory to corrupt data values [5]. Power supply transients, clock signal transients and external electrical field transient techniques are used to create malfunctions that affect only a very small number of machine cycles in smartcard processors. An attacker usually wants to replace conditional jumps or the test instructions preceding them with glitches. Window of vulnerability are created in the processing stages of security applications by which attacker bypass sophisticated cryptographic barriers by simply preventing the execution of the code that detects unsuccessful authentication. Runtime of loops can also be extended by instruction glitches e.g. in serial port output routines, to see more of the memory after the output buffer and to reduce the runtime of loops.

Clock-signal glitches temporarily increase the clock frequency for one or more half cycles, such that some flip-flops sample their input before the new state has reached.[5]. They are simplest and most commonly used attacks. Power analysis is used to monitor how far a program has progressed to find out when a branch instruction is about to be taken. At this point a clock glitch may supply insufficient time for the processor to write the jump address to the program counter, thereby annulling the branch operation [24]. The objective is to apply a glitch in either the clock or the power supply to the chip. There are different number of gate delays in various signal paths and the varying parameters of the circuits on the chip. Only some signals are affected and by varying the precise timing and duration of the glitch, the CPU can be made to execute a number of wrong instructions. These will vary from one instance of the chip to another but can be found by a systematic search using simple hardware.

3.3.1 SPECIFIC GLITCH ATTACK ON RSA IMPLEMENTATION

RSA signature S on a message M modulo n = p*q is computed by computing it mod p and mod q separately. After combining them and using Chinese Remainder Theorem, if an error can be induced in either of the former computations, then the attacker can factor n at once [18]. [2] described another related clock-glitch attack against RSA and DES [11].

3.3.2 SPECIFIC GLITCH ATTACK ON DES IMPLEMENTATION

There are several simple ways to attack DES if we can cause an instruction of our choice to fail. We can take away one of the 8-bit xor operations that are used to combine the round keys with the inputs to the S-boxes from the last two rounds of the cipher and repeat this for each of these key bytes [18]. Erroneous cipher text outputs differing from the genuine cipher text in the output of usually two, and sometimes three S-boxes were obtained by [10]. They obtained about 5 bits of information that were not xor'ed as a result of the induced fault using the techniques of differential cryptanalysis [10]. [10]

described faster attack to reduce the number of rounds in DES to one or two by corrupting the appropriate loop variable or conditional jump. So, DES can be fully compromised with somewhere between one and ten faulty cipher texts. [2] explained clock-glitch attack against RSA and DES [11].

3.3.3 SPECIFIC GLITCH ATTACK ON RC5 IMPLEMENTATION

RC5 is vulnerable to glitches by design of the specific cipher and may be worst possible choice for hardware applications, where some implementations may be vulnerable to glitch attacks [10].

3.4 OTHER ATTACKS

Snooping is unauthorized access to another person's data, an easy ways to intercept and alter data being transmitted over the air. Operation interruption may be created for communication between the reader and the card. Using inapt electronic waves, cards can be destroyed or deflated leading to Rejection of service. Fraudulent merchants using fake readers may generate Hidden transactions. Sharing of underlying chip is done in dual mode chip cards so that the only difference is the way the data is transmitted to the I/O buffer of the chip card.

4. COUNTERMEASURES FOR ATTACKS

Basic smart card security features pertaining to hardware include closed package, memory encapsulation, fuses, security logics (Sensors), cryptographic coprocessors and random generator. Software security features include decoupling applications and operating system, application separation (Java Card), restricted file access, life cycle control and various cryptographic algorithms and protocols. Although smart cards are supposed to be hacker resistant, they are not hacker proof. The vulnerabilities described in the previous sections are intended to give a flavor of some of attacks that can be made against smart cards. There have been recommendations made on how to protect against most of these attacks and smart card community is working hard to address known issues. The countermeasures for attacks are describes as follows.

4.1 COUNTERMEASURES FOR INVASIVE ATTACKS

Invasive attacks require physical manipulations on semiconductors and are powerful and expensive attack class. While the requisite equipment for micro probing microscope, probes, micro positioners, amplifiers may sound expensive, the second-hand market makes all attack equipment available even for individuals. This makes it important to research on low-cost protection mechanisms. Advanced chip designs are accompanied with significant improvement in physical security with following measures.

(a) Alarm: Signals measuring variables such as temperature, light, power supply and clock frequency can be used to disable the chip as soon as out-off-bound situations are detected. So, possibility of live data analysis on a prepared chip can be reduced.

(b) Active grid: An active grid as a top layer carrying protective signals prevents analysis of live data processing. The shielding layer prevents penetration of non-correlated and frequently changing signals.

(c) Feature size: Smartcards are becoming too small for optical microscopes to analyze and to put needles of probe stations.

(d) Multilayer: Sensitive data lines may be hide underneath other layers holding less sensitive connections for multiple layer chips.

(e) Bus scrambling: Advanced non-constant scrambling technique may be used to scramble the data bus between various building blocks.

(f) Glue logic/ redundant logic: Attackers may get confused in analyzing the physical structure of the chip and identifying the functional building blocks by creating glue logic or redundant logic.

4.2 COUNTERMEASURES FOR NON-INVASIVE ATTACK

In non-invasive attack the cryptographic device is essentially exploited as it is and directly accessible interfaces are abused. In these attacks no permanent alteration of device is made. Moreover attacks can be performed with relatively inexpensive equipment and are serious threat to the security of cryptographic devices. A lot of research efforts have been devoted to countermeasure against non-invasive attacks, however effectiveness of these Countermeasures was generally evaluated qualitatively and contained case studies. Following is the description of the countermeasures for non-invasive attacks.

4.2.1 COUNTERMEASURES FOR LOGICAL ATTACKS

Logical attacks are reliant on growing software complexity. With the size of the software, code number of bugs grows and new flaws are introduced. Cautious design and validation are necessary to increase the difficulty of abusing the flaws. Moreover, we should restrict and verify command coding and file access, limit command availability, verify conformance ,test file access mechanisms (PIN, AUT etc.), exclude non-valid behavior, publish algorithms and initiate public discussion, evaluate crypto algorithm and protocol, life cycle management etc. to reduce abuses against cryptographic device.. The countermeasures are described as follows:

(a) Structural strategy may lead software in small functional building blocks for easy understanding and validation.

(b) Proper Verification should be made to verify the soundness of functions using mathematical models.

(c)* E*xperimental authentication of the implementation by testing.

(d) Interface and Application Standardization should be maintained so that using verified software can decreases the chance of flaws.

(e) An object oriented language i.e. ***Java Card Operating System*** is conceptually more secured than the older monolithic operating systems.

(f) Evaluation Labs should be monitored closely for Valuation and certification.

4.2.2 COUNTERMEASURES OF SIDE CHANNEL ATTACKS

Defense against side channel attacks is never expected to be absolute and sooner or later attacker will be able to break an implementation with immense amount of resources. The engineering contest is driving in enough countermeasures to make attack too expensive. Countermeasures products offering high level of security can be implemented at different level. At the transistor level circuits and logical gates may be built so that the information leakage is reduced. Dummy instructions can be inserted randomly to make the alignment of traces more difficult at the program level. Cryptographic algorithm operations are computed to reduce information leakage at the algorithmic level. At this level defense depends on the basic operations used in the algorithm so, choice of operations in the cryptographic primitive is relevant. The protocol may be designed to limits the number of computations an attacker can provoke with a given key at protocol level. Following countermeasure reduces vulnerability to side channel attacks.

4.2.2.1 ***Hardware Countermeasures*** decrease the signal to noise ratio and reduce the vulnerability to side channel analysis making attack more difficult. Power signal can be lowered by balancing the circuits and reducing electromagnetic emissions. Noise level amplitude can be increased by carrying out concurrent random processes. To prevent or hamper alignment of traces, process interrupts and variable clock speeds with timing noise may be introduced.

4.2.2.2 ***Software Countermeasures*** are used to reduce the emission of useful information from the side channels by decreasing the signal to noise ratio. To reduce applicable signals, random process checking for parallel algorithm substitutions may be performed. To deteriorate the quality of the differential trace and hamper alignment of traces, random delays or alternating paths may be performed to add timing noise. To eliminate time dependencies in key material and intermediate values, time constant key operations may be implemented and simple power analysis by visual inspection of traces can be avoided. To prevent useful information leakage, random values should be added that are to be subtracted later to blind intermediate values. These are cautiously designed to compensate the deviation caused by random values.

4.2.2.3 ***Application countermeasures*** include PIN verification blocks guarding against differential analysis. Performance of differential analysis can be reduced by limited input and output visibility of cryptographic algorithms.

To sum balancing or equalizing the power, shielding the emission to reduce processor signal, added noise to processor activity, eradication of time relation of processed key and data, flexible ordering of processes, striking of intermediate values with random values, redo counters and crypto input and output with limited control and visibility are important countermeasures against side channel attacks

4.2.3 COUNTERMEASURES FOR POWER GLITCHING

The most common strategy against power glitching attacks is adamant use of sensors for voltage, frequency and temperature. However, malfunctions may be produced in some terminals due to sensor setting. To detect and recover from fault injection software and application countermeasures may be implemented. Fault detection is performed by Checking the crucial program flow decisions and cryptographic conclusions. Light, supply voltage, frequency detectors, active shield and hardware redundancy are some active protections whereas checksums, execution randomization, variable redundancy, execution redundancy, ratification counters and baits are some passive protections that increase the difficulty of successfully attacking a device.

4.2.4 COUNTERMEASURES FOR OTHER ATTACKS

Users should update knowledge of Smartcards properties to address the abuses accordingly. Crypto-coprocessors and Public Key Cryptography needs to be enhanced from time to time. Important aspect to be monitored is encryption of the data being exchanged and mutual authentication.

5. CONCLUSION

This paper outlined attacks against smart card implementation and some basic countermeasures. Smartcards are precisely designed for security and integrates mechanisms for detection of recovery from security problems. Smart cards are secure and getting everyday better but are not perfect. A regular revision and risk analysis on the emerging threats should be measured to compare options for the assessment and improvement of mechanisms to ensure desired level security.

REFERENCES

[1] Schnier B., A. Snostaxck (1999), “Breaking us is hard to do: modeling security threats for smart cards”, *USENIX workshop on smart card technology*, *USENIX press* pp 175-185.

[2] Anderson R., and M. Kuhn (1997), How cost attacks on temper resistant devices", security protocol.

[3] Gobioff et al. (1996), "Smart card in hostile environment", in proceeding of the second USENIX workshop on electronic commerce, Oakland CA.

[4] Hoon Ko & Ronnie D. Caytiles (2011), "A Review of smart card security issues", *journal of security engineering.*

[5] "Design Principles for Tamper-Resistant Smartcard Processors", By Oliver Kommerling and Markus G. Kuhn

[6] "An Overview of Smart Card Security", By Siu-cheung Charles Chan

[7]"Tamper Resistance - a Cautionary Note", By Ross Anderson and Markus Kuhn

[8] "Tamperproofing of Chip Card", By Ross Anderson

[9] E Bovenlander, "invited talk on smartcard security", Eurocrypt 97

[10] "*Low Cost Attacks on Tamper Resistant Devices"*, By Ross Anderson and Markus Kuhn

[11] "Smart Cards and Private Currencies", By J. Orlin Grabbe

[12] "Cryptanalysis of the m-Permutation Protection Schemes", By Hongjun Wu, Feng Bao, Dingfeng Ye and Robert H. Deng

[13] "Biham-Shamir Differential Fault Analysis of DES", By Eli Biham and Adi Shamir

[14] Peter Gutmann, "Secure deletion of data from magnetic and solid-state memory", 6th USENIX Security Symposium. San Jose, California, July 22-25, 1996.

[15] "Probing Attacks on Temper-Resistant Devices", By Helena Handschuh, pascal Paillier and Jacques Stern.

[16] "*Java Smart Cards"*, By Y. L. Chan and H. Y. Chan

[17] *"A Timing Attack on RC5"*, By Helena Handschuh and Howard M. Heys

[18] "Known attacks against Smartcards", By Hagai Bar-El, Research Group,Discretix Technologies.

[19] "*DPA Q&A"*, By Paul Kocher, Joshua Jaffe, and Benjamin Jun

[20] "Introduction to Differential Power Analysis and Related Attacks", By Paul Kocher, Joshua Jaffe, and Benjamin Jun

[21] *"Differential Power Analysis"*, By Paul Kocher, Joshua Jaffe, and Benjamin Jun

[22] "Defending against DFA", By Sandy Harris

[23] "*New Attacks to Public Key Cryptosystems on Tamperproof Devices"*, By Feng Bao, Robert Deng, Yongfei Han, Albert Jeng, Desai Narasimhalu and Teow Hin Nagir

[24] "Improving Smartcard Security Using Self-timed Circuit Technology", By Simon Moore, Ross Anderson, and Markus Kuhn