ISSN 2348-1196 (print)
International Journal of Computer Science and Information Technology Research ISSN 2348-120X (online)
Vol. 3, Issue 1, pp: (427-441), Month: January - March 2015, Available at: www.researchpublish.com

Latest Side-Channel Attacks and Its Countermeasure Attacks: Attacks Based On Cryptography

HARSHIL.B. JANI

M.TECH (E.C), INDUS UNIVERSITY

Abstract: **latest Side-channel attacks are very easy to execute powerful attacks against crypto graphic implementations, and their targets range from primitives, protocols, modules, devices to even systems. These attacks harm a serious threat to the security of cryptographic modules. Also, cryptographic implementations have been executed for their resistivity against such attacks and the disturbance of different countermeasures has to be considered. This paper also deals with the methods and techniques employed in these attacks, the destructive effects of such attacks, the countermeasures against such attacks and evaluation of their feasibility and applicability.**

Keywords: **Information security, Side channel attack, Cryptographic module.**

1. INTRODUCTION

Security has long been a major concern in computing and communications systems, and substantial research effort has been developed to address it. Cryptographic algorithms, including symmetric ciphers, public-key ciphers, and hash functions, form a set of primitives that can be used as building blocks to construct security mechanisms that are target to specific objectives. For example, network security protocols, such as SSH and TLS, combine these primitives to provide authentication and security between communicating entities, and ensure that the confidentiality and integrity of communicated data. In practice, these security mechanisms only specify what functions are to be performed, irrespective of how these functions are executed. For example, the specification of a security protocol is usually independent of whether the encryption algorithms are implemented in software running on an general processor /using custom hardware units, and whether the memory used to store intermediate data during these computations is on the same chip(same unit) as the computing unit or on a separate chip.

This kind of "separation of concerns" between security mechanisms and their implementation has been enabled (and is, arguably, necessary for) r theoretical analysis and design of cryptosystems and security protocols. Further, in the process, various assumptions are made about the implementation of security mechanisms. For e.g, it is typically assumed that the implementations of cryptographic computations are ideal "black-boxes "also whose internals can neither be observed nor be disturbed with by any malicious node. Aided by these assumptions, the level of security is widely used in quantified in terms of the mathematical properties of the cryptographic modules and algorithms and their key sizes.

Also, however, these security mechanisms alone are being complete security solution For security purpose. It is wrong to assume that attackers will attempt it directly to take on the computational complexity of breaking the cryptographic primitives employed in security mechanism. An interesting analogy can be done in this regard between strong cryptographic algorithm and a powerful secure lock on the front door of a home. Thief attempting to break into a home will not try for combinations necessary to pick such a lock; they may break in through windows, break a door at its hinges, or rob owners of a key as they are to get into the home.

We can say that most of all known security attacks on cryptographic systems achieve weaknesses in the implementation and deployment of mechanisms and their cryptographic algorithm. Thus weaknesses of the attackers tend to completely go, or probably weaken, the strength of security solutions. Involved in it.

Further adding, a cryptographic system remains secure it is possible that the secret keys, that it uses to perform the required security services, are not useful in any way. Since cryptographic algorithms themselves have been observed for a long time by a panel of experts, hackers are likely to attack on the hardware and system within which the cryptographic unit is homed.

A new VERSION OF CLASS of attacks has been introduced in the last few year by Kocher. These attacks work because there is a correlation between the physical measurements taken at different points during the computation and the internal state of the processing device, which is related to the secret key algorithm.

Rarely, in, cryptographic algorithms are always implemented in software or hardware which is being dealt on physical devices which are interactful with and are influence by their environments and surroundings.. These physical interactions can be observed and monitored by adverse, like Attacker, and may result in information useful in cryptanalysis. This type of information is called side-channel information, and the attacks exploiting side-channel information are called side-channel attacks (SCA in the sequel). Thus underlying idea of SCA attacks is to look at the way cryptographic algorithms are implemented, rather than at the algorithm itself.

It is not easy to see that conventional cryptal analysis threats cryptographic algorithms as purely mathematical objects, among the side-channel cryptal analysis also takes the implementations of the algorithms into account of attacker Hence, SCA attacks are also called implementation attacks word is used. Even any cryptographic algorithm must be encoded in order to function properly, so such an encoded algorithms must not yield to the private key information used, despite the adversary's to built up an observe and manipulate to the running algorithm.

The first official information which is used to related to SCA attack dates back to the year 1965. P. Wright (a prominent scientist with GCHQ at that time) reported in [that MI5, the British intelligence agency, had tried to broke a cipher text which is used by the Egyptian guy in London, but with best efforts were done by them to stop of their computational power. so, Wright suggested placing a microphone near the rotor-cipher machine which gave them a huge robust used by the Egyptian to spy the click-sound the machine produced. So, By listening to the clicks of the rotors as cipher clerks reset them in each morning they observed it successfully.

Thus, MI5 successfully deduced the core position of 2 or 3 of the machine's rotors. This additional information reduced the computational effort needed to break the cipher, and could spy on the embassy communication for years and months.

On the other hand, the original seminal work is being carried out, as well as many other ideas, on SCA attacks on public cryptography research community are all due to the help of MR.PAULKOCHER.

The main principle of SCA attacks are very easy to catch by. SCA attacks work because there is a huge link between the physical measurements taken during computational attempt (e.g., power consumption, computing time, EMF radiation, etc.) and the internal state of the given processing device, which is itself related to the secret key algorithm.

It is the correlation which is in between the cipher & the side channel information and the operation is related to the secret key that the SCA attack tries to locate SCA attacks have been proven to be several orders of magnitude more effective that the conventional method is more helpful in analysis of S.C.A based attacks and are much more practical TO implement it.

In the given area of protocol which is used to design or even software develop, one can apply a range of techniques to model the device in question, to model the range of affects & actions, and then to reason about the correctness properties to the given the device is supposed to provide useless. One can obtain at least some confirmation about it that, it is within the block diagram of the model, the device may resist malicious attacks.

Further attacks, when we move from an basic notion of security to its instant as a real process in the physical world, things become difficult to implement. All the real-world scenario is that the model of it is insignificant. Now is the boundary of this cryptographic device is, in the high. Also, What is the outputs that an attacker might observe, and the inputs an enemy may manipulate in order to act on the given device.

These answers are hard to achieve, but designe of architecture is very hard to defend against arbitrary attacks which requires an attempt to get them.

Moreover, the physical action of computational effect is seen in can result in physical effects an attacker may observe; these observations can sometimes betray sensitive internal data the cryptographic module architecture was supposed to protect them. This style attack of is also called side-channel analysis (S.C.A), since its implementation of given the module or device leaks information via other channels other than its main intended interfaces are affected it.

By physically looking in to a cryptographic device, the attacker hopes to defend its security properties also somehow, by extracting it by some secret the device was not supposed to reveal Its identity.

At first glance, the natural way to achieve the above goal is the direct approach on:

(1) Somehow to pass the cryptographic modules protection.

(2) To be fortunate, in design practice, this direct attack can be easily hacked by so called tamper-resistant technique which is observed. Even though this direct approach cannot often prove rather successful, a rather sophisticated family of indirect approaches has been emerged towards it, where the attacker instead tries to check an error into the module operation via some link failure; if the module continues to operate under given the error, it may ends up reveals through leaked information for the attackers to reconstruct the secret. Researchers are at Bellcore lab which originally described this attack, in a theoretical context of inducing errors in cryptographic security that carried out the RSA attacks on it. This results in it.

The above result generated via a disturbance of follow-on results, some of which became known as differential fault analysis. The theoretical attacks eventually become a practical and demonstrable, and eventually earned the name Bellcore attacks after the authors of their original work.

Jargon attack of system security today may be the Trusted Platform Module (TPM in the sequel). TPM usually takes the form of a cryptographic secure module and is the core of the trusted cloud computing platform. A key component of such cryptographic modules is that they keep and use secrets, despites attempts bynan attackers on — perhaps with direct physical access — to obtain them.

Single-chip devices — particularly smart cards — which have received much attention in the attacker community, perhaps due to the misuse of smart cards in low-end commerce applications dealing towards it (providing motivation), and the low cost (making experiment and destructive analysis feasible for device).Referring to Anderson and Kuhn's works also provides an huge enlightening (and entertaining) survey of the various techniques they found effective in practice.

Recently, two attacks which are related to SCA research in Europe should caught a catch the eyes of the cryptography community worldwide, especially those who are interested in the research of SCA attacks: SCARD (Side Channel Analysis Resistant Design Flow) project and ECRYPT(European Network of Excellence for Cryptology) project . Both of these two projects: international joint project plans among European research members from both cryptography research institutes and relevant industries.

In SCARD, it was proposed to enhance the typical micro- flow — from high level system to given description over register transfer layer description down to gate level net lists, and finally placement and routing of the micro-chip—in order to provide for designing side-channel analysis resistant circuits and systems.

Moreover, it is referred to study the whole phenomenon of side-channel analysis in a consistent manner, and also to provide appropriate analysis tools to design tools for the designer of secure systems. In fact, it is observed that these additional ingredients of the traditional design flow of microchips are considered to be necessary in order to obtain the design of the next generation of secure and dependable devices such as ECRYPT is a 4-year layer network of excellence funded within the Information Societies Technology Programme of the European Commission which falls under the action line towards a global dependability and security framework and its objective to intensify the given collaboration in European researchers in information security, and more in particular in cryptology and digital watermarking.

In order to reach this goal, leading players integrate their research capabilities within 5 virtual labs focused on different core research areas, with one being secure and efficient implementations.

One of the four Working Groups of VAMPIRE is the research group on SCA analysis.

From these two attackers alone, it is estimated that the Europe, in opinion, it is likely one step further ahead over the other continents in the internationally collaborative research on SCA attacks based on research.

It is an interesting story based on assumption that SCA attacks evaluation was already explicitly had an idea many years ago it was encompassed in cryptographic algorithm evaluation in many international standards bodies, such as 3GPP security architecture.

However, due to lack of methods of research and practical availability, this suggestion virtually is like empty shapes in sight. So it is easy to execute that the final evaluation report of these standard bodies draw the conclusion at that time that in the designing process it was not to be able to design a general algorithm which have a framework that it would not be so leaky to side channel attack in it.

Recently someone presented a digital VLSI design flow to create safe security. Even though this is the first significant attempt in the secure design of IC and chip design, they only considered the power analysis attack in the top-down automated synchronous VLSI design flow that has constant power dissipation. a systematic security system design approach is useful.

In this case, the concept of trusted code base was introduced, which resembles the trusted computing base in the context of safe and secure operating system.

The threat of SCA attacks also gave a to caught the attention from research community presented a framework for security of by providing network level symmetric key cryptography for key distribution and at the core level on illustrating modification of software with extremely high low overheads for added security against power attacks .

So carefully observing a cryptographic algorithm which is strong with respect to conventional cryptanalytic attacks is not useful & if it cannot be implemented securely on a broad range of platform. Already during the AES process, the cryptographic community has come to this way of conclusion.

Some basic Parameters of:

(1) To understand the history AND Nature of SCA attacks.

(2) To recognize the serious threats which are harmful to SCA attacks .

(3) To acknowledge the various countermeasures and steps against SCA attacks.

(4) To evaluate and measure the impacts of SCA attacks on the security testing of cryptographic module.

(5) To identify the possible research trends in this area and so on FOR FURTHER RESEARCH.

2. MODELS OF SIDE CHANNEL ATTACKS

A cryptographic is the basic idea behind:

(1) It can be Observed as an idea for mathematical object (a transformation, possibly by a key, giving some input in to some output); on the other hand, this METHOD will in fine have to be implemented in a program that will run on a given processor, in a given environment, and will therefore present specific characteristics.

- **ADVANTAGES OF SCA:**

1. Side-channel cryptanalysis takes advantage OF implementation-specific characteristics to recover the secret parameters involved in the computational algorithm
2. It is therefore much less useful — since it is specific to a given implementation but often much more powerful than cryptanalysis, and is considered very serious by cryptographic devices implementors.
3. In traditional cryptanalysis, when checking the security of a cryptographic protocol, one usually assumes that the adversary has a complete description of the protocol, is in possession of all public keys, and is only lack of knowledge of the secret keys. In advance , the attackers may have interest in some data exchanged between the legitimate participants, and may even have some control over the nature of this data (e.g., by selecting the messages in a chosen-

ISSN 2348-1196 (print)
International Journal of Computer Science and Information Technology Research ISSN 2348-120X (online)
Vol. 3, Issue 1, pp: (427-441), Month: January - March 2015, Available at: www.researchpublish.com

message attack on a given signature , or by selecting the cipher text in a chosen-cipher text attack on a public-key algorithm scheme).

We can conclude that, the attacker attempts to compromise the protocol goals by either solving an underlying problem assumed to exploit, or by exploiting some design flaw in the protocol.

In this process, mathematical WAY can be a very useful tool in the study of cryptographic primitives. Cryptographers often evaluate the security of ciphers by considering them as mathematical functions used in the scenario similar to the one described in Figure 1.

Traditionally, secure cryptographic algorithms provide security against an adversary who has only black-box access to the secret information of honest parties. However, such models are not always adequate. In particular, the security of these algorithms may completely break under (feasible) attacks that tries to tamper with the secret key algoriythm..

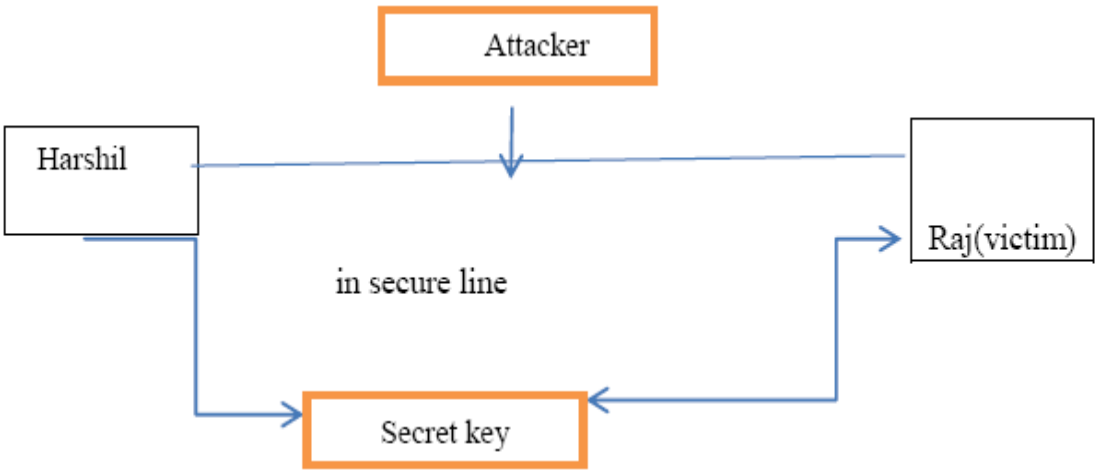

Figure 1: Basic idea of traditional cryptographic model

The attacks considered in this traditional security model exploit concept used by RAJ & HARSHIL the mathematical specification of the protocol. In recent years, researchers have become increasing awareness of the possibility of attacks that exploit specific properties of the implementation and operating environment.

By Given Such SCA attacks utilize information leak during the protocol's execution and are not considered in traditional security models. For e.G, the may be able to monitor the power consumed or the EMT radiation emitted by a smart card while it performs private-key operations such as decryption and signature generation.

The attacker may also be able to measure the time it takes to perform a cryptographic operation, or analyse how a cryptographic device behaves when certain errors are encountered. Side-channel information may be easy to gather in practice, and therefore it is observe that the threat of SCA attacks be quantified when assessing the overall security of a system, see the scenario illustrated in Figure 2.

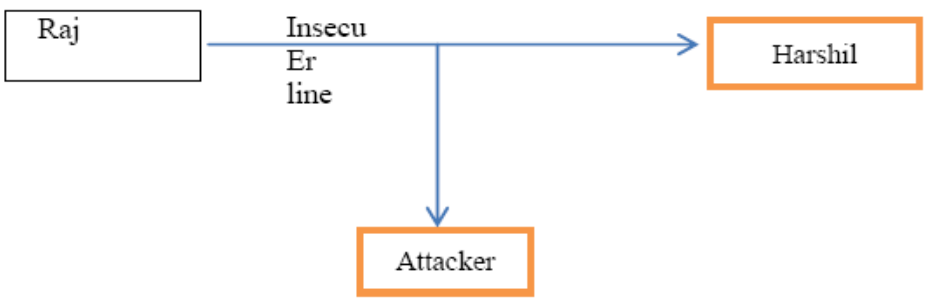

Figure 2: basic idea using the block of cryptographic model including side-channel

Side Channels are defined as an uninterrupted output channels from a system. Paul Kocher 1996 published the seminal paper "Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS, and Other Systems" showing that un-constant running time of ciphertext cans leaky information about the key. When implementations take advantage of optimizations, the problem may become more complicated.

It should be emphasize on the fact that a particular side-channel attack may not be a realistic fact on threat observed threat in some environments. For example, attacks which measures power consumption of a cryptographic device can be considered very plausy if the device is a smart card that draws power from an external, untrusted device. On the other hand, if the device is a work station located in a secure office, then power consumption attacks are not a significant threat is considered

3. LEVEL BASED ATTACKS

The standard to be used by (US) Federal organizations when specifying cryptographic-based security systems is to provide protection for sensitive or valuable data (maintaining the confidentiality and integrity of the given information). The standard specifies the security requirements to be satisfied by a cryptographic module in four increasing, qualitative levels of security (Level 1 to 4, from low to high) as summarized in the following:

● Level 1: which provides the lowest level of security? It specifies basic security requirements for a cryptographic module. (For software implementation only).

● Level 2 which improves the physical security of a Security Level 1 cryptographic module by adding the requirement for tamper evident coating/ seals, for pick-resistant locks.

● Level 3 requires enhanced physical security, attempting to prevent the attacker from gaining access to critical security parameters which are held within the module.

● Security Level 4 provides the highest level of security. Level 4 physical security provides an envelope of protection around the cryptographic module to detect a problem penetration of the device from any direction.

These levels are intended to cover the wide range of potential applications and environments in which cryptographic modules may be covered. The security requirements cover eleven areas related to the secure design and implementation of the cryptographic module.

These areas include the following:

(1) Cryptographic module specification

(2) Cryptographic module ports and interfaces roles, services, and authentication;

(3) Finite state model;

(4) Physical security;

(5) Operational environment;

(6) Cryptographic key management;

(7) Electromagnetic interference/electromagnetic compatibility (EMI/ EMC)

(8) Self tests

(9) Design assurance and affect of other attacks.

The standard protocol is designed after five years., is standard specifies the security requirements to be satisfied by a cryptographic module used within a strict security system protecting unclassified information within computer and telecommunications systems (including voice systems.

Actually, there are problems with the current version of. First of all, this version of standard is mainly focused on hardware modules, and is not well adapted to software modules. It is expected this status may change in the coming version of it. Secondly, this version of standard covers somewhat too narrow scopes of the system to be tested. Better alignment with the Common Criteria is required, to it and the security issues of functional protocols need to be

addressed better. Finally, the functional requirements of this version of standard are already not available. The requirements specified in FIPS have lagged behind the actual needs of the information security both in theory as well as practice.

Meanwhile, the complexity of the given process of validation shows an excellent in creating solid cryptographic algorithms and modules is difficult to achieve. Of the eleven areas, the following four areas are likely of greatest difficulty: physical security, self-tests, random number generation and key management.

Specifically, as far as SCA attack is concerned, FIPS 140-1 did not explicitly mention the security of cryptographic modules against side channel attacks, while only deal briefly with the specification of given of attacks for which no testable requirements are not currently available.

4. CLASSIFICATIONS OF SIDE CHANNEL ATTACKS

Side channel attacks are usually classified in literatures along the following three orthogonal axes:

(1) Classifications depending the control over the computation process;

(2) Classifications depending on the way of accessing the module;

(3) Classifications depending on the method used in the analysis process.

4.1 Controls over the Computation Process:

Depending on the control over the computation process by attackers, SCA attacks can be broadly divided into two main categories: passive attacks and active attacks. We have refer passive attacks to those that do not noticeably disturb with the operation of the target system; the attacker gains some information about the target system's operation, but the object system behaves exactly as if no attack occured.

- In active attack, on the other hand, the attacker impacts some influence on the behavior of the object system. While the actively attacked system may / may not be able to detect such influence, an outsider observer would notice a difference in the operation of the system. It is mainly important to note that the distinction between active attacks and passive attacks has more to do with the intrinsic nature of the attack than the intrusiveness of any physical device and implementation of the attacker.

4.2 Ways of Accessing the Module:

- When analyzing the security attacks of a cryptographic hardware module, it can be said that the useful to perform a systematic review of the attack surface — the set of physical, electrical and logical interfaces that are exposed to a potential opponent.
- According to this observation, and REFERENCE OF: Anderson divided the attacks into the following classes:
- Invasive attacks, semi-invasive attacks and non-invasive attacks.

4.2.1 Invasive Attacks:

An Invasive attack involves deploying to get direct access to the internal components of cryptographic modules or devices. A typical example of this is that the attackers may open a hole in the passive layer of a cryptographic module and place a probe in to the needle on a data bus to see the data transfer.

Tamper resistant or responsive mechanisms are usually implemented in hardware to effectively counter invasive attacks. For e.g, some cryptographic modules of higher security level will affect all their memories when tampered data are detected.

4.2.2 Semi-invasive Attacks:

The concept of semi-invasive attack is first developed by Skorobogatov and Anderson [95]. This kind of attack involves the access to the device, but without affecting the passive layer or making electrical contact between the other than with the authorized user. For e.g, in a fault-induced attack, the attacker may use a laser beam to ionize a device to change some of its memories and thus change the output of this device. For security purpose.

4.2.3 Non-invasive Attacks:

A non-invasive attack involves close observation/manipulation of the device's operation. This attack only exploits externally available information that is often unintentionally leaked. A typical e.g of such an attack is timing analysis:

(1) Measuring the time consumed by a device to execute an operation and correlating this with the computation performed by the device in order to deduce the value of the secret keys.

(2) One important characteristic of non-invasive attack is that this attack is completely undetect. For e.g., there is no way for a smart card to figure out that its running time is currently being measured. On the other hand, compared with invasive attacks that require individual processing of each attacked device, non-invasive attacks are usually of low-cost to deploy on a large scale from an economical point of view. In this sense, non-invasive attacks constitute therefore a bigger WAY for the smart card industry.

4.3 Methods and ways Used in the Analysis Process:

Depending on the methods used in the process of analyzing the sampled data, SCA attacks can be divided simple side channel attack (SSCA in the sequel) and differential side channel attack (DSCA in the sequel).

In a SSCA, the attack exploits the side-channel output mainly depending on the performed operations. Typically, a single trace is used in an SSCA analysis, and therefore the secret key can be directly read up to from the side-channel trace. Obviously, the side-channel information is related to the attackers instructions (the signal) needs to be larger than the side-channel information related to the unrelated instructions (the noise) .

On the other hand, when SSCA is not advisable due too much noise in the measurements, DSCA using statistical methods is tried. In DSCA, the attack exploits the side-channel output mainly depending on the performed data. Typically, many traces are used in a DSCA analysis, and then statistical methods are used to deploy the possible secret keys. With regard to this, one can claim that DSCA is more powerful than SSCA.

Differential side-channel attackers involve the correlation between the data and the instantaneous side-channel leakage of the cryptographical device. As this correlation is usually very small, statistical methodical must be used to exploit it efficiently. In a differential side-channel attack, an attacker uses a hypothetical model of the device under attack. The quality of this model depends on the capability of the attackers.

The hypothetical model is used to predict that the side-channel output of the deviceis not it may output several values. These could be either values describing one type of information leakage for several time slots, or it could be values predicting the leakage of different side-channels. In case only one single output-value is used for an attack, then the attack is called first-order attack.

- If two or more output values for the same side-channel are used in an attack, then the attack is called second-order attack and higher-order attack, respectively.

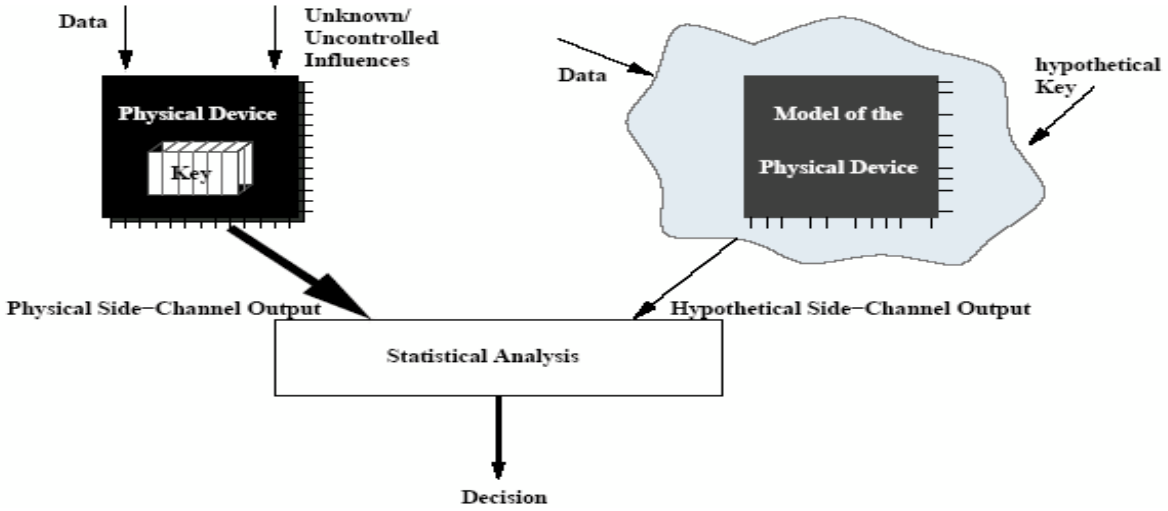

Figure 3: The general idea behind DSCA [132]

Conclusion: The above three axes sometimes are well orthogonal: an invasive attack may completely avoid disturbing the device's behavior, and a passive attack may require a preliminary deploying for the required information to be observable, an active and invasive attack may also belong to the DSCA.

5. BASIC KNOWN SIDE CHANNEL ATTACKS AND CONCRETE COUNTERMEASURES ATTACK

In this section, we will review the known methods and techniques employed in SCA attacks, the destructive effects of such attacks, the countermeasures proposed so far against such attacks and evaluation of their attacks and applicability.

So far, SCA attacks have been successfully used to break the hardware or software implementations of many cryptosystems including block ciphers(such as DES, AES, Camellia, IDEA, Misty1, etc.), stream ciphers(such as RC4, RC6 ,A5/1, SOBER-t32, etc.), public key ciphers(such as RSA-type ciphers, ElGamal-type ciphers, ECC, XTR, etc.), to break the implementations of signature schemes, to break the message authentication code schemes, to break the implementation of cryptographic protocols, to break the implementation of cryptosystems, and even to break the networking systems.

As many security experts have pointed out, security does not equal to cryptography and good cryptographic algorithms do not automatically guarantee the security of application systems. Every component is secure does not necessarily mean that the whole system is secure. For complex systems, security should be studied under various attacks from various angles very carefully. Without doubt, SCA attack is definitely such kind of useful angel to be explored more throughly.

5.1 Known Side Channel Attacks:

SCA attacks against cryptographic modules exploit characteristic information extracted from the implementation of the cryptographic primitives and protocols. This characteristic information can be extracted from timing, power consumption or electromagnetic radiation features. Other forms of side-channel information can be a result of hardware or software faults, computational errors, and changes in frequency or temperature. SCA attacks make use of the characteristics of the hardware and software elements as well as the implementation structure of the cryptographic primitive. Therefore, in contrast to analyzing the mathematical structure and properties of the cryptographic primitives only, side-channel analysis also includes the implementation.

All these facts sum up to one fact that the concrete implementation is very critical to security and a tiny difference in implementations could make a big difference in security. Hence engineers who implement the security schemes should be very carefully in following every step of the schemes. Moreover, attackers will more likely choose the weakest link in the security chain. When peer reviewed cryptographic algorithms and protocols are used, cryptanalysis will almost certainly not be the weakest link. Systems designers must strive to be aware of unintentional "back doors" which are not secure against attacks.

Until today, at least more than ten kinds of important side channels have been explored. We will discuss them one by one in this section.

5.1.1 Timing Attacks based on attacker:

Implementations of cryptographic algorithms often perform computational performance in non-constant time, due to performance optimizations. If such operations involve secret parameters, these timing variations can leaky some information and, provided enough knowledge of the implementation is at hand, a careful statistical analysis could even lead to the total recovery of these secret parameters. This idea was introduced by Kocher and was developed in which a practical timing attack against an actual smart card implementation of the RSA was conducted.

A timing attack is, essentially based on useful a way of obtaining some user's private information by carefully measuring the time it takes the user to carry out cryptographic module. The principle of this attack is very simple: to give exploit the timing variance in the operation.

The basic assumptions of timing analysis are:

1. The run time of a cryptographic operation dependent to some extent on the key. With present hardware this is likely to be the case, but note that there are various efficient hardware based approach proposals to make the timing attack less.

Software approaches to make the timing attack useless are based on the idea that the computations in two branches of a conditional should take the same amount of time

2. A sufficiently large number of encryptions can be carried out, during which time the key does not change. A challenge response protocol is ideal for timing attacks.

3. Time can be measured with known error. The smaller the error, the fewer time measurements are required.

Timing attacks were introduced in 1996 by Kocher where, RSA modular exponentiation was being attacked. Schindler presented timing attacks on implementation of RSA exponentation that employ the Chinese Remainder Theorem (CRT in the sequel) Experimental results for an RSA implementation on a smart card were reported by Dhem et al. [55].

OpenSSL is a well-known free (open source) crypto library which is often used on Apache 2 server on it At the 10th senix Security Symposium, presented timing analysis of keystrokes and timing Attacks on SSH protocol. They applied traffic-analysis techniques to interactive SSH connections in order to infer information about the encrypted connection contents. They concluded that the keystroke timing data observable from SSH implementations reveals a dangerously significant amount of information about user terminal sessions — enough to locate typed passwords in the session data stream and reduce the computational work involved in guessing those passwords by a factor of 50.

By observing the timing of the reject signs from the decryption oracle, Sakurai et al. [52] presented a timing attack against the EPOC-2 public-key cryptosystem that was proved to be IND-CCA2 secure under the factoring assumption in the random oracle model. More interestingly, EPOC-2 was already written into a standard specification P1363 of IEEE, and has been a candidate of the public-key cryptosystem in several international standards (or portfolio) on cryptography, e.g. NESSIE, CRYPTREC, ISO, etc.

Basically, countermeasures for timing attacks must be modelled more rigorously so that we can study how effective the proposed measures are. Two common countermeasures that are currently in use (i.e. noise injection and branch equalization) appear to be fundamentally different in the sense that noise injection weakens the power of the timing attack but it does not defeat it, whereas branch equalisation does defeat the attack but at significant costly.

It it worth noting that even the timing attack exploits the timing variation in each operation of the algorithm, the individual timing of each operation can not be measured in practice. Only the total executing time of all the operations of the algorithm can be measured, and then statistical methods are being applied to deduce (part of) the secret key.

5.1.2 Faulty Attack:

Mostly all of the devices that perform various cryptographic operations are usually assumed to operate reliably when we use them, so we might not think to question if the security of such operations depend on the reliability of these devices that implement them. In spite of this assumption, hardware faults and errors occurring during the operation of a cryptographic module in fact have been demonstrated to seriously affect the security. These faulty behaviors or outputs may also become important side channels, and will even greatly increase a cipher's vulnerability to cryptanalysis sometimes. Fault attacks present practical and effective attacking against the cryptographic hardware devices such as smart cards. Therefore, we mainly focus on the fault attacks on hardware devices here.

Fault attacks on cryptographical algorithms have been studied since 1996 and since then, nearly all the cryptographic algorithms have been broken by using such kinds of attacks. Fault attacks offer the attacker plenty of possibilities to attack a cryptosystem. The ways to exploit a faulty result are very different from one algorithm to another. The feasibility of a fault attack (or at least its efficiency) depends on the exact capabilities of the attackers and the type of faults she can induce.

Generally, a fault model should at least specify the following aspects:

- The precision an attacker can reach in choosing the time and location on which the fault occurs during the execution of a cryptographic module.
- The length of the data affected by a fault; for example, only one bit, or one byte.
- The persistence of the fault; whether the fault is transient or permanent.
- The type of the fault; such as flip one bit; flip one bit, but only in one direction (e.g. from 1 to 0); byte changed to a random (unknown) value; and so on.

There are two major kinds of fault side channels. The first ones are channels which are induced by computational faults occurring during cryptographic computation in an attacked module. These faults can be either random or intentional, caused, for instance, by a precise voltage manipulation Having the ability to introduce computational faults, this kind of attack can be used on almost every kind of cryptographic mechanism and it is regarded as one of the the most effective side channel attacks at all. The second kinds of fault side channels are those which are induced by sending an intentionally corrupted input data to the attacked module. For the module, this means a non-standard situation which must be handled in a special way. Usually the module has to use an error message to inform the user (the module can hardly know whether this is an ordinary user or an attacker) that the computation has been stopped due to some reasons.

Figure 4: Faulty attacks against a smart card

Fault analysis attacks were first considered in 1997 by Boneh , who described such attacks on the RSA signature scheme and the Fiat-Shamir and Schnorr identification protocols. Bao et al. presented fault analysis attacks on the ElGamal, Schnorr and DSA signature schemes.

Fault analysis attacks on elliptic curve public-key encryption schemes were presented by Biehl et al. Their attacks succeed if an error during the decryption process produces a point that is not on the valid elliptic curve. The attacks can be prevented by ensuring that points that are the Result of a cryptographic calculation indeed lie on the correct elliptic curve. Biham and Shami presented fault analysis attacks on the DES symmetric-key encryption scheme. Anderson and Kuhn discussed some realistic ways of inducing transient faults, which they call glitches.

Skorobogatov proposed a powerful yet surprisingly practical optical fault attack. They demonstrated that inexpensive equipment can be used to induce faults in a smart card by illuminating specific transistors; they also proposed countermeasures to these optical fault induction attacks. This attack can again convince the reader that fault injection is definitely a problem worth considering in the design and testing of a secure system or device.

More interstingly, Yen et al. hecking the correctness of the computed result before giving it to others may not be enough to prevent a hardware fault-based cryptanalysis.

Another countermeasure suggested to protect public key algorithms from some specific fault attacks is to check the integrity of the secret key at the end of signature computation. Other general tricks irrespective of concrete algorithms were also proposed, including checksums, execution randomization, ratification counters and baits, repeated refreshments [102].

SUMMARY:

To summarize, faulty attacks are real and big threats for any secure token (whatever the form factor) and must be taken into consideration at all steps of the product design and specification. Countermeasure and protection against fault attacks can be designed in both hardware and software. Devising and analyzing fault attacks are necessary as they permit us to estimate the strength of the countermeasures to be deployed.

5.1.3 Power Analysis Attack:

In addition to its running time and its faulty behaviour, the power consumption of a cryptographic device may provide much information about the operations that take place and the involved parameters. This is the very idea of power analysis attack. Certainly, power analysis attack is applicable only to hardware implementation of the cryptosystems. Power analysis attack is particularly effective and proven successful in attacking smart cards or other dedicated embedded systems storing the secret key.

Of all types of SCA attacks known today, the number of literatures on power analysis attacks and the relevant countermeasures is the biggest. Roughly calculating, there are at least more than 200 papers published currently in this area. Power analysis attack is actually the current research focus of side-channel attacks.

Power analysis attacks have been demonstrated to be very powerful attacks for most straightforward implementations of symmetric and public key ciphers [30, 31, 32, 33, 34]. For simplicity, we use Elliptic Curve Cryptosystems (ECC) to illustrate the power analysis attacks in this section. Yet, many of the relevant attacking methods and various countermeasures are applicable also to other cryptosystems.

Basically, power analysis attack can be divided into Simple and Differential Power Analysis (referred to as SPA and DPA, respectively). In SPA attacks, the aim is essentially to guess from the power trace which particular instruction is being executed at a certain time and what values the input and outputs have. Therefore, the adversary needs an exact knowledge of the implementation to mount such an attack. On the other hand, DPA attack does not need the knowledge of the implementation details and alternatively exploiting statistical methods in the analysis process. DPA is one of the most powerful SCA attacks, yet it can be mounted using very little resources.

More advanced differential power analysis looks at subtle statistical correlations between the secret bits and power consumption. DPA is a strong attack, but it only works in certain cases (e.g. Smartcards). In its classic instantiation, the adversary collects a large set {Ti, Ci} of trace-ciphertext pairs. The adversary also picks a selection function D that takes a ciphertext and a guess of part of the key and outputs one bit. The idea is that if the guess is right, this bit reflects something that actually shows up in the computation, but if the guess is wrong, then Dthe ciphertexts. will be random across The adversary then makes a guess K g and uses this guess and the selection function D to partition the set of traces into two sets: the one for which D(Ci , K g) = 0 and the other one for which D(Ci , K g) = 1 . He averages the traces in each set, and then looks at the difference between these average traces. If K g was wrong, these two sets are uncorrelated, and the differential trace becomes flat as the sample size increases. However, if K g was right, the differential approaches the correlation of D and power consumption, which will be spiky.

SPA and DPA attacks were introduced in 1999 by Kocher et al. [59]. They carried out a practical power analysis attack against an DES implementation in hardware. Coron [13] was the first to apply these attacks to elliptic curve cryptographic schemes, and proposed the SPA-resistant method for point multiplication, and the DPA-resistant method of randomizing projective coordinates. Oswald [159] showed how a multiplier k can be determined using the partial information gained about NAF(k) from a power trace of an execution of the binary NAF point multiplication method. Experimental results with power analysis attacks on smart cards were reported by Akkar et al. [160] and Messerges et al. [31], while those on a DSP processor core are reported by Gebotys et al. [161].

Chari et al. [75] Presented some general SPA and DPA countermeasures, and a formal methodology for evaluating their effectiveness. Proposals for hardware-based defenses against power analysis attacks include using an internal power source, randomizing the order in which instructions are executed (May et al. [162]), randomized register renaming (May et al. [83]), and using two capacitors, one of which is charged by an external power supply and the other supplies power to the device (Shamir [163]).

One effective method for guarding against SPA attacks on point multiplication is to employ elliptic curve addition formulas that can also be used for doubling. This approach was studied by Liardet et al. [15] for curves in Jacobi form, by Joye er al. [17] for curves in Hessian form, and by Brier and Joye [16] for curves in general Weierstrass form. Izu et al. [164] devised an active attack (not using power analysis) on the Brier-Joye formula that can reveal a few bits of the private key in elliptic curve schemes that use point multiplication with a fixed multiplier. Hasan [168] studied power analysis attacks on point multiplication for Koblitz curves and proposed some countermeasures which do not significantly degrade performance.

Another strategy for SPA resistance is to use point multiplication algorithms such as Coron's method [13] where the pattern of addition and double operations is independent of the multiplier. Other examples are Montgomery point multiplication (see Okeya et al.'s methods [18]), and the methods presented by Möller [22, 24], Hitchcock et al. [165], and Izu and Takagi [21]. The security and efficiency of (improved versions) of the Möller [22] and Izu-Takagi [21] methods were carefully analyzed by Izu et al. [166]. Another approach taken by Trichina et al. [167] and Gebotys et al. [161] is to devise formulas for the addition and double operations that have the same pattern of field operations (addition, subtraction, multiplication and squaring).

Joye et al. [14] proposed using a randomly chosen elliptic curve isomorphic to the given one, and a randomly chosen representation for the underlying fields, as countermeasures to DPA attacks. Goubin [25] showed that even if point multiplication is protected with an SPA-resistant method (such as Coron's method [13]) and a DPA-resistant method (such as randomized projective coordinates, randomized elliptic curve, or randomized field representation), the point multiplication may still be vulnerable to a DPA attack in situations where an attacker can select the base point (as is the case, for example, with ECIES). Goubin's observations highlight the difficulty in securing point multiplication against power analysis attacks.

The SPA simply observes several power consumptions of the device, and the DPA is additionally allowed to use a statistical tool in order to guess the secret information. An SPA-resistant scheme can be converted to be a DPA-resistant one by randomizing the parameters of the underlying system (See [13, 14], for example).

There are three different types of SPA-resistant schemes, available at present , for ECC scalar multiplication: (1) indistinguishable addition formula that uses one formula for both of elliptic addition and doubling [15,16,17]; (2) addition chain that always computes elliptic addition and doubling for each bit [13,16,18,20,21]; (3) window based addition chain with fixed pattern [19,22,23,24].

Defenses against differential power analysis are difficult, since they essentially only reduce the signal the adversary is reading, rather than eliminate it. Interestingly, an efficient randomization technique, using some random variables within the point addition operation, has also been proposed as a possible countermeasure against a DPA-style attack on the window-family algorithm in [29].

5.1.4 EM Attack:

As electrical devices, the components of a computer often generate electromagnetic radiation as part of their operation. An adversary that can observe these emanations and can understand their causal relationship to the underlying computation and data may be able to infer a surprising amount of information about this computation and data. This ability can be devastating, should the computer be a trusted computing platform intended to keep this information from the adversary.

Similar to the power analysis attacks, Electromagnetic Analysis (EMA) attacks can also be divided into two main categories: Simple Electromagnetic Analysis (SEMA) and Differential Electromagnetic Analysis (DEMA).

The potential of exploiting electromagnetic emanations has been known in military circles for a long time. For example, see the recently declassified TEMPEST document written by the National Security Agency [136] that investigates different compromising emanations including electromagnetic radiation, line conduction, and acoustic emissions. The unclassified literature on attack techniques and countermeasures is also extensive. For example, Kuhn et al. [138] discuss software-based techniques for launching and preventing attacks based on deducing the information on video screens from the electromagnetic radiations emitted. Experimental results on electromagnetic analysis attacks on cryptographic devices such as smart cards and comparisons to power analysis attacks were first presented by Quisquater et al. [137] and Gandolfi et al. [135]. The most comprehensive unclassified study on EMA attacks to date is the work of Agrawal et al. [133]. They showed that not only can EM emanations be used to attack cryptographic devices where the power side-channel is unavailable; they can even be used to break power analysis countermeasures.

Countermeasures against EM attacks on specific implementations fall into two broad categories: signal strength reduction and signal information reduction. Techniques for signal strength reduction include circuit redesign to reduce egregious unintentional emanations and the use of shielding and physically secured zones to reduce the strength of compromising signals.

6. CONCLUSION

The security of cryptographic modules for providing a practical degree of protection against the given white-box (total access) attacks should be examined in a totally un-trusted execution environment.

As Dr. Bruce Schneier already pointed out in 1998 that, "Strong cryptography is very powerful when it is done right, but it is not a panacea. Focusing on cryptographic algorithms while ignoring other aspects of security is like defending your home not by building a fence around it, but by putting an immense stake in the. Ground and hoping that your adversary runs right into it".

We have surveyed and observed side-channel attacks and the relevant countermeasures. A wide array of countermeasures against side channel attacks has been developed by researchers to provide the protections. We believe that a clear understand of attacks as well as the trade-offs associated with deploying countermeasures will enable a system architect to develop a truly secure system.

Finally, the most important conclusion from this paper is that it is not only a necessity but also a must, in the coming version, to evaluate cryptographic modules for their resistivity against SCA attacks.

ACKNOWLEDGEMENT

I Would to sincerely thank my H.O.D[E.C DEPARTMENT]MR.R.N .MUTAGI for their kind cooperation and support and all other faculty members for their faith and believing in me and support in my paper & research

REFERENCES

[1] J. Black, H. Urtubia. Side-channel attacks on symmetric encryption schemes: the case for authenticated encryption. Proc of 11th USENIX Security Symposium, pp.327-338, 2002.

[2] R. Anderson, M. Kuhn. Tamper resistance—a cautionary note. Proc of the 2nd USENIX Workshop on Electronic Commerce, pp.1-11, 1996.

[3] R. Anderson, M. Kuhn. Low cost attacks on tamper resistant devices. Proc of the 1997 Security Protocols Workshop, pp.125-136, LNCS 1361, 1997.

[4] W. Schindler. A Combined Timing and Power Attack. PKC 2002, LNCS 2274,pp.263-279,2002.

[5] D. Agrawal, J.R. Rao, P. Rohatgi. Multi-channel Attacs. CHES 2003, LNCS 2779,pp.2-16,2003.

[6] A. Shamir, E.Tramer. Acoustic cryptanalysis: on nosy people and noisy machines. Eurocrypt 2004 rump session, 2004.

[7] M. Kuhn. Optical Time-Domain Eavesdropping Risks of CRT Displays. Proc of the 2002

[8] Symposium on Security and Privacy, pp.3-18, 2002.

[9] 3GPP TS 35.205(V4.0.0). 3rd Generation Partnership Project;Technical Specification Group Services and System Aspects;3G Security;Specification of the MILENAGE Algorithm Set,April 2001.

[10] ETSI SAGE 3GPP AF Task Force. Report on the design and evaluation of 3GPP Authentication and Key Generation Functions.

[11] K. Okeya, K. Miyazaki, K. Sakurai. A Fast Scalar Multiplication Method with Randomized Projective Coordinates on a Montgomery-Form Elliptic Curve Secure against Side Channel Attacks. ICICS 2001, LNCS 2288, pp.428-439, 2002.

[12] C.H. Gebotys, R.J. Gebotys. A Framework for Security on NoC Technologies. Proc of the IEEE Computer Society Annual Symposium on VLSI (ISVLSI'03), 2003.

[13] K. Okeya, T. Takagi. A More Flexible Countermeasure against Side Channel Attacks Using Window Method. CHES'2003, LNCS 2779, pp.397-410, 2003.

[14] J.S. Coron. Resistance against Differential Power Analysis for Elliptic Curve Cryptosystems. CHES'99, LNCS 1717, pp.292-302, 1999.

[15] M. Joye,C. Tymen. Protections against differential analysis for elliptic curve cryptography: An algebraic approach. CHES'2001, LNCS 2162, pp.377-390, 2001.

[16] P.Y. Liardet, N.P Smart. Preventing SPA/DPA in ECC systems using the Jacobi form. CHES 2001, LNCS 2162, pp.391-401, 2001.

[17] E. Brier, M. Joye. Weierstrass Elliptic Curves and Side-Channel Attacks. PKC 2002, LNCS 2274, pp.335-345, 2002.

[18] M. Joye, J.J. Quisquater. Hessian elliptic curves and side-channel attacks. CHES 2001, LNCS 2162, pp.402-410, 2001.

[19] Okeya, K., Sakurai, K., Power Analysis Breaks Elliptic Curve Cryptosystems even secure against the Timing Attack. INDOCRYPT 2000, LNCS1977, pp.178-190, 2000.

[20] K. Okeya, T. Takagi. The Width-w NAF Method Provides Small Memory and Fast Elliptic Scalar Multiplications Secure against Side Channel Attacks. CT-RSA 2003, LNCS 2612, pp.328-342, 2003.

[21] W. Fischer, C. Giraud, E.W. Knudsen, J.P. Seifert. Parallel scalar multiplication on general elliptic curves .