Performance Analysis of Side Channel Attacks on Elliptic Curve Cryptosystem

Praful V. Barekar
Dr. Kavita R. Singh
Ms. Bhagyashree V. Ambulkar

Publication Partner: IJSRP INC.

Performance Analysis of Side Channel Attacks On Elliptic Curve Cryptosystem

Praful V. Barekar
Dr. Kavita R. Singh
Ms. Bhagyashree V. Ambulkar

Publishing Partner:
IJSRP Inc.
www.ijsrp.org

ISSN 2250-3153
9 772250 315302

Publication Partner:
International Journal of Scientific and Research Publications (ISSN: 2250-3153)

Preface

Commercial cryptographers have long been concerned with how much execution time their cryptographic implementations require.

The amount of time used to encrypt a message or produce a digital signature is often used as a benchmark when comparing different cryptographic schemes; with all other factors being equal, the fastest scheme is considered the most efficient and is hence the most marketable.

The amount of time it takes to compute a cryptographic function depends not only on what that function does but also what inputs are passed to it. Certain encodings of messages may require less time to encrypt because of the mathematical operations used. For example, an encryption function based on integer multiplication might be quick to evaluate with pen and paper if the message to encrypt is a power of ten.

A prudent cryptographer might then try to express every message as a power of ten to exploit this computational shortcut. However, in addition to messages, cryptographic functions often take secret keys as input and so the value of a key might influence publicly observably timing characteristics.

Publication Partner:

International Journal of Scientific and Research Publications (ISSN: 2250-3153)

Copyright and Trademarks

Publication Partner:

International Journal of Scientific and Research Publications (ISSN: 2250-3153)

Authors

Mr. Praful V. Barekar

Assistant Professor, Department of Computer Technology,
Yeshwantrao Chavan College of Engineering,
Nagpur

Dr. Kavita R. Singh

Associate Professor, Department of Computer
Technology, Yeshwantrao Chavan College of
Engineering,
Nagpur

Ms. Bhagyashree V. Ambulkar

Assistant Professor, Department of Computer Science,
G. H. Raisoni Institute of Engineering and Technology,
Nagpur

Publication Partner:
International Journal of Scientific and Research Publications (ISSN: 2250-3153)

Table of Content

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS

Sr. No.	Symbol	Symbol Description
1	CC	Common Criteria
2	DLP	Discrete Logarithmic Problem
3	DPA	Differential Power Analysis
4	DCA	Differential Collision Analysis
5	DEMA	Differential Electromagnetic Analysis
6	DFA	Differential Fault Analysis
7	DSCA	Differential Side Channel Analysis
8	DSS	Digital Signature Standard
9	EC	Elliptic Curve
10	ECC	Elliptic Curve Cryptosystem
11	ECDLP	Elliptic Curve Discrete Logarithm Problem
12	ECDSA	Elliptic Curve Digital Signature Algorithm
13	GF	Galois Fields
14	RSA	Rivest Shamir Adleman
15	SCA	Side Channel Analysis
16	ST	Security Target
17	SPA	Simple Power Analysis
18	SEMA	Simple Electromagnetic Analysis
19	SFA	Simple Fault Analysis
20	SSCA	Simple Side Channel Analysis
21	SSL	Secured Socket Layer

22	TA	Timing Analysis
23	TOE	Target of Evaluation
24	WTLS	Wireless Transport Layer Security

Publication Partner:

International Journal of Scientific and Research Publications (ISSN: 2250-3153)

LIST OF PUBLICATIONS

Sr. No.	Paper Title	Conference/Journal	Date
1	Survey of Side Channel Attacks on ECC	International Journal of Advanced Research in Computer Science and Software Engineering VOLUME 2, ISSUE NO 4 ISSN 2277 128X Page No. 361-365	04 May 2012
2	Performance Analysis of Timing Attack on Elliptic Curve Cryptosystem	International Journal of Computational Engineering Research VOLUME 2. ISSUE NO 3 ISSN: 2250–3005 Page No. 740-743	15 June 2012

Publication Partner:
International Journal of Scientific and Research Publications (ISSN: 2250-3153)

ABSTRACT

Side-channel analysis is a powerful technique. The principle consists in monitoring some side-channel information like the running time, the power consumption, or the electromagnetic radiation. From the monitored data, the attacker tries to deduce the inner-workings of the algorithm and thereby to retrieve some secret information. Smaller key size is the main advantage of elliptic curve cryptography, which is mostly used for public key infrastructure, but the side channel analysis is now well understood by the implementer. A different amount of running time is taken by cryptosystems depending on the input and the used key. To derive information about the secret key timing information can be used, which is named as timing attack on cryptosystems.To guess the private key Timing attacks attempt to exploit the variations in computational time for private key operations. This type of attack is primitive in the sense that no specialized equipment is needed. Simply measuring the computational time required by the user inputs and recording those user inputs an attacker can break a key.We are basically focusing on the Scalar Point Multiplication algorithm of Elliptic Curve Cryptography. Scalar point Multiplication consists of sequences of elementary field-operations of point addition and point doubling.The sequence of elementary field-operations that form the point-addition operation has a different running Time required than the sequence of elementary field operations that form the point-doubling operation. We are calculating the time required for sequences of point addition and point doubling, and on the basis of calculated time we are analyzing the secrete key which is called as timing attack.We are analysing the performance of Timing Attack on Elliptic Curve Cryptosystem.

Key Words: Cryptosystem, Side Channel Attack, Timing Attack, Running Time, Elliptic Curve Cryptography, Public key Infrastructure.

1. INTRODUCTION

1.1 Background

Mathematical abstraction 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 a scenario similar to the one described in Figure 1.1.

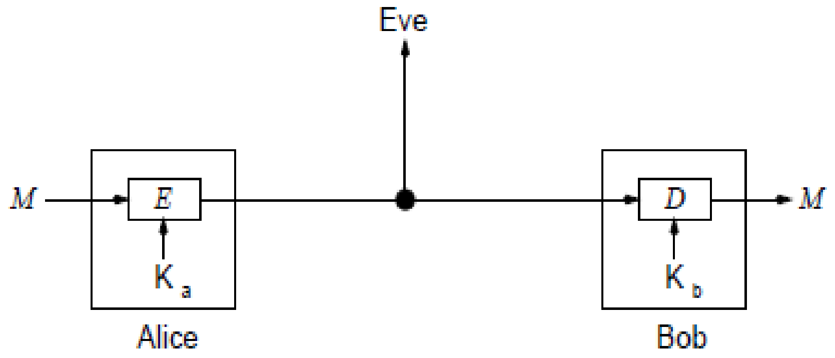


Figure 1.1: The Traditional Cryptographic Model

In this model, two people, Alice and Bob, attempt to use a cipher to engage in a private conversation across a public channel. An eavesdropper, Eve, monitors the public channel and tries to deduce what Alice and Bob are talking about. Eve has at her disposal all the details of the cipher, except for the secret key (this is known as Kirchhoff's assumption), a few plaintext-cipher text pairs generated by either Alice or Bob, as well some reasonable amount of computing power. Traditionally, any cipher which resisted Eve's scrutiny in this model was thought to be secure. Whether or not such a cipher would be implemented in the real world was then a matter of practicality (e.g., key length, encryption speed, memory requirements). However, as this thesis will illustrate, ciphers which are secure when specified as mathematical functions are not necessarily secure in real world implementations. In reality, ciphers are implemented on physical devices which interact with and are influenced by their environments. Electronic devices, like pagers and smartcards, consume power and emit radiation as they operate; they also react to temperature changes and electromagnetic fields. These physical interactions can be instigated and monitored by adversaries, like Eve, and may result in information which is useful in cryptanalysis.

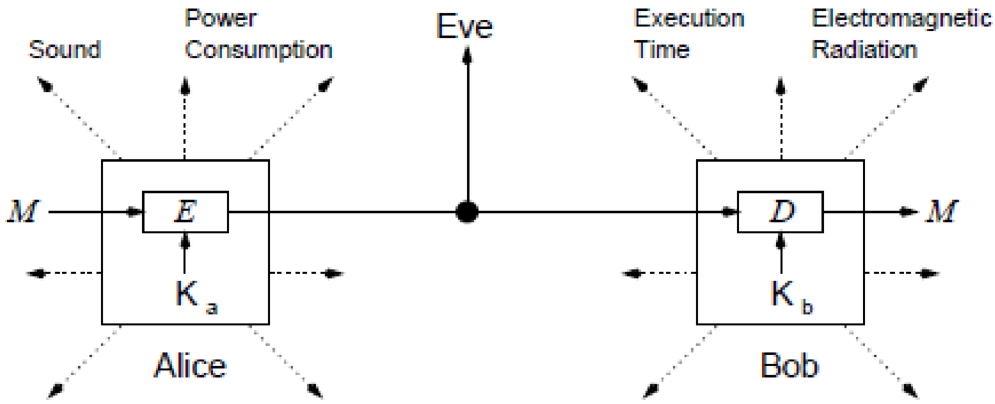


Figure 1.2: A Model Which Includes Side Channels

The traditional cryptographic model does not account for the physical side effects of using ciphers in the real world. A more realistic model can be described using the concept of a side channel, as shown in Figure 1.2. A side channel is source of information that is inherent to a physical implementation. The chapters of this thesis demonstrate how the analysis of side channel information can be used in cryptanalysis. In particular, one kind of side channel is examined: execution time. The academic research in this topic was initiated by Kocher [11].

1.2 Research Objective

Aside from standard algebraic attacks, which are not a threat with a good choice of parameters for the cryptosystem, side channel attacks are particularly effective for breaking the cryptography running on embedded devices. This powerful class of attacks exploits the information leakage through side channels such as timings or power consumption. Elliptic curve cryptosystems have the enormous advantage to consume only little memory and require few computing power. However, side channel attacks can break their common implementations. Finding countermeasures against side channel attacks on elliptic curve is a very active research theme. In this project, proposed system review side channel attacks on elliptic curves.The principle consists in monitoring some side-channel information like the running time, the power consumption, or the electromagnetic radiation. From the monitored data, the attacker tries to deduce the inner-workings of the algorithm and thereby to retrieve some secret information. The objective of the project is finding out the running time required for sequences of elementary field operations used in Scalar Point Multiplication algorithm of Elliptic Curve Cryptosystem. Timing Attack is use to analyze the secrete key on the basis of running time required for sequences of elementary field operations of point addition and point doubling. On the basis of Timing Attack we are analyzing the performance of Elliptic Curve Cryptosystem.

1.3 Thesis Organization

Chapter 2 explains the related work, which consists of basics about the Side Channel Attacks. The different leakage sources on the basis of which attackers try to monitor the internal working of the cryptosystem and countermeasures and Public Key Cryptosystem environment these topics are also covered in chapter 2. The basic and required information about ECC is explained in Chapter 3. This chapter covers the Point addition and Point Doubling operations of ECC and the elliptic curve on Prime field Fp operations. In chapter 4, basic concepts of Side Channel Attacks are described. Different types of Side Channel Attacks are also covered in this chapter. Proposed mechanism for the Timing Attack and the implementation of proposed Timing Attack mechanism is given in chapter 5. Chapter 6 includes the scenario of Implemented Elliptic Curve Cryptosystem and the experimental results are discussed. Finally chapter 7 concludes the work of this study and points out the future work.

2. LITERATURE REVIEW

2.1 Overview

Side channel attacks have been introduced at the abstract level, a powerful class of attacks which is particularly effective against embedded cryptography. The silent assumption of cryptography is that the secret key is stored in secure area. Recently, a new approach has been proposed to cut the Gordian knot: side channel attacks. The model of side channel attacks assumes that there exists some correlation between the secret key and side channel information such as timing or power consumption. By analyzing the information leakage through side channels, attackers may reveal the secret key. With limited investment, it is possible to deploy side channel analysis and break the implementation of most of the cryptosystems, unless specific precautions are taken.Provable security becomes more and more popular in the cryptographic community. It is now common to see it as an attribute of a cryptosystem. Provable security is at the protocol level, a harder task may be to evaluate the security of a cryptosystem at the implementation level. Rather than considering a cryptosystem as a black-box, it has been assumed that some sensitive data can leak during the course of the execution of a (naively implemented) crypto-algorithm.Proposed system is mainly focusing on the Timing Attack.

Timing Attacks were first introduced by Kocher [11]. Kocher describes the general idea of Timing Attack and shortly reports on some results for the RSAREF implementation of the RSA cryptosystem. The same idea can be used for discrete logarithm based cryptosystems like Diffie-Hellman, DSS, and other systems claims by Kocher. Later Kocher's timing attack on RSA was modified and practically examined, e.g. by Dhem et.al. on a RSA smart card implementation [8], and by Brumley and Boneh on the RSA implementation of the Open SSL library [9]. It has been observed that no successful practical results for timing attacks when applied to elliptic curve cryptosystems.

Elliptic curve (EC) cryptosystems have gained large support in recent years after several standard documents on public key cryptography included EC cryptosystems and EC signature schemes [15]. Their main advantage in comparison to RSA is significantly smaller key sizes for similar security levels.

2.2 Leakage Sources

The typical sources for side channel information are timings, power consumption and electromagnetic radiations. The assumption of timing attacks is that the duration of the execution of an algorithm depends on the secret key, and analyzing these durations provides some information on the secret key. Power attacks rely on measurements of the power consumption: the instructions of the CPU have characteristic power traces. Similarly, the electromagnetic radiation emitted by the CPU also depends on individual operations. The knowledge of the operation sequence may reveal the secret key itself.

2.3 Sources of Side Channel Attacks

Side channel attacks are not only a theoretical work of a circle of researchers, infact, an attacker without any special mathematical background and a simple oscilloscope may be able to break some cryptosystems if the designers did not take side channel attacks into account. Some situations are particularly sensitive.

2.3.1 Weak Hardware

The power source of embedded systems is sometimes provided by an extern source: this is the case for smartcards. Using relative cheap hardware such as a numeric oscilloscope and a computer, an attacker has all the necessary tools for side channel attacks. Smartcards do not have any protection to mask their power consumption: an attacker has only to plug wires at the right place to be able to measure and exploit it. In other words, even though the key itself is not directly accessible, it may be correlated with some physical information, which can be easily measured.

2.3.2 Weak Algorithms

Some cryptosystems are easy sources for side channel attacks. For example, elliptic curve cryptosystems are based on two operations, which usually present two totally different power traces. Therefore, it is realistic to expect attackers to be able to retrieve the operation sequence, which is often intimately connected to the secret key. Generally speaking, correlations with the secret key and the distinguish ability of individual operations in the side channel information make the task of the attacker easier.

2.4 Protection against Side Channel Attacks

There are many solutions to achieve resistance against SCA. But embedded systems such as smartcard must remain cheap while achieving sufficient security and the right solution has to be chosen while keeping in mind both of its cost and its security.

2.4.1 Hardware Protection

It is possible to mask side channel information thanks to special hardware. However, to embed such hardware protection, the industrialization processes have to be revised, and the whole line of products might become obsolete, which causes consequent investments for both of the clients and the constructor of the device. This is often unacceptable. Besides, some protections against power attacks may be ineffective against EM or timing attacks. Cumulating expensive protections becomes rapidly unpractical.

2.4.2 Algorithmic Protection

It is also possible to keep all existing technologies if an algorithmic solution is found. This is much cheaper and more flexible than modifying the hardware itself. Even if security is generally achieved at the expense of performances, the price to pay is usually very affordable if we consider that these countermeasures can offer performances that are almost equivalent to the fastest known algorithms.

2.5 Public key Cryptosystem

The first notion of symmetric key cryptography dates from thousands of years ago. Julius Caesar encrypted his secret documents by replacing each character by the character that is located three positions further in the alphabet. Although it is obvious that this encryption technique is not free of flaws, it can be used as an example to explain the basics of symmetric key cryptography. In Caesar’s scheme, encrypting a message means shifting each character over a certain number of positions in the alphabet. The decryption operation shifts each character over the same number of positions back in the alphabet. The secret key in this scheme is the number of positions over which the characters are shifted. In symmetric key cryptography, the encryption and decryption keys are equal or can be derived easily from each other. This is illustrated in Fig. 1.1, where Alice encrypts a plaintext m using an encryption function E and a key k, resulting incipher text c = Ek(m). Bob uses the same key for decrypting the cipher textin order to recover Alice's original message m = Dk(c). An eavesdropper, called Eve in Fig. 1.1, cannot recover the plaintext from the cipher text without knowing the secret key k. Eve is allowed, however, to have full knowledge of the encryption and decryption schemes E and D. This is known as Kirchhoff's principle: "A cryptosystem should be secure, even if an adversary knows everything about the system, except for the key".

3. OVERVIEW OF ELLIPTIC CURVE CRYPTOSYSTEM

3.1 Elliptic Curve Cryptosystems

All readers familiar with public key cryptography know the definition of the discrete logarithm problem in the multiplicative group of a finite field. An analog definition for the group of points on an elliptic curve as follows.

Definition: If E is an elliptic curve over Fq and B is a point of E, then the discrete log problem on E (to the base B) is the problem, given a point P ϵ E, of finding an integer x ϵ Z such that xB = P if such an integer x exists.

Until 1990, the only discrete log algorithms known for an elliptic curve cryptosystem were the ones that work in any group. These are exponential time algorithms, provided that the order of the group is divisible by a large prime factor. Menezes, Okamoto and Vanstone found a new approach to the discrete log problem on an elliptic curve. They used the Weil pairing to embed the group of points on E into the multiplicative group of some extension field Fqk. It is essential for the extension degree k to be small. The only elliptic curves for which k is small are the so-called “supersingular” curves.

3.2 ECC Basics

Based on the algebraic structure of elliptic curves over finite fieldsElliptic curve cryptography (ECC) is an approach to public-key cryptography .It represents a different way to do public-key cryptography. It is an alternative to the older RSA system—and also offers certain advantages. It was discovered in 1985 by Victor Miller and Neil Koblitz as an alternative mechanism for implementing public-key cryptography. Public-key algorithms create a mechanism for sharing keys among large numbers of participants or entities in a complex information system. Unlike other popular algorithms such as RSA, ECC is based on discrete logarithm that is much more difficult to challenge at equivalent key lengths.

The mathematical operations of ECC are defined over the elliptic curve:

$$\mathbf{y^2 = x^3 + ax + b}$$

where$\mathbf{4a^3 + 27b^2 \neq 0}$. Each value of the 'a' and 'b' gives a different elliptic curve. All points (x, y) which satisfies the above equation plus a point at infinity lies on the elliptic curve. the private key is a random number and the public key is a point in the curve. Multiplying the private key with the generator point G in the curve the public key is obtained. The domain parameters of ECC are generator point G, the curve parameters 'a' and 'b', together with few more constants. there are other parameters that must be agreed by both parties involved in secured and trusted communication using ECC, Apart from the curve parameters a and b. The domain parameters for Elliptic curve over Fp are **p**, **a**, **b**, **G**, **n** and **h**.

p is the prime number defined for finite field F_p . a and b are the parameters defining the curve y^2 mod p= x^3 + ax + b mod p. G is the generator point (x_G, y_G), a point on the elliptic curve chosen for cryptographic operations. n is the order of the elliptic curve. The scalar for point multiplication is chosen as a number between 0 and n – 1. h is the number of points on an elliptic curve.

3.3 Discrete Logarithm Problem

The security of ECC depends on the difficulty of Elliptic Curve Discrete Logarithm Problem. Let P and Q be two points on an elliptic curve such that kP = Q, where k is a scalar. Given P and Q, it is computationally infeasible to obtain k, if k is sufficiently large. k is the discrete logarithm of Q to the base P. Hence the main operation involved in ECC is point multiplication. i.e. multiplication of a scalar k with any point P on the curve to obtain another point Q on the curve.

Elliptic curve cryptosystems rely on the difficulty of solving the ECDLP. If an eavesdropper is able to solve the ECDLP then the eavesdropper will be able to break the system. Therefore, it is of great importance to understand the methods of tackling the ECDLP. For, the success of these methods can be use as a measure of the security of the system.

3.4 Scalar Point Multiplication

In Scalar point multiplication a point P on the elliptic curve is multiplied with a scalar k using elliptic curve equation to obtain another point Q on the same elliptic curve i.e. kP=Q.

Scalar Point multiplication is achieved by two basic elliptic curve operations

- Point addition, adding two points J and K to obtain another point L i.e. L= J + K.
- Point doubling, adding a point J to itself to obtain another point L i.e. L = 2J.

3.4.1 Point Addition

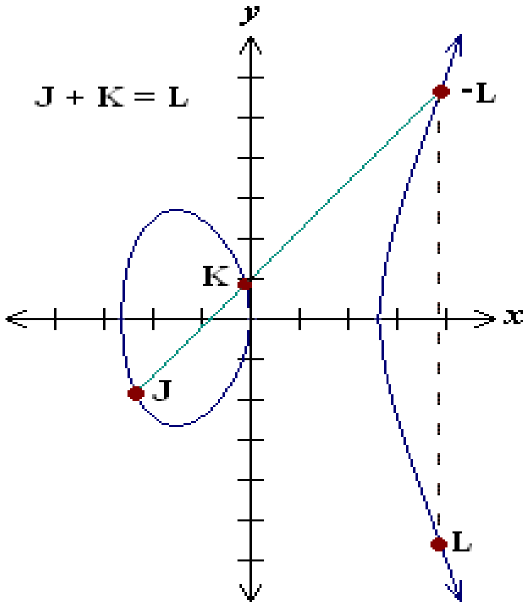


Figure 3.1: Point Addition

Addition of two points J and K on an elliptic curve to obtain another point L on the same elliptic curve is the Point addition.

Consider two points J and K on an elliptic curve as shown in figure 3.1. If K ≠ -J then aline drawn through the points J and K will intersect the elliptic curve at exactly one morepoint –L. The reflection of the point –L with respect to x-axis gives the point L, which is the result of addition of points J and K. Thus on an elliptic curve L = J + K. The point at infinity is the identity element.

Consider two distinct points J and K such that J = (x_J, y_J) and K = (x_K, y_K)

Let L = J + K where L = (x_L, y_L), then

$x_L = s^2 – x_J – x_K \bmod p$

$y_L = -y_J + s (x_J – x_L) \bmod p$

$s = (y_J – y_K)/(x_J – x_K) \bmod p$, s is the slope of the line through J and K.

If K = -J i.e. K = (x_J, $-y_J$ mod p) then J + K = O. where O is the point at infinity.

If K = J then J + K = 2J then point doubling equations are used.

Also J + K = K + J.

3.4.2 Point Doubling

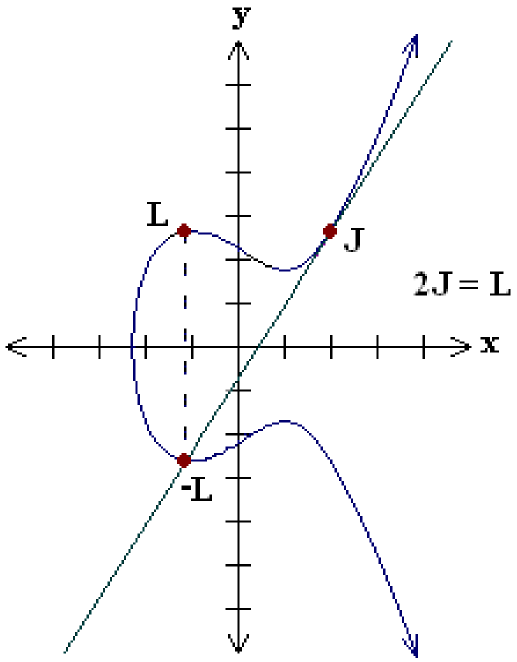


Figure 3.2: Point Doubling

Addition of a point J on the elliptic curve to itself to obtain another point L on the same elliptic curve is the Point doubling.

To double a point J to get L, i.e. to find L = 2J, consider a point J on an elliptic curve as shown in figure 4.2. If y coordinate of the point J is not zero then the tangent line at J will intersect the elliptic curve at exactly one more point –L. The reflection of the point –L with respect to x-axis gives the point L, which is the result of doubling the point J. Thus L = 2J.

Consider a point J such that J = (x_J, y_J), where $y_J \neq 0$

Let L = 2J where L = (x_L, y_L), Then

$x_L = s2 - 2x_J \bmod p$

$y_L = -y_J + s\,(x_J - x_L) \bmod p$

$s = (3x_J^2 + a) / (2y_J) \bmod p$, s is the tangent at point J and a is one of the parameters chosen with the elliptic curve

If $y_J = 0$ then $2J = O$, where O is the point at infinity.

3.5 Elliptic Curve on Prime Field F_p

An Elliptic Curve over F_p, where p is a prime number, can be defined as the points (x, y) satisfying the elliptic curve equation: $y^2 \bmod p= x^3 + ax + b \bmod p$, where $4a^3 + 27b^2 \bmod p \neq 0$. Here the elements of the finite field are integers between 0 & p–1. The prime number p is chosen such that there is finitely large number of points on the elliptic curve to make the cryptosystem secure.

The graph for this elliptic curve equation is not a smooth curve. Hence the geometrical explanation of point addition and doubling as in real numbers will not work here. However, the algebraic rules for point addition and point doubling can be adapted for elliptic curves over Fp.

4. SIDE CHANNEL ATTACKS

4.1 Terms

Attacks on cryptographic devices differ from attacks on cryptographic algorithms and protocols. They benefit from the fact that not only mathematical analysis or design analysis (and their application in an attack on the TOE via the designed user or developer interfaces) is possible but also physical analysis of the device that actually executes operations using secret data. This opens the field of passive and active attacks using the intended interface of the device, as well as invasive, semi-invasive and non-invasive attacks that open and/or exploit additional, non-intended channels which, in consequence, may leak information on the secrets such as private key material. Side-Channel cryptanalysis uses physical observables resulting from internal states and processes of a cryptographic computation as additional information source for cryptanalysis. The term 'physical observable' is understood to be any property of a physical system state determined by a physical operation. The outcomes of the measurement of physical observables are real-valued vectors. In other words, the internal state changes of the physical device (including the state changes caused by operations with secret keys) cause instantaneous leakage that can be exploited. Such side-channel information may arise from observations of

- Varying execution time of operations (on all granularity levels from clock cycles, over single CPU instructions to whole algorithm executions),
- Varying power consumption during operation,
- Varying electro-magnetic emanation during operation,
- Enforced unexpected behavior (not designed system state changes, that cause leakage on the designed TOE interface or on side-channels) in consequence of transient or permanent device faults, e.g. induced by irradiation, and more.

These typical examples should be considered and complemented, if necessary, by the evaluator. Other side-channels or combinations of it might be relevant depending on the Security Target (ST) and the TOE.

The observables of the attack are real-valued vectors, that are recorded and stored in such a way that they are available as input for mathematical (especially statistical) analysis. Side-channel cryptanalysis typically has the aim to recover secret (private) cryptographic keys, or plaintext, or dependent on the scheme, also other data. In order to compromise cryptographic keys, side-channel cryptanalysis can also be used to learn blinding values, intermediate results or parts of the cryptographic key. The complete key is then recovered by brute force or mathematical cryptanalysis. Further, it might be used in an intermediate attack step to learn secret values used in countermeasures such as blinding or masking. Other scenarios, such as reverse engineering of unknown devices with the goal to learn critical design information from a black box are used in practice, but for the purpose of a CC evaluation of an ECC TOE, they are not relevant, because this knowledge has to be passed to the evaluator by the manufacturer and is assumed to be known a-prior to the evaluator.

An adversary is successful if side-channel enhanced cryptanalysis leads to a critical entropy loss of a secret cryptographic key. In the context of CC, it is sufficient to calculate the attack potential required by an attacker to perform an attack. The underlying objective is to aid in expressing the total effort required to mount a successful attack.

The side-channel analysis methods include, but are not limited to,

- Simple Power analysis (SPA),
- Simple Electromagnetic Analysis (SEMA),
- Timing Analysis (TA),
- Differential Power Analysis (DPA),
- Differential Electromagnetic Analysis (DEMA),
- Profiling Analysis,
- Differential Collision Analysis (DCA),
- Higher Order Differential Side-Channel Analysis, e.g. second order DPA,
- Multivariate Analysis (Template Attacks),
- Simple Fault Analysis (SFA),
- Differential Fault Analysis (DFA),
- Stochastic Approach.

SPA attacks directly interpret power consumption traces, that are usually preprocessed (e.g. by filtering) to make inner parts of the algorithm execution visible. In this context average traces or traces are also treated as the standard deviation / variance of the collected power consumption traces.

SEMA attacks work accordingly with another side-channel, the electro-magnetic emanation during the execution of the cryptographic algorithm. SPA and SEMA are sometimes called SSCA (Simple Side-Channel Analysis).

Timing attacks exploit differences in the time needed for the execution of data-dependent steps of the cryptographic algorithm implementation in order to recover an unknown secret parameter, e.g. the cryptographic key used. The measurement of the time can focus on the overall runtime, e.g. between message input and reception of a result or on local steps that are identified by means of another side-channel such as power traces in SPA.

DPA attacks exploit the data dependency of the power consumption of cryptographic devices. They use a large number of (possibly preprocessed) power traces to analyse the power consumption at a fixed point in time as a function of the processed data. They can be organized in five steps:

- Choosing an intermediate value of the executed cryptographic algorithm, that depends on the secret to be attacked
- Measuring the power consumption
- Calculating hypothetical intermediate values
- Mapping hypothetical intermediate values to hypothetical power consumption values
- Comparing the hypothetical power consumption values of each key hypothesis with the recorded power traces by computing the dependency traces (e.g. correlation traces) between them.

DEMA attacks work accordingly with the electro-magnetic emanation during the execution of the cryptographic algorithm. DPA and DEMA are sometimes called DSCA (Differential Side-Channel Analysis).

Profiling Analysis works similar to DPA with the same amount of data. In contrast to DPA, it does not set up hypotheses on secret values but uses known input, output or intermediate values instead. In this way an attacker can, e.g., identify the time at which the value is loaded or computed and hence narrow the time interval for critical executions with sensitive data, and eventually observe hardware characteristics such as register widths.

Template attacks can be performed, e.g., as SPA or as DPA. They consist of a profiling stage with identification of the so called points of interest and a key recovery stage. At the profiling stage, for each key-dependency, a template, i.e. a multivariate characterization of the key-dependent leakage signal, is produced that is later used in the key recovery stage.

Fault analysis (fault channel cryptanalysis) uses physical means in order to modify internal states of the TOE, aiming at obtaining additional information for cryptanalysis by observation of faults and their differences to normal behavior. A fault is defined as an abnormal condition or defect of a component which may lead to a transient or permanent malfunction. Fault analysis can be applied in a simple or differential approach (SFA / DFA).

The Stochastic Approach can be seen as a further development of the template attack. It consists of a profiling step and a key-recovery stage and operates on a small key subspace. But in contrast to the template attack, it uses a specific approximation for the estimation of the part of the side-channel signal (random variable) that only depends on the sub key and the input message.

Side-channel analysis methods are called non-invasive if the cryptographic boundary of the TOE is left intact. This is typically the case for TA, SPA/DPA, SEMA/DEMA, DCA, stochastic approach and template attacks. Even SFA and DFA can be considered non-invasive if power glitches are applied to the TOE, but typically at least a semi-invasive approach is necessary for fault analysis, where e.g. a chip is de-packaged, but the chip surface is not harmed.

Invasive attacks physically access components such as buses, memory and processor registers of the TOE, violating the cryptographic boundary.

4.2 Types of Side Channel Attacks

Through side-channel information like the running time, the power consumption, or the electromagnetic radiation attackers try to analyse some secret information, on the basis of secret information analysed by the attacker there are three types of Side Channel Attacks as follows.

4.2.1 Timing Attack

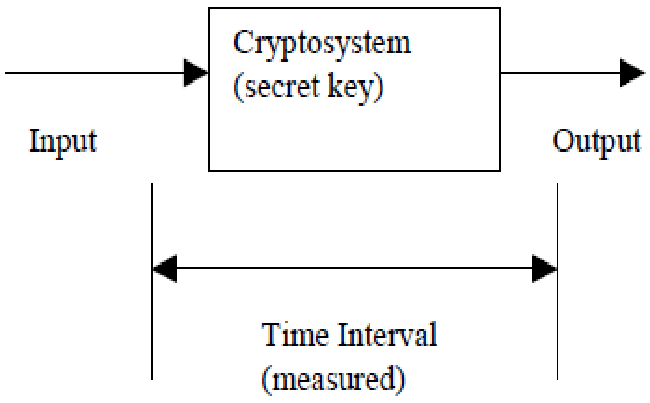


Figure 4.1: Timing attack

Kocher presented the timing-attacks: Attackers carefully measure the amount of time required to perform the private key operations. This attack could be applicable to the elliptic curve cryptosystems including ECDSA. Time required to perform the conventional scalar multiplication algorithm based on the Weierstrass-form depends on the bit-patterns (and on the ratio between the number of zeros and the number of ones) of the secret value. Whereas the scalar multiplication on the Montgomery-form elliptic curve does not depend on the bit-patterns (nor on the ratio between the number of zeros and the number of ones) of the secret value. It has exactly seven multiplications and four square-multiplications on **F***p*per bit. This is due to the specific algorithm for computing scalar multiplication *nP*from *P*, which repeatedly calculates either $(2mP, (2m + 1)P)$ or $((2m + 1)P, (2m + 2)P)$ from $(mP, (m + 1)P)$ in the Montgomery-form elliptic curves. The computation via by choosing a representive in the projective coordinates randomly is also useful for making it more difficult to measure the amount of time required.

Compute the scalar *d* multiplications on the a. coordinates (*x, y*) via a corresponding projective coordinates (*kx, ky, k*), where *k* is randomly choose. Thus, Montgomery-form elliptic curves are shown to be useful for public-key cryptosystems from the point of view of not only efficient implementation but also protection against timing-attacks.

4.2.2 Power Attack

Power attacks arise from the actual implementation of the cryptosystems, which differ from the ideal cryptosystems. There are leakages of information in addition to input and output data while the cryptographic devices (e.g. smart card) execute cryptographic transactions (e.g. signature, encryption, decryption). An attacker may use the leakages for his estimate. In 1996, Kocher proposed timing attack. Timing attack is one of the power attacks in which an attacker uses timing of execution for estimate of the secret key. Recently, Kocher et al. proposed DPA (Differential Power Analysis) and SPA (Simple Power Analysis) DPA is a power attack in which an attacker uses power consumption and analyzes such data statistically, and SPA is an attack without statistical analysis. Coron generalized DPA to elliptic curve cryptosystems with the following SPA-immune scalar multiplication algorithm. For the purpose of constructing cryptosystems with immunity to DPA, characteristics of DPA are described, how an attacker estimates the secret key in the attack. The point of this attack is “a difference between executing procedures (non-symmetry)” and “an appearance of a predicted special value”. First, the executing procedure of typical cryptographic transaction depends on the secret key. Consequently, the executing procedure of cryptographic transaction differs from secret key to secret key. If an attacker finds the difference of the executing procedure from leakages, attackeris able to derive the information on the secret key. Actually, since it is hard to find the difference of executing procedure as it is, attacker treats it statistically and makes its bias big, and finally attacker finds the difference of the executing procedure. Next, if an appearance of some specific value on the cryptographic transaction depends on the secret key, an attacker is able to detect the secret key by whether the value appears on the execution or not.

4.2.2.1 Simple Power Analysis

SPA is executed on one single power trace, where the attackers identify the power signature of special operations which are related to some data of the secret key. Typically, to find conditional branches depending on the secret key in the implementation of the cryptosystem it can be expected, such as:

if condition on the secret key then do operation A else do operation B

If it is possible to distinguish the power signatures of operations A and B, the condition on the secret key can be retrieved.

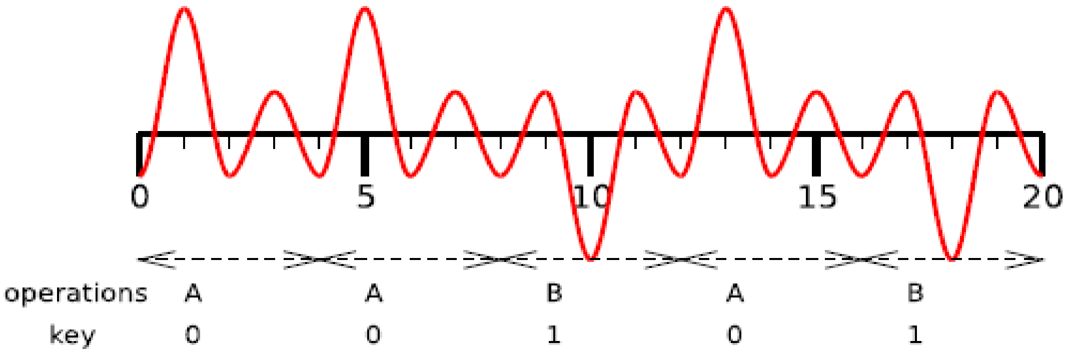


Figure 4.2: Simple power Analysis

4.2.2.2 Differential Power Analysis

Sometimes the dependence on the secret key induces tiny differences in the power trace, and these differences are embedded in the noise. To extract such information, attackers may use the averaging technique, where the noise level is decreased by averaging the power traces with different inputs to the cryptosystem. The principle of DPA is to guess the value of some bit of the secret key and verify the validity of the assumption with the collected power traces. More precisely, given the assumption on the secret key and the input of the cryptosystem, the corresponding power traces are classified into two classes C0 and C1 in which a specific attribute remains the same. For example, there is a special spike in every power trace of C0 and C1. Inside a class, the spike is always the same, but it is different in the two classes. Because the characteristic attribute of the classes is usually too small to be detected within one power trace, a function for explicitly computing the class of a the power trace is needed:

F: (input, bit of the secret key) → {class 0, class 1}

By computing the difference of the averaged power trace in each class, the constant common part of the power trace tends to vanish and only the attribute, which is different in each class, remains in the form of spikes. However, if the assumption on the bit of the secret key was not correct, the classification is random and the average power traces are the same in the two classes. Therefore, if spikes are visible in the differential signal, the assumption on the secret key is correct, and if no spikes are visible, the assumption is wrong.

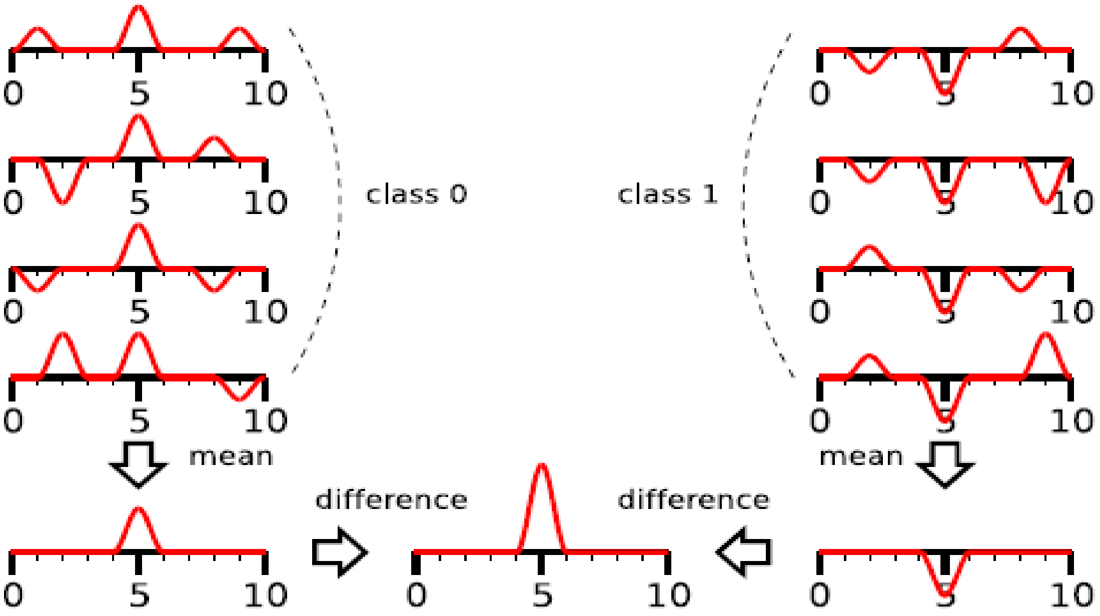


Figure 4.3: Differential Power Analysis

4.2.3 Electromagnetic Attack

Nowadays, CMOS is by far the most commonly used technology to implement digital integrated circuits. A CMOSgate consists of a pull-up network with p-MOS transistors and a pull-down network with n-MOS transistors. Those networks are complementary: when the input is stable, only one of the two networks conducts. The most simple logic gate is an inverter; its power consumption is representative for all logic ports and gives a general image of the power consumption in a CMOS circuit. During the functioning of the inverter, 3 types of power consumption can be distinguished. The leakage current, the current that flows from the power source to the ground during the switching from 0 to 1 (short-circuit current) and the current used to charge and discharge the different capacitors in a digital network (dynamic power consumption). The last one causes the biggest power consumption in present designs. Important to note is that these capacitors are necessary to maintain the two different logic levels. In addition, all capacitors for each gate differ, which results in a different power consumption of the different gates according to the data being processed. The sudden current pulse that occurs during the transition of the output of a CMOS gate causes a variation of the electromagnetic field surrounding the chip; this can be monitored for example by inductive probes which are particularly sensitive to the related impulsion.

Two types of electromagnetic analysis attacks are distinguished. In a simple electromagnetic analysis (SEMA) attack, an attacker uses the information from one electromagnetic radiation measurement directly to determine (parts of) the secret key. In a differential electromagnetic analysis(DEMA) attack, many measurements are used in order to filter out noise and the key is derived using a statistical analysis. While SEMA exploits the relationship between the executed operations and the electromagnetic radiation, DEMA exploits the relationship between the processed data and the electromagnetic radiation. A SEMA attack is typically used when there is a conditional branch in the algorithm, which results in a different radiation pattern whenever the branch is taken. A DEMA attack uses the property that processing different data needs a distinct amount of power and radiates a different field.

5. DESIGN AND IMPLEMENTATIONOF PROPOSED TIMING ATTACK MECHANISM

5.1 Proposed Timing Attack Mechanism

Depending on the input and the used key, cryptosystems often take slightly different amounts of running time. To derive information about the secret key this timing information can be used. Timing Attacks is a new class of attacks on implementations of cryptosystems. Timing attacks attempt to exploit the variations in computational time for private key operations to guess the private key. This type of attack is primitive in the sense that no specialized equipment is needed. An attacker can break a key by simply measuring the computational time required by the user inputs and recording those user inputs. The focus is basically on the Scalar Point Multiplication algorithm of Elliptic Curve Cryptography. Scalar point Multiplication consists of sequences of elementary field-operations of point addition and point doubling. The sequence of elementary field-operations that form the point-addition operation has a different running Time required than the sequence of elementary field operations that form the point-doubling operation. The time required for sequences of point addition and point doubling is calculated, and on the basis of calculated time the secrete key i. e. decryption key is analysed.

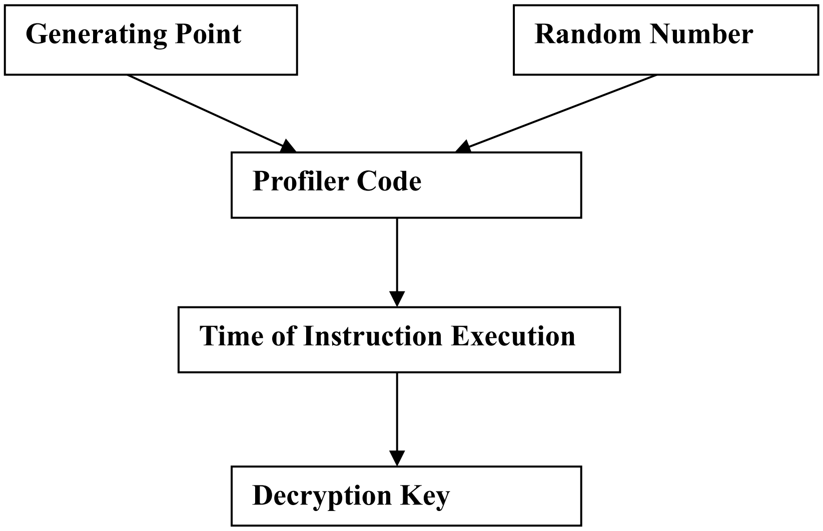


Figure 5.1: Proposed Timing Attack Mechanism

Figure 5.1 shows the flow of proposed timing attack mechanism. Profiler code analyses the time required to execute the set of instructions written under it. Profiler evaluate the running time of set of instructions written for Addition and Doubling operations of elliptic curve cryptosystem for calculating the public keys of the user involved in communication, as the running time required for Addition operation is different than the running time required for Doubling operation of the Elliptic curve cryptosystem. On the basis of these running time required for different size of prime number, we calculate the Probable Secrete key.

5.2 Scalar Point Multiplication Algorithm

In Scalar point multiplication a point P on the elliptic curve is multiplied with a scalar k using elliptic curve equation to obtain another point Q on the same elliptic curve i.e. kP=Q.

Algorithm 1: (Scalar Point Multiplication)

```
Input: Point P, integer k > 0 with binary
representation k = (1 kw-2... k0)2.
Output: Point k · P
1. Let Q = P
2. for j = w - 2 down to 0 do
    Q = 2 · Q
    ifkj = 1 then Q = Q + P
  done
3. return Q
```

The **Elliptic Curve Discrete Logarithm Problem** (ECDLP) is defined as the problem computing the integer k only with knowledge of the elliptic curve E and the two points P and Q = K·P∈E(GF(p)) without prior knowledge of k. ECDLP is generally considered to be a very difficult problem if the field characteristic p is sufficiently large (at least 160 bit), and the order of the group of points on the elliptic curve is prime (or "almost prime"). Up to now, there exists no algorithm that can solve ECDLP for such curves in reasonable time. Therefore point multiplication was chosen as the "trapdoor one-way function" used in elliptic curve public key cryptosystems (ECC) and elliptic curve signature schemes (ECDSA). Timing attack use the following scenario: Assume that the unknown secret key k is stored securely and cannot be accessed directly. Compute k · Pi for arbitrarily many random points Pi. Moreover it is posible to determine the running time Ti of each of these computations (in our simulation, we uses Algorithm 1 for fast multiplication). Running times for a single point addition or point doubling for arbitrarily chosen points might also be determined.

5.3 The Attacker's Task

The attacker has the ability to observe a sequence of elliptic curve operations, thus, the attacker's aim is to calculate and exploit the probabilities of certain sequences of bits given an observed sequence of elliptic curve operations. Using the information of such conditional probabilities, the key-space that has to be searched to find the correct ephemeral key, can be significantly reduced. This is because certain combinations of patterns in the power trace and certain combination of digits are less likely than the others (or even not possible at all). The attacker's task can be stated in a more formal way. Let X be a random variable that denotes a sequence of elliptic-curve operations and $|X|$ the length of X (i.e. the number of elliptic-curve operations in this sequence). For example, X="DDD" (i.e. the realization of the random variable X consists of three consecutive elliptic-curve point-double operations) thus $|X| = 3$, or X="DAD" (i.e. the realization of the random variable X consists of an elliptic-curve point-double operation, an elliptic-curve point-addition operation and an elliptic-curve point-double operation) thus $|X| = 3$. Let Y be a random variable that denotes a sequence of digits in the digit representation of k and $|Y|$ the length of Y (i.e. the number of digits). For example Y = "000" (i.e. the realization of the random variable Y consists of three consecutive zeros) thus $|Y| = 3$, or Y = "01" (i.e. the realization of the random variable Y consists of a zero and a one digit) thus $|Y| = 2$. Then the attacker's goal is to calculate and exploit the conditional probability.

5.4 Mathematical Model

Let us denote a set of inputs (plaintexts) to the system by $S_M = \{M_1,M_2,M_3,\ldots\ldots M_n\}$ All the possible keys compose the key set denoted by $S_K = \{K_1,K_2,K_3,\ldots\ldots\ldots\ldots K_d\}$ where d is the number of possible keys. If the cryptosystem implementation we want to attack is vulnerable to timing attacks, the timing distribution of the input will be dependent on the key used in the system. Thus for key K_i , we will have a timing distribution donated by) $P_i(t)= F(S_M , K_i)$ which is different from that of other keys.

For the system to attack, the timing information is measured for a set of input values from the set S_M, and form a timing distribution P(t). The attack to the system will be reduced to a usual detection problem which tries to detect K_i knowing P_i (t) and P(t). regular detection solutions can be applied to solve the problem at least in theory. For example, the detection problem has a general form of the solution: if T (P(t), K_i) > Threshold (K_i , S_M) , K_i is detected. As long as the proper transform function T() and the threshold functions are found, the system is broken.

5.5 Timing Attack applied to Elliptic Curve

Timing attack is based on the following idea. Denote by kj the j-th bit of the secret key k. the following equation for the total running time is got in algorithm1 with input point Pi :

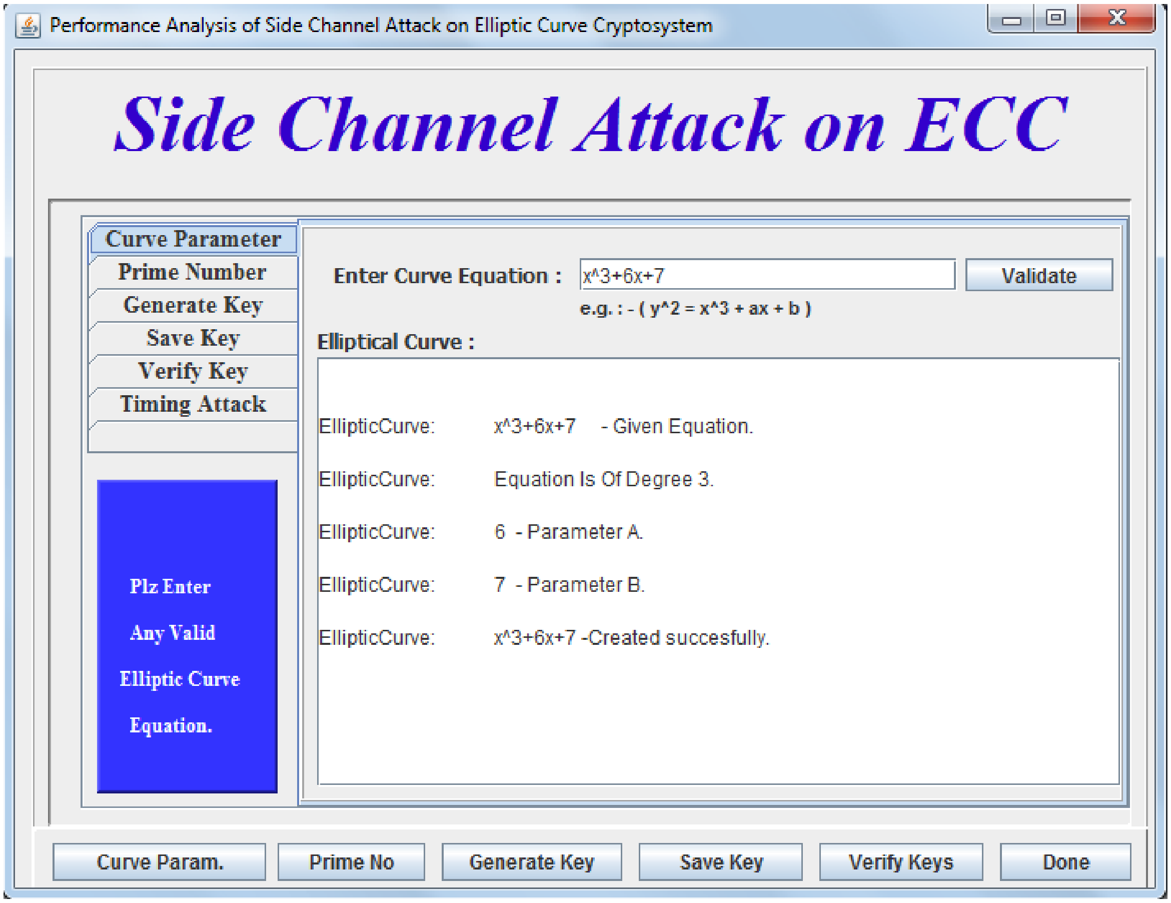
$$T_i = e_i + \sum_{j=0}^{w-1} (D_{i,j} + k_j A_{i,j})$$

In this formula, Di,j denotes the time needed for a point doubling operation for bit j, and Ai,j denotes the time for an addition operation for bit j, ei is some "noise" (run time for looping, if-operation, and other external influences). Note that both the doubling time Di,j and the addition time Ai,j depend on the chosen random point Pi (index i) and the iteration index j. Let 0 < r < w, and assume that we already know the "upper" bits kw-1 , ... , kr+1 of the binary representation of the secret key k. The fundamental idea is the fact that the values of Di,r and Ai,r can be determine with this information. For every sample point Pi the known bits of k are use to determine the value of the point Q at the beginning of iteration j = r in Algorithm 1. Once these points are known, both Di,r and Ai,r can be determine using the decryption device (It is assumed that the run time for a single point operation can be determine).

6. RESULTS AND DISCUSSION

6.1Scenario Description

Scenario consists of two users A and B try to communicate under public key cryptosystem environment secured through Elliptic Curve Cryptosystem mechanism. Different snapshots will describe the overview of scenario.



Enter Curve Equation : x^3+6x+7

e.g. : - (y^2 = x^3 + ax + b)

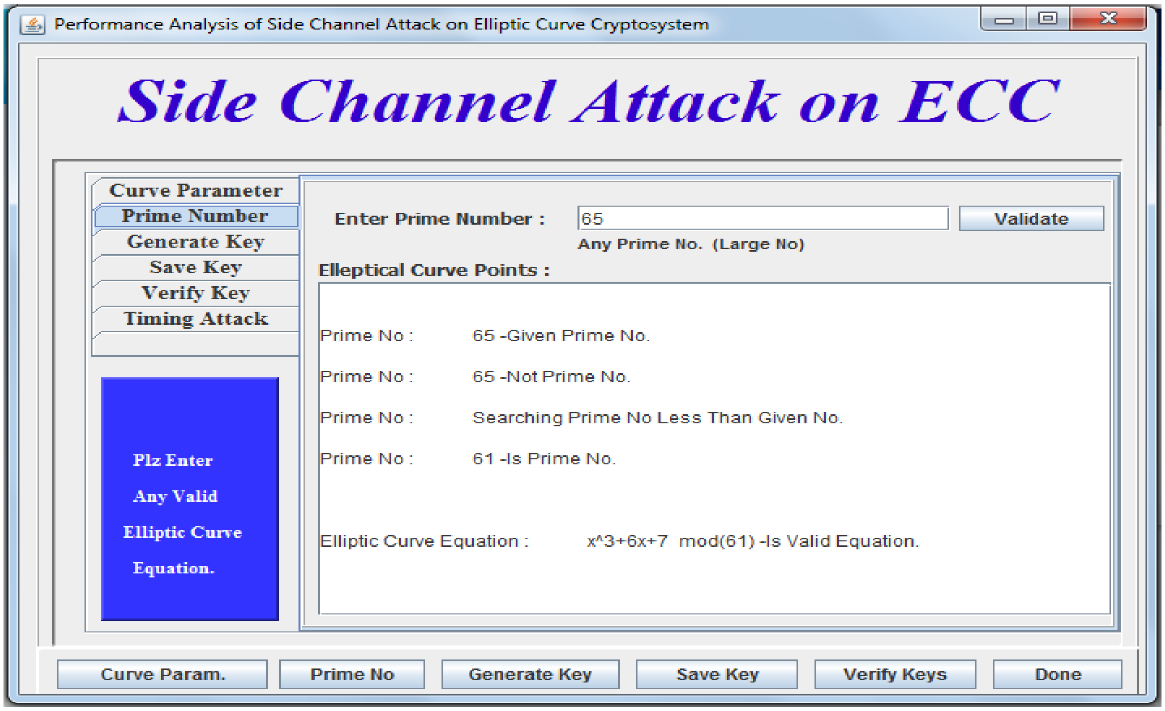
Elliptical Curve :

```
EllipticCurve:        x^3+6x+7     - Given Equation.

EllipticCurve:        Equation Is Of Degree 3.

EllipticCurve:        6  - Parameter A.

EllipticCurve:        7  - Parameter B.

EllipticCurve:        x^3+6x+7 -Created succesfully.
```

Plz Enter
Any Valid
Elliptic Curve
Equation.

Figure 6.1: Curve Validation

As shown in Figure 6.1, the validity of the Elliptic curve is being checked. Whether the entered curve equation is of degree three or not, under the condition $\mathbf{4a^3 + 27b^2 \neq 0}$.Calculate the parameter A and B.



Enter Prime Number : 65

Any Prime No. (Large No)

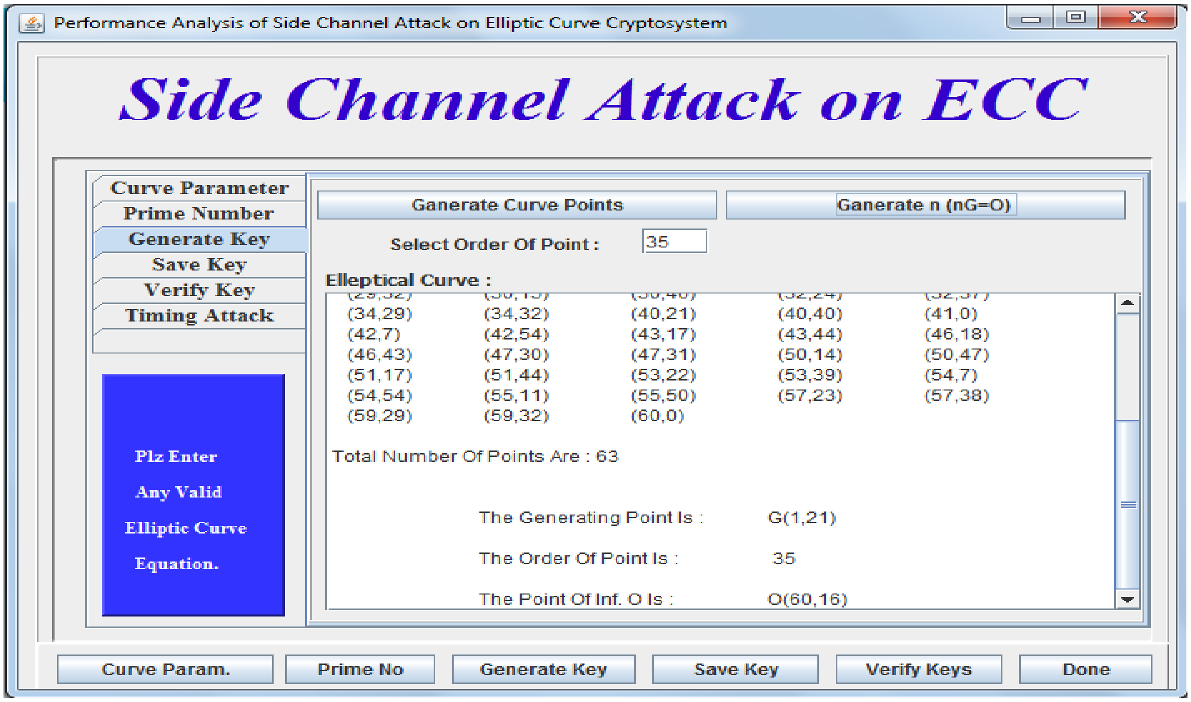
Plz Enter Any Valid Elliptic Curve Equation.

Elleptical Curve Points :

```
Prime No :        65 -Given Prime No.
Prime No :        65 -Not Prime No.
Prime No :        Searching Prime No Less Than Given No.
Prime No :        61 -Is Prime No.

Elliptic Curve Equation :       x^3+6x+7  mod(61) -Is Valid Equation.
```

Figure 6.2: Prime Number Validation

As shown in the figure 6.2 the validity of prime number is being checked, whether entered number is prime or not, otherwise calculate the valid prime number just smaller than the entered number by user.



Plz Enter Any Valid Elliptic Curve Equation.

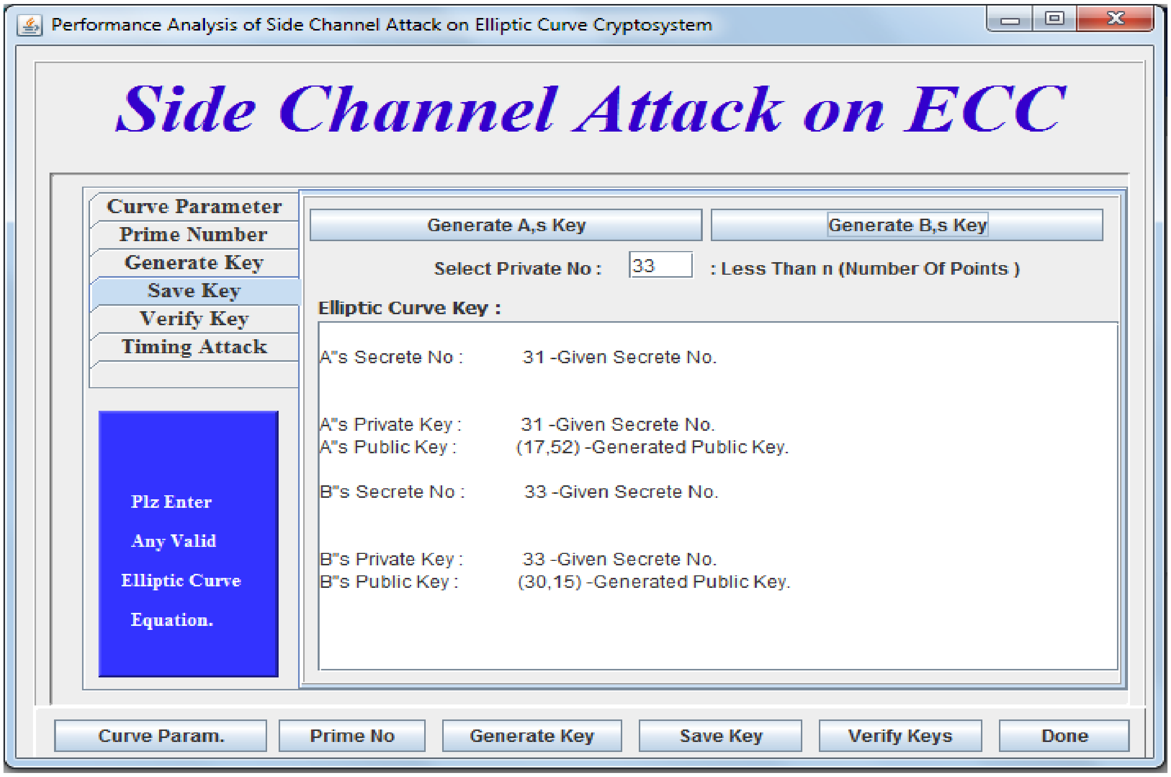
Select Order Of Point : 35

Elleptical Curve :

```
(29,32)        (30,13)        (30,40)        (32,24)        (32,37)
(34,29)        (34,32)        (40,21)        (40,40)        (41,0)
(42,7)         (42,54)        (43,17)        (43,44)        (46,18)
(46,43)        (47,30)        (47,31)        (50,14)        (50,47)
(51,17)        (51,44)        (53,22)        (53,39)        (54,7)
(54,54)        (55,11)        (55,50)        (57,23)        (57,38)
(59,29)        (59,32)        (60,0)

Total Number Of Points Are : 63


               The Generating Point Is :        G(1,21)

               The Order Of Point Is :           35

               The Point Of Inf. O Is :         O(60,16)
```

Figure 6.3: Curve point generation and select the Order of Point for calculating the Generating Point and Point of Infinity.

Figure 6.3 shows the total number of curve points generated on the Elliptic Curve. The order of point is being sated for calculating the Generating point and Point of Infinity.



Performance Analysis of Side Channel Attack on Elliptic Curve Cryptosystem

Side Channel Attack on ECC

Select Private No : 33 : Less Than n (Number Of Points)

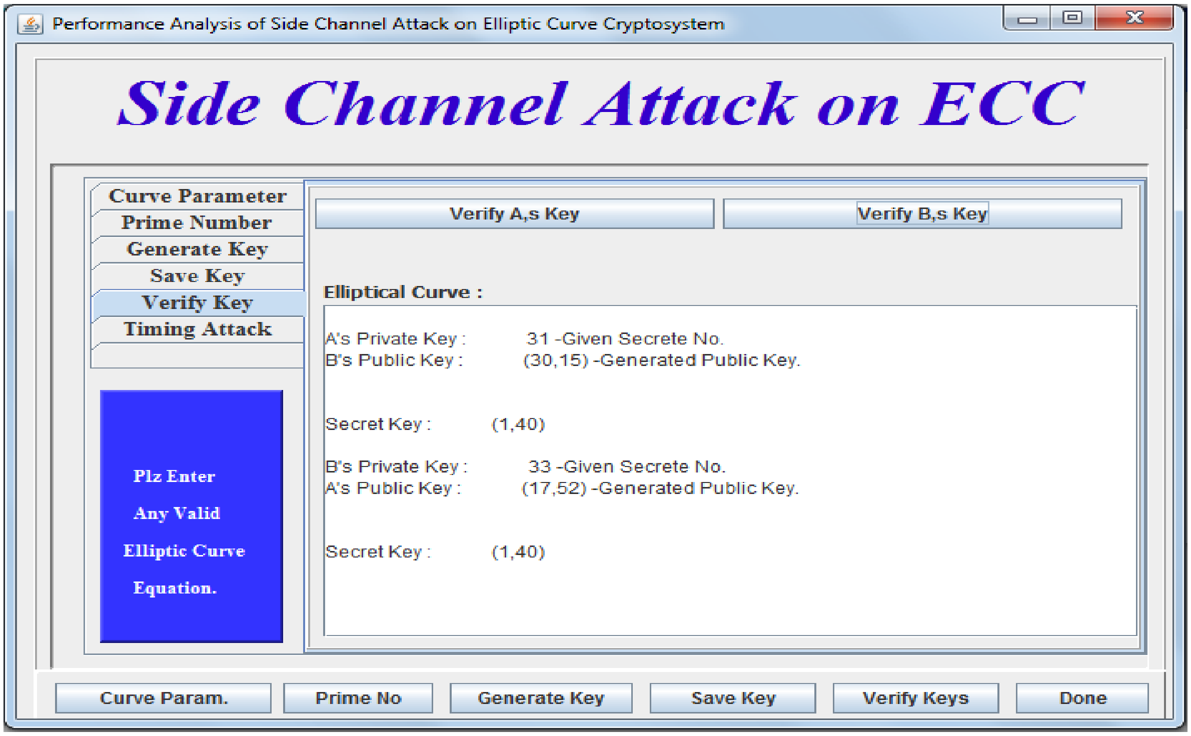
Elliptic Curve Key :

```
A"s Secrete No :        31 -Given Secrete No.

A"s Private Key :       31 -Given Secrete No.
A"s Public Key :       (17,52) -Generated Public Key.

B"s Secrete No :        33 -Given Secrete No.

B"s Private Key :       33 -Given Secrete No.
B"s Public Key :       (30,15) -Generated Public Key.
```

Plz Enter
Any Valid
Elliptic Curve
Equation.

Figure 6.4: Public key generation for user

Figure 6.4 describes the public Key generation for the users A and B on the basis of Secrete number selected by the user A and B respectively. Public key of the user is calculated with the help of Random number i. e. secrete number selected by the user and the Generating point set with the help of order of point by performing addition and doubling operation on them.



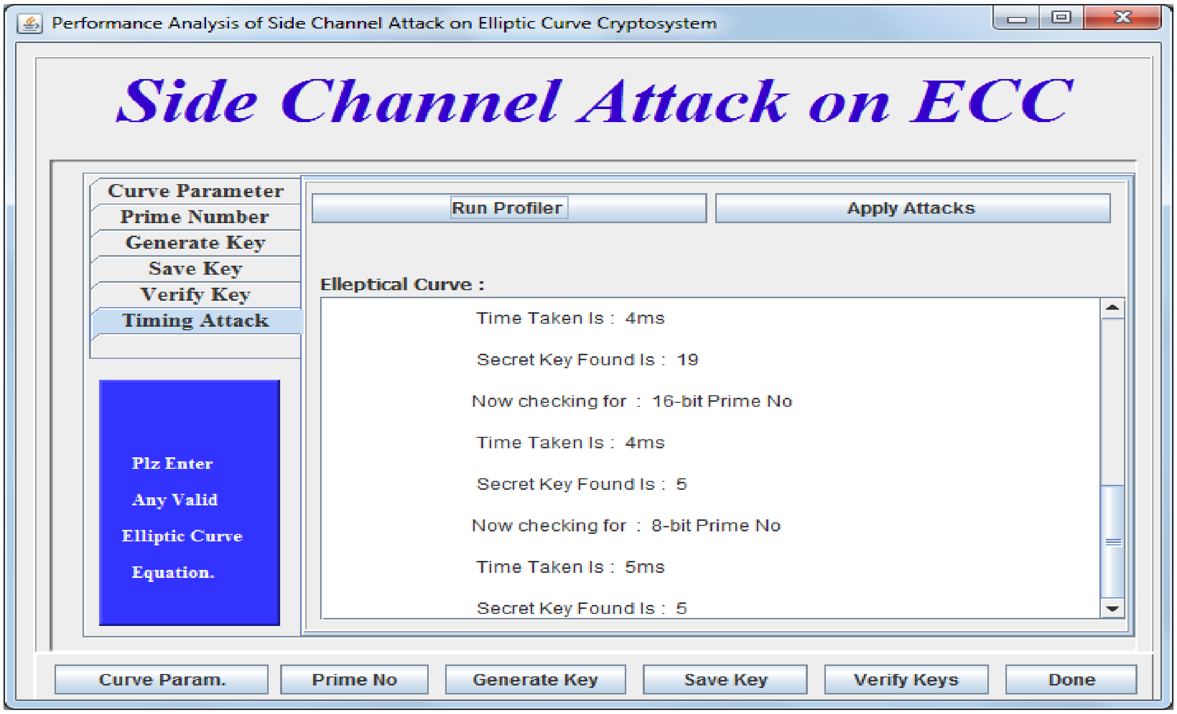
```
Elliptical Curve :

A's Private Key :        31 -Given Secrete No.
B's Public Key :        (30,15) -Generated Public Key.


Secret Key :        (1,40)

B's Private Key :        33 -Given Secrete No.
A's Public Key :        (17,52) -Generated Public Key.


Secret Key :        (1,40)
```

Figure 6.5: Secrete key generation

Figure 6.5 shows the secrete key or shared key. Which is calculated for user A by private key of user A and Public key of user B, and for user B by private key of user B and public key of user A.

6.1.1 Implementation of Profiler:

Profiler code analyses the time required to execute the set of instructions written under it. Profiler evaluate the running time of set of instructions written for Addition and Doubling operations of elliptic curve cryptosystem for calculating the public keys of the user involved in communication, as the running time required for Addition operation is different than the running time required for Doubling operation of the Elliptic curve cryptosystem. On the basis of these running time required   for different size of prime number, the Probable Secrete key is calculated. Figure 6.6 shows the Elapse Time for the respective size of Prime number. Figure 6.7 shows the probable secrete key after applying the Attack.

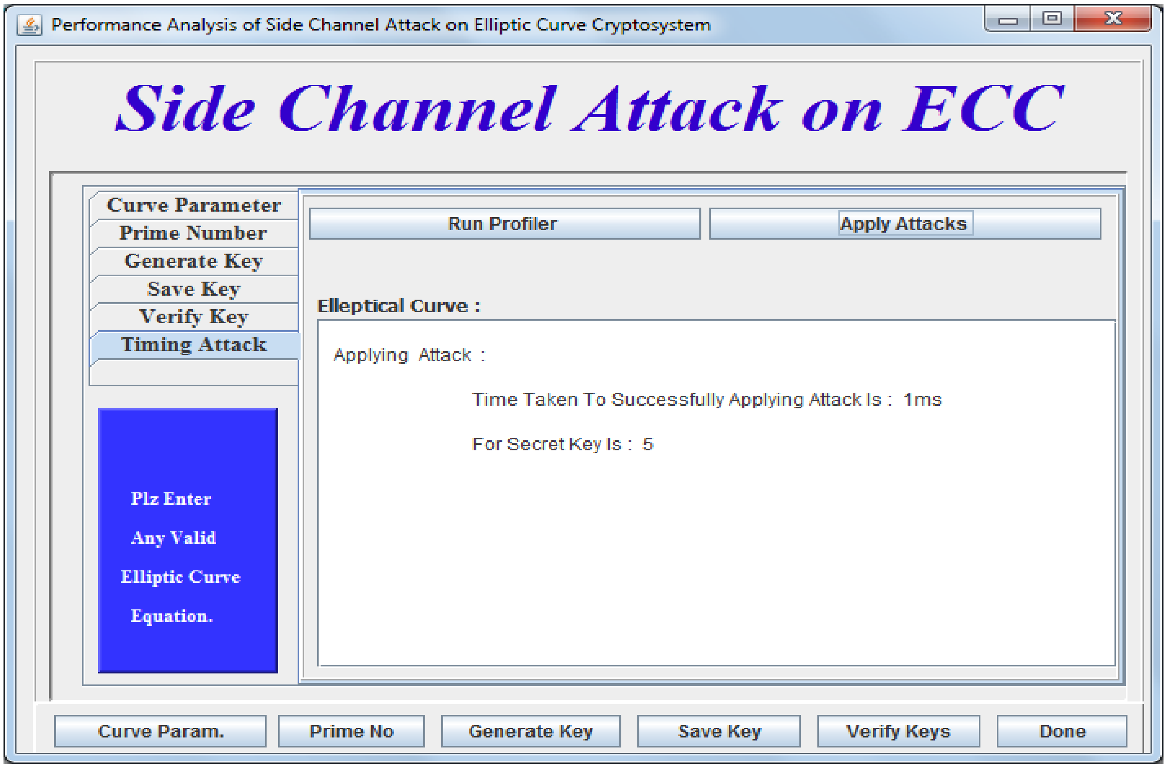


Plz Enter Any Valid Elliptic Curve Equation.

Elleptical Curve :

```
Time Taken Is :  4ms
Secret Key Found Is :  19
Now checking for  :  16-bit Prime No
Time Taken Is :  4ms
Secret Key Found Is :  5
Now checking for  :  8-bit Prime No
Time Taken Is :  5ms
Secret Key Found Is :  5
```

Figure 6.6: Elapse Time for respective bits of Prime Number



Plz Enter Any Valid Elliptic Curve Equation.

Elleptical Curve :

```
Applying  Attack  :
          Time Taken To Successfully Applying Attack Is :  1ms
          For Secret Key Is :  5
```

Figure 6.7: Probable key after applying the attack.

6.2 Experimental results

Table 6.1 describes some practical information applied to random scalars of different sizes. The length of prime number, the complete running time, and the average number of iterations for one bit of the scalar are shown. It should be noted that all the point computations and field inversions were repeated several times, until the variance of the single timings for these operations was sufficiently small (and hopefully the timing error sufficiently small). Seeing the number of iterations (i.e. backtracks) especially for large scalars, several parameters of our implementation of the timing attack algorithm (e.g. the sample number, the definition of "sufficiently different") are concluded. Further examinations and determination of optimal parameters have to be done in future.

Length of Prime Number	Average number of iterations per bit	Total Running Time
2^{24}-1	3.54	11 min 13 sec
2^{24}	0.71	1 min 20 sec
2^{32}-1	1.16	18 min 37 sec
2^{32}	0.84	7 min 1 sec
2^{48}-1	1.10	27 min 15 sec
2^{48}	0.94	12 min 8 sec
2^{96}-1	2.33	53 min 32 sec
2^{96}	1.28	22 min 7 sec
2^{128}-1	4.06	58 min 8 sec
2^{128}	1.37	43 min 32 sec
2^{159}-1	7.38	59 min 26 sec
2^{159}	1.88	53 min 15 sec

Table 6.1: Experimental Results of timing Attack

Figure 6.8 shows the graph of Length of Prime Number and Total Running Time Required, where as figure 6.9 shows the graph of Length of Prime Number and the Average Number of Iterations per Bit.

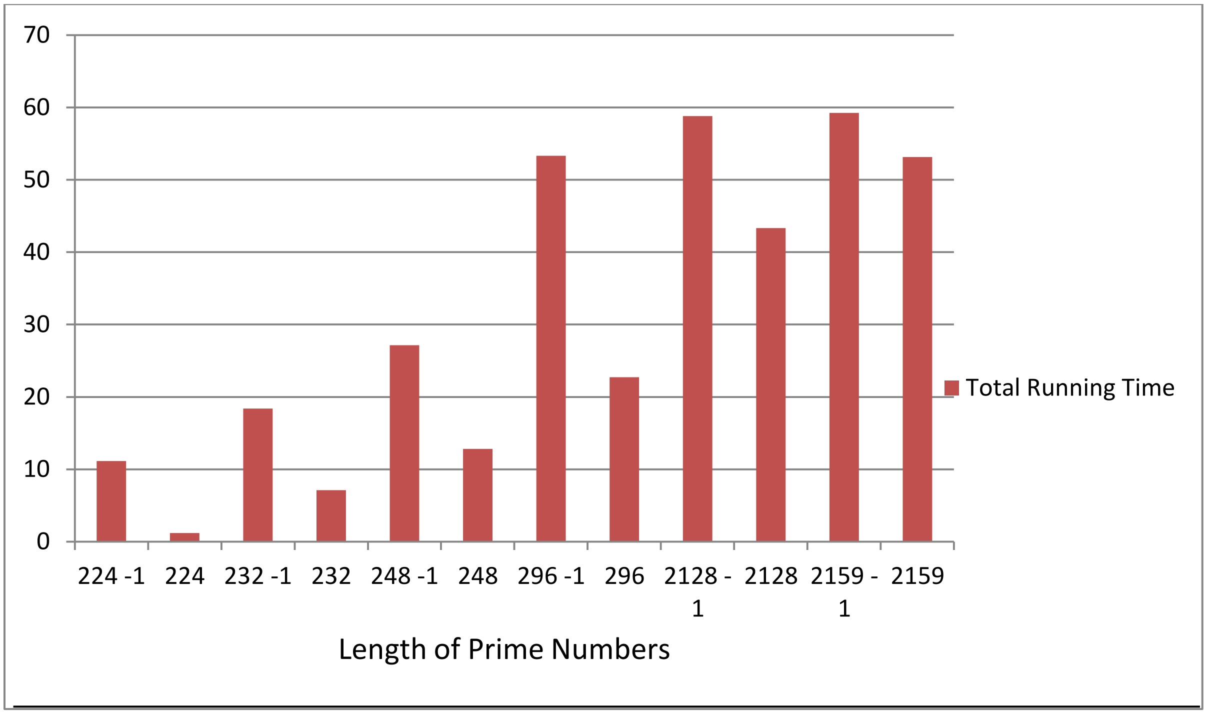


Figure 6.8: Graph of Length of Prime Number Vs Total Running Time

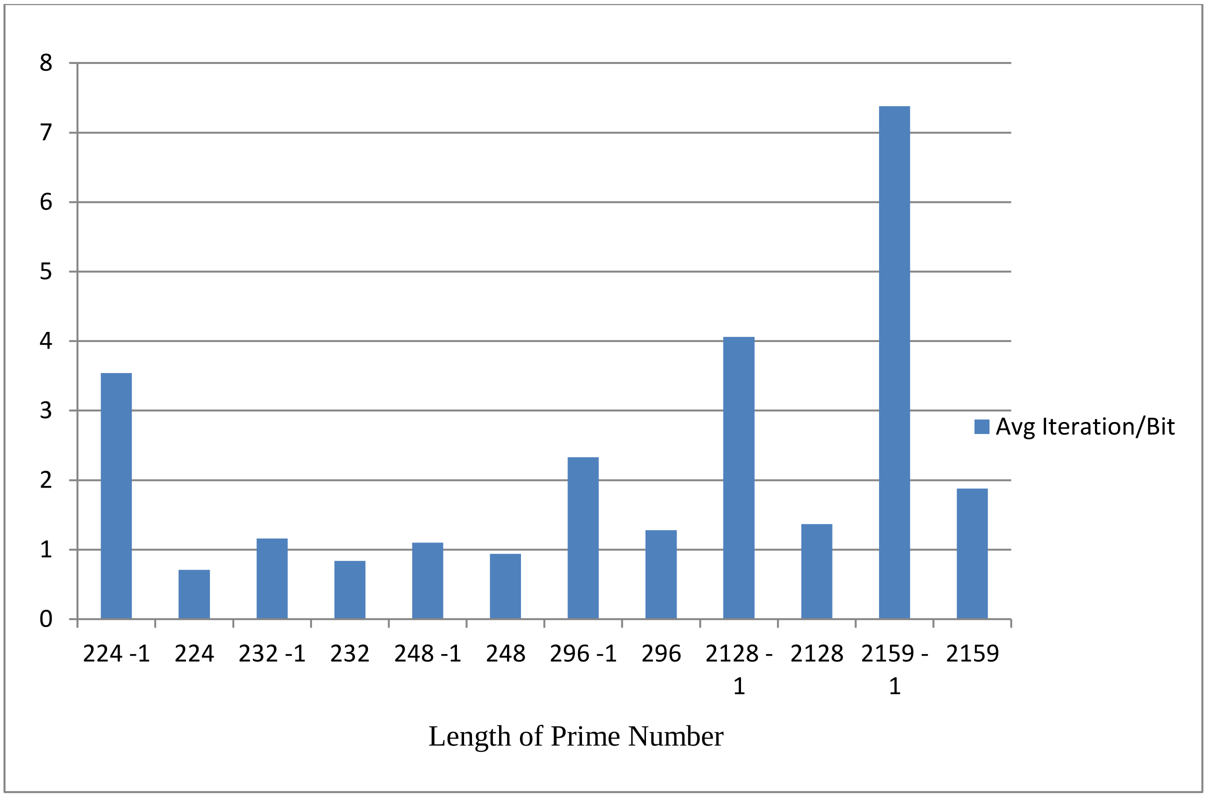


Figure 6.9: Graph of Length of Prime Number Vs Avg Iteration/Bit

Publication Partner:
International Journal of Scientific and Research Publications (ISSN: 2250-3153)

7. CONCLUSION

Proposed system reviewed Timing attack on elliptic curve cryptosystems. The running time of the attack can be several minutes; it always succeeded to determine the secret scalar.. Therefore timing attacks should be considered as a serious threat for EC security system implementations.

With this work, proposed system examined the necessary factors to assess the vulnerability of an ECC system to the Timing Attack. Proposed system have tried to find weaknesses in computational algorithm used in ECC implementation. However, it cannot be claimed so far that all implementations of ECC are resistant to timing attacks. It seems that the basic algorithms used to implement ECC system could cause the system to be vulnerable to a timing attack. The implementation determines whether a cryptosystem is vulnerable to a timing attack.

In future work we are optimistic that further experiments can greatly improve the still large running time by searching for more optimized parameters for Elliptic Curve based Cryptosystems.Further research is needed to examine whether the third timing attack variant can also be applied to different multiplication algorithm, especially the very popular algorithms using signed binary expansions.

REFERENCES

[1] EkambaramKesavulu Reddy, “Elliptic Curve Cryptosystems and Side-channel Attacks”, Published in international Journals of Network Security, 2009.

[2] Kazuo Sakiyama, Elke De Mulder, Bart Preneel, and Ingrid Verbauwhede“Side-channel Resistant System-level Design Flow for Public-key Cryptography” GLSVLSI’07, March 11–13, 2007, Stresa-Lago Maggiore, Italy.

[3] Marc Joye, “Elliptic curves and side channel analysis”, published in ST Journal of System Research, 2003.

[4] Zhang Tao, Fan Mingyu&ZhengXiaoyu“ Secure and efficient elliptic curve cryptography resists side-channel attacks” published in Journal of Systems Engineering and Electronics Vol. 20, No. 3, 2009, pp.660–665.

[5] Werner Schindler1, Kerstin Lemke, and Christof Paar, “A Stochastic Model for Differential Side Channel Cryptanalysis”, International Association for Cryptologic Research, 2005.

[6] Elisabeth Oswald, “Enhancing Simple Power-Analysis Attacks on Elliptic Curve Cryptosystems”, Springer-Verlag Berlin Heidelberg, 2003.

[7] KatsuyukiOkeya and Kouichi Sakurai, “Power Analysis Breaks Elliptic Curve Cryptosystems Even Secure against the Timing Attack”, Springer-Verlag Berlin Heidelberg, 2000.

[8] Dhem J.-F., Koeune F., Leroux P.-A., Mestre P., Quisquater J.-J., and Willem J.- L.s, “A practical implementation of the timing attack”, Proc. of the Third Working Conference on Smart Card Research and Advanced Applications (CARDIS 1998), LNCS 1820, 1998.

[9] Boneh D. and Brumley D. Remote timing attacks are practical, To appear in the 12thUsenix Security Symposium, 2003.

[10] Lu Z., Mah M., Neve M., and Peeters E., “Timing Attacks on Elliptic Curve Cryptosystems”, Project Presentation for course "CS588: Cryptology Principles and Applications, Fall 2001", University of Virginia, Department of Computer Science.

[11] Kocher P., Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS, and the Systems, In N. Koblitz, editor, Advances in Cryptology - CRYPTO'96, Lecture Notes in

Computer Science vol. 1109, pp. 104 - 113, 1996.

[12] Wireless Application Protocol Forum, Wireless Transportation Layer Security, Version WAP-261-WTLS-20010406-a,21. April 2001.

[13] KatsuyukiOkeya, Hiroyuki Kurumatani, and Kouichi Sakurai, “ Elliptic Curves with the Montgomery-Form and Their Cryptographic Applications”, Springer-Verlag Berlin Heidelberg, 2003.

[14] Bodo M¨oller, “Securing Elliptic Curve Point Multiplication against Side-Channel Attacks”, Springer-Verlag Berlin Heidelberg, 2003.

[15] An Implementation Tutorial on “ Elliptic Curve Cryptography”, By Anoop MS.

[16] Dimitrios L. Delivasilis and Sokratis K. Katsikas, “Side Channel Analysis on Biometric-based Key Generation Algorithms on Resource Constrained Devices”,International Journal of Network Security, Vol.3, No.1, PP.44–50, July 2006.

[17] Ernie Brickell, Gary Graunke, Michael Neveand Jean-Pierre Seifert, “Software mitigations to hedge AES against cache-based software side channel vulnerabilities”

[18] Eric Brier and Marc Joye, “Weierstraß Elliptic Curves and Side-Channel Attacks”, [Published in D. Naccache and Pascal Paillier, Eds., Public Key Cryptography,vol. 2274 of Lecture Notes in Computer Science, pp. 335–345,Springer-Verlag, 2002.]

[19] Marc Joye and Christophe Tymen “Protections against Di_erential Analysis for Elliptic Curve Cryptography An Algebraic Approach”, [Published in C_ .K. Ko_c, D. Naccache, and C. Paar, Eds., Cryptographic Hardware and Embedded Systems, CHES 2001, vol. 2162 of Lecture Notes in Computer Science, pp. 377, Springer-Verlag, 2001.]

[20] Marc Joye and Jean-Jacques Quisquater, “Hessian Elliptic Curves and Side-Channel Attacks”, [Published in C. K. Koc D. Naccache, and C. Paar, Eds., Cryptographic Hardware and Embedded Systems– CHES 2001, vol. 2162 of Lecture Notes in Computer Science, pp. 402–410, Springer-Verlag, 2001.]

[21] Jean-Sebastien Coronand_EcoleNormaleSuperieure, “Resistance Against Di_erential Power Analysis for Elliptic Curve Cryptosystems”, [Published in C_ .K. Ko_c and C. Paar, Eds.,

Cryptographic Hardware and Embedded Systems, vol. 1717 of Lecture Notes in Computer Science, pp. 292 Springer-Verlag, 1999.]

[22]TetsuyaIzu, Bodo M¨oller, and Tsuyoshi Takagi, "Improved Elliptic Curve Multiplication Methods Resistant against Side Channel Attacks", [Appears in A. Menezes, P. Sarkar (Eds.): Progress in Cryptology – INDOCRYPT 2002, Springer-VerlagLNCS 2551, pp. 296–313, ISBN 3-540-00263-4.]

[23] Benoıt Chevallier-Mames, Mathieu Ciet, and Marc Joye, "Low-Cost Solutions for Preventing Simple Side-Channel Analysis: Side-Channel Atomicity", [Published in IEEE Transactions on Computers 53(6):760–768, 2004.]

[24] KatsuyukiOkeya and Kouichi Sakurai, "On Insecurity of the Side Channel Attack Countermeasure Using Addition-Subtraction Chains under Distinguishability between Addition and Doubling", [L. Batten and J. Seberry (Eds.): ACISP 2002, LNCS 2384, pp. 420–435, 2002. c Springer-Verlag Berlin Heidelberg 2002]

[25] KatsuyukiOkeya and Tsuyoshi Takagi, "A More Flexible Countermeasure against Side Channel Attacks Using Window Method", [C.D. Walter et al. (Eds.): CHES 2003, LNCS 2779, pp. 397–410, 2003. c Springer-Verlag Berlin Heidelberg 2003]

[26] KatsuyukiOkeya and Tsuyoshi Takagi, " The Width-*w* NAF Method Provides Small Memory and Fast Elliptic Scalar Multiplications Secure against Side Channel Attacks", [M. Joye (Ed.): CT-RSA 2003, LNCS 2612, pp. 328–343, 2003. c Springer-Verlag Berlin Heidelberg 2003]

[27] Werner Schindler, Kerstin Lemke, and Christof Paar, "A Stochastic Model for Differential Side Channel Cryptanalysis", [J.R. Rao and B. Sunar (Eds.): CHES 2005, LNCS 3659, pp. 30–46, 2005.c International Association for Cryptologic Research 2005]