TAMPEREEN TEKNILLINEN YLIOPISTO
TAMPERE UNIVERSITY OF TECHNOLOGY

SOHAIB UL HASSAN

EM SIDE CHANNEL ANALYSIS ON COMPLEX SOC ARCHITECTURES

Master of Science thesis

Examiner: Prof. Billy Bob Brumley
Examiner and topic approved by the
Faculty Council of the Faculty of
Computing and Electrical Engineering
on 8th June 2016

ABSTRACT

SOHAIB UL HASSAN: EM Side Channel Analysis on Complex SoC Architectures
Tampere University of Technology
Master of Science thesis, 66 pages
November 2016
Master's Degree Programme in Information Technology
Major: Pervasive Systems
Examiner: Prof. Billy Bob Brumley
Keywords: electromagnetic, side channel attacks, elliptic curve cryptography, wavelet decisioning, software defined radio

The EM side channel analysis is a very effective technique to attack cryptographic systems due to its non invasive nature and capability to launch an attack even with limited resources. The EM leakage from devices can give information about computations on the processor, which can in turn reveal the internal state of the algorithm. For security sensitive algorithms, these EM radiations can be exploited by the adversary to extract secret key dependent operations hence EM side channel must be studied for evaluating the security of these algorithms. Modern embedded devices composed of System-on-Chip architectures are considered hard targets for EM side channel analysis mainly due to their complex architecture. This thesis explores the viability of EM side channel attacks on such targets. There is a comprehensive literature overview of EM side channel analysis followed by a practical side channel attack on a SoC device using well know cryptographic library OpenSSL. The attack successfully extracts the secret key dependent operation which can be used to retrieve the private key in security protocols such as TLS and SSH. The thesis concludes, with practical single trace attacks, that cryptographic implementations can still be broken using EM side channel analysis, and a complex nature of the device have no significant effect when combined with signal processing methods for extracting side channel information, hence the cryptographic software implementations must address these issues.

PREFACE

The MSc thesis work was carried out at the Department of Pervasive Computing, Tampere University of Technology, supported by the project "Hardware Rooted Security (HRS)" from TEKES – the Finnish Funding Agency for Innovation.

I would start by expressing special thanks and acknowledgments to my advisor Prof. Billy Bob Brumley for providing me with this opportunity and for his help, guidance and valuable feedback in understanding and carrying out my academic work, and compiling my thesis. Thanks to Prof. Jarmo Harju and my coworkers at NISEC for providing me with a great and productive work environment. I would also like to thank Dr. Daniel Page and Jake Longo Galea from Bristol University for their guidance.

I thank my wife for her love and care, and for being there with me in everything, and my kid for being a source of happiness. Special thanks to my Parents who helped, supported and gave me good advice throughout my life, in particular my Dad for being a source of motivation.

Thanks to all my friends and family who supported me and my teachers who helped and guided me in the studies.

Tampere, 10.10.2016

Sohaib ul Hassan

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS AND SYMBOLS

EC	Elliptic Curves
ECC	Elliptic Curve Cryptography
ECDSA	Elliptic Curve Digital Signature
EM	Electromagnetic
SDR	Software Defined Radio
SoC	System on Chip
SSH	Secure Shell
TLS	Transport Layer Security

1. INTRODUCTION

We are living in times flooded with ubiquitous embedded devices constantly talking to each other and sharing information, creating a huge ecosystem of connected devices. Over the past decade there has been a rapid growth in intelligence and interconnectedness of embedded devices. Embedded systems have been the building block of most modern devices, but the use of more complex System-on-Chip architectures and design reuse have enabled the developers to release these devices for consumer application market at a more rapid rate. Theses complex micro architectures range from mobile devices to payment systems and to more recent trends towards Internet-of-Things (IoT). User applications running on these devices involving financial transactions, on-line shopping and digital currency use security protocols like Transport Layer Security (TLS), Secure Shell (SSH), Bitcoin which can be exploited if not thoroughly tested for any security vulnerabilities.

The increased architectural complexity and rapid deployment of devices brings numerous design and development challenges for security experts such as demand of computational and energy efficiency which also significantly impact the device security and functionality. According to [50] the processing of security primitives on embedded devices are constrained by the computational power and throughput referred as "security processing gaps", while embedded systems powered by battery are constrained by energy consumption "battery gaps", which sets a limit on the amount of energy a secure operation can draw, requiring both an efficient software and hardware implementations. There is another gap which is perhaps more challenging from information theoretic and cryptographic point of view: between security critical operations running as data encryption standards or client server communication protocols, and their provable secure properties. In order to address these gaps information security experts must develop cryptographic implementations that are not only computationally efficient but also secure at the same time.

Typically cryptanalysis focuses on targeting the mathematical design of the crypto-

graphic systems to evaluate the strength of the cryptographic system. Side channels or Covert Channels on the other hand are those communication channels that take advantage of the physical characteristics of the secure systems. They are also termed as implementation attacks as they try to target the weak implementation of the cryptographic systems. These channels can be exploited to gather information in a way that can breach the confidentiality and integrity of cryptographic systems. There are different side channels that have bee studied such as

- Timing Attack [12]
- Power Analysis Attack [39]
- Electromagnetic (EM) Side Channel Analysis Attack [40]
- Acoustic Side Channel Analysis Attack [25]

The techniques used to target these side channels usually vary based on the physical properties of the channel, however the information obtained is more or less similar. These channels can either reveal complete information to break the system or additional cryptographic/mathematical techniques are used on the gathered information.

The public key cryptographic implementations such as Rivest Shamire Adleman (RSA) modular exponentiation and Elliptic Curve Cryptography (ECC) point multiplication contain operations that are dependent on the secret key. EM side channel analysis can give information about these cryptographic operations and may reveal the secret key. However, using only one EM signal to exploit this attack is considered a hard problem. Recently advanced signal processing methods have opened new research area for applying cheap EM side channel attacks in presence of limited side channel data [24, 22].

For evaluating the effectiveness of EM side channel analysis single trace attacks using cheap equipment are demonstrated on ECC and RSA using well known cryptographic libraries. The results indicated strong presence of EM side channel leakage on SoC device revealing the secret key dependent operations. The architectural complexity of SoC does play some role by introducing noise to EM measurements however signal processing methods can be used to overcome the noise. This suggest that EM side channel attacks are viable and can exploit the weakness in software

implementation with low cost setup. Also since EM signals can be captured at some distance from the device they can have practical implications, never the less they give insight about vulnerabilities in cryptographic implementations and must be investigated with scrutiny.

Chapter 2 gives a theoretical background of cryptography. Although cryptography is broadly divided into symmetric and asymmetric categories, the thesis work is done on asymmetric cryptography. There is more detailed discussion on RSA and ECC. The idea is to give an insight on how these cryptographic systems are designed and implemented, so it is easier to follow the rest of the thesis when the actual side channel attack is discussed.

Chapter 3 is a brief introduction and background on the topic side channel analysis. It tries to develop from brief survey on different side channel analysis techniques and finally more details of Electromagnetic Side Channel Analysis which is the basic theme of the thesis. The explanation is complemented with discussion on some previous work in the field. This chapter will set theoretical background of what is to be followed as the actual work.

Chapter 4 discusses about application of signal processing techniques for EM side channel analysis. It explains the theory behind signal representation of the EM side channels and the signal processing method to extract the compromising side channel frequencies, applied in the side channel attack presented in thesis.

Chapter 5 highlights the experiments carried out on ECC and RSA cyptosystems running on SoC device and presents the results of the attacks. Chapter 6 explains some possible ways to mitigate these side channel attacks, while Chapter 7 is the discussion on lessons learned during the thesis and discusses some useful application of the results obtained for carrying out future research.

2. THEORY OF CRYPTOGRAPHY

Cryptography is the science of transforming semantic structure of information. It deals with using mathematical tools and techniques for protecting critical information. According to [42, p. 5] cryptography aims at providing a set of information security goals as highlighted in Table 2.1. While the list is not exhaustive, it gives some of the basic principles to qualify as a cryptographic system. In modern cryptography, a cryptographic system is seen as an algorithm which provides at least a set of two operations, "encryption" -transform and hide the information and "decryption" -transform the encrypted information back into its original form.

Table 2.1 *Information Security Goals of Cryptography*

Confidentiality	Only authorized parties can access information
Data Integrity	Information cannot be tampered by any means while transferred between engaged parties
Entity Authentication	Way to prove the identity of engaged parties
Message Authentication	Way to prove that information came from the authorized party
Signature	Bind the information to the legal owner of information

Broadly speaking any cryptographic system can be classified as:

Symmetric Key Cryptography also known as private-key cryptography involves the use of a same shared secret key between engaged parties, using the same encryption and decryption method. There is a very brief discussion on symmetric cryptography in Section 2.3, as the focus of the thesis towards asymmetric cryptography. Advanced Encryption Standard (AES) and Data Encryption Standard (DES) are well known symmetric cryptography solutions.

Asymmetric Cryptography also known as public-key cryptography makes use of two different keys for doing encryption and decryption. Section 2.4 explains it in more detail, with examples on RSA and ECC.

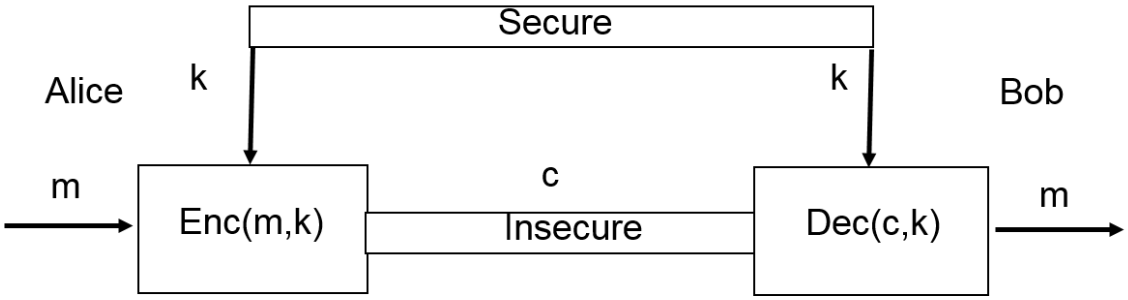

Figure 2.1 *Symmetric Key Cryptography*

Protocols are applications of cryptography algorithms on top of communication layer protocols for secure transfer of information between sender and receiver. TLS is one of the most widely used protocols on the Internet for maintaining secure sessions between client and server on the Internet.

2.1 Symmetric Key Cryptography

Symmetric key cryptography can be best understood by Figure 2.1. The two engaging parties Alice and Bob communicate over an insecured channel. Alice encrypts the secret message m with private key k using an encryption function enc to output a cipher text c and send it to Bob. On the receiving end Bob decrypts the cipher text c using the same private key k and decryption function dec to get back the original message m. For this scheme to work, the secret key k must be shared among the communicating parties through a secure channel. Another important aspect, encryption and decryption functions in symmetric key cryptography are usually same. The private key sharing is one of the major challenges in symmetric key cryptography. Symmetric key cryptosystems are designed either as:

Stream Ciphers perform encryption or decryption on each incoming bit of message stream using a stream of random non repeating key. The idea is to transform each bit of plaintext to different bits of cipher text. For a secure and robust stream cipher the key should be true random number.

Block Ciphers breaks down the message into fixed size blocks and perform encryption and decryption operation on the block.

More detailed information on symmetric key cipher can be found here [42, p. 15].

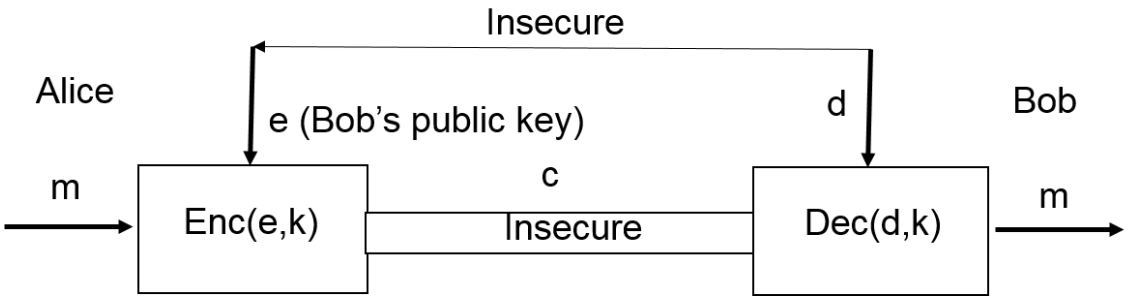

Figure 2.2 *Asymmetric Key Cryptography*

The remaining of the thesis will talk about asymmetric cryptography.

2.2 Asymmetric Cryptography

Asymmetric cryptography deals with the problem of private key distribution by using a pair of keys, *private key* -which is not shared and kept secret and *public key* -which is shared and publicly available hence the term public key cryptography. Unlike the symmetric key cryptosystems, here the encryption and decryption functions are different, and in a typical case the public key is used to perform encryption while the decryption uses private key. Figure 2.2 depicts a typical public key cryptosystem. Here Alice wants to send Bob an encrypted message which only Bob can see. Bob generates a pair of public and private keys *(e,d)* and share the public key e. Allice uses the public key e provided by Bob to encrypt the message m and generate the cipher text c. Bob can then use his private key d to decrypt c and get the original message m. For a secure public key cryptosystem it is not possible to generate the private key given the public key. The only drawback is that these cryptosystems are computationally expensive especially on embedded devices with low computational power. Public key cryptosystems are widely studied and practiced in modern cryptography with applications such as secret key exchange, digital signature, digital certificates or use as encryption tools [42, p. 283].

2.2.1 Rivest Shamire Adleman (RSA)

RSA is one of the widely deployed cryptosystems. Most of the security protocols on the Internet use RSA for secret key exchange, digital signature schemes and issuing certificates. RSA is based on integer factorization of the product of two large prime

numbers which is considered a hard mathematical problem; however this also adds to the computational complexity and requires use of more efficient implementation techniques.

Mathematics of RSA [42, p. 285] is based on modular arithmetic in integer ring Zn [42, p. 76]. RSA first generates key using these steps

- Generate two large prime numbers p and q where $p \neq q$ and p is roughly same size as q
- Calculate the modulus $n = pq$
- Calculate the Euler Totient Function [42, p. 65] $\phi(n) = (p-1)(q-1)$
- Choose public exponent randomly from $e \in 1, 2, ..., \phi(n) - 1$ where $\gcd(e, \phi(n)) = 1$
- Compute a unique d which is the private exponent such that $d = e^{-1} \bmod (\phi(n))$
- Here (n, e) becomes the public key and d is the private part. Knowing n and e it should be infeasible to generate d [31, p. 12].

Using the public exponent e RSA performs encryption whereas decryption is performed using the private exponent d.

Encryption: $c = m^e(\text{mod}(n))$ where c is the cipher text and m is the message
Decryption: $m = c^d(\text{mod}(n))$ to get back the original message m

Implementation of RSA heavily relies on modular exponentiation of large prime numbers. Usually p and q are chosen to be 512 -2048 bit numbers. This means that maximum bit length of encryption or decryption is $n = pq$. Also n should be chosen in a way that it gives many p and q pairs. Exponentiation of big numbers is computationally very slow for practical use. As an example consider a 1024 bit number n. The simplest method for calculating the modular exponentiation is Square followed by sequence of Multiply operations

$Square(n) -> Multiply(n^2) -> Multiply(n^3) -> Multiply(n^4) -> ...$

Considering that we need this exponentiation for creating HTTPS session on a mobile device we need 2^{1024} multiplications, which is infeasible. This requires use of Fast Exponentiation methods. The most commonly used method is the **left-to-right square and multiply** or **binary** algorithm [36] which scans the exponent bits from left MSB to right LSB. The accumulator c is first initialized to the input x. If the current scanned bit is '1' then a multiply operation is followed after a square and stored in the accumulator c. The method is listed in Algorithm 2.1.

```
Square_and_Multiply( x,d,n )
        x-> input
        d-> exponent bits
        n-> modulus
        c = x
        for i = size(d)-1 to 0
                c = c^2 mod(n)
                if d(i)==1
                        c = c * x mod(n)
        return c
```

Algorithm 2.1 *RSA left-to-right Square and Multiply Method*

There is another more efficient implementation called ***m-ary algorithm*** [36] listed in Algorithm 2.2. This technique requires scanning the bits of the exponent $\log_2 m$ at a time instead of just one bit like in the previous case. The method uses binary expansion of the exponent using a partition function. The idea is to partition the exponent into blocks s of fixed length r such that $sr = k$ with 0 padding to make it equal to length k if necessary. Rest of the algorithm works similar to Algorithm 1 i.e. scanning bits of exponents and raising the input 2-to-the-power of r at each step. There is a multiplication of the input with x to the power Wi where Wi is the non-zero value of r bits scanned in each step.

The problem with m-ary algorithm method is the number of multiplication operations depends on the size of the partition of exponent bits. If its an n-bit partition, the number of multiplications increase as n approaches to 0. However as n approaches to infinity the probability of multiplication increase, considering there is equal probability for occurrence of 0 and 1 in exponent. For an optimal solution the exponent bits are decomposed into zero and non-zero words Wi of variable length, by making sure that for each non-zero word the LSB is 1, hence effectively increasing the probability of zero word occurrence. The technique is known as ***sliding window***

[36] and the algorithm is listed in Algorithm 2.3. Square and Multiply operations takes more time to execute then just Square operation, this kind of timing variation is critical in side channel attacks. In EM analysis the amount of power consumed is directly correlated to the side channel leakage, so heavy operations should give more EM leakage.

```
m_Ary( x,d,n )
        x-> input
        d-> exponent bits
        n-> modulus
        w-> window size
        Precompute x^i(mod(n)) for i=2,3 .. m-1
        Decompose d into r-bit words W(i) for i = 0,1,2,..,s-1
        c = x^W(s-1) mod(n)
        for i = s-2 to 0
                c = c^(2^r) mod(n)
                if W(i) != 0
                        c = c * x^W(i) mod(n)
        return c
```

Algorithm 2.2 *RSA m-ary method*

```
Sliding_Window( x,d,n )
        x-> input
        d-> exponent bits
        n-> modulus
        w-> window size
        Precompute x^i(mod(n)) for i=3,5,7 .. 2^(d-1)
        Decompose d into zero and non zero words W(i) of fixed
           ↪ size where i = 0,1,2,..,s-1
        c = x^W(s-1) mod(n)
        for i = s-2 to 0
                c = c^(2^length(W(i))) mod(n)
                if W(i) != 0
                        c = c * x^W(i) mod(n)
        return c
```

Algorithm 2.3 *RSA Sliding Window Method*

2.2.2 Eliptic Curve Cryptography (ECC)

The use of large prime factorization hence large key size means more power consumption and memory requirements which is not practical in power and memory constrained devices such as mobile devices. Also large digital certificates can become bottleneck in wireless applications. As an alternate solution the first breakthrough was discovered by Koblitz and Miller who purposed Elliptic Curve Cryptography with cryptographic proven properties while using smaller secret key.

Mathematics of ECC

The elliptic curve E is a set of points over finite field F defined by the Weierstrass equation [31, p. 76]

$$E : y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6$$

The above equation is further reduced for prime finite field F_p where $p > 3$

$$E : y^2 = x^3 + ax + b$$

These set of points on the elliptic curve belongs to $E(F_p)$ which is an additive abelian group over the prime field with identity element at point of infinity (∞). The Elliptic curve cryptography is based on the fact that point multiplication of a point P with a secret scalar d yields a new point Q on the curve over F_p where p is large, it should be computationally infeasible to find d knowing P and Q. Given the points $P = (x_1, y_1)$ and $Q = (x_2, y_2)$

- point addition $P + Q$ gives a third point $R = (x_3, y_3)$ on the curve, which is the reflection of point of intersection between the curve and line with slope λ passing through P and Q, given by equation

$$x3 = \lambda^2 - x_1 - x_2 \text{ and } y_3 = \lambda(x_1 - x_3) - y_1 \text{ where } \lambda = \frac{y_2 - y_1}{x_2 - x_1}$$

- similarly point doubling $P + P = 2P$ is the reflection of point of intersection between the curve and the tangent line passing through P

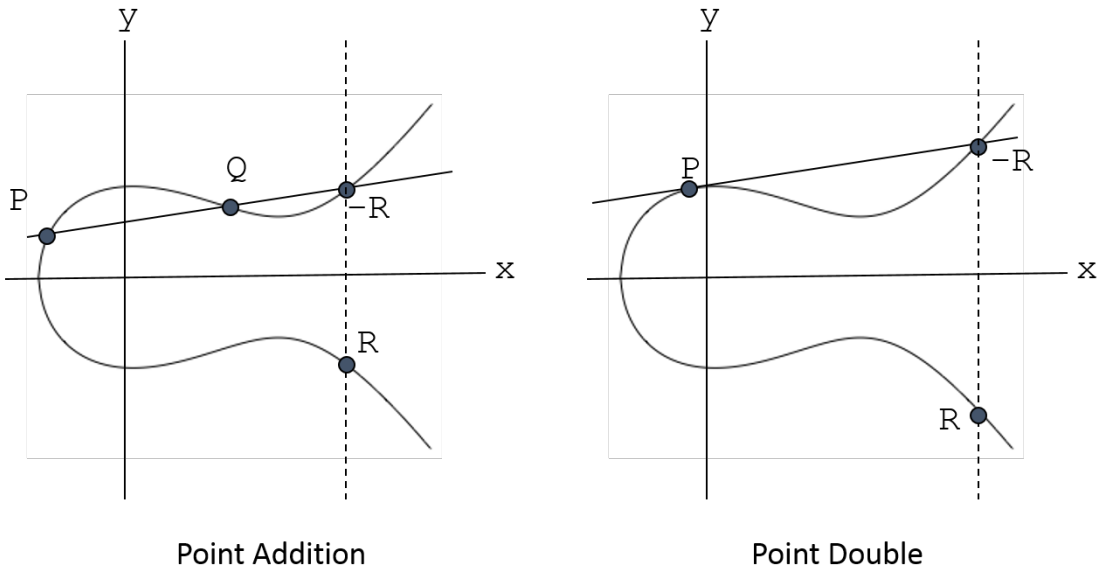

Figure 2.3 *Elliptic Curve Point Addition and Double*

$$x_3 = \lambda^2 - 2x_1 \text{ and } y_3 = \lambda(x_1 - x_3) - y_1 \text{ where } \lambda = \frac{3x_1^2+a}{2y_1}$$

The geometric interpretation of point addition and doubling is shown in Figure 2.3. The point addition generates a new point R which is a projection about x-axis of point of intersection $-R$ on the curve with line passing through P and Q, where as point doubling generates a new point R which is a projection about x-axis of point of intersection $-R$ on the curve with tangent line at point P.

Implementation of ECC

Elliptic curve cryptography requires adding points on a curve in finite filed repeatedly d times to generate a new point on the curve, where d is the secret scalar

$$Add(P) -> Add(2P) -> Add(3P) -> Add(4P) -> ...$$

This point multiplication with scalar is the most computationally expensive operation in ECC so more efficient methods are required for speed up. The use of point doubling where appropriate can reduce execution times. The reason is that each point addition and doubling consists of multiple square and multiply operation where point addition dominates in the number of operations. Before looking into the actual point multiplication method, we will first establish how the individual point

addition and point double operations execute. Looking at the mathematical interpretation of point addition and doubling in affine coordinates we can observe that it requires finite field inversions, which are not efficient to perform on the hardware. Hence more efficient methods to minimize these inversions have come forth [17, 48] which uses alternate coordinate systems such as projective coordinates, Jacobian coordinates, Chudnovsky Jacobian coordinates and Lambda coordinates. Often mixed coordinate systems are used to take advantage of the different coordinate system based on the operation on the point.

```
Point_Addition( P,Q,E )
        P = (X_1 , Y_1 , Z_1) -> input point in Jacobian
           ↪ coordinate
        Q = (X_2, x_2) -> input point in Affine coordinate
        E: y^2 = x^3 + a * x +b -> input curve where a=-3
        P + Q = ( X_3 , Y_3 , Z_3) -> output in Jacobian
           ↪ coordinate
        if Q = infinity return P
        if P = infinity return (X_2 , Y_2 , 1)
        T_1 = Z_1^2
        T_2 = T_1 * Z_1
        T_1 = T_1 * x_2
        T_2 = T_2 * y_2
        T_1 = T_1 - X_1
        T_2 = T_2 - Y_1
        if T_1 = 0 and T_2 = 0 return (P+P)
        if T_1 = 0 and T_2 != 0 return infinity
        Z_3 = Z_1 * T_1
        T_3 = T_1^2
        T_4 = T_3 * T_1
        T_3 = T_3 * X_1
        T_1 = 2(T_3)
        X_3 = T_2^2
        X_3 = X_3 - T_1
        X_3 = X_3 - T_4
        T_3 = T_3 - X_3
        T_3 = T_3 * T_2
        T_4 = T_4 * Y_1
        Y_3 = T_3 - T_4
        return( X_3 , Y_3 , Z_3)
```

Algorithm 2.4 *Point Addition in Affine-Jacobian coordinates*

The choice of coordinate representation depends on the curve and the type of elliptic

curve point multiplication method. For example, the point addition in Jacobian coordinate takes 12 multiply and 4 square operations while point doubling takes 4 multiply and 4 square operations, while on the other hand point additions and point doubling take 12 multiply 2 squares and 7 multiply 3 squares respectively in projective coordinate system [31, p. 92]. Clearly we can see point addition for

```
Point_Doubling( P,Q,E )
        P = (X_1 , Y_1 , Z_1) -> input point in Jacobian
            ↪ coordinate
        E: y^2 = x^3 + a * x +b -> input curve where a=-3
        P + P = 2(P) = ( X_3 , Y_3 , Z_3) -> output in Jacobian
            ↪ coordinate
        if P = infinity return infinity
        T_1 = Z_1^2
        T_2 = X_1 - T_1
        T_1 = X_1 + T_1
        T_2 = T_2 * T_1
        T_2 = 3(T_2)
        Y_3 = 2(Y_1)
        Z_3 = Y_3 * Z_1
        Y_3 = Y_3^2
        T_3 = Y_3 * X_1
        Y_3 = Y_3^2
        Y_3 = Y_3 / 2
        X_3 = T_2^2
        T_1 = 2(T_3)
        X_3 = X_3 - T_1
        T_1 = T_3 - X_3
        T_1 = T_1 * T_2
        Y_3 = T_1 - Y_3
        return( X_3 , Y_3 , Z_3)
```

Algorithm 2.5 *Point Double in Jacobian coordinates*

projective takes more time to execute than Jacobian coordinates however point multiplication takes more time in Jacobian coordinates. Algorithm 2.4 gives the listing for instructions performed during one point addition operations in mixed Affine-Jacobian coordinate giving 8 multiply and 3 square operations, where as Algorithm 2.5 gives point double operation in Jacobian coordinates with 4 multiply and 4 square operations. The two operations are still expensive but due to difference in the execution order and number of instructions in point addition and double there is a timing variation in their execution which can be exploited through side channel

leakage as discussed in Chapter3 and Chapter5.

In order to compute point multiplication dP efficiently i.e. sequence of double and add operations, there are a number of methods [31, 7]. The most generalized method for point multiplications is **left-to-right binary method** which is analogous to modular exponentiation method of RSA with square and multiply replaced with double and add operations. The Algorithm 2.6 algorithm first initialize Q to ∞, then scans from MSB to LSB of the scalar d and performs a double operation every time, except when the bit is 1 both double and add operations are performed. The probability of occurrence of add operations is 1/2 the length of scalar d.

```
Left_to_Right_Double_ADD( P , d )
        P-> input point on the curve
        d-> scalar bits
        Q-> point at infinity
        for i = size(d)-1 to 0
                Q = Q + Q
                if d(i)==1
                        Q = Q + P
        return Q
```

Algorithm 2.6 *Left to Write Doable and Add*

As the add operations are computationally heavy more efficient implementation use the secret scalar with low density of '1' bits known as **Non Adjacent Form (NAF) point multiplication** [31, p. 98]. The NAF representation of scalar has less probability for occurrence of non-zero digit hence effectively reducing the number of add operations in point multiplication. The following hold true for a NAF representation of scalar d

- NAF representation ensures that no two adjacent digits are non-zero
- NAF representation is balanced meaning it allows both negative and positive digits $\{-1, 0, +1\}$
- probability for occurrence of non-zero digit is 1/3 since the set contains three digits
- the size of scalar d is smaller by at least one or less for a given NAF representation such that $2^s/3 < d < 2^{s+1}/3$ where s is the size of $NAF(d)$

NAF representation is very effective in reducing the Hamming weight by splitting the scalar to $-1, 0, 1$. Point multiplication with NAF representation first calculates the NAF digits of the scalar and then use a method similar to binary left-to-right. Algorithm 2.7 gives listing for NAF point multiplication. Whenever the digit is non zero there is a point addition.

```
NAF_Point_Multiplication( P , d )
        P-> input point on the curve
        d-> scalar bits
        Q-> point at infinity

        Compute NAF digits
        i = 0
        while d > 0
                if d = odd
                        d(i) = 2 - (d mod 4)
                else
                        d(i) = 0
                d = d - d(i)
                d = d / 2
                i = i + 1
        end while

        Compute scalar multipliation
        for i = size(d)-1 to 0
                Q = Q + Q
                if d(i) > 0
                        Q = Q + P
                if d(i) < 0
                        Q = Q - P
        return Q
```

Algorithm 2.7 *NAF point multiplication*

To achieve better running times for NAF method instead of scanning the bits one digit at a time it is scanned w bits at a time similarl to m-arry method in Algorithm 2. The method is **wNAF point multiplication** where the window size w are the digits of scalar d computed per iteration. These pre-computed points increase the overall performance with expense of some extra memory. Similar to the NAF representation wNAF possess these properties

- wNAF representation ensures the adjacency property holds for w digits i.e. no

two adjacent w digits are non-zero

- wNAF representation is balanced meaning it allows both negative and positive digits $\{-2^{w-1}, ..., -3, -1, 0, +1, +3, ..., 2^{w-1}\}$
- probability for occurrence of non-zero digit is $1/(w+1)$

```
W_NAF_Point_Multiplication( P , d )
        P-> input point on the curve
        d-> scalar bits
        w-> window size
        Q-> point at infinity

        Compute NAF digits
        i = 0
        while d > 0
                if d = odd
                        d(i) = d mods 2^w
                        d = d - d(i)
                else
                        d(i) = 0

                d = d / 2
                i = i + 1
        end while

        Pre-compute 3(P),5(P),7(P),..,(2^[w-1] - 1)(P)

        Compute scalar multiplication
        for i = size(d)-1 to 0
                Q = Q + Q
                if d(i) > 0
                        Q = Q + P_d(i)
                if d(i) < 0
                        Q = Q - P_d(i)
        return Q
```

Algorithm 2.8 *wNAF point multiplication method*

The use of negative digits has a very nice property, because the field addition is very fast operation to perform on the Elliptic Curves, if there are precomputed points, we only need to store the positive digits. This holds as the digits in signed representation are in the range $-2^{l-1} \leq d \leq 2^{l-1}$ so any digit greater than 2^{l-1} can

be calculated easily by taking the difference from the scalar d to shift in the range $-2^{l-1} \leq d \leq 0$. The wNAF method is listed in Algorithm 2.8. The point double and add operations in all the methods discussed depend on the secret scalar bits, which is exploited in the side channel attack discussed in the thesis.

2.3 Asymmetric Cryptography in Practice

The asymmetric cryptography algorithms are widely deployed in modern cryptographic systems and secure communication protocols. Standards such as key exchange and digital signatures are the building blocks of these protocols. For the purpose of explaining practical implications of side channel analysis on cryptographic algorithms, **Elliptic Curve Digital Signature (ECDSA)** scheme is discussed here. In short a digital signature allows two parties to verify each other. This is achieved by Signature Generation where a sender creates a signature for the message to be sent using his/her private key, and Signature Verification where the receiver of the message verifies the signature with sender's public key. The ECDSA scheme use Elliptic Curve cryptography for Signature Creation and Verification. The Algorithm 2.9 shows the steps required for generating digital signature using ECDSA [31, p. 98].

```
ECDSA_Sign( {q, FR, S, a, b, P, n, h} , m ,d )

        {q, FR, S, a, b, P, n, h}- > Domain parameters
        m-> message to sign
        d-> private key/scalar
        choose k from [1,n-1]
        (x_1, y_1) = k * P      //point multiplication
        r = x_1 mod n
        if r = 0
                choose new k and start again
        e = Hash(m)
        s = k^-1(e + d * r) mod n
        if s = 0
                choose new k and start again
        return signature pair (r,s)
```

Algorithm 2.9 *ECDSA signature*

The domain parameters [54] are required set of parameters that ensure interoperability and secure selection of curve parameters over the finite field. Field element q

is the field size of elliptic curve in finite field F_q represented by field representation FR, which is either prime F_p or binary F_{2^m}. a and b are field elements in F_q satisfying the elliptic curve equation $E(F_q)$ i.e. $y^2 = x^3 + ax + b$ for prime field. $P = (x, y)$ is the generator points over F_q and n is the order of the base point P in the finite field defined by F_q, such that $n \simeq q$ and $\#G(F_q)$ is almost prime. The cofactor is defined by $h = \#G(F_q)/n$ where as S is the seed required to generate the elliptic curve parameters randomly.

The line 5 of Algorithm 2.9 is doing a point multiplication of scalar k with base point P. A new value of k is generated for every signature ensuring that line 10 cannot be solved for the private part d. Side channel attacks, however, are still a threat, as the attacker can have access to key dependent side channel leakage and signatures, which can be used to recover the private key. One of the most popular uses for ECDSA now a days is in on-line digital currency Bitcoin [8]. Bitcoin uses ECDSA private key for client authentication and digital signatures for verification. The curve used by Bitcoin is **secp256k1**, which got a lot of attention recently for both performance improvement and secure side channel properties. The same curve is attacked using EM Side channel leakage with a detailed discussion in Chapter 5.

3. CRYPTANALYSIS OF SIDE CHANNELS

Experts from the field of cryptography often deploy techniques to analyze the security resilience of cryptographic systems by trying to break them. These techniques involve creating mathematical and information theoretic proofs to validate the security when exposed to certain attack environments also referred as cryptanalysis. These attack models rely on the fact that attacker is aware of all or some of the working details of the cryptosystem, such as the algorithm, set of input (plain text) and output (cipher text) and access to the secure system to perform operations (encryption and decryption). The idea is to recover the secret (key) by using this knowledge, hence breaking the system by decrypting all future secure communications. According to [51] typical attack models can be classified as black box, grey box and white box. In black box the adversary can see the cryptographic implementations as a black box meaning without any knowledge of internals, while having access to the input and outputs and try to analyze the statistical dependence of the plain and cipher texts e.g. In case of grey box attack model, the adversary is assumed to have some limited knowledge of the cryptographic implementation that can be exploited to break the system. They are more practical in open software implementations such as OpenSSL as the attacker has some knowledge of the implementation details of cryptographic primitives. In case of embedded systems, the security evaluation of secure implementations is modeled as grey box. Side Channel Analysis falls in this category, where an adversary in addition to some implementation knowledge also possesses some extra information observed during the operation of devices. The third case white box model assumes the attacker has full control over the cryptographic implementation and its application. The purpose of white box is to find any security vulnerability knowing complete details in order to evaluate the security features of the given cryptographic system and develop countermeasures for developed attacks. Such attacks can be used for example to target the memory of the device and extract the key from there. The work in this thesis has followed grey box approach where some knowledge of cryptographic software implementation is available to the attacker.

3.1 Side Channel Attacks

A device performing cryptographic operations is usually exploitable by an attacker using two channels. The main channel which is the intended output of the cryptographic functions and a side channel which is the unintended output due to physical behavior of the device. Where as the traditional attack tries to deploy the mathematical and statistical methods to try and find the relationship between the input and output in presence of limited or no knowledge of cryptography algorithm, side channel attacks on the other hand also utilize the physical covert channels to break the cryptographic systems. Hence these attacks are proven to be more powerful as they can even target cryptographic implementations with mathematically proven properties. Side channels are caused due to physical nature of the device and implementation details of the algorithm, hence attacks using side channel information are also termed as physical attacks or implementation attacks. Mathematically secure algorithm can still be broken in presence of side channel information with a limited resources if the implementation is weak. One of the first published works on side channel attacks by P. Kocher [37] explained how the timing difference in various key dependent operations in cryptographic primitives show strong correlations to the side channels such as power consumption of the device or memory access pattern. Broadly speaking the side channel information a.k.a. side channel leakage can be targeted in different ways to launch an attack depending on the side channel physical properties such as power consumption or electromagnetic radiations, the device under attack such as embedded processor, cryptographic primitive implementation such as Openssl RSA and the ratio between the amount of side channel leakage to the noise. For a successful attack, it is important to first identify the side channel which is susceptible to leak information that can be exploited. This require in depth analysis of the channel itself while observing the behavior of the target system. A weak correlation between the side channel and device operations for example can limit the success rate of the attack or may require additional resources in terms of cost or equipment. The cryptanalysis which falls under the category of side channel analysis is still relatively new, while there is no doubt about its effectiveness, there is still room for improvement by identifying new side channels and attack methods.

3.1.1 Previous Exploited Side Channels

Previous research has been able to exploit various side channels listed in this section, whilst it is not a conclusive list.

Timing Attacks is one of the first published side channel attacks for modern cryptography. It relies on the fact that different computations takes different time to execute on a processor. These timing variations if dependent on the secret key bits can be exploited to recover the key by identifying different operations, such the square and multiply of RSA [37] . First practical timing attack was demonstrated by D. Brumley and D. Boneh on OpenSSL RSA implementation of modular exponentiation [13]. The attack used a TLS handshake between OpenSSL client and server, while probing the time it took for the server to respond over the local network and use this timing information to retreat the private key. B. Brumley and N. Tuveri [12] demonstrated a similar attack on OpenSSL TLS handshake for Elliptic Curve implementation targeting the constant time scalar multiplication method.

Power Analysis Attacks is based on the idea of timing attacks but uses the power consumption model of the hardware device running cryptographic primitives. Introduced by Kocher et al. [38] power analysis was first utilized for smart cards and microprocessors by analyzing the power fluctuations while cryptographic algorithms are running. For example a square consumes less time and power to execute as compared square and multiply, hence the difference in power consumption can be used to recover the secret key.

Electromagnetic (EM) analysis attack is similar to power analysis but uses EM radiations caused by current flow due to fluctuations in power consumption of the device or microprocessor. These attacks does not require to tap into the physical lines but can be measured from a distance as opposed to power cosumption. The first practical attack was demonstrated by Gandolfi et al. [20] on microchips showed EM attacks to be more effective compared to the power analysis attacks. Their study was backed by performing attacks on multiple cryptographic algorithms using different probes for capturing EM radiations. More recent EM attacks on RSA and ECC [22, 26, 24] have demonstrated that these attacks are not only practical on modern devices such as smart phones, embedded systems and even personal computers but they can be very effective and cheap to carry out. More detailed discussion about previous work on EM analysis in next section.

Acoustic Attacks tries to recover secret key dependent operations using the sound emitted due to vibrations of components on a device. Fist practical attack on 4096 bit RSA was published by Genkin et al. [25] where they collected acoustic information from PCs using microphone at very low frequencies. However these

attacks are not as practical due to poor quality of acoustic signals.

3.1.2 Categorization of side channel attacks

Target Device Control refers to the amount of control the attackers have on the target device and how much influence can be made on the way the operations are executed.

- **Active attacks** allow greater control over the target device meaning an attacker can manipulate the normal behavior of the device running some cryptographic primitive.
- **Passive attacks** on the other hand provide an attacker with limited or no control over the target device. Such attacks are performed while a device operates in its normal condition, which makes it more powerful as the victim is not aware of the actual attack being carried out.

Target Device Access puts the attacks into different sets according to the level of physical access to the target device [3].

- **Invasive Attack** requires physical access to the cryptographic device such as dismantling of a microprocessor to expose individual data lines. Power measurements on these data lines can be used in classical power analysis attacks. In modern complex SoC systems these attacks are not so practical due to tightly packed interconnected components. Also very minute circuitry make it very hard and time consuming to try and find individual data or address lines and may require very expensive equipment to do so.
- **Non-Invasive Attacks** on the other hand are a powerful class of attacks that can target the cryptographic device without any physical access. Electromagnetic analysis falls in this class, as the EM signals can be captured from a distance and can still be utilized to attack. These attacks are more practical and can be used to attack real life systems such as mobile devices running cryptographic operations. Such attacks are relatively quick to carry out, with limited resources.

- **Semi-Invasive Attack** requires some physical access to the device but without dismantling to access bare metal. In fact these attacks require some physical access to enhance the capability of non-invasive attack without actually accessing the electrical circuitry. One example is to remove shielding from a mobile device to enhance EM leakage from the microprocessor.

The thesis is focused on EM side channel attack using non-invasive, passive attack model. The remainder of the chapter will discuss EM side channel analysis in depth.

3.2 Electromagnetic Side Channel Analysis

EM side channel maps the key dependent cryptographic operations to the EM leakage. This leakage is caused due to flow of current through the device. This current flow can also be measured by tapping on the power lines of the target device, however there are two main disadvantages, first this require semi-invasive or invasive approach to access power lines of interest and in case of SoC device many small power lines are packed together in layers which makes them hard to access, second often these lines are noisy due to current and voltage regulators which filters the power lines hence making it hard to find compromising leakage. This make EM side channel an attractive choice, as there is no physical access to the device. Moreover it adds spatial dimension to the measurement as the target device such as a microprocessor radiates EM signals with varying intensities depending on the location on the surface, EM measurements can be targeted on these locations increasing the chances of finding compromising side channel leakage. In order to analyze how to target EM side channel, we must understand the source and physical nature of EM radiations.

3.2.1 Source of EM Radiations

The Electromagnetic radiations are time varying electrical and magnetic fields which travel through space carrying energy between two points, hence source of transferring information. The origin of these electromagnetic fields are due to the flow of charged particles i.e. the current through a conductor. A charged particle such as an electron creates an electric field around it while the movement of these charged particles through conductor causes magnetic fields. The higher the concentration of charged

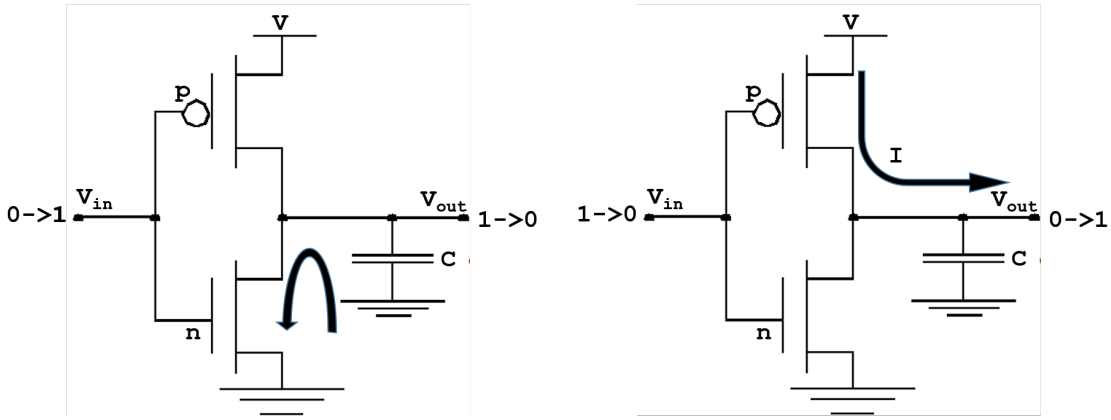

Figure 3.1 *CMOS Inverter Charging and Discharging*

particles in space the stronger the electric field, similarly the higher the velocity of these charged particles the stronger the magnetic field. The EM radiation travels as sinusoidal waves due to changing electric and magnetic fields, by sinusoidal it means they have frequency, phase and time information, which is very important from side channel perspective. In order to really understand the origin of compromising EM radiations or signals in the cryptographic device, we need to need to first understand the source of current that flows through various components of the device.

The building block of modern complex SoC are digital logic circuits composed of the CMOS inverter technology. A basic CMOS inverter is shown in Figure 3.1. There are two transistors: upper $p-transistor$ and lower $n-transistor$. This CMOS circuit acts as a logic switch, with states ON '1' and OFF '0'. For understanding purpose consider $p-transistor$ and $n-transistor$ as two different switches. These switches turn on and off based on the potential difference across V_{in}. Applying a positive voltage V at V_{in} causes the $p-transistor$ to switch off and $n-transistor$ to switch on. This sets the V_{out} to ground while discharging the capacitor C. On the other hand ground or low voltage at V_{in} causes the $p-transistor$ to switch on and $n-transistor$ to switch off generating a high voltage at V_{out}. This will make current I to flow through $p-transistor$ charging the capacitor C. This explains the rationale behind the term inverter logic circuit. Here charging and discharging of the capacitor cause a change in logic level between 1 and 0 caused due to current flow. This shows correlation between the data bits hamming distance and the current hence the resulting electromagnetic side channels are data dependent. There is another type of current I_{short} caused due to short circuit when both $p-transistor$

and $n - transistor$ switch on at the same time i.e. both cause the flow of current. This is caused due to very small switching times between transistors, and may cause them to turn on together. Even though they generate short burst of EM radiations, they are not data dependent hence are not useful from side channel perspective, in fact they can be a source of noise in the signal. There is a third kind of current source drawn by the CMOS circuit, I_{static} which is the static charge dissipated while the circuit is in equilibrium condition i.e. when the circuit is in steady-state. In steady-state there is no direct path between the input and the ground hence there is no transition in output states, in other words output state is maintained. Evidence suggests that static current is also susceptible to side channel attacks [45, 40]. Clock edge and relative bits determine the logic state, and the events occurring on the device. These states have a direct relationship with current flow as explained above. EM emissions give information about current flow and hence information about the state of the device.

Having discussed about source of EM emissions it is important to distinguish between the various type of EM radiation emitted from the device in order to identify the target EM radiations for side channel attacks.

3.2.2 Types of EM Radiations

EM side channel from a device can be viewed as a sum of multiple EM radiations emitting from different components. Take an example of a modern SoC device, it contains various components such as clock oscillators, blue-tooth modules, data lines, WiFi, processor, memory components. Hence any data dependent EM side channel leakage from such devices contains information about various other activities on the device. Therefore it is important to identify the correct EM signal for side channel attack. This requires an understanding of properties of EM radiations.

It is important to understand the concept of frequency and wavelength of an EM radiation or signal at this point. A naive mathematical relationship for an EM wave can be expressed as

$$f = c\lambda$$

where f is the frequency which is the number of cycles of the EM signal, c is the

speed of light and λ is the wavelength of the EM signal which is distance between two peaks or cycles.

From side channel point of view the frequency component of the EM signals is very important as the EM side channel contains several frequency components, and it is important to identify the side channel leaking frequency.

In order to better understand the EM side channel the resulting EM radiations can be put into two main categories [1].

- **Deliberate EM Radiations** are those emission which are part of the normal operation of the device. By normal operation it means all those components which are intended to generate EM radiations. For example a microprocessor running at clock frequency 100 MHz is supposed to generate a EM signal at 100 MHz. Similarly components such as wireless antennas, data lines and even LCD devices emit radiations on various frequencies. These radiations are normally found at higher bandwidth and frequencies and may require expensive equipment for acquiring and analyzing the side channel signals.
- **Modulated EM Radiations** are modulated signals on some carrier signals such as the CPU clock oscillator. They are caused due to transfer of electrical signals between tightly packed components on a device, where each component acts as a transmitting antenna. For example EM signals from data lines may modulate themselves on the strong signals such as main clock oscillator increasing the amplitude of the resulting EM signal. This kind of signal is amplitude modulated (AM). It may also be possible that EM signals are frequency modulated (FM) when the electric signals from data lines modulate on top of weak carrier frequencies.

While Deliberate EM Radiations may give more information at higher frequencies exploiting them may be difficult due to interfering noisy signals, in addition they are relatively weak which means it is best to capture them very close to the target device. On the other hand modulated signals can travel greater distance due to the carrier signal hence making it possible to find compromising frequencies even at a distance. Modulated signals are often observed at lower frequencies and bandwidths making it possible to exploit signals on difficult targets using cheap equipment and signal processing techniques.

3.2.3 EM Side Channel Techniques

For exploiting the EM side channel leakage, we need to analyze the properties of side channels along with the target software and a hardware implementation in order to develop the appropriate technique. There are a number of techniques mentioned in the literature which are described here.

Simple Side Channel Analysis (SCA) relies on the fact that power consumption on the device such as a microprocessor directly correlates with the EM leakage and the operations running on the processor. The EM radiations collected in SCA can directly extract the secret key related information by visual inspection [39, 55]. This is a powerful method as it requires just a single trace to break the cryptographic implementation. This also shows the weakness in the implementation of the cryptographic primitive and the simple architecture of the target hardware. However they require use of expensive equipment to capture EM radiations. In modern complex SoC architectures SCA is not very useful due to noisy side channels.

Differential Side Channel Analysis This technique was first published by Kocher et al. [39]. These attacks rely on the fact that EM radiations contain information about the data being processed on the device such as the activity on data lines or dynamic power consumption due to hamming distance or static power due to hamming weight of that data being processed. The data related information are usually in the form of very weak signals and cannot be detected by inspecting the EM side channel. Typically EM side channel contain more information related to strong signals usually due to processor activity. Collecting many similar traces and applying statistical methods can identify these weak signals. Figure 3.2 shows a simple differential attack model in a nutshell. A typical attack scenario works by first capturing the EM traces for some number of encryption operations using the same key k. The attack assumes that at some point in time t an intermediate value X is available. This value depends on small portion of the key k. The traces are split into two buckets $B_0 = X|b_{k_i^h} = 1$ and $B_1 = X|b_{k_i^h} = 0$, where $b_{k_i^h}$ is the value of the i_{th} bit of the hypothetical key k^h. This is repeated until all the bits of small key portion k are exhausted. The traces in both the buckets are averaged to a mean value Σ and a difference Δ of the averaged traces are taken.

$$\Delta(t) = \Sigma(t)B_0 - \Sigma(t)B_1$$

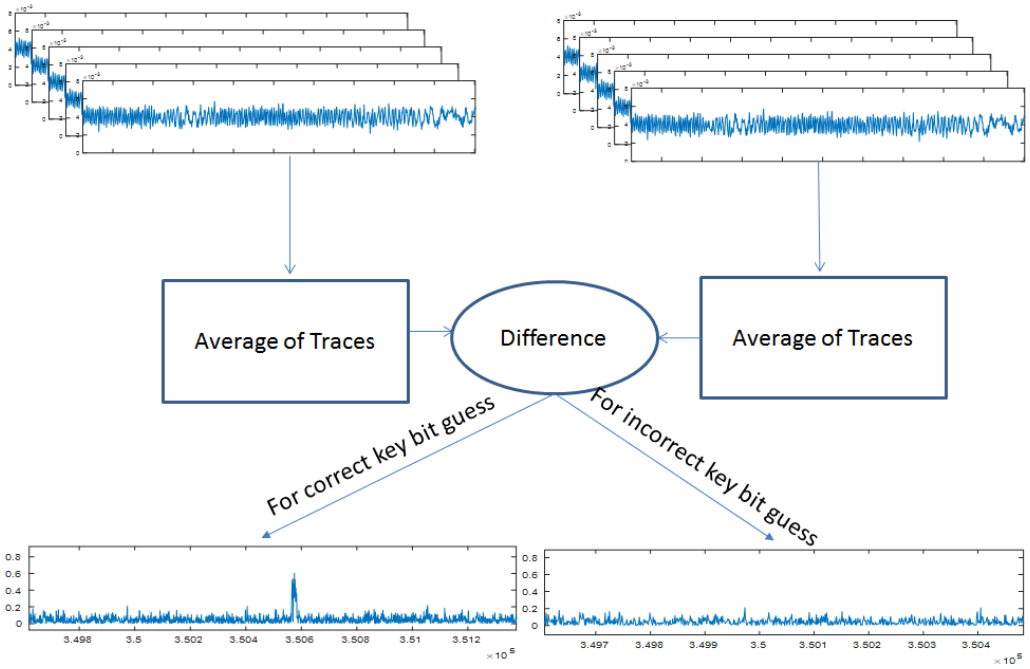

Figure 3.2 *Differential Side Channel Analysis*

For the correct key bit guess $b_{k_i^h}$ the difference is shown as huge spike in the resulting EM trace and for the incorrect key guess the resulting trace is mostly flat. This is repeated for all the key bits, where $k^h = k$ for correct key guess. This only works if the intermediate value X always fall in the same point in time in all the traces and the traces are properly aligned in time. Also sufficiently large number of traces are available.

Template Attacks Template attacks also rely on the fact that device power consumption is data dependent, however unlike differential side channel attacks, template attacks consist of two steps, template building and template matching [41]. Template attacks also consider noise as part of the signal and make no efforts to reduce it. During template building the attacker gets hold of a profiling device identical to the device under attack. Cryptographic operations are executed on the profiling device with a choice of secret key and plain text while multiple EM traces are captured for each operation. An average of the multiple EM traces for each operation is the signal part of a template and the difference between average and the signal is the noise. During template matching, if the target device performs one of the operations similar to profiling device, the EM traces from target should correlate with templates with high probability hence finding the correct operation.

3.3 Previous work on EM Side Channel Attacks

The first documented evidence about using EM radiations as covert channels dates back to 1982 when NSA released a document under the name TEMPEST program. This document provides a guideline about possible compromising EM radiations and shielding requirements for the electronic devices to protect against leaking sensitive information through EM radiations.

In 2001, the first practical attack was published by Gandolfi et al. [20] in which they demonstrated the use of EM side channel to attack three different CMOS chips. This paper demonstrated that the EM side channel is a sum of EM leakage from smaller sub components and hence can be separated to find leakage source from individual components. Moreover they found that most compromising EM signals were found around the main processing unit. They compared results with power analysis attacks and found EM attacks to be more effective. Their hand made copper-wire magnetic loop probes were able to capture EM signals close to the target device.

Agrawal et al. [2] were able to classify different kind of EM radiations, one that are intentional and other that are modulate on some carrier. They attacked DES, RSA and COMP128 implementation on smart card and SSL accelerators. They found out EM side channels are indeed composed of multiple compromising frequencies that manifest themselves as integer multiples of system clock harmonics. Their results showed that modulated EM signals can be targeted using cheap equipment and provides more useful side channel leakage than compared to signals at clock frequency.

Gebotys et al. [21] targeted a practical application of Rijndael and ECC on an embedded PDA device. Their work was based on frequency domain analysis using a spectrogram in order to capture both time and frequency information. They used frequency information to develop a differential EM analysis as the time domain signal was subjected to misalignments.

G. Kenworthy and P. Rohatgi [35] showed side channel vulnerabilities of modern mobile devices by targeting RSA and ECC implementations using cheap EM signal acquisition setup. They turned off all radio and wireless communications so signals observed in EM trace are only from processor related activity. They successfully extracted keys using near and far field antennas while using cheap radio equipment and digitizers.

Montminy et al. [44] in 2013 showed a successful differential EM attack on AES running on a 32 bit microprocessor. They used cheap software defined radios and demonstrated the EM signal acquisition at sub Nyquist sampling rate, where Nyquist sampling rate states the minimum sampling frequency must be twice the maximum frequency component of the signal to avoid signal aliasing. They sampled EM signals at 4 MSa/s and 2 MSa/s for center frequencies in the range 50 to 70 MHz. Results showed that at lower sampling rates there is more noise in the collected signal which was compensated by collecting many EM traces 5000-10000 to perform correlation based differential EM side channel attack.

G. Goller and G. Sigl [26] were able to demonstrate an attack on modern SoC based smart phones using far field antennas and cheap software defined radios. They also showed how the compromising EM signals are found at processor clock frequency. Their work was based on RSA square-and-multiply algorithm. The attack utilized averaging multiple traces of RSA modular exponentiation using the same key. They were able to emphatically prove using correlation the minimum number of traces required for averaging to successfully extract key. Moreover the number of traces required increased as the distance from the device increased due to additional environmental noise and weak EM signals. They were able to extract the key at 80cm from the target device while using 1894 EM traces.

Longo et al. [40] in 2015 conducted an in depth analysis of EM side channel on complex SoC ARM development board. The target implementations were software based OpenSSL AES implementations running on the ARM core as well as hardware AES on crypto co-processor and the ARM NEON core. Their method used a leakage detection Test Vector Leakage Assessment while scanning the SoC surface to select the EM traces. They found the compromising EM signals at frequencies around 50 MHz. They also demonstrated the effect of different clock rates on ARM core which is feature of modern SoC device. Their results suggested that significant amount of effort is required to find the EM frequencies that leak side channel information, however a leakage detection test can significantly improve finding compromising frequencies. In order to improve the signal to noise ratio of the EM signals they utilized decisioning methodology Wavelet Analysis. A template matching based attack was developed on the collected traces. The paper also demonstrated the misalignment effect of preemptive hardware and software interrupts on the EM trace during AES execution and demonstrated a template based matching of interrupted versus clean trace to identify the interrupts. However all interrupted traces were

discarded.

Genkin et al. [22] demonstrated an attack on a different class of devices such as PCs and laptops. The target software implementation was RSA and ElGamal fixed window exponentiation of GnuPG library. The attack used very low grade software defined radio and hand made co-axial cable loop antenna. The frequency modulated EM signals at frequency between 1.5 to 2 MHz were targeted around a relatively narrow bandwidth of around 100kHz. Their attack was made successfully using signal processing methods. Unlike [40] this paper dealt with interrupts by removing them from the traces. They applied a correlation based search of interrupted with clean traces on small window of the signal and then removing samples that corresponds to interrupts. The misalignment in time was also corrected using a similar technique. The key recovery was possible with a few hundred traces using a chosen cipher text for decryption.

Recently in 2016 two papers have been published with attacks on Elliptic Curve cryptography. The first paper [23] demonstrated an attack on Elliptic Curve Diffie-Hellman key exchange ECDH scheme on PCs. This attack features a very low bandwidth attack using cheap equipment. They targeted GnuPG point multiplication which uses NAF representation. In order to recover the key, a trace aggregation was performed for multiple point multiplications. The interrupts and phase shifts were compensated before averaging. The attack did not rely on extracting individual Double and Add operations, rather the Add operations were identified using a chosen cipher text. They also demonstrated how to distinguish between an Add with 1 and -1 digits since it is a signed representation. The attack was able to extract a key with 75 traces in 3.75 seconds. The second paper [24] is based on OpenSSL Elliptic Curve Digital Signature (ECDSA) algorithm. Their work demonstrated the partial extraction of Double and Add sequence from a single EM trace at very low frequencies around 200kHz. They were able to identify the location of individual Add operations in the EM trace using a blind source separation technique called Singular Spectrum Analysis. However they did not identify the individual Double operations, but they were rather estimated by calculating the distance between the Add operations. A lattice based attack was used to recover the key used for digital signature from partial information of the random scalar in each point multiplication during the signing process.

The work in this thesis has combined the understanding obtained from previous

research, while attacking a different target device and OpenSSL Elliptic Curve point multiplication. Moreover the attack demonstrated in the thesis will extract complete sequence of Double and Add operations using a single trace.

4. SIGNAL PROCESSING OF EM SIDE CHANNELS

EM radiations must be captured and processed before any side channel information can be extracted. While the capturing phase is dependent on the acquisition equipment such as Software Defined Radio (SDR), Oscilloscopes, Digitizers, the processing stage is mostly done using digital signal processing tools. A digital signal is a set of samples $x[n]$ sampled at fixed point in time such that $x[n] = y(nT)$ where T is the sampling rate for continuous signal $y(t)$. The sampling rate is dependent on the Analogue to Digital Converter (ADC) of the acquisition device. Theoretically a signal must be sampled at least twice the maximum frequency component or bandwidth represented in the signal also known as Nyquist rate. In side channel analysis the sampling rate is selected based on the maximum frequency component of the signal of interest. However use of cheap equipment such as SDR allow to capture signals at sub Nyquist rate. A lower sampling rate introduces more noise in the signal and requires additional signal processing steps to compensate for that. There are also other factors such as interfering signals in addition to the side channel signals that add noise. EM side channel signal processing is the identification and extraction of the side channel related components in the signal. In order to do that we need to understand what are these components, how are they contained in the signal and how to extract them. Perhaps one of the most important if not the most important aspect of EM side channel analysis is the signal processing and analysis step. This section gives a brief theoretical background on signals, followed by the signal processing techniques usually applied for EM side channel analysis. Although there are many different techniques developed for signal processing of EM side channel depending on the target device and attack methodology, the ones presented here are in the context of side channel attacks presented in the thesis.

4.1 Frequency Analysis of EM signals

An EM side channel consists of several different signal components also refereed to as frequency components. Mathematically these frequency components can be represented with Fourier Series

$$f(x) = \frac{1}{2}\, a_0 + \sum_{n=1}^{\infty} a_n \cos(\tfrac{n\,\pi\,x}{K}) + b_n \sin(\tfrac{n\,\pi\,x}{K})$$

where $(a_n,\, b_n)$ are expressed as

$$a_n = \tfrac{1}{K} \int_{-K}^{K} f(x) \cos(\tfrac{n\,\pi\,x}{K})\, dx, \quad b_n = \tfrac{1}{K} \int_{-K}^{K} f(x) \sin(\tfrac{n\,\pi\,x}{K})\, dx.$$

The term $\cos(\frac{n\,\pi\,x}{K})$ and $\sin(\frac{n\,\pi\,x}{K})$ are periodic signals such that $\cos(\frac{n\,\pi\,(x+T)}{K})$ =$\cos(\frac{n\,\pi\,x}{K})$ where T is the period of the signal which is a single oscillation in unit time. The frequency f which is the number of oscillations per second is the reciprocal of time period T. The symbol K is a positive integer which specifies the periodicity in the signal where as n is the multiple harmonics of the signal.

$$T = \tfrac{2K}{n}, \quad f = \tfrac{n}{2K}$$

In side channel analysis EM radiations are composed of multiple harmonics of the target frequencies, so it is possible that side channel leakage information is contained in several different frequencies. To analyze the signal in frequency domain the signal is decomposed into its frequency components. For a discrete signal $x[n]$ a Fourier Transform shows the frequency spectrum of the signal

$$X(\omega) = \sum_{n=-\infty}^{\infty} x[n]\cos(\omega n) - j\sin(\omega n)$$

where $\omega = \frac{2\pi}{2K} = \frac{\pi}{K}$ is the angular frequency.

Frequency domain analysis works well when we need to find periodic components in a signal, however in EM side channel analysis we need to find the signal behavior in particular time, in other words we need to know which frequency components are active during a specific period in time. For this a tool known as Spectrogram a.k.a. Short Term Fourier Transform is utilized which allow us to take Fourier transforms in

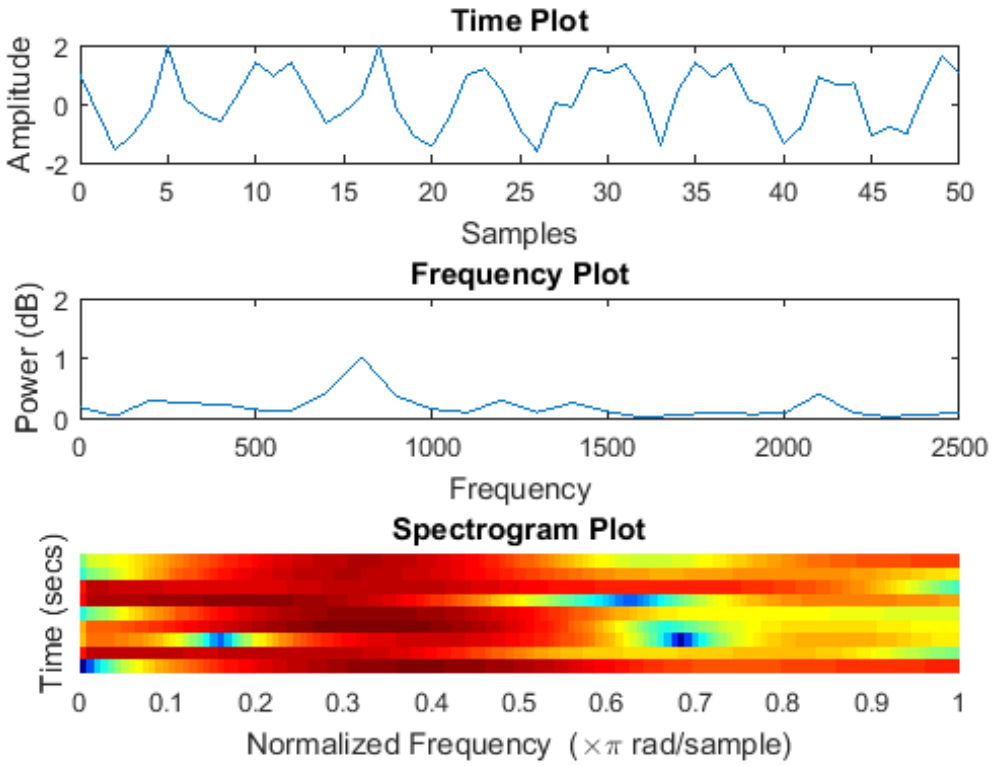

Figure 4.1 *Time plot, Frequency plot and Spectrogram plot of the same signal*

short window of time. The smaller the window size, the higher is the time resolution and lower the frequency resolution and vice versa. Mathematically it is a function of both time and frequency

$$X_s(\omega, \tau) = \sum_{n=-\infty}^{\infty} x[n]w(n-\tau)\cos(\omega n) - j\sin(\omega n)$$

where $w[n]$ is the window function and τ is the small shift in the window over the signal samples in time.

Figure 4.1 shows a frequency plot and a spectrogram of the same signal. The frequency plot is only giving overall view of each frequency component in the entire signal where as the spectrogram also shows the time information of these frequencies. This is very useful in identifying the timing information of cryptographic operations and extracting frequencies only at specific points in time.

4.2 Extraction of Target Frequencies

The term target frequencies refers to those frequencies that contain signals which can be exploited to attack cryptographic systems. For identifying side channel patterns in the signal these frequencies must be extracted from the rest of the signal using filters. Although filters can be both digital and analog, the ones described here are the digital filters since we are dealing with digital signals. In broader terms a filter is designed to either block frequencies above a certain cutoff frequency ω_c known as Low Pass Filter, pass all the frequencies above ω_c which is a High Pass Filter, or pass frequencies within a range of frequencies between ω_{c1} and ω_{c2} which is a Band Pass Filter. The choice of filter depends on the application, for example in EM side channel the first step for analysis is to apply a digital band pass filter to isolate a small stream of frequencies from a large river [40, 44]. Low pass filters in EM side channel analysis are particular useful to suppress high frequency noise and smooth the signal to identify the shape of the signal [22] or to find signals in very low frequencies [23, 24]. A digital filter falls into two main categories Finite Impulse Response (FIR) and Infinite Impulse Response (IIR). The FIR filters are weighted sum of the past inputs whereas the IIR are the weighted sums of the input as well as output which makes it infinite as there is a feedback from the output to the input. Talking in terms of side channel analysis we need a filter to respond to certain frequencies also known as Frequency Response of the filter which is mathematically expressed as:

for FIR,

$$H(e^{j\omega}) = \sum_{n=0}^{M} b_n e^{-j\omega n}$$

and for IIR,

$$H(e^{j\omega}) = \frac{\sum_{n=0}^{M} b_n e^{-j\omega n}}{\sum_{n=0}^{N} a_n e^{-j\omega n}}$$

where $e^{-j\omega n} = \cos(\omega n) - j\sin(\omega n)$ using Euler's theorem. A filter tries to approximate the Frequency response of the input signal to desired Frequency response using the set of coefficients b_n in FIR and an_n, b_n in IIR filter. The order of the filter is the number of coefficients used in the weighted sum. There are no strict rules to choose between IIR and FIR, however there are few significant differences. The FIR are

more stable as they can converge to a linear solution which is not the case in IIR due to their infinite response. However FIR needs to have more number of coefficients to get the desired frequency response hence computationally more expensive. The design of IIR is more close to analog filters where the magnitude response is limited to the frequency selection such as high pass low pass or band pass. FIR filters on the other hand can be used to design filters that have arbitrary response and are useful to extract frequencies that are not possible using normal filters. In EM side channel analysis while IIR filters can be useful in filtering out frequencies with low attenuation around some cutoff point with low order filers [44] or for removing certain amplitude offset from the signal [34], the FIR filters are widely used in EM side channel properties due to their stable design and the flexibility of optimizing filter coefficient based on the signal characteristics. For instance the class of FIR filters called matching filters can be particular useful to improve the differential and template attacks in presence of noise in the signal. [46, 30].

4.3 EM Signal Noise Treatment

Although simple filters are useful for removing unwanted frequencies, they might not remove all the noise present in the signal. Such noise may arrive from quantization error due to low sampling rate, or very high energy signal frequency interfering with low energy side channel frequency, or due to certain side channel frequency present for a very short period of time in the EM signal. There are various methods developed for treating noisy signals in EM side channel literature. The most applied method for both differential analysis and simple analysis is averaging multiple traces to increase the signal to noise ratio [26, 40, 22]. However there are two major problems with this method, first it requires multiple traces depending on the noise level they may vary between 100 to 10000. Which means each trace should be identical and must be aligned properly with each other. Secondly they only work if the secret key is fixed for each encryption or decryption operation, which means it will not work with schemes such as ECDSA where the secret scalar is generated for every new signature. This is why differential attacks which require statistical approximation of multiple traces are not practical against ECDSA schemes.

Another technique applied for denoising in side channel attacks is *Blind Source Separation* which tries to extract a specific signal component without having any knowledge about the signal components and how different components in signal are present [47]. This is a very effective technique in side channel attacks where there is

limited signal information available or in some cases only a single EM trace to work with [4, 43, 24].

The denoising method applied in this thesis is based on *Wavelet Transforms*. Like blind source separation wavelet based denoising also decomposes the signal into various components and separates the time localized frequency components from the overall signal. However there is one difference, for wavelet a prior signal knowledge can be useful and it requires only wavelet coefficients for denoising. Blind source separation on the other hand also requires a matrix for the source signals in other words we require to store signal information for approximating denoising coefficients [47]. In differential and template based EM side channel analysis wavelet denoising technique has proven to significantly improve the quality of signals where side channel features are present for a very short period of time and over shadowed by a long trend in noise [18, 40].

4.3.1 Wavelet Transform

Wavelet transforms can be seen as the extension of spectrogram analysis, however unlike spectrogram the window size in wavelet is not fixed and can be both scaled and shifted on the signal samples [28]. This is achieved by using a wavelet function known as *Mother Wavelet*. It is also worth mentioning that in contrast to frequency analysis which is a series of sin and cosine functions the wavelet can be represented by much wider selection of functions for signal analysis. Mathematically a wavelet transforms splits a function or a signal $f(t)$ into set of wavelet coefficients or basis derived from mother wavelet

$$WT(s,l) = \int_{t=-\infty}^{\infty} \psi_{s,l}^{*}(t) f(t) dt$$

where $\psi_{s,l}^{*}$ the complex conjugate of

$$\psi_{s,l}(t) = \frac{1}{\sqrt{|s|}} \psi\left(\frac{t-l}{s}\right)$$

is the *Mother Wavelet* as a function of shifting factor l and scaling factor s. The shifting factor l shifts the wavelet function over the signal in time where as the scaling factor s selects the appropriate frequencies and scales the window span. The

coefficients are calculated by continuously scaling and shifting the wavelet function over the signal which is why they are called continuous wavelet transform. *Mother Wavelet* function can be chosen from a set referred to as wavelet family.

As the EM signals are sampled at discrete intervals and we are looking for signal features in specific points in time, using continuous wavelets will give us redundant coefficients. As these coefficients are put into a transformation matrix to transform the input signal, they must be minimized for achieving computational efficiency. This brings us to the notion of *Discrete Wavelet Transforms* which applies translation and scaling at discrete samples. The Mother Wavelet Transform for Discrete Wavelet Transforms is given by [9]

$$\psi_{s,l}(t) = \frac{1}{\sqrt{j^s}}\psi\left(\frac{t-lkj^s}{j^s}\right)$$

where scaling factor $j = 2$ is the dyadic scaling and the translation factor $k = 1$ is the dyadic translation.

Form the implementation point of view Discrete Wavelet Transforms can be seen consisting of High pass H_p and Low pass L_p filters and Decimation factor D. The wavelet transformation decomposes the signal $x[n]$ into N levels where each level decomposition is expressed by

$$d_i = DH_p \text{ and } c_i = DL_p \text{ where } i = 0...N-1$$

The coefficients d_i and c_i are called *detail coefficients* and *approximation coefficients* respectively. In other words at each level of decomposition the signal frequencies are decomposed into high and low, with D decimation factor of two applied at each level. The decimation factor is used as the number of samples in decomposed signals contain half the samples of the original signal. This is a recursive decomposition where each level approximation coefficient c_i is fed as an input to the next level i.e. at level N the decomposition coefficients are $[c_N, d_N, ..., d_0]$. In order to reconstruct the signal back the same process is applied in the reverse order, i.e. the coefficients d_N and c_N are first up-sampled with a factor of two and the reconstruction high pass and low pass filter are convolved to construct c_{N-1}. This is also known as Mallat algorithm. [9, 53].

4.3.2 Wavelet Denosing

Wavelet denoising is performed on the input signal using the following steps

wavelet decomposition $\rightarrow$ *coefficient thresholding* $\rightarrow$ *wavelet reconstruction*

The core of denoising through wavelets is the selection of appropriate thresholding factor λ. This thresholding factor depends on the level of wavelet decomposition. The thresholding is applied to high frequency components which are the detail coefficients d_i as most of the noise is present in high frequencies. Wavelet thresholding can either be *Hard thresholding* or *Soft thresholding*. Hard thresholding sets all the detail coefficients to zero which are less than or equal to λ i.e. $Th(d_i) = 0$ for $d_i \leq \lambda$ and $Th(d_i) = d_i$ for $d_i > \lambda$. On the other hand soft thresholding not only sets the detail coefficients to zero but also subtracts the threshold value from any details coefficient greater than threshold i.e. $Th(d_i) = 0$ for $d_i > \lambda$ and $Th(d_i) = d_i - \lambda$ for $d_i > \lambda$. Soft thresholding also known as *wavelet shrinkage* is a more effective method for denoising as it tries to approximate all the detail coefficient to zero resulting in a greater bias towards smooth signal [19].

The effectiveness of this denoising technique is demonstrated during a practical attack on the SoC device in Chapter 5. It is also worth mentioning that unlike previous work on wavelet transformation on EM side channel analysis which is based on differential and template attacks [18, 40, 16] the attack mentioned here uses a single EM trace and wavelet denoising helped clean interfering noise as the probe collects EM signals around 3 cm from the device.

5. EM SIDE CHANNEL ANALYSIS IN PRACTICE

The EM side channel techniques are proven to be very effective against complex SoC architectures [22, 24, 26]. In oder to evaluate its effectiveness a number of experiments have been performed on cryptographic libraries running on a SoC device. The details of the experiments and the results obtained are discussed here.

5.1 Experimental Setup

This section underlines the setup for acquiring and processing the EM side channel. One of the goals for performing side channel analysis was to use low cost equipment. The tools and equipment are listed

- **EM Probe** Tekbox Near-Field magnetic loop probe H-20 with frequency response in range 500 kHz to 3 Ghz and loop size of 2 cm.
- **Amplifier** Tekbox wide-band amplifier with a flat frequency range of 3 MHz to 3 GHz and gain of 40 dB is used to amplify the weak EM signals.
- **Software Defined Radio** For acquiring and digitizing the EM radiations a software defined radio BladeRF from Nuand was used. BladeRF has a frequency range of 300 MHz to 3.8 GHz. In order to find EM leakage at lower frequencies a transverter expansion board was used to enhance the frequency range below 300 MHz. BladeRF can sample up to 40 MSps but we used a 2 MSps configuration in order to evaluate the effectiveness of the attack at low sampling rates. BladeRF has a 12-bit ADC resolution and 28 MHz bandwidth.
- **Signal Processing** Matlab R2015b was used for signal processing of EM side channel.

5.2 Target SoC Device

The target device was chosen to be Texas Instruments **AM335x Sitara** SoC based on **ARM Cortex-A8** processor embedded on **Beaglebone Black** single board system. As already established, the EM side channel radiations are due to memory and processor activity [1, 14, 15] so we need to have an insight about the architecture of the SoC. The processor supports two instruction sets. A 32-bit Integer integer instruction set and NEON floating point instruction set for Single Instruction Multiple Data (SIMD) instructions with a 128 bits register support. There is dedicated L1 cache for both integer and NEON core, however L2 cache is shared. In addition there is a dedicated shared on-chip memory. The processor has multiple power trains and clock rates which vary depending on the work load. The SoC also contains a dedicated cryptographic co-processor and graphics processing unit (GPU), where all the components are connected via Open Core Protocol (OCP) L3/L4 interconnect. Table 5.1 gives a more detailed description of the specifications. Due to this architectural complexity, the EM radiations from the device also contains information about activity from other components which act as noise in side channel measurements. This requires extraction of the target signals through use of signal processing methods as discussed later in the attack.

Table 5.1 *Sitara XAM3359AZCZ100 Processor Specifications*

Processor	ARM Cortex-A8
Supported Clock-rates	300,600,600,800,1000 (MHz)
Integer Instruction set	pipelined, 10 stages
NEON Instruction set	pipelined, 6 stage
Co-processor	cryptographic accelerator
Graphics	3D accelerator
L1 Cache	64KB
L2 Cache	256 KB
On-Chip Memory	128KB
DRAM	512 MB DDR3

The Beaglebone Black comes with a Linux Ångström based on Debian 7 "wheezy" version 3.8.13-bone70 as default Operating System. The default configurations were kept and no efforts were made what so ever to change the configurations. Which means all the default Operating System services were running. It is also a key point to mention here that the operating system switches between tasks using preemptive scheduling, this switching causes interrupts which also cause noise in the signals.

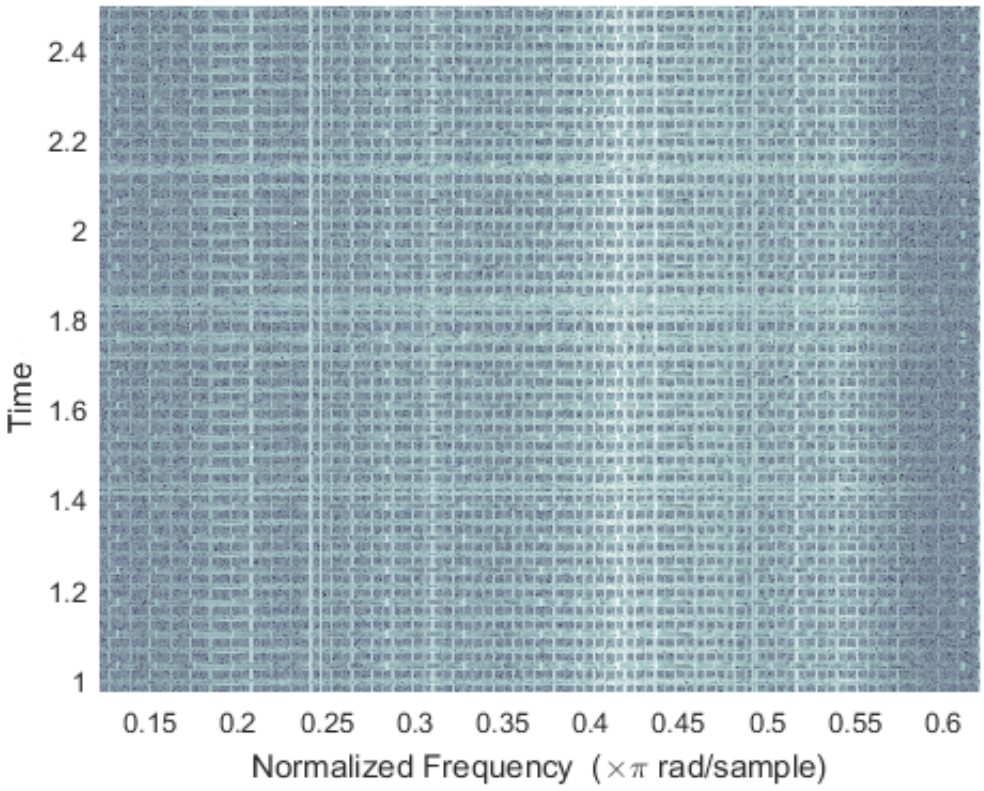

Figure 5.1 *Spectrogram showing the initial test code iterating through add, multiply and memory operations*

The Beaglebone was powered with a +5V power adapter and commands were sent using serial UART interface.

5.3 Identifying Side Channel Leakage

Before the actual attacks on the cryptographic primitives the EM side channel leakage needs to be detected which correlates to certain processor/memory activity. There are two reasons for this initial investigation, first to find the optimal probe position and secondly to access how viable is the side channel information for detecting the operations running on the processor. For this purpose a magnetic probe with a smaller loop size of 5 mm was first used to run an initial scan. Use of smaller probes requires them to be very close to the device, less then 5mm, and captures limited EM radiations only from the surface they are exposed to, which means less noise from surrounding components. For this purpose a test code was executed

which looped through 100 add instructions with a memory fetch and 100 multiply with memory load instructions followed by a sleep for 0.05 seconds. Figure 5.1 shows the spectrum of this test code in frequencies around 10 MHz.

By moving the probe over the SoC surface, one area was identified which yielded the best Signal to Noise ratio. This showed a clear processor and DRAM related side channel leakage was present. The only purpose of this step was to get an idea how the device leaks side channel related information. However in the later experiments while attacking the actual cryptographic implementation, the probe with larger loop size of 2 cm was used, which allowed to capture EM signals from a distance but also introduced noise.

5.4 Attack on Elliptic Curve Point Multiplication

The attack presented utilized a single trace to recover the point Double and Add operations in ECC point multiplication. For this purpose OpenSSL was used which is a popular open source cryptography library implemented in C programming language. The library is widely utilized for implementing Transport Layer Security (TLS) and Secure Socket Layer (SSL) communication protocols. A number of ECC curves are supported by OpenSSL, out of which **secp256k1** was chosen for the attack. This curve has recently gained popularity since its use in Bitcoin digital currency. It is defined over prime field F_p with the following parameters which are fixed

- 256-bit prime field representation FR: FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFFFFFF FFFF FFFF FFFFFFFF FFFE FFFF FC2F
- 256-bit x-coordinate of generator point P in F_p: 79BE 667E F9DC BBAC 55A0 6295 CE87 0B07 029B FCDB 2DCE 28D9 59F2 815B 16F8 1798
- 256-bit y-coordinate of generator point P in F_p: 483A DA77 26A3 C465 5DA4 FBFC 0E11 08A8 FD17 B448 A685 5419 9C47 D08F FB10 D4B8
- 256-bit order n: FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFE BAAE DCE6 AF48 A03B BFD2 5E8C D036 4141
- cofactor h: 01

OpenSSL **secp256k1** is based on Gallant-Lambert-Vanstone (GLV) curves [10].

```
r_is_at_infinity = 1;

    for (k = max_len - 1; k >= 0; k--) {
        if (!r_is_at_infinity) {
            if (!EC_POINT_dbl(group, r, r, ctx))
                goto err;
        }

        for (i = 0; i < totalnum; i++) {
            if (wNAF_len[i] > (size_t)k) {
                int digit = wNAF[i][k];
                int is_neg;

                if (digit) {
                    is_neg = digit < 0;

                    if (is_neg)
                        digit = -digit;

                    if (is_neg != r_is_inverted) {
                        if (!r_is_at_infinity) {
                            if (!EC_POINT_invert(group, r, ctx))
                                goto err;
                        }
                        r_is_inverted = !r_is_inverted;
                    }

                    /* digit > 0 */

                    if (r_is_at_infinity) {
                        if (!EC_POINT_copy(r, val_sub[i][digit >>
                            ↪ 1]))
                            goto err;
                        r_is_at_infinity = 0;
                    } else {
                        if (!EC_POINT_add
                            (group, r, r, val_sub[i][digit >> 1],
                                ↪ ctx))
                            goto err;
                    }
                }
            }
        }
    }
```

Algorithm 5.1 *OpenSSL point multiplication ec_ mult.c*

The point multiplication on this curve is implemented using $wNAF$ representation as described in Algorithm 2.8, however OpenSSL also uses the so called "interleaved scalar" by point multiplication [10] for better efficiency. This method decomposes the scalar into two parts and performs interleaved point multiplications on them. However the scalar interleaving only works if pre computation of point multiplications are enabled using certain conditions in the code which were disabled by default in our case. We used a C code harness that calls OpenSSL point multiplication function pointer **EC_POINT_mul** running in a loop. This function uses a random scalar k and a point Q on the curve. The C harness also contains command line parameters to loop through individual Add operations **EC_POINT_add** and double operations **EC_POINT_dbl** to find EM leakage corresponding to these operations rather than the complete point multiplication leakage. These function pointers invoke the OpenSSL point multiplication method **ec_mult.c** listed in Algorithm 5.1. By looking at the code we can find two loops at line 3 and 9. The first loop iterates over each digit of the scalar k. There is always one point double operation at the start as variable $r\_is\_at\_infinity$ equals 1 in only the first iteration. The second for loop either executes once if no interleaving is applied and twice if the scalar interleaving is applied. From the perspective of EM side channel we are specifically interested in the two operations **EC_POINT_dbl** and **EC_POINT_add** functions. The first loop is executed for every digit calling the **EC_POINT_dbl** function. For a non-zero digit we have either positive digits or negative digits of scalar. A point addition is performed for non zero digits in the inner for loop.

5.4.1 Attack Methodology

The software defined radio captured the EM traces at 2MSa/s while EC point multiplication executes in a loop. We applied a frequency sweep across the spectrum between 1 MHz and 100 MHz. We suspected the side channel information to be in the lower frequencies as the point multiplication is a heavy operation and takes millions of clock cycles and secondly the side channel signals are modulated to lower frequencies due to signal coupling. Because the software defined radio bandwidth was limited to 28MHz, a window based sweep of the spectrum was done by adjusting the center frequency. Figure 5.2 shows the spectrogram plot of the initial frequency sweep from 0-50 MHz sweep. There are a number of frequencies identified in the spectrum with higher signal energy close to 50MHz, so a pre-processing stage was

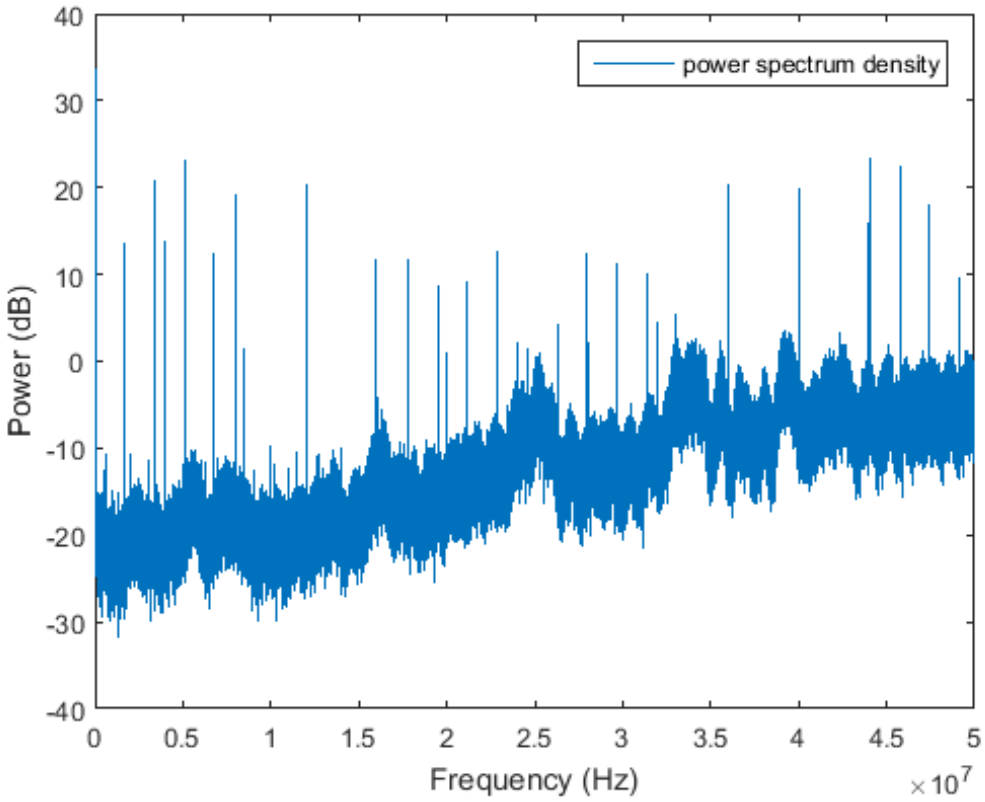

Figure 5.2 *Power Spectrum Density of 0 to 50 MHz while executing point multiplication operation*

applied to extract the frequencies that may yield side channel information. This was done by modifying the code executed on SoC to alternate between multiple iterations of Double followed by multiple iteration of Add operations. This clearly identified two distinct patterns in the spectrogram as shown in Figure 5.3. One of these frequencies can be selected as the target frequency to extract the Double and Add operations. This was a tedious step as it involved selecting each frequency of interest and analyzing it for a possible pattern that exhibits the sequence of operations in point multiplication.

Extracting the Target Frequency After identifying the frequency we applied a band pass filter around it. This was done by selecting the desired center frequency for the SDR, in our case 40 Mhz, and applying a digital FIR filter to the raw signal. By looking at the filtered trace Figure 5.4 we can see some repeating spikes in the signal.

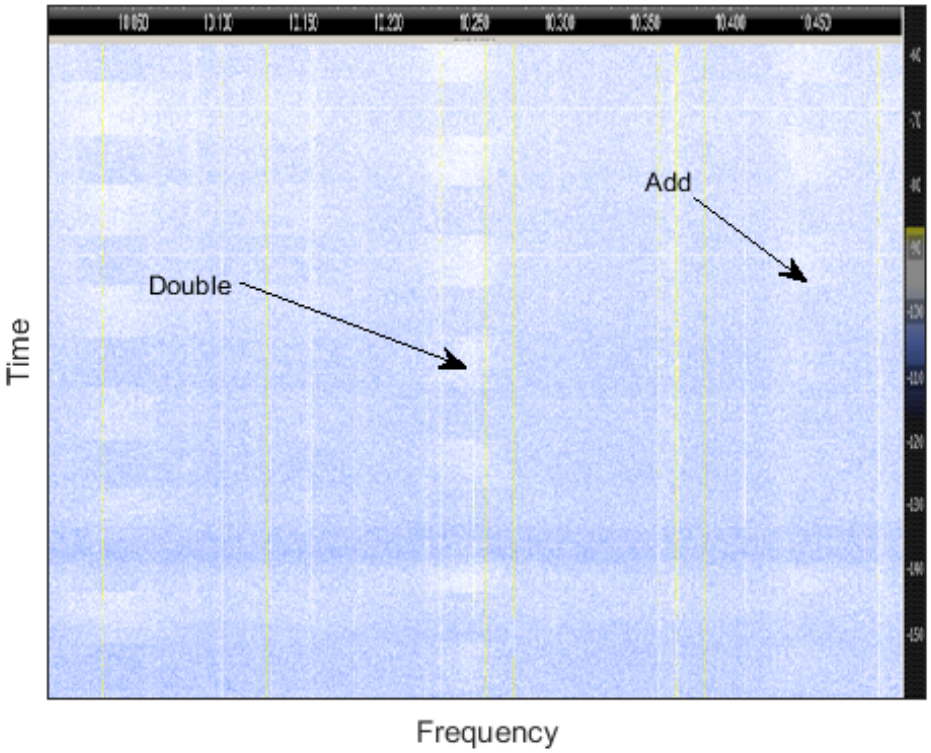

Figure 5.3 *Spectrogram showing multiple invocations of point multiplication and addition around 10 MHz frequency*

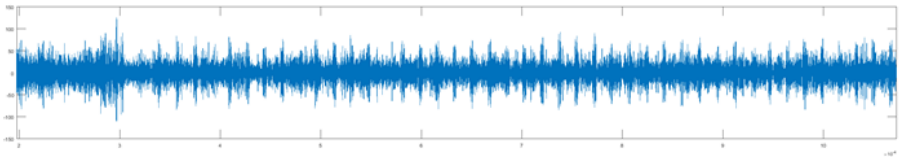

Figure 5.4 *Filtered Signal*

Demodulation The filtered trace was demodulated by applying a discrete Hilbert transform and taking its absolute value. The resulting signal was passed through Savitzky-Golay filter with a polynomial of order 3 and window size 11 which applies the linear least square curve fitting. This helped in suppressing high frequency noise to improve the signal to noise ratio. In Figure 5.5 spikes are clearly visible, however to extract the Double and Add operations we required to extract these peaks. Peak extraction on this signal gave several false positives as there are noisy peaks in the signal and even a carefully selected thresholding did not help. To mitigate this

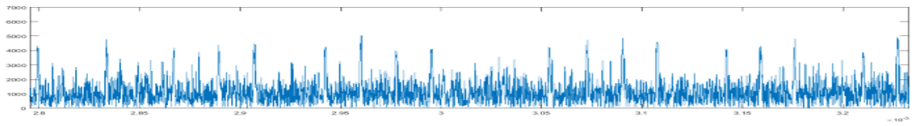

Figure 5.5 *Demodulated Signal*

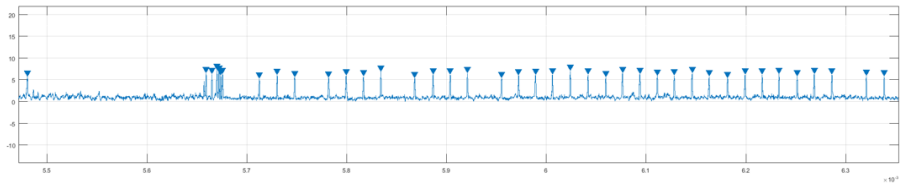

Figure 5.6 *Successfully extracted peaks after denoising*

problem a signal denoising was performed so peaks corresponding to the interesting frequencies can be isolated.

Noise Removal We applied Discrete Wavelet Transform based denoising as discussed in Chapter 4. In our demodulated trace the signal of interest is overlapped by other interfering signals and low frequency noise. So we decomposed the signal using Haar wavelet into 5 levels. This resulted in a set of details coefficients in higher frequency bands for the signal of interest. All the coefficients below a certain threshold were set to zero. This effectively removed the coefficients that correspond to noise. Moreover we applied soft thresholding as suggested by [19]. The soft thresholding not only sets the detail coefficients to zero but also approximates the retained detail coefficients towards zero in order to minimize any residual noise. The resulting signal clearly shows identifiable spikes while all other noisy spikes are removed.

Peak Extraction This step extracted the peaks that identify the sequence of Double and Add operations. This was done by calculating the optimal peak prominence as a threshold to detect all the peaks that correlates to the leaked signals. Due to the denoising performed in the previous step, peak extraction successfully extracted the target peaks with zero error. Figure 5.6 shows denoised signal with extracted peaks.

Double and Add Sequence Detection Although these peaks gave a clear indication about each Double and Add operations, we still need to figure out where each elliptic curve point multiplication starts and ends. In side channel analysis we call this a trigger condition that identifies the start and end of the cryptographic

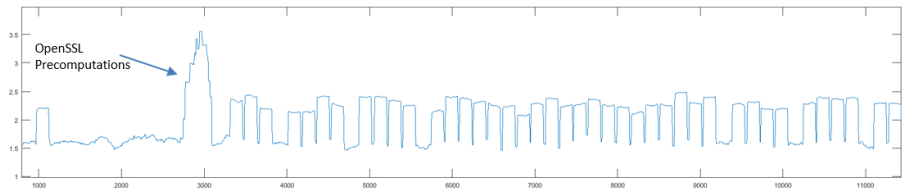

Figure 5.7 *After Envelop Detection showing OpenSSL pre-computations at the start*

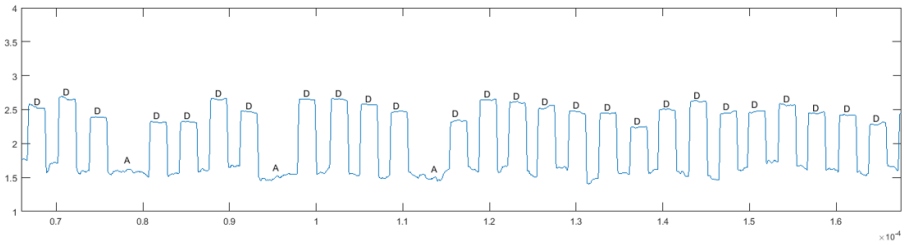

Figure 5.8 *Extracted Double and Add sequence*

operations to extract the key dependent information. In our case the SDR was continuously capturing EM signals so having some interrupt based trigger was not an option, although it was possible to use a GPIO of the SoC to send interrupt signals before start and end of point multiplication. However that would not serve the purpose of using SDR for non-invasive attack model. However by investigating the signal we found a specific patter at the start and end of each point multiplication operation. OpenSSL performs an internal pre-computation stage before each scalar multiplication operation and this might explain the reason for this signal behavior. Figure 5.7 shows the start of the point multiplication, these peaks are not part of actual point multiplication but the Double and Add operations in the pre-computation of OpenSSL. In order to recover the scalar dependent operations we had to separate these peaks from the rest of the signal and also use this information to detect the start of the signal. This was achieved by creating an envelope on the signal in a way that peaks that were close together approximated to a single peak while other peaks stayed separate. The method used for envelope detection was root mean square where we found the window size of 10 samples gave the best results. The resulting signal showed a large enveloped peak at the start of each point multiplication and smaller peaks to identify double and operations which is shown in Figure 5.7. The final signal fully extracted the Double and Add operations. The peaks closer together showing Double operations and peaks further apart are Double and Add

operations.

Interrupt Detection The interrupts caused by the OS and hardware introduce noise to the signals, so we need to identify them to successfully extract key dependent operations. The interrupt in our signal was clearly identified as a huge gap in the signal, as contrary to [40, 24] where authors suggested they appear as spikes. The reason could be the frequencies we were using, the signal processing steps involved and the characteristics of the target device. In order to identify the interrupts we did a correlation analysis of a reference trace with interrupted trace as suggested by [23]. The correlation was performed by first identifying the beginning of signal, then applying small window of reference trace with target trace, and repeating this over the entire signal. If an interrupt was detected that part of the signal was removed by removing the samples. If this step corrupted the signal it was dropped and next signal was taken.

Extracting Scalar Bits The attack successfully extracted the complete Double and Add operations with a success rate of over 95%. These Double and Add operations however do not reveal the secret scalar k involved in point multiplication kP. Taking the use case of point multiplication in the ECDSA signature generation, the scalars k act as nonces i.e. they are randomly generated for each scalar multiplication kP in a signature hence a different sequence of Double and Add in each processed signal. As the scalar is represented in $wNAF$ form where each non-zero digit in scalar belongs to the set $\{\pm 1, \pm 3, \pm 5, \pm 7\}$. So mapping the Double and Add sequence to the key is not straightforward as we need to know which digit is represented from the set for each point addition. The key extraction can be done using Lattice based attack on the sequence of Double and Add operations as demonstrated by Brumley and Hakala [11] and Yuval et al. [6]. The attack starts by utilizing the information that last Double and Add sequence is always represented as a 1 in the LSB of scalar and works itself up by collecting successive signatures of Double and Add sequences. The Table 5.2 shows four possible sequences of Double and Add operations and the probability of occurrence of Add operations in LSB of the scalar. These probabilities follow a Poisson distribution as there is a exponential decay in the number of unknown scalar digits. Collecting a few hundred EM traces of Double and Add sequences can extract the key using Lattice attack. Table 5.3 relates the number of Double and Add sequences to probability of success and time required for extracting the scalar using lattice attack on **secp256k1** curve [6].

Table 5.2 *Sequence of Double and Add in LSB for Lattice attacks*

Double and Add sequence	LSB of scalar	Probability of Add
...DA	1	1/2
...DAA	10	1/4
...DAAA	100	1/8
...DAAAA	1000	1/16

Table 5.3 *Number of sequences and probability of successful scalar bits extraction using Lattice attacks on secp256k1*

Number of Sequences	Probability of Success	Time in Seconds
240	0.5	2.68
300	53.0	13.54
480	87.0	11.55
520	96.0	10.50

5.5 Attack on RSA Modular Exponentiation

In order to attack RSA the **left-to-right square and multiply algorithm** of GMP library was used with a 4096 bit key size. The method is straight forward and finding the Square and Multiply operation directly maps to the secret key bits. Utilizing the same method as discussed above, the signal was extracted around 10.5 MHz, however it was possible to find similar signals in other frequencies. This attack also utilized a single trace to extract the key related operations. After applying FIR band pass filter, the trace was corrected by removing noise using wavelet transforms. Figure 5.9 shows square and multiply operations of fixed key. Here the larger peak is the Square and Multiply where as smaller peaks are Square operations corresponding to a 1 and 0 in the key respectively.

Figure 5.10 shows a processed trace from fixed window algorithm of OpenSSL with 2048 bit key and window size 5. The dips here signify the Square and Multiply operation, where each Multiply operation is followed by 5 Square operations. The peak extraction however also had some false positives even after denoising, one reason could be that there is no significant difference in processing times and power consumption of Square and Square and Multiply operations on this SoC device. However averaging techniques such as the ones applied here [26, 22] can improve the results. The work in this thesis focused on single trace EM side channel analysis using cheap equipment to prove its effectiveness.

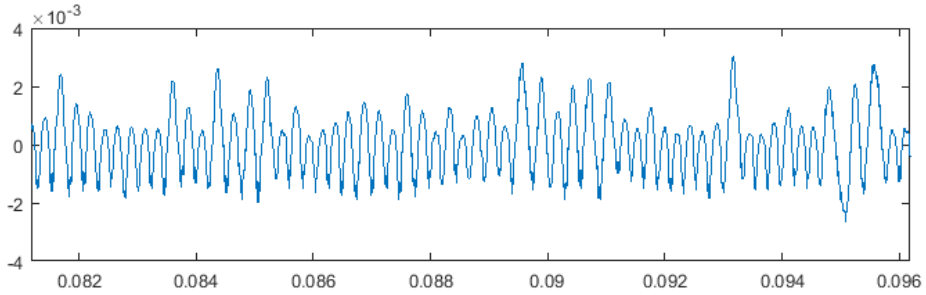

Figure 5.9 *Extracted Square and Multiply operations with low peak as Square and high peak Square and Multiply*

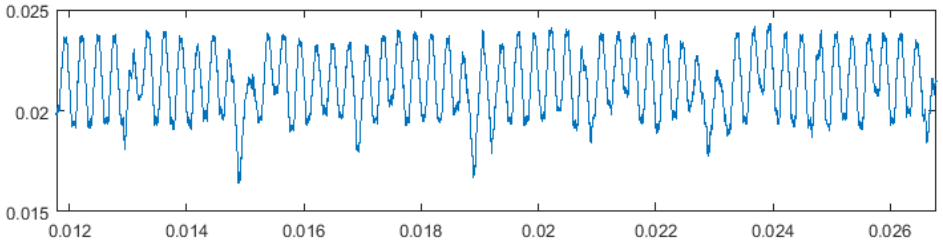

Figure 5.10 *Extracted Square and Multiply operations with a window size 5*

The results obtained for both RSA and ECC suggest strong evidence that cryptographic implementations can leak secrete key dependent information irrespective of the complex architecture of devices. Cryptographers must come up with implementations that have no correlations between secret keys and cryptographic operations. The next chapter discuss some possible countermeasures that can mitigate this side channel vulnerability, however implementing these countermeasures is out of the scope of this thesis.

6. SIDE CHANNEL COUNTERMEASURES

EM side channel is a proven technique to attack cryptographic systems, therefore engineers must come up with implementations to thwart this vulnerability by developing side channel countermeasures. These countermeasures try to minimize any information that relates side channel leakage to the cryptographic algorithm implementation details such as conditional branches, memory access, order of instruction execution on the processor which can reveal the secret. From the perspective of EM side channel this can be achieved in two different ways: either by minimizing the exploitable EM leakage from the device or by developing side channel resistant software implementations. The following sections discuss some side channel countermeasures that can reduce the secret dependent EM leakage exploited in the thesis. Although these countermeasures are discussed for elliptic curve point multiplication, the same analogy can be applied to modular exponentiations.

The EM side channel attacks using just one trace most often utilize the information from timing variations on secret key dependent operations similar to the attacks discussed in the thesis. These timing variations can come from the order and processing times of instruction execution [39] and/or memory access patterns [12]. For defeating the side channel leakage caused due to difference in operations such as elliptic curve point Double and Add from point Double, the straight forward solution is to utilize a method that executes both the Double and Add operations on each key dependent conditional branch of the executed code. An efficient point multiplication method with constant time operations is Montgomery Ladder [32] listed in Algorithm 6.1. We can see that during each conditional branch both the point Addition and point Double is performed on the registers. Furthermore it is shown [32] that the field operations on the curve can be performed only on the x-coordinate of the point, and that y-coordinate can be calculated directly from the x-coordinate, this property makes this method more efficient.

```
Montgomery_Ladder( P , d )
        P-> input point on the curve
        d-> scalar bits
        R_0 = P
        R_1 = P + P
        for i = size(d)-2 to 0
                if d(i)==1
                        R_0 = R_0 + R_1
                        R_1 = R_1 + R_1

                else
                        R_1 = R_0 + R_1
                        R_0 = R_0 + R_0

        return R_0
```

Algorithm 6.1 *Montgomery Ladder point multiplication method*

Recently S. Gueron and V. Krasnov [29] purposed a Montgomery friendly optimized implementation of point multiplication for OpenSSL 256-bit prime curves with good side channel resistance properties. The point multiplication is computed by first converting the point coordinates to Montgomery domain and then converting the result of operations back to residue domain. This method can further achieve performance improvement for 256 bit prime curves by slicing the 256-bits scalar into 4 words of 64-bits each for 64-bit architectures. In order to implement scalar multiplication dP a window based method with Booth encoding using a window size of 7 is used, implies a total of $\lceil \frac{256}{7} \rceil$ tables of precomputed values for field operations on x and y-coordinates. For side channel protection the tables are populated sequentially for each window, and a mask is used which only keeps the required values with all other table entries set to zero. This step ensures there is no memory access pattern depending on the secret scalar, where as the Montgomery point multiplication ensures constant time implementation.

Another countermeasure found in the literature for OpenSSL ECC by B. Brumley [10] uses Regular NAF scalar encoding and software multiplexing. The Regular NAF encoding ensures that the scalar uses a fixed number of zero values between non-zero digits. Software multiplexing is a countermeasure that removes memory access patterns by iterating through the entire table in a fixed order and extracting only required values using a bitwise operation with a mask. This countermeasure ensures no secret scalar dependent timing information is given away.

It is also possible to reduce the amount of EM radiations from the device by applying shielding methods or to redesign the CMOS circuits. On circuit design level, it is required to remove the crosstalk between the external power of the CMOS chip and the internal power of the logic gate. One proposed countermeasure is to use decoupling capacitors in interleaving fashion [52, 49] where one capacitor charges while the other discharges. The decoupling can reduce the side channel information in power analysis, but may still leak EM radiations as these capacitors are used on the chip level and a current flow through the chip wires can still induce EM radiations. A more recent countermeasure [27] suggests use of dedicated decoupling cells for each logic gate, by using an integrated capacitor to remove cross talk between the internal and external power supply of the logic gate. The circuit design ensures that each cell gets its power from a nearby buffering capacitor which makes sure that the current is kept within the cell, which significantly reduces the EM radiations as there is no current flow through long wires. However, this claim is not proved with experimental results and it may be possible to use signal processing methods to enhance the information contained in weak EM signals.

Implementing security solutions on the hardware level can help mitigate some of the problems found on software implementations. Hardware security makes use of hardware-assisted technology with the aid of dedicated processor cores and hardware modules to encapsulate the security critical operations, while the only access to the external environment is passing secure information: for example, a bank transaction authentication token. With an increase in the use of mobile devices, vendors are now moving towards hardware based solutions due to their superior security features and resistance against malware. Examples of such solutuons are Trusted Execution Environments (TEE) and Trusted Platform Modules (TPM).

7. CONCLUSION

This thesis provides evidence about the effectiveness of EM side channel attacks on well known OpenSSL implementations using just one trace. The results proves that even mathematically strong and carefully written cryptographic implementations can have side channel vulnerabilities, hence these implementations must be tested for secure side channel properties. Moreover EM side channels can be very effective due to its non invasive nature and ability to break cryptographic implementation using cheap equipment and signal processing methods. Therefore countermeasures must be developed and tested against these side channels.

There is still no strong evidence to support the correlation between EM side channels and micro-architecture design, such as how manipulating cache would effect the EM radiations. This is particularly important when developing countermeasures. An extension of this attack could be to combine the cache-timing with EM side channels for developing hybrid attacks adding another dimension to the attack. This can also be useful in studying the effects of EM radiations on loading the program to memory. Moreover the countermeasures that address the memory addressing may not be sufficient for EM side channels, as they can still leak information present in the processor registers.

Reverse engineering hardware encryption modules such as Trusted Platform Modules (TPM) using EM side channel templates is a very interesting future work. TPMs are vendor specific solutions and for security reasons there are no specifications available on how the cryptographic algorithms are embedded. Reverse engineering using EM/Power Side channel is one solution. It requires carrying out detailed analysis on these side channels and developing templates on each stage of algorithm execution. These side channel templates can provide some insight into the working of these TPMs and can be used to evaluate the security features.

EM side channel analysis is a promising area to target the Trusted Execution Environments (TEE). TEE are secure sandboxes which effectively creates a defined

boundary between secured code and unsecured code, providing confidentiality and integrity at the hardware level. As of this date, there are no studies to the best of knowledge, to prove any side channel attacks on TEE. Even though these trusted executions are isolated they still share the same device resources such as microprocessor cores and memory resulting in EM side channel properties with similar characteristics. Most of the devices now a days are equipped with TEE solutions such as ARM TrustZone and AMD Platform Security Processor as they provide enhanced security. However, these devices still radiate EM leakage and must be investigated for any side channel vulnerabilities.

BIBLIOGRAPHY

[1] D. Agrawal, B. Archambeault, J. R. Rao, and P. Rohatgi, "The EM side-channel(s)," in *Cryptographic Hardware and Embedded Systems - CHES 2002, 4th International Workshop, Redwood Shores, CA, USA, August 13-15, 2002, Revised Papers*, ser. Lecture Notes in Computer Science, B. S. K. Jr., Ç. K. Koç, and C. Paar, Eds., vol. 2523. Springer, 2002, pp. 29–45. [Online]. Available: http://dx.doi.org/10.1007/3-540-36400-5_4

[2] D. Agrawal, J. R. Rao, and P. Rohatgi, "Multi-channel attacks," in *Cryptographic Hardware and Embedded Systems - CHES 2003, 5th International Workshop, Cologne, Germany, September 8-10, 2003, Proceedings*, ser. Lecture Notes in Computer Science, C. D. Walter, Ç. K. Koç, and C. Paar, Eds., vol. 2779. Springer, 2003, pp. 2–16. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-45238-6_2

[3] R. Anderson, M. Bond, J. Clulow, and S. Skorobogatov, "Cryptographic processors-a survey," *Proceedings of the IEEE*, vol. 94, no. 2, pp. 357–369, Feb 2006.

[4] C. Archambeau, E. Peeters, F. Standaert, and J. Quisquater, "Template attacks in principal subspaces," in *Cryptographic Hardware and Embedded Systems - CHES 2006, 8th International Workshop, Yokohama, Japan, October 10-13, 2006, Proceedings*, ser. Lecture Notes in Computer Science, L. Goubin and M. Matsui, Eds., vol. 4249. Springer, 2006, pp. 1–14. [Online]. Available: http://dx.doi.org/10.1007/11894063_1

[5] L. Batina and M. Robshaw, Eds., *Cryptographic Hardware and Embedded Systems - CHES 2014 - 16th International Workshop, Busan, South Korea, September 23-26, 2014. Proceedings*, ser. Lecture Notes in Computer Science, vol. 8731. Springer, 2014. [Online]. Available: http://dx.doi.org/10.1007/978-3-662-44709-3

[6] N. Benger, J. van de Pol, N. P. Smart, and Y. Yarom, ""ooh aah... just a little bit" : A small amount of side channel can go a long way," in *Cryptographic Hardware and Embedded Systems - CHES 2014 - 16th International Workshop, Busan, South Korea, September 23-26, 2014. Proceedings*, ser. Lecture Notes in Computer Science, L. Batina and

M. Robshaw, Eds., vol. 8731. Springer, 2014, pp. 75–92. [Online]. Available: http://dx.doi.org/10.1007/978-3-662-44709-3_5

[7] I. Blake, G. Seroussi, and N. Smart, *Elliptic Curves in Cryptography*, 1st ed. Cambridge, UK: Cambridge University Press, 1999.

[8] J. W. Bos, J. A. Halderman, N. Heninger, J. Moore, M. Naehrig, and E. Wustrow, "Elliptic curve cryptography in practice," in *Financial Cryptography and Data Security - 18th International Conference, FC 2014, Christ Church, Barbados, March 3-7, 2014, Revised Selected Papers*, ser. Lecture Notes in Computer Science, N. Christin and R. Safavi-Naini, Eds., vol. 8437. Springer, 2014, pp. 157–175. [Online]. Available: http://dx.doi.org/10.1007/978-3-662-45472-5_11

[9] L. M. Bruce, C. H. Koger, and J. Li, "Dimensionality reduction of hyperspectral data using discrete wavelet transform feature extraction," *IEEE Trans. Geoscience and Remote Sensing*, vol. 40, no. 10, pp. 2331–2338, 2002. [Online]. Available: http://dx.doi.org/10.1109/TGRS.2002.804721

[10] B. B. Brumley, "Faster software for fast endomorphisms," *IACR Cryptology ePrint Archive*, vol. 2015, p. 36, 2015. [Online]. Available: http://eprint.iacr.org/2015/036

[11] B. B. Brumley and R. M. Hakala, "Cache-timing template attacks," in *Advances in Cryptology - ASIACRYPT 2009, 15th International Conference on the Theory and Application of Cryptology and Information Security, Tokyo, Japan, December 6-10, 2009. Proceedings*, ser. Lecture Notes in Computer Science, M. Matsui, Ed., vol. 5912. Springer, 2009, pp. 667–684. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-10366-7_39

[12] B. B. Brumley and N. Tuveri, "Remote timing attacks are still practical," in *Computer Security - ESORICS 2011 - 16th European Symposium on Research in Computer Security, Leuven, Belgium, September 12-14, 2011. Proceedings*, ser. Lecture Notes in Computer Science, V. Atluri and C. Díaz, Eds., vol. 6879. Springer, 2011, pp. 355–371. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-23822-2_20

[13] D. Brumley and D. Boneh, "Remote timing attacks are practical," *Computer Networks*, vol. 48, no. 5, pp. 701–716, 2005. [Online]. Available: http://dx.doi.org/10.1016/j.comnet.2005.01.010

[14] R. Callan, A. G. Zajic, and M. Prvulovic, "A practical methodology for measuring the side-channel signal available to the attacker for instruction-level events," in *47th Annual IEEE/ACM International Symposium on Microarchitecture, MICRO 2014, Cambridge, United Kingdom, December 13-17, 2014*. IEEE, 2014, pp. 242–254. [Online]. Available: http://dx.doi.org/10.1109/MICRO.2014.39

[15] ——, "FASE: finding amplitude-modulated side-channel emanations," in *Proceedings of the 42nd Annual International Symposium on Computer Architecture, Portland, OR, USA, June 13-17, 2015*, D. T. Marr and D. H. Albonesi, Eds. ACM, 2015, pp. 592–603. [Online]. Available: http://doi.acm.org/10.1145/2749469.2750394

[16] X. Charvet and H. Pelletier, "Improving the dpa attack using wavelet transform," 2005. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.473.6813

[17] H. Cohen, A. Miyaji, and T. Ono, "Efficient elliptic curve exponentiation using mixed coordinates," in *Advances in Cryptology - ASIACRYPT '98, International Conference on the Theory and Applications of Cryptology and Information Security, Beijing, China, October 18-22, 1998, Proceedings*, ser. Lecture Notes in Computer Science, K. Ohta and D. Pei, Eds., vol. 1514. Springer, 1998, pp. 51–65. [Online]. Available: http://dx.doi.org/10.1007/3-540-49649-1_6

[18] N. Debande, Y. Souissi, M. A. Elaabid, S. Guilley, and J. Danger, "Wavelet transform based pre-processing for side channel analysis," in *45th Annual IEEE/ACM International Symposium on Microarchitecture, MICRO 2012, Workshops Proceedings, Vancouver, BC, Canada, December 1-5, 2012*. IEEE Computer Society, 2012, pp. 32–38. [Online]. Available: http://dx.doi.org/10.1109/MICROW.2012.15

[19] D. L. Donoho, "De-noising by soft-thresholding," *IEEE Trans. Information Theory*, vol. 41, no. 3, pp. 613–627, 1995. [Online]. Available: http://dx.doi.org/10.1109/18.382009

[20] K. Gandolfi, C. Mourtel, and F. Olivier, "Electromagnetic analysis: Concrete results," in *Cryptographic Hardware and Embedded Systems - CHES 2001, Third International Workshop, Paris, France, May 14-16, 2001, Proceedings*,

ser. Lecture Notes in Computer Science, Ç. K. Koç, D. Naccache, and C. Paar, Eds., vol. 2162, no. Generators. Springer, 2001, pp. 251–261. [Online]. Available: http://dx.doi.org/10.1007/3-540-44709-1_21

[21] C. H. Gebotys, S. Ho, and C. C. Tiu, "EM analysis of rijndael and ECC on a wireless java-based PDA," in *Cryptographic Hardware and Embedded Systems - CHES 2005, 7th International Workshop, Edinburgh, UK, August 29 - September 1, 2005, Proceedings*, ser. Lecture Notes in Computer Science, J. R. Rao and B. Sunar, Eds., vol. 3659. Springer, 2005, pp. 250–264. [Online]. Available: http://dx.doi.org/10.1007/11545262_19

[22] D. Genkin, L. Pachmanov, I. Pipman, and E. Tromer, "Stealing keys from pcs using a radio: Cheap electromagnetic attacks on windowed exponentiation," *IACR Cryptology ePrint Archive*, vol. 2015, p. 170, 2015. [Online]. Available: http://eprint.iacr.org/2015/170

[23] ——, "ECDH key-extraction via low-bandwidth electromagnetic attacks on pcs," *IACR Cryptology ePrint Archive*, vol. 2016, p. 129, 2016. [Online]. Available: http://eprint.iacr.org/2016/129

[24] D. Genkin, L. Pachmanov, I. Pipman, E. Tromer, and Y. Yarom, "ECDSA key extraction from mobile devices via nonintrusive physical side channels," *IACR Cryptology ePrint Archive*, vol. 2016, p. 230, 2016. [Online]. Available: http://eprint.iacr.org/2016/230

[25] D. Genkin, A. Shamir, and E. Tromer, "RSA key extraction via low-bandwidth acoustic cryptanalysis," in *Advances in Cryptology - CRYPTO 2014 - 34th Annual Cryptology Conference, Santa Barbara, CA, USA, August 17-21, 2014, Proceedings, Part I*, ser. Lecture Notes in Computer Science, J. A. Garay and R. Gennaro, Eds., vol. 8616. Springer, 2014, pp. 444–461. [Online]. Available: http://dx.doi.org/10.1007/978-3-662-44371-2_25

[26] G. Goller and G. Sigl, "Side channel attacks on smartphones and embedded devices using standard radio equipment," in *Constructive Side-Channel Analysis and Secure Design - 6th International Workshop, COSADE 2015, Berlin, Germany, April 13-14, 2015. Revised Selected Papers*, ser. Lecture Notes in Computer Science, S. Mangard and A. Y. Poschmann, Eds., vol. 9064. Springer, 2015, pp. 255–270. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-21476-4_17

[27] A. Gornik, A. Moradi, J. Oehm, and C. Paar, "A hardware-based countermeasure to reduce side-channel leakage - design, implementation, and evaluation," *IACR Cryptology ePrint Archive*, vol. 2015, p. 348, 2015. [Online]. Available: http://eprint.iacr.org/2015/348

[28] A. Graps, "An introduction to wavelets," *IEEE Comput. Sci. Eng.*, vol. 2, no. 2, pp. 50–61, June 1995. [Online]. Available: http://dx.doi.org/10.1109/99.388960

[29] S. Gueron and V. Krasnov, "Fast prime field elliptic curve cryptography with 256 bit primes," *IACR Cryptology ePrint Archive*, vol. 2013, p. 816, 2013. [Online]. Available: http://eprint.iacr.org/2013/816

[30] S. Hajra and D. Mukhopadhyay, "On the optimal pre-processing for non-profiling differential power analysis," in *Constructive Side-Channel Analysis and Secure Design - 5th International Workshop, COSADE 2014, Paris, France, April 13-15, 2014. Revised Selected Papers*, ser. Lecture Notes in Computer Science, E. Prouff, Ed., vol. 8622. Springer, 2014, pp. 161–178. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-10175-0_12

[31] D. Hankerson, A. J. Menezes, and S. Vanstone, *Guide to Elliptic Curve Cryptography*, 1st ed. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2004. [Online]. Available: http://www.springer.com/us/book/97803879527342

[32] M. Joye and S. Yen, "The montgomery powering ladder," in *Cryptographic Hardware and Embedded Systems - CHES 2002, 4th International Workshop, Redwood Shores, CA, USA, August 13-15, 2002, Revised Papers*, ser. Lecture Notes in Computer Science, B. S. K. Jr., Ç. K. Koç, and C. Paar, Eds., vol. 2523. Springer, 2002, pp. 291–302. [Online]. Available: http://dx.doi.org/10.1007/3-540-36400-5_22

[33] B. S. K. Jr., Ç. K. Koç, and C. Paar, Eds., *Cryptographic Hardware and Embedded Systems - CHES 2002, 4th International Workshop, Redwood Shores, CA, USA, August 13-15, 2002, Revised Papers*, ser. Lecture Notes in Computer Science, vol. 2523. Springer, 2003.

[34] T. Kasper, D. Oswald, and C. Paar, "EM side-channel attacks on commercial contactless smartcards using low-cost equipment," in *Information Security Applications, 10th International Workshop, WISA 2009, Busan, Korea, August 25-27, 2009, Revised Selected Papers*, ser. Lecture Notes in Computer Science,

H. Y. Youm and M. Yung, Eds., vol. 5932. Springer, 2009, pp. 79–93. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-10838-9_7

[35] G. Kenworthy and P. Rohatgi, "Mobile device security: The case for side channel resistance," in *Mobile Security Technologies Conference, Workshop, California, USA, May 24, 2012, Proceedings*, 2012. [Online]. Available: http://www.mostconf.org/2012/papers/21.pdf

[36] C. K. Koc, "High-speed rsa implementation," RSA Laboratories, Tech. Rep., 1994.

[37] P. C. Kocher, "Timing attacks on implementations of diffie-hellman, rsa, dss, and other systems," in *Advances in Cryptology - CRYPTO '96, 16th Annual International Cryptology Conference, Santa Barbara, California, USA, August 18-22, 1996, Proceedings*, ser. Lecture Notes in Computer Science, N. Koblitz, Ed., vol. 1109. Springer, 1996, pp. 104–113. [Online]. Available: http://dx.doi.org/10.1007/3-540-68697-5_9

[38] P. C. Kocher, J. Jaffe, and B. Jun, "Differential power analysis," in *Advances in Cryptology - CRYPTO '99, 19th Annual International Cryptology Conference, Santa Barbara, California, USA, August 15-19, 1999, Proceedings*, ser. Lecture Notes in Computer Science, M. J. Wiener, Ed., vol. 1666. Springer, 1999, pp. 388–397. [Online]. Available: http://dx.doi.org/10.1007/3-540-48405-1_25

[39] P. C. Kocher, J. Jaffe, B. Jun, and P. Rohatgi, "Introduction to differential power analysis," *J. Cryptographic Engineering*, vol. 1, no. 1, pp. 5–27, 2011. [Online]. Available: http://dx.doi.org/10.1007/s13389-011-0006-y

[40] J. Longo, E. D. Mulder, D. Page, and M. Tunstall, "Soc it to EM: electromagnetic side-channel attacks on a complex system-on-chip," *IACR Cryptology ePrint Archive*, vol. 2015, p. 561, 2015. [Online]. Available: http://eprint.iacr.org/2015/561

[41] M. Medwed and E. Oswald, "Template attacks on ECDSA," *IACR Cryptology ePrint Archive*, vol. 2008, p. 81, 2008. [Online]. Available: http://eprint.iacr.org/2008/081

[42] A. J. Menezes, S. A. Vanstone, and P. C. V. Oorschot, *Handbook of Applied Cryptography*, 1st ed. Boca Raton, FL, USA: CRC Press, Inc., 1996.

[43] O. Meynard, D. Réal, S. Guilley, F. Flament, J. Danger, and F. Valette, "Characterization of the electromagnetic side channel in frequency domain," in *Information Security and Cryptology - 6th International Conference, Inscrypt 2010, Shanghai, China, October 20-24, 2010, Revised Selected Papers*, ser. Lecture Notes in Computer Science, X. Lai, M. Yung, and D. Lin, Eds., vol. 6584. Springer, 2010, pp. 471–486. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-21518-6_33

[44] D. P. Montminy, R. O. Baldwin, M. A. Temple, and M. E. Oxley, "Differential electromagnetic attacks on a 32-bit microprocessor using software defined radios," *IEEE Trans. Information Forensics and Security*, vol. 8, no. 12, pp. 2101–2114, 2013. [Online]. Available: http://dx.doi.org/10.1109/TIFS.2013.2287600

[45] A. Moradi, "Side-channel leakage through static power - should we care about in practice?" in *Cryptographic Hardware and Embedded Systems - CHES 2014 - 16th International Workshop, Busan, South Korea, September 23-26, 2014. Proceedings*, ser. Lecture Notes in Computer Science, L. Batina and M. Robshaw, Eds., vol. 8731. Springer, 2014, pp. 562–579. [Online]. Available: http://dx.doi.org/10.1007/978-3-662-44709-3_31

[46] C. O'Flynn and Z. D. Chen, "Channel equalization for side channel attacks," *IACR Cryptology ePrint Archive*, vol. 2014, p. 28, 2014. [Online]. Available: http://eprint.iacr.org/2014/028

[47] P. D. O'Grady, B. A. Pearlmutter, and S. T. Rickard, "Survey of sparse and non-sparse methods in source separation," *Int. J. Imaging Systems and Technology*, vol. 15, no. 1, pp. 18–33, 2005. [Online]. Available: http://dx.doi.org/10.1002/ima.20035

[48] T. Oliveira, J. López, D. F. Aranha, and F. Rodríguez-Henríquez, "Lambda coordinates for binary elliptic curves," in *Cryptographic Hardware and Embedded Systems - CHES 2013 - 15th International Workshop, Santa Barbara, CA, USA, August 20-23, 2013. Proceedings*, ser. Lecture Notes in Computer Science, G. Bertoni and J. Coron, Eds., vol. 8086. Springer, 2013, pp. 311–330. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-40349-1_18

[49] T. Plos, "Evaluation of the detached power supply as side-channel analysis countermeasure for passive UHF RFID tags," in *Topics in Cryptology -*

CT-RSA 2009, The Cryptographers' Track at the RSA Conference 2009, San Francisco, CA, USA, April 20-24, 2009. Proceedings, ser. Lecture Notes in Computer Science, M. Fischlin, Ed., vol. 5473. Springer, 2009, pp. 444–458. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-00862-7_30

[50] S. Ravi, A. Raghunathan, P. C. Kocher, and S. Hattangady, "Security in embedded systems: Design challenges," *ACM Trans. Embedded Comput. Syst.*, vol. 3, no. 3, pp. 461–491, 2004. [Online]. Available: http://doi.acm.org/10.1145/1015047.1015049

[51] P. Sasdrich, A. Moradi, and T. Güneysu, "White-box cryptography in the gray box - A hardware implementation and its side channels," *IACR Cryptology ePrint Archive*, vol. 2016, p. 203, 2016. [Online]. Available: http://eprint.iacr.org/2016/203

[52] A. Shamir, "Protecting smart cards from passive power analysis with detached power supplies," in *Cryptographic Hardware and Embedded Systems - CHES 2000, Second International Workshop, Worcester, MA, USA, August 17-18, 2000, Proceedings*, ser. Lecture Notes in Computer Science, Ç. K. Koç and C. Paar, Eds., vol. 1965. Springer, 2000, pp. 71–77. [Online]. Available: http://dx.doi.org/10.1007/3-540-44499-8_5

[53] M. J. Shensa, "The discrete wavelet transform: wedding the a trous and mallat algorithms," *IEEE Trans. Signal Processing*, vol. 40, no. 10, pp. 2464–2482, 1992. [Online]. Available: http://dx.doi.org/10.1109/78.157290

[54] S. Vaudenay, "Digital signature schemes with domain parameters: Yet another parameter issue in ECDSA," in *Information Security and Privacy: 9th Australasian Conference, ACISP 2004, Sydney, Australia, July 13-15, 2004. Proceedings*, ser. Lecture Notes in Computer Science, H. Wang, J. Pieprzyk, and V. Varadharajan, Eds., vol. 3108. Springer, 2004, pp. 188–199. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-27800-9_17

[55] Y. Zhou and D. Feng, "Side-channel attacks: Ten years after its publication and the impacts on cryptographic module security testing," *IACR Cryptology ePrint Archive*, vol. 2005, p. 388, 2005. [Online]. Available: http://eprint.iacr.org/2005/388