CHAPTER 6

ECDSA OVER GF($2^n$) WITH DYNAMIC SCHEDULING

This chapter suggests yet another enhanced methodology for calculating the point computation on ECDSA over binary field GF($2^n$) through parallel processing. It also points out one of the applications such as smart card for this approach.

6.1 INTRODUCTION

Elliptic Curve Digital Signature Algorithm (ECDSA) over binary field ($2^n$) becomes more popular to use in authentication service for smart card based applications (Liu Wen-yuan et al 2007). It is also used for automatic authentication service in the GRID environment of wireless network (Gyozo Godor et al 2006). The constraints of grid computing on wireless environment are the processing speed, storage area, size of the code, gate count in circuit design, power consumption and communication bandwidth (Keqin Li 2008). Actually, Grid environment is a collection of heterogeneous computers and the resources spread across the world to provide global access for the users. There are many ways to provide the resources for the computational grid in the secure environment. One of methods is digital signature, which provides authentication, authorization, integrity, and confidentiality (Viktors Berstics 2005). It is implemented through a lot of algorithms. But today, Elliptic Curve Digital Signature Algorithm over binary field ECDSA over GF($2^n$) has become more popular than other digital signature algorithms (DSA) because of its hard mathematic concept.
Secondly, the key size is smaller than the RSA based DSA (Westhoff et al 2004).

One of the ECDSA implementation such as ECDSA over binary field $\text{GF}(2^m)$ is used for various hardware applications. The NIST recommends the different fields based on the length of key for these applications. They are listed as $\text{GF}(2^{163})$, $\text{GF}(2^{233})$, $\text{GF}(2^{283})$, $\text{GF}(2^{409})$ and $\text{GF}(2^{571})$ and its corresponding reduction polynomials are listed in Table 6.1 (Brian King 2009).

**Table 6.1 NIST recommended binary field and its corresponding irreducible polynomial for ECDSA over GF($2^n$)**

<table>
<thead>
<tr>
<th>Binary Field</th>
<th>Reduction Polynomial</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\text{GF}(2^{163})$</td>
<td>$F(x) = x^{163} + x^7 + x^6 + x^3 + 1$</td>
</tr>
<tr>
<td>$\text{GF}(2^{233})$</td>
<td>$F(x) = x^{233} + x^{74} + 1$</td>
</tr>
<tr>
<td>$\text{GF}(2^{283})$</td>
<td>$F(x) = x^{283} + x^{12} + x^7 + x^5 + 1$</td>
</tr>
<tr>
<td>$\text{GF}(2^{409})$</td>
<td>$F(x) = x^{409} + x^{87} + 1$</td>
</tr>
<tr>
<td>$\text{GF}(2^{571})$</td>
<td>$F(x) = x^{571} + x^{10} + x^5 + x^2 + 1$</td>
</tr>
</tbody>
</table>

The main problem of ECDSA over $\text{GF}(2^n)$ is the lack of implementations (Arash Reyhani-Masoleh et al 2006). It has more number of the point additions and multiplications, when it is compared with ECC over $\text{GF}(2^n)$. So it is necessary to find out the suitable implementation for ECDSA, which reduces the number of clock cycles.

For this purpose, the Chapter is organized into seven sections. The section 6.2 gives an overview of ECDSA over $\text{GF}(2^n)$ and the section 6.3 proposes an innovative methodology for the implementation of ECDSA over $\text{GF}(2^n)$. The result of the proposed methodology is explained in the section
6.4 and the results are briefly analyzed in the section 6.5. The section 6.6 discusses two applications for this proposed methodology and finally it is concluded in the section 6.7.

6.2 **OVERVIEW OF ECDSA OVER GF(2^n)**

ECDSA is a combination of the Elliptic Curve over binary field GF(2^n) and Digital Signature Algorithm. It also has three main concepts known as digital signing, digital verifying and key generation for authentication. These procedures are implemented with some of the domain parameters as follows (Han Ping et al 2007).

\[
\text{ECDSA\_Domain\_parameters}(q, FR, a, b, \{\text{domain\_parameter\_seed}\}, G, n, h)
\]

where, \(q\)→binary field size \((2^n - 1)\),

\(FR\)→base two

\(a, b\)→binary field elements of the EC

\(\text{domain\_parameter\_seed}\)→an optional bit string that is present if the Elliptic Curve was randomly generated

\(G\)→a base point of higher order of polynomial (i.e., \(G = (xG, yG)\))

\(n\)→the order of the point \(G\),

\(h\)→the cofactor (which is equal to the order of the curve divided by \(n\))

For example, the following parameters shown in Table 6.2 are considered to implement ECDSA over GF(2^{163}) and its procedures are explained based on these domain parameters as follows.
Table 6.2 Domain parameters and its corresponding value of GF
GF(2\(^{163}\)), where \(p\) is 256 bits prime suggested by NIST

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>(q)</td>
<td>(2^{163})</td>
</tr>
<tr>
<td>FR</td>
<td>base two</td>
</tr>
<tr>
<td>(a)</td>
<td>1</td>
</tr>
<tr>
<td>(b)</td>
<td>0x 00000020A601907 B8C953CA 1481EB10 512F7874 4A3205FD</td>
</tr>
<tr>
<td>(a, b)</td>
<td>The coefficients of the elliptic curve (y^2 + xy = x^3 + ax^2 + b)</td>
</tr>
<tr>
<td>(n)</td>
<td>0x 0000004000000000 00000000 00292FE 77E70C12 A4234C33</td>
</tr>
<tr>
<td>(h)</td>
<td>2</td>
</tr>
<tr>
<td>(x)</td>
<td>0x 000003F0EBA162 86A2D57E A0991168 D4994637 E8343E36</td>
</tr>
<tr>
<td>(y)</td>
<td>0x 000000D51FBC6C 71A0094F A2CDD545 B11C5C0C 797324F1</td>
</tr>
<tr>
<td>(P)</td>
<td>(x,y)</td>
</tr>
<tr>
<td>(G)</td>
<td>a base point of higher order of polynomial assume (z^{163}+z^7+z^6+z^3+1)</td>
</tr>
</tbody>
</table>

6.2.1 Key Generation

The key generation produces a pair of keys for digital signature generation and verification for each person and it is defined as follows (Komathy K 2006):
1. Define $E_2^n(a,b)$, where $a, b \rightarrow$ variable and $p$ is an irreducible polynomial in the form of $2^n-1$.

2. Select another irreducible polynomial $q$ where $q$ is smaller than $p$.

3. Choose a binary integer $d$ as private key.

4. Choose $e_1(x,y))$ form EC point set.

5. Find $e_2(x, y)=d \odot e_1(x, y)$.

6. User private key is $d$ and public key is $(e_1, e_2, p, q, a, b)$ where $e_1$ and $e_2$ polynomial generators.

### 6.2.2 Digital Signature Generation

Based on these domain parameters and a pair of keys, the digital signature is created for authentication in secure environment. Its procedure is as follows:

1. choose random binary number $r$ where $1 < r < q-1$.

2. Find $P(x,y)=r \odot e_1(x,y)$.

3. Find $S_1=x \pmod{q}$ from $P$.

4. Find $S_2=(h(M) \oplus d \odot S_1)r^{-1}(\pmod{q})$.

5. Send $M$, $S_1$ and $S_2$ to receiver side.

### 6.2.3 Digital Signature Verification

The signature in the received message is verified through the following procedures, based on the same domain parameters with its corresponding key pairs.

1. Calculate $A=h(M)S_2^{-1}(\pmod{q})$
2. Calculate $B = S_2^{-1}S_1 \pmod{q}$

3. Find $T(x,y) = A \odot e_1(x,y) \odot B \odot e_2(x,y)$

4. Is $x = S_1 \pmod{q}$ than it is verified otherwise it is rejected

In this case, the hash function $(h(m))$ is implemented with SHA-1 to generate a hash code for authentication. It generates 163 bits hash code for the message with the maximum length of $2^{61}$ bytes information or less than $2^{61}$ bytes of information. This entire information is divided into equal number of blocks, which has 64 bytes information. These blocks are manipulated 80 times iteratively with feedback to produce the output block. In this case, the five registers are two byte size and they are used for the four byte length of word processing to produce output ($5 \times 32 \text{ bits} = 160 \text{ bits}$) (William Stallings 2006).

6.3 PROPOSED METHODOLOGY FOR ECDSA OVER GF ($2^n$)

The architecture of ECDSA over GF($2^n$) also has the four layers in ECDSA over GF(p). The bottom most layer is a basic binary field operations over GF($2^n$), and the second layer is a basic group operations of Elliptic Curve over GF($2^n$). The third layer is known as the scheduling layer to arrange the point multiplication code for parallel computation. The final layer is a ECDSA layer, which computes the various point operations of ECDSA over GF($2^n$).

Here also, the point multiplication is the main operation in both digital signing and digital verification operations of ECDSA over GF($2^n$). The operations of scalar multiplication in the inside branches can be executed in parallel through dynamic code scheduling, which is discussed in Chapter 4.3. The computation of point addition and doubling are done in parallel with the help of a few additional parameters to hold intermediate results. These point computations are implemented with the help of logical XOR, logical AND,
and shift operations and its coding is scheduled for concurrent processing.

6.4 EXPERIMENTAL SETUP AND RESULT

The Equations (4.1) and (4.7) are assumed to calculate the point computation for ECDSA over GF(2^n) with same parameters. First, the information is passed to the authentication processor, which is designed based on SHA-1 algorithm to produce the 160 bit message authentication code h(m). Then, it is converted into the EC points over GF(2^n) through mapping techniques (Sklavos N et al 2006). Finally, the signatures are computed based on the proposed point multiplication. The combination of message and signatures (M, S_1, S_2) are transmitted to receiver side. The receiver verifies the received information and regenerates signature to validate sender information. Mainly, the selection of point multiplication algorithm depends upon the platform characteristics, coordinate selection, memory constraints, security considerations and interoperability requirements. So the execution times of ECDSA over GF(2^n) are typically dominated by the number of point multiplications. It is measured with the number of clock cycles.

The experimental setup is based on the y^2=x^3+ax+b Equation for generating the same of points on Elliptic Curve. Then E_2^n(a,b) is defined by usingE_2^4(g^4,g^2). The point (x=g^5,y=g^3) is taken from the set to compute point multiplication for both linear scalar and the proposed binary tree multiplication as shown in Table 4.1. The same language ‘C’ is used for implementing the dynamic scheduling of ECDSA point multiplication under Ubandu version 10.02. The point multiplication of code scheduling is simulated for measuring the number of clock cycles based on the following configurations.
The number of clock cycles of each procedure in ECDSA over GF($2^n$) are measured based on the different of point computations. Particularly, the point multiplication is considered because of its time complexity. So the dynamic scheduling is applied on ECDSA over binary field to reduce the number of point multiplications, and also to create possibility for parallel processing. Here, the number of point multiplications in ECDSA over GF($2^n$) is shown in the following Table 6.3.

**Table 6.3 Number of point multiplication needed for the various operations in ECDSA**

<table>
<thead>
<tr>
<th>ECDSA Operation Name</th>
<th>Equation</th>
<th>Number of Point Multiplication</th>
</tr>
</thead>
<tbody>
<tr>
<td>Key generation</td>
<td>$e_2(x, y) = d \odot e_1(x, y)$</td>
<td>1</td>
</tr>
<tr>
<td>Signing</td>
<td>$P(x, y) = r \odot e_1(x, y)S_2 = (h(M) \oplus d \odot S_1)r^{-1}(mod)q$</td>
<td>2</td>
</tr>
<tr>
<td>Verifying</td>
<td>$h(M)S_2^{-1}(mod)qT(x, y) = (A \odot e_1(x, y)) \odot (B \odot e_2(x, y))$</td>
<td>3</td>
</tr>
<tr>
<td>Total Number of point multiplication in ECDSA</td>
<td>6</td>
<td></td>
</tr>
</tbody>
</table>
The point multiplications in the digital signing and digital verification are avoided through Quadruple processors. Besides, each processor is carrying to avoid the operand dependencies from 12% to 11%, the predictor dependencies from 6% to 2% and the loop carried dependencies from 8% to 3% concurrently.

6.5 DISCUSSION AND APPLICATION

This section suggests two applications for ECDSA over $2^n$ with the modified point multiplication. One is the smart card based application and another is digital authentication on Grid environment for resource sharing.

6.5.1 Smart Card

Today, Smart Cards are transforming business efficiencies around the world for reducing fraud and opening up on-line transactions in banking, personalizing services for mobile communication users and protecting revenues for public telephony operators. It is also used for securing access of corporate networks in grid environments (Chen Yang et al 2007).

These cards are plastic cards and their size is similar to the credit card. It has an embedded microchip with a little memory space. So, it provides the secure authentication tokens and the basic cryptographic services. Some of the examples for smart card based applications are as follows.

- Telephone calling
- Electronic cash payments
- Health care
- Identification of person in other applications
The varieties of smart cards are available in the market, for example: the microprocessor-based cards, cryptographic coprocessor cards and memory-based cards (Hamed Taherdoost 2011). These cards require a number of different cryptographic services with extremely fast performance. So the main objectives of these cards have the fact, that their stored data can be protected against unauthorized access and tampering. But the card has a small memory card (16kb or 32kb), low speed processors for slower execution time (16 bits), very limited processing powers (4 Mega Hertz) and slow in their Input/Output interface limitations.

Recently Elliptic Curve Digital Signature Standard (ECDSA) is used in smart card applications, because it needs less memory space, low bandwidth, limited power, reuse and portability.

In this article, ECDSA over GF($2^n$) is suggested with the proposed point computation to perform its operations. It increases the performance of ECDSA through the number of point addition reductions. So it avoids more power consumption and creates a possibility for supporting parallel computation. It also uses the hardware units in minimum number of times to increase the life time of these hardware units. Hence, it is more suitable for smart card, financial transactions and commercial applications.

6.5.2 Grid Computing

The significant challenges in the secure Grid computing are authentication, delegation, single logon, credential, authorization, privacy, confidentiality, message integrity, policy exchange, secure logging, assurance, manageability, firewall traversal and securing the Open Grid Service Architecture (OGSA) infrastructure. Besides, there is no way to guarantee, that nodes will not drop out of the network at random times. So Grid Security Infrastructure (GSI) needs a strong Grid Security Service (GSS) to provide
the security for distributed computing workstations. It means that, the basic idea of Grid Computing is to utilize the ideal Central Processing Unit (CPU) cycles and the storage of millions of computer systems, across Wide Area Network (WAN) to provide a flexible, pervasive, and inexpensive rights in secure environment (Ezedin S Barka et al. 2006). So GSI supports the execution of programs in a variety of platforms including a secure infrastructure for the Data movement/replication/federation, Resource discovery and Resource management. The GSI has a Globus Toolkit and the Public Key Infrastructure (PKI), which provides the technical framework for protocols, services and standards to support the user authentication, data confidentiality, data integrity, non-repudiation, key management protocols, services, and standards (Avijit Bhowmick et al. 2012).

A Grid environment consists of three layers known as Hard Users, Soft Services and Sensitivity Data Repository. The sensitivity data repository is a bottom layer in the grid environment, and the soft service is the middle layer to stand between the hard users and the data repositories. The hard user is a top layer to provide platform for the users. In addition to, the proposed model has an additional layer between hard user and soft service layer known as an authentication layer. It is used to perform digital signatures and other authentication, which is implemented by using the proposed methodology of ECDSA over GF(2^n) shown in Figure 6.1.
Besides, the grid computation is an expensive supercomputing for doing parallel computation, and it is still in the development stage to improve its efficiency and security (Bart Jacob et al 2005). However, the more computers involved in grid computation with the security problems will become increasingly serious. So it is the best choice to introduce a new layer for handling security services. Here, it is implemented with the proposed ECDSA over GF($2^n$) to perform authentication among systems.

6.6 SUMMARY

ECDSA over GF($2^n$) is more attractive for the high performance servers, small devices in wireless communication and smart cards, because it needs the lack of the power, minimum storage and computational power for fast processing. The proposed ECDSA over GF($2^n$) suggests a stronger and faster processing with smaller key size, high performance, lower computational cost with long life time. It is achieved through code scheduling.
for the hardware. But it needs some additional registers and buffers to hold intermediate results for further processing in the future. The space of integrated circuit may be extended in a little bit, and also the circuit complexity will be increased. In future, the circuit complexity and its design are optimized by using the advanced technology such as Nanotechnology.