Hastening Point Multiplication in the ECC

Ahmed Chalak Shakir, Jia Min and Gu Xuemai
School of Electronics and Information Engineering, Harbin Institute of Technology HIT, Heilongjiang, Harbin, 150001, China
Department of Computer Science, Collage of Science, Kirkuk University, Kirkuk, Iraq

Abstract: The demanding of the lightweight algorithms to produce efficient techniques used for security, is paving the way toward the exploiting of elliptic curve for cryptography. Therefore, there has a trend for substituting the traditional public key cryptography by the Elliptic Curve Cryptography (ECC) due to its efficiency for providing a high security with smaller keys in the comparison with other algorithms. The main problem in elliptic curve cryptography is the complexity of executing the operation of multiplying a point on the elliptic curve by the scalar value which is mainly fulfilled by the doubling and addition operations and is called scalar multiplication or point multiplication. This scalar can be represented by zeros and ones in terms of binary system. In double-and-add method, the number of ones (hamming weight) determines the number of addition operations, while the number of bits that represents the scalar determines the number of doubling operations. This paper produces the encoding method for reducing the hamming weight of the scalar and thereby diminishing the complexity of the scalar multiplication. The proposed method is compared with the one’s complement method and the simulation analysis showed that it gives lower hamming weight than the one’s complement method.

Key words: Elliptic curve cryptography, point multiplication, encoding, hamming weight, doubling and adding operations

INTRODUCTION

Recently, the wireless network is rapidly growing up and has been the backbone of our life, particularly the wireless sensor network. This open area makes the transmission of information among the nodes in the network be prone to the eavesdropping. Hence, the security issues being vital essential (Huang and Sharma, 2010; Huang et al., 2011). Due to the advantages that the ECC has in comparison with other algorithms of cryptography (DES, RSA, AES, etc.) (Rabah, 2006), especially for providing a robust security per bits (Rabah, 2005), therefore it has been the core of many standards. Although the ECC characterized by many advantages over the other algorithms, but it still has a problem for implementing one of its main operations called the scalar multiplication (or point multiplication) which required the complex computations (Wang et al., 2008) and spent 85% of ECC’s execution time (Gura et al., 2004).

This scalar multiplication has the form of kP, where k represents the private key, while P is the point on the elliptic curve (Shou et al., 2013). The double-and-add method (Moon, 2004) is used for performing this operation and depending on the binary number of such scalar, where the hamming (Razak et al., 2009) weight (non-zero elements or the number of ones) is affecting the number of addition operations and measured by, while the length of the number in terms of bits determines the number of doubling operations. In this paper the new encoding method is explored for re-coding such scalar in such a way that can produce smaller number of ones and consequently reducing the entire calculation of multiplication (accelerating the PM). Many previous works considered this problem and how it can be expedited (Reitwiesner, 1960; Kati 2002; Wang et al., 2007; Huang et al., 2010; Huang and Sharma, 2010; Mohamed et al., 2010; Basu, 2012). Once in a while, the proposed method is not only reducing the number of ones, rather than, it also reducing the length of the encoded scalar comparing with normal binary form. Nowadays, the orientation of exploration and studies in the field of securing such tiny devices, as in wireless sensor networks, is concentrating on the use of ECC with some considerations. Minimizing its computations is one of the most significant attribute, which makes it more efficient for the devices that are described by limited resources.

MATERIALS AND METHODS

Mathematical background of elliptic curve: An elliptic curve over real numbers may be defined as the set of
points \((x, y)\) which satisfy an elliptic curve equation (referred to as Weierstrass) as shown in Eq. 1:

\[
y^2 = x^3 + ax + b
\]  
(1)

where, \(x, y, a\), and \(b\) are real numbers. Each choice of numbers \(a\) and \(b\) yields a different elliptic curve. The discriminant of polynomial \(f(x) = x^3 + ax + b\) is shown in Eq. 2:

\[
4a^3 + 27b^2 \neq 0
\]  
(2)

where, if Eq. 2 is satisfied, then the elliptic curve \(y^2 = x^3 + ax + b\) can be used to form a group (Blake et al., 2000). Two main operations can be performed in EC, which are addition and doubling. Each one has its own characteristics and conditions (Kodali and Budwal, 2013).

PM can be defined as the repeated additions of a point along the elliptic curve and it is denoted as in Eq. 3:

\[
kP = P + P + P + + P
\]  
(3)

Repeatedly adding \(P\) to itself \(k\) times.

**Double-and-add method:** It is a method for implementing the PM in terms of number of zeros and ones which can be illustrated in algorithm 1 (Ganapathy and Mani, 2009):

Algorithm 1: Double-and-add method for PM

1. \(Q = 0\)
2. for \(i\) from \(m\) to \(0\) do
   1. \(Q = Q + P\) (using point doubling)
   2. if \(k_i = 1\) then \(Q = Q + P\) (using point addition)
3. Return \(Q\)

Equation 4 (Shah et al., 2010) is used for this computation of the form \(kP\):

\[
k = \sum_{i=0}^{\log_2 k} k_i 2^i, k_i \{0, 1\}
\]  
(4)

where, in example, the integer number \((131)_8\) is taken and then converted to the binary number as \((10000011)_2\). Table 1 and Fig. 1 are depicting the double-and-add method when applied on an integer 131. As a result of this example, the PM requires two additions and seven doubling operations.

Let \(r\) represents the number of ones and \(d\) represents the number of bits (length). Then the number of addition and doubling operations can be computed from the Eq. 5 and Eq. 6, respectively and as shown:

**Table 1:** An example of double-and-add

<table>
<thead>
<tr>
<th>Iterations</th>
<th>Bit-value</th>
<th>(Q + P) &quot;addition&quot;</th>
<th>(Q + 2Q) &quot;doubling&quot;</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>...</td>
<td>2P</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>...</td>
<td>2P</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>...</td>
<td>2(2P)</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>...</td>
<td>2(2(2P))</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>...</td>
<td>2(2(2(2P)))</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>...</td>
<td>2(2(2(2(2P))))</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>(P + 2(2(2(2(2P)))))</td>
<td>2P + 2(2(2(2(2P))))</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>(P + 2(2(2(2(2(2P)))))</td>
<td>...</td>
</tr>
</tbody>
</table>

No. of addition operation \(A = r - 1\)  
(5)

No. of doubling operation \(D = d - 1\)  
(6)

So that the total No. of operations used to implement the scalar multiplication can be computed using Eq. 7:

Total number of operations to achieve PM, \(T = A + D\)  
(7)

In the above example \(r = 3\), \(d = 8\), \(A = 2\) and \(D = 7\). Then \(T = 9\).

**Scalar encoding:** The encoding can be defined as the process of converting information into another form of representation, not inevitably of the same type.

Finding the efficient encoding algorithm for the integer \(k\) in \(kP\) has the direct impact on the efficiency of PM which leads to accelerate the computations and thereby be suitable for use in WSN. This encoding affects the number of point doubling and point addition operations. Whereas the number of bits that represents the integer \(k\) (length of \(k\)) affects the doubling operation, while the number of ones in such representation affects the addition operation. In this paper, we concentrated on the latter and minimized the hamming weight of \(k\) by encoding it using only "0" and "1" as in the binary system, but it differs from it such that using another algorithm for encoding not as in the binary system. Algorithm 2 depicts the idea where two bases are considered and it can be expanded to more than two bases:
To achieve the point multiplication $kP$, the double-add algorithm (Shah et al., 2010) is used before and after applying our method as shown in Table 3 and 4, respectively.

As shown in Table 5, the number of addition operations is diminished from 5-3 which save 2 addition operations, whereas the doubling operation is reduced by one. For comparison, the same example is achieved by using one’s complement recoding method, where the scalar is converted to one’s complement by using the Eq. 8 (Shah et al., 2010):

$$C_i = (2^i-1)-N$$  \hspace{1cm} (8)

Where:

- $C_i$ = One’s complement of the number $a$
- $a$ = No. of bit in $N$
- $N$ = Binary number

Then by applying the Eq. 8 on the scalar 91 yields:

$$C_i = (2^7-1)-1011011$$
$$= 28-1-1011011$$
$$= 127-1011011$$
$$= 111111-1011011$$
$$= 01001000$$ is the one’s complement of (1011011)

The Eq. 8 can be modified to Eq. 9:

$$N = (2^i-C_i)$$  \hspace{1cm} (9)

Then:

$$1011011 = 10000000-0100100-1$$
$$91 = 128-36-1$$
$$= 128-32-4-1$$

In which the hamming weight of the scalar is decreased from 5-4, thereby the proposed method gives better results.
Another method is taken for the comparison with our method called Non-Adjacent Form NAF (Shah et al., 2010) which is a unique sign digit representation. The Matlab code in the CODE1 below can be used for converting the decimal number into the NAF which is useful for decreasing the computation weight and consequently reducing the number of addition operations in the scalar multiplication.

```
Code 1
binF = dec2bin(E);
Z = zeros(1,length(binF)+1);
for i = 1:length(binF)+1
    zpos = length(binF)+i+2;
    if E>0
        if mod(E,2) == 1
            Z(zpos) = 2-mod(E,4);
        else
            Z(zpos) = 0;
        end
        E = (E - Z(zpos))/2;
    end
    i = i+1;
end
```

where, E represents the input number, while Z represents its NAF. Note that Z ∈ {0, 1, -1}.

By applying CODE1 on the same tested number (91), it yields:

```
  1 0 1 0 0 1 0 1
```

### Data manipulation in both sides of transmission:

In order to send the encrypted information from one point to another, both nodes must be well familiar with the protocols which answer the following questions:

- How the scalar is encoded? Recall that this scalar represents the key of the encryption
- Which information is necessary to be sent
- How the integrity of the data can be calculated

The above questions can be answered by the algorithm 3 as follows:

```
Algorithm 3: Data transmission (sender)
Convert the scalar value k from decimal (Xk)10 to new form base (Y)10
Compute R1, where (R1*K2F P) and P is the point on the elliptic curve.
Multiply the first base by the same point, (R1*K2P).
Multiply the second base by the same point, (R2*K2P).
Calculate the summation of such three points, R1+R2+R3.
Send R3 point with (Z) value (where Z is the binary form of Xk).
```

Recalling that, the bases are already known to the nodes. Take with the regard that the doubling-add algorithm is only working properly on the odd number. Therefore, Y is ORed with 1 to always get the prime number.

For example and because the least significant bit of (10100101110) is zero, so, it ORed with 1 to convert it to the odd number as (10100101110) OR (1) = (10100101111). Finally, converting the value of Z in algorithm 4 can be illustrated by the example shown in Fig. 2.

### RESULTS

As mentioned earlier, PM has two main operations-addition and doubling. The addition operation is affected by hamming weight of the scalar, while the doubling operation is affected by the length (e.g., total number of bits) of the scalar. The hamming weight of such scalar is reduced by the proposed method and the results are compared with the one’s complement method in Shah et al. (2010) and the NAF in the prime numbers in-between 1-100 are chosen for the test and simulation. The comparisons between the one’s complement, NAF and the proposed methods using MATLAB for different bases are produced in Table 6 which are analysed in Fig. 3, 4, while the hamming weight ratio (number of ones) for the scalar k (after converting it to the binary code) for each method are depicted in Fig. 5-8, respectively.
Fig. 3: Hamming weight ratios with respect to the length of the scalar for the tested methods

Fig. 4: Reduction ratio of hamming weight for the tested methods

Fig. 5: Bases \{2, 3\} are taken in the proposed method for the comparison with the tested methods in terms of the hamming weight
Fig. 6: Bases \{2, 3, 5\} are taken in the proposed method for the comparison with the tested methods in terms of the hamming weight.

Fig. 7: Bases \{2, 3, 7\} are taken in the proposed method for the comparison with the tested methods in terms of the hamming weight.

Table 6: Comparisons between tested methods in terms of ratio of the hamming weight.

<table>
<thead>
<tr>
<th>Bases</th>
<th>Ratio of ones (one's comp. method)</th>
<th>Ratio of ones (proposed method)</th>
<th>Ratio of ones (normal method)</th>
<th>Ratio of ones (NAF)</th>
<th>Reduction ratio of hamming weight (proposed method)</th>
<th>Reduction ratio of hamming weight (one's comp. method)</th>
<th>Reduction ratio of hamming weight (NAF)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2,3</td>
<td>0.5604</td>
<td>0.5481</td>
<td>0.6453</td>
<td>0.5256</td>
<td>0.0972</td>
<td>0.0849</td>
<td>0.1197</td>
</tr>
<tr>
<td>2,3,5</td>
<td>0.5604</td>
<td>0.5123</td>
<td>0.6453</td>
<td>0.5256</td>
<td>0.1130</td>
<td>0.0849</td>
<td>0.1197</td>
</tr>
<tr>
<td>2,3,7</td>
<td>0.5604</td>
<td>0.5348</td>
<td>0.6453</td>
<td>0.5256</td>
<td>0.1195</td>
<td>0.0849</td>
<td>0.1197</td>
</tr>
<tr>
<td>2,5,7</td>
<td>0.5604</td>
<td>0.5655</td>
<td>0.6453</td>
<td>0.5256</td>
<td>0.0798</td>
<td>0.0849</td>
<td>0.1197</td>
</tr>
</tbody>
</table>

1785
Fig. 8: Bases \( \{2, 5, 7\} \) are taken in the proposed method for the comparison with the tested methods in terms of the hamming weight.

**DISCUSSION**

Four values of bases \( \{2, 3, 5, 7\} \) are taken for the test on the prime numbers from 1-100 and as seen from the simulation results, the base \( \{2, 3, 5\} \) provides the best reduction ratio of the hamming weight. This reduction is calculated by depending on the normal binary method (binary number system) and comparing the results with the one’s complement and the NAF methods. In the teased range, the proposed method gave better hamming weight reduction over the one’s complement method. While it gave the better results over NAF only in the case where the \( \{2, 3, 5\} \) bases are used. Take with the account that if another range of prime numbers is simulated, the result may be different and other bases may give the best reduction of the hamming weight. This is owing to the number of bits and number of ones that each scalar has. Whenever the scalar goes larger, it can be divisible by the large bases like 5 and 7. Consequently, the results are depending on the how big the scalar is and what are the used bases.

**CONCLUSION**

The ECC has been proved by literal; it has many merits over other methods, especially for use in the constraints devices, like in wireless sensor network. Consequently, it has been contained in many standards. Scalar Multiplication \( kP \) is the kernel operation in the ECC, which needs complicated calculations. It can be executed in more efficient manner and appropriate for such limited devices if this operation is hastening. Since, the hamming weight of the scalar \( k \) affecting the number of addition in PM, hence the focusing of this paper was on this point and how it can be reduced. Meanwhile the one’s complement method and the NAF are representing the most important methods used for such acceleration; therefore, the comparisons were presented with them. Experimental result illustrated that the proposed method produced better reduction of the hamming weight over the one’s complement method, while produced better results over the NAF only in the case where the bases are \( \{2, 3, 5\} \). Ultimately, the integrity is tested in the receiver side using the algorithm 4.

**ACKNOWLEDGMENTS**

This study was supported by the National Natural Science Foundations of China (Grant No. 61201143), the Fundamental Research Fund for the Central Universities (Grant No. HIT. NSRIF. 2010091), the National Science Foundation for Post-doctoral Scientists of China (Grant No. 2012M510956) and the Post-doctoral Fund of Heilongjiang Province (Grant No. LBHZ11128).

**REFERENCES**


