Secure Wireless Sensor Networks with Dynamic Window for Elliptic Curve Cryptography

Xu Huang
Faculty of Information Sciences and Engineering,
University of Canberra, 2601, Australia
Xu.Huang@canberra.edu.au

Dharmendra Sharma
Faculty of Information Sciences and Engineering,
University of Canberra, 2601, Australia
Dharmendra.Sharma@canberra.edu.au

Mohammed Aseeri
Ministry of Interior, Saudi Arabia
Riyadh 11596 PoBox 67273
aseeri50@hotmail.com

Sultan Almorqi
ECP
King Abdulaziz City for Science and Technology, Kingdom of Saudi Arabia
Pritam.Shah@canberra.edu.au

Abstract—Elliptic curve cryptosystem (ECC) was proposed by Miller [10] and Koblitz [9] which relies on the difficulty of elliptic curve discrete logarithmic problem (ECDLP). It is gaining wide acceptance as an alternative to the conventional public key cryptosystem such as RSA [24], DSA [25], DH [26]. The security of the WSN becomes one of the major concerns in its applications. It is well known that scalar multiplication operation in ECC accounts for about 80% of key calculation time on wireless sensor network motes. In this paper we present an optimized dynamic window based on our previous research works. The whole quality of service (QoS) has been improved under this algorithm in particularly the power consuming is more efficiently. The simulation results showed that the average calculation time, due to fuzzy conditions decreased from previous 26 to current 9 as a whole the calculation time, decreased by approximately 18% in comparison to our previous algorithms in an ECC wireless sensor network [23].

Keywords—Elliptic curve cryptography (ECC), scalar multiplication, non-adjacent form, slide window, fuzzy control

I. INTRODUCTION

The high demand for various applications shows the fact that the rapid progress of wireless sensor networks has become popular in our daily life. With the growth in very large scale integrated (VLSI) technology, embedded systems and micro electro mechanical systems (MEMS) has enabled production of inexpensive sensor nodes, which can transmit data over a distances with free media and efficient use of power [1, 22, 23]. In the WSN systems, the sensor node will detect the interested information, processes it with the help of an in-built microcontroller and communicates results to a sink or base station. Normally the base station is a more powerful node, which can be linked to a central station via satellite or internet communication to form a network. There are many deployments for wireless sensor networks depending on various applications such as environmental monitoring e.g. volcano detection [2,3], distributed control systems [4], agricultural and farm management [5], detection of radioactive sources [6], and computing platform for tomorrows’ internet [7]. Generally speaking, a typical WSN architecture can be shown in Figure 1.

Contrast to traditional networks, a wireless sensor network normally has many resource constraints [4] due to the limited size. As an example, the MICA2 mote consists of an 8 bit ATmega 128L microcontroller working on 7.3 MHz. As a result nodes of WSN have limited computational power. Radio transceiver of MICA motes can normally achieve maximum data rate of 250 Kbits/s, which restricts available communication resources. The flash memory that is available on the MICA mote is only 512 Kbyte. Apart from these limitations, the on board battery is 3.3.V with 2A-Hr capacity. Therefore, the above restrictions with the current state of art protocols and algorithms are expensive for sensor networks due to their high communication overhead.

Figure 1: A Typical WSN architecture

Recalled that the Elliptic Curve Cryptography was first introduced by Neal Koblitz [9] and Victor Miller [10,11] independently in the early eighties. ECC offers the same level of security with smaller key size and it leads to the better performance in limited environments like cellular phones, PDA, sensor networking, etc. For example, ECC with a key size of 160 bits provides the same level of security as RSA [24], DSA [25] and DH [26] with a key size of 1024 bits. In summary, the ECC problem can only be solved in exponential time and, to date, there is a lack of sub exponential methods to attack ECC.
An elliptic curve $E$ over $GF(p)$ can be defined by

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

where $a, b \in GF(p)$ and

$$4a^3 + 27b^2 \neq 0$$

The point $(x, y)$ on the curve satisfies the above equation and the point at infinity denoted by $\infty$ is said to be on the curve.

II. ELLIPTIC CURVE DIFFIE-HELLMAN SCHEME PROPOSED FOR WSN

Before we get into our innovation method, we need to have a closer look at the popular legacy scheme for WSN. As per [13] the original Diffie-Hellman algorithm with RSA requires a key of 1024 bits to achieve sufficient security but Diffie Hellman based on ECC can achieve the same security level with only 160 bit key size.

In ECC, two heavily used operations are involved: scalar multiplication and modular reduction. Gura et al. [14] showed that 85% of execution time is spent on scalar multiplication. Scalar Multiplication is the operation of multiplying point $P$ on an elliptic curve $E$ defined over a field $GF(p)$ with positive integer $k$ which involves point addition and point doubling. Operational efficiency of $kP$ is affected by the type of coordinate system used for point $P$ on the elliptic curve and the algorithm used for recoding of integer $k$ in scalar multiplication.

The number of zeros and number of ones in the binary form, their places and the total number of bits will affect the computational cost of scalar multiplications. The Hamming weight as represented by the number of non-zero elements, determines the number of point additions and bit length of integer $K$ determines the number of point doublings operations in scalar multiplication.

One point addition when $P \neq Q$ requires one field inversion and three field multiplications [13]. Squaring is counted as regular multiplication. This cost is denoted by $1I + 3M$, where $I$ denotes the cost of inversion and $M$ denotes the cost of multiplication.

One point doubling when $P = Q$ requires $1I + 4M$ as we can neglect the cost of field additions as well as the cost of multiplications by small constant 2 and 3 in the above formulae.

**Binary Method**

Scalar multiplication is the computation of the form $Q = kP$, where $P$ and $Q$ are the elliptic curve points and $k$ is positive integer. This is obtained by repeated elliptic curve point addition and doubling operations. In binary method the integer $k$ is represented in binary form:

$$k = \sum_{j=0}^{l-1} K_j 2^j, \quad K_j \in \{0,1\}$$

The binary method scans the bits of $K$ either from left-to-right or right-to-left.

**Signed Digit Representation Method**

The subtraction has virtually the same cost as addition in the elliptic curve group. The negative of point $(x, y)$ is $(x, -y)$ for odd characters. This leads to scalar multiplication methods based on addition–subtraction chains, which help to reduce the number of curve operations. When integer $k$ is represented with the following form, it is a binary signed digit representation:

$$k = \sum_{j=0}^{l} S_j 2^j, \quad S_j \in \{0,1, -1\}$$

When a signed-digit representation has no adjacent non zero digits, i.e. $S_j S_{j+1} = 0$ for all $j \geq 0$ it is called a non-adjacent form (NAF).

NAF usually has fewer non-zero digits than binary representations. The average hamming weight for NAF form is $(n - 1)/3.0$. So generally it requires $(n - 1)$ point doublings and $(n - 1)/3.0$ point additions. The binary method can be revised accordingly and is given another algorithm for NAF, and this modified method is called the *Addition Subtraction* method.

III. FUZZY CONTROLLER FOR DYNAMITIC WINDOW IN ECC- PROPOSED ALGORITHM BASED

We are going to use the algorithm, based on *subtraction by utilization of the 1’s complement*, is most common in binary arithmetic. The 1’s complement of any binary number may be found by the following equation [19-22]:

$$C_1 = 2^n - 1 - N$$

where $C_1$ = 1’s complement of the binary number, $a = \text{number of bits in } N \text{ in terms of binary form, } N = \text{binary number.}$

From a closer observation of the equation (1), it reveals that any positive integer can be represented by using minimal non-zero bits in its 1’s complement form provided that it has a minimum of 50% Hamming weight. The minimal non-zero bits in positive integer scalar are very important to reduce the number of intermediate operations of multiplication, squaring and inverse calculations used in elliptical curve cryptography as we have seen in previous sections.

The equation (1) can therefore be modified as per below:

$$N = 2^n - C_1 - 1$$

For example, we may take $N=1788$ then it appears $N=(110111111100)_2$, in its binary form

$$C_1= \text{1's Complement of the number of } N=(00100000011)_2$$

$a$ is in binary form so we have $a = 11$

After putting all the above values in the equation (2) we have:

$$1788 = 2^{11} - 00100000011 - 1$$

this can be reduced as below:

$$1788 = 10000000000 - 00100000011 - 1$$

So we have

$$1788 = 10000000000 - 00100000011 - 1$$
1788= 2048 − 256 − 2 − 1 − 1

As is evident from equation (3), the Hamming weight of scalar \( N \) has reduced from 8 to 5 which will save 3 elliptic curve addition operations.

The above recoding method based on one’s complement subtraction combined with sliding window method provides a more optimized result.

Let us compute \([763]\) \( P \) (in other words \( k = 763 \)) as an example, with a sliding window algorithm with \( K \) recoded in binary form and window sizes ranging from 2 to 10.

Now we present the details for the different window size to find out the optimal window size using the following example:

**Window Size \( w = 2 \)**

\[ 763 = (1011111011)_2 \]

No of pre-computations = \( 2^w - 1 = 2^2 - 1 = [3] \) \( P \)

The intermediate values of \( Q \) are


Computational cost = 9 doublings, 4 additions, and 1 pre-computation.

**Window Size \( w = 3 \)**

No of pre-computations = \( 2^w - 1 = 2^3 - 1 = [7] \) \( P \)


\[ 763 = 1011111011 \]

The intermediate values of \( Q \) are


Computational cost = 7 doublings, 3 additions, and 3 pre-computations.

We continue to derive the remaining calculations for Window Size \( w = 6 \), Window Size \( w = 7 \), Window Size \( w = 8 \), Window Size \( w = 9 \), and Window Size \( w = 10 \). The analysis results will show “no. of Doublings” and “no. of Precomputation” will dominate in the beginning and then the later will control all the process.

IV. FUZZY CONTROLLER SYSTEM IN ECC

It is clear, from above description, that there is a tradeoff between the computational cost and the window size as shown in Table 1. However, this tradeoff is underpinned by the balance between computing cost (or the RAM cost) and the pre-computing (or the ROM cost) of the node in the network.

It is obviously to observe that the variety of wireless network working states will make this control complex and calculations could be relatively more expensive.

Therefore, we propose a fuzzy dynamic control system, to provide dynamic control to ensure the optimum window size is obtained by tradeoff between pre-computation and computation cost.

The fuzzy decision problem introduced by Bellman and Zadeh has as a goal the maximization of the minimum value of the membership functions of the objectives to be optimized.

Accordingly, the fuzzy optimization model can be represented as a multi-objective programming problem as follows [21]:

\[
Max : \min \{ \mu_i(D) \} \& \min \{ \mu_i(U) \} \quad \forall s \in S \& \forall l \in L
\]

such that \( A_l \leq C_1 \)

\[
\sum_{r \in R} x_r = 1 \quad \forall p \in P \& \forall s \in S,
\]

\[
x_r = 0 \quad \text{or} \quad 1 \quad \forall r \in R \& \forall s \in S
\]

In above equation, the objective is to maximize the minimum membership function of all delays, denoted by \( D \), and the difference between the recommend value and the measured value, denoted by \( U \).

The Fuzzy control system is extended from our previous design and shown in Figure 2. For the accurate control, we designed a three inputs fuzzy controller. The first input is storage room, which has three statuses, showing storage room in one of the three, namely (a) low, (b) average, and (c) high. The second input is pre-computing working load (PreComputing) in one of three states, namely (a) low, (b) average, and (c) high. The third input is Doubling, expressing how much working load for the calculation “doubling” which has three cases, namely (a) low, (b) average, and (c) high. The output is one, calledWindowSize, to express the next window size should be moved in which way, which has three states for the window sizes, namely (a) down, (b) stay, and (c) up.

There are 26 Fuzzy Rules listed as shown in Table 2 (weights are unit). Note: H=high; A=average; L=low; U=up; S=stay; D1=down.

<table>
<thead>
<tr>
<th>StorgeRoom</th>
<th>PreComputing</th>
<th>Doubling</th>
<th>WindowSize</th>
</tr>
</thead>
<tbody>
<tr>
<td>L</td>
<td>L</td>
<td>L</td>
<td>U</td>
</tr>
<tr>
<td>L</td>
<td>L</td>
<td>A</td>
<td>U</td>
</tr>
<tr>
<td>L</td>
<td>L</td>
<td>H</td>
<td>S</td>
</tr>
<tr>
<td>L</td>
<td>A</td>
<td>L</td>
<td>U</td>
</tr>
<tr>
<td>L</td>
<td>A</td>
<td>A</td>
<td>U</td>
</tr>
<tr>
<td>L</td>
<td>H</td>
<td>A</td>
<td>S</td>
</tr>
<tr>
<td>L</td>
<td>H</td>
<td>H</td>
<td>S</td>
</tr>
<tr>
<td>A</td>
<td>L</td>
<td>L</td>
<td>U</td>
</tr>
<tr>
<td>A</td>
<td>L</td>
<td>A</td>
<td>U</td>
</tr>
<tr>
<td>A</td>
<td>L</td>
<td>H</td>
<td>S</td>
</tr>
<tr>
<td>A</td>
<td>A</td>
<td>L</td>
<td>U</td>
</tr>
<tr>
<td>A</td>
<td>A</td>
<td>A</td>
<td>S</td>
</tr>
<tr>
<td>A</td>
<td>A</td>
<td>H</td>
<td>D1</td>
</tr>
<tr>
<td>A</td>
<td>H</td>
<td>A</td>
<td>S</td>
</tr>
<tr>
<td>A</td>
<td>H</td>
<td>H</td>
<td>S</td>
</tr>
<tr>
<td>H</td>
<td>L</td>
<td>L</td>
<td>S</td>
</tr>
<tr>
<td>H</td>
<td>L</td>
<td>A</td>
<td>S</td>
</tr>
<tr>
<td>H</td>
<td>L</td>
<td>H</td>
<td>D1</td>
</tr>
<tr>
<td>H</td>
<td>A</td>
<td>L</td>
<td>S</td>
</tr>
<tr>
<td>H</td>
<td>A</td>
<td>A</td>
<td>S</td>
</tr>
<tr>
<td>H</td>
<td>H</td>
<td>A</td>
<td>D1</td>
</tr>
<tr>
<td>H</td>
<td>H</td>
<td>L</td>
<td>D1</td>
</tr>
<tr>
<td>H</td>
<td>H</td>
<td>H</td>
<td>D1</td>
</tr>
</tbody>
</table>
It is noted that above 26 fuzzy rules did work in our previous designed system, but it is noted that there are only 9 Fuzzy Rules listed in above table will be dominated the whole fuzzy control procedure (if we still keep the weights are unit) due to the fact that StorageRoom can be ignored

Figure 2: Three inputs fuzzy window control system

Therefore, the above 26 fuzzy rules become the major 9 fuzzy rules, which will decrease the system latency and improve the quality of service (QoS). The 9 fuzzy rules can be described as below:

1. If 1·(PreComputing is low) and 1·(Doubling is low) then 1·(WindowSize is Up)
2. If 1·(PreComputing is low) and 1·(Doubling is average) then 1·(WindowSize is Up)
3. If 1·(PreComputing is low) and 1·(Doubling is high) then 1·(WindowSize is stay)
4. If 1·(PreComputing is average) and 1·(Doubling is low) then 1·(WindowSize is Up)
5. If 1·(PreComputing is average) and 1·(Doubling is average) then 1·(WindowSize is Up)
6. If 1·(PreComputing is average) and 1·(Doubling is high) then 1·(WindowSize is stay)
7. If 1·(PreComputing is high) and 1·(Doubling is low) then 1·(WindowSize is Up)
8. If 1·(PreComputing is high) and 1·(Doubling is average) then 1·(WindowSize is stay)
9. If 1·(PreComputing is high) and 1·(Doubling is high) then 1·(WindowSize is stay)

The number at each fuzzy condition being with a bracket is defined as “the weight number”, currently it is unit. Later we shall change it with different weight according to the running situations as described in the next. The three inputs with 9 fuzzy rules in Mamdani model running fuzzy controller part. The three inputs are StorageRoom, PreComputing and Doubling.

As our target is to making window size at closer to or sitting on the optimum “window size,” the output is taking WindowSize. It is noted that the “StorageRoom” has low, average, high with other two parameters’s combinations.

Now if we change the weight for above fuzzy rules as such the rules 1,5 10, 13, 15, 16, 18, 20 ,21, 22, 23, 25, and 26 are set in 0.5 (the rest will keep the same) due to the major functions are controlled by the storage room, and doubling will rapidly increasing by the window size larger. The outputs will changed as the average storage room will increased 0.04% and the other two inputs are decreased by 0.02% the output become window staying a little wider side by 0.003%.

It is clear that this fuzzy controller for the dynamic window is also involved a tradeoff between accuracy and control costs. For example the same system may go further for the second order parameters, i.e. not just check the changes about the input variables but also check the change tendencies of the variables, which will be discussed in another paper.

If we keep the StorageRoom constant, e.g. StorageRoom = 0.4, and the output of the “PreComputing” vs. “Doubling” is shown in Figure 3. Now in order to show the “StorageRoom” is not dominated factor as we discussed above, we took another value, “StorageRoom” = 0.8 and the output of the “PreComputing” vs. “Doubling” is shown in Figure 4.

Figure 3: The output of the surface for the StorageRoom vs. Doubling.

It is observed that the output of the “PreComputing” vs. “Doubling” is not much difference between Figures 3 and 4.

Figure 4: The output of the surface for the StorageRoom = constant (0.8) and PreComputing vs. Doubling.
The simulations of the example described in above were implemented. Also it is noted that with equation (2), the computational cost has been reduced from 3 additions as in the binary method to only 1 addition in one’s complement subtraction form. The number of pre-computations has remained the same.

In our simulations, the proposed method together with a fuzzy window size controller makes the ECC calculation in the current algorithm is about 18% more efficient (in terms of average) than the methods in [23] with the same QoS level.

V. CONCLUSION

In this paper, it has been investigated that the positive integer in point multiplication may be recorded with one’s complement subtraction to reduce the computational cost involved in this heavy mathematical operation for wireless sensor network platforms. As the NAF method involves modular inversion operation to get the NAF of binary number, the one’s complement subtraction can provide a very simple way of recording the integer. There is always a decision between pre-computing and computing, the former is related to the storage and the latter is associated with computing capability and capacity. The window size may be the subject of trade-off between the available RAM and ROM at a particular instance on a sensor node, which can be controlled by fuzzy controller. The final simulation in a sensor wireless network shows that about 18% more efficient than our previous method [23] can be obtained with an ECC sensor network.

REFERENCES


