http://ansinet.com/itj

ITJ

ISSN 1812-5638

# INFORMATION TECHNOLOGY JOURNAL

ANSInet
Asian Network for Scientific Information
308 Lasani Town, Sargodha Road, Faisalabad - Pakistan

Information Technology Journal 12 (9): 1780-1787, 2013
ISSN 1812-5638 / DOI: 10.3923/itj.2013.1780.1787


# Hastening Point Multiplication in the ECC

[1, 2]Ahmed Chalak Shakir, [1]Jia Min and [1]Gu Xuemai
[1]School of Electronics and Information Engineering, Harbin Institute of Technology HIT, Heilongjiang, Harbin,150001, China
[2]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 the 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; Katti 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

**Corresponding Author:** Gu Xuemai, School of Electronics and Information Engineering, Harbin Institute of Technology HIT, Heilongjiang, Harbin, 150001, China Tel: 045186418071

points (x, y) which satisfy an elliptic curve equation (referred to as Weierstrass) as shown in Eq. 1:

$$y^2 = x^3+ax+b \tag{1}$$

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:

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

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:

$$kP = \overbrace{P+P+P+P+.........+P+}^{k\ \text{times}} \tag{3}$$

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 (Ganopathy and Mani, 2009):

Algorithm 1: Double-and-add method for PM

1. Q = 0
2. for i from m to 0 do
2.1 Q := 2Q (using point doubling)
2.1 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:

$$k = \sum_{j=0}^{l-1} k_j 2^j, k_j \varepsilon \{0,1\} \tag{4}$$

where, in example, the integer number $(131)_{10}$ 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:

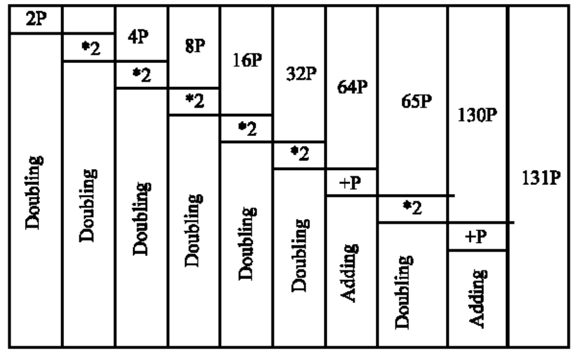


Fig. 1: Double-and-add procedure for the number 131

Table 1: An example of double-and-add

| Iterations No. | Bit-value | Q = Q + P "addition" | Q = 2 Q "doubling" |
|---|---|---|---|
| 0 | 1 | --- | 2P |
| 1 | 0 | --- | 2(2P) |
| 2 | 0 | --- | 2(2(2P)) |
| 3 | 0 | --- | 2(2(2(2P))) |
| 4 | 0 | --- | 2(2(2(2(2P)))) |
| 5 | 0 | --- | 2(2(2(2(2(2P))))) |
| 6 | 1 | P+2(2(2(2(2(2P))))) | 2(P+2(2(2(2(2(2P)))))) |
| 7 | 1 | P+2(P+2(2(2(2(2(2P)))))) | --- |

$$\text{No. of addition operation } A = r-1 \tag{5}$$

$$\text{No. of doubling operation } D = d-1 \tag{6}$$

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

$$\text{Total number of operations to achieve PM, } T = A+D \tag{7}$$

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:

Algorithm 2: Encoding representation of scalar *k*, where two bases are used:

```
Read the scalar k, k is Integer
Read the bases A, A = (b1, b2)
While k = 1 do
 If k mod b1 = 0 then
 k = k/b1
 Concatenate 0 to the right of the binary form
 Else
 If k mod b2=0 then
 k = k/b2
 Concatenate 0 to the right of the binary form
 Else
 k = (k-1)/b1
Concatenate 1 to the right of the binary form
 End if
Next
```

For reducing the number of ones, more than one base is used. The proposed method is comprehended by the example illustrated in Table 2.

Where, || symbol refers to the remainder of the division and modulo. As depicted in Table 1, number of ones is reduced from 5-3, which saves two elliptic curve addition operations, as well as the length is also reduced from 7 to 6 which saves one elliptic curve doubling operation. For more clarification and for example, the binary representation of the scalar $(91)_{10}$ which is shown in Table 1 is used before and after applying our method by multiplying this scalar by the elliptic point (12, 2) under the field of 23, such that a and b values of the coefficients are 0 and 1 respectively. This can be depicted in Table 3.

Table 2: Hamming weight diminishing example of scalar *k* using the proposed method

| k prime | Bases | Normal binary for k | Applying the proposed method | Proposed encoding of k |
|---|---|---|---|---|
| $(91)_{10}$ | 2,3 | 1011011 | (91-1)/2\|\|1 = 45/3\|\|0 = 15/3\|\|0 = (5-1)/2\|\|1 = 2/2\|\|0 = 1mod2\|\|1 | 101001 |

Table 3: Example of the scalar multiplication

| EC equation | Scalar value $(91)_{10}$ | Elliptic point | kP |
|---|---|---|---|
| $y^2 = x^3+x \bmod 23$ | Before: 1011011 | (12,2) | (1,5) |
| $y^2 = x^3+x \bmod 23$ | After: 101001 | (12,2) | (1,18) |

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):

$$C_1 = (2^a-1)-N \tag{8}$$

Where:

$C_1$ = One's complement of the number
a = No. of bit in N
N = Binary number

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

$$\begin{aligned} C_1 &= (2^7-1)-1011011 \\ &= 28-1-1011011 \\ &= 127-1011011 \\ &= 111111-1011011 \\ &= 0100100 \text{ is the one's complement of } (1011011) \end{aligned}$$

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

$$N = (2^a-C_1-1) \tag{9}$$

Then:

$$1011011 = 10000000-0100100-1$$

$$\begin{aligned} 91 &= 128-36-1 \\ &= 128-32-4-1 \end{aligned}$$

In which the hamming weight of the scalar is decreased from 5-4, thereby the proposed method gives better results.

Table 4: Computing kP using double-and-add method prior of applying the proposed method

| Scalar | Normal binary | Bit-value | Even-odd | Addition R = R+Q | Doubling Q = 2Q |
|---|---|---|---|---|---|
| $(91)_{10}$ | 1011011 | 1 | Odd | (12,2) | (2,20) |
| | | 1 | Odd | (15,8) | (0,22) |
| | | 0 | Even | ---- | (0,1) |
| | | 1 | Odd | (12,21) | (0,22) |
| | | 1 | Odd | (15,8) | (0,1) |
| | | 0 | Even | ---- | (0,22) |
| | | 1 | Odd | (1,5) | (0,1) |

Table 5: Computing kP using double-and-add method after applying the proposed method

| Scalar | Proposed encoding | Bit-value | Even-odd | Addition R = R+Q | Doubling Q = 2Q |
|---|---|---|---|---|---|
| - | 101001 | 1 | Odd | (12,2) | (2,20) |
| | | 0 | Even | --- | (0,22) |
| | | 0 | Even | --- | (0,1) |
| | | 1 | Odd | (15,15) | (0,22) |
| | | 0 | Even | --- | (0,1) |
| | | 1 | Even | (1,18) | (0,22) |

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 convert the decimal number into the NAF which is useful for decreasing the hamming weight and consequently reducing the number of addition operations in the scalar multiplication.

Code 1

```
binE = dec2bin(E);
Z = zeros(1,length(binE)+1);
for i =1:length(binE)+1
 zpos =length(binE)-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
Z
```

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

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 $(X)_{10}$ to new form base $(Y)_{b1,b2}$ (suppose $b_1,b_2$ are 2 and 3 respectively).
Compute R1, where ($R_1 \leftarrow K^* P$) and P is the point on the elliptic curve.
Multiply the first base by the same point, ($R_2 \leftarrow b_1 * P$)
Multiply the second base by the same point, ($R_3 \leftarrow b_2 * P$)
Calculate the summation of such three points, $R_s \leftarrow R_1+R_2+R_3$
Send $R_s$ point with (Z) value (where Z is the binary form of $(X)_{10}$).

| | | | | | |
|---|---|---|---|---|---|
| 1 | **Always 1** | | | | |
| 0 | **(2)** | **2×2+1** | | | |
| 1 | **(2)** | | | | |
| 0 | **(3)** | | **5×3** | | |
| 0 | **(3)** | | | **15×3** | |
| 1 | **(2)** | | | | **45×2+1 = 91** |

Fig. 2: Example of retrieving the scalar number from its encoded binary pattern

Algorithm 4: Data transmission (receiver)

Receiving X value and $R_s$ point
Using ECC algorithm and $R_s$ , calculating the value of $K_s$ (when the $K_s$ is the number that the point multiply by it and the result it $R_s$)
Evaluate the value of k such that:
$k \leftarrow K_s - (b_1+b_2)$
Using the bases $b_1$ and $b_2$ to convert the value of Z to Y.
Check the message
If ((Y OR 1) == k) then
The message is received correctly
Else
The message doesn't receive correctly and it must be resent again.

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.

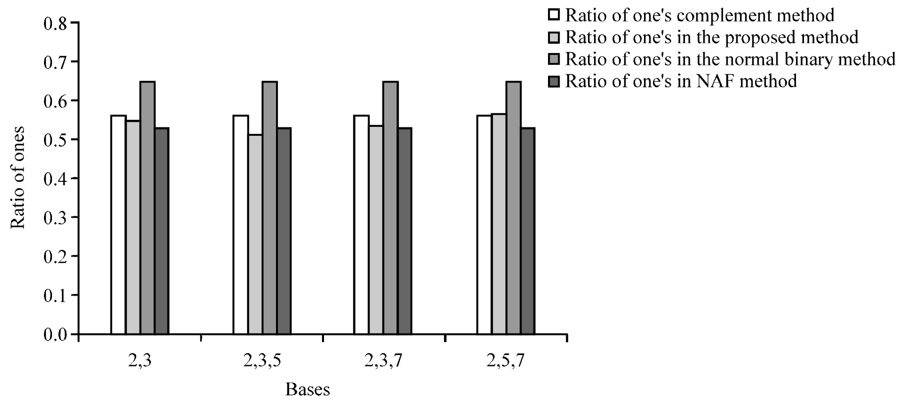


Fig. 3: Hamming weight ratios with respect to the length of the scalar for the tested methods

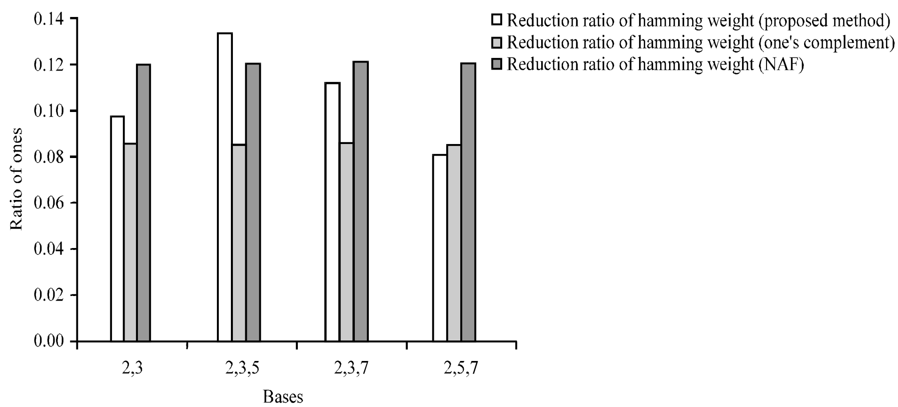


Fig. 4: Reduction ratio of hamming weight for the tested methods

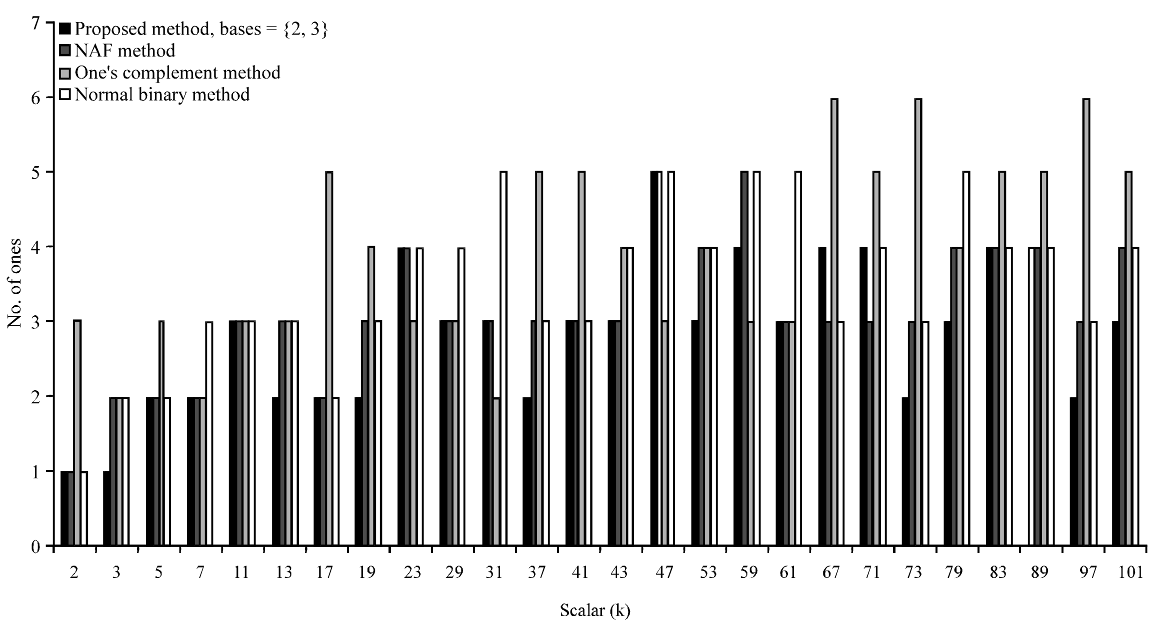


Fig. 5: Bases {2, 3} are taken in the proposed method for the comparison with the tested methods in terms of the hamming weight

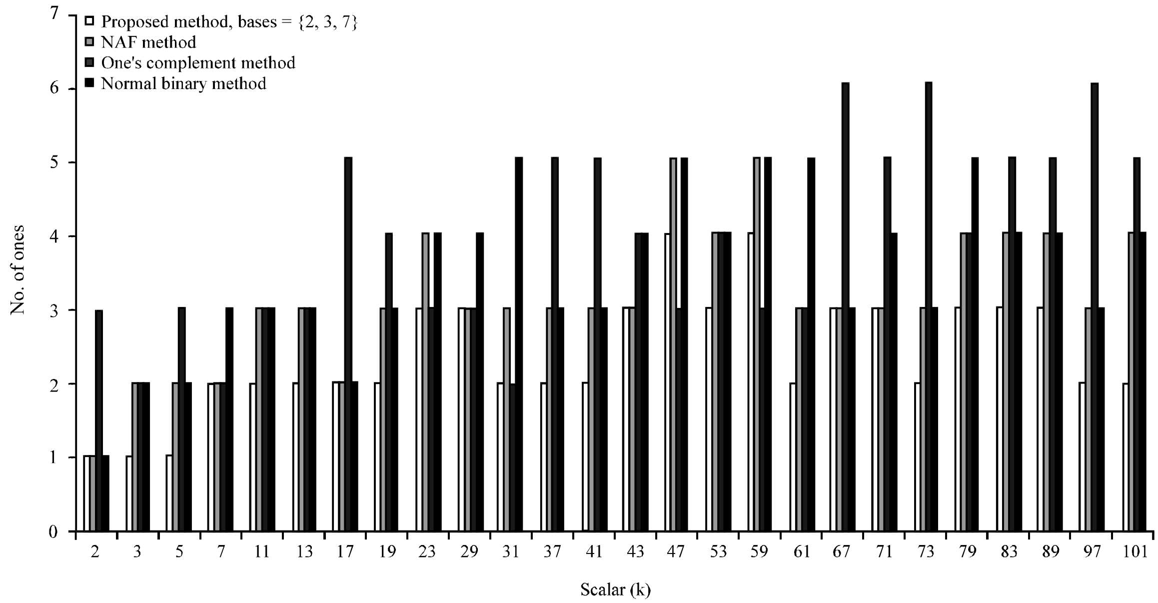


Fig. 6: Bases {2, 3, 5} are taken in the proposed method for the comparison with the tested methods in terms of the hamming weight

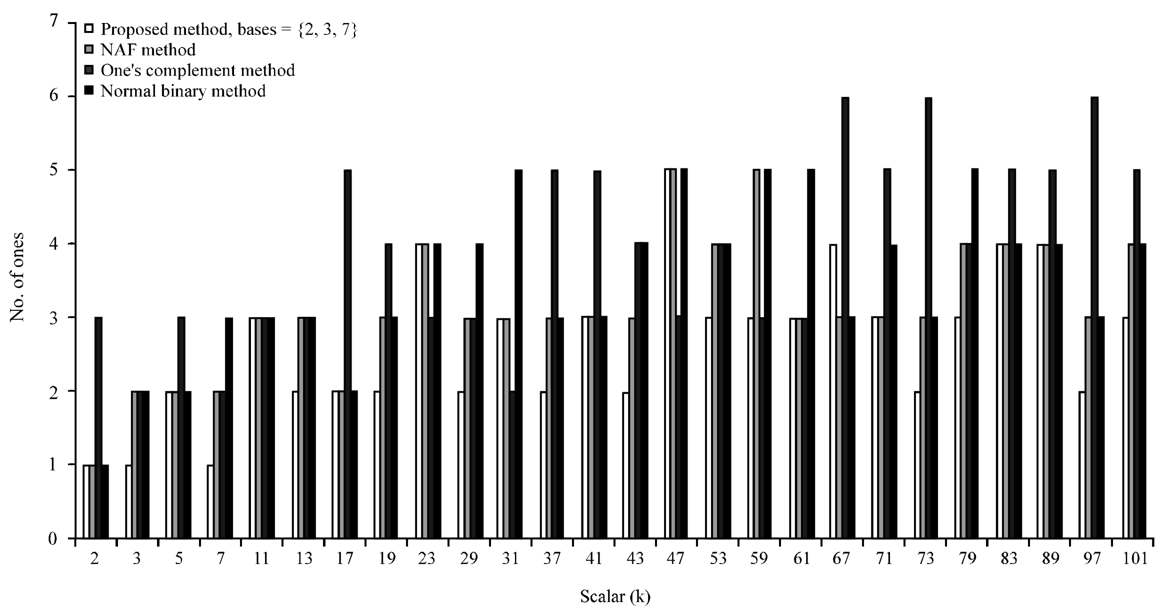


Fig. 7: Bases {2, 3, 7} are taken in the proposed method for the comparison with the tested methods in terms of the hamming weight

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

| Bases | Ratio of ones (one's comp. method) | Ratio of ones-(proposed method) | Ratio of ones-(normal method) | Ratio of ones (NAF) | Reduction ratio of hamming weight (proposed method) | Reduction ratio of hamming weight (one's comp. method) | Reduction ratio of hamming weight (NAF) |
|---|---|---|---|---|---|---|---|
| 2,3 | 0.5604 | 0.5481 | 0.6453 | 0.5256 | 0.0972 | 0.0849 | 0.1197 |
| 2,3,5 | 0.5604 | 0.5123 | 0.6453 | 0.5256 | 0.1330 | 0.0849 | 0.1197 |
| 2,3,7 | 0.5604 | 0.5348 | 0.6453 | 0.5256 | 0.1105 | 0.0849 | 0.1197 |
| 2,5,7 | 0.5604 | 0.5655 | 0.6453 | 0.5256 | 0.0798 | 0.0849 | 0.1197 |

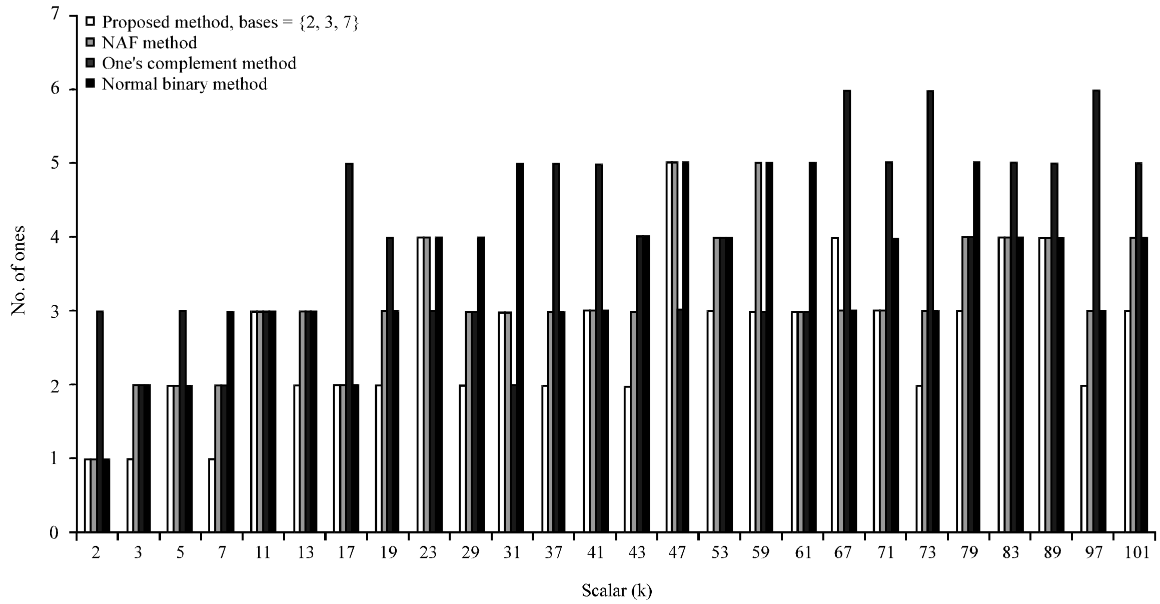


Fig. 8: Bases {2, 5, 7} are taken in the proposed method for the comparison with the tested methods in terms of the hamming weight

## DISCUSSION

Four values of bases (2, 3, 5 and 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 dividable 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 hasting. 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

Basu, S., 2012. A new parallel window-based implementation of the elliptic curve point multiplication in multi-core architectures. Int. J. Network Security, 14: 101-108.

Blake, I., G. Seroussi and N. Smart, 2000. Elliptic Curves in Cryptography: London Mathematical Society Lecture Note Series 265. Cambridge University Press, UK, ISSN: 0-521-65374-6.

Ganopathy, G. and K. Mani, 2009. Maximization of speed in elliptic curve cryptography using fuzzy modular arithmetic over a microcontroller based environment. Proceedings of the World Congress on Engineering and Computer Science, October 20-22, 2009, San Francisco, USA..

Gura, N., A. Patel, A . Wander, H. Eberle and C.S. Sheueling, 2004. Comparing elliptic curve cryptography and rsa on 8-bit cpus. Proceedings of the Workshop on Cryptographic Hardware and Embedded Systems, LNCS., 3156, August 11-13, 2004, Springer, Berlin, Heidelberg, pp: 119-132.

Huang, X. and D. Sharma, 2010. Fuzzy controller for a dynamic window in elliptic curve cryptography wireless networks for scalar multiplication. Proceedings of the 16th Asia-Pacific Conference on Communications, October 31-November 3, 2010, Auckland, pp: 458-463.

Huang, X., P. G. Shah and D. Sharma, 2010. Fast scalar multiplication for elliptic curve cryptography in sensor networks with hidden generator point. Proceedings of the International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery, October 10-12, 2010, Huangshan, pp: 243-249.

Huang, X., D. Sharma, M. Aseeri and S. Almorqi, 2011. Secure wireless sensor networks with dynamic window for elliptic curve cryptography. Proceedings of the Electronics, Communications and Photonics Conference, April 24-26, 2011, Riyadh, pp: 1-5.

Katti, R., 2002. Speeding up elliptic cryptosystems using a new signed binary representation for integers. Proceedings of the Euromicro Symposium on Digital System Design, (DSD'02), Dortmund, Germany, pp: 380-384.

Kodali, R.K. and H.S. Budwal, 2013. High performance scalar multiplication for ECC. Proceedings of the International Conference on Computer Communication and Informatics, January 4-6, 2013, Coimbatore, Tamil Nadu, pp: 1-4.

Mohamed, M.A., M.R.M. Said, K.A.M. Atan and Z.A. Zulkarnain, 2010. An improved binary method for scalar multiplication in elliptic curve cryptography. J. Math. Stat., 6: 28-33.

Moon, S., 2004. Elliptic curve scalar point multiplication using radix-4 Booth's algorithm. Proceedings of the International Symposium on Communications and Information Technologies, October 26-29, 2004, Hangzhou, China, pp: 80-83.

Rabah, K., 2005. Implementation of elliptic curve diffie-hellman and ec encryption schemes. Inf. Technol. J., 4: 132-139.

Rabah, K., 2006. Elliptic curve cryptography over binary finite field GF (2m). Inform. Technol. J., 5: 204-229.

Razak, Z., N.A. Ghani, E.M. Tamil, M.Y.I. Idris and N.M. Noor *et al.*, 2009. Off-line jawi handwriting recognition using hamming classification. Inform. Technol. J., 8: 971-981.

Reitwiesner, G.W., 1960. The determination of carry propagation length for binary addition. IRE Trans. Elect. Comput., EC-9: 35-38.

Shah, P.G., X. Huang and D. Sharma, 2010. Algorithm based on one's complement for fast scalar multiplication in ECC for wireless sensor network. Proceedings of the 2010 IEEE 24th International Conference on Advanced Information Networking and Applications Workshops (WAINA), April 20-23, 2010, Perth, WA., pp: 571-576.

Shou, Y., H. Guyennet and M. Lehsaini, 2013. Parallel Scalar Multiplication on Elliptic Curves in Wireless Sensor Networks. In: Distributed Computing and Networking, Frey, D., M. Raynal, S. Sarkar, R. Shyamasundar and P. Sinha (Eds.). Vol. 7730, Springer, Berlin, Heidelberg, ISBN: 978-3-642-35667-4, pp: 300-314.

Wang, B.J., H.G. Zhang and Y.H. Wang, 2007. An efficient elliptic curves scalar multiplication for wireless network. Proceedings of the Ifip International Conference on Network and Parallel Computing Workshops, September 18-21, 2007, China, pp: 131-134.

Wang, H., B. Sheng, C.C. Tan and Q. Li, 2008. Comparing symmetric-key and public-key based security schemes in sensor networks: A case study of user access control. Proceednigs of the 28th International Conference on Distributed Computing Systems, June 17-20, 2008, IEEE., pp: 11-18.