TRIPLING FORMULAE OF ELLIPTIC CURVE OVER BINARY FIELD IN LOPEZ-DAHAB MODEL

[1]SHARIFAH MD YASIN, [2]ZAITON MUDA
[1,2]Faculty of Computer Science and Information Technology, Universiti Putra Malaysia, 43400 Serdang, Malaysia
Email: {[1]ifah, [2]zaitonm}@upm.edu.my

ABSTRACT

In elliptic curve cryptosystem (ECC), scalar multiplication is the major and most costly operation. Scalar multiplication involves with point operations such as point addition, point doubling, and point tripling. Scalar multiplication can be improved by using efficient point operations. This research focuses on point tripling operation for elliptic curves over the binary field in Lopez-Dahab (LD) model. Currently, there is no existing tripling formula for this model. Traditionally, tripling is computed using one doubling followed by one addition (i.e. 3P=2P+P) with cost of 18M+8S, where M is field multiplication and S is field squaring. In this paper, we proposed tripling formulae with cost of 12M+7S. We proved the formulae and proposed its algorithm. The tripling saved 6M+1S which contribute to cost reduction in multiplication and squaring by 33% and 12.5% respectively when compared with the traditional method. For National Institute of Standards and Technology (NIST) curve (i.e. where $a = 1$), the cost of the tripling is further reduced to 10M+7S which saved 8M+1S from the traditional one. Further cost reduction in multiplication and squaring by 44% and 12.5% respectively.

Keywords: *Elliptic curve over binary field, scalar multiplication, point tripling, Lopez-Dahab.*

1. INTRODUCTION

Elliptic curve cryptosystem (ECC) is a public key cryptosystem and independently introduced in 1985 by Victor Miller [1] and Neal Koblitz [2]. Another well-known public key cryptosystem is the Rivest, Shamir, and Adleman (RSA) cryptosystem. Public key cryptosystem is also called asymmetric cryptosystem which requires a public and a private key that should correspond to each other. The security of ECC is based on the difficulty of the Elliptic Curve Discrete logarithm Problem (ECDLP). ECC has the advantage of having smaller key sizes for equivalent security level with its competitor, the RSA cryptosystem. For example, a 160-bit ECC has equivalent security strength of 1024-bit RSA. Also, 160-bit key requires 1/6 of the space that required by the RSA. It uses smaller memory and processor requirement than that of the RSA and also suitable for wireless devices which have limited bandwidth and processing power. ECC is an important cryptosystem in electronic banking and financial institutions [3]. ECC uses the properties of elliptic curves to provide the same functionality as other public key cryptosystem such as encryption, decryption, key generation, and digital signature. Most ECC implementation demands for high performance and low power ECC architectures [4].

In ECC, scalar multiplication is the major operation of the cryptosystem. The main challenge in ECC research is to perform efficient scalar multiplication. The implementation of ECC scalar multiplication is the most time consuming operation [5]. The scalar multiplication is achieved by repeatedly doing elliptic curve point operations: point addition, point doubling, point tripling and etc. In the literature, some researcher optimizing the scalar multiplication operation by improving the efficiency of the point operations. Generally, the cost of the scalar multiplication depends on the selection of elliptic curve parameters, representation of a scalar k, point operations and field operations. This research mainly focuses on improving the efficiency of point operation for elliptic curves over the binary field in the Lopez-Dahab Model.

Generally, point operation can be done in various coordinate representations. Traditionally, point operations use affine coordinates which involved with expensive inversion operations. Many ECC implementations using projective coordinates like Jacobian and Lopez-Dahab (LD), in order to represent the points on the curve by reducing inversion/division to one [4]. Different coordinate system gives different cost for point operations [6][7][8][9]. LD coordinate gives the best performance for elliptic curve over the binary field [10]. Certain coordinate system may be superior when performing doublings, but not when performing addition [11]. Hybrid or mixed coordinates can be very efficient in improving performance of scalar multiplication [12][13]. Mixed coordinates means that different coordinate systems are used for inputs and outputs. Sometimes two different coordinates are used for input and another coordinate system for output.

In this paper, we propose a new tripling formulae and algorithm in LD coordinates for elliptic curves over the binary field. Related work is presented in Section 2 and a brief introduction on Lopez-Dahab coordinates are in Section 3. Proposed tripling formula, algorithm and proofs of the formula is in Section 4. Concluding remarks are in Section 5.

2. RELATED WORK

Table 1: Cost of Point Operation in Lopez-Dahab vs. Jacobian Coordinates for Elliptic Curve Over Binary Field

Point Operation	Lopez-Dahab	Jacobian
Point Addition	14M+6S [14] 13M+4S [16]	16M+3S [15]
Point Doubling	5M+5S [14] 5M+4S [17]	5M+5S [15]
Mixed Addition	9M+5S [18, 19] Affine-Lopez	11M+3S [15] Affine-Jacobian
General Tripling	18M + 8S (Traditionally, compute point doubling followed by point addition)	15M+7S [20]

Table 1 shows estimated cost of point operation for LD and Jacobian coordinates for elliptic curve over binary field. The costs are measured by counting the number of field operation performed: Multiplication (M) and Squaring (S). Traditionally, point addition and point doubling are proposed by [14]. Improved point addition proposed by [16] and improved point doubling proposed by [17]. Al-Daoud et al [18][19] proposed a mixed addition formulae using affine and LD coordinates with better computation cost than the traditional addition. No existing tripling formula for LD coordinates and it is traditionally computed using one doubling operation then followed by one addition for the cost of 18M+8S.

Point addition and point doubling for Jacobian have cost of 16M+3S and 5M+5S respectively. Mixed addition using affine and Jacobian coordinates has cost of 11M+3S. Dimitrov et al [20] proposed efficient tripling formula in Jacobian with cost of 15M+7S. He also proposed tripling formula in affine for elliptic curve over binary field [21]. The tripling operation is using 3P=2P+P where a doubling operation is computed first then followed by an addition. Meloni [22] proposed special addition formula that can be used for efficient scalar multiplication. Moon [23] proposed point quadruple (4P) operation to accelerate the quad-and-add scalar multiplication algorithm. Edward [24] introduced new coordinates of elliptic curves. Hisil et al [25] proposed new doubling and tripling formula for Jacobian coordinates. Mishra and Dimitrov [20] proposed quintupling (5P) formulas in Jacobian coordinates for elliptic curves over prime fields. The 5P formula is used to speed up the scalar multiplication algorithm.

3. ELLIPTIC CURVE OVER BINARY FIELD IN LOPEZ-DAHAB (LD) MODEL

Affine coordinates are represented by two coordinates x and y. The scalar multiplication in affine coordinates involves with expensive inversion operations. Whereas LD is a projective coordinates that omits inversion operation. LD coordinate is in the form (X, Y, Z) where $(Z\neq0)$ corresponding to the point $(X/Z, Y/Z^2)$ in the affine coordinate [14]. LD projective equation is given below:

$$Y^2 + XYZ = X^3Z + aX^2Z^2 + bZ^4. \qquad (1)$$

The triple $(X_1: Y_1: Z_1)$ represents the affine point $(X_1/Z_1, Y_1/Z1^2)$ when $Z_1 \neq 0$. The point at infinity P_∞ corresponds to $(1:0:0)$, while the negative of $(X_1: Y_1 : Z_1)$ is $(X : XZ + Y : Z)$. LD point doubling and addition formulae can be derived by substituting $x = X/Z$ and $y = Y/Z^2$ in the affine formulae and clearing the denominators. Lopez-Dahab [14] proposed a general addition formula which costs 14M+6S where M denotes multiplication, and S is a squaring. Higuchi and Takagi [16] improved the LD addition formula which costs 13M+4S. Al-Daoud et. al [19] proposed mixed addition using affine-LD coordinates. In this research, this formula is used to derive the tripling formulae. Mixed addition is as follows:

Consider $P = (X_1, Y_1, 1)$, $Q = (X_2, Y_2, Z_2)$ such that $P \neq \pm Q$. P is an affine and Q is in LD coordinates, then, $P \oplus Q = (X_3, Y_3, Z_3)$ is given by

$$\begin{gathered} A=Y_2+Y_1Z_2^2 \\ B = X_2+X_1Z_2 \\ C = BZ_2 \\ Z_3=C^2 \\ D=X_1Z_3 \\ X_3=A^2+C(A+B^2+aC) \\ Y_3=(D+X_3)(AC+Z_3)+(Y_1+X_1)Z_3^2 \end{gathered} \tag{2}$$

Mixed addition costs 9M+5S which is cheaper than the general addition. If $a \in \{0, 1\}$ then only eight general multiplications are required. Lopez Dahab [14] proposed doubling formula which costs 5M+5S. Lange [17] improved LD doubling formula which costs 5M+4S. In this research, this formula is also used to derive a tripling formula and as shown below:

Consider $P=(X_1{:}Y_1{:}Z_1)$ then $2P =(X_2{:}Y_2{:}Z_2)$

$$\begin{gathered} \mathrm{S} = \mathrm{X}_1^{\,2}, \\ \mathrm{U} = \mathrm{S} + \mathrm{Y}_1, \\ \mathrm{T} = \mathrm{X}_1\mathrm{Z}_1, \\ \mathrm{R} = \mathrm{UT} \\ \mathrm{Z}_2 = \mathrm{T}^2, \\ \mathrm{X}_2 = \mathrm{U}^2 + \mathrm{R} + a\mathrm{Z}_2, \\ \mathrm{Y}_2 = (\mathrm{Z}_2 + \mathrm{R})\mathrm{X}_2 + \mathrm{S}^2\mathrm{Z}_2 \end{gathered} \tag{3}$$

4. PROPOSED TRIPLING FORMULAE IN LOPEZ DAHAB MODEL

In this research, the tripling operation is computed as 3P=2P+P using one doubling (Equation 3) and followed by one mixed addition (Equation 2). The expected tripling cost for one doubling and one mixed addition is given below:

$$(5M+4S) + (9M+5S) = (14M+9S). \tag{4}$$

3.1 Proposed Tripling Formulae

In this research, we proposed a tripling formula with cost 12M+7S. Consider $P=(X_1,Y_1,1)$, then, $3P=(X_3,Y_3,Z_3)$. Then, (X_3,Y_3,Z_3) is as follows:

$$\begin{gathered} A = (Z_2 + E)X_2 + Z_2^2U, \\ B = X_2 + X_1Z_2, \\ C = BZ_2, \\ Z_3 = C^2, \\ D = X_1Z_3, \\ Z_2 = X_1^2 \\ E=UX_1, \\ X_2 = U^2+E+aZ_2, \\ X_3 = A^2 + C(A + B^2) + aZ_3, \\ Y_3 = (D + X_3)(AC + Z_3) + (Y_1 + X_1)Z_3^2 \end{gathered} \tag{5}$$

As in Table 1, the traditional tripling has cost (18M+8S). The tripling cost in Equation 5 is better than the traditional tripling.

3.2 Proof of the Proposed Tripling Formula

To prove the tripling, we use affine coordinates. In affine, a nonsupersingular elliptic curve over binary field, E(F_{2^m}) is defined by parameter $a, b \in F_{2^m}$ consists of the set of solutions or points P(x, y) where $x, y \in F_{2^m}$ for the equation $y^2 + xy = x^3 + ax^2 + b$.

Addition of two points, $P(x_P, y_P)$ and $Q(x_Q, y_Q)$, are distinct points such that $P \neq \pm Q$, then

$$P(x_P, y_P) + Q(x_Q, y_Q) = R(x_R, y_R)$$

where

$$x_R \equiv \lambda^2 + \lambda + x_P + x_Q + a \text{ in } F_{2^m};$$

$$y_R \equiv \lambda(x_P + x_R) + x_P + y_P \text{ in } F_{2^m}$$

and $$\lambda \equiv \frac{(y_P + y_Q)}{(x_P + x_Q)} \text{ in } F_{2^m} \tag{6}$$

Doubling a point, provided that $y_P \neq 0$, then

$$P(x_P, y_P) + Q(x_P, y_P) = R(x_R, y_R)$$

where

$$x_R \equiv \lambda^2 + \lambda + a \text{ in } F_{2^m}$$

$$y_R \equiv \lambda(x_P + x_R) + x_R + y_P = x_P^2 + (\lambda + 1)x_R \text{ in } F_{2^m}$$

and $$\lambda \equiv x_P + \frac{(y_P)}{(x_P)} \text{ in } F_{2^m} \tag{7}$$

Firstly, let $P=(x_1, y_1)$, from (Equation 7), then $2P=(x_2, y_2)$ is as the following:

$$x_2 = \lambda_1^2 + \lambda_1 + a$$
$$y_2 = (x_1 + x_2)\lambda_1 + x_2 + y_1$$
$$\lambda_1 = \frac{y_1 + x_1^2}{x_1} \qquad (8)$$

Secondly, use x_2, y_2, and λ_1 from (Equation 8) compute point tripling as $3P = 2P + P = (x_2, y_2) + (x_1, y_1) = (x_3, y_3)$. Then, from (Equation 7), *(x₃,y₃)* is as the following:

$$x_3 = \lambda_2^2 + \lambda_2 + x_1 + x_2 + a$$
$$y_3 = (x_1 + x_3)\lambda_2 + x_3 + y_1$$
$$\lambda_2 = \frac{y_1 + y_2}{x_1 + x_2} \qquad (9)$$

Thirdly, using (Equation 5) and (Equation 9), need to prove that $\frac{X_3}{Z_3} = x_3$ and $\frac{Y_3}{Z_3^2} = y_3$. The process is shown below:

$$\frac{X_3}{Z_3} = \frac{A^2 + C(A+B^2) + aZ_3}{C^2} = \frac{A^2}{C^2} + \frac{A}{C} + \frac{B^2}{C} + a$$
$$= \frac{[(Z_2+E)X_2 + Z_2^2U]^2}{(X_2+X_1Z_2)^2Z_2^2} + \frac{[(Z_2+E)X_2 + Z_2^2U]}{(X_2+X_1Z_2)Z_2} + \frac{(X_2+X_1Z_2)^2}{(X_2+X_1Z_2)Z_2} + a$$
$$= \frac{(Y_2+Y_1Z_2)^2}{(X_2+X_1Z_2)^2Z_2^2} + \frac{(Y_2+Y_1Z_2)}{(X_2+X_1Z_2)Z_2} + \frac{(X_2+X_1Z_2)^2}{(X_2+X_1Z_2)Z_2} + a$$

Use $Z_1 = 1$, also $\lambda_2 = \frac{y_1 + y_2}{x_1 + x_2}$, then,

$$x_3 = \frac{(y_1 + y_2)^2}{(x_1 + x_2)^2} + \frac{(y_1 + y_2)}{(x_1 + x_2)} + \frac{(x_1 + x_2)^2}{(x_1 + x_2)} + a$$
$$x_3 = \lambda_2^2 + \lambda_2 + x_1 + x_2 + a(proven)$$

$$\frac{Y_3}{Z_3^2} = \frac{(D + X_3)(AC + Z_3) + (Y_1 + X_1)Z_3^2}{Z_3^2}$$
$$= \frac{(X_1Z_3 + X_3)\left[(Y_2 + Y_1Z_2^2)BZ_2 + Z_3\right] + (Y_1 + X_1)Z_3^2}{(X_2 + X_1Z_2)^4Z_2^4}$$
$$= \frac{(X_1Z_3 + X_3)\left[(Y_2 + Y_1Z_2^2)BZ_2\right]}{(X_2 + X_1Z_2)^4Z_2^4} + \frac{(X_1Z_3 + X_3)Z_3}{(X_2 + X_1Z_2)^4Z_2^4} + Y_1 + X_1$$
$$= \frac{(X_1Z_3 + X_3)\left[(Y_2 + Y_1Z_2^2)\right]}{(X_2 + X_1Z_2)^3Z_2^3} + \frac{(X_1Z_3 + X_3)}{(X_2 + X_1Z_2)^2Z_2^2} + Y_1 + X_1$$

Use $X_3 = x_3 . Z_3$, then substitute $Z_2=1$ and $Z_3=1$,

$$y_3 = \frac{(x_1 + x_3)(y_1 + y_2)}{(x_1 + x_2)} + x_1 + x_3 + y_1 + x_1$$
$$y_3 = (x_1 + x_3)\lambda_2 + x_3 + y_1(proven)$$

Thus, x_3, y_3 and λ_2 are the same as (Equation 9).

3.3 Proposed Tripling Algorithm Based on The New Tripling Formulae

We also proposed a tripling algorithm based on our formula. The algorithm is as follows:

Algorithm: Tripling with Mixed Addition

Input: Two points $P = (X_1, Y_1, 1)$ and $2P = (X_2, Y_2, Z_2)$ where *P* is in affine and *2P* is in LD coordinates.

Output: Point $3P = (X_3, Y_3, Z_3)$

```
1.  $T_1 \leftarrow X_1$
2.  $T_2 \leftarrow Y_1$
3.  $Z_2 \leftarrow (T_1)^2$
4.  $T_3 \leftarrow Z_2 + T_2$
5.  $T_4 \leftarrow T_3 * T_1$
6.  $T_5 \leftarrow a$
7.  $X_2 \leftarrow (T_3)^2$
8.  $X_2 \leftarrow X_2 + T_4 + (T_5 * Z_2)$
9.  $T_6 \leftarrow (Z_2)^2$
10. $T_6 \leftarrow T_6 * T_3$
11. $T_6 \leftarrow T_6 + (Z_2 + T_4) * X_2)$
    $T_4 \leftarrow X_2 + (T_1 * Z_2)$
12. $Z_2 \leftarrow T_4 * Z_2$
13. $T_3 \leftarrow (T_6)^2$
14. $T_4 \leftarrow (T_4)^2$
15. $T_4 \leftarrow T_4 + T_6 + (T_5 * Z_2)$
    $T_4 \leftarrow T_4 * Z_2$
16. $X_3 \leftarrow T_4 + T_3$
17. $Z_3 \leftarrow (Z_2)^2$
18. $T_4 \leftarrow T_1 * Z_3$
19. $T_6 \leftarrow (T_6 * Z_2) + Z_3$
20. $T_4 \leftarrow T_4 + X_3$
21. $T_6 \leftarrow T_6 * T_4$
22. $T_3 \leftarrow (Z_3)^2$
23. $T_3 \leftarrow (T_1 + T_2) * T_3$
```

```
24. $Y_3 \leftarrow T_6 + T_3$
25. return (3P)
```

5. CONCLUSION

As in Table 1, the traditional tripling has cost (18M+8S). The tripling (Equation 5) has cost (12M+7S). This tripling saved (2M+2S) from the traditional tripling, which saved 14% multiplication cost and 22% squaring cost from the traditional method. If National Institute of Standards and Technology (NIST) curve is used, the value of *a* is equal to 1, and the new tripling cost is further reduced to (10M+7S) which saved 28.6 % multiplication cost and 22% squaring cost from the traditional method.

For future work, the tripling might be further improved in time. These tripling can be used in Double Base Number System (DBNS) scalar multiplication and promotes efficient implementation of DBNS based scalar multiplication. The tripling can also be used for {0,1,3}-NAF scalar multiplication [26].

REFERENCES:

[1] V. Miller, "Use of elliptic curves in cryptography," *Advances in Cryptology–Proceedings of CRYPTO'85*, 1986, LNCS, vol. 218, pp. 417-426.

[2] N. Koblitz, 1987. "Elliptic curve cryptosystems," *Mathematics of computation*, vol. 48, 1987, pp. 203-209.

[3] M. Mogollan, 2007. "Cryptography and Security Services: Mechanisms and Applications," *Cybertech Publishing*, Hershey, New York. 2007.

[4] H. Moderas, "A scalar multiplication in elliptic curve cryptography with binary polynomial operations in Galois Field," *Master Thesis*. University Malaya, Malaysia. 2009.

[5] H. Pathak and M. Sanghi, "Speeding up computation of scalar multiplication in elliptic curve cryptosystem," *International Journal on Computer Science and Engineering (IJCSE)*, vol. 2, no. 4, 2010, pp. 1024-1028.

[6] Y. Hitchcock, E. Dawson, A. Clark, A. and P. Montague, "Implementing an efficient elliptic curve cryptosystem over GF(p) on a smart card," *ANZIAM J*, vol 44. No. E, 2003, pp. C354-C377.

[7] S. Kumar, "Elliptic Curve Cryptography For Constrained Devices," *Dissertation, Rurh-University Bochum,* 2006.

[8] M. Morales-Sandoval, M. "An Interoperability and Reconfigurable Hardware Architecture for Elliptic Curve Cryptography," *PhD Thesis proposal, National Institute for Astro Physics, Optics and Electronics, Tonantzintla, Puebla, Mexico*, 2006.

[9] M. Morales-Sandoval and C. Feregrino-Uribe, "GF(2m) Arithmetic modules for elliptic curve cryptography," *Proceedings of IEEE International Conference on Reconfigurable Computing and FPGA's*, 2006, pp. 1-8.

[10] Z. Yan, and H. Shi, "Software implementation of elliptic curve cryptography," *International Journal of Network Security*, vol. 7, no. 2, 2008, pp. 157-166.

[11] L. Elmegaard-Fessel, "Efficient Scalar Multiplication and Security Against Power Analysis in Cryptosystems based on the NIST Elliptic Curves Over Prime Field," *Master Thesis, University of Copenhagen, Denmark,* 2006.

[12] H. Cohen, A. Miyaji, and T. Ono, "Efficient elliptic curve exponentiation using mixed coordinates," *Springer-Verlag*, 1998, pp. 51-65.

[13] M. Bednara, M. Daldrup, J. Gathen, J. Shokrollahi, and J. Teich, 2002. "Reconfigurable implementation of elliptic curve crypto algorithms," *Proceedings of the International Parallel and Distributed Processing Symposium (IPDPS' 02),* 2002, pp. 157-164.

[14] J. Lo´pez and R. Dahab, "Improved Algorithms for Elliptic Curve Arithmetic in GF(2n)," *Tech. Report IC-98-39*, October 1998.

[15] H. Cohen, and G. Frey, "Handbook of Elliptic and Hyperelliptic Curve Cryptography," *Discrete Mathematics and Its Application, Chapman & Hall/CRC*, 2006.

[16] A. Higuchi and N. Takagi, "A fast addition algorithm for elliptic curve arithmetic in GF(2n) using projective coordinates," *Information Processing Letter*, vol 76, no. 3, 2000, pp. 101-103.

[17] T. Lange, "A note on Lopez-Dahab coordinates," *http://eprint.iacr.org/2004/323.pdf.*

[18] E. Al-Daoud, "An Improved Public Key Cryptography Based on the Elliptic Curve," *PhD Thesis*, University Putra Malaysia. 2002.

[19] E. Al-Daoud, R. Mahmod, M. Rushdan, and A. Kilicman, "A new addition formula for elliptic curves over GF(2^n)," *IEEE Transactions on Computers*, Vol. 51, 2002, pp. 972-975.

[20] V. S. Dimitrov, L. Imbert, and P. Mishra, "Efficient and secure elliptic curve point multiplication using double-base chains," *LNCS, Springer-Verlag*, vol. 3788, pp. 59-78.

[21] P. Mishra and V. Dimitrov, "Efficient quintuple formulas for elliptic curves and efficient scalar multiplication using multibase number representation," *LNCS, Springer-Verlag*, vol. 4779, 2007, pp. 390-406.

[22] N. Meloni, "Fast and Secure Elliptic Curve Scalar Multiplication Over Prime Fields Using Special Addition Chains," *http://eprint.iacr.org/2006/216.pdf.*

[23] S. Moon, "A binary redundant scalar point multiplication in secure elliptic curve cryptosystems," *International Journal of Network Security*, Vol. 3, no. 2, 2006, pp. 132–137.

[24] H. Edwards, "A normal form for elliptic curves," *Bulletin of the AMS*, Vol. 44, no. 3, 2007, pp. 393-422.

[25] H. Hisil, G. Carter, and E. Dawson, E. "New Formulae for efficient elliptic curve arithmetic," *LNCS, Springer-Verlag*, Vol. 4859, 2007, pp. 138-151.

[26] M. Y. Sharifah, "New Signed Digit {0,1,3}-NAF Scalar Multiplication Algorithm for Elliptic Curve Over Binary Field," *PhD Thesis*, University Putra Malaysia. 2011.