IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received August 18, 2021, accepted September 13, 2021, date of publication September 17, 2021,
date of current version September 27, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3113792

Efficient Septuple Formula for Elliptic Curve and Efficient Scalar Multiplication Using a Triple-Base Chain Representation

SHUANGGEN LIU AND LIJUAN ZHANG
School of Cyberspace Security, Xi'an University of Posts and Telecommunications, Xi'an 710121, China

Corresponding author: Shuanggen Liu (liusgxupt@163.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 61872058, and in part by the Key Research and Development Program of Shaanxi under Grant 2021NY-211.

ABSTRACT Because of shorter key and higher security, elliptic curve cryptosystem has attracted people's attention and it is widely used in various fields, such as wireless sensor networks. Scalar multiplication is one of the most basic and critical links in the realization of elliptic curve public key systems, and its operation efficiency directly affects the implementation efficiency of the entire cryptographic system. To improve speed up the efficiency of the scalar multiplication algorithm, we express k for scalar multiplication kP by using triple-base chain representation of the scalar using $\{2, 3, 7\}$ as basis of the triple-base chain in this paper. However, the efficiency of scalar multiplication is not only related to the length of representation but also the numbers and costs of doubling, tripling, septupling and addition. Therefore, we improve septuple formula of the elliptic curve by using Co_Z operation, which costs decreased by 8.3%. Due to the high redundancy of the triple-base chain representation, the algorithm can resist side channel attacks. The experimental results show that the proposed algorithm compared with that of other scalar multiplication algorithms, it requires less cost.

INDEX TERMS Co_Z operation, elliptic curve cryptography, triple-base number system, triple-base chain, scalar multiplication, septuple.

I. INTRODUCTION

A. BACKGROUND

With the rapid development of wireless communication technology, wireless sensor networks of low-cost, low-power, and multi-functional are used in many civilian fields, such as environmental and ecological monitoring, health monitoring, home automation, and traffic control. However, users have high requirements for the security capabilities of wireless sensor networks in these application environments. Therefore, the introduction of encryption technologies that can ensure information security has become a major research hotspot at the moment, and it is vital to the development of wireless sensor networks. However, elliptic curve cryptosystem has attracted people's attention with shorter key and higher security, and it is widely used in various fields, such as wireless sensor networks. Elliptic curve cryptography (ECC) has been a research topic for many scholars since it was first discovered by Miller [1] and Koblitz [2] in 1985. In particular, substantial research has been conducted on how to improve the overall computing efficiency of elliptic curves and how to resist side-channel attacks. The security of ECC depends on the computational intractability of the elliptic curve discrete logarithm problem (ECDLP). In contrast to the discrete logarithm problem in finite fields and the integer factorization problem, there is no known sub-exponential time algorithm to solve the discrete logarithm problem on a well-chosen elliptic curve. In terms of security, compared with RSA [3] public key schemes and ElGamal [4] public key schemes, the elliptic curve cryptosystem provides higher security strength per bit. For instance, the security strength provided by the 160-bit key length elliptic curve cipher is equivalent to the security strength provided by the 1024-bit key RSA password. Therefore, ECC is simpler and more universal in implementation than other public key cryptographic systems, and its application is increasingly widespread. ECC is particularly suitable

The associate editor coordinating the review of this manuscript and approving it for publication was Aniello Castiglione.

for application environments with limited storage or computing resources, such as smart cards and PDAs. When using the elliptic curve cryptosystem for encryption, decryption, digital signature and digital signature verification [5]–[7], the most time-consuming operation is scalar multiplication, and its operation efficiency effects the implementation efficiency of the elliptic curve cryptosystem.

Scalar multiplication is denoted by kP, where k is a scalar and P is a point on an elliptic curve. How to accelerate scalar multiplication is a challenging task, especially in resource-constrained environments. Methods to improve the efficiency of scalar multiplication can be considered based on the following aspects: on one hand, k can be represented in binary form [8], or non-adjacent form(NAF) [9]; on the other hand, the cost of point addition and doubling operation is decreased. In order to describe the cost of field operations, we will respectively denote the cost of field inversion, field multiplication, field squaring and field cube by using I, M, S and C. We shall always ignore the costs of field addition, field subtraction and multiplication by small constants. For large prime field, we assume that the quantity relationship among them is $S/M = 0.8$, as is customary in software implementation.

In recent years, the double-base or multi-base representation of integer k has researched by many scholars and has made great progress. The double-base number system (DBNS) representation of integers was first proposed by Dimitrov and Cooklev in 1995 [10], where an integer K is represented as $k = \Sigma_{i=1}^{l} 2^{a_i} 3^{b_i}$. In 1997, Dimitrov *et al.* present a rigorous theoretical analysis of the main properties of a double base number system with using base 2 and 3 and emphasize the sparseness of the representation in particular. And its potential areas of application is computation of modular exponentiations in cryptography [11]. After that, Dimitrov *et al* used DBNS to improve the performance of modular exponentiations and gave an theoretical about DBNS, which the greedy algorithms terminates after steps $k = o(\frac{logx}{loglogx})$ [12]. Due to the DBNS seems to be not that efficient in the case of a randomly chosen base point. In order to overcome this problem, Dimitriv *et al.* [13] introduced the concept of double-base chain(DBC), which as a special case of double-base number systems(DBNSs), requires the restrictions $b_1 \geq b_2 \geq \ldots \geq b_l \geq 0$ and $t_1 \geq t_2 \geq \ldots \geq t_l \geq 0$. The algorithm accelerate scalar multiplication by fewer point additions and can be protected against simple and differential side-channel analysis by using side-channel atomicity and classical randomization techniques. However, the new restriction conditions increasing the number of double-base chain terms. To overcome this problem, a triple-base number system(TBNS) appeared.

Triple-base chain(TBC) as a special case of triple-base number systems(TBNSs) have the characteristics of shorter scalar representation length and fewer non-zero bits. In 2007, Mishra and Dimitrov [14] presented efficient formulas for point quintupling and introduced a triple-base number system (TBNS) for computing scalar multiplication more efficiently based on $\{2, 3, 5\}$. In 2013, Wei *et al.* [15], [16] proved the number of TBNSs and indicated that the sub-linear bound is still valid. Meloni and Hasan [17] introduced a class of constrained DNBSs that were restricted to exponents of base $\{2, 3\}$ in 2015. Yunqi *et al.* [18] proposed a constrained TBNS on the basis of a constrained TBNS based on $\{2, 3, 5\}$. After that, Yunqi *et al.* [19] proved that the upper bound on expansion length of a constrained TBNS is still sub-linear. This result provides a more practical boundary of the TBNS to accelerate scalar multiplication. However, since the length of the TBC representation has not been visibly became short, but the number of points doublings, point triplings, and point septuplings increased greatly, the computational overhead of scalar multiplication may be very large. Therefore, it is of great significance for TBC to optimize to reduce the computational complexity of the underlying filed by improving operation on the bottom layer such as field inverse, multiplication and so on. Table 1 shows the operations needed in this paper.

TABLE 1. Cost of elliptic curve point operations in F_P.

Operation	costs
P+Q (ADD)	$4M + 2S(5.6M)$
2P (DBL)	$1M + 5S(5M)$
3P (TPL) [28]	$7M + 6S(11.8M)$
5P (QPL) [23]	$12M + 13S(22.4M)$

B. CONTRIBUTION

In the present work, our main contributions are as follows:

First, we improve septuple formula of the elliptic curve by using Co_Z operation, which costs decreased by 8.3%.

Second, an efficient scalar multiplication algorithm of a point P on an elliptic curve is proposed using TBC representation of the scalar using $\{2, 3, 7\}$ as basis of the TBC.

Third, we apply the improving algorithm to wireless sensor networks. It can effectively improve the quality of wireless sensor network broadcast authentication service.

C. ORGANIZATION

The rest of this work is organized as follows: in section II, we provide the background on the ECC and MBRS. In section III, we improve the septuple operation formula. Then, we introduce a fast TBC scalar multiplication algorithm with the scalar using $\{2, 3, 7\}$ as basis of the TBC in section IV. In section V, We analyse the efficiency of the group operation and the efficiency of the TBC scalar multiplication algorithm. And we describes the application of the algorithm. Finally, in section VI concludes the paper.

II. MATHEMATICAL BACKGROUND

In this section, we give a brief review of the basic knowledge used in the paper.

A. ELLIPTIC CURVE CRYPTOGRAPHY

Definition 2.1: An elliptic curve E over a finite field K is defined by the following equation:

$$E : y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6 \tag{1}$$

where $a_1, a_2, a_3, a_4, a_6 \in K$ are such that, for each point (x, y) on E, the particle derivatives do not vanish simultaneously.

The Weierstrass equation(1) can be highly dependent on the characteristics of K and can be simplified by applying admissible changes in variables. If the characteristic of K is not equal to 2 or 3, equation (1) can be simplified as:

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

where $a, b, \in K$ and $\Delta = 4a^3 + 27b^2 \neq 0$. When the characteristic of K is equal to 2, we use the nonsupersingular form of an elliptic curve and the Weierstrass equation(1) is represented as:

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

where $a, b \in K$ and $\Delta \neq 0$. In this paper, we only consider curves defined over a prime finite field ($K = F_p$) of characteristics greater than 3. The set of points $E(K)$ defined over K forms an abelian group. Elliptic curve group law computation has been a very active research area. If we assume that $P_1 = (x_1, y_1)$ and $P_2 = (x_2, y_2)$ are arbitrary points on the elliptic curve and P_1, P_2 are not equal, point addition (ADD) is defined by $P_3 = P_1 + P_2$ and point doubling (DBL) is defined by $P_3 = 2P_1$. The equations for ADD and DBL are as follows: ADD operation $P_3 = P_1 + P_2 = (x_3, y_3)$

$$\begin{cases} x_3 = (\dfrac{y_1 - y_2}{x_1 - x_2})^2 - x_1 - x_2, \\ y_3 = (\dfrac{y_1 - y_2}{x_1 - x_2})(x_1 - x_3) - y_1. \end{cases} \tag{4}$$

DBL operation $P_3 = 2P_1 = (x_3, y_3)$.

$$\begin{cases} x_3 = (\dfrac{3x_1^2 + a}{2y_1})^2 - 2x_1, \\ y_3 = (\dfrac{3x_1^2 + a}{2y_1})(x_1 - x_3) - y_1. \end{cases} \tag{5}$$

B. THE JACOBIAN COORDINATE ON OPERATION

In the affine coordinate system, the formulas of point addition and point doubling on the elliptic curve in F_p involve an inverse operation, which is computationally expensive. In order to avoid inversion, Jacobian projective coordinates we would be introduced: for Jacobian coordinates, we set $x = \dfrac{X}{Z^2}$ and $y = \dfrac{Y}{Z^3}$, giving the equation

$$E_J : Y^2 = X^3 + aXZ^4 + bZ^6. \tag{6}$$

The point addition formula is as follows. Let $P = (X_1, Y_1, Z_1)$, $Q = (X_2, Y_2, Z_2)$ and $P+Q = R = (X_3, Y_3, Z_3)$.

$$\begin{cases} X_3 = R^2 + G - 2V, \\ Y_3 = R(V - X_3). \\ Z_3 = ((Z_1 + Z_2)^2 - I_1 - I_2)H. \end{cases} \tag{7}$$

where, $R = 2(K_1 - K_2)$, $G = FH$, $V = U_1F$, $K_1 = Y_1J_2$, $K_2 = Y_2J_1$, $F = (2H)^2$, $H = U_1 - U_2$, $U_1 = X_1I_2$, $U_2 = X_2I_1$, $J_1 = I_1Z_1$, $J_2 = I_2Z_2$, $I_1 = Z_1^2$, $I_2 = Z_2^2$ [25].

The point doubling formula is as follows: Let $P = (X_1, Y_1, Z_1)$, and $2P = R = (X_3, Y_3, Z_3)$.

$$\begin{cases} X_3 = A^2 - 2B, \\ Y_3 = A(B - X_3) - 8E^2. \\ Z_3 = C. \end{cases} \tag{8}$$

where $A = 3D + aF^2$, $B = 2((X_1 + E)^2 - D - E^2)$, $C = ((Y_1 + Z_1)^2 - E - F)$, $D = X_1^2$, $E = Y_1^2$, $F = Z_1^2$

According to the above formulas, under Jacobian coordinates, point addition costs $11M + 5S$ and point doubling costs $1M + 8S + 1c$, where c denotes the cost of a multiplication by curve parameter a.

Moreover, it is important to attention that it has been proposed that the parameter can be fixed at $a = -3$ for efficiency purposes. In fact, most curves recommended by public-key standards use $a = -3$, which has been shown to not impose significant restrictions on the cryptosystem. In this case, the cost of point doubling is reduced to only $3M + 5S$ [20], [21].

C. MULTI-BASE REPRESENTATION OF AN INTEGER

1) DOUBLE-BASE REPRESENTATION OF AN INTEGER

In this section, we introduce the concept of a DBNS; the details can be found in [22].

Definition 2.2 (S-Integer): Given a set of primes S, an S-integer is a positive integer whose prime factors all belong to S.

Definition 2.3 (Double-Base Number System): Given p, q, two relatively prime positive integers, the DBNS is a representation scheme into which every positive integer n is represented as the sum or difference of p, q-integers, i.e., numbers of the form $p^a q^b$:

$$n = \Sigma_{i=1}^{l} s_i p^{a_i} q^{b_i}, \tag{9}$$

with $s_i \in \{-1, 1\}$ and $a_i, b_i \geq 0$. The size, or length, of a DBNS expansion is equal to the number of terms l in equation (10). In the following, we consider only expansions of n as sums of 2, 3-integers; i.e., DBNS with $p = 2, q = 3$.

This representation is highly redundant regardless of whether the integer K is expanded signed or unsigned. For example, if we assume only unsigned double-base representations, we can prove that the DBNS representations of 10 have exactly 5 different form, the DBNS representations of 100 have exactly 402 different form, the DBNS representations of 1, 000 have exactly 1, 295, 579 different form, etc.

2) MULTI-BASE REPRESENTATION OF AN INTEGER

In this section, we give the concept of TBNS and TBC the details can be found in [14], [24].

TBC has higher redundancy as an extension of the DBC. Compared with other scalar representations, its chain length is shorter and the number of nonzero bits is less, which can

effectively reduce the amount of required scalar multiplication. TBNS first proposed by Mishra and Dimitrov [14]. Let K be a large integer and $B = b_1, \ldots, b_l$ be a set of small integers. That is, $k = \Sigma_{i=1}^{l} s_i b_1^{e_{i_l}} \ldots b_l^{e_{i_l}}, s_i \in \{-1, 1\}$ where s_i is the sign bit, and the integer l is the length of the expression, then k is called the multi-base representation using the basis set B.

Definition 2.4 (TBNS): Given three relatively prime positive integers p_1, p_2, p_3, every integer k is represented as

$$k = \Sigma_{i=1}^{l} s_i p_1^{a_i} p_2^{b_i} p_3^{e_i}, \tag{10}$$

where $s_i \in \{-1, 1\}$, l is the chain length. We will refer to terms of the form $p_1^{a_i} p_2^{b_i} p_3^{e_i}$, as 3-integers. A general triple-base representation, although very short, is not suitable for a scalar multiplication algorithm. So we are interested in a special representation with restricted exponents, such as TBC.

Definition 2.5 (TBC): Given three relatively prime positive integers p_1, p_2, p_3, every integer k is represented as

$$k = \Sigma_{i=1}^{l} s_i p_1^{a_i} p_2^{b_i} p_3^{e_i}, \tag{11}$$

where $s_i \in \{-1, 1\}$, l is the chain length, $a_1 \geq a_2 \geq \ldots \geq a_l \geq 0, b_1 \geq b_2 \geq \ldots \geq b_l \geq 0$ and $e_1 \geq e_2 \geq \ldots \geq e_l \geq 0$.

For example, in [23], if $k = 21,962$ then k can be represented by following.

$$\begin{aligned} k &= 21962 = 2^5 3^3 5^3 + 2^3 3^2 5^1 + 2 \\ &= 2(2^2 3^2 5^1 (2^2 3^1 5^1 + 1) + 1) \end{aligned} \tag{12}$$

Algorithm 1 uses a greedy algorithm, which can convert an integer to a triple-base representation.

Algorithm 1 Greedy Algorithm to Compute DBNS Representations

Input: A positive k, $max2$, $max3$

Output: the sequence$(s_i, a_i, b_i)_{i>0}$, such that $k = \Sigma_{i=1}^{l} s_i 2^{a_i} 3^{b_i}$

$s_1 \leftarrow 1, i \leftarrow 1$
for b = 0 to $max2$, t = 0 to $max3$ **do**
  $z = T[a, b]$, the best approximation of K
  print (s, a, b)
  $max2 \leftarrow a, max3 \leftarrow b$
  if $k < z$ **then**
    $s \leftarrow -s$
  else
    $s \leftarrow s$
  end if
  $k \leftarrow |k - z|$
end for
end

III. OPTIMIZED 7P METHOD ON CO-Z OPERATION

A. ADDITION FORMULA ON CO-Z OPERATION

The Co-Z operation was first proposed by Meloni [25]. Its main idea is that two points with different Z coordinates on the elliptic curve use the same Z coordinate through transformation during calculation. Let $P = (X_1, Y_1, Z)$, $Q = (X_2, Y_2, Z)$ and $P_1 + P_2 = (X_3, Y_3, Z_3)$.

$$\begin{cases} X_3 = U - T_1 - T_2, \\ Y_3 = (Y_1 - Y_2)(T_1 - X_3) - Y_1(T_1 - T_2). \\ Z_3 = Z(X_1 - X_2). \end{cases} \tag{13}$$

where, $W = (X_1 - X_2)^2$, $T_1 = X_1 W$, $T_2 = X_2 W$, $U = (Y_1 - Y_2)^2$.

This operation is called as the ZADD operation. The key observation in Equation (11) is that the computation of $R = P + Q$ yields for free an equivalent representation for input point P with its Z-coordinate equal to that of output point R, namely $(X_1(X_1 - X_2)^2 : Y_1(X_1 - X_2)^3 : Z_3) \sim P$.

From Equation (11), the unified Z-coordinate transformation has been calculated in the Co_Z point addition operation, so no extra calculations are needed. At this time, the point addition costs $5M + 2S$. If Z-coordinate of P and Q are equal to 1, the corresponding point addition costs $4M + 2S$.

B. OPTIMIZED 7P METHOD ON CO-Z OPERATION

For the first time, G. N. Purohit proposed a method of calculating 7P directly in affine coordinates over a binary field using the idea of the following division polynomial [24]. Later, Lai Zhongxi and others used the idea of transforming inversion into multiplication and proposed an algorithm for calculating 7P in affine coordinates over a large prime field [26]. And operation cost of setupling is $I + 18M + 12S$. Longa *et al* obtained operation cost of setupling that is $13M + 18S$ by presenting new improvements in the point operation formulae [27].

In this section, we further improve the point septupling formula for elliptic curves over large prime fields. We use the Co_Z operation to compute setupling in Jacobian coordinates. The calculation process of septupling is described in detail below. Let P be (X_1, Y_1, Z_1) be a point on elliptic curve given by equation (13) (14)(15)(16).

Firstly, when $Z_1 = 1$, we use equation (8) to give the result of the point $2P$ calculation. let $2P = (X_3, Y_3, Z_3)$ we can get that,

$$\begin{cases} X_3 = A^2 - 2B, \\ Y_3 = A(B - X_3) - 8E^2. \\ Z_3 = 2Y_1. \end{cases} \tag{14}$$

where $A = 3D, B = 2((X_1 + E)^2 - D - E^2), D = X_1^2, E = Y_1^2$.

Secondly, in the same way, we use equation (8) to give the result of the point $4P$ calculation. When $Z_1 = 1$, let $4P = 2(2P) = (X_4, Y_4, Z_4)$ we can get that,

$$\begin{cases} X_4 = {A_1}^2 - 2B_1 \\ Y_4 = A_1(B_1 - X_4) - 8E_1^2 \\ Z_4 = C \end{cases} \tag{15}$$

where $A_1 = 3(X_3 - 4E^2)(X_3 + 4E^2), B_1 = 2((X_3 + E_1)^2 - D_1 - E_1^2), C_1 = ((Y_3 + Z_3)^2 - E_1 - 4E^2), D_1 = X_3^2, E_1 = Y_3^2, F_1 = Z_3^2 = 2Y_1$.

Thirdly, we use equation (11) to give the result of the point $3P$ calculation. When $Z_1 = 1$, $P_1^{(1)}$ and $2P$ have the same Z-coordinate. let $P = (X_1, Y_1, 1)$, $P_1^1 \sim (X_1^{(1)}, Y_1^{(1)}, Z_1^{(1)}) = (4X_1Y_1^2, 8Y_1^4, 2Y_1)$, and $3P = 2P + P = (X_5, Y_5, Z_5)$ we can get that,

$$\begin{cases} X_5 = U' - T_1' - T_2', \\ Y_5 = (Y_1^{(1)} - Y_3)(T_1' - X_5) - Y_1^{(1)}(T_1' - T_2'). \\ Z_5 = Z_3(X_1^{(1)} - X_3). \end{cases} \tag{16}$$

where, $W' = (X_1^{(1)} - X_3)^2$, $T_1' = X_1^{(1)}W'$, $T_2' = X_3W'$, $U' = (Y_1^{(1)} - Y_3)^2$.

Finally, we continue to use equation (11) to give the result of the point setupling. Then, using Co_Z point addition between $4P$ and $3P$ can get the formula of $7P$, At this time, $4P \sim P^{(2)} = (X_3^{(2)}, Y_3^{(2)}, Z_3^{(2)}) = (X_5(2Y_3Z_3)^2, Y_5(2Y_3Z_3)^3, Z_5(2Y_3Z_3)$, then, $7P = 3P + 4P = (X_6, Y_6, Z_6)$. See Algorithm 2

$$\begin{cases} X_6 = U'' - T_1'' - T_2'', \\ Y_6 = (Y_3^{(2)} - Y_5)(T_1'' - X_6) - Y_3^{(2)}(T_1'' - T_2''). \\ Z_6 = Z_5(X_3^{(2)} - X_5). \end{cases} \tag{17}$$

Algorithm 2 Greedy Algorithm to Convert a Triple-Base Representation

Input: A positive k, $maxa_i$, $maxb_i$, $maxe_i$, with $s_i \in \{-1, 1\}$, l is the chain length, $a_1 \geq a_2 \geq \ldots \geq a_l \geq 0$, $b_1 \geq b_2 \geq \ldots \geq b_l \geq 0$ and $e_1 \geq e_2 \geq \ldots \geq e_l \geq 0$.
Output: the sequence$(s_i, a_i, b_i, e_i)_{i>0}$, such that $k = \Sigma_{i=1}^{l} s_i p_1^{a_i} p_2^{b_i} p_3^{e_i}$

$s_1 \leftarrow 1, i \leftarrow 1$
while $k > 0$ **do**
  find $\{p_1, p_2, p_3\}$-integer $k = p_1^{a_i} p_2^{b_i} p_3^{e_i}$ the best approximation of k
  print (s, a, b, e)
  $a_i \leftarrow a, b_i \leftarrow b, e_i \leftarrow e$
  if $k < z$ **then**
    $s \leftarrow -s$
  else
    $s \leftarrow s$
  end if
  $k \leftarrow |k - z|$
end while
end

The computational complexity of each step is analyzed in Table 2. According to the costs of point doubling and point addition, it costs $12M + 15S$.

IV. NEW SCALAR MULTIPLICATION ALGORITHM OF TBC OF USING {2, 3, 7} AS BASIS

A. A TBC REPRESENTATION OF USING {2, 3, 7} AS BASIS

According to the algorithm 2 in section II, by TBC representation of k in this paper, we mean a representation of the form.

$$k = \Sigma_{i=1}^{l} s_i 2^{a_i} 3^{b_i} 7^{r_i}, \tag{18}$$

Algorithm 3 Seven-Tuple Formula on Co-Z Operation

Input: $P_1 = (X_1, Y_1, Z_1)$
Output: $7P = (X_6, Y_6, Z_6)$

$W'' \leftarrow (X_3^{(2)} - X_5)^2$
$T_1'' \leftarrow X_3^{(2)} W''$
$T_2'' \leftarrow X_5 W''$
$U'' \leftarrow (Y_4 - Y_5)^2$.
$X_6 \leftarrow U'' - T_1'' - T_2''$
$Y_6 \leftarrow (Y_3^{(2)} - Y_5)(T_1'' - X_6) - Y_3^{(2)}(T_1'' - T_2'')$
$Z_6 \leftarrow Z_5(X_3^{(2)} - X_5)$
return (X_6, Y_6, Z_6)
end

with $s_i \in \{-1, 1\}$, l is the chain length, $a_1 \geq a_2 \geq \ldots \geq a_l \geq 0$, $b_1 \geq b_2 \geq \ldots \geq b_l \geq 0$ and $r_1 \geq r_2 \geq \ldots \geq r_l \geq 0$. Obviously, the exponents a_i, b_i and r_i form three separate monotonic decreasing sequence.

Algorithm 4 Greedy Algorithm to Compute a TBC Representation of Using {2, 3, 7} as Basis

Input: A positive k, $max2$, $max3$, $max7$, with $s_i \in \{-1, 1\}$, l is the chain length, $a_1 \geq a_2 \geq \ldots \geq a_l \geq 0$, $b_1 \geq b_2 \geq \ldots \geq b_l \geq 0$ and $r_1 \geq r_2 \geq \ldots \geq r_l \geq 0$.
Output: the sequence$(s_i, a_i, b_i, r_i)_{i>0}$, such that $k = \Sigma_{i=1}^{l} s_i 2^{a_i} 3^{b_i} 7^{r_i}$

$s_1 \leftarrow 1, i \leftarrow 1$
while $k > 0$ **do**
  find $\{2, 3, 7\}$-integer $k = 2^a 3^b 7^r$ the best approximation of k
  print (s, a, b, r)
  $a_i \leftarrow a, b_i \leftarrow b, r_i \leftarrow r$
  if $k < z$ **then**
    $s \leftarrow -s$
  else
    $s \leftarrow s$
  end if
  $k \leftarrow |k - z|$
end while
end

Algorithm 4 shows that if $k = 21962$, k is represented by following:

$$k = 21962 = 2^6 7^3 + 2^3 + 2 = 2(2^2(2^3 7^3 + 1) + 1) \tag{19}$$

B. NEW SCALAR MULTIPLICATION ALGORITHM OF TBC OF USING {2, 3, 7} AS BASIS

According to Algorithm 3, the scalar k can be expressed as a TBC. we fastly compute to scalar multiplication by using a recursive formula using following equation for recursive calculations for purpose of implementing the scalar multiplication.

$$k_1 = 1, \quad k_i = k = 2^x 3^v 7^u + s_i, \quad s_i \in \{-1, 1\} \tag{20}$$

TABLE 2. Cost of elliptic curve point setupling operation in F_p.

Step	Knomn Terms	Computation
D	X_1^2	$1S$
E	Y_1^2	$1S$
A	$3D$	$-$
B	$2[(X_1+E)^2-D-E^2]$	$2S$
X_3	A^2-2B	$1S$
Y_3	$A(B-X_3)-8E^2$	$1M$
Z_3	$2Y_1$	$-$
D_1	X_3^2	$1S$
E_1	Y_3^2	$1S$
A_1	$3(X_3-4E^2)(X_3+4E^2)$	$1M$
B_1	$2[(X_3+E_1)^2-D_1-E_1^2]$	$2S$
C_1	$[(Y_3+Z_3)^2-E_1-4E^2]$	$1S$
X_4	$A_1^2-2B_1$	$1S$
Y_4	$A_1(B_1-X_4)-8E_1^2$	$1M$
Z_4	C	$-$
$U^{'}$	$(Y_1^{(1)}-Y_3)^2$	$1S$
$T_1^{'}$	$X_1^{(1)}W^{'}$	$1M$
$T_2^{'}$	$X_3W^{'}$	$1M$
$W^{'}$	$(X_1-X_3)^2$	$1S$
X_5	$U^{'}-T_1^{'}-T_2^{'}$	$-$
Y_5	$(Y_1^{(1)}-Y_3)(T_1^{'}-X_5)-Y_1^{(1)}(T_1^{'}-T_2^{'})$	$2M$
Z_5	$Z_3(X_1^{(1)}-X_3)$	$1M$
$U^{''}$	$(Y_3^2-Y_5)^2$	$1S$
$T_1^{''}$	$X_3^{(2)}W^{''}$	$1M$
$T_2^{''}$	$X_5W^{''}$	$1M$
$W^{''}$	$(X_3^{(2)}-X_5)^2$	$1S$
X_6	$U^{''}-T_1^{''}-T_2^{''}$	$-$
Y_6	$(Y_3^{(2)}-Y_5)(T_1^{''}-X_6)-Y_3^{(3)}(T_1^{''}-T_2^{''})$	$2M$
Z_6	$Z_5(X_3^{(2)}-X_5)$	$1M$
$Total$	$-$	$12M+15S$

TABLE 3. Method of calculating 21962P in different iterations.

i	k	s	x	u	v	cost
1	$k_1=1$	1	0	0	0	$-$
2	$k_2=2^3 7^3+k_1$	1	3	0	3	$92.6M$
3	$k_3=2^2k_2+1$	1	2	0	0	$108.2M$
4	$k_4=2k_3$	0	1	0	0	$113.2M$

where x is the difference of two consecutive binary exponents, v is the difference of two consecutive ternary exponents and u is the difference of two consecutive septenary exponents. we use the algorithm4 to implement it.

From Algorithm 4, it can be concluded that the amount number of points doubling, points tripling, points setupling and points addition. respectively is x, v, u and $l-1$. And TBC does not require any pre-computations but in this method the expansion of the scalar reduces the cost of the scalar multiplication making it faster. This algorithm requires $l-1$ iterations, and the calculation amount of costing is denoted as W

$$W = X * DBL + v * TBL + u * SPL + (l-1) * ADD \quad (21)$$

For example, from equation 18, if $k = 21962$, the scalar multiplied by kP is

$$kP = 21962 = 2(2^2(2^3 7^3 P + P) + P) \quad (22)$$

However, from equation 12, if $k = 21962$, the scalar multiplied by kP is

$$k = 21962 = 2(2^2 3^2 5^1 (2^2 3^1 5^1 + P) + P) \quad (23)$$

Table 3 is method of calculating 21962P in different iterations and the scalar using {2, 3, 7} as basis of the TBC. It costs $113.2M$. Table 4 is method of calculating 21962P in different iterations and the scalar using {2, 3, 5} as basis of the TBC. It costs $117M$. Compared with using {2, 3, 5} as basis of the TBC for 21962P, the total cost of the scalar using {2, 3, 7} as basis decrease 3.3%.

V. ANALYSIS OF ALGORITHM

In this section, we conduct software simulation to compare the efficiency of our algorithm with that of existing methods. We analyse two aspects: the efficiency of group operations and the overall operating efficiency of the scalar multiplication algorithm.

TABLE 4. Method of calculating 21962P in different iterations.

i	k	s	x	u	v	cost
1	$k_1 = 1$	1	0	0	0	–
2	$k_2 = 2^2 3^1 5^1 + k_1$	1	2	1	1	$49.8M$
3	$k_3 = 2^2 3^2 5^1 k_2 + 1$	1	2	2	1	$111.4M$
4	$k_4 = 2k_3$	0	1	0	0	$117M$

TABLE 5. Cost of elliptic curve point setupling in F_P ($I = 10M$, $S = 0, 8M$).

Algorithm	I	S	M	Cost($\approx M$)
literature [26]	1	12	18	37.6
literature [27]	-	15	14	26
Ours	-	15	12	24

Algorithm 5 New Scalar Multiplication Algorithm of TBC by Using {2, 3, 7} as Basis

Input: A integer $k = \Sigma_{i=1}^{l} s_i 2^{b_i} 3^{t_i} 7^{r_i}$, such that $b_1 \geq b_2 \geq \ldots \geq b_l \geq 0$, $t_1 \geq t_2 \geq \ldots \geq t_l \geq 0$, and $r_1 \geq r_2 \geq \ldots \geq r_l \geq 0$ and a point P on elliptic curve

Output: $Q = kP \in E(F_P)$

$Q \leftarrow s_1 P$
for $i = 2, \ldots, l-1$ **do**
  $x \leftarrow b_i - b_{i-1}$
  $v \leftarrow t_i - t_{i-1}$
  $u \leftarrow r_i - r_{i-1}$
  for $i = l-1$ to i **do**
    for $i = m-1$ to x **do**
      $Q \leftarrow 2Q$
    end for
    for $i = n-1$ to v **do**
      $Q \leftarrow 3Q$
    end for
    for $i = h-1$ to u **do**
      $Q \leftarrow 7Q$
    end for
  end for
  if $s_i = 1$ **then**
    $Q \leftarrow Q + P$
  else
    $Q \leftarrow Q - P$
  end if
  $i = i + 1$
end for
return Q

A. GROUP OPERATION ANALYSIS

Group operation is the underlying field operation of the elliptic curve scalar multiplication algorithm, which plays a key role in the efficiency of the entire system. The comparison of the total computational cost of different setuple formula is shown in Fig. 1.

Table 5 shows that detailed data of computational cost of the improved setuple formula and the previous formulas. Compared with that of [26], the cost of calculation is reduced to $3.6M + I$, and the cost decreased by 56.7%. Compared with [27], the cost of calculation is reduced to $2M$, and the cost decreased by 8.3%. And the cost of point doubling, point tripling and point addition, in this paper, is shown in Table 5.

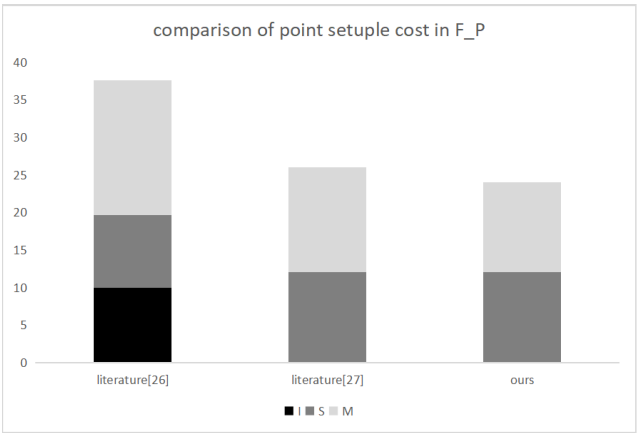

FIGURE 1. Comparison of point setuple cost in F_P.

B. PERFORMANCE ANALYSIS OF ALGORITHMS

In order to analyze the performance of the scalar multiplication algorithm proposed in this paper. Let us compare the performance of the proposed scalar multiplication scheme to some of the schemes existing in the literature. The experiments were conducted on the elliptic curve recommended by the National Institute of Standards and Technology (NIST) that it is NIST B-160, and the size of the large prime field was selected as 160-bit. For each curve and each set of parameters, I. We generate 10,00 pseudo random integers in $\{0, \ldots, 2^{160} - 1\}$. II. The integers are converted into a TBC representations. III. We calculate the costs for the algorithm in terms of scalar multiplication. The experimental environment is: the hardware environment is Intel (R) Core (TM) i5 CPU, the installed memory is 16 GB, the software environment is windows operating system, and the algorithms are implemented by using JAVA language.

In the section, under in the case of different fields such as large prime fields and over binary fields, our proposed

TABLE 6. Cost comparison of 160-bit scalar multiplication in different field.

Algorithm	proposed	field	cost
MMNR $\{2,3,7\}$	[14]	F_{2^m}	$1341M$
Our algorithm$\{2,3,7\}$	this paper	F_p	$1236M$

TABLE 7. Cost comparison of 160-bit scalar multiplication in same field.

Algorithm	proposed	field	cost
DBC $\{2,3\}$	[13]	F_p	$1863M$
CTBC New Algorithm 1 $\{2,3,5\}$	[18]	F_p	$1437.9M$
CTBC New Algorithm 2 $\{2,3,5\}$	[18]	F_p	$1562.2M$
Our algorithm$\{2,3,7\}$	this paper	F_p	$1236M$

algorithm compared with the MBNR proposed algorithm in [14] for the same field size. See table 6.

Table 6 shows that when the key length is 160- bits and in the case of different fields such as large prime fields and over binary fields, the total cost of the proposed algorithm in this paper is 105M lower than that of MMNR. And the efficiency of the algorithm is increased by 8.5%. As a result, in the case of different fields such as large prime fields and over binary fields and the same field size, the scalar multiplication algorithm we propose is more efficient.

Second, under in the case of the same fields such as large prime fields, our proposed algorithm compared with the DBC proposed algorithm [13], the CTBC proposed algorithm in [18] for the same field size. See table 7.

Table 7 shows that when the key length is 160- bits and in the case of the same such as large prime fields, the total cost of the proposed algorithm in this paper is 627M lower than that of DBC, 201.9M lower than that of CTBC New Algorithm 1, and 326.2M lower than that of CTBC New Algorithm 2. And the efficiency of the algorithm is increased by 50.7%, 16.3% and 26.4%. As a result, in the case of the same fields such as large prime fields and the same field size, the scalar multiplication algorithm we propose is more efficient.

C. APPLICATION

At present, because wireless sensor networks are deployed in hostile environments, broadcast authentication as a fundamental security service. But the slow signature verification in existing schemes makes high energy consumption and long verification delay for broadcast authentication. However, the ECC uses smaller parameters than other cryptosystems such as RSA and DSA, it can obtain faster processing speed, lower power consumption, and save memory and bandwidth. Therefore, ECC is suitable for resource-constrained devices such as wireless sensor networks. So, we apply the improving algorithm to wireless sensor networks. It can effectively improve the quality of wireless sensor network broadcast authentication service.

VI. CONCLUSION

The elliptic curve cryptographic algorithm is one of the most widely used public key cryptographic algorithms, and the performance of the scalar multiplication operation is key to its application. In this paper we have presented efficient scalar multiplication algorithms. First, we have introduced an innovative methodology to derive operations of the septuple formula by applying the special addition with identical z-coordinate to the setting of over prime fields. The new septuple formula are shown to be faster than operations of previous formulae. Second, we have proposed a new algorithm for using TBC representation of an integer and combining with the scalar multiplication. The purpose is to accelerate the scalar multiplication algorithm on elliptic curve cryptography and the scalar uses $\{2,3,7\}$ as basis of the TBC. In the future, we will continue to work on more efficient elliptic curve scalar multiplication algorithms. The triple-base chain is more redundant than the double-base chain. Therefore, to improve the efficiency of the elliptic curve scalar multiplication algorithm, future work will attempt to find the optimal triple-base chain.

REFERENCES

[1] V. S. Miller, "Use of elliptic curves in cryptography," in *Advances in Cryptology*, vol. 3, J. Peters, Ed. New York, NY, USA: McGraw-Hill, 1964, pp. 15–64.

[2] N. Koblitz, "Elliptic curve cryptosystems," *Math. Comput.*, vol. 48, no. 177, pp. 203–209, 1987.

[3] R. L. Rivest, "A method for obtaining digital signatures and public-key cryptosystems," *Commun. ACM*, vol. 21, no. 2, pp. 120–126, 1978.

[4] T. E. Gamal, "A public key cryptosystem and a signature scheme based on discrete logarithms," in *Proc. Adv. Cryptol.*, Santa Barbara, CA, USA, Aug. 1984, pp. 469–472.

[5] L. Han, Q. Xie, and W. Liu, "An improved biometric based authentication scheme with user anonymity using elliptic curve cryptosystem," *Int. J. Netw. Secur.*, vol. 19, no. 3, pp. 469–478, 2017.

[6] D. Singh, B. Kumar, S. Singh, and S. Chand, "A secure IoT-based mutual authentication for healthcare applications in wireless sensor networks using ECC," *Int. J. Healthcare Inf. Syst. Informat.*, vol. 16, no. 2, pp. 21–48, Apr. 2021. [Online]. Available: https://ideas.repec.org/a/igg/jhisi0/v16y2021i2p21-48.html

[7] D. Rangwani and H. Om, "A secure user authentication protocol based on ECC for cloud computing environment," *Arabian J. Sci. Eng.*, vol. 46, no. 4, pp. 3865–3888, 2021.

[8] G. Jianguang, "Scalar multiplication algorithm resisting power analysis attacks using NAF with threshold window," *Comput. Eng.*, vol. 45, pp. 296–299, 2019.

[9] D. Khleborodov, "Fast elliptic curve point multiplication based on window non-adjacent form method," *Appl. Math. Comput.*, vol. 334, pp. 41–59, Oct. 2018.

[10] D. T. Cooklev, "Two algorithms for modular exponentiation using non-standard arithmetics," *IEICE Trans. Fundam. Electron. Commun. Comput. Sci.*, vol. 78, no. 1, pp. 82–87, 1995.

[11] V. S. Dimitrov, G. A. Jullien, and W. C. Miller, "Theory and applications for a double-base number system," in *Proc. 13th IEEE Symp. Comput. Arithmetic*, Dec. 1997, pp. 44–51.
[12] V. S. Dimitrov, G. A. Jullien, and W. C. Miller, "An algorithm for modular exponentiation," *Inf. Process. Lett.*, vol. 66, no. 3, pp. 155–159, May 1998.
[13] V. Dimitrov, L. Imbert, and P. Mishra, "Efficient and secure elliptic curve point multiplication using double-base chains," in *Proc. Int. Conf. Theory Appl. Cryptol. Inf. Secur.* Berlin, Germany: Springer, 2005, pp. 59–78.
[14] P. Mishra and V. Dimitrov, "Efficient quintuple formulas for elliptic curves and efficient scalar multiplication using multibase number representation," in *Proc. 10th Int. Conf. Inf. Secur.*, Valparaíso, Chile, Oct. 2007, pp. 390–406.
[15] W. Yu, "Triple-base number system for scalar multiplication," in *Proc. Int. Conf. Cryptol. Afr.* Berlin, Germany: Springer, 2013, pp. 433–451.
[16] W. Yu, "Research on several algorithms of elliptic curve cryptography," Ph.D. dissertation, School Cyberspace Sci. Technol., Univ. Sci. Technol. China, Hefei, China, 2013.
[17] N. Méloni and M. A. Hasan, "Efficient double bases for scalar multiplication," *IEEE Trans. Comput.*, vol. 64, no. 8, pp. 2204–2212, Aug. 2015.
[18] D. Yunqi, W. Jiang, M. Chuangui, and W. Fushan, "Secure and efficient ECC speeding up algorithms for wireless sensor networks," *Soft Comput.*, vol. 21, no. 19, pp. 5665–5673, Oct. 2017, doi: 10.1007/s00500-016-2142-x.
[19] D. Yunqi, W. Jiang, M. Chuangui, and W. Fushan, "Revisiting the expansion length of triple-base number system for elliptic curve scalar multiplication," *J. Inf. Sci. Eng.*, vol. 34, no. 3, pp. 721–732, 2018.
[20] M. Fossorier, T. Holdt, and A. Poli, "Fast point multiplication on elliptic curves through isogenies," in *Proc. 15th Int. Conf. Appl. Algebra*, 2004, pp. 43–50.
[21] *IEEE Standard Specifications for Public-Key Cryptography*, Standard 1363-2000, 2002.
[22] V. S. Dimitrov, L. Imbert, and P. K. Mishra, "The double-base number system and its application to elliptic curve cryptography," *Math. Comput.*, vol. 77, no. 262, pp. 1075–1104, 2008.
[23] S. M. Cho, S. G. Gwak, C. H. Kim, and S. Hong, "Faster elliptic curve arithmetic for triple-base chain by reordering sequences of field operations," *Multimedia Tools Appl.*, vol. 75, no. 22, pp. 1–13, 2016.
[24] G. N. Purohit and A. S. Rawat, "Fast scalar multiplication in ECC using the multi-base number system," *Int. J. Comput. Sci. Issues*, vol. 8, no. 1, p. 44, 2011.
[25] N. Méloni, "New point addition formulae for ECC applications," in *Proc. 1st Int. workshop Arithmetic Finite Fields*, 2007, pp. 189–201.
[26] Z. Lai, A. Zhang, and Z. Zhang, "Improved algorithms for computing 7P and 7 kP on elliptic curves," *Comput. Eng. Appl.*, vol. 52, no. 1, pp. 29–32 and 156, 2016.
[27] P. Longa and C. H. Gebotys, "Fast multibase methods and other several optimizations for elliptic curve scalar multiplication," in *Proc. Int. Conf. Pract. Theory Public Cryptogr.* Heidelberg, Germany: Springer-Verlag, 2009, pp. 443–462.
[28] S. Liu, Y. Zhang, and S.-Y. C. Y. Ding, "Co_Z point addition algorithm on elliptic curve over characteristic two," *J. Wuhan Univ.*, vol. 65, pp. 207–212, Jul. 2021.

SHUANGGEN LIU received the Ph.D. degree in cryptography from Xidian University, in 2008. He is currently an Associate Professor with the School of Cyber Security, Xi'an University of Posts and Telecommunications, Xi'an, China. His recent research interests include crptography and information security. He is a member of China Computer Federation and the Chinese Association for Cryptologic Research.

LIJUAN ZHANG is currently pursuing the degree with Xi'an University of Posts and Telecommunications. She is mainly engaged in the research of elliptic curve cryptosystems.