International Journal of Networking and Computing – www.ijnc.org, ISSN 2185-2847
Volume 8, Number 2, pages 341–350, July 2018

# An Implementation of ECC with Twisted Montgomery Curve over 32nd Degree Tower Field on Arduino Uno

Yuta Hashimoto, Md. Al-Amin Khandaker, Yuta Kodera, Takuya Kusaka and Yasuyuki Nogami
Graduate School of Natural Science and Technology, Okayama University,
3-1-1 Tsushima-naka, Kita-ku Okayama-city, Okayama, 700-8530, Japan

Taehwan Park and Howon Kim
School of Computer Science and Engineering, Pusan National University,
San-30, JangJeon-Dong, Geumjeong-Gu, Busan, 609-735, Republic of Korea

Received: February 1, 2018
Revised: May 17, 2018
Accepted: June 2, 2018
Communicated by Toru Nakanishi

**Abstract**

The security of Internet of Things (IoT) devices is one of the most important problems to be addressed by the cryptographers and security engineers. The processing ability of IoT devices is limited, therefore light-weight and secure cryptographic tools are necessary for security of them. This paper shows the implementation of 256-bit Elliptic Curve Cryptography (ECC) on an 8-bit microcontroller. The proposed implementation applies towering technique for extension field of degree 32 with a certain 8-bit prime characteristic instead of the 256-bit prime characteristic. It enables to execute 256-bit ECC operations without complicated multiple-precision arithmetic on small computers like 8-bit microcontrollers. This approach efficiently realizes the scalability of the ECC encryption strength. In addition, the authors use a twisted Montgomery curve with a Montgomery ladder technique which enables fast calculations without inversions referring to Curve25519. It is considered resistant to the Side Channel Attack (SCA) since it applies the Montgomery ladder technique for scalar multiplication (SCM). This ECC implementation on Arduino UNO, an 8-bit microcontroller board, can be utilized for a key agreement protocol among IoT devices.

*Keywords:* ECC, Twisted Montgomery Curve, Montgomery ladder, Tower of fields, IoT security, Microcontroller

## 1 Introduction

In the IoT era, many devices are connected to the Internet. But, every device does not have enough processing ability to use powerful encryption methods which are used on high-performance devices such as PCs. Therefore, an encryption method which enables enough security with fewer computations is necessary.

RSA is the most popular public-key cryptography which is based on the difficulty of the factoring problem, though the key size needs not less than 2048-bit to ensure the security. On the other hand, Elliptic Curve Cryptography (ECC) is an another choice for public-key cryptography. ECC is based on the intractability of Elliptic Curve Discrete Logarithm Problem (ECDLP). It is attracting the attention as a next-generation public-key cryptography because of the efficiency at the security level per bit of the key size. For example, a 256-bit ECC public-key should provide comparable security to a 3072-bit RSA public-key.

Among the elliptic curves, Curve25519 [1] which is proposed for Elliptic Curve Diffie Hellman (ECDH) in TLS 1.3 [2], is one of the fastest curves. However, computing encryption on Curve25519 which handles 256-bit integers is computationally expensive for smaller devices such as an 8-bit microcontroller in IoT devices.

Then, two approaches are available for the implementation of 256-bit ECC on 8-bit microcontrollers.
1. Using the multiple-precision arithmetic over a 256-bit prime field.
2. Using the arithmetic over an $m$-th degree extension field of an $n$-bit characteristic such that $m * n = 256$.

Approach 1 may be general for implementations of the large bit ECC regardless of its computational resources. If the device is small such as IoT devices, it is usually implemented with the hard optimization for the fixed parameters of cryptography in order to enable the encryption/decryption to work enough fast. However, the security parameters for cryptography have been rapidly changing with the growth of processing ability of computers. In the case of implementation with the multiple-precision arithmetic for fixed parameters of the curve, it becomes too hard to change the parameters by redesigning most arithmetic codes due to the hard optimization. Therefore, the authors consider that this approach is not the best at the scalability of the security strength of the implementation.

This paper introduces an ECC implementation on Arduino Uno [3], an 8-bit microcontroller board, by Approach 2 with $m = 32, n = 8$. The proposed implementation is suitable for 8-bit microcontrollers because of the same security level with the scaled-down integers and the towering technique of extension fields. It enables to execute 256-bit ECC encryption just by handling 8-bit integers over $\mathbb{F}_{q^{32}}$ tower field where $q$ is an 8-bit prime number. Then, the complicated multiple-precision arithmetic is not required for it. It has also the scalability of the ECC encryption strength, because the degree of the tower field can be treated as a variable parameter. The parameter can be increased simply and easily to strengthen the encryption. Furthermore, it applies the combination of a twisted Montgomery curve [4] and a Montgomery ladder [5] referring to Curve25519. It provides an efficient SCM algorithm and high resistance to SCA [6].

# 2 Fundamentals

This section briefly introduces finite field, elliptic curve, Montgomery ladder, Montgomery curve, tower field and Weil's theorem.

## 2.1 Finite Field

### 2.1.1 Group

Group is an algebraic system defined as follows.

**Definition 2.1** (Group)**.** *A group $\langle \mathbb{G}, \circ \rangle$ is a nonempty set with a binary operation $\circ$ that satisfies the following group axioms:*

**G1 : (Closure)** *For $^\forall a, ^\forall b \in \mathbb{G}$, the result of $a \circ b$ is also in $\mathbb{G}$.*

**G2 : (Associativity)** $(a \circ b) \circ c = a \circ (b \circ c), \quad a, b, c \in \mathbb{G}$.

**G3 : (Unity)** *For $^\forall a \in \mathbb{G}$, there exists an element $e \in \mathbb{G}$ such that $a \circ e = e \circ a = a$, where $e$ is called unity (unit element).*

**G4 : (Inverse Element)** *For $^\forall a \in \mathbb{G}$, there exists an element $x \in \mathbb{G}$ such that $a \circ x = x \circ a = e$, where $x$ is called inverse element of $a$.*

**Definition 2.2** (Commutative Group)**.**

**AG5 : (Commutativity)** *A group $\mathbb{G}$ is said to be commutative (or abelian), if $a \circ b = b \circ a$ for $^\forall a, b \in \mathbb{G}$.*

### 2.1.2 Field

Field is an algebraic system defined as follows.

**Definition 2.3** (Field)**.** *A field $\langle \mathbb{F}, +, \cdot \rangle$ has two binary operations denoted by $+$ and $\cdot$, such that:*

**F1 : (Additive Group)** *$\mathbb{F}$ is a commutative group with respect to $+$.*

**F2 : (Multiplicative Group)** *$\mathbb{F}^*$is a group with respect to $\cdot$, where $\mathbb{F}^*$is the set that consists of every element distinct from the unity (zero element) with respect to $+$.*

**F3 : (Distributive law)** *For all $a, b, c \in \mathbb{F}$, $a \cdot (b + c) = a \cdot b + a \cdot c$ and $(b + c) \cdot a = b \cdot a + c \cdot a$ are satisfied.*

**Definition 2.4** (Order of Field)**.** *The order is the number of elements in $\mathbb{F}$ is called the order of field $\mathbb{F}$. If the order of $\mathbb{F}$ is finite, $\mathbb{F}$ is called finite field.*

**Definition 2.5** (Characteristic of Field)**.** *The least positive number $n$ such that $n \cdot a = 0$ for every $a \in \mathbb{F}$ is called characteristic.*

### 2.1.3 Prime Field

A prime field $\mathbb{F}_q$ is defined as a finite field with the prime characteristic $q$. The elements in $\mathbb{F}_q$ is represented by integers between 0 and $q - 1$. For two elements $a, b \in \mathbb{F}_q$, their addition and multiplication are defined as follows.

$$a + b \quad \coloneqq \quad a + b \qquad (\mathrm{mod}\ q) \tag{1a}$$

$$a \cdot b \quad \coloneqq \quad a \cdot b \qquad (\mathrm{mod}\ q) \tag{1b}$$

Then, a division can be considered as a multiplication by the inverse element of the divisor. The calculation of the multiplicative inverse is called inversion.

### 2.1.4 Extension Field

An extension field $\mathbb{F}_{q^m}$ produces a vector space over the prime field $\mathbb{F}_q$ where $q$ is the characteristic of $\mathbb{F}_q$ and $m$ is the extension degree. The extension degree $m$ means the dimension of the vector space. In order to represent an arbitrary vector, a certain basis such as $\{\alpha_0, \alpha_1, \cdots, \alpha_{m-1}\}$ is necessary. Then, an arbitrary vector $A \in \mathbb{F}_{q^m}$ is represented as follows.

$$A = \sum_{i=0}^{m-1} a_i \alpha_i, \qquad a_i \in \mathbb{F}_q \tag{2}$$

## 2.2 Elliptic Curve

An elliptic curve $E$ over the prime field $\mathbb{F}_q$ for Elliptic Curve Cryptography (ECC) is defined as follows.

$$E : \ y^2 = x^3 + ax + b, \qquad a, b \in \mathbb{F}_q, \ \ x, y \in \mathbb{F}_{q^m}. \tag{3}$$

For $\mathbb{F}_q$-rational points $P(x_1, y_1), Q(x_2, y_2)$ which satisfy this equation, the addition $P + Q = (x_3, y_3)$ is defined as follows.

$$\lambda = \begin{cases} \dfrac{y_1 - y_2}{x_1 - x_2} & \text{when} \quad x_1 \neq x_2, \\ \dfrac{3x_1^2 + a}{2y_1} & \text{when} \quad P = Q \text{ and } y_1 \neq 0, \\ \phi & \text{otherwise.} \end{cases} \tag{4a}$$

$$(x_3, y_3) = \begin{cases} (\lambda^2 - (x_1 + x_2), \lambda(x_3 - x_1) + y_1) & \text{when } \lambda \neq \phi, \\ O & \text{when } \lambda = \phi. \end{cases} \tag{4b}$$

When $P(x_1, y_1) = Q(x_2, y_2)$, it becomes the doubling of $\mathbb{F}_q$-rational point $P(x_1, y_1)$.

In the case of $\lambda = \phi$, the result of this addition becomes the point at infinity $O$ which is the unity as $P + O = O + P = P$.

The Scalar Multiplication (SCM) for a rational point $P$ and a scalar $s$ is denoted by $[s]P = \sum_{i=0}^{s-1} P$.

## 2.3 Montgomery Ladder

Binary method is well known as an efficient SCM technique though it has a risk of side channel attack.

Then, the Montgomery ladder is an another efficient SCM technique that stands up to side channel attacks. SCM with Montgomery ladder is calculated as following Alg.1.

**Algorithm 1**: SCM with Montgomery ladder

**Input:** $P, s = \{s_{n-1}, s_{n-2}, ..., s_1, s_0\}$
**Output:** $T_1 (= [s]P)$

```
T1 ← O
T2 ← P
for i = n − 1 to 0 do
    if s_i = 1 then
        T1 ← T1 + T2
        T2 ← 2T2
    else
        T2 ← T1 + T2
        T1 ← 2T1
    end if
end for
return T1
```

## 2.4 Montgomery Curve

The Montgomery curve is defined as follows.

$$E_{AB}: \ By^2 = x^3 + Ax^2 + x, \qquad A, B \in \mathbb{F}_q, \ \ x, y \in \mathbb{F}_{q^m} \tag{5}$$

where $B(A^2 - 4) \neq 0 \pmod q$.

A rational point $P = (x, y)$ on the Montgomery curve is represented in Montgomery coordinates $P = (X : Z)$ where $P = (X : Z)$ are projective coordinates and $x = X/Z$ for $Z \neq 0$.

For the rational points $P_i(X_i : Z_i) = [i]P(X : Z)$, $P_j(X_j : Z_j) = [j]P(X : Z)$, the addition $P_i + P_j = P_{i+j}(X_{i+j} : Z_{i+j})$ is calculated as follows.

$$X_{i+j} = Z_{i-j}((X_i - Z_i)(X_j + Z_j) + (X_i + Z_i)(X_j - Z_j))^2, \tag{6a}$$

$$Z_{i+j} = X_{i-j}((X_i - Z_i)(X_j + Z_j) - (X_i + Z_i)(X_j - Z_j))^2. \tag{6b}$$

When $i = j$, the doubling $2P_j = P_{2j}(X_{2j} : Z_{2j})$ is calculated as follows.

$$X_{2j} = (X_j + Z_j)^2 (X_j - Z_j)^2, \tag{7a}$$

$$Z_{2j} = T((X_j - Z_j)^2 + \frac{A+2}{4} \cdot T), \tag{7b}$$

$$T = (X_j + Z_j)^2 - (X_j - Z_j)^2. \tag{7c}$$

It is calculated with the Montgomery ladder efficiently.

#### 2.4.1 Curve25519

Curve25519 is known as one of the most efficient elliptic curves. It is designed for ECDH and offers 128-bit security. It is defined by

$$E_{25519}: \ y^2 = x^3 + 486662x^2 + x, \tag{8}$$

which is a Montgomery curve with the fixed prime number $q = 2^{255} - 19$.

An SCM of the rational points on this curve does not need any inversion in $\mathbb{F}_q$ without a final division because of using a Montgomery curve, Montgomery ladder, and projective $(X : Z)$ coordinates [1].

This curve is equivalent to a certain twisted Edwards curve [1].

### 2.5 Weil's Theorem

Let $E(\mathbb{F}_q)$ be the set of all rational points on $E$ over $\mathbb{F}_q$, then it forms an additive group on the elliptic curve addition Eq.(4). $\#E(\mathbb{F}_q)$ denotes the order of $E(\mathbb{F}_q)$.

Let $t_1 = q + 1 - \#E(\mathbb{F}_q)$ be the Frobenius trace of $E(\mathbb{F}_q)$. When $E$ is defined over the extension field $\mathbb{F}_{q^m}$, the order is given by

$$\begin{aligned} \#E(\mathbb{F}_{q^m}) &= q^m + 1 - t_m, \\ t_m &= \alpha^m + \beta^m, \end{aligned} \tag{9}$$

where $\alpha, \beta$ are complex numbers which satisfy $\alpha\beta = q$ and $\alpha + \beta = t_1$, and $t_m$ is the Frobenius trace of $E(\mathbb{F}_{q^m})$. It is well known as Weil's theorem [7].

### 2.6 Twists of Curves

A twist of elliptic curve $E$ is given by

$$E': \ y = x^3 + aA^2x + bA^3, \tag{10}$$

where $A$ is a non-zero element in the definition field $\mathbb{F}_{q^m}$.

Corresponding to whether $A$ is a quadratic residue (QR) or a quadratic non-residue (QNR) in $\mathbb{F}_{q^m}$, the order $\#E'(\mathbb{F}_{q^m})$ of the twisted elliptic curve $E'$ is given as follows.

$$\#E'(\mathbb{F}_{q^m}) = \begin{cases} q^m + 1 - t_m & \text{when } A \text{ is a QR}, \\ q^m + 1 + t_m & \text{when } A \text{ is a QNR}. \end{cases} \tag{11}$$

### 2.7 Tower Field

Let $q$ be a prime which satisfies $4 \mid (q-1)$ and $\alpha_{n-1}$ be a QNR element in $\mathbb{F}_{q^{2^{n-1}}}$. Then, the basis $\{1, \alpha_n\}$ of $\mathbb{F}_{q^{2^n}}$ where $\alpha_n$ is a square root of $\alpha_{n-1}$ over $\mathbb{F}_{q^{2^{n-1}}}$ is given as a chain.

$$\begin{aligned} \mathbb{F}_{q^2} &= \mathbb{F}_q[\alpha_1]/(\alpha_1^2 - \alpha_0), \\ \mathbb{F}_{q^4} &= \mathbb{F}_{q^2}[\alpha_2]/(\alpha_2^2 - \alpha_1), \\ &\vdots \\ \mathbb{F}_{q^{2^n}} &= \mathbb{F}_{q^{2^{n-1}}}[\alpha_n]/(\alpha_n^2 - \alpha_{n-1}). \end{aligned} \tag{12}$$

This technique realizes efficient arithmetic operations in the tower field $\mathbb{F}_{q^{2^n}}$.

## 3 Twisted Montgomery Curve over Tower Fields

This section describes elliptic curve over tower fields and the definition of a twisted Montgomery curve over tower field.

## 3.1 Elliptic Curves over Tower Fields

$E(\mathbb{F}_{q^m})$ denotes the set of rational points on the elliptic curve $E$ over the $m$-th degree extension field $\mathbb{F}_{q^m}$. The sub-fields of $E(\mathbb{F}_{q^{32}})$ satisfy the following inclusion relation.

$$E(\mathbb{F}_q) \subseteq E(\mathbb{F}_{q^2}) \subseteq E(\mathbb{F}_{q^4}) \subseteq E(\mathbb{F}_{q^8}) \subseteq E(\mathbb{F}_{q^{16}}) \subseteq E(\mathbb{F}_{q^{32}}) \tag{13}$$

It has a negative point from security viewpoints such that the order $\#E(\mathbb{F}_{q^{32}})$ has many factors. The order of curves used for ECC must be a large prime or divisible by a large prime in order to ensure its security. At this point, the twisted curve $E'$ solves the security problem since the order of a twisted curve $\#E'(\mathbb{F}_{q^{2n}})$ can have a lager factor as described by the paper [8].

## 3.2 Definition of a Twisted Montgomery Curve over a Tower Field

When let $B$ be 1, the Montgomery curve over $\mathbb{F}_{q^{32}}$ becomes as follows.

$$E_A : \ y^2 = x^3 + Ax^2 + x, \quad A \in \mathbb{F}_q, \ x, y \in \mathbb{F}_{q^{32}}. \tag{14}$$

The order of $E_A$ is given by the Weil's theorem as follows.

$$\#E_A(\mathbb{F}_{q^{32}}) = q^{32} + 1 - t_{32}. \tag{15}$$

Then, the twisted Montgomery curve of $E_A$ is defined by

$$E'_A : \ y^2 = x^3 + A\theta x^2 + \theta^2 x, \quad A \in \mathbb{F}_q, \ x, y, \theta \in \mathbb{F}_{q^{32}}, \tag{16}$$

where $\theta$ is a QNR in $\mathbb{F}_{q^{32}}$ and the order of $E'_A$ is also given by the Weil's theorem as follows.

$$\#E'_A(\mathbb{F}_{q^{32}}) = q^{32} + 1 + t_{32}. \tag{17}$$

This order $\#E'_A(\mathbb{F}_{q^{32}})$ can be a large prime number or a composite number with a large prime factor. In addition, the twisted Montgomery curve $E'_A$ has the following efficiency.

For $\mathbb{F}_{q^{32}}$-rational points $R(X_1 : Z_1), S(X_2 : Z_2), P(X_P : Z_P) \in E'_A(\mathbb{F}_{q^{32}})$ which are represented in projective coordinates on this curve and satisfy $S - R = P$, addition $R + S = (X_3 : Z_3)$ is calculated as follows.

$$X_3 = \theta^2 Z_P((X_1 - \theta Z_1)(X_2 + \theta Z_2) + (X_1 + \theta Z_1)(X_2 - \theta Z_2))^2, \tag{18a}$$

$$Z_3 = X_P((X_1 - \theta Z_1)(X_2 + \theta Z_2) - (X_1 + \theta Z_1)(X_2 - \theta Z_2))^2. \tag{18b}$$

Let $P(X_P : Z_P) = (\delta : 1)$ be the base point, they become more simple formulas as follows.

$$X_3 = ((X_1 - \theta Z_1)(X_2 + \theta Z_2) + (X_1 + \theta Z_1)(X_2 - \theta Z_2))^2, \tag{19a}$$

$$Z_3 = ((X_1 - \theta Z_1)(X_2 + \theta Z_2) - (X_1 + \theta Z_1)(X_2 - \theta Z_2))^2. \tag{19b}$$

In addition, for $\mathbb{F}_q$-rational points $R(X_1 : Z_1)$, doubling $2R = (X_2 : Z_2)$ is calculated as follows.

$$X_2 = \theta(X_1 + \theta Z_1)^2 (X_1 - \theta Z_1)^2, \tag{20a}$$

$$Z_2 = T((X_1 + \theta Z_1)^2 + \frac{A-2}{4} \cdot T), \tag{20b}$$

$$T = (X_1 + \theta Z_1)^2 - (X_1 - \theta Z_1)^2. \tag{20c}$$

Then, SCM is calculated with the Montgomery ladder as following Alg.2 where $A_{24}$ is $(A-2)/4$.

**Algorithm 2**: SCM with the Montgomery ladder

**Input:** $s = \{s_{n-1}, s_{n-2}, ..., s_1, s_0\}$
**Output:** $Q(= [s]P) = (X_1 : Z_1)$

$Q \leftarrow O$
$X_2 \leftarrow \theta$
$Z_2 \leftarrow 1$
**for** $i \leftarrow n - 1$ **to** $0$ **do**
  $t_0 \leftarrow Z_1 \times \theta$
  $t_1 \leftarrow X_1 + t_0$
  $t_2 \leftarrow X_1 - t_0$
  $t_0 \leftarrow Z_2 \times \theta$
  $t_3 \leftarrow X_2 + t_0$
  $t_4 \leftarrow X_2 - t_0$
  **if** $s_i = 1$ **then**
    $t_5 \leftarrow t_1 \times t_4$
    $t_6 \leftarrow t_2 \times t_3$
    $t_0 \leftarrow t_5 + t_6$
    $X_1 \leftarrow t_0 \times t_0$
    $t_0 \leftarrow t_5 - t_6$
    $Z_1 \leftarrow t_0 \times t_0$
    $t_1 \leftarrow t_3 \times t_3$
    $t_2 \leftarrow t_4 \times t_4$
    $t_0 \leftarrow t_1 \times t_2$
    $X_2 \leftarrow t_0 \times \theta$
    $t_0 \leftarrow t_1 - t_2$
    $t_3 \leftarrow t_0 \times A_{24}$
    $t_4 \leftarrow t_1 + t_3$
    $Z_2 \leftarrow t_0 \times t_4$
  **else**
    $t_5 \leftarrow t_1 \times t_4$
    $t_6 \leftarrow t_2 \times t_3$
    $t_0 \leftarrow t_5 + t_6$
    $X_2 \leftarrow t_0 \times t_0$
    $t_0 \leftarrow t_5 - t_6$
    $Z_2 \leftarrow t_0 \times t_0$
    $t_3 \leftarrow t_1 \times t_1$
    $t_4 \leftarrow t_2 \times t_2$
    $t_0 \leftarrow t_3 \times t_4$
    $X_1 \leftarrow t_0 \times \theta$
    $t_0 \leftarrow t_3 - t_4$
    $t_1 \leftarrow t_0 \times A_{24}$
    $t_2 \leftarrow t_1 + t_3$
    $Z_1 \leftarrow t_0 \times t_2$
  **end if**
**end for**
**return** $Q$

# 4 Implementation on Arduino

This section describes the detail of the proposed implementation on Arduino.

## 4.1 Arduino Uno

Arduino is an open-source electronics platform and it is commonly used in IoT products. SCM for ECC is executed on Arduino Uno, a microcontroller board based on the ATmega328P which is an Atmel 8-bit AVR microcontroller. Program codes for Arduino are written in Arduino language based on C/C++ with Arduino IDE or AVR-C language with AVR toolchains.

Figure 1: Arduino Uno

The specification for Arduino Uno is shown in Table 1.

Table 1: Specification

| Device | Arduino Uno |
|---|---|
| Microcontroller | ATmega328P |
| Clock Speed | 16 MHz |
| Flash memory (Program space) | 32 KB |
| SRAM (Variables space) | 2 KB |

The program code of the proposed implementation is written in AVR-C language because it can be compiled with harder optimization options than that of Arduino language.

## 4.2 Modulo $q$ arithmetic over the prime field $\mathbb{F}_q$

In the proposed implementation, the operator % in AVR-C language is never used because the operation constructed is too time-consuming on Arduino Uno. Instead of it, the modulo $q$ arithmetic over the prime field $F_q$ is implemented with inline assembly codes to reduce the computational costs. It applies a modulo $q$ technique by the relation $2^t \equiv c \pmod{q}$ where $q = 2^t - c$. This technique realizes the modulo $q$ arithmetic with just addition, multiplication and bit shift operations. In the case of $q = 239$, an 8-bit prime number, it is represented as $q = 2^8 - 17$. Then, more than 8-bit values are reduced by the relation $2^8 \equiv 17 \pmod{239}$.

## 4.3 Parameter Setting

Let $q = 239$, $m = 32$ and $A = 26$ for the twisted Montgomery curve over the tower field, then the group order of the proposed curve becomes as shown in Table 2. In comparison with the parameters of Curve25519 shown in Table 3, the bit-size of the group order is smaller than that of Curve25519. There are also $q = 251$ and $q = 241$ which are 8-bit primes larger than $q = 239$, but the group order of the curve has some large value factors in these case.

$q = 239$ satisfies $4 \nmid (q-1)$ and $3 \nmid (q-1)$. Let $\alpha$ be the root of the sixth cyclotomic polynomial $\Phi_6 = x^2 - x + 1$ which is an irreducible polynomial because of $3 \nmid (q-1)$. Then, $\alpha + 2$ becomes a QNR over $\mathbb{F}_{q^2}$. The definition field $\mathbb{F}_{q^{32}}$ is constructed as follows.

$$
\begin{aligned}
\mathbb{F}_{q^2} &= \mathbb{F}_q[\alpha]/(\alpha^2 - \alpha + 1),\\
\mathbb{F}_{q^4} &= \mathbb{F}_{q^2}[\beta]/(\beta^2 - (\alpha + 2)),\\
\mathbb{F}_{q^8} &= \mathbb{F}_{q^4}[\gamma]/(\gamma^2 - \beta),\\
\mathbb{F}_{q^{16}} &= \mathbb{F}_{q^8}[\delta]/(\delta^2 - \gamma),\\
\mathbb{F}_{q^{32}} &= \mathbb{F}_{q^{16}}[\theta]/(\theta^2 - \delta).
\end{aligned}
\tag{21}
$$

Let $P = (\delta : 1)$ be the base point, because the order of $P$ becomes $\#E'_A/4$, a large prime shown in Table 2 in the case of $q = 239$, $A = 26$ and it reduces the cost of the addition because of Eq.(19). For the comparison, parameters of Curve25519 is shown in Table 3.

Table 2: Parameters of the proposed curve

| The proposed curve | |
|---|---|
| $q$ (Prime number) | $2^8 - 17$ |
| $A$ | 26 |
| Group order $\#E'_A$ | $2^{252}$ + 560795661594583649320467314004776048711275624289804345271834587710640851866 0 |
| Order of the base point | $2^{250}$ + 140198915398645912330116828501194012177818906072451086317958646927660212966 5 |

Table 3: Paramters of Curve25519

| Curve25519 | |
|---|---|
| $q$ (Prime number) | $2^{255} - 19$ |
| $A$ | 486662 |
| Group order $\#E_{25519}$ | $2^{255}$ + 221938542218978828286815502327069187944 |
| Order of the base point | $2^{252}$ + 27742317777372353535851937790883648493 |

## 4.4 Experimental Results

It computes operations of the proposed curve with arithmetic operations over $\mathbb{F}_{q^{32}}$. Execution times of those is shown in following Table 4.

Table 4: Results of arithmetic operations over $\mathbb{F}_{q^{32}}$

| Arithmetic operations over $\mathbb{F}_{q^{32}}$ | Execution time $[ms]$ |
|---|---|
| $\mathbb{F}_{q^{32}}$ addition | 0.035 |
| $\mathbb{F}_{q^{32}}$ multiplication | 1.12 |
| $\mathbb{F}_{q^{32}}$ squaring | 0.75 |

The comparison results of SCM operation between the proposed implementation and the implementation of Curve25519 provided by μNaCl [9] [10] is shown in following Table 5.

Table 5: Comparison results of the SCM operation

| SCM operation | Execution time $[s]$ | Clock cycles |
|---|---|---|
| The proposed curve | 1.96 | 31340271 |
| Curve25519 | 0.88 | 14087567 |

### 4.5 Consideration

According to the above results, the proposed implementation takes about 2 seconds to execute the SCM operation that is about 2 times longer than that of Curve25519 provided by μNaCl. Therefore, the proposed implementation should be improved by a more careful implementation.

## 5 Conclusion

This paper has introduced the implementation which realizes 256-bit ECC with 8-bit integer arithmetic instead of multiple-precision arithmetic on Arduino. It applies an efficient SCM algorithm with a Montgomery curve and Montgomery ladder referring to Curve25519. This paper also has shown the idea of twisted Montgomery curve which provides the security of ECC with a Montgomery curve over the tower field. The proposed implementation can be utilized for a key agreement protocol among IoT devices. As the future work, the execution time of SCM should be scaled down for the practical use. Moreover, the evaluation of the resistance to SCA is also a future work.

## Acknowledgement

This work is partially supported by the Strategic Information and Communications R&D Promotion Programme (SCOPE) of Ministry of Internal Affairs and Communications, Japan, and the MSIT(Ministry of Science and ICT), Korea, under the ITRC(Information Technology Research Center) support program(IITP-2017-2014-0-00743) supervised by the IITP(Institute for Information & communications Technology Promotion)

## References

[1] D. J. Bernstein. Curve25519: new Diffie-Hellman speed records. In *International Workshop on Public Key Cryptography*, pages 207–228, Springer, Berlin, Heidelberg, 2006.

[2] https://tools.ietf.org/html/draft-ietf-tls-tls13-20. [Accessed 20 July 2017].

[3] http://www.arduino.org/products/boards/arduino-uno. [Accessed 20 July 2017].

[4] P. L. Montgomery. Speeding the Pollard and elliptic curve methods of factorization. In *Mathematics of computation*, number 177, pages 243–264, 1987.

[5] M. Joye and S. M. Yen. The Montgomery powering ladder. In *CHES*, volume 2, pages 291–302, 2002.

[6] P. C. Kocher. Timing attacks on implementations of Diffie-Hellman, RSA, DSS, and other systems. In *Annual International Cryptology Conference*, pages 104–113, Springer, Berlin, Heidelberg, 1996.

[7] I.Blake, G.Seroussi, and N.Smart. *Elliptic curves in cryptography*, volume 265. Cambridge university press, 1995.

[8] Y. Nogami and Y. Morikawa. Fast Generation of Elliptic Curves with Prime Order over $\mathbb{F}_{p^{2^e}}$. In *Proc. of Workshop on Coding and Cryptography 2003*, pages 347–3, 2003.

[9] M. Düll, B. Haase, G. Hinterwälder, M. Hutter, C. Paar, A. H. Sánchez, and P. Schwabe. High-speed Curve25519 on 8-bit, 16-bit, and 32-bit microcontrollers. In *Designs, Codes and Cryptography 2015*, pages 291–302, 2015.

[10] https://munacl.cryptojedi.org/atmega.shtml. [Accessed 22 September 2017].