Efficient Pairing-Based Cryptography on Raspberry Pi

Yuki Nanjo,[1] Md. Al-Amin Khandaker,[2] Takuya Kusaka[2] and Yasuyuki Nogami[2]
[1] Department of Electrical and Communication Engineering Okayama University, Japan.
[2] Graduate School of Natural Science and Technology Okayama University, Japan.
Email: yuki.nanjo@s.okayama-u.ac.jp; khandaker@s.okayamau.ac.jp; {kusaka-t, yasuyuki.nogami}@okayamau.ac.jp

Abstract —In the age of IoT, pairing-based cryptography (PBC) can play an important role as a public key cryptography since it enables several innovative protocols such as anonymous encryption and certificate-less authentication. However, due to the computation complexity, PBC is often regarded computationally unfeasible for IoT devices. Therefore, this paper tries to push that limit by efficiently calculating the pairing operation together with scalar multiplication and exponentiation over the Barreto-Naehrig (BN) curve by applying the state of art techniques. In addition to the theoretical explanation of the applied techniques, the authors also show a high-level implementation using C programming on a raspberry pi model 3 B, instead of hardware specific implementation.

Index Terms—Pairing-based cryptography, BN curve, efficient implementation.

I. Introduction

In 1976, the historic work of Whitfield Diffie and Martin Hellman [1] initiated a new era in information security known as public key cryptography. Two years later Rivest-Shamir-Adleman proposed RSA cryptography [2], which is still the most widely used public key cryptosystem. In the mid 80's the independent work of Miller [3] and Koblitz [4] began the journey of the elliptic curve cryptosystem (ECC). However, due to the shorter key length for same security level than RSA, ECC became popular among researchers. At the beginning of this century when the debate between RSA and ECC was at its peak, a new paradigm of cryptography called pairing-based cryptography, which is based on an elliptic curve came into the limelight by the independent work of Sakai *et al*. [5] and Joux [6]. Since then, researchers have proposed many innovative cryptography applications based on pairing such as ID-based encryption [7] and attribute based encryption [8]. Several pairing techniques such as ate [9], optimal-ate [10] and χ-ate [11] have been developed over the years. In general, pairing is a bilinear map from two additive rational point groups, $\mathbb{G}_1$ and $\mathbb{G}_2$, to a multiplicative group, $\mathbb{G}_3$, denoted as $\mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_3$. The efficiency of the pairing computation depends mostly on:

- Miller's algorithm
- Final exponentiation
- Scalar multiplication in the $\mathbb{G}_1$ and $\mathbb{G}_2$ groups
- Exponentiation in $\mathbb{G}_3$.

This paper, focuses on efficiently optimizing the above operations and their implementation in C programming to verify the enhancement of the calculation efficiency on the Barreto-Naehrig (BN) curve [12].

Unlike RSA, the security of ECC and PBC depends on the discrete logarithm problem (DLP). However, Kim *et al*. [13] have recently proposed a new algorithm to solve the DLP at CRYPTO2016. Therefore, the previous parameters, such as the length of the prime number $\lfloor \log_2 p \rfloor$, should be updated. This paper applies the recent parameters of 128 bit security proposed by Barbulescu *et al*. [14], where $\lfloor \log_2 p \rfloor = 462$ bit.

Since pairing on the BN curve enables the calculation on the twisted isomorphic group, the authors adopted a skew Frobenius map [15] on the sextic twisted curve to efficiently carry out the elliptic curve scalar multiplication on $\mathbb{G}_1$ and $\mathbb{G}_2$. This paper also applies 7-sparse multiplication [16] for an efficient Miller's algorithm and the method of Fuentes *et al.*[17] for the final exponentiation. Finally, this paper also shows a high-level implementation using C programming language to evaluate the performance of the proposal on a raspberry pi 3 B. The optimized implementation for a raspberry pi 3 B using the ARM-NEON assembly is not the focus of this work. However, the proposed method can be implemented using the ARM-NEON assembly and to exploit the Quad Core CPU, a SIMD parallel computing technique can also be applied.

II. Fundamentals

A. *Elliptic Curve* [18]

Let E be the elliptic curve defined over the prime field $\mathbb{F}_p$ as follows:

$$E/\mathbb{F}_p : y^2 = x^3 + ax + b \tag{1}$$

where $4a^3 + 27b^2 \neq 0$ and $a, b \in \mathbb{F}_p$. Points satisfying Eq. (1) are known as the rational points on the curve. The set of rational points including the point at infinity $\mathcal{O}$ on the curve forms an additive Abelian group denoted by $E(\mathbb{F}_p)$, whose total number of points $\#E(\mathbb{F}_p)$ can be obtained as follows:

$$\#E(\mathbb{F}_p) = p + 1 - t \tag{2}$$

where t is the Frobenius trace of $E(\mathbb{F}_p)$.

- Point addition and doubling.

Manuscript received July 29, 2017; revised February 1, 2018.
Corresponding author email: yuki.nanjo@s.okayama-u.ac.jp.
doi:10.12720/jcm.13.2.88-93

Let $L = (x_l, y_l)$ and $M = (x_m, y_m)$ be two rational points on E. Their addition $N = L + M$, where $N = (x_n, y_n)$ and $L, M, N \in E(\mathbb{F}_p)$. The x and y coordinates of N are given as follows:

$$(x_n, y_n) = ((\lambda^2 - x_l - x_m), (x_l - x_n)\lambda - y_l) \quad (3)$$

where λ is given as follows:

$$\lambda = \begin{cases} (y_m - y_l)(x_m - x_l)^{-1} \ (L \neq M) \\ (3x_l^2 + a)(2y_l)^{-1} \ \ (L = M) \end{cases} \quad (4)$$

Here λ is the tangent point on the curve and $\mathcal{O}$ is the additive unity in $E(\mathbb{F}_p)$. When $L \neq M$, $L + M$ is called the elliptic curve addition (ECA). If $L = M$, then $L + M = 2L$, which is known as elliptic curve doubling (ECD).

- Scalar multiplication.

Let r be the order of the target rational point group and s be a scalar such that $0 \leq s < r$. Scalar multiplication of the rational point M, typically denoted as $[s]M$, can be calculated using $(s-1)$-times additions of M as follows:

$$[s]M = \underbrace{M + M + \cdots + M}_{s-1 \text{ times additions}}. \quad (5)$$

When $s = r$, where r is the order of the curve, then $[r]M = \mathcal{O}$. If $[s]M = N$ and s are unknown, then solving s from M and N is called an elliptic curve discrete logarithm problem (ECDLP). The security of elliptic curve cryptography lies on the difficulty of solving the ECDLP.

B. *Barreto-Naehring curve* [12]

The Barreto-Naehrig (BN) curve is a type of non-super-singular (ordinary) pairing-friendly elliptic curve of the embedding degree, $k = 12$, defined over $\mathbb{F}_{p^{12}}$, given as follows:

$$E: y^2 = x^3 + b, \ b \neq 0 \quad (6)$$

As a typical feature of the BN curve, its characteristic p, Frobenius trace t and order r are given as a polynomial of an integer χ, also known as the mother parameter as follows:

$$p(\chi) = 36\chi^4 + 36\chi^3 + 24\chi^2 + 6\chi + 1 \quad (7a)$$

$$r(\chi) = 36\chi^4 + 36\chi^3 + 18\chi^2 + 6\chi + 1 \quad (7b)$$

$$t(\chi) = 6\chi^2 + 1 \quad (7c)$$

The smallest positive integer k such that r divides $p^k - 1$ is called the *embedding degree*.

C. *Extension field arithmetic*

Pairing-based cryptography requires performing the arithmetic operation in extension fields of degree $k \geq 6$. Bailey *et al*. [19] explained the optimal extension field by towering using irreducible binomials, which is adopted for the BN curve as follows:

- Towering of the $\mathbb{F}_{p^{12}}$ extension field.

Let $6|(p-1)$, where p is the characteristic of the BN curve. In the context of the BN, where $k = 12$, $\mathbb{F}_{p^{12}}$ is constructed as a tower field with an irreducible binomial as follows:

$$\begin{cases} \mathbb{F}_{p^2} = \mathbb{F}_p[\alpha]/(\alpha^2 + 1) \\ \mathbb{F}_{p^4} = \mathbb{F}_{p^2}[\beta]/(\beta^2 - (\alpha + 1)) \\ \mathbb{F}_{p^{12}} = \mathbb{F}_{p^4}[\gamma]/(\gamma^3 - \beta) \end{cases} \quad (8)$$

To construct this tower of extension field, -1 should be a quadratic non-residue in $\mathbb{F}_p$ and $(\alpha + 1)$ should be a quadratic and cubic non-residue in $\mathbb{F}_{p^2}$.

D. *Ate and optimal-ate pairing on the BN Curve*

In the context of pairing-based cryptography, especially on the BN curve, two additive rational point groups $\mathbb{G}_1$ and $\mathbb{G}_2$, and a multiplicative group $\mathbb{G}_3$ of order r are considered. From [20], $\mathbb{G}_1$, $\mathbb{G}_2$ and $\mathbb{G}_3$ are defined as follows:

$$\mathbb{G}_1 = E(\mathbb{F}_{p^k})[r] \cap \mathrm{Ker}(\phi_p - [1])$$

$$\mathbb{G}_2 = E(\mathbb{F}_{p^k})[r] \cap \mathrm{Ker}(\phi_p - [p])$$

$$\mathbb{G}_3 = \mathbb{F}_{p^k}^* / \left(\mathbb{F}_{p^k}^*\right)^r$$

$E(\mathbb{F}_{p^k})[r]$ denotes the rational points of order r and $[n]$ denotes the n times scalar multiplication for a rational point. ϕ_p denotes the Frobenius mapping given by $\phi_p : (x, y) \mapsto (x^p, y^p)$ and $\mathrm{Ker}\,(\cdot)$ is a set whose elements are mapped to the zero element by $\cdot$. The Ate pairing [9] is generally given as follows:

$$e = \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_3, \quad (9)$$

In the case of BN curve, the above $\mathbb{G}_1$ is just $E(\mathbb{F}_p)$. In what follows, rest of this paper considers $P \in \mathbb{G}_1 \subset E(\mathbb{F}_p)$ and $Q \in \mathbb{G}_2 \subset E(\mathbb{F}_{p^{12}})$ for BN curve. Ate pairing $e\,(Q, P)$ is given as follows:

$$e(Q, P) = f_{t-1,Q}(P)^{\frac{p^{12}-1}{r}}, \quad (10)$$

where $f_{t-1,Q}(P)$ symbolizes the output of Miller's algorithm and $\lfloor \log_2(t-1) \rfloor$ is the loop length, where t is the Frobenius trace given in Eq. (7c). The bilinearity of Ate pairing is satisfied after calculating the final exponentiation $\frac{p^{12}-1}{r}$.

Vercauteren proposed a more efficient variant of the Ate pairing named as the optimal-Ate pairing [10], where the Miller's loop length was reduced to $\lfloor \log_2 s \rfloor$, $s = 6\chi + 2$.

$$e_{opt}(Q, P) = (f_{s,Q}(P) \cdot l_{[s]Q,[p]Q}(P) \cdot l_{[s+p]Q,[-p^2]Q})^{\frac{p^{12}-1}{r}} \quad (11)$$

where f is the main Miller loop's outcome. The authors applied 7-sparse multiplication for the line evaluations of Miller's algorithm described in Section 4. The ECA and

ECD are also calculated efficiently in the twisted curve. The $[p]Q$ and $[p^2]Q$ terms are calculated efficiently by applying a skew Frobenius map over $\mathbb{F}_{p^2}$, since Q can be mapped in the sub-field twisted isomorphic group. The final exponentiation is calculated by applying Fuentes et al.'s work for the BN curve [17].

E. Twist of the BN curves

There exists a twisted curve E' of order r isomorphic to the group where the rational point $Q \in E(\mathbb{F}_{p^k})$ belongs to. This sub-field isomorphic rational point group includes an isomorphic point of Q, typically denoted as $Q' \in E'(\mathbb{F}_{p^{k/d}})$, where k is the embedding degree and d is the twist degree. Since the points on the twisted curve are defined over a smaller field than $\mathbb{F}_{p^k}$, the ECA and ECD therefore become faster. However, when required in the pairing calculation, such as for the line evaluation, they can be quickly mapped to a point on $E(\mathbb{F}_{p^k})$. In the context of the BN curve, there exists a 6-th degree twist, also known as a sextic twist since $6|k$.

- Sextic twist of the BN curve.

When the embedding degree $k = 6e$, where $e = 2$ is a positive integer, the *sextic* twist is given as follows:

$$E : y^2 = x^3 + b,\ b \in \mathbb{F}_p \tag{12}$$

$$E' : y^2 = x^3 + b(\alpha + 1) \tag{13}$$

where $\alpha + 1$ is a quadratic and cubic non-residue in $E(\mathbb{F}_{p^e})$ and $3|p^e - 1$. The isomorphism between $E'(\mathbb{F}_{p^e})$ and $E(\mathbb{F}_{p^{6e}})$ is given as follows:

$$\psi_6 : \begin{cases} E'(\mathbb{F}_{p^e}) \to E(\mathbb{F}_{p^{6e}}), \\ (x, y) \mapsto \left(x(\alpha+1)^{-\frac{1}{3}}, y(\alpha+1)^{-\frac{1}{2}}\right) \end{cases} \tag{14}$$

III. Proposed Implementation Techniques

This section describes the efficient pairing implementation techniques on the BN curve for the parameters given in [14]. The overall contributions are can be summarized as follows:

- Efficient line evaluation of Miller's algorithm by 7-sparse multiplication.
- Efficient scalar multiplication of $\mathbb{G}_1$ and $\mathbb{G}_2$ by applying skew Frobenius mapping and Frobenius mapping over the twisted isomorphic curve $E'(\mathbb{F}_{p^2})$.
- Efficient exponentiation of the $\mathbb{G}_3$ points.

Fig. 1 shows the sextic twisted isomorphic mapping of $Q' \in E'(\mathbb{F}_{p^2}) \mapsto Q \in E(\mathbb{F}_{p^{12}})$ and isomorphic mapping of $P \in E(\mathbb{F}_p) \mapsto P' \in E'(\mathbb{F}_{p^{12}})$. The following sub-sections give the explicit formulas to implement the above ideas and in Section 5 give the comparative implementation results.

A. 7-Sparse Multiplication

Since the line equations inside of the Miller loop and outside are sparsely obtained (7 zero coefficients and 5 no-zero coefficients), the following optimized line calculation can be given as follows:

- Elliptic curve doubling $T = Q$.

$$A = \frac{1}{2y_{T'}}, B = 3x_{T'}^2, C = AB, D = 2x_{T'}, x_{2T'} = C^2 - D,$$
$$E = Cx_{T'} - y_{T'}, y_{2T'} = E - Cx_{2T'},$$
$$l_{T,T}(P) = y_P + (\alpha+1)^{-1}E\beta - (\alpha+1)^{-1}Cx_P\beta\gamma^2$$

- Elliptic curve addition $T \neq Q$.

$$A = \frac{1}{x_{Q'} - x_{T'}}, B = y_{Q'} - y_{T'}, C = AB, D = x_{T'} + x_{Q'},$$
$$x_{R'} = C^2 - D, E = Cx_{T'} - y_{T'}, y_{R'} = E - Cx_{R'},$$
$$l_{T,Q}(P) = y_P + (\alpha+1)^{-1}E\beta - (\alpha+1)^{-1}Cx_P\beta\gamma^2$$

Here the temporary variables A to E are in $\mathbb{F}_{p^2}$ and Q', T' are in $E'(\mathbb{F}_{p^2})$. The basis element β and $\beta\gamma^2$ identifies the coordinates position in the $\mathbb{F}_{p^{12}}$ vector.

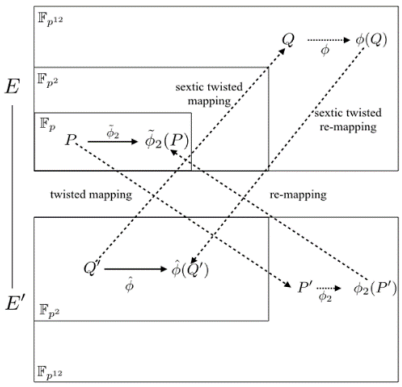


Fig. 1. The isomorphic map of $P \mapsto P'$ and the sextic twisted map of $Q' \mapsto Q$.

B. Scalar Multiplication in $\mathbb{G}_1$

- Skew Frobenius Map in $\mathbb{G}_1$.

The skew Frobenius map $\tilde{\phi}_e$ for $\mathbb{G}_1$ is defined as follows:

$$\tilde{\phi}_e : \begin{cases} E(\mathbb{F}_p) \to E(\mathbb{F}_p), \\ (x_p, y_p) \mapsto (x_p/v^{2(p^e-1)/d}, y_p/v^{3(p^e-1)/d}), \end{cases} \tag{15}$$

where $v = (\alpha + 1)^5$. In the case of the BN curves, since $k = 12$, $d = 6$ and $e = k/d = 2$, here $(\alpha + 1)^{5(p^2-1)/2}$ becomes -1 since $(\alpha + 1)$ is quadratic non-residue in $\mathbb{F}_{p^2}$ and $(\alpha + 1)^{5(p^2-1)/3}$ is a primitive cube root of 1, ϵ_3. Finally, the skew Frobenius map is given as follows:

$$\tilde{\phi}_2 : (x_p, y_p) \mapsto (x_p/(\alpha+1)^{5(p^2-1)/3}, y_p/v^{5(p^2-1)/2}),$$
$$= (x_p\epsilon_3, -y_p). \tag{16}$$

$\tilde{\phi}_2$ and p have following relationship and we can calculate $[p^2]P$ easily.

$$[p^2]P = \tilde{\phi}_2(P). \tag{17}$$

- $\mathbb{G}_1$ scalar multiplication with $\tilde{\phi}_e$.

The previous work of Sakemi *et al*. [15] shows the following relationships.

$$6\chi^2 - 4\chi + 1 \equiv (-2\chi + 1)p^2 \bmod r \tag{18}$$

Using this relationship, we can get the following relationships for the $\mathbb{G}_1$ scalar multiplication to consider the $(6\chi^2 - 4\chi + 1)$-adic representation of scalar s.

$$s = (s_5 - s_4)p^2 + (s_2 - s_5) \bmod r \quad (19)$$

However, s_2, s_4 and s_5 satisfiy these conditions using $\nu = 6\chi^2 - 4\chi + 1$ and $\mu = -2\chi + 1$.

$$\begin{cases} s = \nu s_1 + s_2 \\ \mu s_1 = \nu s_3 + s_4 \\ s_5 = \mu s_3 \end{cases}$$

When $A = s_5 - s_4$ and $B = s_2 - s_5$, $s[P]$ is calculated as follows:

$$[s]P = [A]\tilde{\phi}_2(P) + [B]P \quad (20)$$

C. Scalar multiplication in $\mathbb{G}_2$

- Skew Frobenius Map in $\mathbb{G}_2$

The skew Frobenius map $\hat{\phi}^l$ for $\mathbb{G}_2$ is defined as follows:

$$\hat{\phi}^l : \begin{cases} E'(\mathbb{F}_{p^2}) \to E'(\mathbb{F}_{p^2}) \\ (x, y) \mapsto (v^{1/3}(v^{-1/3}x)^{p^l}, v^{1/2}(v^{-1/2}y)^{p^l}) \end{cases} \quad (21)$$

In the case of the BN curves, $v = (\alpha + 1)$, where α is a root of the polynomial $x^2 + 1$. Then, $\hat{\phi}^l$ becomes

$$\hat{\phi}^l : (x, y) \mapsto ((\alpha + 1)^{1/3}((\alpha + 1)^{-1/3}x)^{p^l}, (\alpha + 1)^{1/2}((\alpha + 1)^{-1/2}y)^{p^l}) \quad (22)$$

$\hat{\phi}^l$ and p have following relationships and we can calculate $[6\chi]P$, $[6\chi^2]P$ and $[36\chi^3]P$ easily as given in [15].

$$[6\chi]P = -\{(1 + \hat{\phi}) + \hat{\phi}^3(1 - \hat{\phi})\}P$$
$$[6\chi^2]P = \hat{\phi}(P)$$
$$[36\chi^3]P = -\hat{\phi}^3\{(1 + \hat{\phi}) + \hat{\phi}^3(1 - \hat{\phi})\}P \quad (23)$$

- $\mathbb{G}_2$ scalar multiplication with $\hat{\phi}^l$

We can get following relationships for the $\mathbb{G}_2$ scalar multiplication to consider the $(6\chi^2)$-adic representation of scalar s.

$$s = (6\chi^2)A + B \quad (24)$$

Then, we consider (6χ)-adic for A and B as follows:

$$A = (6\chi)s_1 + s_2$$
$$B = (6\chi)s_3 + s_4 \quad (25)$$

Using this relationship between A and B, s can be represented as follows:

$$s = s_1(36\chi^3) + s_2(6\chi^2) + s_3(6\chi) + s_4 \quad (26)$$

Therefore, using Eq. (23) we can reduce the scalar multiplication by s as follows:

$$s = s_1[36\chi^3]P + s_2[6\chi^2]P + s_3[6\chi]P + s_4P \quad (27)$$

D. Exponentiation in $\mathbb{G}_3$

We can calculate the $\mathbb{G}_3$ exponentiation like the $\mathbb{G}_2$ scalar multiplication.

- Frobenius map in $\mathbb{G}_3$.

The Frobenius map ϕ^l is defined as follows:

$$\phi^l : \begin{cases} \mathbb{F}^*_{p^k} \to \mathbb{F}^*_{p^k}, \\ X \mapsto X^{p^l}. \end{cases} \quad (28)$$

ϕ^l and p have following relationship and we can calculate $f^{6\chi}$, $f^{6\chi^2}$ and $f^{36\chi^3}$ as follows:

$$f^{6\chi} = \{(f \cdot \phi(f)) \cdot \phi^3(f \cdot \phi(f)^{-1})\}^{-1},$$
$$f^{6\chi^2} = \phi(f),$$
$$f^{36\chi^3} = \phi^3\left(\{(f \cdot \phi(f)) \cdot \phi^3(f \cdot \phi(f)^{-1})\}^{-1}\right) \quad (29)$$

- $\mathbb{G}_3$ exponentiation with ϕ^l

We can get the same relationship as the $\mathbb{G}_2$ scalar multiplication. Then, s also can be represented as Eq. (26). Therefore, using Eq. (29) we can efficiently calculate the exponentiation using s as follows:

$$f^s = (f^{36\chi^3})^{s_1} \cdot (f^{6\chi^2})^{s_2} \cdot (f^{6\chi})^{s_3} \cdot f^{s_4}. \quad (30)$$

IV. Evaluation of the Results

This section gives a comparative implementation of the techniques described in Section 3.

- Environment

We implemented the proposal using C language. For large-integer arithmetic GMP 6.1.1 has been used. The programs were tested on a Raspberry Pi 3 B and PC. The Raspberry Pi was specified as 1.2 GHz 64 bit CPU, 1 GB RAM, gcc compiler version 4.9.2 on Raspbian 4.9.2-10 OS. The PC is equipped with a core i5 3.3 GHz CPU, 8 GB RAM with Ubuntu 16.04 OS, gcc ver-5.4.0. In both cases only a single core was utilized.

- Parameter and result analysis.

As said before this paper uses the most recent parameters [14] where the mother parameter is $\chi = 2^{114} + 2^{101} - 2^{14} - 1$, which is resistant to exTNFS [13] for 128 bit security level. The size of the ECDL, length$(r) = 462$ bit and the DLP length $(p^k) = 5544$ bit. For $\mathbb{G}_1$ and $\mathbb{G}_2$ SCM an integer $s \approx r$ was considered. Table I shows the time comparison of the implementation. The source code can be found in Github.[1]

Table I: The Time Comparison of Pairing the SCM in $\mathbb{G}_1$, $\mathbb{G}_2$ and Exponentiation in $\mathbb{G}_3$

Operation		Time [ms]	
Device type		Raspberry pi	PC
Pairing	Miller Algo.	$\mathbf{8.54} \times 10^1$	$\mathbf{8.15} \times 10^0$
	Final Exp.[17]	2.29×10^2	1.94×10^1
	Total	3.15×10^2	2.76×10^1
$\mathbb{G}_1$ SCM	Previous	2.81×10^1	3.76×10^0
	This	$\mathbf{1.66} \times 10^1$	$\mathbf{2.24} \times 10^0$
$\mathbb{G}_2$ SCM	Previous	7.96×10^1	8.00×10^0

[1] https://github.com/YukiNanjo/BN12raspi

Operation		Time [ms]	
	This: 2-split	4.62×10^1	4.53×10^0
	This: 4-split	$\mathbf{2.71} \times 10^1$	$\mathbf{2.76} \times 10^0$
$\mathbb{G}_3$ Exp.	Previous	2.31×10^2	2.02×10^1
	This	$\mathbf{8.48} \times 10^1$	$\mathbf{6.62} \times 10^0$

Here 2-split and 4-split refers to dividing the scalar s into 2 and 4 parts, respectively. "Previous" refers to the implementation without any optimization and "This" refers to the optimization given in Section 3. It is clear from the results that the given implementation methods are faster than the "Previous" method. The time is different in two environments but the ratio is same. In both environments the $\mathbb{G}_1$ SCM is two times faster than the "Previous" approach. Similarly, the $\mathbb{G}_2$ 2-split is about 2 times, 4 splits are 3 times and $\mathbb{G}_3$ exp is about 3 times faster than the "Previous" method. However, since it shows a high-level general implementation, it is not practical for raspberry pi. Implementing the proposed optimized methods using the ARM-NEON assembly will be the most efficient, which is not the focus of this work.

V. Conclusions

This paper shows the techniques for efficient pairing calculations on the BN curve for 128 bit security level. In addition, it also shows the state of art techniques for efficient scalar multiplication in the $\mathbb{G}_1$, $\mathbb{G}_2$ groups together with $\mathbb{G}_3$ exponentiation. The implementation result on the raspberry pi and PC substantiated the proposed techniques efficiency. As future work, the authors would like to apply this pairing on a customized SSL/TLS suite for authentication using ID-based encryption.

Acknowledgments

This work was partially supported by the Strategic Information and Communications R&D Promotion Programme (SCOPE) of Ministry of Internal Affairs and Communications, Japan.

References

[1] W. Diffie and M. Hellman, "New directions in cryptography," *IEEE Transactions on Information Theory*, vol. 22, no. 6, pp. 644–654, 1976.

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

[3] V. S. Miller, "Use of elliptic curves in cryptography," in *Proc. Conference on the Theory and Application of Cryptographic Techniques*, Springer, 1985, pp. 417–426.

[4] N. Koblitz, "Elliptic curve cryptosystems," *Mathematics of Computation*, vol. 48, no. 177, pp. 203–209, 1987.

[5] R. Sakai, "Cryptosystems based on pairing," in *Proc. Symposium on Cryptography and Information Security, Okinawa*, Japan, Jan. 2000, pp. 26–28.

[6] A. Joux, "A one round protocol for tripartite diffie–hellman," in *Proc. International Algorithmic Number Theory Symposium*., Springer, 2000, pp. 385–393.

[7] D. Boneh, B. Lynn, and H. Shacham, "Short signatures from the weil pairing," *Cryptology-ASIACRYPT 2001*, pp. 514–532, 2001.

[8] A. Sahai and B. Waters, "Fuzzy identity-based encryption," in *Proc. Annual International Conference on the Theory and Applications of Cryptographic Techniques*, Springer, 2005, pp. 457–473.

[9] H. Cohen, G. Frey, R. Avanzi, C. Doche, T. Lange, K. Nguyen, and F. Vercauteren, *Handbook of Elliptic and Hyperelliptic Curve Cryptography*, CRC press, 2005.

[10] F. Vercauteren, "Optimal pairings," *IEEE Transactions on Information Theory*, pp. 455–461, 2010.

[11] Y. Nogami, M. Akane, Y. Sakemi, H. Kato, and Y. Morikawa, "Integer variable χ–based ate pairing," in *Proc. International Conference on Pairing-Based Cryptography*, Springer, 2008, pp. 178–191.

[12] P. S. Barreto and M. Naehrig, "Pairing-friendly elliptic curves of prime order," in *Proc. International Workshop on Selected Areas in Cryptography*, Springer, 2005, pp. 319–331.

[13] T. Kim and R. Barbulescu, "Extended tower number field sieve: A new complexity for the medium prime case," in *Proc. Advances in Cryptology - CRYPTO 2016 Proceedings, Part I*. Springer, 2016, pp. 543–571.

[14] R. Barbulescu and S. Duquesne, "Updating key size estimations for pairings," Cryptology ePrint Archive, Report 2017/334, 2017.

[15] Y. Sakemi, Y. Nogami, K. Okeya, H. Kato, and Y. Morikawa, "Skew frobenius map and efficient scalar multiplication for pairing–based cryptography," in *Proc. International Conference on Cryptology and Network Security*, Springer, 2008, pp. 226–239.

[16] G. Grewal, R. Azarderakhsh, P. Longa, S. Hu, and D. Jao, "Efficient implementation of bilinear pairings on arm processors," in *Proc. International Conference on Selected Areas in Cryptography*, Springer, 2012, pp. 149–165.

[17] L. Fuentes-Castaneda, E. Knapp, and F. Rodrıguez-Henrıquez, "Faster hashing to g2," *Selected Areas in Cryptography*, vol. 7118, pp. 412–430, 2011.

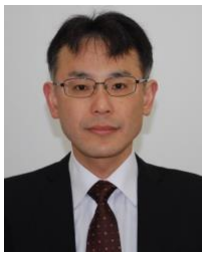
[18] L. C. Washington, *Elliptic Curves: Number Theory and Cryptography*, CRC press, 2008.

[19] D. V. Bailey and C. Paar, "Efficient arithmetic in finite field extensions with application in elliptic curve cryptography," *Journal of Cryptology*, vol. 14, no. 3, pp. 153–176, 2001.

[20] Y. Mori, S. Akagi, Y. Nogami, and M. Shirase, "Pseudo 8–sparse multiplication for efficient ate–based pairing on barreto–naehrig curve," *Pairing-Based Cryptography–Pairing*, pp. 186–198, 2013.

Yuki Nanjo is in the final year of her Bachelor degree in Electrical and Communication Engineering at Okayama University, Japan. She is working on information security engineering under the supervision of Professor Dr. Yasuyuki NOGAMI. Her main fields of research are information security and pairing-based cryptography.

Md. Al-Amin Khandaker graduated from Jahangirnagar University in 2011. He is now pursuing his Ph.D. in the field of Finite field theory and cryptography in Okayama University under the supervision of Professor Dr. Yasuyuki NOGAMI. His main fields of research are pairing-based cryptography and its applications. He is a graduate student member of the IEEE.



Takuya Kusaka received his B.E. degree in Electric Engineering from Kobe University in 1994 and then received M.E. and Ph.D. degrees in Information Science from the Graduate School of Information Science, Nara Institute of Science and Technology in 1996 and 1999, respectively. In 2004, he joined Okayama University.

Yasuyuki Nogami graduated from Shinshu University in 1994 and received his Ph.D. degree in 1999 from Shinshu University. He is now a Professor at Okayama University. His research focuses on finite field theory and its applications such as recent public key cryptographies. Recently, he is a member of security research group in Okayama University and particularly focuses on IoT security. He is a member of the IEICE and IEEE.