RESEARCH ARTICLE

Security analysis of elliptic curves with embedding degree 1 proposed in PLOS ONE 2016

Tadanori Teruya *

Cyber Physical Security Research Center, National Institute of Advanced Industrial Science and Technology, Aomi, Koto-ku, Tokyo, Japan

* tadanori.teruya@aist.go.jp

Abstract

Wang et al. proposed a method for obtaining elliptic curves with embedding degree 1 for securing critical infrastructures, and presented several elliptic curves generated by their method with torsion points of 160 bits and 189 bits orders. They also presented some experimental results and claimed that their implementation of an elliptic curve generated with their method is faster than an implementation for embedded devices presented by Bertoni et al. In this paper, we point out that the security and efficiency claims given by Wang et al. are flawed. Specifically, we show that it is possible to solve finite field discrete logarithm problems defined over their elliptic curves in practice. On the elliptic curves with torsion points of 160 bits orders generated by Wang et al., their instances of finite field discrete logarithm problems are solved in around 4 hours by using a standard desktop PC. On the torsion points of 189 bits orders, their instances are solved in around 10 days by using two standard desktop PCs. The hardness of the finite field discrete logarithm problems is one of the most important bases of security; therefore, their elliptic curves should not be used for cryptographic purposes.

OPEN ACCESS

Citation: Teruya T (2019) Security analysis of elliptic curves with embedding degree 1 proposed in PLOS ONE 2016. PLoS ONE 14(2): e0212310. <https://doi.org/10.1371/journal.pone.0212310>

Editor: Hua Wang, Victoria University, AUSTRALIA

Received: February 10, 2018

Accepted: January 31, 2019

Published: February 19, 2019

Copyright: © 2019 Tadanori Teruya. This is an open access article distributed under the terms of the [Creative Commons Attribution License](https://creativecommons.org/licenses/by/4.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Data Availability Statement: All relevant data are within the paper and its Supporting Information files.

Funding: The author received no specific funding for this work.

Competing interests: The author has declared that no competing interests exist.

Introduction

Since 2000, many researchers have proposed efficient and useful cryptographic schemes for securing systems using a *pairing*, which is a bilinear map defined over an elliptic curve. For example, Sakai et al. proposed a non-interactive key-exchange scheme [1], Joux proposed a tripartite key-exchange scheme [2], Boneh and Franklin proposed an identity-based encryption scheme [3], Boneh et al. proposed a short digital signature scheme [4], and Groth and Sahai proposed efficient non-interactive zero-knowledge proof systems [5]. This research field is called *pairing-based cryptography* because a pairing is used as a building block. As mentioned above, the pairings allow us to implement many efficient and useful cryptographic schemes, and pairing-based cryptography is currently one of the major fields of cryptographic research [6, 7].

For security requirements, it is necessary for any implementation of a pairing-based cryptographic scheme to appropriately select an elliptic curve. Its efficiency should then be improved for practical performance requirements. Since the computational costs of pairings and group operations defined over elliptic curves are expensive, investigation of fast algorithms and implementations is an important research topic, and there have been many studies on the mathematical foundation for algorithms [8–26], and efficient implementations [27–34]. A comprehensive survey was presented by [7].

Thanks to these studies, the development of pairing-based cryptosystems has progressed in not only theory but also practice. However, implementation is still difficult. The most difficult problem is the selection of appropriate parameters to instantiate the schemes securely and efficiently. This is because all algorithms correctly work, even if a selected parameter is vulnerable. It is necessary to carefully evaluate whether the selected parameters are robust against cryptanalysis.

Recently, Wang et al. [35] proposed a method for obtaining elliptic curves with embedding degree 1 in order to instantiate pairing-based cryptographic schemes for securing critical infrastructures. They presented some experimental results and claimed that their pairing implementation of an elliptic curve generated with their method is faster than another secure and efficient implementation presented by Bertoni et al. [36, 37]. The aim of Bertoni et al. [36, 37] is to develop efficient implementation of pairings for embedded devices.

In this paper, we point out that there is a serious issue with the security and efficiency claims given by Wang et al. [35]. In short, their elliptic curves are insecure and should not be used for cryptographic purposes. Wang et al. [35] presented several elliptic curves generated by their method. The bit lengths of orders of these curves are 160 bits and 189 bits, and these elliptic curves are described in the files [38, 39] of their supporting information. According to [25, 40, 41], these bit lengths are large enough to guarantee security in practice. However, we demonstrate that finite field discrete logarithm problems defined on these elliptic curves are solvable in practice by using standard desktop PCs. The hardness of finite field discrete logarithm problems is one of the most important bases of security; therefore, their elliptic curves should not be used for cryptographic purposes. In addition, we present security and efficiency analyses of the method of Wang et al. [35]. Based on these analyses, we conclude that the claims of security and efficiency given by Wang et al. [35] are flawed.

Recently, Chatterjee et al. [24] proposed proper constructions of pairing-based cryptographic schemes on elliptic curves with embedding degree 1. Their considerations and analyses of security and efficiency of their constructions are comprehensively discussed based on state-of-the-art results; therefore, we do not discuss how to repair the method of Wang et al. [35] in this paper. We strongly recommend that the readers refer to the study by Chatterjee et al. [24].

The rest of this paper is organized as follows. We introduce preliminaries of pairing-based cryptography. Next, we also introduce finite field discrete logarithm problems and elliptic curve discrete logarithm problems in pairing-based cryptography. We show how to produce finite field discrete logarithm problems from elliptic curves generated by Wang et al. [35], and then we show that they are solvable by using standard desktop PCs. Then we explain why their claims are flawed. Finally, we conclude the paper.

Materials and methods

Mathematical preliminaries of pairings

In this section, we introduce the notations and terminology of pairings defined over elliptic curves [6, 7].

Let S be a finite set, then we denote the number of elements in S by $\#S$. Let p be a prime number, and k be a positive integer. We denote by $\mathbb{F}_p$ a finite field whose field order is p , by $\mathbb{F}_{p^k}$ its k -th extension field, and by $\mathbb{F}_{p^k}^*$ the multiplicative group of $\mathbb{F}_{p^k}$. For $\mathbb{F}_{p^k}$, its characteristic and extension degree are p and k , respectively. We assume that $p > 3$, and an elliptic curve E defined over $\mathbb{F}_p$ is defined by the Weierstrass equation

$$Y^2 = X^3 + aX + b, \quad (1)$$

where X and Y are two variables, and $a, b \in \mathbb{F}_p$ with $4a^3 + 27b^2 \neq 0$. We define the $\mathbb{F}_p$ -rational points of E as

$$E(\mathbb{F}_p) := \{(x, y) \in \mathbb{F}_p \times \mathbb{F}_p \mid y^2 = x^3 + ax + b\} \cup \{\infty\}, \quad (2)$$

where ∞ is the point at infinity. Note that $E(\mathbb{F}_p)$ and its group operation, which is denoted by addition symbol “+” in this paper, forms an abelian group whose unit element is ∞ and inverse operation is $-(x, y) := (x, -y)$. Let $P \in E(\mathbb{F}_p)$ and let a be an integer. We denote the scalar multiplication of P by a as

$$[a]P := \underbrace{P + P + \cdots + P}_{a \text{ repetitions}} \quad (3)$$

if $a > 0$, $[a]P := [-a](-P)$ if $a < 0$, and $[0]P := \infty$. We say that P is order n if n is the smallest positive integer such that $[n]P = \infty$. We also denote an additive cyclic group generated by P as

$$\langle P \rangle := \{Q \mid \text{there exists } a \in \mathbb{Z} \text{ such that } Q = [a]P\}, \quad (4)$$

and a multiplicative cyclic group generated by x as

$$\langle x \rangle := \{y \mid \text{there exists } a \in \mathbb{Z} \text{ such that } y = x^a\}. \quad (5)$$

We call P and x are generators of $\langle P \rangle$ and $\langle x \rangle$, respectively. We also call the number of elements of a group the group order.

Let r be a prime number with $r \neq p$. Then we define the r -torsion points as

$$E[r] := \{P \in E(\overline{\mathbb{F}}_p) \mid [r]P = \infty\}, \quad (6)$$

where $\overline{\mathbb{F}}_p$ is the algebraic closure of $\mathbb{F}_p$, and we call r the order of $E[r]$. The *embedding degree* k of E with respect to r is the smallest positive integer such that $r \mid (p^k - 1)$, and this property ensures $\mu_r \subseteq \mathbb{F}_{p^k}^*$ and $E[r] \subseteq E(\mathbb{F}_{p^k})$, where $\mu_r := \{x \in \mathbb{F}_{p^k}^* \mid x^r = 1\}$ is the r -th roots of unity [42]. Note that $\mu_r = \langle x \rangle$ for all $x \in \mu_r \setminus \{1\}$ if r is prime.

Let $P, Q \in E[r]$, then we define the *reduced Tate pairing* $t: E[r] \times E[r] \rightarrow \mu_r$. Note that the reduced Tate pairing t has the following three properties: First, it is non-degenerate, i.e., for all elements $P \in E[r] \setminus \{\infty\}$, there is an element $Q \in E[r]$ with $e(P, Q) \neq 1$, and for all elements $Q \in E[r] \setminus \{\infty\}$, there is an element $P \in E[r]$ with $e(P, Q) \neq 1$. Second, it is bilinear, i.e., for all $P, Q, S \in E[r]$, $t(P + S, Q) = t(P, Q) \cdot t(S, Q)$ and $t(P, Q + S) = t(P, Q) \cdot t(P, S)$. Third, it is efficiently computable, i.e., its computational time complexity is in the polynomial of $\log r$ (see [8, 9]).

Discrete logarithm problems of pairing-based cryptography

Every pairing-based cryptographic scheme requires the hardness of underlying mathematical problems in order to guarantee security in practice. In this section, we introduce two discrete

logarithm problems which are important underlying mathematical problems for pairing-based cryptographic schemes.

Let E be an elliptic curve defined over a finite field $\mathbb{F}_p$, let r be a prime number with $r \neq p$, and let k be the embedding degree of E with respect to r . We define the following two discrete logarithm problems of E :

- *Elliptic curve discrete logarithm problem in $E[r]$ (ECDLP)*: Given $P \in E[r] \setminus \{\infty\}$ and $Q = [x]P$, where x is a randomly chosen integer from $\{1, \dots, r-1\}$, find x . We call a pair of P and Q an instance of ECDLP, and denote its solution by $x = \log_P Q$.
- *Finite field discrete logarithm problem in $\mu_r \subseteq \mathbb{F}_{p^k}^*$ (FFDLP)*: Given $g \in \mu_r \setminus \{1\}$ and $h = g^y$, where y is a randomly chosen integer from $\{1, \dots, r-1\}$, find y . We call a pair of g and h an instance of FFDLP, and denote its solution by $y = \log_g h$.

In general, the hardness of solving ECDLP is determined by the robustness against the best solving algorithm for ECDLP. The computational time complexity of the algorithm strongly depends on the bit length of order r , and the larger the bit length of r is, the harder ECDLP becomes. On the other hand, the hardness of solving FFDLP is also determined by the robustness against the best solving algorithm for FFDLP. Also, the computational time complexity of the algorithm strongly depends on the bit length of field order p^k , and the larger the bit length of p^k is, the harder FFDLP becomes.

Note that Menezes et al. [43], and Frey and Rück [44] pointed out that ECDLP is reduced to FFDLP by using the Weil pairing or the (reduced) Tate pairing. We call this reduction the *pairing reduction*. For example, given an instance of ECDLP P and $Q = [x]P$ as above, one can choose an arbitrary element $T \in E[r]$ with $t(P, T) \neq 1$ then construct an FFDLP instance $g := t(P, T)$ and $h := t(Q, T) = t([x]P, T) = t(P, T)^x$; thus, $\log_P Q = \log_g h$. As mentioned above, every cryptographic scheme requires the hardness of ECDLP and FFDLP. Here it is important to note that *the overall hardness of every cryptographic scheme is determined by the weakest underlying problem*. Hence, developers of pairing-based cryptosystems should use appropriate elliptic curves such that *both ECDLP and FFDLP are intractable simultaneously*.

As mentioned above, the longer the bit lengths of r and p^k are, the harder ECDLP and FFDLP, respectively, become and imply stronger robustness against cryptanalysis. However, the longer bit lengths of r and p^k cause significant efficiency loss. Hence, it is a very important task to find an elliptic curve which has r and p^k achieving reasonable robustness (i.e., security) and efficiency simultaneously. Since we focus on solving FFDLP reduced from ECDLP of elliptic curves generated by Wang et al. [35], finding appropriate elliptic curves is out of scope of this paper, and we refer the reader to [25, 40, 45, 46] for the details.

Elliptic curves with embedding degree 1 proposed in PLOS ONE 2016

Wang et al. [35] presented two types of elliptic curves with embedding degree 1 generated by their method. These are described in the files [38, 39] of their supporting information. For each type, there are 10 elliptic curves. Hence, there are 20 elliptic curves in total.

In this paper, we denote these two types of elliptic curves [38] and [39] as W160 and W189, respectively. In W160, each elliptic curve E is defined over a finite field $\mathbb{F}_{p_1}$ with $E[r_1] \subseteq E(\mathbb{F}_{p_1})$, where p_1 and r_1 are two distinct prime numbers. The bit lengths of r_1 and p_1 are 160 bits and 319 bits, respectively. In W189, each elliptic curve E is defined over a finite field $\mathbb{F}_{p_2}$ with $E[r_2] \subseteq E(\mathbb{F}_{p_2})$, where p_2 and r_2 are two distinct prime numbers. The bit lengths of r_2 and p_2 are 189 bits and 377 bits, respectively.

Results

In this section, we demonstrate that the FFDLP instances reduced from ECDLP instances of elliptic curves generated by Wang et al. [35] are easily solvable. Wang et al. [35] claimed that implementations of pairings defined over these elliptic curves are more efficient than an implementation of Bertoni et al. [36, 37]. To the best of our knowledge, ECDLP and FFDLP of an elliptic curve implemented by Bertoni et al. [36, 37] are intractable, so that we naturally expected that ECDLP and FFDLP of the elliptic curves implemented by Wang et al. [35] are also intractable. However, this is not the case, and hence, the security claim given by Wang et al. [35] is clearly flawed, and the efficiency claim is also flawed because FFDLP instances reduced from ECDLP instances of their elliptic curves are solvable in practice.

Problem instance generation

In this section, we explain how we generate the FFDLP instances reduced from ECDLP instances, which we solve.

To demonstrate that FFDLP instances are easily solvable in the W160 and W189 elliptic curves, the FFDLP instances should be generated without knowing their solutions. However, recall that the definition of FFDLP is that its solution is randomly chosen by an instance generator, and this means that the instance generator knows the solution. To generate FFDLP instances without knowing their solutions, we use well-known methodology [47–50] and the pairing reduction [43, 44].

Concretely, we generate h , which is a part of FFDLP instances, by using the ratio π of a circle's circumference to its diameter. For an elliptic curve E defined over a finite field $\mathbb{F}_p$, we compute $h := (\lfloor \pi \cdot 2^\ell \rfloor)^c \bmod p$, where ℓ is the largest integer such that $\lfloor \pi \cdot 2^\ell \rfloor < p$, and $c = (p - 1)/r$. Note that $h \in \mu_r$. For the W160 and W189 elliptic curves, we find the following two largest integers $\lfloor \pi \cdot 2^{316} \rfloor$ and $\lfloor \pi \cdot 2^{374} \rfloor$, respectively, that are less than p_1 and p_2 , respectively. We use $h_1 := (\lfloor \pi \cdot 2^{316} \rfloor)^{c_1} \bmod p_1$ and $h_2 := (\lfloor \pi \cdot 2^{374} \rfloor)^{c_2} \bmod p_2$ for W160 and W189, respectively, where $c_1 = (p_1 - 1)/r_1$ and $c_2 = (p_2 - 1)/r_2$, respectively. The integer values of $\lfloor \pi \cdot 2^{316} \rfloor$, h_1 , $\lfloor \pi \cdot 2^{374} \rfloor$, and h_2 are described in files S1 and S2 Files of our supporting information.

Next, Wang et al. [35] provided elements $P, Q \in E[r]$ for all the elliptic curves, so there are 20 pairs of P, Q . We use them to generate 20 elements $g = t(P, Q)$ for the remaining part of FFDLP instances. Note that, to compute $t(P, Q)$, we use a function of the reduced Tate pairing implemented in Sage [51].

Equipment

We explain our equipment to solve FFDLP instances. To solve FFDLP instances reduced from ECDLP instances of the W160 elliptic curves, we use one desktop PC with a Core i7-6700 (3.4) CPU. For the W189 elliptic curves, we use two desktop PCs with Core i7-6700 (3.4) and Core i7-4770 (3.4) CPUs. In these PCs, the size of equipped random access memories is 32, and the operating system is Ubuntu 16.04. We use the mathematics software system Sage [51] version 7.6 to generate FFDLP instances, the FFDLP solver CADO-NFS [52] version 2.3.0-rc1, and compiler gcc version 5.4.0.

Solutions and calculation times of finite field discrete logarithm problems

We solve the FFDLP instances defined above. For the W160 and W189 elliptic curves, calculation times to obtain solutions are around 3 hours 45 minutes and 240 hours 4 minutes,

respectively. Lists of FFDLP instances and their solutions are shown in [S1](#) and [S2](#) Files of our supporting information. Additionally, we present a verification script in [S3 File](#).

According to the results of our demonstration, we conclude that the security and efficiency claims given by Wang et al. [35] are flawed, and all the elliptic curves described in the files [38, 39] of their supporting information should not be used for cryptographic purposes.

Discussion

We have already shown that the FFDLP instances reduced from ECDLP instances of all the elliptic curves generated by Wang et al. [35] are practically solvable. In this section, we discuss why the security and efficiency claims given by Wang et al. [35] are flawed.

Wang et al. [35] presented the performance comparison of the reduced Tate pairing between their implementation of a W160 elliptic curve [38] and an implementation presented by Bertoni et al. [36]. Wang et al. [35] claimed that their implementation is faster than the latter. To revisit this comparison, we briefly introduce the implementation for embedded devices presented by Bertoni et al. [36]. Bertoni et al. [36, 37] implemented a pairing defined over another elliptic curve whose embedding degree is 2, and this elliptic curve is defined over $\mathbb{F}_p$, where the bit length of p is 512 bits. We denote this elliptic curve as B160. Now, we show the bit lengths of orders r and field orders p^k of the W160, W189, and B160 elliptic curves in [Table 1](#).

We consider the hardness of ECDLP and FFDLP of these elliptic curves based on state-of-the-art reports of hardness evaluation.

For ECDLP, Bernstein et al. [41] gave a state-of-the-art report on December 2nd 2016. They solved an ECDLP instance, where the bit length of order r is 117.35 bits. In general, the hardness of ECDLP strongly depends on the bit length of r . Therefore, solving ECDLP instances should be considered feasible if the bit length of r is less than 117.35 bits.

On the other hand, according to Grémy and Guillevic [53], state-of-the-art reports for solving FFDLP were given by Adrian et al. [54], Kleinjung et al. [50], Fried et al. [55], and Barbu-lescu et al. [49] on May 20th 2015, June 16th 2016, October 10th 2016, and April 29th 2015, respectively, and they solved FFDLP instances over finite fields $\mathbb{F}_p^*$ and $\mathbb{F}_{p^2}^*$, where the bit lengths of their field orders are 512 bits, 768 bits, 1024 bits, and 595 bits, respectively. In general, the hardness of FFDLP strongly depends on the bit length of p^k . Fried et al. [55] solved the largest bit length of FFDLP; however, their technique to solve FFDLP is only applicable to certain finite fields that have a special property, and thus this result is not relevant for the discussion of the hardness of W160, W189, and B160 elliptic curves (and to the best of our knowledge, it seems to be not satisfied by these curves). Therefore, for embedding degrees $k = 1$ and 2, solving FFDLP instances should be considered feasible if the bit length of p^k is less than 768 bits. It is clear that FFDLP instances of the W160 and W189 elliptic curves [38, 39] are practically solvable, but instances of B160 are not. In fact, we solved the FFDLP instances of the W160 and W189 elliptic curves [38, 39], and this immediately implies that the performance comparison given by Wang et al. [35] is definitely unfair.

Table 1. Bit lengths of orders and field orders of elliptic curves proposed by Wang et al. and Bertoni et al.

	Bit length of r (bit)	Bit length of p^k (bit)
W160	160	319
W189	189	377
B160	160	1023

<https://doi.org/10.1371/journal.pone.0212310.t001>

For a fair comparison, both ECDLP and FFDLP should be intractable, and the hardness of both ECDLP and FFDLP should ideally be equal. To the best of our knowledge, using the method of Wang et al. [35] to achieve the same hardness with B160, the bit lengths of r and p should be greater than or equal to at least 160 bits and 1023 bits, respectively, and the resulting elliptic curve should have almost the same hardness of FFDLP as a B160 elliptic curve [36, 37]. However, their method seems to output an elliptic curve, where the bit length of the resulting p is twice that of the resulting r . Therefore, the bit length of r should be around 512 bits, and this is larger than for the B160 elliptic curve [36, 37]. Both ECDLP and FFDLP defined over these elliptic curves should be intractable, and the hardness of FFDLP is almost the same because the bit length of p^k is almost the same. However, the resulting elliptic curve is defined over $\mathbb{F}_p$, where the bit length of p is around 1023 bits and this is larger than that of B160 (recall that the B160 elliptic curve is defined over $\mathbb{F}_p$, where the bit length of p is 512 bits, because its embedding degree is 2). In fact, larger bit lengths of p and r generally cause efficiency loss. Therefore, in the *same security level*, the implementation of B160 [36, 37] should be faster than implementations of elliptic curves generated by the method of Wang et al. [35].

From the discussions and considerations above, we conclude that there is no merit of the method proposed by Wang et al. [35].

Conclusion

In this paper, we demonstrated that instances of finite field discrete logarithm problems derived from elliptic curves [38, 39] are solvable in practice. These elliptic curves are generated by the method of Wang et al. [35]. In our demonstration, the instances were generated without knowing their solutions, and they were solved by using standard desktop PCs in reasonable time. The hardness of discrete logarithm problems is one of the most important bases of security; therefore, the elliptic curves described in [35, 38, 39] should not be used for cryptographic purposes. We also pointed out that the efficiency evaluation given by Wang et al. [35] is unfair. From our demonstrations and discussions, it is clear that the security and efficiency claims given by Wang et al. [35] are flawed.

Finally, we recommend that the readers refer to the paper by Chatterjee et al. [24]. They presented careful and comprehensive discussions and proper constructions of pairing-based cryptographic schemes on elliptic curves whose embedding degree is 1, based on state-of-the-art results.

Supporting information

S1 File. List of inputs and solutions of FFDLP instances of 160 bits (W160) elliptic curves.

This file contains five integers r_1 , p_1 , ℓ_1 , s_1 , and h_1 , and ten sets of values b , P , Q , g , and y , and these symbols are defined as follows:

- r_1 is the order of each $E[r_1] \subseteq E(\mathbb{F}_{p_1})$.
- p_1 is the field order of defining the finite field $\mathbb{F}_{p_1}$ of each E .
- $\ell_1 = 316$.
- $s_1 = \lfloor \pi \cdot 2^{\ell_1} \rfloor$, where π is the ratio of a circle's circumference to its diameter.
- $h_1 = s_1^c \in \mu_{r_1}$, where $c = (p_1 - 1)/r_1$.
- b is a coefficient of each Weierstrass equation, which defines each E .
- P and Q are contained in each $E[r_1]$.

- $g = t(P, Q)$.
- $y = \log_g h_1$.

Note that all p_1 , r_1 , b , P , and Q are generated by Wang et al. [35] and are described in the file [38].

(TXT)

S2 File. List of inputs and solutions of FFDLP instances of 189 bits (W189) elliptic curves.

This file contains five integers r_2 , p_2 , ℓ_2 , s_2 , and h_2 , and ten sets of values b , P , Q , g , and y , and these symbols are defined as follows:

- r_2 is the order of each $E[r_2] \subseteq E(\mathbb{F}_{p_2})$.
- p_2 is the field order of defining the finite field $\mathbb{F}_{p_2}$ of each E .
- $\ell_2 = 374$.
- $s_2 = \lfloor \pi \cdot 2^{\ell_2} \rfloor$, where π is the ratio of a circle's circumference to its diameter.
- $h_2 = s_2^c \in \mu_{r_2}$, where $c = (p_2 - 1)/r_2$.
- b is a coefficient of each Weierstrass equation, which defines each E .
- P and Q are contained in each $E[r_2]$.
- $g = t(P, Q)$.
- $y = \log_g h_2$.

Note that all p_2 , r_2 , b , P , and Q are generated by Wang et al. [35] and are described in the file [39].

(TXT)

S3 File. Sage script for verification. This Sage script verifies our results described in S1 and S2 Files.

(SAGE)

Author Contributions

Conceptualization: Tadanori Teruya.

Data curation: Tadanori Teruya.

Investigation: Tadanori Teruya.

Methodology: Tadanori Teruya.

Project administration: Tadanori Teruya.

Resources: Tadanori Teruya.

Software: Tadanori Teruya.

Validation: Tadanori Teruya.

Writing – original draft: Tadanori Teruya.

Writing – review & editing: Tadanori Teruya.

References

1. Sakai R, Ohgishi K, Kasahara M. Cryptosystems Based on Pairing. In: Symposium on Cryptography and Information Security; 2000. p. 26–28.
2. Joux A. A One Round Protocol for Tripartite Diffie-Hellman. *J Cryptology*. 2004; 17(4):263–276. <https://doi.org/10.1007/s00145-004-0312-y>
3. Boneh D, Franklin MK. Identity-Based Encryption from the Weil Pairing. *SIAM J Comput*. 2003; 32(3): 586–615. <https://doi.org/10.1137/S0097539701398521>
4. Boneh D, Lynn B, Shacham H. Short Signatures from the Weil Pairing. *J Cryptology*. 2004; 17(4): 297–319. <https://doi.org/10.1007/s00145-004-0314-9>
5. Groth J, Sahai A. Efficient Noninteractive Proof Systems for Bilinear Groups. *SIAM J Comput*. 2012; 41(5):1193–1232. <https://doi.org/10.1137/080725386>
6. Hoffstein J, Pipher J, Silverman JH. An Introduction to Mathematical Cryptography. 2nd ed. Undergraduate Texts in Mathematics. Springer-Verlag; 2014.
7. Mrabet N, Joye M, editors. Guide to Pairing-Based Cryptography. CRC Press; 2017.
8. Miller VS. Short Programs for functions on Curves; 1986. Available from: <http://crypto.stanford.edu/miller/>.
9. Miller VS. The Weil Pairing, and Its Efficient Calculation. *J Cryptology*. 2004; 17(4):235–261. <https://doi.org/10.1007/s00145-004-0315-8>
10. Hess F, Smart NP, Vercauteren F. The Eta Pairing Revisited. *IEEE Trans Information Theory*. 2006; 52(10):4595–4602. <https://doi.org/10.1109/TIT.2006.881709>
11. Matsuda S, Kanayama N, Hess F, Okamoto E. Optimised Versions of the Ate and Twisted Ate Pairings. *IEICE Transactions*. 2009; 92-A(7):1660–1667. <https://doi.org/10.1587/transfun.E92.A.1660>
12. Galbraith SD, Paterson KG, Smart NP. Pairings for cryptographers. *Discrete Applied Mathematics*. 2008; 156(16):3113–3121. <https://doi.org/10.1016/j.dam.2007.12.010>
13. Ogura N, Uchiyama S, Kanayama N, Okamoto E. A Note on the Pairing Computation Using Normalized Miller Functions. *IEICE Transactions*. 2012; 95-A(1):196–203. <https://doi.org/10.1587/transfun.E95.A.196>
14. Stange KE. The Tate Pairing Via Elliptic Nets. In: Takagi T, Okamoto T, Okamoto E, Okamoto T, editors. Pairing-Based Cryptography—Pairing 2007, First International Conference, Tokyo, Japan, July 2–4, 2007, Proceedings. vol. 4575 of Lecture Notes in Computer Science. Springer; 2007. p. 329–348. Available from: https://doi.org/10.1007/978-3-540-73489-5_19.
15. Onuki H, Teruya T, Kanayama N, Uchiyama S. The optimal ate pairing over the Barreto-Naehrig curve via parallelizing elliptic nets. *JSIAM Letters*. 2015; 8:9–12. <https://doi.org/10.14495/jsiaml.8.9>
16. Onuki H, Teruya T, Kanayama N, Uchiyama S. Faster Explicit Formulae for Computing Pairings via Elliptic Nets and Their Parallel Computation. In: Ogawa K, Yoshioka K, editors. Advances in Information and Computer Security—11th International Workshop on Security, IWSEC 2016, Tokyo, Japan, September 12–14, 2016, Proceedings. vol. 9836 of Lecture Notes in Computer Science. Springer; 2016. p. 319–334. Available from: https://doi.org/10.1007/978-3-319-44524-3_19.
17. Vercauteren F. Optimal pairings. *IEEE Trans Information Theory*. 2010; 56(1):455–461. <https://doi.org/10.1109/TIT.2009.2034881>
18. Hess F. Pairing Lattices. In: Galbraith SD, Paterson KG, editors. Pairing-Based Cryptography—Pairing 2008, Second International Conference, Egham, UK, September 1–3, 2008. Proceedings. vol. 5209 of Lecture Notes in Computer Science. Springer; 2008. p. 18–38. Available from: https://doi.org/10.1007/978-3-540-85538-5_2.
19. Galbraith SD, Scott M. Exponentiation in Pairing-Friendly Groups Using Homomorphisms. In: Galbraith SD, Paterson KG, editors. Pairing-Based Cryptography—Pairing 2008, Second International Conference, Egham, UK, September 1–3, 2008. Proceedings. vol. 5209 of Lecture Notes in Computer Science. Springer; 2008. p. 211–224. Available from: https://doi.org/10.1007/978-3-540-85538-5_15.
20. Sakemi Y, Nogami Y, Okeya K, Katou H, Morikawa Y. Skew Frobenius Map and Efficient Scalar Multiplication for Pairing-Based Cryptography. In: Franklin MK, Hui LCK, Wong DS, editors. Cryptology and Network Security, 7th International Conference, CANS 2008, Hong-Kong, China, December 2–4, 2008. Proceedings. vol. 5339 of Lecture Notes in Computer Science. Springer; 2008. p. 226–239. Available from: https://doi.org/10.1007/978-3-540-89641-8_16.
21. Kanayama N, Teruya T, Okamoto E. Scalar Multiplication on Pairing Friendly Elliptic Curves. *IEICE Transactions*. 2011; 94-A(6):1285–1292. <https://doi.org/10.1587/transfun.E94.A.1285>
22. Teruya T, Saito K, Kanayama N, Kawahara Y, Kobayashi T, Okamoto E. Constructing Symmetric Pairings over Supersingular Elliptic Curves with Embedding Degree Three. In: Cao Z, Zhang F, editors. Pairing-Based Cryptography—Pairing 2013—6th International Conference, Beijing, China, November

- 22-24, 2013, Revised Selected Papers. vol. 8365 of Lecture Notes in Computer Science. Springer; 2013. p. 97–112. Available from: https://doi.org/10.1007/978-3-319-04873-4_6.
23. Zhang X, Wang K. Fast Symmetric Pairing Revisited. In: Cao Z, Zhang F, editors. Pairing-Based Cryptography—Pairing 2013—6th International Conference, Beijing, China, November 22–24, 2013, Revised Selected Papers. vol. 8365 of Lecture Notes in Computer Science. Springer; 2013. p. 131–148. Available from: https://doi.org/10.1007/978-3-319-04873-4_8.
24. Chatterjee S, Menezes A, Rodríguez-Henríquez F. On Instantiating Pairing-Based Protocols with Elliptic Curves of Embedding Degree One. *IEEE Trans Computers*. 2017; 66(6):1061–1070. <https://doi.org/10.1109/TC.2016.2633340>
25. Freeman D, Scott M, Teske E. A Taxonomy of Pairing-Friendly Elliptic Curves. *J Cryptology*. 2010; 23(2):224–280. <https://doi.org/10.1007/s00145-009-9048-z>
26. Tibouchi M, Kim T. Improved elliptic curve hashing and point representation. *Des Codes Cryptography*. 2017; 82(1-2):161–177. <https://doi.org/10.1007/s10623-016-0288-2>
27. Beuchat J, González-Díaz JE, Mitsunari S, Okamoto E, Rodríguez-Henríquez F, Teruya T. High-Speed Software Implementation of the Optimal Ate Pairing over Barreto-Naehrig Curves. In: Joye M, Miyaji A, Otsuka A, editors. Pairing-Based Cryptography—Pairing 2010—4th International Conference, Yamana Hot Spring, Japan, December 2010. Proceedings. vol. 6487 of Lecture Notes in Computer Science. Springer; 2010. p. 21–39. Available from: https://doi.org/10.1007/978-3-642-17455-1_2.
28. Aranha DF, Karabina K, Longa P, Gebotys CH, López J. Faster Explicit Formulas for Computing Pairings over Ordinary Curves. In: Paterson KG, editor. Advances in Cryptology—EUROCRYPT 2011—30th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Tallinn, Estonia, May 15–19, 2011. Proceedings. vol. 6632 of Lecture Notes in Computer Science. Springer; 2011. p. 48–68. Available from: https://doi.org/10.1007/978-3-642-20465-4_5.
29. Aranha DF, Barreto PSLM, Longa P, Ricardini JE. The Realm of the Pairings. In: Lange T, Lauter KE, Lisonek P, editors. Selected Areas in Cryptography—SAC 2013—20th International Conference, Burnaby, BC, Canada, August 14–16, 2013, Revised Selected Papers. vol. 8282 of Lecture Notes in Computer Science. Springer; 2013. p. 3–25. Available from: https://doi.org/10.1007/978-3-662-43414-7_1.
30. Zavattoni E, Perez LJD, Mitsunari S, Sánchez-Ramírez AH, Teruya T, Rodríguez-Henríquez F. Software Implementation of an Attribute-Based Encryption Scheme. *IEEE Trans Computers*. 2015; 64(5):1429–1441. <https://doi.org/10.1109/TC.2014.2329681>
31. Ghosh S, Verbaauwhede I, Chowdhury DR. Core Based Architecture to Speed Up Optimal Ate Pairing on FPGA Platform. In: Abdalla M, Lange T, editors. Pairing-Based Cryptography—Pairing 2012—5th International Conference, Cologne, Germany, May 16–18, 2012, Revised Selected Papers. vol. 7708 of Lecture Notes in Computer Science. Springer; 2012. p. 141–159. Available from: https://doi.org/10.1007/978-3-642-36334-4_9.
32. Yao GX, Fan J, Cheung RCC, Verbaauwhede I. Faster Pairing Coprocessor Architecture. In: Abdalla M, Lange T, editors. Pairing-Based Cryptography—Pairing 2012—5th International Conference, Cologne, Germany, May 16–18, 2012, Revised Selected Papers. vol. 7708 of Lecture Notes in Computer Science. Springer; 2012. p. 160–176. Available from: https://doi.org/10.1007/978-3-642-36334-4_10.
33. Unterluggauer T, Wenger E. Efficient Pairings and ECC for Embedded Systems. In: Batina L, Robshaw M, editors. Cryptographic Hardware and Embedded Systems—CHES 2014—16th International Workshop, Busan, South Korea, September 23–26, 2014. Proceedings. vol. 8731 of Lecture Notes in Computer Science. Springer; 2014. p. 298–315. Available from: https://doi.org/10.1007/978-3-662-44709-3_17.
34. Fujimoto D, Nagahama Y, Matsumoto T. How to design hardware prime field multipliers for bilinear pairing. In: International SoC Design Conference, ISOCC 2016, Jeju, South Korea, October 23–26, 2016. IEEE; 2016. p. 203–204. Available from: <https://doi.org/10.1109/ISOCC.2016.7799858>.
35. Wang M, Dai G, Choo KK, Jayaraman P, Ranjan R. Constructing Pairing-Friendly Elliptic Curves under Embedding Degree 1 for Securing Critical Infrastructures. *PLOS ONE*. 2016; 11(8):1–13. <https://doi.org/10.1371/journal.pone.0161857>
36. Bertoni GM, Chen L, Fragneto P, Harrison KA, Pelosi G. Computing Tate Pairing on Smartcards; 2005. https://web.archive.org/web/20060316075737/http://www.st.com/stonline/products/families/smartcard/ches2005_v4.pdf (Accessed 2017/06/25).
37. Bertoni G, Breveglieri L, Chen L, Fragneto P, Harrison KA, Pelosi G. A pairing SW implementation for Smart-Cards. *Journal of Systems and Software*. 2008; 81(7):1240–1247. <https://doi.org/10.1016/j.jss.2007.09.022>
38. Wang M, Dai G, Choo KK, Jayaraman P, Ranjan R. Pairing-friendly elliptic curves under embedding degree 1 with 160 bits;. Available from: <https://doi.org/10.1371/journal.pone.0161857.s001>.
39. Wang M, Dai G, Choo KK, Jayaraman P, Ranjan R. Pairing-friendly elliptic curves under embedding degree 1 with 190 bits;. Available from: <https://doi.org/10.1371/journal.pone.0161857.s002>.

40. Lenstra AK, Verheul ER. Selecting Cryptographic Key Sizes. *J Cryptology*. 2001; 14(4):255–293. <https://doi.org/10.1007/s00145-001-0009-4>
41. Bernstein DJ, Engels S, Lange T, Niederhagen R, Paar C, Schwabe P, et al. Faster elliptic-curve discrete logarithms on FPGAs; 2016. Cryptology ePrint Archive, Report 2016/382. Available from: <https://eprint.iacr.org/2016/382>.
42. Balasubramanian R, Koblitz N. The Improbability That an Elliptic Curve Has Subexponential Discrete Log Problem under the Menezes—Okamoto—Vanstone Algorithm. *J Cryptology*. 1998; 11(2):141–145. <https://doi.org/10.1007/s001459900040>
43. Menezes A, Okamoto T, Vanstone SA. Reducing elliptic curve logarithms to logarithms in a finite field. *IEEE Trans Information Theory*. 1993; 39(5):1639–1646. <https://doi.org/10.1109/18.259647>
44. Frey G, Rück HG. A Remark Concerning m -divisibility and the Discrete Logarithm in the Divisor Class Group of Curves. *Mathematics of Computation*. 1994; 62(206):865–874. <https://doi.org/10.1090/S0025-5718-1994-1218343-6>
45. Menezes A, Sarkar P, Singh S. Challenges with Assessing the Impact of NFS Advances on the Security of Pairing-based Cryptography; 2016. Cryptology ePrint Archive, Report 2016/1102. Available from: <http://eprint.iacr.org/2016/1102>.
46. Barbulescu R, Duquesne S. Updating key size estimations for pairings; 2017. Cryptology ePrint Archive, Report 2017/334. Available from: <http://eprint.iacr.org/2017/334>.
47. Hayashi T, Shimoyama T, Shinohara N, Takagi T. Breaking Pairing-Based Cryptosystems Using η T Pairing over $\text{GF}(3^6)$. In: Wang X, Sako K, editors. *Advances in Cryptology—ASIACRYPT 2012—18th International Conference on the Theory and Application of Cryptology and Information Security*, Beijing, China, December 2–6, 2012. Proceedings. vol. 7658 of *Lecture Notes in Computer Science*. Springer; 2012. p. 43–60. Available from: https://doi.org/10.1007/978-3-642-34961-4_5.
48. Hayashi T, Shinohara N, Wang L, Matsuo S, Shirase M, Takagi T. Solving a 676-Bit Discrete Logarithm Problem in $\text{GF}(3^6)$. *IEICE Transactions*. 2012; 95-A(1):204–212. <https://doi.org/10.1587/transfun.E95.A.204>
49. Barbulescu R, Gaudry P, Guillevic A, Morain F. Improving NFS for the Discrete Logarithm Problem in Non-prime Finite Fields. In: Oswald E, Fischlin M, editors. *Advances in Cryptology—EUROCRYPT 2015—34th Annual International Conference on the Theory and Applications of Cryptographic Techniques*, Sofia, Bulgaria, April 26–30, 2015, Proceedings, Part I. vol. 9056 of *Lecture Notes in Computer Science*. Springer; 2015. p. 129–155. Available from: https://doi.org/10.1007/978-3-662-46800-5_6.
50. Kleinjung T, Diem C, Lenstra AK, Priplata C, Stahlke C. Computation of a 768-Bit Prime Field Discrete Logarithm. In: Coron J, Nielsen JB, editors. *Advances in Cryptology—EUROCRYPT 2017—36th Annual International Conference on the Theory and Applications of Cryptographic Techniques*, Paris, France, April 30—May 4, 2017, Proceedings, Part I. vol. 10210 of *Lecture Notes in Computer Science*; 2017. p. 185–201. Available from: https://doi.org/10.1007/978-3-319-56620-7_7.
51. The Sage Developers. SageMath, the Sage Mathematics Software System; 2017. Available from: <http://www.sagemath.org>.
52. The CADO-NFS Development Team. CADO-NFS, An Implementation of the Number Field Sieve Algorithm; 2017. Available from: <http://cado-nfs.gforge.inria.fr/>.
53. Grémy L, Guillevic A. DiscreteLogDB, a database of computations of discrete logarithms; 2017. <https://gitlab.inria.fr/dldb/discretelogdb> (Accessed 2017/07/24).
54. Adrian D, Bhargavan K, Durumeric Z, Gaudry P, Green M, Halderman JA, et al. Imperfect Forward Secrecy: How Diffie-Hellman Fails in Practice. In: Ray I, Li N, Kruegel C, editors. *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, Denver, CO, USA, October 12–6, 2015. ACM; 2015. p. 5–17. Available from: <http://doi.acm.org/10.1145/2810103.2813707>.
55. Fried J, Gaudry P, Heninger N, Thomé E. A Kilobit Hidden SNFS Discrete Logarithm Computation. In: Coron J, Nielsen JB, editors. *Advances in Cryptology—EUROCRYPT 2017—36th Annual International Conference on the Theory and Applications of Cryptographic Techniques*, Paris, France, April 30—May 4, 2017, Proceedings, Part I. vol. 10210 of *Lecture Notes in Computer Science*; 2017. p. 202–231. Available from: https://doi.org/10.1007/978-3-319-56620-7_8.