Network Working Group
Internet-Draft
Intended status: Experimental
Expires: August 1, 2019

S. Yonezawa
Lepidum
S. Chikara
NTT TechnoCross
T. Kobayashi
T. Saito
NTT

January 28, 2019

Pairing-Friendly Curves
draft-yonezawa-pairing-friendly-curves-00

Abstract

This memo introduces pairing-friendly curves used for constructing pairing-based cryptography. It describes recommended parameters for each security level and recent implementations of pairing-friendly curves.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on August 1, 2019.

Table of Contents

1. Introduction

1.1. Pairing-Based Cryptography

Elliptic curve cryptography is one of the important areas in recent cryptography. The cryptographic algorithms based on elliptic curve cryptography, such as ECDSA, is widely used in many applications.

Pairing-based cryptography, a variant of elliptic curve cryptography, is attracted the attention for its flexible and applicable functionality. Pairing is a special map defined over elliptic curves. Generally, elliptic curves is defined so that pairing is not efficiently computable since elliptic curve cryptography is broken if pairing is efficiently computable. As the importance of pairing

grows, elliptic curves where pairing is efficiently computable are studied and the special curves called pairing-friendly curves are proposed.

Thanks to the characteristics of pairing, it can be applied to construct several cryptographic algorithms and protocols such as identity-based encryption (IBE), attribute-based encryption (ABE), authenticated key exchange (AKE), short signatures and so on. Several applications of pairing-based cryptography is now in practical use.

1.2. Applications of Pairing-Based Cryptography

Several applications using pairing-based cryptography are standardized and implemented. We show example applications available in the real world.

IETF issues RFCs for pairing-based cryptography such as identity-based cryptography [9], certificateless signatures [10], Sakai-Kasahara Key Encryption (SAKKE) [11], and Identity-Based Authenticated Key Exchange (IBAKE) [12]. SAKKE is applied to Multimedia Internet KEYing (MIKEY) [13] and used in 3GPP [14].

Pairing-based key agreement protocols are standardized in ISO/IEC [15]. In [15], a key agreement scheme by Joux [16], identity-based key agreement schemes by Smart-Chen-Cheng [17] and by Fujioka-Suzuki-Ustaoglu [18] are specified.

MIRACL implements M-Pin, a multi-factor authentication protocol [19]. M-Pin protocol includes a kind of zero-knowledge proof, where pairing is used for its construction.

Trusted Computing Group (TCG) specifies ECDAA (Elliptic Curve Direct Anonymous Attestation) in the specification of Trusted Platform Module (TPM) [20]. ECDAA is a protocol for proving the attestation held by a TPM to a verifier without revealing the attestation held by that TPM. Pairing is used for constructing ECDAA. FIDO Alliance [21] and W3C [22] also published ECDAA algorithm similar to TCG.

Zcash implements their own zero-knowledge proof algorithm named zk-SNARKs (Zero-Knowledge Succinct Non-Interactive Argument of Knowledge) [23]. zk-SNARKs is used for protecting privacy of transactions of Zcash. They use pairing for constructing zk-SNARKS.

Cloudflare introduced Geo Key Manager [24] to restrict distribution of customers' private keys to the subset of their data centers. To achieve this functionality, attribute-based encryption is used and pairing takes a role as a building block.

DFINITY utilized threshold signature scheme to generate the decentralized random beacons [25]. They constructed a BLS signature-based scheme, which is based on pairings.

In Ethereum 2.0, project Prysm applies signature aggregation for scalability benefits by leveraging DFINITY's random-beacon chain playground [26]. Their codes are published on GitHub.

1.3. Goal

The goal of this memo is to consider the security of pairing-friendly curves used in pairing-based cryptography and introduce secure parameters of pairing-frindly curves. Specifically, we explain the recent attack against pairing-friendly curves and how much the security of the curves is reduced. We show how to evaluate the security of pairing-friendly curves and give the parameters for 100 bits of security, which is no longer secure, 128 and 256 bits of security.

1.4. Requirements Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [1].

2. Preliminaries

2.1. Elliptic Curve

Let p > 3 be a prime and F_p be a finite field. The curve defined by the following equation E is called an elliptic curve.

$$E : y^2 = x^3 + A * x + B,$$

where A, B are in F_p and satisfies $4 * A^3 + 27 * B^2 != 0 \bmod p$.

Solutions (x, y) for an elliptic curve E, as well as the point at infinity, O_E, are called F_p-rational points. If P and Q are two points on the curve E, we can define R = P + Q as the opposite point of the intersection between the curve E and the line that intersects P and Q. We can define P + O_E = P = O_E + P as well. The additive group is constructed by the well-defined operation in the set of F_p-rational points. Similarly, a scalar multiplication S = [a]P for a positive integer a can be defined as an a-time addition of P.

Typically, the cyclic additive group with a prime order r and the base point G in its group is used for the elliptic curve

cryptography. Furthermore, we define terminology used in this memo as follows.

O_E: the point at infinity over an elliptic curve E.

#E(F_p): number of points on an elliptic curve E over F_p.

h: a cofactor such that h = #E(F_p)/r.

k: an embedding degree, a minimum integer such that r is a divisor of p^k - 1.

2.2. Pairing

Pairing is a kind of the bilinear map defined over an elliptic curve. Examples include Weil pairing, Tate pairing, optimal Ate pairing [2] and so on. Especially, optimal Ate pairing is considered to be efficient to compute and mainly used for practical implementation.

Let E be an elliptic curve defined over the prime field F_p. Let G_1 be a cyclic subgroup generated by a rational point on E with order r, and G_2 be a cyclic subgroup generated by a twisted curve E' of E with order r. Let G_T be an order r subgroup of a field F_p^k, where k is an embedded degree. Pairing is defined as a bilinear map e: (G_1, G_2) -> G_T satisfying the following properties:

(1) Bilinearity: for any S in G_1, T in G_2, a, b in Z_r, we have the relation e([a]S, [b]T) = e(S, T)^{a * b}.

(2) Non-degeneracy: for any T in G_2, e(S, T) = 1 if and only if S = O_E. Similarly, for any S in G_1, e(S, T) = 1 if and only if T = O_E.

(3) Computability: for any S in G_1 and T in G_2, the bilinear map is efficiently computable.

2.3. Barreto-Naehrig Curve

A BN curve [3] is one of the instantiations of pairing-friendly curves proposed in 2005. A pairing over BN curves constructs optimal Ate pairings.

A BN curve is an elliptic curve E defined over a finite field F_p, where p is more than or equal to 5, such that p and its order r are prime numbers parameterized by

p = 36u^4 + 36u^3 + 24u^2 + 6u + 1

$$r = 36u^4 + 36u^3 + 18u^2 + 6u + 1$$

for some well chosen integer u. The elliptic curve has an equation of the form E: y^2 = x^3 + b, where b is an element of multiplicative group of order p.

BN curves always have order 6 twists. If w is an element which is neither a square nor a cube in a finite field F_p^2, the twisted curve E' of E is defined over a finite field F_p^2 by the equation E': y^2 = x^3 + b' with b' = b/w or b' = bw.

A pairing e is defined by taking G_1 as a cyclic group composed by rational points on the elliptic curve E, G_2 as a cyclic group composed by rational points on the elliptic curve E', and G_T as a multiplicative group of order p^12.

2.4. Barreto-Lynn-Scott Curve

A BLS curve [4] is another instantiations of pairings proposed in 2002. Similar to BN curves, a pairing over BLS curves constructs optimal Ate pairings.

A BLS curve is an elliptic curve E defined over a finite field F_p by an equation of the form E: y^2 = x^3 + b and has a twist of order 6 defined in the same way as BN curves. In contrast to BN curves, a BLS curve does not have a prime order but its order is divisible by a large parameterized prime r and the pairing will be defined on the r-torsions points.

BLS curves vary according to different embedding degrees. In this memo, we deal with BLS12 and BLS48 families with embedding degrees 12 and 48 with respect to r, respectively.

In BLS curves, parameterized p and r are given by the following equations:

BLS12:

$$p = (u - 1)^2 (u^4 - u^2 + 1)/3 + u$$

$$r = u^4 - u^2 + 1$$

BLS48:

$$p = (u - 1)^2 (u^{16} - u^8 + 1)/3 + u$$

$$r = u^{16} - u^8 + 1$$

for some well chosen integer u.

3. Security of Pairing-Friendly Curves

3.1. Evaluating the Security of Pairing-Friendly Curves

The security of pairing-friendly curves is evaluated by the hardness of the following discrete logarithm problems.

- The elliptic curve discrete logarithm problem (ECDLP) in G_1 and G_2
- The finite field discrete logarithm problem (FFDLP) in G_T

There are other hard problems over pairing-friendly curves, which are used for proving the security of pairing-based cryptography. Such problems include bilinear computational Diffie-Hellman (BCDH) problem, bilinear decisional Diffie-Hellman (BDDH) problem, gap BDDH problem, etc [27]. Almost all of these variants are reduced to the hardness of discrete logarithm problems described above and believed to be easier than the discrete logarithm problems.

There would be the case where the attacker solves these reduced problems to break the pairing-based cryptography. Since such attacks have not been discovered yet, we discuss the hardness of the discrete logarithm problems in this memo.

The security level of pairing-friendly curves is estimated by the computational cost of the most efficient algorithm to solve the above discrete logarithm problems. The well-known algorithms for solving the discrete logarithm problems includes Pollard's rho algorithm [28], Index Calculus [29] and so on. In order to make index calculus algorithms more efficient, number field sieve (NFS) algorithms are utilized.

In addition, the special case where the cofactors of G_1, G_2 and G_T are small should be taken care [30]. In such case, the discrete logarithm problem can be efficiently solved. One has to choose parameters so that the cofactors of G_1, G_2 and G_T contain no prime factors smaller than |G_1|, |G_2| and |G_T|.

3.2. Impact of the Recent Attack

In 2016, Kim and Barbulescu proposed a new variant of the NFS algorithms, the extended number field sieve (exTNFS), which drastically reduces the complexity of solving FFDLP [5]. Due to exTNFS, the security level of pairing-friendly curves asymptotically dropped down. For instance, Barbulescu and Duquesne estimates that

the security of the BN curves which was believed to provide 128 bits of security (BN256, for example) dropped down to approximately 100 bits [6].

Some papers show the minimum bitlength of the parameters of pairing-friendly curves for each security level when applying exTNFS as an attacking method for FFDLP. For 128 bits of security, Menezes, Sarkar and Singh estimated the minimum bitlength of p of BN curves after exTNFS as 383 bits, and that of BLS12 curves as 384 bits [7]. For 256 bits of security, Kiyomura et al. estimated the minimum bitlength of p^k of BLS48 curves as 27,410 bits, which implied 572 bits of p [8].

4. Security Evaluation of Pairing-Friendly Curves

We give security evaluation for pairing-friendly curves based on the evaluating method presented in Section 3. We also introduce secure parameters of pairing-friendly curves for each security level. The parameters introduced here are chosen with the consideration of security, efficiency and global acceptance.

For security, we introduce 100 bits, 128 bits and 256 bits of security. We note that 100 bits of security is no longer secure and recommend 128 bits and 256 bits of security for secure applications. We follow TLS 1.3 which specifies the cipher suites with 128 bits and 256 bits of security as mandatory-to-implement for the choice of the security level.

Implementors of the applications have to choose the parameters with appropriate security level according to the security requirements of the applications. For efficiency, we refer to the benchmark by mcl [31] for 128 bits of security, and by Kiyomura et al. [8] for 256 bits of security and choose sufficiently efficient parameters. For global acceptance, we give the implementations of pairing-friendly curves in section Section 5.

4.1. For 100 Bits of Security

Before exTNFS, BN curves with 256-bit size of underlying finite field (so-called BN256) were considered to have 128 bits of security. After exTNFS, however, the security level of BN curves with 256-bit size of underlying finite field fell into 100 bits.

Implementors who will newly develop the applications of pairing-based cryptography SHOULD NOT use BN256 as a pairing-friendly curve when their applications require 128 bits of security. In case an application does not require higher security level and is sufficient to have 100 bits of security (i.e. IoT), implementors MAY use BN256.

4.2. For 128 Bits of Security

A BN curve with 128 bits of security is shown in [6], which we call BN462. BN462 is defined by a parameter u = 2^114 + 2^101 - 2^14 - 1 for the definition in Section 2.3. Defined by u, the elliptic curve E and its twisted curve E' are represented by E: y^2 = x^3 - 4 and E': y^2 = x^3 - 4 * (1 + i), where i is an element of an extension field F_p^2, respectively. The size of p becomes 462-bit length.

A BLS12 curve with 128 bits of security shown in [6] is parameterized by u = -2^77 - 2^71 - 2^64 + 2^37 + 2^35 + 2^22 - 2^5, which we call BLS12-461. Defined by u, the elliptic curve E and its twisted curve E' are represented by E: y^2 = x^3 - 2 and E': y^2 = x^3 - 2 / (1 + i), respectively. The size of p becomes 461-bit length. The curve BLS12-461 is subgroup-secure.

There is another BLS12 curve stating 128 bits of security, BLS12-381 [32]. It is defined by a parameter u = -0xd201000000010000. Defined by u, the elliptic curve E and its twisted curve E' are represented by E: y^2 = x^3 + 4 and E': y^2 = x^3 + 4(i + 1), respectively.

We have to note that, according to [7], the bit length of p for BLS12 to achieve 128 bits of security is calculated as 384 bits and more, which BLS12-381 does not satisfy. Although the computational time is conservatively estimated by 2^110 when exTNFS is applied with index calculus, there is no currently published efficient method for such computational time. They state that BLS12-381 achieves 127-bit security level evaluated by the computational cost of Pollard's rho.

4.3. For 256 Bits of Security

As shown in Section 3.2, it is unrealistic to achieve 256 bits of security by BN curves since the minimum size of p becomes too large to implement. Hence, we consider BLS48 for 256 bits of security.

A BLS48 curve with 256 bits of security is shown in [8], which we call BLS48-581. It is defined by a parameter u = -1 + 2^7 - 2^10 - 2^30 - 2^32 and the elliptic curve E and its twisted curve E' are represented by E: y^2 = x^3 + 1 and E': y^2 = x^3 - 1/w, where w is an element of an extension field F_p^8. The size of p becomes 581-bit length.

5. Implementations of Pairing-Friendly Curves

We show the pairing-friendly curves selected by existing standards, applications and cryptographic libraries.

ISO/IEC 15946-5 [33] shows examples of BN curves with the size of 160, 192, 224, 256, 384 and 512 bits of p. There is no action so far after the proposal of exTNFS.

TCG adopts an BN curve of 256 bits specified in ISO/IEC 15946-5 (TPM_ECC_BN_P256) and of 638 bits specified by their own (TPM_ECC_BN_P638). FIDO Alliance [21] and W3C [22] adopt the BN curves specified in TCG, a 512-bit BN curve shown in ISO/IEC 15946-5 and another 256-bit BN curve.

MIRACL [34] implements BN curves and BLS12 curves.

Zcash implemented a BN curve (named BN128) in their library libsnark [35]. After exTNFS, they propose a new parameter of BLS12 as BLS12-381 [32] and publish its experimental implementation [36].

Cloudflare implements a 256-bit BN curve (bn256) [37]. There is no action so far after exTNFS.

Ethereum 2.0 adopts BLS12-381 (BLS12_381), BN curves with 254 bits of p (CurveFp254BNb) and 382 bits of p (CurveFp382_1 and CurveFp382_2) [38]. Their implementation calls mcl [31] for pairing computation.

Cryptographic libraries which implement pairings include PBC [39], mcl [31], RELIC [40], TEPLA [41], AMCL [42], Intel IPP [43] and a library by Kyushu University [44].

Table 1 shows the adoption of pairing-friendly curves in existing standards, applications and libraries.

Category	Name	100 bit	128 bit	256 bit
standards	ISO/IEC [33]	BN256	BN384	
	TCG	BN256		
	FIDO/W3C	BN256		
applications	MIRACL	BN254	BLS12	
	Zcash	BN128 (CurveSNARK)	BLS12-381	
	Cloudflare	BN256		

	Ethereum	BN254	BN382 (*) / BLS12-381 (*)	
libraries	PBC	BN254 / BN_SNARK1	BN381_1 (*) / BN462 / BLS12-381	
	mcl	BN254 / BN_SNARK1	BN381_1 (*) / BN462 / BLS12-381	
	RELIC [40]	BN254 / BN256	BLS12-381 / BLS12-455	
	TEPLA	BN254		
	AMCL	BN254 / BN256	BLS12-381 (*) / BLS12-383 (*) / BLS12-461	BLS48
	Intel IPP	BN256		
	Kyushu Univ.			BLS48

(*) There is no research result on the security evaluation, but the implementers states that they satisfy 128 bits of security.

Table 1: Adoption of Pairing-Friendly Curves

6. Security Considerations

This memo entirely describes the security of pairing-friendly curves, and introduces secure parameters of pairing-friendly curves. We give these parameters in terms of security, efficiency and global acceptance. The parameters for 100, 128 and 256 bits of security are introduced since the security level will different in the requirements of the pairing-based applications.

7. IANA Considerations

This document has no actions for IANA.

8. Acknowledgements

The authors would like to thank Akihiro Kato for his significant contribution to the early version of this memo.

9. Change log

NOTE TO RFC EDITOR: Please remove this section in before final RFC publication.

10. References

10.1. Normative References

[1] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", RFC 2119, March 1997.

[2] Vercauteren, F., "Optimal pairings", Proceedings IEEE Transactions on Information Theory 56(1): 455-461 (2010), 2010.

[3] Barreto, P. and M. Naehrig, "Pairing-Friendly Elliptic Curves of Prime Order", Selected Areas in Cryptography-SAC 2005. volume 3897 of Lecture Notes in Computer Science, pages 319-331, 2006.

[4] Barreto, P., Lynn, B., and M. Scott, "Constructing Elliptic Curves with Prescribed Embedding Degrees", Security in Communication Networks - SCN 2002 LNCS 2576, pp. 257--167, Springer, 2002.

[5] Kim, T. and R. Barbulescu, "Extended tower number field sieve: a new complexity for the medium prime case.", CRYPTO 2016 LNCS, vol. 9814, pp. 543.571, 2016.

[6] Barbulescu, R. and S. Duquesne, "Updating Key Size Estimations for Pairings", Journal of Cryptology 2018, January 2018.

[7] Menezes, A., Sarkar, P., and S. Singh, "Challenges with Assessing the Impact of NFS Advances on the Security of Pairing-Based Cryptography", Paradigms in Cryptology - Mycrypt 2016 LNCS 10311, pp. 83--108, Springer, 2017.

[8] Kiyomura, Y., Inoue, A., Kawahara, Y., Yasuda, M., Takagi, T., and T. Kobayashi, "Secure and Efficient Pairing at 256-Bit Security Level", ACNS 2017 LNCS, vol. 10355, pp. 59.79, 2017, 2017.

10.2. Informative References

[9] Boyen, X. and L. Martin, "Identity-Based Cryptography Standard (IBCS) #1: Supersingular Curve Implementations of the BF and BB1 Cryptosystems", RFC 5091, DOI 10.17487/RFC5091, December 2007, <https://www.rfc-editor.org/info/rfc5091>.

[10] Groves, M., "Elliptic Curve-Based Certificateless Signatures for Identity-Based Encryption (ECCSI)", RFC 6507, DOI 10.17487/RFC6507, February 2012, <https://www.rfc-editor.org/info/rfc6507>.

[11] Groves, M., "Sakai-Kasahara Key Encryption (SAKKE)", RFC 6508, DOI 10.17487/RFC6508, February 2012, <https://www.rfc-editor.org/info/rfc6508>.

[12] Cakulev, V., Sundaram, G., and I. Broustis, "IBAKE: Identity-Based Authenticated Key Exchange", RFC 6267, DOI 10.17487/RFC6267, March 2012, <https://www.rfc-editor.org/info/rfc6539>.

[13] Groves, M., "MIKEY-SAKKE: Sakai-Kasahara Key Encryption in Multimedia Internet KEYing (MIKEY)", RFC 6509, DOI 10.17487/RFC6509, February 2012, <https://www.rfc-editor.org/info/rfc6509>.

[14] 3GPP, "Security of the mission critical service (Release 15)", 3GPP TS 33.180 15.3.0, September 2018.

[15] ISO/IEC, "ISO/IEC 11770-3:2015", ISO/IEC Information technology -- Security techniques -- Key management -- Part 3: Mechanisms using asymmetric techniques, 2015.

[16] Joux, A., "A One Round Protocol for Tripartite Diffie-Hellman", ANTS-IV LNCS 1838, pp. 385--393, Springer-Verlag, 2000.

[17] Chen, L., Cheng, Z., and N. Smart, "Indentity-based Key Agreement Protocols From Pairings", International Journal of Information Security Volume 6 Issue 4, pages 213--241, Springer-Verlag, June 2007.

[18] Fujioka, A., Suzuki, K., and B. Ustaoglu, "Ephemeral Key Leakage Resilient and Efficient ID-AKEs That Can Share Identities, Private and Master Keys", Pairing-Based Cryptography - Pairing 2010 LNCS 6487, pp. 187--205, Springer, 2010.

[19] Scott, M., "M-Pin: A Multi-Factor Zero Knowledge Authentication Protocol", <https://www.miracl.com/miracl-labs/m-pin-a-multi-factor-zero-knowledge-authentication-protocol>.

[20] Trusted Computing Group (TCG), "TPM 2.0 Library Specification", September 2016, <https://trustedcomputinggroup.org/resource/tpm-library-specification/>.

[21] Lindemann, R., "FIDO ECDAA Algorithm - FIDO Alliance Review Draft 02", July 2018, <https://fidoalliance.org/specs/fido-v2.0-rd-20180702/fido-ecdaa-algorithm-v2.0-rd-20180702.html>.

[22] Balfanz, D., Czeskis, A., Hodges, J., Jones, J., Jones, M., Kumar, A., Liao, A., Lindemann, R., and E. Lundberg, "Web Authentication: An API for accessing Public Key Credentials Level 1 - W3C Candidate Recommendation", July 2018, <https://www.w3.org/TR/webauthn/>.

[23] Lindemann, R., "What are zk-SNARKs?", July 2018, <https://z.cash/technology/zksnarks.html>.

[24] Sullivan, N., "Geo Key Manager: How It Works", September 2017, <https://blog.cloudflare.com/geo-key-manager-how-it-works/>.

[25] Hanke, T., Movahedi, M., and D. Williams, "DFINITY Technology Overview Series Consensus System Rev. 1", <https://dfinity.org/pdf-viewer/library/dfinity-consensus.pdf>.

[26] Jordan, R., "Ethereum 2.0 Development Update #17 - Prysmatic Labs", November 2018, <https://medium.com/prysmatic-labs/ethereum-2-0-development-update-17-prysmatic-labs-ed5bcf82ec00>.

[27] ECRYPT, "Final Report on Main Computational Assumptions in Cryptography", January 2013.

[28] Pollard, J., "Monte Carlo Methods for Index Computation (mod p)", Proceedings Mathematics of Computation, Vol.32, 1978.

[29] Hellman, M. and J. Reyneri, "Fast computation of discrete logarithms in GF(q)", Advances in Cryptology: Proceedings of CRYPTO '82 pp. 3-13, 1983.

[30] Barreto, P., Costello, C., Misoczki, R., Naehrig, M., Pereira, G., and G. Zanon, "Subgroup security in pairing-based cryptography", Cryptology ePrint Archive http://eprint.iacr.org/2015/247.pdf, 2015.

[31] Mitsunari, S., "mcl - A portable and fast pairing-based cryptography library", 2016, <https://github.com/herumi/mcl>.

[32] Bowe, S., "BLS12-381: New zk-SNARK Elliptic Curve Construction", March 2017, <https://blog.z.cash/new-snark-curve/>.

[33] ISO/IEC, "ISO/IEC 15946-5:2017", ISO/IEC Information technology -- Security techniques -- Cryptographic techniques based on elliptic curves -- Part 5: Elliptic curve generation, 2017.

[34] MIRACL Ltd., "MIRACL Cryptographic SDK", 2018, <https://github.com/miracl/MIRACL>.

[35] SCIPR Lab, "libsnark: a C++ library for zkSNARK proofs", 2012, <https://github.com/zcash/libsnark>.

[36] zkcrypto, "zkcrypto - Pairing-friendly elliptic curve library", 2017, <https://github.com/zkcrypto/pairing>.

[37] Cloudflare, "package bn256", <https://godoc.org/github.com/cloudflare/bn256>.

[38] Prysmatic Labs, "go-bls - Go wrapper for a BLS12-381 Signature Aggregation implementation in C++", 2018, <https://godoc.org/github.com/prysmaticlabs/go-bls>.

[39] Lynn, B., "PBC Library - The Pairing-Based Cryptography Library", 2006, <https://crypto.stanford.edu/pbc/>.

[40] Aranha, D. and C. Gouv, "RELIC is an Efficient LIbrary for Cryptography", 2013, <https://code.google.com/p/relic-toolkit/>.

[41] University of Tsukuba, "TEPLA: University of Tsukuba Elliptic Curve and Pairing Library", 2013, <http://www.cipher.risk.tsukuba.ac.jp/tepla/index_e.html>.

[42] The Apache Software Foundation, "The Apache Milagro Cryptographic Library (AMCL)", 2016, <https://github.com/apache/incubator-milagro-crypto>.

[43] Intel Corporation, "Developer Reference for Intel Integrated Performance Primitives Cryptography 2019", 2018, <https://software.intel.com/en-us/ipp-crypto-reference-arithmetic-of-the-group-of-elliptic-curve-points>.

[44] Kyushu University, "bls48 - C++ library for Optimal Ate Pairing on BLS48", 2017, <https://github.com/mk-math-kyushu/bls48>.

Appendix A. Test Vectors of Optimal Ate Pairing

(TBD)

Authors' Addresses

Shoko Yonezawa
Lepidum

EMail: yonezawa@lepidum.co.jp

Sakae Chikara
NTT TechnoCross

EMail: chikara.sakae@po.ntt-tx.co.jp

Tetsutaro Kobayashi
NTT

EMail: kobayashi.tetsutaro@lab.ntt.co.jp

Tsunekazu Saito
NTT

EMail: saito.tsunekazu@lab.ntt.co.jp