Exploring Pairing Based Cryptography*

Stéphane Vincent

Sikoba Research

December 2018

Abstract

One of the key cryptographic primitives behind various constructions, including privacy-preserving authentication [1], short signature schemes [2], and zero-knowledge proofs [3] is the bilinear mapping technique that uses pairings over elliptic curves. Initially used in cryptography to break the discrete logarithm problem in the group of points of some elliptic curves [4], pairings are now considered to be one of the most suitable mathematical tools to design secure and efficient cryptographic protocols.

This paper provides an introduction to Pairing Based Cryptography (PBC), explains which elliptic curves are suitable for PBC and discusses existing PBC libraries. An extended list of references is also provided.

1 Introduction

The cryptography relying on pairings is known under the generic term of "pairing-based cryptography" (or PBC for short). PBC has been studied extensively [5], because it has many beautiful and elegant properties and one powerful argument that can support the use of PBC is verifiable computation. A client can outsource a complicated task to a server in the cloud and get back the results. To convince the client that the computation is correct the server may include a non-interactive argument of correctness with the result.

The authors in [6] found an insightful construction with a common reference string for proving arithmetic circuit satisfiability. Useful work on implementation has followed the above theoretical advances, e.g. in virtual currencies such as Pinocchio coin [7] and Zcash [8]. However, the benefit of using PBC, compared to traditional constructions, comes with a price: for most of the people involved in implementation, it is difficult to understand the pairing computation and there are few industrial products being integrated with pairing-based cryptosystems.

The paper is organized as follows: bilinear pairings are introduced in section 2. Relevant properties of pairing-friendly elliptic curves are reviewed in section 3. Section 4 presents several PBC libraries. This is followed by concluding remarks in section 5.

*Research supported by Fantom Foundation

Table 1 shows a list of elliptic curve (EC) and PBC cryptrographic primitives. A checkmark ✓denotes that the cryptographic primitive can be implemented, while an × denotes that it cannot.

	Standard EC Crypto	PBC
Multisignature	✓	✓
Threshold Signature	✓	✓
Aggregate Signature	×	✓
ZKP in Log. Setting	✓	✓
ZKP in Pairing Setting	×	✓
Partially Homomorphic Encryption*	✓	✓
Somewhat Homomorphic Encryption †	×	✓

Table 1: A non-exhaustive survey of EC and PBC cryptographic primitives.

2 PBC In a Nutshell

A pairing is a function that takes a pair of two points on an elliptic curve and outputs an element in a finite field. The pairings we consider are also bilinear. This means that the pairing map preserves the additive structure of the elliptic curve, and carries it over into the finite field. For example,

$$e(P_1 + P_2,\ Q) = e(P_1,\ Q) \cdot e(P_2,\ Q)$$

Where the $\cdot$ symbol denotes multiplication in the finite field. P_1, P_2 and Q are elements of the additive groups of elliptic curves defined over finite fields. More precisely,

$$e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_3$$

Where the groups $\mathbb{G}_1$ and $\mathbb{G}_2$ are subgroups or quotient groups of an elliptic curve defined over a finite field $\mathbb{F}_P$ or one of its extensions and $\mathbb{G}_3$ is a subgroup or a quotient group of $\mathbb{F}^*_{p^k}$ where k is called the embedding degree. Therefore, PBC is working with elements that are defined over the finite base field $\mathbb{F}_p$ (its parameters have size $O(\log p)$ bits) and elements defined over the extension field $\mathbb{F}_{p^k}$ (its parameters have size $O(k \log p)$ bits).

A suitable pairing for cryptographic applications requires that the discrete logarithm problem is sufficiently difficult on these three groups. The security of pairings defined over $\mathbb{F}_p$ having embedding degree k and group order r is determined by:

1. The cost of the discrete logarithm problem (DLP) on an order r subgroup or quotient group of an elliptic curve defined over $\mathbb{F}_p$ (the curve side);
2. The cost of DLP in a quotient of the multiplicative group $\mathbb{F}_{p^k}$ (the finite field side).

The security evaluation on the curve is simple: if s is the desired level of security, we select r such that $log_2 r \geq 2s$ because of Pollard's rho algorithm [9] (and by consequence, $\log_2 p \geq 2s$). Table 2 shows the parameter size for recommended security level adapted from [10].

To actually implement any pairing-based cryptogrpahic protocol, it is necessary to also choose a specific pairing function e. The two most commonly used pairings are the Well and Tate pairings,

*Partially Homomorphic Encryption allows only one type of operation with an unlimited number of times (i.e. no bound on the number of usages.

†Somewhat Homorphic Encryption (SWHE) allows some types of operations with a limites number of times

Security level s (bits)	Group order r (bits)	p^k (bits)
80	160	960-1280
128	256	3000-5000
256	512	14000-18000

Table 2: Recommended security levels. As with secp256k1, the security level is 128 bits.

but researchers have discovered several new pairings. However, we note that various pairings are not interchangeable; the Eta pairing can only be defined for supersingular curves. Hence, the choice of pairing (and elliptic curve) is important and pairings should not be treated as "black boxes".

While PBC is an emerging technology, with active research and development, PBC security has generated a large volume of research - many references can be found in [11]. So far, research has fond that PBC maintains the same level of security when compared with standard elliptic curve cryptography.

3 Friendly-PBC Curves

Although in theory pairings exist for any elliptic curve, in practice there are curves whose pairings are not suitable for cryptographic applications. Associated to each elliptic curve, there is a parameter that can be calculated and is known as the embedding degree k. This embedding degree represents the difficulty of turning an elliptic curve system into a classical discrete logarithm system. More precisely, we should talk about the embedding degree of a subgroup of an elliptic curve.

To efficiently implement pairings for use in cryptography, we need k to be relatively small, certainly less than 100. Regular elliptic curve constructions are designed primarily for digital signatures and have very large k. In fact, k is usually about the same size as p.

The elliptic curve secp256k1, which is used in Bitcoin and Ethereum [12], has a 256 bits k value [13],

$k = 1929868153955269923726183083478131754729273798458173971008605235863602490 56$

This makes PBC implementations impractical over secp256k1.

For PBC, it is desired to have elliptic curve E over $\mathbb{F}_q$ such that:

- There is a large prime r dividing $E(\mathbb{F}_p)$, with gcd(r,p)= 1.
- The discrete logarithm problem in $E(\mathbb{F}_p)[r]$ is hard;
- The discrete logarithm problem in $\mathbb{F}^*_{p^k}$ is hard;
- Computation in $E(\mathbb{F}_p)$ and $\mathbb{F}^*_{p^k}$ is efficient;
- Elements of $E(\mathbb{F}_p)[r]$ and $\mathbb{F}^*_{p^k}$ can be represented

Elliptic curves with these properties are called pairing-friendly curves. Note that the conditions are incompatible: for the DLP in $\mathbb{F}^*_{p^k}$ to be hard it is necessary that p^k be large (say, at least 3000 bits) to resist index calculus attacks, whereas to represent elements of $\mathbb{F}^*_{p^k}$ compactly we would like p^k to be small. Luckily, there is a large amount of literature on pairing-friendly elliptic curves and we can use techniques such as those described in [14] for constructing curves of a given embedding degree.

Freeman, Scott and Teske [14] propose the following classification of pairing-friendly elliptic curve to navigate through the forest of constructions:

- Curves not in families. The constructions are based on the complex multiplication method [15]. The algorithm takes as input a prime power p and an integer n, and constructs an elliptic curve over $\mathbb{F}_p$ with n points.
- Families of curves. These methods produce polynomials $p(x)$ and $r(x)$ such that if $p(x_0)$ is a prime power for some value $x_0 \in \mathbb{Z}$, there is an elliptic curve E over $\mathbb{F}_{p(x_0)}$ with a subgroup of order $r(x_0)$ and embedding degree k with respect to r_0. Parametric families have the advantage that the sizes of the finite field and the prime-order subgroup can be varied by specifying x_0.

We will now briefly describe the most popular families of pairing friendly curves.

A Barreto-Naehrig (BN) curve [16] is an elliptic curve E defined over a finite field $\mathbb{F}_p$, $p > 5$, such that its order r and p are prime numbers parametrized by

$$r(x) = 36x^4 + 36x^3 + 18x^2 + 6x + 1$$
$$p(x) = 36x^4 + 36x^3 + 24x^2 + 6x + 1$$

BN curves have an equation of the

$$y^2 = x^3 + b$$

where $b \in \mathbb{F}_p^*$. BN curves have an embedding degree equal to 12.

A Barreto-Lynn-Scott curve (BSL) [17] is also defined over a parametrized prime field $\mathbb{F}_p$ by an equation of the form

$$y^2 = x^3 + b$$

BLS curves are available for different embedding degrees. The parametrizations with $k = 12$ are given by

$$r(x) = x^4 - x^2 + 1$$
$$p(x) = \frac{(x-1)^2(x^4 - x^2 + 1)}{3} + x$$

A Kachia-Schaefer-Scott curve (KSS) [18] is also available for different embedding degrees. If the required embedding degree is 18, this is very similar to BLS curves.

4 PBC Libraries

There are numerous libraries and software frameworks to compute cryptographic pairings and related arithmetic operations in the groups. The interested reader may consult [19] for a list of PBC libraries. In this section, we briefly introduce three of them: Pairing-Based Cryptography library [20], MIRACL Cryptographic SDK [21] and RELIC [22].

The Pairing-Based Cryptography library library is a free C library built on the GMP library, which performs the mathematical operations underlying PBC cryptosystems. The library is designed to be the backbone of implementations of pairing-based cryptosystems. It provides routines such as elliptic curve generation, elliptic curve arithmetic and pairing computations. Pairing times are reasonable and the C language enables implementation in many environments and operating systems. The set of functions is abstract enough that the library can be used even if the programmer possesses only an elementary understanding of pairings.

MIRACL Crypto SDK is a software library written in C/C++ which is widely recognized by developers as the best open source standard for elliptic curve cryptography and is used for studies requiring high performance. MIRACL provides PBC primitives. Since 2011, the library is commercial.

RELIC is a cryptographic tool with emphasis on efficiency and flexibility. It can be used to build efficient and usable cryptographic toolkits tailored for specific security levels and algorithmic choices. RELIC operates under the LGPL license. In terms of PBC, RELIC implements several types of pairings and pairing-based protocols, including pairings over Barreto-Naehrig curves and other parameterized curves at different security levels.

5 Summary

As we have seen, PBC has much to offer. Pairing-based schemes, such as aggregate signature, provide special properties that cannot be provided through traditional PKI in a straightforward way. Therefore, PBC schemes would make a useful addition to the cryptographic toolkit af many blockchain platforms.

References

[1] D. Balfanz, G. Durfee, N. Shankar, D. Smetters, J. Staddon, and Hao-Chi Wong. Secret handshakes from pairing-based key agreements. In *Proceedings of the 2003 IEEE Symposium on Security and Privacy*, SP '03, pages 180–. IEEE Computer Society, 2003.

[2] D. Boneh, B. Lynn, and H. Shacham. Short signatures from the weil pairing. *Journal of Cryptology*, 17(4):297–319, 2004.

[3] Jens Groth. Pairing-based non-interactive zero-knowledge proofs. In *Proceedings of the 4th International Conference on Pairing-based Cryptography*, Pairing'10, pages 206–206. Springer-Verlag, 2010.

[4] A. Menezes, S. Vanstone, and T. Okamoto. Reducing elliptic curve logarithms to logarithms in a finite field. In *Proceedings of the Twenty-third Annual ACM Symposium on Theory of Computing*, STOC '91, pages 80–89. ACM, 1991.

[5] Dan Boneh. Pairing-based cryptography: Past, present, and future. In *Advances in Cryptology – ASIACRYPT 2012*. Springer Berlin Heidelberg, 2012.

[6] R. Gennaro, C. Gentry, B. Parno, and M. Raykova. Quadratic span programs and succinct nizks without pcps. In *Advances in Cryptology – EUROCRYPT 2013*, pages 626–645. Springer Berlin Heidelberg, 2013.

[7] G. Danezis, C. Fournet, M. Kohlweiss, and B. Parno. Pinocchio coin: building zerocoin from a succinct pairing-based proof system. In *PETShop'13, Proceedings of the 2013 ACM Workshop on Language Support for Privacy-Enhancing Technologies, Co-located with CCS 2013, November 4, 2013, Berlin, Germany*, pages 27–30, 2013.

[8] `https://z.cash/`.

[9] J. M. Pollard. A monte carlo method for factorization. *BIT Numerical Mathematics*, 15(3):331–334, 1975.

[10] D. Moody, R. Peralta, R. A. Perlner, Regenscheid A., Roginsky A., and L. Chen. Report on pairing-based cryptography. *Journal of research of the National Institute of Standards and Technology*, 120:11–27, 2015.

[11] R. Granger, D. Page, and N. P. Smart. High security pairing-based cryptography revisited. In *Algorithmic Number Theory*, pages 480–494. Springer Berlin Heidelberg, 2006.

[12] `http://ethdocs.org/en/latest/ethereum-clients/cpp-ethereum/architecture.html`.

[13] `https://safecurves.cr.yp.to/transfer.html`.

[14] D. Freeman, M. Scott, and E. Teske. A taxonomy of pairing-friendly elliptic curves. *Journal of Cryptology*, 23(2):224–280, 2010.

[15] A. O. L. Atkin and F. Morain. Elliptic curves and primality proving. *Math. Comp.*, 61:29–68, 1993.

[16] Paulo S. L. M. Barreto and M. Naehrig. Pairing-friendly elliptic curves of prime order. In *Selected Areas in Cryptography*, pages 319–331. Springer Berlin Heidelberg, 2006.

[17] Paulo S. L. M. Barreto, B. Lynn, and M. Scott. Constructing elliptic curves with prescribed embedding degrees. In *Proceedings of the 3rd International Conference on Security in Communication Networks*, SCN'02, pages 257–267. Springer-Verlag, 2003.

[18] E. J. Kachisa, E. F. Schaefer, and M. Scott. Constructing brezing-weng pairing-friendly elliptic curves using elements in the cyclotomic field. In *Pairing-Based Cryptography – Pairing 2008*, pages 126–135. Springer Berlin Heidelberg, 2008.

[19] `https://gist.github.com/artjomb/f2d720010506569d3a39`.

[20] `https://crypto.stanford.edu/pbc/`.

[21] `https://github.com/miracl/MIRACL`.

[22] `https://github.com/relic-toolkit/relic`.