Pairings are not dead, just resting

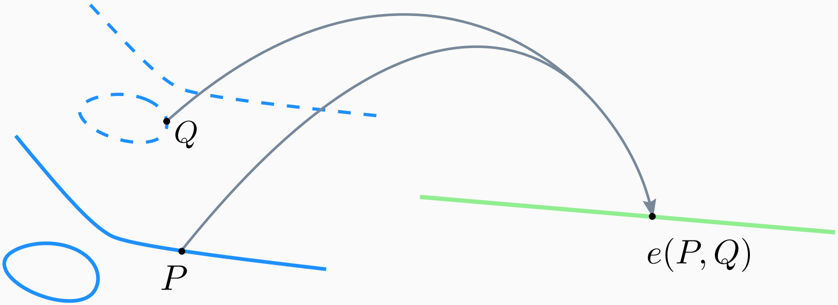
ECC 2017

Diego F. Aranha

December 8, 2018

Institute of Computing – University of Campinas

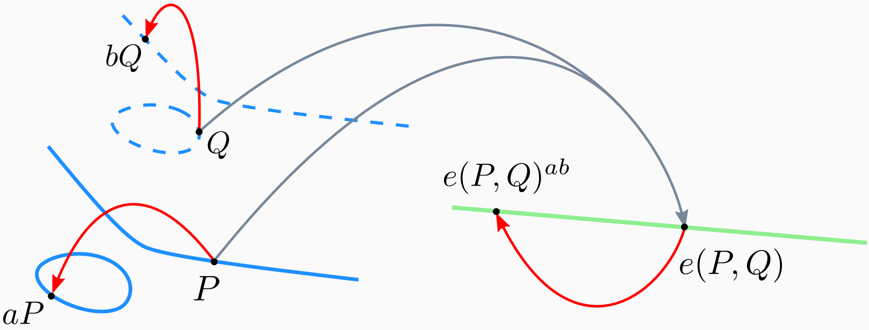
Bilinear pairings



Bilinear pairings

$$e(P+R,Q) = e(P,Q)\cdot e(R,Q) \text{ and } e(P,Q+S) = e(P,Q)\cdot e(P,S)$$

.



Introduction

Elliptic Curve Cryptography (ECC):

- Underlying problem **harder** than integer factoring (RSA)
- Same security level with **smaller** parameters
- Efficiency in storage (**short** keys) and execution time

Pairing-Based Cryptography (PBC):

- Initially **destructive**
- Allows for **innovative** protocols
- Makes curve-based cryptography more **flexible**

Introduction

Pairing-Based Cryptography (PBC) enables many elegant solutions to cryptographic problems:

- **Implicit certification schemes** (IBE, CLPKC, etc.)
- **Short signatures** (in group elements, BLS, BBS)
- **More efficient key agreements** (Joux's 3DH, NIKDS)
- **Low-depth homomorphic encryption** (BGN and variants)
- **Isogeny-based cryptography** (although not postquantum)

Not dead: Pairings are not only interesting for research, but actually deployed in practice!

Disclaimer: I have no conflict of interest with any of the following applications. This is not an endorsement.

Classic: IBE in Voltage's SecureMail

Implemented with supersingular curve over large characteristic [BF01].

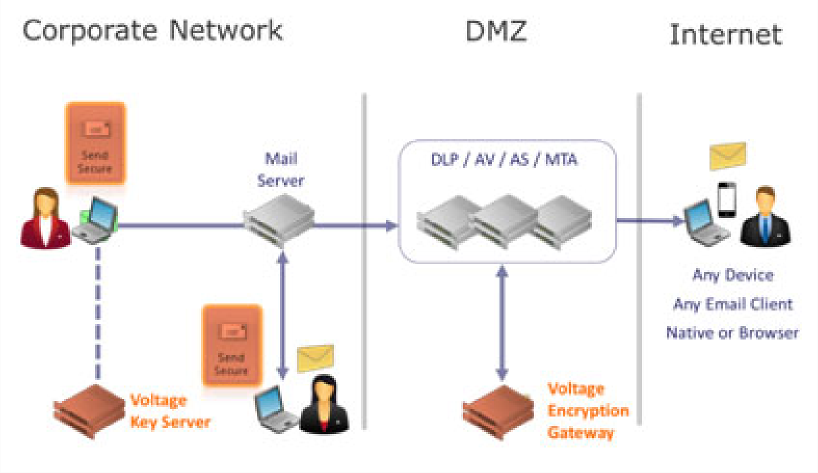


Figure 1: Source: `http://www.securemailworks.com/SecureMail.asp`

Modern applications

IBE in Cloudflare's Geo Key Manager

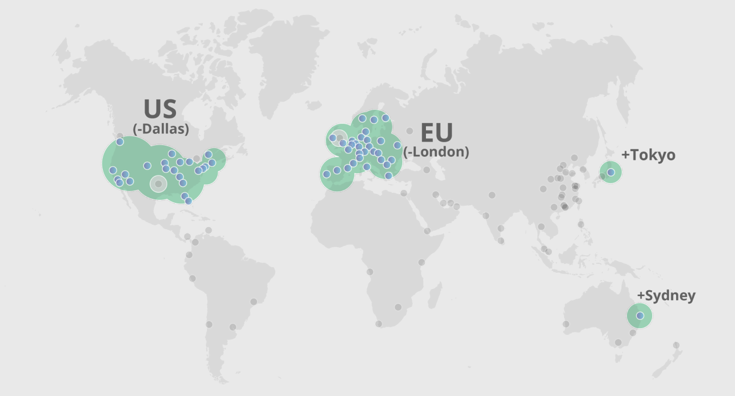


Figure 2:
https://blog.cloudflare.com/geo-key-manager-how-it-works/

IBE in Cloudflare's Geo Key Manager

Implemented using a 256-bit Barreto-Naehrig curve [BN05]

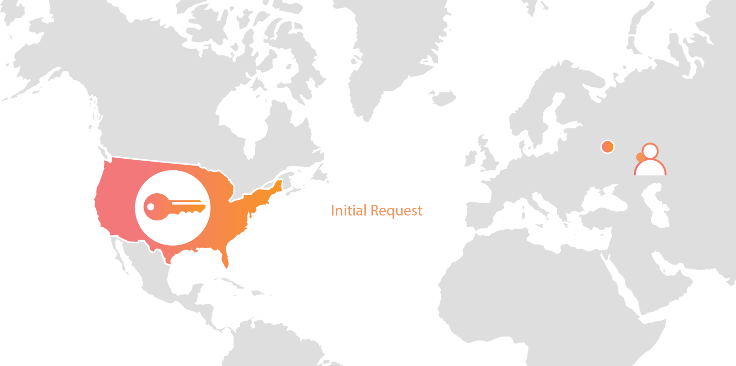


Figure 3:
`https://blog.cloudflare.com/geo-key-manager-how-it-works/`

Remote attestation in Intel SGX

Remote attestation scheme employs a pairing-based anonymous group signature by Brickell and Li (EPID) [BL12].

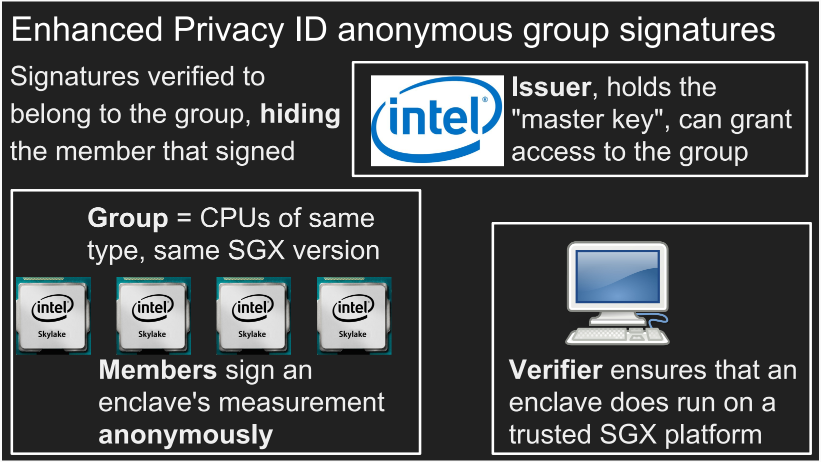


Figure 4: Slides from BlackHat 2016 talk by Aumasson and Merino [AM16].

Remote attestation in Intel SGX

Implemented using a 256-bit Barreto-Naehrig curve [BN05].

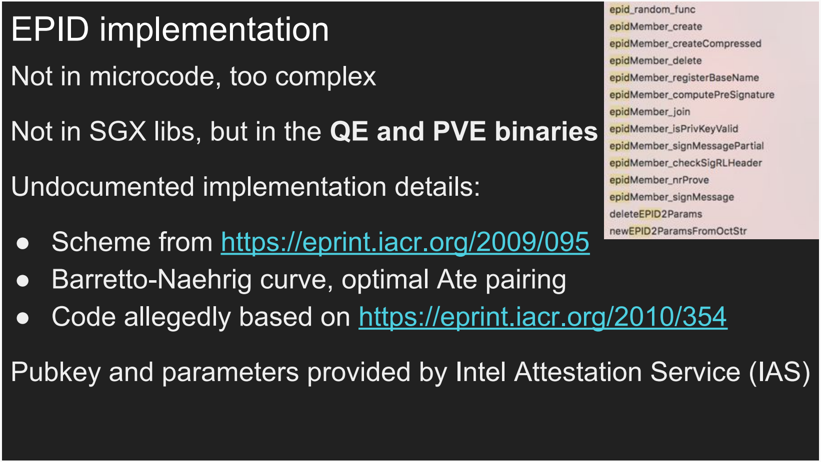


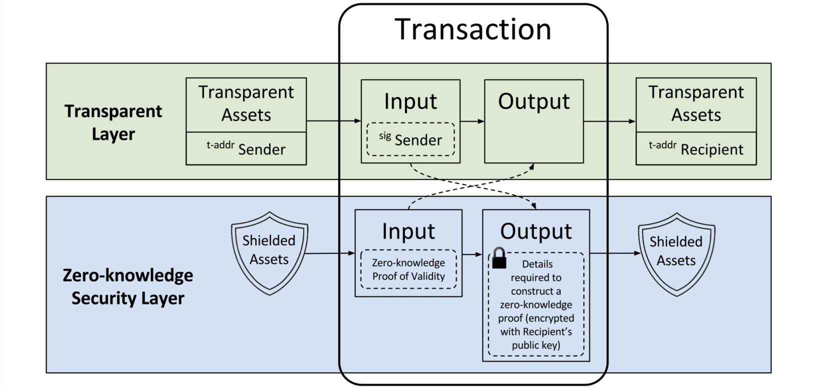
Figure 5: Slides from BlackHat 2016 talk by Aumasson and Merino [AM16].

Authentication in voting machines

Short signature scheme due to Boneh and Boyen [BB04] to link voting machines to specific polling places, using BN 160-bit curve.

Zcash cryptocurrencies

zk-SNARKs by Ben-Sasson et al. [BCG$^+$14] for privacy-preserving cryptocurrencies, also recently adopted by Ethereum.



What is dead about pairings?

However, some things about pairings are dead:

1. **Pairings over small char**, due to many advances in the DLP, including a quasi-polynomial algorithm by Barbulescu et al. [BGJT14]

What is dead about pairings?

However, some things about pairings are dead:

1. **Pairings over small char**, due to many advances in the DLP, including a quasi-polynomial algorithm by Barbulescu et al. [BGJT14]
2. **Pairing conference series** after 6 editions, last one in 2013.

Figure 6: Source: `http://www.ieccr.net/2013/pairing2013/`

What is dead about pairings?

Beware of the **fake** knock-off:

What is dead about pairings?

Beware of the **fake** knock-off:

Presentation Program
Conference Program

SESSION 1

Chair : Phutthiwat Waiyawuththanapoom

1 **Factors Associated with Hotel Employees' Loyalty: A Case Study of Hotel Employees in Bangkok, Thailand**
Kevin Wongleedee
International College, Suan Sunandha Rajabhat University Thailand

2 **Motivation Needs in Working of the Employees in Rayong Province: A Case Study of Panakom Co., Ltd.**
Ganratchakan Ninlawan, Witthaya Mekhum
Suan Sunandha Rajabhat University Thailand

Background

Pairing groups

Let $\mathbb{G}_1 = \langle P \rangle$ and $\mathbb{G}_2 = \langle Q \rangle$ be additive groups and $\mathbb{G}_T$ be a multiplicative group such that $|\mathbb{G}_1| = |\mathbb{G}_2| = |\mathbb{G}_T| =$ prime r.

A general pairing

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

- $\mathbb{G}_1$ is typically a subgroup of $E(\mathbb{F}_p)$.
- $\mathbb{G}_2$ is typically a subgroup of $E(\mathbb{F}_{p^k})$.
- $\mathbb{G}_T$ is a multiplicative subgroup of $\mathbb{F}_{p^k}^*$.

Hence pairing-based cryptography involves arithmetic in $\mathbb{F}_{p^k}$, for **embedding degree** k.

Pairing operations

A general pairing

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

Cryptographic schemes require multiple operations in pairing groups:

1. **Exponentiation**, **membership testing**, **compression** in $\mathbb{G}_1$, $\mathbb{G}_2$ and $\mathbb{G}_T$.
2. **Hashing** strings to $\mathbb{G}_1$, $\mathbb{G}_2$.
3. **Efficient maps** between $\mathbb{G}_1$ and $\mathbb{G}_2$.
4. Efficient **pairing computation**.

Problem: In practice, we want small k for efficient pairing!

Curve families

At some point, pairing-based cryptography had an **explosion** of parameter choices to choose from:

BN curves: $k = 12, \ \rho \approx 1$ $p(x) = 36x^4 + 36x^3 + 24x^2 + 6x + 1$ $r(x) = 36x^4 + 36x^3 + 18x^2 + 6x + 1, \quad t(x) = 6z^2 + 1$
BLS12 curves: $k = 12, \ \rho \approx 1.5$ $p(x) = (x-1)^2(x^4 - x^2 + 1)/3 + x,$ $r(x) = x^4 - x^2 + 1, \quad t(x) = x + 1$
KSS18 curves: $k = 18, \ \rho \approx 4/3$ $p(x) = (x^8 + 5x^7 + 7x^6 + 37x^5 + 188x^4 + 259x^3 + 343x^2 + 1763x + 2401)/21$ $r(x) = (x^6 + 37x^3 + 343)/343, \ \ t(x) = (x^4 + 16z + 7)/7$
BLS24 curves: $k = 24, \ \rho \approx 1.25$ $p(x) = (x-1)^2(x^8 - x^4 + 1)/3 + x,$ $r(x) = x^8 - x^4 + 1, \quad t(x) = x + 1$

Barreto-Naehrig curves

Let $x \in \mathbb{Z}$ such that $p(x)$ and $r(x)$ are prime:

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

Then $E : y^2 = x^3 + b,\ b \in \mathbb{F}_p$ is a curve of **order** r and **embedding degree** $k = 12$ [BN05] and E' its **twist** of degree $d = 6$.

Fix $x = -(2^{62} + 2^{55} + 1)$ and $b = 2$, the towering can be:

- $\mathbb{F}_{p^2} = \mathbb{F}_p[i]/(i^2 - \beta)$, where $\beta = -1$
- $\mathbb{F}_{p^4} = \mathbb{F}_{p^2}[s]/(s^2 - \epsilon)$, where $\xi = 1 + i$
- $\mathbb{F}_{p^6} = \mathbb{F}_{p^2}[v]/(v^3 - \xi)$, where $\xi = 1 + i$
- $\mathbb{F}_{p^{12}} = \mathbb{F}_{p^4}[v]/(t^3 - s)$ or $\mathbb{F}_{p^6}[w]/(w^2 - v)$

Until recently: BN curves **were** king at the 128-bit security level and got even close to **standardization** (IETF RFC).

Barreto-Naehrig curves

Instantiating pairings over BN curves had **many** performance features:

1. Implementation-friendly parameters, with **fast towering** and compact generators [GJNB11].
2. **Prime-order** group $\mathbb{G}_1$, facilitating protocols.
3. Twist of **maximum degree**, reducing size of $\mathbb{G}_2$.
4. Gallant-Lambert-Vanstone [GLV01] **endomorphism** in $\mathbb{G}_1$.
5. Galbraith-Scott **homomorphism** [GS08] in $\mathbb{G}_2$, $\mathbb{G}_T$.
6. Compressed squarings for **exponentiation** in $\mathbb{G}_T$.

Barreto-Naehrig curves

Instantiating pairings over BN curves had **many** performance features:

1. Implementation-friendly parameters, with **fast towering** and compact generators [GJNB11].
2. **Prime-order** group $\mathbb{G}_1$, facilitating protocols.
3. Twist of **maximum degree**, reducing size of $\mathbb{G}_2$.
4. Gallant-Lambert-Vanstone [GLV01] **endomorphism** in $\mathbb{G}_1$.
5. Galbraith-Scott **homomorphism** [GS08] in $\mathbb{G}_2$, $\mathbb{G}_T$.
6. Compressed squarings for **exponentiation** in $\mathbb{G}_T$.

Alfred Menezes, 2007

"These curves should not exist, they are too good to be true."

Recent DLP attacks on the medium-prime case

In 2015, Kim and Barbulescu [KB16] proposed a variant of the NFS that **reduces the complexity** of the DLP in $\mathbb{F}_{p^k}$ in time $L[1/3, \left(\frac{48}{9}\right)^{1/3}]$ or $L[1/3, \left(\frac{32}{9}\right)^{1/3}]$ for special primes p.

Direct consequences of these attacks on BN curves:

1. BLS signatures are not as **short** anymore. You can obtain similar sizes with Schnorr and **preimage-resistant** hashing [NSW09].
2. Previous curves at 128-bit security now provide 100 bits of security. **Not much impact** on curves at the 80-bit level.
3. Pairings may not be viable anymore on **memory-constrained** devices.

Curve families

And now we are somewhat **back** to that situation again. Recently proposed parameters, from the most conservative:

1. Elliptic curves with embedding degree $k = 1$ (**large base field**) [CMR17]
2. Symmetric pairings with prime embedding degree $k = 2, 3$ (**still large base field**) [Sco05, ZW13]
3. Elliptic curves with **less smooth** embedding degrees (ordinary with $k = 9, 13, 15, 21, 27$)

→ **Adjusted** field sizes and smooth embedding degrees such as Barreto-Lynn-Scott (BLS) and Kachisa-Scott-Schaefer (KSS) curves [BLS02, KSS08].

Previous work has demonstrated that BLS12 curves were **promising** at the **old** 192-bit security level [AFK$^+$12].

Implementation techniques

Arithmetic levels

Protocols

$e(P, Q)$

$E(\mathbb{F}_q)$ $E(\mathbb{F}_{q^k})$

$\mathbb{F}_{q^k}$

$\mathbb{F}_q$

Low-level backend

Software libraries

There are many different open-source software implementations of pairings:

- **PBC**: on top of GMP, **outdated**.
- **Panda**: not as efficient anymore, but **constant-time**.
- **Ate-pairing:** CINVESTAV, **previous** state of the art.
- **MIRACL**: special support for constrained platforms.
- **Apache Milagro**: fast C and bindings to many languages.
- **OpenPairing**: OpenSSL patch, never merged.
- **mcl:** new library at **new** 128-bit level by Shigeo Mitsunari.

Software libraries

There are many different open-source software implementations of pairings:

- **PBC**: on top of GMP, **outdated**.
- **Panda**: not as efficient anymore, but **constant-time**.
- **Ate-pairing:** CINVESTAV, **previous** state of the art.
- **MIRACL**: special support for constrained platforms.
- **Apache Milagro**: fast C and bindings to many languages.
- **OpenPairing**: OpenSSL patch, never merged.
- **mcl:** new library at **new** 128-bit level by Shigeo Mitsunari.

$\rightarrow$ **RELIC**: UNICAMP, flexible and **current** state of the art.

Finite field arithmetic

Target platform: Desktop processor.

1. An efficient 64-bit implementation of the base field arithmetic typically employs:
    - **Montgomery** representation.
    - Wide multiplication instructions `MUL` and `MULX`.
    - **Lazy reduction**:

    $$(a \cdot b) \bmod p + (c \cdot d) \bmod p = (a \cdot b + c \cdot d) \bmod p$$

    Open: Can CPU vector instruction improve the asymptotically faster **Residue Number Systems** (RNS)?

2. Techniques for extension field arithmetic:
    - **Small** quadratic/cubic non-residues and **change of representation**.
    - **Fastest** formulas available in the literature (asymmetric squarings due to [CH07].
    - **General** lazy reduction: k reductions for $\mathbb{F}_{p^k}$ arithmetic [AKL$^+$11].

Operations in $\mathbb{G}_1$ and $\mathbb{G}_2$

Scalar multiplications in $\mathbb{G}_1$ and $\mathbb{G}_2$ follow standard techniques, such as projective coordinates and signed recodings.

Scalars can be decomposed using the GLV method when **endomorphism** ψ is available: $\ell \equiv \ell_0 + \lambda\ell_1 \pmod{r} \rightarrow [\ell]P = [\ell_0]P + [\ell_1]\psi(P)$.

Hashing to $\mathbb{G}_1$ and $\mathbb{G}_2$ involves hashing to point and multiplying by **cofactor** represented in base p [SBC$^+$09, FKR11].

Operations in $\mathbb{G}_T$

Pairing result is an element of the **cyclotomic subgroup** $\mathbb{G}_{\phi_k}(\mathbb{F}_{p^{k/d}})$.

Given $C(g)$, efficient to compute $C(g^2)$ as shown by Karabina in [Kar13].

Idea: $g^{|u|=2^a-2^b+1}$ can now be computed in three steps:

1. Compute $C(g^{2^i})$ for $1 \leq i \leq a$ and store $C(g^{2^b})$ and $C(g^{2^a})$
2. Compute $D(C(g^{2^a})) = g^{2^a}$ and $D(C(g^{2^b})) = g^{2^b}$
3. Compute $g^{|x|} = g^{2^a} \cdot (g^{2^b})^{k/2} \cdot g$

Remark 1: Montgomery's simultaneous inversion allows **simultaneous decompression**.

Remark 2: For dense exponent, plain cyclotomic squarings can be used instead [GS10]. **Signed recodings** can be used because inversion is **conjugation**, and base-$(t-1)$ expansions due to $g^p = g^{t-1}$.

Pairing computation

Algorithm 1 Tate pairing [BKLS02].

Input: $r = \sum_{i=0}^{\log_2 r} r_i 2^i, P, Q$.

Output: $e_r(P, Q)$.

$T \leftarrow P$

$f \leftarrow 1$

for $i = \lfloor \log_2(r) \rfloor - 1$ **downto** 0 **do**

  $T \leftarrow 2T$

  $f \leftarrow f^2 \cdot l_{T,T}(Q)$

  if $r_i = 1, i \neq 0$ **then**

    $T \leftarrow T + P$

    $f \leftarrow f \cdot l_{T,P}(Q)$

  end if

end for

return $f^{(q^k - 1/r)}$

Pairing computation

A pairing computation essentially consists in the **Miller loop** followed by the **final exponentiation**.

1. An efficient implementation of the Miller loop requires:
   - **Low Hamming weight** of the integer parameter.
   - Efficient formulas for **curve arithmetic** (homogeneous coordinates).
   - Curve arithmetic combined together with computation of the **line evaluations**.

2. And the final exponentiation:
   - For even k, split the final exponent as $(p^k - 1)/\phi_k(p) \cdot \phi_k(p)/r$.
   - Easy part computed with **Frobenius**.
   - Hard part computed with decomposition in base p and **vectorial addition chain**.
   - Compressed squarings in cyclotomic subgroup.

Pairing computation

Other optimizations are possible:

1. **Optimal ate construction** to minimize integer parameter by $\phi(k)$ [Ver10].
2. **Fixed argument pairings** precomputes Miller loop when argumets are fixed [CS10].
3. **Product of pairings** to share final exponentiation when evaluating $\prod_{i=0}^{m} e(P_i, Q_i)$.

Subgroup security

A security property mandating that cofactors have only large prime factors to prevent small subgroup attacks [BCM+15]. Started as "$\mathbb{G}_T$-strong" notion of security [Sco13].

In general, **subgroup membership testing** is easy in $\mathbb{G}_1$ (validity or scalar multiplication).

In $\mathbb{G}_2$, we can exploit $n = p - t + 1$ and check if $[p]Q = [t-1]Q$.

Subgroup security

A security property mandating that cofactors have only large prime factors to prevent small subgroup attacks [BCM+15]. Started as "$\mathbb{G}_T$-strong" notion of security [Sco13].

In general, **subgroup membership testing** is easy in $\mathbb{G}_1$ (validity or scalar multiplication).

In $\mathbb{G}_2$, we can exploit $n = p - t + 1$ and check if $[p]Q = [t-1]Q$.

Faster: protocols can be modified instead to **multiply by cofactors**.

In a subgroup-secure curve with prime $\phi_k(p)/r$, membership testing in $\mathbb{G}_T$ is easy by checking if $g^{\phi_k(p)} = 1$.

Impact: subgroup-secure curves slightly penalize pairing computation but save on membership tests.

New results

Implementation

Characteristics of the implementation:

- Target platform: Intel Skylake 64-bit processors.
- Library: RELIC is an Efficient LIbrary for Cryptography (`github.com/relic-toolkit/relic`)
- Compiler: GCC 7.2.0 with flags `-O3 -fomit-frame-point -funroll-loops`

Open: Still under heavy development!

Implementation

Characteristics of the implementation:

- Target platform: Intel Skylake 64-bit processors.
- Library: RELIC is an Efficient LIbrary for Cryptography (`github.com/relic-toolkit/relic`)
- Compiler: GCC 7.2.0 with flags `-O3 -fomit-frame-point -funroll-loops`

Open: Still under heavy development!

Comparison between two sets of parameters:

1. BN vs BLS12 curves.
2. BLS12 vs KSS16 curves.

BN vs BLS12

Parameter sizes suggested by Menezes et al. [MSS16]: subgroup-secure BN-382 tweeted by Barreto, and BLS12-381 from ZCash (Sapling).

Operation	BN-254	BN-382	BLS12-381
kP in $\mathbb{G}_1$	200	564	386
kQ in $\mathbb{G}_2$	459	1465	968
g^k in $\mathbb{G}_T$	719	2284	1500
H to $\mathbb{G}_1$	58	180	500
H to $\mathbb{G}_2$	248	760	960
Test $\mathbb{G}_1$	0.306	0.691	323
Test $\mathbb{G}_2$	173	519	391
Test $\mathbb{G}_T$	271	713 (9[1])	3911
$e(P, Q)$ (M+F)	583+406=989	1950+1291=3241	1310+1512=2822

Table 1: Timings from RELIC in 10^3 cycles in Skylake processor measured as average of 10^4 executions (HT and TB disabled).

[1](*) Faster test in $\mathbb{G}_{\phi_k}(\mathbb{F}_{p^{k/d}})$.

BLS12 vs KSS16

Parameters suggested by Barbulescu and Duquesne [BD17]: curves BLS12-461 and KSS16-340. Advantages of BLS12 over KSS16:

1. Twist with **larger degree** and smaller $\mathbb{G}_2$ representation.
2. Compressed squarings due to $d = 6$.
3. Subgroup security.

Operation	KSS16-340	BLS12-461
$e(P, Q)$ (M+F)	1567+3856=5423	2547+2604=5151

Table 2: Timings from RELIC in 10^3 cycles in Skylake processor measured as average of 10^4 executions (HT and TB disabled).

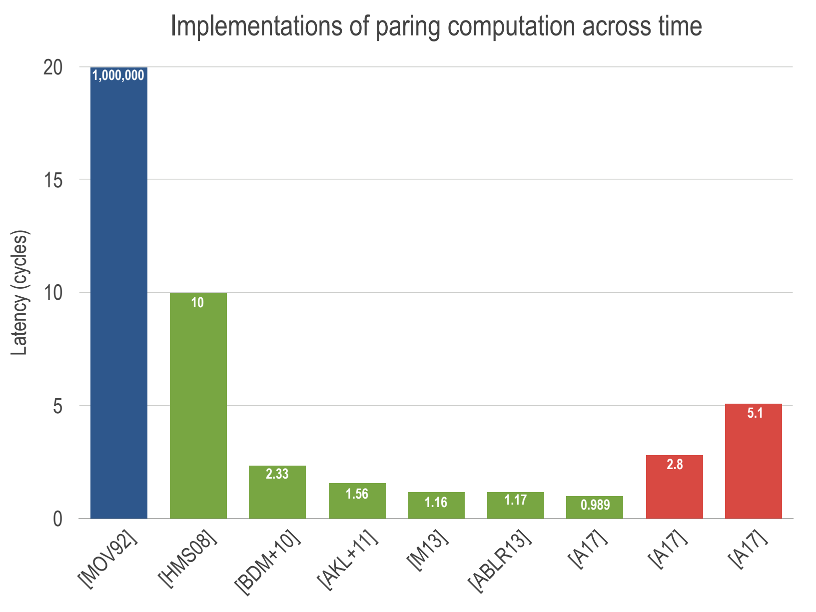
Beware: There is still **plenty** to do in terms of optimizing arithmetic in the recently proposed KSS16 curve.

History of pairing implementations

Implementation	Curve	(10^6 cycles)
MOV92	Supersingular	Billions
HMS08	256-bit BN	10.0
NNS10	256-bit BN	4.38
BDM+10	256-bit BN	2.33
AKL+11	254-bit BN	1.56
M13	254-bit BN	1.16
ABLR13	254-bit BN	1.17
This work	254-bit BN	0.99
This work (optimistic)	381-bit BLS12	2.82
This work (conservative)	461-bit BLS12	5.15

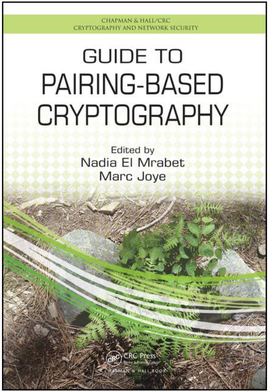
Table 3: Speed records for pairing computation in the past decades.

History of pairing implementations



Further reading

1. *Pairings for Beginners*, by Craig Costello.
2. Guide to Pairing-Based Cryptography:



Questions?

D. F. Aranha

dfaranha@ic.unicamp.br

@dfaranha

References i

- Diego F. Aranha, Laura Fuentes-Castañeda, Edward Knapp, Alfred Menezes, and Francisco Rodríguez-Henríquez.
  Implementing pairings at the 192-bit security level.
  In *Pairing*, volume 7708 of *Lecture Notes in Computer Science*, pages 177–195. Springer, 2012.
- Diego F. Aranha, Koray Karabina, Patrick Longa, Catherine H. Gebotys, and Julio López.
  Faster explicit formulas for computing pairings over ordinary curves.
  In *EUROCRYPT*, volume 6632 of *Lecture Notes in Computer Science*, pages 48–68. Springer, 2011.
- Jean Philippe Aumasson and Luis Merino.
  Sgx secure enclaves in practice: Security and crypto review.
  BlackHat, 2016.

Dan Boneh and Xavier Boyen.
Short signatures without random oracles.
In *EUROCRYPT*, volume 3027 of *Lecture Notes in Computer Science*, pages 56–73. Springer, 2004.

Eli Ben-Sasson, Alessandro Chiesa, Christina Garman, Matthew Green, Ian Miers, Eran Tromer, and Madars Virza.
Zerocash: Decentralized anonymous payments from bitcoin.
In *IEEE Symposium on Security and Privacy*, pages 459–474. IEEE Computer Society, 2014.

Paulo S. L. M. Barreto, Craig Costello, Rafael Misoczki, Michael Naehrig, Geovandro C. C. F. Pereira, and Gustavo Zanon.
Subgroup security in pairing-based cryptography.
In *LATINCRYPT*, volume 9230 of *Lecture Notes in Computer Science*, pages 245–265. Springer, 2015.

Razvan Barbulescu and Sylvain Duquesne.
Updating key size estimations for pairings.
IACR Cryptology ePrint Archive, 2017:334, 2017.

Dan Boneh and Matthew K. Franklin.
Identity-based encryption from the weil pairing.
In *CRYPTO*, volume 2139 of *Lecture Notes in Computer Science*, pages 213–229. Springer, 2001.

Razvan Barbulescu, Pierrick Gaudry, Antoine Joux, and Emmanuel Thomé.
A heuristic quasi-polynomial algorithm for discrete logarithm in finite fields of small characteristic.
In *EUROCRYPT*, volume 8441 of *Lecture Notes in Computer Science*, pages 1–16. Springer, 2014.

- Paulo S. L. M. Barreto, Hae Yong Kim, Ben Lynn, and Michael Scott.
  Efficient algorithms for pairing-based cryptosystems.
  In *CRYPTO*, volume 2442 of *Lecture Notes in Computer Science*, pages 354–368. Springer, 2002.
- Ernie Brickell and Jiangtao Li.
  Enhanced privacy ID: A direct anonymous attestation scheme with enhanced revocation capabilities.
  IEEE Trans. Dependable Sec. Comput., 9(3):345–360, 2012.
- Paulo S. L. M. Barreto, Ben Lynn, and Michael Scott.
  Constructing elliptic curves with prescribed embedding degrees.
  In *SCN*, volume 2576 of *Lecture Notes in Computer Science*, pages 257–267. Springer, 2002.

Paulo S. L. M. Barreto and Michael Naehrig.
Pairing-friendly elliptic curves of prime order.
In *Selected Areas in Cryptography*, volume 3897 of *Lecture Notes in Computer Science*, pages 319–331. Springer, 2005.

Jaewook Chung and M. Anwar Hasan.
Asymmetric squaring formulae.
In *IEEE Symposium on Computer Arithmetic*, pages 113–122. IEEE Computer Society, 2007.

Sanjit Chatterjee, Alfred Menezes, and Francisco Rodríguez-Henríquez.
On instantiating pairing-based protocols with elliptic curves of embedding degree one.
IEEE Trans. Computers, 66(6):1061–1070, 2017.

Craig Costello and Douglas Stebila.
Fixed argument pairings.
In *LATINCRYPT*, volume 6212 of *Lecture Notes in Computer Science*, pages 92–108. Springer, 2010.

Laura Fuentes-Castañeda, Edward Knapp, and Francisco Rodríguez-Henríquez.
Faster hashing to ${\mathbb G}_2$.
In *Selected Areas in Cryptography*, volume 7118 of *Lecture Notes in Computer Science*, pages 412–430. Springer, 2011.

C. C. F. Pereira Geovandro, Marcos A. Simplício Jr., Michael Naehrig, and Paulo S. L. M. Barreto.
A family of implementation-friendly BN elliptic curves.
Journal of Systems and Software, 84(8):1319–1326, 2011.

Robert P. Gallant, Robert J. Lambert, and Scott A. Vanstone.
Faster point multiplication on elliptic curves with efficient endomorphisms.
In *CRYPTO*, volume 2139 of *Lecture Notes in Computer Science*, pages 190–200. Springer, 2001.

Steven D. Galbraith and Michael Scott.
Exponentiation in pairing-friendly groups using homomorphisms.
In *Pairing*, volume 5209 of *Lecture Notes in Computer Science*, pages 211–224. Springer, 2008.

Robert Granger and Michael Scott.
Faster squaring in the cyclotomic subgroup of sixth degree extensions.
In *Public Key Cryptography*, volume 6056 of *Lecture Notes in Computer Science*, pages 209–223. Springer, 2010.

Koray Karabina.
Squaring in cyclotomic subgroups.
Math. Comput., 82(281):555–579, 2013.

Taechan Kim and Razvan Barbulescu.
Extended tower number field sieve: A new complexity for the medium prime case.
In *CRYPTO (1)*, volume 9814 of *Lecture Notes in Computer Science*, pages 543–571. Springer, 2016.

Ezekiel J. Kachisa, Edward F. Schaefer, and Michael Scott.
Constructing brezing-weng pairing-friendly elliptic curves using elements in the cyclotomic field.
In *Pairing*, volume 5209 of *Lecture Notes in Computer Science*, pages 126–135. Springer, 2008.

Alfred Menezes, Palash Sarkar, and Shashank Singh.
Challenges with assessing the impact of NFS advances on the security of pairing-based cryptography.
In *Mycrypt*, volume 10311 of *Lecture Notes in Computer Science*, pages 83–108. Springer, 2016.

Gregory Neven, Nigel P. Smart, and Bogdan Warinschi.
Hash function requirements for schnorr signatures.
J. Mathematical Cryptology, 3(1):69–87, 2009.

Michael Scott, Naomi Benger, Manuel Charlemagne, Luis J. Dominguez Perez, and Ezekiel J. Kachisa.
Fast hashing to $\mathbb{G}_2$ on pairing-friendly curves.
In *Pairing*, volume 5671 of *Lecture Notes in Computer Science*, pages 102–113. Springer, 2009.

Michael Scott.
Computing the tate pairing.
In *CT-RSA*, volume 3376 of *Lecture Notes in Computer Science*, pages 293–304. Springer, 2005.

Michael Scott.
Unbalancing pairing-based key exchange protocols.
IACR Cryptology ePrint Archive, 2013:688, 2013.

- Frederik Vercauteren.
  Optimal pairings.
  IEEE Trans. Information Theory, 56(1):455–461, 2010.
- Xusheng Zhang and Kunpeng Wang.
  Fast symmetric pairing revisited.
  In *Pairing*, volume 8365 of *Lecture Notes in Computer Science*, pages 131–148. Springer, 2013.