# Optimized and secure pairing-friendly elliptic curve suitable for one layer proof composition

Youssef El Housni[1] Aurore Guillevic[2]

[1]Ernst & Young, Inria and École polytechnique, Paris, France
youssef.el.housni@fr.ey.com

[2]Université de Lorraine, CNRS, Inria, LORIA, Nancy, France
aurore.guillevic@inria.fr

ZKProof Lightning talks, April 2020

# Overview

1. Proof composition
   - Notations
2. Our work
   - Theory
   - Implementation
3. Applications

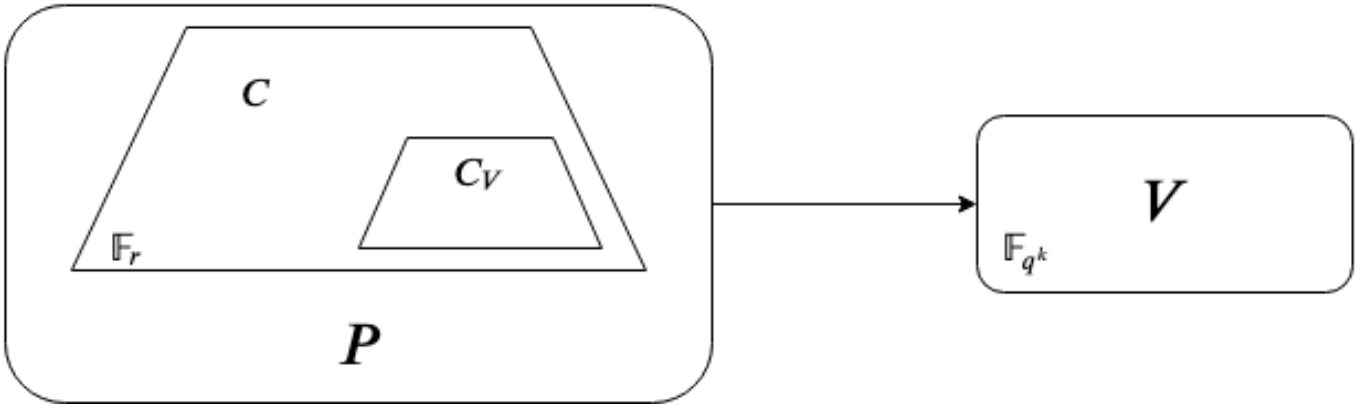
# Notations

## Pairing-based zkSNARK for NP-language

- $E: y^2 = x^3 + ax + b$ elliptic curve defined over $\mathbb{F}_q$, $q$ a prime power.
- $r$ prime divisor of $\#E(\mathbb{F}_q) = q + 1 - t$, $t$ Frobenius trace.
- $k$ embedding degree, smallest integer $k \in \mathbb{N}^*$ s.t. $r \mid q^k - 1$.
- pairing $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ where:
  - $\mathbb{G}_1 \subset E(\mathbb{F}_q)$ and $\mathbb{G}_2 \subset E(\mathbb{F}_{q^k})$ two groups of order $r$.
  - $\mathbb{G}_T \subset \mathbb{F}_{q^k}^*$ group of $r$-th roots of unity.

# Proof composition

## A proof of a proof

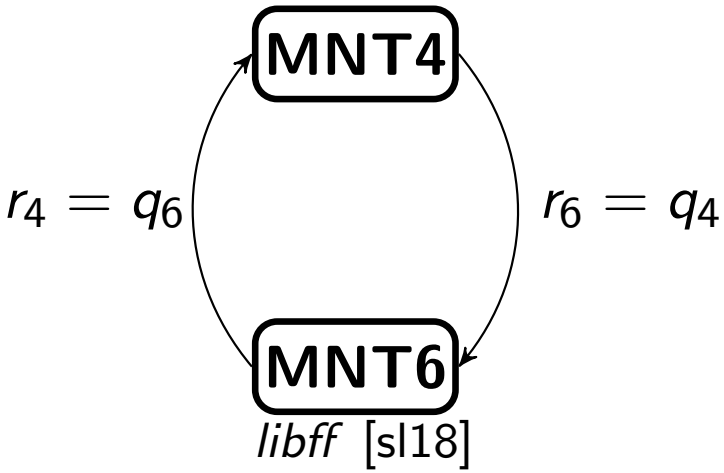


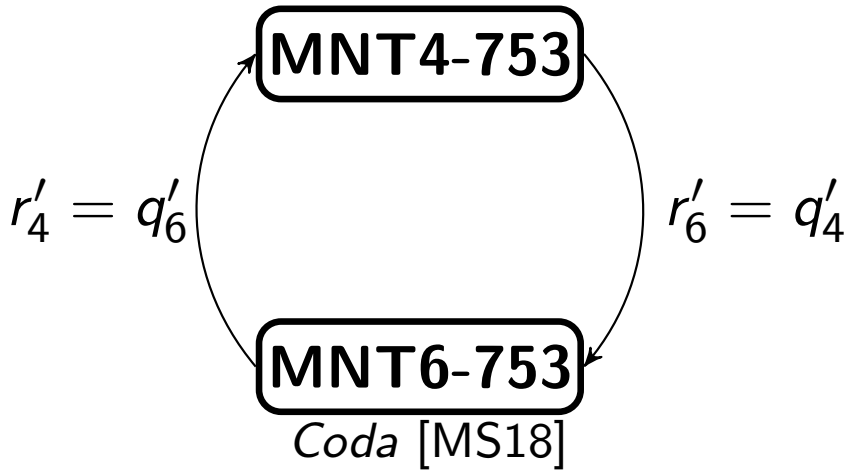

The verification algorithm $V$ is a NP program $\implies$ generate a new proof that verifies the correctness the old proof.
For pairing-based SNARKs, $V$ is in $\mathbb{F}_{q^k}$ and $P$ in $\mathbb{F}_r$, where $q$ is the field size of an elliptic curve $E$ and $r$ its prime subgroup order.

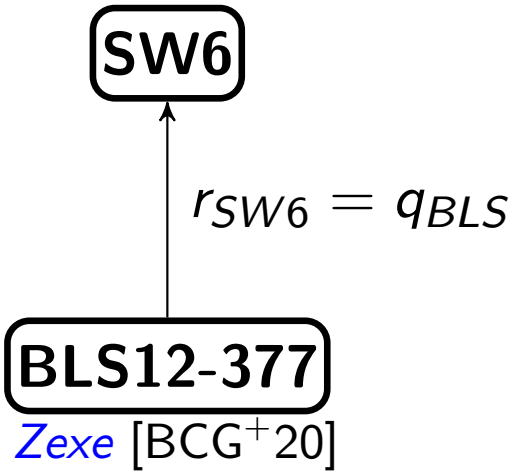
- $1^{\text{st}}$ attempt: choose a curve for which $q = r$ (impossible)
- $2^{\text{nd}}$ attempt: simulate $\mathbb{F}_q$ operations via $\mathbb{F}_r$ operations ($\times \log q$ blowup)
- $3^{\text{rd}}$ attempt: use a cycle/chain of pairing-friendly elliptic curves [BCTV14, BCG+20]

# Proof composition

## cycles and chains of pairing-friendly elliptic curves



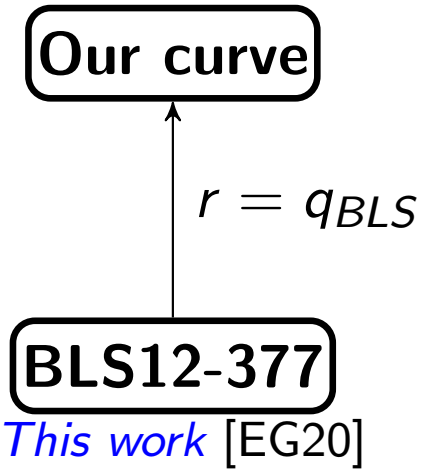




Figure: Examples of pairing-friendly amicable cycles and chains.

# Proof composition

## cycles and chains of pairing-friendly elliptic curves

| $E/\mathbb{F}_q$ | $q$ | $r$ | $k$ | $d$ | $a, b$ | $\lambda$ |
|---|---|---|---|---|---|---|
| MNT4 | $q_4 = r_6$ (298b) | $r_4 = q_6$ (298b) | 4 | 2 | $a = 2, b = *$ | 32 |
| MNT6 | $q_6 = r_4$ (298b) | $r_6 = q_4$ (298b) | 6 | 2 | $a = 11, b = *$ | 50 |
| MNT4-753 | $q'_4 = r'_6$ (753b) | $r'_4 = q'_6$ (753b) | 4 | 2 | $a = 2, b = *$ | 128 |
| MNT6-753 | $q'_6 = r'_4$ (753b) | $r'_6 = q'_4$ (753b) | 6 | 2 | $a = 11, b = *$ | 128 |
| BLS12-377 | $q_{BLS}$ (377b) | $r_{BLS}$ (253b) | 12 | 6 | $a = 0, b = 1$ | 128 |
| SW6 | $q_{SW6}$ (782b) | $r_{SW6} = q_{BLS}$ (377b) | 6 | 2 | $a = 5, b = *$ | 128 |
| This work | $q$ (761b) | $r = q_{BLS}$ (377b) | 6 | 6 | $a = 0, b = -1$ | 128 |

Table: 2-cycle and 2-chain examples.

Recall that $E/\mathbb{F}_q : y^2 = x^3 + ax + b$ has a subgroup of order $r$, an embedding degree $k$, a twist of order $d$ and an approximate security of $\lambda$-bit.

# Our work

## ZK-curves

- SNARK
  - $E/\mathbb{F}_q$ — BN, BLS12, BW12?, KSS16? ... [FST10]
    - pairing-friendly
    - $r-1$ highly 2-adic
- Recursive SNARK (2-cycle)
  - $E_1/\mathbb{F}_{q_1}$ and $E_2/\mathbb{F}_{q_2}$ — MNT4/MNT6 [FST10, Sec.5], ? [CCW19]
    - both pairing-friendly
    - $r_2 = q_1$ and $r_1 = q_2$
    - $r_{\{1,2\}} - 1$ highly 2-adic
    - $q_{\{1,2\}} - 1$ highly 2-adic
- Recursive SNARK (2-chain)
  - $E_1/\mathbb{F}_{q_1}$ — BLS12 ($seed \equiv 1 \mod 3.2^{adicity}$) [BCG+20], ?
    - pairing-friendly
    - $r_1 - 1$ highly 2-adic
    - $q_1 - 1$ highly 2-adic
  - $E_2/\mathbb{F}_{q_2}$ — Cocks–Pinch algorithm
    - pairing-friendly
    - $r_2 = q_1$

# Our work

## The suggested curve: BW6-761

We found the following curve $E : y^2 = x^3 - 1$ over $\mathbb{F}_q$ of 761-bit. The parameters are expressed in polynomial forms and evaluated at the seed $x_0 = 0\text{x}8508\text{c}00000000$. For pairing computation we use the M-twist curve $E' : y^2 = x^3 + 4$ over $\mathbb{F}_q$ to represent $\mathbb{G}_2$ coordinates.

| Our curve, $k = 6$, $D = 3$, $r = q_{BLS12-377}$ |
|---|
| $r(x) = (x^6 - 2x^5 + 2x^3 + x + 1)/3 = q_{BLS12-377}(x)$ |
| $t(x) = x^5 - 3x^4 + 3x^3 - x + 3 + h_t r(x)$ |
| $y(x) = (x^5 - 3x^4 + 3x^3 - x + 3)/3 + h_y r(x)$ |
| $q(x) = (t^2 + 3y^2)/4$ |
| $q_{h_t=13,h_y=9}(x) = (103x^{12} - 379x^{11} + 250x^{10} + 691x^9 - 911x^8$ |
| $-79x^7 + 623x^6 - 640x^5 + 274x^4 + 763x^3 + 73x^2 + 254x + 229)/9$ |

# Our work

## Features

- The curve is over 761-bit instead of 782-bit, we save one machine-word of 64 bits.
- The curve has an embedding degree $k = 6$ and a twist of order $d = 6$, allowing $\mathbb{G}_2$ coordinates to be in $\mathbb{F}_q$ (factor 6 compression).
- The curve parameters have polynomial expressions, allowing fast implementation.
- The curve has a very efficient optimal ate pairing.
- The curve has CM discriminant $-D = -3$, allowing fast GLV multiplication on both $\mathbb{G}_1$ and $\mathbb{G}_2$.
- The curve has fast subgroup checks and fast cofactor multiplication on $\mathbb{G}_1$ and $\mathbb{G}_2$ via endomorphisms.
- The curve has fast and secure hash-to-curve methods for both $\mathbb{G}_1$ and $\mathbb{G}_2$.

# Our work

## Cost estimation of a pairing

$e(P,Q) = f_{t-1,Q}(P)^{(q^6-1)/r}$ $\quad (t-1)$ of 388 bits, $Q \in \mathbb{F}_{q^3}$

$e(P,Q) = (f_{x_0+1,Q}(P) f^q_{x_0^3-x_0^2-x_0,Q}(P))^{(q^6-1)/r}$ $\quad x_0$ of 64 bits, $Q \in \mathbb{F}_q$

$$(q^6-1)/r = \underbrace{(q^3-1)(q+1)}_{\text{easy part}} \underbrace{(q^2-q+1)/r}_{\text{hard part}} = \begin{cases} \text{easy part} \times (w_0 + q w_1) \\ \text{easy part} \times f(x_0, q^i) \end{cases}$$

| Curve | Prime | Pairing | Miller loop | Exponentiation | Total |
|---|---|---|---|---|---|
| BLS12 | 377-bit | ate | 6705 $m_{384}$ | 7063 $m_{384}$ | 13768 $m_{384}$ |
| SW6 | 782-bit | ate | 47298 $m_{832}$ | 10521 $m_{832}$ | 57819 $m_{832}$ |
| This | 761-bit | opt. ate | 7911 $m_{768}$ | 5081 $m_{768}$ | 12992 $m_{768}$ |

$m_b$ base field multiplication, $b$ bitsize in Montgomery domain on a 64-bit platform

x4.5 less operations in a smaller field by one machine-word

# Our work

## C++ implementation timings

Implemented in `libff` library [sl18] (with GMP 6.1.2_2) and tested on a 2.2 GHz Intel Core i7 x86_64 processor with 16 Go 2400 MHz DDR4 memory running macOS Mojave 10.14.6. C++ compiler is clang 10.0.1. Profiling routines use `clock_gettime` and `readproc` calls.

url: `https://github.com/EYBlockchain/zk-swap-libff/tree/ey/libff/algebra/curves/bw6_761`

| Curve | Pairing | Miller loop | Exponentiation | Total | Eq. ?? |
|---|---|---|---|---|---|
| BLS12 | ate | 0.0025s | 0.0049s | 0.0074s | 0.0149s |
| SW6 | ate (proj.) | 0.0388s | 0.0110s | 0.0499s | 0.1274s |
| SW6 | ate (aff.) | 0.0249s | 0.0110s | 0.0361s | 0.0875s |
| This | opt. ate | 0.0053s | 0.0044s | 0.0097s | 0.0203s |

x5 faster to compute a pairing (in projective coordinates)
x6.27 faster to verify a Groth16 proof (in projective coordinates)
x3.7 faster to compute a pairing (in affine coordinates)
x4.22 faster to verify a Groth16 proof (in affine coordinates)

# Applications

## Blockchain projects

- Zexe: user-defined assets, decentralized exchanges and policy-enforcing stablecoins
- Celo: batched verification of BLS signatures
- Filecoin: circuit splitting
- Baseline protocol (EY, Consensys) [ECM20]: batching zkSNARK proofs

# Thank you

- Paper: `https://eprint.iacr.org/2020/351.pdf`
- C++ (*libff*): `https://github.com/EYBlockchain/zk-swap-libff/tree/ey/libff/algebra/curves/bw6_761`
- SageMath and Magma: `https://gitlab.inria.fr/zk-curves/bw6-761/`
- Rust WIP (*zexe*): `https://github.com/scipr-lab/zexe/issues/152`

e-mail: youssef.el.housni@fr.ey.com
twitter: @YoussefElHousn3
telegram: @ElMarroqui

# References I


Sean Bowe, Alessandro Chiesa, Matthew Green, Ian Miers, Pratyush Mishra, and Howard Wu.
Zexe: Enabling decentralized private computation.
In *2020 IEEE Symposium on Security and Privacy (SP)*, pages 1059–1076, Los Alamitos, CA, USA, may 2020. IEEE Computer Society.
https://ia.cr/2018/962.

Eli Ben-Sasson, Alessandro Chiesa, Eran Tromer, and Madars Virza.
Scalable zero knowledge via cycles of elliptic curves.
In Juan A. Garay and Rosario Gennaro, editors, *CRYPTO 2014, Part II*, volume 8617 of *LNCS*, pages 276–294. Springer, Heidelberg, August 2014.

# References II


Alessandro Chiesa, Lynn Chua, and Matthew Weidner.
On cycles of pairing-friendly elliptic curves.
*SIAM Journal on Applied Algebra and Geometry*, 3(2):175–192, 2019.

EY, Consensys, and Microsoft.
Baseline protocol, 2020.
https://github.com/ethereum-oasis/baseline.

Youssef El Housni and Aurore Guillevic.
Optimized and secure pairing-friendly elliptic curves suitable for one layer proof composition.
Cryptology ePrint Archive, Report 2020/351, 2020.
https://eprint.iacr.org/2020/351.

# References III



David Freeman, Michael Scott, and Edlyn Teske.
A taxonomy of pairing-friendly elliptic curves.
*Journal of Cryptology*, 23(2):224–280, April 2010.


Izaak Meckler and Evan Shapiro.
Coda: Decentralized cryptocurrency at scale.
O(1) Labs whitepaper, 2018.
https://cdn.codaprotocol.com/v2/static/coda-whitepaper-05-10-2018-0.pdf.


scipr lab.
libff: C++ library for finite fields and elliptic curves., 2018.
https://github.com/scipr-lab/libff.