Elliptic curves for SNARK and proof systems

Diego F. Aranha$^1$, Youssef El Housni$^2$, Aurore Guillevic$^3$

$^1$Aarhus University, Denmark, $^2$Consensys – Linea, NYC US, $^3$Inria Rennes, France

Journées Numération, Arithmétique, Cryptographie
February 29 – March 1, 2024

https://webusers.imj-prg.fr/~jean-claude.bajard/NAC2024/
Outline

zk-SNARK

Elliptic Curves

Pairings

Pairing-friendly curves

SNARK-friendly curves
Zero-knowledge proofs (ZKP)

Alice
I know the solution to this complex equation

Bob
No idea what the solution is but Alice claims to know it

Challenge Response

• Sound: Alice has a wrong solution $\Rightarrow$ Bob is not convinced.
• Complete: Alice has the solution $\Rightarrow$ Bob is convinced.
• Zero-knowledge: Bob does NOT learn the solution.
Zero-knowledge proofs (ZKP)

Alice

I know the solution to this complex equation

Bob

No idea what the solution is but Alice claims to know it

Sound: Alice has a wrong solution $\implies$ Bob is not convinced.
Zero-knowledge proofs (ZKP)

Alice
I know the solution to this complex equation

Bob
No idea what the solution is but Alice claims to know it

Challenge
Response

• **Sound**: Alice has a **wrong solution** $\implies$ Bob is not convinced.
• **Complete**: Alice has the **solution** $\implies$ Bob is convinced.
Zero-knowledge proofs (ZKP)

**Alice**
I know the solution to this complex equation

**Bob**
No idea what the solution is but Alice claims to know it

- **Sound**: Alice has a wrong solution $\implies$ Bob is not convinced.
- **Complete**: Alice has the solution $\implies$ Bob is convinced.
- **Zero-knowledge**: Bob does NOT learn the solution.
Example: Sigma protocol

Alice

I know $x$ such that $g^x = y$

Bob
Example: Sigma protocol

**Alice**

I know $x$ such that $g^x = y$

$n \leftarrow \mathbb{Z}_r$

**Bob**

$A = g^n$

$n \cdot g^x \cdot c = g^n + x \cdot c$

slide Y. El Housni
Example: Sigma protocol

Alice

I know $x$ such that $g^x = y$

$n \leftarrow \mathbb{Z}_r$

Bob

$A = g^n$

$c \leftarrow \mathbb{Z}_r$

$c \leftarrow \mathbb{Z}_r$
Example: Sigma protocol

I know \( x \) such that \( g^x = y \)

\[
\begin{align*}
n &\leftarrow \mathbb{Z}_r \\
s &= n + c \cdot x \\
A &= g^n \\
c &\leftarrow \mathbb{Z}_r
\end{align*}
\]

slide Y. El Housni
Example: Sigma protocol

Alice

I know $x$ such that $g^x = y$

$n \leftarrow \mathbb{Z}_r$

$s = n + c \cdot x$

Bob

$A = g^n$

$c \leftarrow \mathbb{Z}_r$

$g^s = A \cdot y^c$

with $A \cdot y^c = g^n \cdot g^{x \cdot c}$

then $g^n \cdot g^{x \cdot c} = g^{n+x \cdot c}$
Non-Interactive Zero-Knowledge (NIZK) Sigma protocol

Alice

I know $x$ such that $g^x = y$

$n \leftarrow \mathbb{Z}_r$

$g^n = A$

$c = H(A, y)$

$s = n + c \cdot x$

Bob
Non-Interactive Zero-Knowledge (NIZK) Sigma protocol

**Alice**

I know $x$ such that $g^x = y$

$n \leftarrow \mathbb{Z}_r$

$g^n \cdot A = g^n$

$c = H(A, y)$

$s = n + c \cdot x$

**Bob**

$\pi = (A, c, s)$
Non-Interactive Zero-Knowledge (NIZK) Sigma protocol

**Alice**

I know $x$ such that $g^x = y$

$n \leftarrow \mathbb{Z}_r$

$g^n ; A = g^n$

$c = H(A, y)$

$s = n + c \cdot x$

$\pi = (A, c, s)$

**Bob**

$g^s \overset{?}{=} A \cdot y^c$

$c \overset{?}{=} H(A, y)$
Non-Interactive Zero-Knowledge (NIZK) Sigma protocol

**Alice**

I know $x$ such that $g^x = y$

$n \xleftarrow{\$} \mathbb{Z}_r$

$g^\cdot A = g^n$

Setup

$c = H(A, y)$

$s = n + c \cdot x$

**Bob**

$g^s \overset{?}{=} A \cdot y^c$

$c = H(A, y)$

Proof

$\pi = (A, c, s)$

Verify

slide Y. El Housni
ZKP literature landmarks

- First ZKP work [GMR85]
- Non-Interactive ZKP [BFM88]
- Succinct ZKP [Kil92]
- Succinct Non-Interactive ZKP [Mic94]
ZKP literature landmarks

- First ZKP work [GMR85]
- Non-Interactive ZKP [BFM88]
- Succinct ZKP [Kil92]
- Succinct Non-Interactive ZKP [Mic94]
- Pairing-based succinct NIZK [Gro10]

slide Y. El Housni
ZKP literature landmarks

- First ZKP work [GMR85]
- Non-Interactive ZKP [BFM88]
- Succinct ZKP [Kil92]
- Succinct Non-Interactive ZKP [Mic94]
- Pairing-based succinct NIZK [Gro10]
- “SNARK” terminology and characterization of existence [BCCT12]
- Pairing-based SNARK in quasi-linear prover time [GGPR13]
- Pairing-based SNARK with shortest proof and verifier time [Gro16]
ZKP literature landmarks

- First ZKP work [GMR85]
- Non-Interactive ZKP [BFM88]
- Succinct ZKP [Kil92]
- Succinct Non-Interactive ZKP [Mic94]
- Pairing-based succinct NIZK [Gro10]
- “SNARK” terminology and characterization of existence [BCCT12]
- Pairing-based SNARK in quasi-linear prover time [GGPR13]
- Pairing-based SNARK with shortest proof and verifier time [Gro16]
- SNARK with universal and updatable setup [GKM+18], [MBKM19] (Sonic), [GWC19] (PlonK), [CHM+20] (Marlin), ...
What is a zero-knowledge proof?

“I have a sound, complete and zero-knowledge proof that a statement is true”.

[GMR85]

Sound
False statement $\implies$ cheating prover cannot convince honest verifier.

Complete
True statement $\implies$ honest prover convinces honest verifier.

Zero-knowledge
True statement $\implies$ verifier learns nothing other than statement is true.
zk-SNARK: Zero-Knowledge Succinct Non-interactive ARgument of Knowledge

"I have a *computationally sound, complete, zero-knowledge, succinct, non-interactive* proof that a statement is true and that I know a related secret”.

**Succinct**
A proof is very *short* and *easy* to verify.

**Non-interactive**
No interaction between the prover and verifier for proof generation and verification (except the proof message).

**ARgument of Knowledge**
Honest verifier is convinced that a computationally bounded prover knows a secret information.
Preprocessing zk-SNARK for NP language

$F$: public NP program, $x, z$: public inputs, $w$: private input (witness)

$z := F(x, w)$
Preprocessing zk-SNARK for NP language

\( F \): public NP program, \( x, z \): public inputs, \( w \): private input (witness)
\( z := F(x, w) \)

A zk-SNARK consists of algorithms \( S, P, V \) s.t. for a security parameter \( \lambda \):

\[
\begin{align*}
\text{Setup} & : (pk, vk) \quad \leftarrow \quad S(F, 1^\lambda) \\
\text{Prove} & : \pi \quad \leftarrow \quad P(x, z, w, pk) \\
\text{Verify} & : \text{false/true} \quad \leftarrow \quad V(x, z, \pi, vk)
\end{align*}
\]
Preprocessing zk-SNARK for NP language

\[ F: \text{public NP program, } x, z: \text{public inputs, } w: \text{private input (witness)} \]

\[ z := F(x, w) \]

A zk-SNARK consists of algorithms \( S, P, V \) s.t. for a security parameter \( \lambda \):

\[ \text{Setup: } (pk, vk) \leftarrow S(F, 1^\lambda) \]
\[ \text{Prove: } \pi \leftarrow P(x, z, w, pk) \]
\[ \text{Verify: } \text{false/true} \leftarrow V(x, z, \pi, vk) \]

Anyone
\[ (pk, vk) \leftarrow S(F, 1^\lambda) \]

Alice (prover)
\[ \pi \leftarrow P(x, z, w, pk) \]

Bob (verifier)
\[ \pi \rightarrow V(x, z, \pi, vk)? \]
zk-SNARKs in a nutshell

Main ideas:
zk-SNARKs in a nutshell

Main ideas:

1. Reduce a general statement satisfiability to a polynomial equation satisfiability.
zk-SNARKs in a nutshell

Main ideas:

1. Reduce a general statement satisfiability to a polynomial equation satisfiability.
2. Use Schwartz–Zippel lemma to succinctly verify the polynomial equation with high probability.
zk-SNARKs in a nutshell

Main ideas:

1. Reduce a general statement satisfiability to a polynomial equation satisfiability.
2. Use Schwartz–Zippel lemma to succinctly verify the polynomial equation with high probability.
3. Use homomorphic hiding cryptography to blindly verify the polynomial equation.
zk-SNARKs in a nutshell

Main ideas:

1. Reduce a general statement satisfiability to a polynomial equation satisfiability.
2. Use Schwartz–Zippel lemma to succinctly verify the polynomial equation with high probability.
3. Use homomorphic hiding cryptography to blindly verify the polynomial equation.
4. Make the protocol non-interactive.
Data flow

Statement → Arithmetic circuit → Intermediate representation → Polynomial identities → zk-SNARK proof

(knowledge of a hash function preimage, a discrete log...)

(QAP: Quadratic Arithmetic Program)
Data flow

Statement $\rightarrow$ Arithmetic circuit $\rightarrow$ Intermediate representation $\rightarrow$ Polynomial identities $\rightarrow$ zk-SNARK proof

(knowledge of a hash function preimage, a discrete log...)

Group $\langle g \rangle$ of order $r$, arithmetic over $\mathbb{F}_q$

\[
g^s \overset{?}{=} A \cdot y^c
\]
\[
c \overset{?}{=} H(A, y)
\]
Data flow

Statement $\rightarrow$ Arithmetic circuit $\rightarrow$ Intermediate representation $\rightarrow$ Polynomial identities $\rightarrow$ zk-SNARK proof

(knowledge of a hash function preimage, a discrete log...)

Group $\langle g \rangle$ of order $r$, arithmetic over $\mathbb{F}_q$

Compiler (internal machinery)

$g^s \overset{?}{=} A \cdot y^c$
$c \overset{?}{=} H(A, y)$

(QAP: Quadratic Arithmetic Program)
Data flow

Statement $\rightarrow$ Arithmetic circuit $\rightarrow$ Intermediate representation $\rightarrow$ Polynomial identities $\rightarrow$ zk-SNARK proof

(knowledge of a hash function preimage, a discrete log...)

Group $\langle g \rangle$ of order $r$, arithmetic over $\mathbb{F}_q$

$g^s \equiv A \cdot y^c$

$c \equiv H(A, y)$

(QAP: Quadratic Arithmetic Program)

Group of order $q$, arithmetic over $\mathbb{F}_p$

$\mathbb{F}_q$ in the exponent
Outline

zk-SNARK

Elliptic Curves

Pairings

Pairing-friendly curves

SNARK-friendly curves
Elliptic curves in cryptography

- 1985 (published in 1987) Hendrik Lenstra Jr., Elliptic Curve Method (ECM) for integer factoring
- 1985, Koblitz, Miller: Elliptic Curves over a finite field form a group suitable for Diffie–Hellman key exchange
- 1985, Certicom: company owning patents on ECC
Elliptic curves in cryptography

- 1985 (published in 1987) Hendrik Lenstra Jr., Elliptic Curve Method (ECM) for integer factoring
- 1985, Koblitz, Miller: Elliptic Curves over a finite field form a group suitable for Diffie–Hellman key exchange
- 1985, Certicom: company owning patents on ECC
- 2000 Elliptic curves in IEEE P1363 standard
- 2000 Bilinear pairings over elliptic curves
- NSA cipher suite B, elliptic curves for public-key crypto
Elliptic curves in cryptography

- 1985 (published in 1987) Hendrik Lenstra Jr., Elliptic Curve Method (ECM) for integer factoring
- 1985, Koblitz, Miller: Elliptic Curves over a finite field form a group suitable for Diffie–Hellman key exchange
- 1985, Certicom: company owning patents on ECC
- 2000 Elliptic curves in IEEE P1363 standard
- 2000 Bilinear pairings over elliptic curves
- NSA cipher suite B, elliptic curves for public-key crypto
- 2014: Quasi-polynomial-time algorithm for discrete log computation in GF($2^n$), GF($3^m$)
  No more pairings on elliptic curves over these fields
- 2015: Tower Number Field Sieve in GF($p^n$)
  Pairing-friendly curves should have larger key sizes
Elliptic curves in cryptography

- 1985 (published in 1987) Hendrik Lenstra Jr., Elliptic Curve Method (ECM) for integer factoring
- 1985, Koblitz, Miller: Elliptic Curves over a finite field form a group suitable for Diffie–Hellman key exchange
- 1985, Certicom: company owning patents on ECC
- 2000 Elliptic curves in IEEE P1363 standard
- 2000 Bilinear pairings over elliptic curves
- NSA cipher suite B, elliptic curves for public-key crypto
- 2014: Quasi-polynomial-time algorithm for discrete log computation in $\text{GF}(2^n)$, $\text{GF}(3^m)$
  No more pairings on elliptic curves over these fields
- 2015: Tower Number Field Sieve in $\text{GF}(p^n)$
  Pairing-friendly curves should have larger key sizes
- 2016: NIST Post-Quantum competition
  Isogenies on elliptic curves
Examples of elliptic curves

\[ y^2 = x^3 - 2 \]
\[ y^2 = x^3 - 4x + 4 \]
\[ y^2 = x^3 - 6x + 4 \]
Chord and tangent rule

\[ \ell_{P,Q}(x, y) \]

\[ P(x_1, y_1), \; Q(x_2, y_2), \; x_1 \neq x_2 \]

slope \( \lambda = \frac{\Delta y}{\Delta x} = \frac{y_2 - y_1}{x_2 - x_1} \)

line \( L \) through \( P \) and \( Q \) has equation

\[ L: y = \lambda(x - x_1) + y_1 \]

\( P, Q, R \in L \cap E \)
Elliptic curves over finite fields

$E/\mathbb{F}_{17}: y^2 = x^3 + x + 7$
Elliptic curves over finite fields

$E/\mathbb{F}_{17} : y^2 = x^3 + x + 7$
Elliptic curves over finite fields

$E/\mathbb{F}_{17} : y^2 = x^3 + x + 7$
Elliptic curves over finite fields

\[ E/\mathbb{F}_{17} : y^2 = x^3 + x + 7 \]
Elliptic curves over finite fields

\[ E/\mathbb{F}_{17} : y^2 = x^3 + x + 7 \]
Elliptic curves over finite fields

\[ E_{\mathbb{F}_{17}}: y^2 = x^3 + x + 7 \]

\[ P + Q \]
Outline

zk-SNARK

Elliptic Curves

Pairings

Pairing-friendly curves

SNARK-friendly curves
What is a pairing?

$(G_1, +), (G_2, +), (G_T, \cdot)$ three cyclic groups of large prime order $n$

Pairing: map $e : G_1 \times G_2 \rightarrow G_T$

1. bilinear: $e(P_1 + P_2, Q) = e(P_1, Q) \cdot e(P_2, Q)$, $e(P, Q_1 + Q_2) = e(P, Q_1) \cdot e(P, Q_2)$

2. non-degenerate: $e(G_1, G_2) \neq 1$ for $\langle G_1 \rangle = G_1$, $\langle G_2 \rangle = G_2$

3. efficiently computable.

Most often used in practice:

$$e([a]P, [b]Q) = e([b]P, [a]Q) = e(P, Q)^{ab}.$$ 

$\sim$ Many applications in asymmetric cryptography.
Pairing setting: elliptic curves

\[ E/\mathbb{F}_p : y^2 = x^3 + ax + b, \ a, b \in \mathbb{F}_p, \ p \geq 5 \]

- proposed in 1985 by Koblitz, Miller
- \( E(\mathbb{F}_p) \) has an efficient group law (chord and tangent rule) \( \rightarrow \ G_1 \)
- \#\( E(\mathbb{F}_p) = p + 1 - t \), trace \( t \): \( |t| \leq 2\sqrt{p} \)
- efficient group order computation (point counting)
Pairing setting: elliptic curves

\[ E/F_p : y^2 = x^3 + ax + b, \ a, b \in F_p, \ p \geq 5 \]

- proposed in 1985 by Koblitz, Miller
- \( E(F_p) \) has an efficient group law (chord an tangent rule) \( \rightarrow G_1 \)
- \#E(F_p) = p + 1 - t, trace \( t: |t| \leq 2\sqrt{p} \)
- efficient group order computation (point counting)
- large subgroup of prime order \( n \) s.t. \( n \mid p + 1 - t \) and \( n \) coprime to \( p \)
- \( E(F_p)[n] = \{ P \in E(F_p) : [n]P = O \} \) has order \( n \)
- \( E[n] \sim \mathbb{Z}/n\mathbb{Z} \times \mathbb{Z}/n\mathbb{Z} \) (for crypto)
- only generic attacks against DLP on well-chosen genus 1 and genus 2 curves
- optimal parameter sizes
Tate pairing

From its definition to its efficient implementation

• John Tate, 1958
• Stephen Lichtenbaum, 1969
• Victor Miller, 1986, Miller algorithm for $f_P$
• Frey–Rück, 1994: the MOV attack with the Tate pairing instead of the Weil pairing
• Harasawa, Shikata, Suzuki, Imai, 1999, 161467 s (112 days)
  163-bit supersingular curve, $G_T \subset \mathbb{F}_{p^2}$ of 326 bits.
• Antoine Joux, 2000: how to compute Miller algorithm more efficiently
  1 s on a supersingular 528-bit curve, $G_T \subset \mathbb{F}_{p^2}$ of 1055 bits
Cryptographic pairing

Modified Weil or Tate pairing on an elliptic curve
Discrete logarithm problem with one more dimension.

\[ e : E(\mathbb{F}_p)[n] \times E(\mathbb{F}_{p^k})[n] \to \mathbb{F}_{p^k}^*, \quad e([a]P, [b]Q) = e(P, Q)^{ab} \]
Cryptographic pairing

Modified Weil or Tate pairing on an elliptic curve
Discrete logarithm problem with one more dimension.

\[ e : E(\mathbb{F}_p)[n] \times E(\mathbb{F}_{p^k})[n] \rightarrow \mathbb{F}_{p^k}^\ast, \quad e([a]P, [b]Q) = e(P, Q)^{ab} \]

Attacks
Cryptographic pairing

Modified Weil or Tate pairing on an elliptic curve
Discrete logarithm problem with one more dimension.

\[ e : E(\mathbb{F}_p)[n] \times E(\mathbb{F}_{p^k})[n] \rightarrow \mathbb{F}_{p^k}^*, \quad e([a]P, [b]Q) = e(P, Q)^{ab} \]

Attacks

• inversion of \( e \): hard problem (exponential)
Cryptographic pairing

Modified Weil or Tate pairing on an elliptic curve
Discrete logarithm problem with one more dimension.

\[ e : E(\mathbb{F}_p)[n] \times E(\mathbb{F}_{p^k})[n] \longrightarrow \mathbb{F}_{p^k}^* \]
\[ e([a]P, [b]Q) = e(P, Q)^{ab} \]

Attacks
- inversion of \( e \) : hard problem (exponential)
- discrete logarithm computation in \( E(\mathbb{F}_p) \) : hard problem (exponential, in \( O(\sqrt{n}) \))
Cryptographic pairing

Modified Weil or Tate pairing on an elliptic curve
Discrete logarithm problem with one more dimension.

\[ e : E(\mathbb{F}_p)[n] \times E(\mathbb{F}_{p^k})[n] \rightarrow \mathbb{F}_{p^k}^*, \quad e([a]P, [b]Q) = e(P, Q)^{ab} \]

Attacks
- inversion of \( e \): hard problem (exponential)
- discrete logarithm computation in \( E(\mathbb{F}_p) \): hard problem (exponential, in \( O(\sqrt{n}) \))
- discrete logarithm computation in \( \mathbb{F}_{p^k}^* \): easier, subexponential \( \rightarrow \) take a large enough field
Jens Groth’s proof composition [Gro16]

Given an instance \( \Phi = (a_0, \ldots, a_\ell) \in \mathbb{R}^\ell \) of a public NP program \( F \)

- \((pk, vk) \leftarrow S(F, \tau, 1^\lambda)\) where

\[ vk = (vk_{\alpha, \beta}, \{vk_{\pi_i}\}_{i=0}^\ell, vk_{\gamma}, vk_\delta) \in G_T \times G_1^{\ell+1} \times G_2 \times G_2 \]

- \(\pi \leftarrow P(\Phi, w, pk)\) where

\[ \pi = (A, B, C) \in G_1 \times G_2 \times G_1 \quad (O_\lambda(1)) \]

- \(0/1 \leftarrow V(\Phi, \pi, vk)\) where \(V\) is

\[ e(A, B) = vk_{\alpha, \beta} \cdot e(vk_x, vk_{\gamma}) \cdot e(C, vk_\delta) \quad (O_\lambda(|\Phi|)) \quad (1) \]

and \(vk_x = \sum_{i=0}^\ell [a_i]vk_{\pi_i}\) depends only on the instance \(\Phi\) and \(vk_{\alpha, \beta} = e(vk_\alpha, vk_\beta)\) can be computed in the trusted setup for \((vk_\alpha, vk_\beta) \in G_1 \times G_2.\)
Applications not in cryptocurrencies

ZK Microphone: Trusted audio in the age of deepfakes
https://ethglobal.com/showcase/zk-microphone-8161v
Proving sound authenticity

Using ZK Proofs to Fight Disinformation
Proving image authenticity

A Tool for Proving Software Vulnerabilities in Zero Knowledge
https://galois.com/blog/2024/02/
introducing-cheesecloth-a-tool-for-proving-software-vulnerabilities-in-zero-knowledge/
Outline

zk-SNARK

Elliptic Curves

Pairings

Pairing-friendly curves

SNARK-friendly curves
First ordinary pairing-friendly curves: MNT

Miyaji, Nakabayashi, Takano, \#\(E(\mathbb{F}_p) = p(u) + 1 - t(u) = q(u)\)

\[
\begin{align*}
k = 3 \quad & \left\{ 
\begin{array}{l}
t(u) = -1 \pm 6u \\
q(u) = 12u^2 \mp 6u + 1 \\
p(u) = 12u^2 - 1 \\
Dy^2 = 12u^2 \pm 12u - 5
\end{array} \right.
\end{align*}
\]

\[
\begin{align*}
k = 4 \quad & \left\{ 
\begin{array}{l}
t(u) = -u, \ u + 1 \\
q(u) = u^2 + 2u + 2, \ u^2 + 1 \\
p(u) = u^2 + u + 1 \\
Dy^2 = 3u^2 + 4u + 4
\end{array} \right.
\end{align*}
\]

\[
\begin{align*}
k = 6 \quad & \left\{ 
\begin{array}{l}
t(u) = 1 \pm 2u \\
q(u) = 4u^2 \mp 2u + 1 \\
p(u) = 4u^2 + 1 \\
Dy^2 = 12u^2 - 4u + 3
\end{array} \right.
\end{align*}
\]

CODA [MS18]:

\(k = 6\), 753 bits, \(E_6 \approx 137\) bits of security, \(D = -241873351932854907\), seed \(u = 0\times a3a58eb20d1fec36e5e772ee6d3ff28c296465f137300399db8a5521e18d33581a262716214583d3b89820dd0c000\)

\(k = 4\), 753 bits, \(E_4 \approx 113\) bits of security
Cycle of curves: unlimited chains of SNARKs [BCTV14]

- Elliptic curve $E_0(\mathbb{F}_q)$ of prime order $p$ over a field $\mathbb{F}_q$
- Statement in a group of prime order $p$ over a field $\mathbb{F}_q$
- Elliptic curve $E_1(\mathbb{F}_p)$ of prime order $q$ over a field $\mathbb{F}_p$
- Statement in a group of prime order $q$ over a field $\mathbb{F}_p$
MNT-4 and MNT-6 curves form a cycle

\[ k = 4, \text{ MNT-4 parameters} \quad t_4 = -v, \quad q_4 = v^2 + 1, \quad p_4 = v^2 + v + 1 \]

\[ k = 6, \text{ MNT-6 parameters} \quad t_6 = 1 - 2u, \quad q_6 = 4u^2 + 2u + 1, \quad p_6 = 4u^2 + 1 \]

\[ q_4 = p_6 \quad \text{and} \quad v = 2u \quad \text{and} \quad p_4 = q_6 \quad q_4, q_6 \text{ are primes} \]

Unique known cycle of pairing-friendly curves.

Impossibility results:

- **Alessandro Chiesa, Lynn Chua, and Matthew Weidner.**
  On cycles of pairing-friendly elliptic curves.

- **Marta Bellés-Muñoz, Jorge Jiménez Urroz, and Javier Silva.**
  Revisiting cycles of pairing-friendly elliptic curves.
Very popular pairing-friendly curves: Barreto-Naehrig (BN)

\[ E_{BN} : \quad y^2 = x^3 + b, \quad p \equiv 1 \mod 3, \quad D = -3 \text{ (ordinary)} \]

\[
\begin{align*}
p &= 36x^4 + 36x^3 + 24x^2 + 6x + 1 \\
t &= 6x^2 + 1 \\
q &= p + 1 - t = 36x^4 + 36x^3 + 18x^2 + 6x + 1 \\
t^2 - 4p &= -3(6x^2 + 4x + 1)^2 \rightarrow \text{no CM method needed}
\end{align*}
\]

Comes from the Aurifeuilllean factorization of \( \Phi_{12} : \)
\[ \Phi_{12}(6x^2) = q(x)q(-x) \]

<table>
<thead>
<tr>
<th>Security level</th>
<th>( \log_2 q )</th>
<th>finite field</th>
<th>( k )</th>
<th>( \log_2 p )</th>
<th>deg ( P, \ p = P(u) )</th>
<th>( \rho )</th>
</tr>
</thead>
<tbody>
<tr>
<td>102</td>
<td>256</td>
<td>3072</td>
<td>12</td>
<td>256</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>123</td>
<td>384</td>
<td>4608</td>
<td>12</td>
<td>384</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>132</td>
<td>448</td>
<td>5376</td>
<td>12</td>
<td>448</td>
<td>4</td>
<td>1</td>
</tr>
</tbody>
</table>

Formerly BN-254 in Ethereum with seed 0x44e992b44a6909f1
Barreto, Lynn, Scott method. Becomes more and more popular, replacing BN curves

\[ E_{\text{BLS}} : \ y^2 = x^3 + b, \ p \equiv 1 \mod 3, \ D = -3 \ (\text{ordinary}) \]

\[
\begin{align*}
p &= (u - 1)^2 / 3(u^4 - u^2 + 1) + u \\
t &= u + 1 \\
q &= (u^4 - u^2 + 1) = \Phi_{12}(u) \\
p + 1 - t &= (u - 1)^2 / 3(u^4 - u^2 + 1) \\
\quad \quad \quad \text{cofactor} \\
t^2 - 4p &= -3y(u)^2 \rightarrow \text{no CM method needed}
\end{align*}
\]

BLS12-381 (Zcash \[\text{Bow17}\]) with seed \(-0xd201000000010000\)
BLS12-377 (Zexe \[\text{BCG}^+18\]) with seed \(0x8508c000000000001\)
Outline

zk-SNARK

Elliptic Curves

Pairings

Pairing-friendly curves

SNARK-friendly curves
CØCØ embedded curve: Kosba et al. construction [KZM
+15]

- **Elliptic curve** $E_0(\mathbb{F}_r)$ of order $4s$
- **Statement** in a group of prime order $s$ over a field $\mathbb{F}_r$
- **Arithmetisation**
- **Polynomials** in $\mathbb{F}_r[X]$
- **Proof of the circuit**
- **SNARK** with a pairing $e: G_1 \times G_2 \rightarrow G_T$
  \[ \#G_i = r \]

**CØCØ**: given $r$, search for a curve $E_0$ over $\mathbb{F}_r$ of order 4 times a prime
Embedded SNARK-friendly curves

Usually a twist-secure elliptic curve in Montgomery or (twisted) Edwards form

Input: field $\mathbb{F}_p$
Output: an embedded curve of order $4s$ or $8s$ with prime $s$
Procedure: Increment the curve coefficient(s) until a suitable curve is found

CØCØ [KZM⁺15] with BN-254a,
JubJub [ZCa21] or Bandersnatch [MSZ21] with BLS12-381, ...
2-chains of elliptic curves

Given $p$, search for a pairing-friendly curve $E_1$ of order $h \cdot p$ over a field $\mathbb{F}_q$
Geppetto construction [CFH⁺15]

SNARK-1 with a pairing $e : G_1 \times G_2 \rightarrow G_T$

polynomials in $\mathbb{F}_p[X]$ of prime order $r$

arithmetisation

proof of the circuit

SNARK-2 with a pairing $e : G'_1 \times G'_2 \rightarrow G'_T$

# $G_i = r$

pairing-friendly BN curve $E_1(\mathbb{F}_p)$ of prime order $r$

pairing-friendly BW curve $E_2(\mathbb{F}_q)$ of order $h \times p$

Geppetto: given $p$, search for a pairing-friendly curve BW6 (Brezing–Weng) of order $h \cdot p$ over a field $\mathbb{F}_q$
2-chains of pairing-friendly curves

- Geppetto [CFH\textsuperscript{+}15]: BN254b + BW6-509
- Zexe [BCG\textsuperscript{+}18]: BLS12-377 + CP6-782
- BLS12-377 + BW6-761 [EHG20] for Gorth\textsubscript{16}
- BLS24-315 + BW6-633 [EHG22] For KZG / universal SNARK
More plain/hybrid cycles of curves

Plain cycles: 2 plain prime-order elliptic curves (no pairing)

- HALO: Tweedledum/tweedledee curves [https://github.com/daira/tweedle](https://github.com/daira/tweedle)
- HALO2: Pallas-Vesta – Pasta curves [https://github.com/zcash/pasta_curves](https://github.com/zcash/pasta_curves)

Hybrid cycles: a plain curve and a BN pairing-friendly curve, both prime order

- BN254-Grumpkin [https://hackmd.io/@aztec-network/ByzgNxBfd](https://hackmd.io/@aztec-network/ByzgNxBfd)
- Pluto (BN446) - Eris [https://github.com/daira/pluto-eris/](https://github.com/daira/pluto-eris/)
## Conclusion

<table>
<thead>
<tr>
<th>Statement embedded curve</th>
<th>SNARK 1 inner curve</th>
<th>SNARK 2 outer curve</th>
</tr>
</thead>
<tbody>
<tr>
<td>CØCØ [KZM⁺15]</td>
<td>BN254a Ethereum</td>
<td></td>
</tr>
<tr>
<td>$E₀$</td>
<td>BN254b</td>
<td>BW6-509 Geppetto [CFH⁺15]</td>
</tr>
<tr>
<td>$E₀'$</td>
<td>BLS12-377 [BCG⁺18]</td>
<td>CP6-782 [BCG⁺18]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BW6-761 [EHG20]</td>
</tr>
<tr>
<td>$E₀''$</td>
<td>BLS24-315</td>
<td>BW6-633 [EHG22]</td>
</tr>
</tbody>
</table>

Survey paper [AEHG23]

 Diego F. Aranha, Youssef El Housni, and Aurore Guillevic.  
A survey of elliptic curves for proof systems.  
Félicitations Jean-Claude et bonne retraite bientôt en Bretagne !
References

Diego F. Aranha, Youssef El Housni, and Aurore Guillevic.
A survey of elliptic curves for proof systems.
Special Issue: Mathematics of Zero-Knowledge.

Nir Bitansky, Ran Canetti, Alessandro Chiesa, and Eran Tromer.
From extractable collision resistance to succinct non-interactive arguments of knowledge, and back again.

Sean Bowe, Alessandro Chiesa, Matthew Green, Ian Miers, Pratyush Mishra, and Howard Wu.
Zexe: Enabling decentralized private computation.

Eli Ben-Sasson, Alessandro Chiesa, Eran Tromer, and Madars Virza.
Scalable zero knowledge via cycles of elliptic curves.

Manuel Blum, Paul Feldman, and Silvio Micali.
Non-interactive zero-knowledge and its applications (extended abstract).

Marta Bellés-Muñoz, Jorge Jiménez Urroz, and Javier Silva.
Revisiting cycles of pairing-friendly elliptic curves.
References II

Sean Bowe.
BLS12-381: New zk-SNARK elliptic curve construction.
https://electriccoin.co/blog/new-snark-curve/.

Alessandro Chiesa, Lynn Chua, and Matthew Weidner.
On cycles of pairing-friendly elliptic curves.

Geppetto: Versatile verifiable computation.
ePrint 2014/976.

Alessandro Chiesa, Yuncong Hu, Mary Maller, Pratyush Mishra, Psi Vesely, and Nicholas P. Ward.
Marlin: Preprocessing zkSNARKs with universal and updatable SRS.

Youssef El Housni and Aurore Guillevic.
Optimized and secure pairing-friendly elliptic curves suitable for one layer proof composition.
Youssef El Housni and Aurore Guillevic.
Families of SNARK-friendly 2-chains of elliptic curves.

Rosario Gennaro, Craig Gentry, Bryan Parno, and Mariana Raykova.
Quadratic span programs and succinct NIZKs without PCPs.

Jens Groth, Markulf Kohlweiss, Mary Maller, Sarah Meiklejohn, and Ian Miers.
Updatable and universal common reference strings with applications to zk-SNARKs.

Shafi Goldwasser, Silvio Micali, and Charles Rackoff.
The knowledge complexity of interactive proof-systems (extended abstract).

Jens Groth.
Short pairing-based non-interactive zero-knowledge arguments.

Jens Groth.
On the size of pairing-based non-interactive arguments.
Ariel Gabizon, Zachary J. Williamson, and Oana Ciobotaru.
PLONK: Permutations over lagrange-bases for oecumenical noninteractive arguments of knowledge.

Joe Kilian.
A note on efficient zero-knowledge proofs and arguments (extended abstract).

Ahmed Kosba, Zhichao Zhao, Andrew Miller, Yi Qian, Hubert Chan, Charalampos Papamanthou, Rafael Pass, abhi shelat, and Elaine Shi.
C∅c∅: A framework for building composable zero-knowledge proofs.

Mary Maller, Sean Bowe, Markulf Kohlweiss, and Sarah Meiklejohn.
Sonic: Zero-knowledge SNARKs from linear-size universal and updatable structured reference strings.

Silvio Micali.
CS proofs (extended abstracts).

Izaak Meckler and Evan Shapiro.
Coda: Decentralized cryptocurrency at scale.
O(1) Labs whitepaper, 2018.
Bandersnatch: a fast elliptic curve built over the BLS12-381 scalar field.

ZCash.
What is jubjub?