Fast cube roots in $\mathbb{F}_{p^2}$ via the algebraic torus

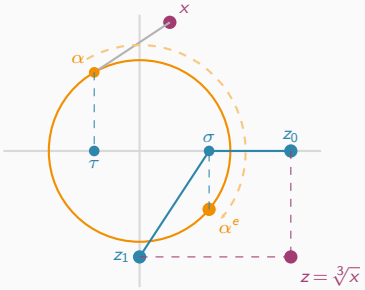
Project, Rotate, Recover

Youssef El Housni

Consensys, New York, USA

25 November 2026

CANS – Wollongong, Australia



Why $\mathbb{F}_{p^2}$ cube roots?

Applications of $\mathbb{F}_{p^2}$ cube roots

- **Hash-to-curve** — Icart; Koshelev for $y^2 = x^3 + b$ and $\mathbb{G}_2$ [Ica09, Kos24]
- **Point decompression** on $j = 0$ curves, two at a time [Kos25]
- **Algebraic VDF analysis** [BFH+24]
- **Depressed cubics** / Cardano [EB26]
- **3-isogeny walks** [COR25] — one cube root *per step*

In the first four it is *one* operation among many.

In the last, it **is** the computation.

Two people said this out loud

Koshelev [Kos24], on hashing to $\mathbb{G}_2$:

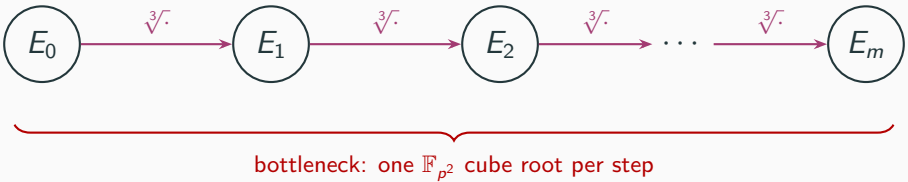
"In contrast to a square $\mathbb{F}_{q^2}$-root, **the author does not know how to express $\sqrt[3]{\cdot} \in \mathbb{F}_{q^2}$ through a few $\mathbb{F}_q$-roots**."

Biryukov et al. [BFH+24], on countermeasures for Sloth++:

"While taking square roots over $\mathbb{F}_{p^2}$ reduces to ... taking square roots over $\mathbb{F}_p$, **this doesn't appear to be the case for cube roots or higher**."

This talk: that reduction — and it is also *faster*.

3-isogeny walk



- Radical 3-isogenies [CDV20]: each step *is* a cube root plus a few field operations
- $\mathrm{CGLHash}_3$ / VDFs: public walk; QFESTA [NO24], CSIDH: secret-dependent walk

$\Rightarrow$ the walk costs $m\times$ (one $\mathbb{F}_{p^2}$ cube root). Nothing else matters.

What is done in practice

Everyone computes $z = x^e$ **directly in** $\mathbb{F}_{p^2}$: exponent read off $p^2 \bmod 9$, about $2\log_2 p$ bits.

- Reference isogeny code [COPR25]: binary square-and-multiply, hardcoded **759-bit exponent**
- Free and unused: a short **addition chain** $\Rightarrow$ our *Direct* baseline
- Free and unused: **Frobenius splitting** $x^e = x^{e_0}\overline{x}^{e_1}$ $\Rightarrow$ our *Frob.* baselines

All of them stay **inside** $\mathbb{F}_{p^2}$. We want to leave.

This talk

1. Why $\mathbb{F}_{p^2}$ cube roots?
2. Roots in $\mathbb{F}_p$
3. From $\mathbb{F}_p$ to $\mathbb{F}_{p^2}$
4. Our algorithm
5. Benchmarks
6. Constant-time & conclusion

Roots in $\mathbb{F}_p$

Square root in $\mathbb{F}_p$

- Simple cases via direct exponentiation:

  $p \equiv 3 \pmod 4$: $\sqrt{a} = a^{(p+1)/4}$, $p \equiv 5 \pmod 8$: Atkin's formula, ...

- General($v_2(p-1) \geq 3$): **Tonelli–Shanks**

  Exponentiation + discrete-log correction in 2-Sylow subgroup

- Cost $\approx 1.5 \log_2(p)\ \mathbf{M}_p$ (via square-and-multiply)
- Cost $\approx 1.2 \log_2(p)\ \mathbf{M}_p$ (via addition chain)

Cube root in $\mathbb{F}_p$

For a nonzero cubic residue a:

Prime congruence	Root or initial candidate m_0
$p \equiv 2 \pmod 3$	$a^{(2p-1)/3}$ (unique root)
$p \equiv 4 \pmod 9$	$a^{(2p+1)/9}$
$p \equiv 7 \pmod 9$	$a^{(p+2)/9}$
$p \equiv 10 \pmod{27}$	$a^{(2p+7)/27}$
$p \equiv 19 \pmod{27}$	$a^{(p+8)/27}$

- Last two cases: $m_0^3 = a\omega^j$, with $\omega^3 = 1$; correct using a **ninth** root of unity.
- If $v_3(p-1) \geq 3$: Adleman–Manders–Miller [AMM77], with a discrete-log correction in the 3-Sylow subgroup.

Square root vs. cube root in $\mathbb{F}_p$

v		**Square root** ($v_2(p-1)$)	**Cube root** ($v_3(p-1)$)
1	Formula	1 exp	1 exp
2	Formula	1 exp + root-of-unity fix	1 exp + ninth-root fix
≥ 3	Algorithm	Tonelli–Shanks	AMM [AMM77]

Always: **one exponentiation**, then a **discrete log in the ℓ-Sylow**.

All six primes here have $v_3(p-1) \leq 2 \quad \Rightarrow \quad$ no AMM loop.

From $\mathbb{F}_p$ to $\mathbb{F}_{p^2}$

$\mathbb{F}_{p^2}$ arithmetic recap

$$\mathbb{F}_{p^2} = \mathbb{F}_p[u]/(u^2 - \beta), \quad \beta = -c \text{ non-square}, \quad \mathsf{N}(x) = x_0^2 + c x_1^2$$

- Conjugation: $\overline{x_0 + x_1 u} = x_0 - x_1 u$
- Frobenius: $x^p = \bar{x}$
- $\mathbf{M}_{p^2} = 3\,\mathbf{M}_p$ (Karatsuba)
- $\mathbf{S}_{p^2} = 2\,\mathbf{M}_p$ (Complex)

$\sqrt{\cdot}$ in $\mathbb{F}_{p^2}$ reduces to $\mathbb{F}_p$ — and why

Which powers of x land in $\mathbb{F}_p$?

$$x^k \in \mathbb{F}_p \iff (p+1) \mid k \qquad \Longrightarrow \qquad \text{smallest: } x^{p+1} = \mathsf{N}(x)$$

And $p+1$ is even, so the half-power is still an integer power:

$$b = x^{(p+1)/2}: \qquad b^2 = \mathsf{N}(x) \in \mathbb{F}_p, \qquad b^{p-1} = \pm 1 \implies \boxed{b \in \mathbb{F}_p}$$



$\sqrt[3]{\cdot}$ in $\mathbb{F}_{p^2}$: what exactly goes wrong

Not the problem: $\sqrt[3]{\mathrm{N}(x)}$ **is** in $\mathbb{F}_p$.

1. No "third of the norm": we want $x^{(p+1)/3}$, but $3 \nmid (p+1)$.

$$\underbrace{2 \mid (p-1) \ \textbf{and} \ 2 \mid (p+1)}_{\text{halving works on } \textit{both} \text{ sides}} \qquad \text{versus} \qquad \underbrace{3 \mid (p-1) \ \textbf{only}}_{\text{torus side untouched}}$$

2. The norm loses a dimension: eliminating z_1 from $z^3 = x$ leaves a **cubic over** $\mathbb{F}_p$ — and Cardano then asks for a cube root in $\mathbb{F}_{p^2}$.

Circular — the *casus irreducibilis*.

Our algorithm

The obvious idea: split along $p^2 - 1 = (p-1)(p+1)$

$$\underbrace{\mathbb{T}_2(\mathbb{F}_p)}_{\text{order } p+1} \qquad \text{and} \qquad \underbrace{\mathsf{N}(\mathbb{F}_{p^2}^*)}_{\text{order } p-1}$$

A thoroughly classical move:

- Pohlig–Hellman [PH78] for DL; AMM [AMM77] for roots; CRT for RSA
- LUC [SS95, BBL95], XTR [LV00], Rubin–Silverberg [RS03]
- $\mathbb{F}_{p^2}$ *square* roots already reduce this way [AR14, Sco20, AAA+25]

So why has nobody done it for cube roots?

Because the naive split does not beat it

Step		$\mathbb{F}_p$ cost	
1. **Project**	$\tau = \mathrm{Tr}(x^{1-p})$	1 **inv**	$1.3\,\ell$
2. **Rotate**	$\sigma = V_e(\tau)$	1 exp	$1.8\,\ell$
3. **Recover**	$m = \sqrt[3]{n}$	1 exp	$1.0\,\ell$
	two divisions, batched	$+1$ **inv**	$1.3\,\ell$
Total	four $\mathbb{F}_p$ exponentiations	$\approx$ **5.4** ℓ $\mathbf{M}_p$	

Direct: one $\mathbb{F}_{p^2}$ exponentiation, $\approx 2\ell$ bits $\implies$ $\approx$ **5.3** ℓ $\mathbf{M}_p$.

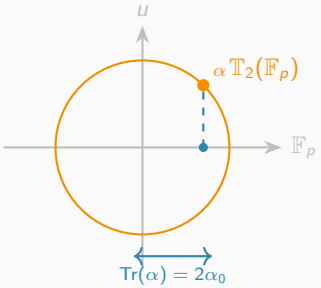
A wash — so why bother?

Those four exponentiations are *not independent*: **fuse them**.

The algebraic torus $\mathbb{T}_2(\mathbb{F}_p)$

$$\mathbb{T}_2(\mathbb{F}_p) = \{\alpha \in \mathbb{F}_{p^2}^* : \mathsf{N}(\alpha) = 1\}, \qquad |\mathbb{T}_2| = p + 1$$

- $\alpha^p = \bar{\alpha} = \alpha^{-1}$
- $\mathsf{Tr}(\alpha) = 2\alpha_0 \in \mathbb{F}_p$ determines α up to inversion



Key observation

1. $x^{1-p} \in \mathbb{T}_2(\mathbb{F}_p)$ (since $\mathsf{N}(x^{1-p}) = 1$)
2. $\gcd(3, p+1) = 1$ for $p \equiv 1 \pmod 3$

   $\Rightarrow$ **cube map is a bijection** on $\mathbb{T}_2(\mathbb{F}_p)$
3. Cube root on torus $=$ exponentiation by $e = 3^{-1} \bmod (p+1)$
4. **Connection to Lucas V-sequence:**

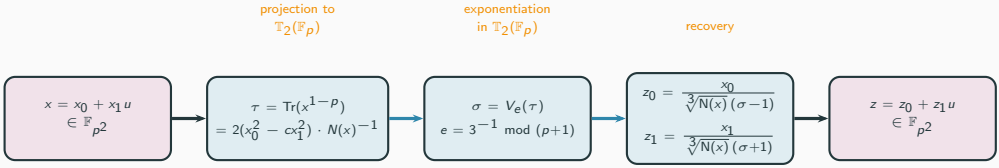
   For $\alpha \in \mathbb{T}_2$ with $\tau = \mathsf{Tr}(\alpha) \in \mathbb{F}_p$,

   $$\mathsf{Tr}(\alpha^e) = V_e(\tau) \quad \text{(Lucas identity)}$$

   $\Rightarrow$ compute the trace of α^e via $V_e(\tau)$ **entirely in** $\mathbb{F}_p$

$$\sigma = V_e(\tau), \quad \tau = \mathsf{Tr}(\alpha), \quad e = 3^{-1} \bmod (p+1)$$

Algorithm overview: before fusion



Lucas V-sequence on the torus



$$V_0 = 2, \quad V_1 = \tau$$

$$V_{2k} = V_k^2 - 2, \qquad V_{2k+1} = V_k \cdot V_{k+1} - \tau$$

- Cost: $1.8\,\mathbf{M}_p$ per bit $\quad (1\,\mathbf{M}_p + 1\,\mathbf{S}_p$, with $\mathbf{S}_p = 0.8\,\mathbf{M}_p)$
- Fixed public exponent: input-independent operation sequence
- Total: $\approx 1.8\,\ell\,\mathbf{M}_p$ where $\ell = \log_2(p)$

$1.8\,\mathbf{M}_p$ **per bit over** ℓ **bits** — against the direct method's $\sim 2.2\,\mathbf{M}_p$ per step over a ladder **twice as long**.

Fusion 1: the cube root and the inverse share a chain

Both are exponentiations **of the same base** n, over the same long chain $t = n^h$:

$$m_0 = n^{1+sh}, \qquad n^{-1} = n^{p-2}, \qquad h = \tfrac{p-r}{d}, \quad \chi = n\,t^s.$$

Proposition. $a = r - 2, \quad b = d - as \implies \chi^a\, t^b = n^{-1}.$

p	(a, b)	p	(a, b)
7 mod 9	$(5, 4)$	10 mod 27	$(8, 11)$
4 mod 9	$(2, 5)$	19 mod 27	$(17, 10)$

One chain ($\approx 1.0\,\ell\,\mathbf{M}_p$), **two outputs**.

The inverse drops from $\approx 1.3\,\ell\,\mathbf{M}_p$ to $\leq 8.6\,\mathbf{M}_p$.

Main path, v1 — one fusion

	Step	Cost
1.	$n \leftarrow x_0^2 + c\,x_1^2$	—
2.	$(m, n^{-1}) \leftarrow \texttt{FusedExp}(n)$	$1.0\,\ell$
3.	$\tau \leftarrow 2(x_0^2 - c\,x_1^2)n^{-1}$	$1\,\mathbf{M}_p$
4.	$\sigma \leftarrow V_e(\tau)$	$1.8\,\ell$
5.	z_0, z_1 from m, σ	$1.3\,\ell$ (1 **inv**)
6.	verify $z^3 = x$	$5\,\mathbf{M}_p$
	Total	$\approx$ **4.1** ℓ $\mathbf{M}_p$

Direct is $5.3\,\ell \implies$ already $\mathbf{1.3}\times$.

One inversion survives — and it is now the second most expensive line. **Can we kill it too?**

Fusion 2: killing the last inversion — Okeya–Sakurai

Montgomery ladder: carries only x-coordinates. Okeya–Sakurai recover $y(kP)$ from the *adjacent* point $x((k+1)P)$ — which the ladder computed anyway.

Lucas ladder: carries only traces, and already gives σ_{e+1}. With $\alpha = A + Bu$, $\gamma = \alpha^e = C + Du$:

$$W_0 = \sigma_{e+1} - A\,\sigma_e = -2cBD, \qquad W_1 = B\,\sigma_e = 2BC$$

$\Longrightarrow$ γ is determined, up to **one division by** $2cB$.

And the divisor is ours to choose: $U = -16c\,n\,(x_0 x_1)^2 = -4c\,n^3 B^2$, fused into the *same* chain, $w = U^3 n$:

$$(m,\ n^{-1},\ U^{-1}) \qquad \Longrightarrow \qquad k = 2B\,n^3 U^{-1} = \tfrac{-1}{2cB}$$

$$z = x\,\overline{\gamma}\,(m^2 n^{-1}) \ \text{—}\ 15\,\mathbf{M}_p + 2\,\mathbf{S}_p, \textbf{ no inversion}.$$

Main path, v2 — the new algorithm

Main path: $x_0 x_1 \neq 0$, $\beta = -c$, $e = (p+2)/3$.

1. **Prepare:** $n = x_0^2 + c x_1^2$, $U = -16cn(x_0 x_1)^2$, $w = U^3 n$.
2. **Fuse:** one addition chain yields (m, n^{-1}, U^{-1}), with $m^3 = n$.
3. **Project:** $\alpha = x^{1-p}$, $\tau = 2(x_0^2 - c x_1^2) n^{-1}$.
4. **Lucas ladder:** compute $(\sigma_e, \sigma_{e+1}) = (V_e(\tau), V_{e+1}(\tau))$.
5. **Recover and verify:** use both traces and the fused inverses; check $z^3 = x$.

One $\mathbb{F}_p$ addition chain + one Lucas ladder; **no inversion at all**.

$\approx 1.0\,\ell + 1.8\,\ell + O(1) = \mathbf{2.8}\,\ell + \mathbf{36}\ \mathbf{M}_p$ vs. $4.1\,\ell$ (v1) vs. $5.3\,\ell$ (direct)

Verification and ninth-root correction

`FusedExp` first adjusts the base-field root so that $m^3 = n$. The final check verifies the recovered $\mathbb{F}_{p^2}$ candidate and rejects non-cubes.

For $p \equiv 1 \pmod 9$, let ω be a primitive cube root of unity and ζ a primitive ninth root, with $\zeta^3 = \omega^2$.

Check	Result
$z^3 = x$	z
$z^3\omega^2 = x$	$z\zeta$
$z^3\omega = x$	$z\zeta^2$
otherwise	$\perp$

- For cubic inputs, the adjusted m already yields $z^3 = x$.
- The extra branches also handle an unadjusted candidate.
- For $p \equiv 4, 7 \pmod 9$, no root-of-unity correction is needed.

Cost comparison ($\ell = \log_2 p$)

Weighted cost: $W = \#\mathbf{M}_p + 0.8\#\mathbf{S}_p + 3\#\mathbf{M}_{p^2} + 2\#\mathbf{S}_{p^2}$.

Method	**Approx. cost**	**BLS12-377**	**Speed-up**
Direct addition chain	5.3ℓ	1913	$1\times$
Frobenius 1-bit	4.25ℓ	1602	$1.19\times$
Frobenius 2-bit	3.4ℓ	1282	$1.49\times$
Torus	$2.8\ell + 36$	1085	$1.76\times$

- Costs in $\mathbf{M}_p$ units; additions $\mathbf{A}_p$ and constant multiplications $\mathbf{C}_p$ are reported separately in the paper.
- Direct and Torus use exact chain counts; Frobenius columns use estimates.
- Ratios are relative to Direct. Base-field arithmetic avoids extension-field overhead.

Benchmarks

Target primes

Prime	$\log_2 p$	$p \bmod 9$	β	**Source**
p-p254	254	1	-1	Pairing [BCTV14] (Ethereum)
p-p377	377	7	-5	Pairing [BCG$^+$20] (Linea)
p-p381	381	1	-1	Pairing [Bow17] (Ethereum)
i-p381	381	4	-1	Isogeny [COR25]
i-p575	575	4	-1	Isogeny [COR25]
i-p765	765	4	-1	Isogeny [COR25]

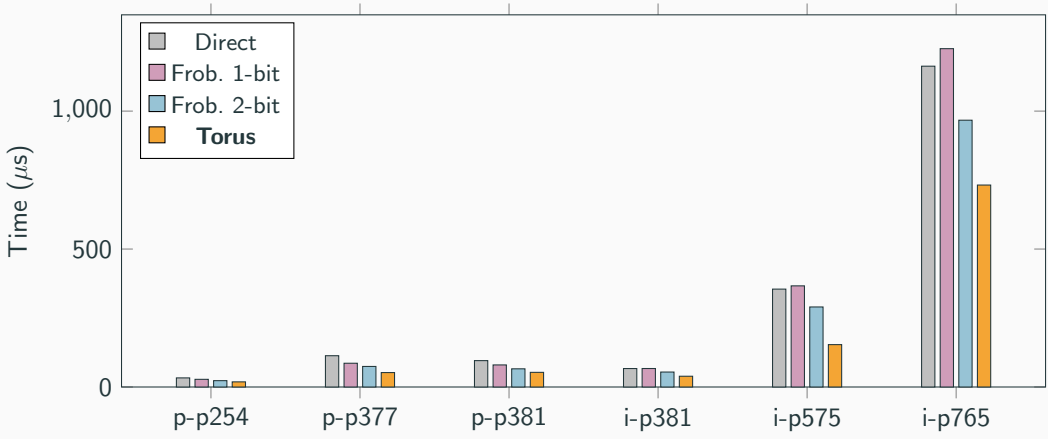
Benchmark results

AMD EPYC 9R14, Go 1.24, median of 4 runs

Prime	Direct (μs)	Frob. 1 (μs)	Frob. 2 (μs)	Torus (μs)	vs. Direct	vs. Frob. 2
p-p254	33.0	27.8	23.0	**18.6**	1.77×	1.24×
p-p377	113.4	86.1	74.5	**52.3**	2.17×	1.43×
p-p381	95.6	80.0	65.9	**53.1**	1.80×	1.24×
i-p381	66.9	67.0	54.2	**39.0**	1.72×	1.39×
i-p575	354.8	366.6	290.1	**153.6**	2.31×	1.89×
i-p765	1162.8	1226.3	967.0	**732.0**	1.59×	1.32×

Torus helper: `addchain` for five primes; generic square-and-multiply for i-p765.

Visual comparison



Constant-time & conclusion

Constant-time considerations

The kernels are constant-time: chain, ladder and table indices depend only on the public prime.

The full algorithm, as written, is not:

- early return on $x_0 = 0$, $x_1 = 0$
- `FusedExp` tests and adjusts the base-field root
- final verification branches on the input

Fine for **public** inputs — decompression, $\mathrm{CGLHash}_3$, public VDF inputs.

For **secret** inputs — hash-to-curve, QFESTA / CSIDH walks — mask the exceptional cases and select with masks. Mechanical, not free, not yet measured.

Conclusion

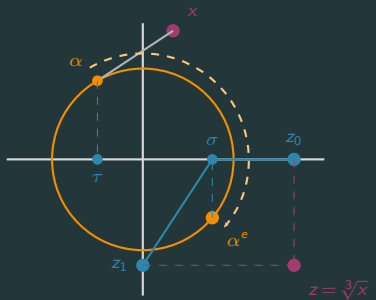
- $\mathbb{F}_{p^2}$ cube roots **entirely in** $\mathbb{F}_p$: torus + Lucas sequences.
- One fused chain (m, n^{-1}, U^{-1}); adjacent traces $\Rightarrow$ **no inversion**.
- $5.3\,\ell \to 4.1\,\ell \to \mathbf{2.8}\,\ell$. In practice **1.6–2.3**×.

Other characteristics	**Other roots**	**Other extensions**
$p \equiv 2 \pmod 3$: fine iff $v_3(p+1) \leq 1$	odd k, $\gcd(k, p+1) = 1$	$\mathbb{T}_6(\mathbb{F}_p)$, XTR-style
but $p \equiv 8 \pmod 9$ needs a fix *inside* $\mathbb{T}_2$	*but* chain and recovery are k-specific	*but* recovery unexplored

ia.cr/2026/392 · github.com/yelhousni/fp2-cbrt

Thank you!

github.com/yelhousni/fp2-cbrt



References i


Marius A. Aardal, Gora Adj, Arwa Alblooshi, Diego F. Aranha, Isaac Andrés Canales Martinez, Jorge Chávez-Saab, Décio Luiz Gazzoni Filho, Krijn Reijnders, and Francisco Rodríguez-Henríquez.
Optimized one-dimensional SQIsign verification on intel and cortex-M4.
IACR TCHES, 2025(1):497–522, 2025.


Leonard M. Adleman, Kenneth Manders, and Gary L. Miller.
On taking roots in finite fields.
18th Annual Symposium on Foundations of Computer Science (FOCS), pages 175–178, 1977.


Gora Adj and Francisco Rodríguez-Henríquez.
Square root computation over even extension fields.
IEEE Transactions on Computers, 63(11):2829–2841, 2014.

References ii


Daniel Bleichenbacher, Wieb Bosma, and Arjen K. Lenstra.
Some remarks on Lucas-based cryptosystems.
In Don Coppersmith, editor, *CRYPTO'95*, volume 963 of *LNCS*, pages 386–396. Springer, Berlin, Heidelberg, August 1995.


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*, pages 947–964. IEEE Computer Society Press, May 2020.


Eli Ben-Sasson, Alessandro Chiesa, Eran Tromer, and Madars Virza.
Succinct non-interactive zero knowledge for a von neumann architecture.
In Kevin Fu and Jaeyeon Jung, editors, *USENIX Security 2014*, pages 781–796. USENIX Association, August 2014.

References iii


Alex Biryukov, Ben Fisch, Gottfried Herold, Dmitry Khovratovich, Gaëtan Leurent, María Naya-Plasencia, and Benjamin Wesolowski.
Cryptanalysis of algebraic verifiable delay functions.
In Leonid Reyzin and Douglas Stebila, editors, *CRYPTO 2024, Part III*, volume 14922 of *LNCS*, pages 457–490. Springer, Cham, August 2024.


Sean Bowe.
BLS12-381: New zk-SNARK elliptic curve construction.
Zcash blog, March 11 2017.
`https://blog.z.cash/new-snark-curve/`.


Wouter Castryck, Thomas Decru, and Frederik Vercauteren.
Radical isogenies.
In Shiho Moriai and Huaxiong Wang, editors, *ASIACRYPT 2020, Part II*, volume 12492 of *LNCS*, pages 493–519. Springer, Cham, December 2020.

References iv


Jesús-Javier Chi-Domínguez, Eduardo Ochoa-Jiménez, and Ricardo-Neftalí Pontaza-Rodas.

Let us walk on the 3-isogeny graph: efficient, fast, and simple, 2025.


Jesús-Javier Chi-Domínguez, Eduardo Ochoa-Jiménez, and Ricardo Neftali Pontaza Rodas.

Let us walk on the 3-isogeny graph: efficient, fast, and simple.
IACR TCHES, 2025(4):644–666, 2025.


Youssef El Housni and Benedikt Bünz.
On the security of constraint-friendly map-to-curve relations.
Cryptology ePrint Archive, Report 2026/590, 2026.

References v


Thomas Icart.
How to hash into elliptic curves.
In Shai Halevi, editor, *CRYPTO 2009*, volume 5677 of *LNCS*, pages 303–316. Springer, Berlin, Heidelberg, August 2009.


Dmitrii Koshelev.
Some remarks on how to hash faster onto elliptic curves.
Journal of Computer Virology and Hacking Techniques, 20:593–605, 2024.


Dmitrii Koshelev.
Batch point compression in the context of advanced pairing-based protocols.
Applicable Algebra in Engineering, Communication and Computing, 36:611–629, 2025.


Arjen K. Lenstra and Eric R. Verheul.
The XTR public key system.
In Mihir Bellare, editor, *CRYPTO 2000*, volume 1880 of *LNCS*, pages 1–19. Springer, Berlin, Heidelberg, August 2000.


Kohei Nakagawa and Hiroshi Onuki.
QFESTA: Efficient algorithms and parameters for FESTA using quaternion algebras.
In Leonid Reyzin and Douglas Stebila, editors, *CRYPTO 2024, Part V*, volume 14924 of *LNCS*, pages 75–106. Springer, Cham, August 2024.


Stephen C. Pohlig and Martin E. Hellman.
An improved algorithm for computing logarithms over $GF(p)$ and its cryptographic significance.
IEEE Transactions on Information Theory, 24(1):106–110, 1978.



Karl Rubin and Alice Silverberg.
Torus-based cryptography.
In Dan Boneh, editor, *CRYPTO 2003*, volume 2729 of *LNCS*, pages 349–365. Springer, Berlin, Heidelberg, August 2003.

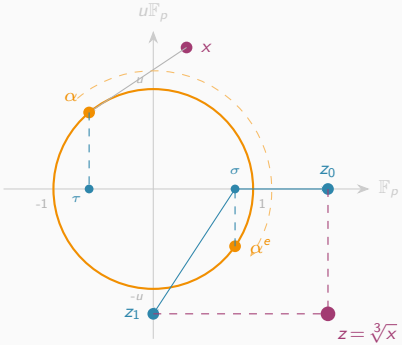


Michael Scott.
A note on the calculation of some functions in finite fields: Tricks of the trade.
Cryptology ePrint Archive, Report 2020/1497, 2020.



Peter Smith and Christopher Skinner.
A public-key cryptosystem and a digital signature system based on the Lucas function analogue to discrete logarithms.
In Josef Pieprzyk and Reihaneh Safavi-Naini, editors, *ASIACRYPT'94*, volume 917 of *LNCS*, pages 357–364. Springer, Berlin, Heidelberg, November / December 1995.

Appendix: Visual – $p \equiv 1 \pmod 3$



$\mathbb{T}_2(\mathbb{F}_p)$, $|\mathbb{T}_2| = p+1$

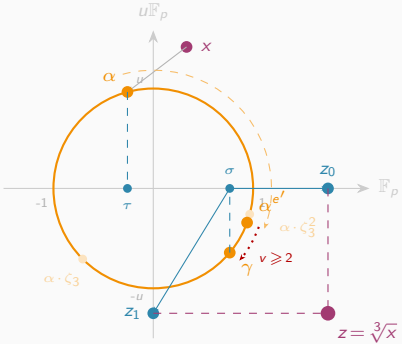
$\gcd(3,\, p+1) = 1$

$\Rightarrow$ cube map **bijective**

- Project: $\alpha = x^{1-p}$
- Rotate: $e = 3^{-1} \bmod (p+1)$
- Recover: $z_0,\, z_1 \in \mathbb{F}_p$

Ninth-root correction if $p \equiv 1 \pmod 9$

Appendix: Visual – $p \equiv 2 \pmod 3$



$\mathbb{T}_2(\mathbb{F}_p)$, $|\mathbb{T}_2| = 3^v \cdot \mu$

$3 \mid (p+1)$

$\Rightarrow$ cube map **3-to-1**

$\alpha,\ \alpha\cdot\zeta_3,\ \alpha\cdot\zeta_3^2$

all cube to α^3 on $\mathbb{T}_2$

- Project: $\alpha = x^{1-p}$
- Rotate: $e' = 3^{-1} \bmod \mu$

  $\alpha^{e'}$; exact if $v = 1$
- Correct $\alpha^{e'} \to \gamma$

  3-Sylow in $\mathbb{F}_{p^2}$, if $v \geqslant 2$
- Recover z_0, z_1

$\mathbb{F}_p$ cbrt unique: $a^{(2p-1)/3}$

No base-field root adjustment

Appendix: Beyond cube roots

Let k be odd and $\gcd(k, p+1) = 1$.

$$\alpha \in \mathbb{T}_2(\mathbb{F}_p), \qquad e = k^{-1} \bmod (p+1), \qquad \mathrm{Tr}(\alpha^e) = V_e(\mathrm{Tr}(\alpha)).$$

- The k-th power map is a bijection on the torus.
- Lucas sequences evaluate its inverse at the trace level.
- Efficient fused exponentiation and recovery formulas depend on k and require separate analysis.
- Deriving and evaluating those formulas is future work.

Appendix: the full fused chain (m, n^{-1}, U^{-1})

Set $w = U^3 n$, $h = (p - r)/d$, $t = w^h$, $\chi = wt^s$.

$$w^{-1} = \chi^a t^b, \qquad a = r - 2, \quad b = d - as.$$

p congruence	d	r	s	(a, b)
7 mod 9	9	7	1	$(5, 4)$
4 mod 9	9	4	2	$(2, 5)$
10 mod 27	27	10	2	$(8, 11)$
19 mod 27	27	19	1	$(17, 10)$

$$U^{-1} = U^2 n w^{-1}, \qquad n^{-1} = U^3 w^{-1}, \qquad m_0 = \chi U^{-1}.$$

When $p \equiv 1 \pmod 9$, adjust m_0 by a ninth root of unity so $m^3 = n$.

Three outputs from one chain: (m, n^{-1}, U^{-1}).

Appendix: recovery detail and exceptional inputs

Let $\alpha = A + Bu = x^{1-p}$, $A = \tau/2$, $B = 2x_0x_1n^{-1}$. The ladder gives $\sigma_e = \mathrm{Tr}(\gamma)$ and $\sigma_{e+1} = \mathrm{Tr}(\alpha\gamma)$, where $\gamma = \alpha^e$.

$$W_0 = \sigma_{e+1} - A\sigma_e, \qquad W_1 = B\sigma_e, \qquad k = 2Bn^3U^{-1} = -\frac{1}{2cB}.$$

$$\gamma = (-cW_1k) + (W_0k)u, \qquad \boxed{z = x\,\overline{\gamma}\,(m^2n^{-1})}.$$

Recovery cost: $15\mathbf{M}_p + 2\mathbf{S}_p + 2\mathbf{C}_p + 5\mathbf{A}_p$; no inversion.

Exceptional inputs handled first:

- $x = 0$: return 0; $x_1 = 0$: use the $\mathbb{F}_p$ cube root of x_0.
- $x_0 = 0$: return au with $a^3 = -x_1/c$; verify or reject.