On the Security of Constraint-Friendly Map-to-Curve Relations

Youssef El Housni [1] Benedikt Bünz [2]

Consensys, Linea [1]

New York University [2]

May 27th, 2026 – zkSecurity

consensys Linea

Where Hash-to-Curve Shows Up

Many cryptographic protocols need a deterministic map

$$H_{\text{curve}} : \{0,1\}^* \longrightarrow E(\mathbb{F}_q).$$

BLS signatures

$$e(\sigma, G_2) = e(H_{\text{curve}}(M), \text{pk}).$$

Verifiers must recompute the curve point before checking the pairing.

Multiset hashing

$$S = \{m_i\}_i \longmapsto \sum_i H_{\text{curve}}(m_i).$$

Used to compress unordered data, e.g. memory or state consistency checks.

Classical Hash-to-Curve Pipeline

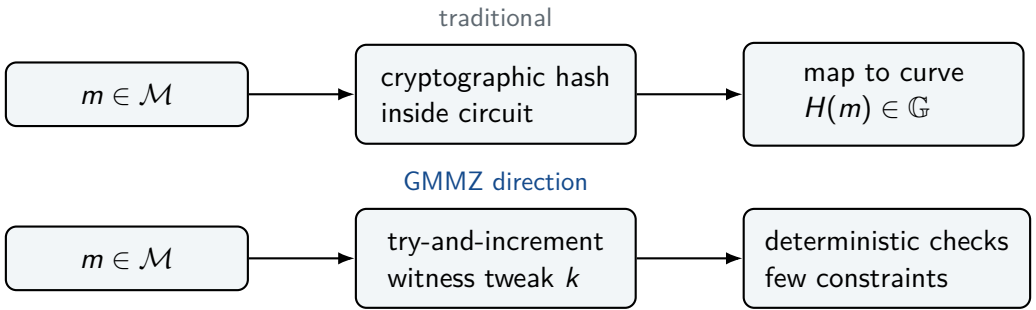
Standard maps use two phases:

$$m \xrightarrow{\text{hash-to-field}} u \in \mathbb{F}_q \xrightarrow{\text{map-to-curve}} P \in E(\mathbb{F}_q).$$

Method	$m \mapsto u \in \mathbb{F}_q$	$u \mapsto E(\mathbb{F}_q)$
Try-and-increment	hash function	try $x = u + k$, $k = 0, 1, \ldots$, until $g(x)$ is square; set $y = \sqrt{g(x)}$
SWU	hash function	compute fixed rational candidates $x_i(u)$; choose one with square $g(x_i)$
SSWU	hash function	derive two candidates from u^2; choose one with square $g(x)$
WB-SSWU	hash function	run SSWU on isogenous E', then apply isogeny $E' \to E$
Elligator2	hash function	use a rational map to a Montgomery/Edwards curve; recover the corresponding point

Problem Setting

- Hash-to-curve inside zk circuits is expensive because the inner hash dominates cost.
- GMMZ [1] chose try-and-increment: the prover witnesses k, then the circuit checks $x = mT + k$ and the curve equation.
- Goal: keep verifier constraints tiny while preserving signature and multiset-hash security.

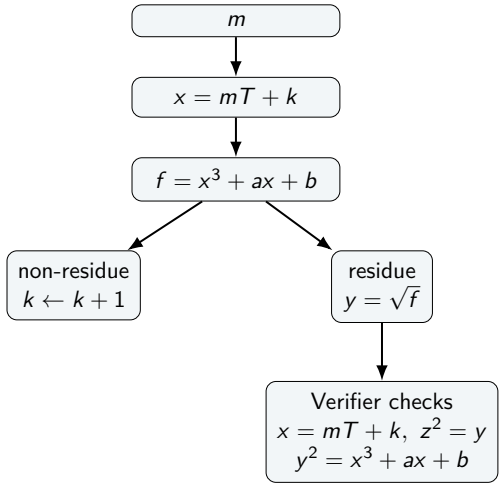


GMMZ x-increment Construction

- Encode message into an x-coordinate window:

$$x = mT + k, \qquad 0 \le k < T.$$

- Search for a tweak k such that (x, y) is on the curve.
- For $q \equiv 3 \pmod 4$, use $z^2 = y$ to exclude the inverse point $(x, -y)$.



GMMZ Properties

Disjoint encoding windows

$$x = mT + k, \qquad 0 \le k < T, \qquad x \in [mT, (m+1)T).$$

Distinct messages use disjoint x-ranges, and (m, k) is recovered by Euclidean division.

Inverse exclusion

For $q \equiv 3 \pmod 4$, exactly one of y and $-y$ is square. The witness $z^2 = y$ selects the canonical point.

Failure probability

Each candidate x succeeds with probability $\approx 1/2$. All T trials fail with probability $\approx 2^{-T}$.

What We Show

Three gaps in the original analysis

1. The security threshold is not $MT \ll q$; the right boundary is $MT < \sqrt{q}$.
2. The construction is not field-agnostic, but can be extended naturally.
3. EC-GGM [2] misses the automorphism structure of deployed $j = 0$ curves, yielding concrete BLS forgeries once the signing-oracle tweak condition is met.

- Countermeasure: switch the encoding coordinate from x to y.

Abstract Forgery Target

What the attacker wants

Find two valid relation instances

$$P_i = (x_i, y), \qquad x_i = m_i T + k_i < MT,$$

for $m_1 \neq m_2$, with a known scalar relation

$$P_2 = [\mu]P_1.$$

At least one side must use the signing oracle's canonical tweak $k_s = k_s^*$; the forged side only needs a valid relation tweak.

Why this forges BLS

Query a signature

$$\sigma_s = [\mathsf{sk}]P_s.$$

Then output

$$\sigma_t = [\mu]\sigma_s = [\mathsf{sk}]P_t,$$

so $e(\sigma_t, G_2) = e(P_t, \mathsf{pk})$.

Question: can we find two small x-coordinates related by a known scalar?

Why the Attack Exists on $j = 0$ Curves

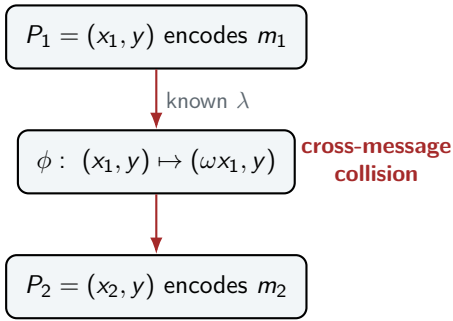
For curves $E : y^2 = x^3 + b$, there is an order-3 automorphism

$$\phi(x, y) = (\omega x, y) = [\lambda](x, y).$$

- It scales the encoded coordinate linearly: $x \mapsto \omega x$.
- If two small values satisfy $x_2 \equiv \omega x_1 \pmod{q}$, then

$$P_2 = \phi(P_1) = [\lambda]P_1.$$

- A known scalar relation between two encoded points is exactly what BLS forgery needs.



The $\sqrt{q}$ Threshold

The automorphism gives the lattice

$$L = \{(a, b) \in \mathbb{Z}^2 : b \equiv \omega a \pmod{q}\}.$$

- $\det(L) = q$.
- Minkowski gives a non-zero vector with

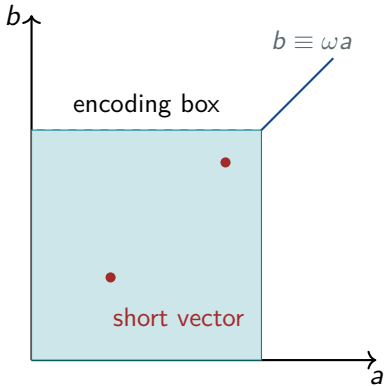
$$\max(|a|, |b|) \leq \sqrt{q}.$$

- If $MT \gtrsim \sqrt{q}$, both coordinates can land inside the encoding range.

This also matches GMMZ's own bound:

$$\mathrm{Adv} \lesssim \frac{cMTQ + Q^2}{p}$$

after Q queries:
structural hit: $cMTQ/p$
birthday hit: Q^2/p



Finding Candidate Coordinates

Generic $j = 0$ curve

Find a short basis of

$$L = \{(a, b) : b \equiv \omega a \pmod{q}\}.$$

Half-GCD, lattice reduction, and Cornacchia/CM all give two independent vectors $(a_1, b_1), (a_2, b_2)$. Search

$$(X, Y) = i(a_1, b_1) + j(a_2, b_2)$$

with $|X|, |Y| < MT$, then check valid tweaks and $k_s = k_s^*$.

Parameterized family

For BN curves the basis is symbolic:

$$q(u) = 36u^4 + 36u^3 + 24u^2 + 6u + 1,$$
$$\omega(u) = 18u^3 + 18u^2 + 9u + 1,$$
$$a_u = 6u^2 + 2u + 1, \qquad b_u = 2u,$$

with basis

$$(a_u, b_u), \qquad (a_u + b_u, -a_u).$$

The coefficient search is finite and exhaustive once M, T are fixed.

The key extra check is whether either side is the canonical signing-oracle point for its message.

Concrete Outcomes at $M = 2^{120}$, $T = 128$

For 254-bit BN curves, $MT = 2^{127} \approx \sqrt{q}$. The exhaustive coefficient range is $|i| \le 2,\ |j| \le 1$.

Beuchat/TEPLA BN254B: complete

$$E : y^2 = x^3 + 5, \qquad u_B = 2^{62} - 2^{54} + 2^{44}.$$

Source-compatible triples:

$$(-2, 1, 1),\ (-2, 1, 2),\ (2, -1, 1),\ (2, -1, 2).$$

$$\begin{array}{lcl} x_1 & = & \texttt{0x5f408fd00600000000000000000001} \\ x_2 & = & \texttt{0x5f408fd0060000017e80600000000001} \\ k_1 & = & 1, \quad k_1^* = 0 \\ k_2 & = & 1, \quad k_2^* = 1 \end{array}$$

Query m_2, output a forgery on (m_1, k_1):

$$P_1 = \phi^2(P_2), \qquad \sigma_1 = [\lambda^2]\sigma_2.$$

Ethereum BN254: relation only

$$E : y^2 = x^3 + 3, \qquad u_0 = \texttt{0x44e992b44a6909f1}.$$

Valid relation triples:

$$(-2, 1, 1), (-2, 1, 2), (-1, 0, 1), (-1, 0, 2),$$
$$(1, 0, 1), (1, 0, 2), (2, -1, 1), (2, -1, 2).$$

Example $(-1, 0, 1)$:

$$\begin{array}{lcl} x_1 & = & \texttt{0x6f4d8248eeb859fc8211bbeb7d4f1129} \\ x_2 & = & \texttt{0x89d3256894d213e2} \\ k_1 & = & 41, \quad k_1^* = 0 \\ k_2 & = & 98, \quad k_2^* = 0 \end{array}$$

Both tweaks are valid relation witnesses, but neither side is canonical for a message-only signing query.

Complete Ethereum BN254 Forgery at $M = 2^{124}$

Increasing only the message bound gives $MT = 2^{131}$. The exhaustive coefficient range grows to $|i| \le 36$, $|j| \le 18$, and the search finds 32 source-compatible triples.

Example triple $(-29, 15, 1)$

$$
\begin{aligned}
x_1 &= \texttt{0x6163d1ffd0e14ebc7099915c02203c600} \\
x_2 &= \texttt{0x6858aa245fccd45db3bf53fa5336e4201} \\
m_1 &= \texttt{0xc2c7a3ffa1c29d78e13322b8044078c} \\
m_2 &= \texttt{0xd0b15448bf99a8bb677ea7f4a66dc84} \\
k_1 &= 0, \quad k_1^* = 0 \\
k_2 &= 1, \quad k_2^* = 1
\end{aligned}
$$

Both sides are canonical signing-oracle inputs.

Forgery

$$P_1 = (x_1, y), \qquad P_2 = \phi(P_1) = [\lambda]P_1.$$

Query the signing oracle on m_1:

$$\sigma_1 = [\mathsf{sk}]P_1.$$

Output

$$\sigma_2 = [\lambda]\sigma_1 = [\mathsf{sk}]P_2,$$

which verifies on the fresh message m_2.

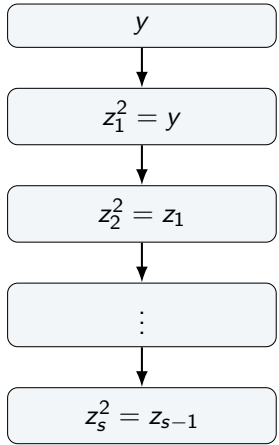
The Sage artifact confirms the pairing checks for both the Beuchat BN254B and Ethereum BN254 complete forgeries.

Field Restriction in x-increment

- Inverse exclusion relies on exactly one of $y, -y$ being a quadratic residue.
- That only holds when $q \equiv 3 \pmod 4$.
- For general q, one can replace $z^2 = y$ with $z^{2^s} = y$, where $s = v_2(q-1)$.

Verifier cost: $3 + s$ multiplications

but witness generation succeeds with probability about 2^{-s}, so high-2-adicity fields become impractical.



cheap for $s = 1$
painful for $s = 46$

Countermeasure: y-increment

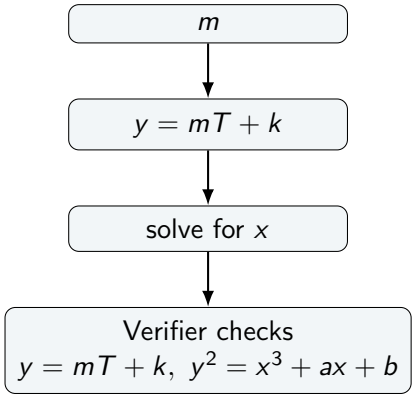
Encode the message into the y-coordinate instead:

$$y = mT + k, \qquad 0 \le k < T.$$

- Recover x from $y^2 = x^3 + ax + b$.
- On $j = 0$: one cube root.
- On $j \neq 0, 1728$: solve a depressed cubic.

Immediate benefits

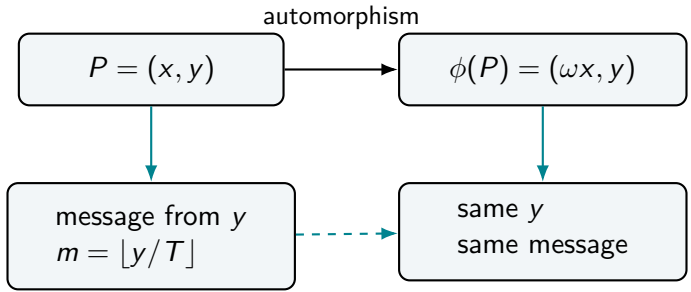
- No q mod 4 restriction.
- No inverse-exclusion witness chain.
- The order-3 automorphism fixes y, so it cannot cross messages.



encode in y

automorphism-safe on $j = 0$

Why y-increment Neutralizes the Automorphism



Why y-increment Does Not Need Inverse Exclusion

x-increment: same encoded coordinate

The message is recovered from

$$x = mT + k.$$

But inverse points share x:

$$(x, y) \quad \text{and} \quad (x, -y).$$

So the verifier must choose one sign of y. GMMZ does this with $z^2 = y$, which is the inverse-exclusion constraint.

y-increment: inverse leaves the range

The message is recovered from

$$y = mT + k, \qquad 0 \le y < MT.$$

The inverse has y-coordinate $-y \bmod q = q - y$, which lies near q, not in the low interval $[0, MT)$, when $MT < q/2$. The range check rejects it automatically.

Witness Generation for y-increment

$j = 0$ curves (BN254, BLS12-381, BLS12-377, secp256k1)

For curves $y^2 = x^3 + b$,

$$x^3 = y^2 - b.$$

Witness generation reduces to a cube root in $\mathbb{F}_q$ (often just one exponentiation plus a fixed cube root of unity).

$j \neq 0, 1728$ curves (P-256)

Solve $x^3 + ax + (b - y^2) = 0$ with a depressed-cubic solver. In our implementation this is Cardano's method; all arithmetic stays in $\mathbb{F}_q$ even when the discriminant case requires $\mathbb{F}_{q^2}$. We use the algebraic torus idea from [3].

Security Statement for y-increment

Main theorem

On $j = 0$ curves, if $MT < \sqrt{q}$, the y-increment relation is secure in the *standard* EC-GGM:

$$\mathrm{Adv}^{\mathrm{forge}} \leq \frac{cMTQ + Q^2}{p}.$$

EC-GGM modeling gap. Standard EC-GGM only accounts for the sign symmetry

$$\{\pm 1\}, \qquad \pi(-i) = -\pi(i).$$

Some curves have more automorphisms, e.g.

$$\begin{aligned} j = 0 &: \quad \{\pm 1, \pm\phi, \pm\phi^2\}, \\ j = 1728 &: \quad \{\pm 1, \pm\psi\}. \end{aligned}$$

What we do. In the paper we propose a ϕ-augmented EC-GGM, where applying the automorphism costs a query.

For y-increment, no augmentation is needed: ϕ preserves y, so it cannot create cross-message collisions.

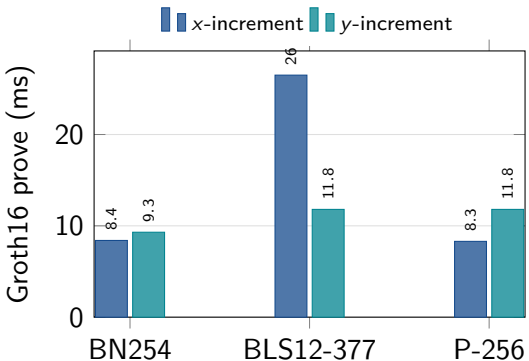
The verifier remains simple: 3 multiplications plus one range check.

Constraint Counts

Setting	Curve	x-increment R1CS	y-increment R1CS
Native	BN254	18	**16**
Native	BLS12-377	63	**16**
Native	Grumpkin	45	**16**
Emulated	BN254	977	**821**
Emulated	BLS12-377	7,741	**1,065**
Emulated	P-256	1,065	**911**

- Native y-increment is flat across curves: 16 constraints regardless of $s = v_2(q-1)$.
- Biggest win is emulated BLS12-377: 7.3$\times$ fewer constraints.

Benchmark Snapshot



- Verification time is essentially unchanged.
- Proving time improves where the 2^s-witness chain was expensive.
- On emulated BLS12-377:

$$26.5 \text{ ms} \rightarrow 11.8 \text{ ms}$$

for Groth16.

Noir Circuit Attack

What the tests show

We added attack tests to the GMMZ Noir circuits:

$$P_2 = \phi(P_1) = (\omega x_1, y_1).$$

The same y and QR witness pass the relation, while the scaled x-coordinate encodes a different message.

Where it appears

The attack is confirmed in both circuit variants:

- native Grumpkin circuit;
- non-native limb circuit;
- BLS-forgery tests using the known scalar relation.

This is the algebraic attack at the circuit level, not just a proof-model issue.

Range Check Miss in Noir

Missing implementation check

The formal relation needs $0 \leq t < T$. The Noir circuit checks $x = mT + t$, but omits the range check on t.

Injectivity breaks

With $T = 256$,

$$7T + 0 = 6T + 256 = 1792.$$

The same curve point is accepted for two different messages because $t = 256$ is not rejected.

Consequences

The test suite demonstrates:

- zero-message forgery: $m = 0,\ t = x$;
- multiset-hash collisions;
- zkVM memory-consistency forgery.

This is a Noir implementation gap, separate from the formal GMMZ construction.

Takeaways

- Constraint-friendly relations are useful, but the algebraic structure of real curves matters.
- For x-increment, the right security boundary is $MT < \sqrt{q}$, and $j = 0$ automorphisms break concrete parameters.
- y-increment keeps the low-cost verifier, removes the field restriction, and avoids the automorphism attack by construction.
- The full attack and circuit benchmarks are implemented in SageMath, Noir, and `gnark`.

Questions?

Email: youssef.elhousni@consensys.net
GitHub: @yelhousni
Telegram: @ElMarroqui

References

Groth, J., Malvai, H., Miller, A., Zhang, Y.N.: Constraint-friendly map-to-elliptic-curve-group relations and their applications. In: Hanaoka, G., Yang, B.Y. (eds.) ASIACRYPT 2025, Part II. LNCS, vol. 16246, pp. 511–543. Springer, Singapore (Dec 2025). https://doi.org/10.1007/978-981-95-5096-8_16

Groth, J., Shoup, V.: On the security of ECDSA with additive key derivation and presignatures. In: Dunkelman, O., Dziembowski, S. (eds.) EUROCRYPT 2022, Part I. LNCS, vol. 13275, pp. 365–396. Springer, Cham (May / Jun 2022). https://doi.org/10.1007/978-3-031-06944-4_13

Housni, Y.E.: Fast cube roots in fp2 via the algebraic torus. Cryptology ePrint Archive, Paper 2026/392 (2026), https://eprint.iacr.org/2026/392