Fast and simple constant-time hashing to the BLS12-381 elliptic curve

(and other curves, too!)

Riad S. Wahby, Dan Boneh

Stanford

December 3rd, 2019

Motivation

Our initial motivation: BLS signatures [BLS01]

Motivation

Our initial motivation: BLS signatures [BLS01]

- Also: VRFs, OPRFs, PAKEs, IBE, ...

Motivation

Our initial motivation: BLS signatures [BLS01]

- Also: VRFs, OPRFs, PAKEs, IBE, ...

Why simple and constant time?

Motivation

Our initial motivation: BLS signatures [BLS01]

- Also: VRFs, OPRFs, PAKEs, IBE, ...

Why simple and constant time?

- Avoids side channels (e.g. Dragonblood [VR19]), without requiring randomized blinding

Motivation

Our initial motivation: BLS signatures [BLS01]

- Also: VRFs,

fixed-modulus arithmetic only

Why simple and constant time?

- Avoids side channels (e.g. Dragonblood [VR19]), without requiring randomized blinding

Motivation

Our initial motivation: BLS signatures [BLS01]

- Also: VRFs,

fixed-modulus arithmetic only

Why simple and constant time?

- Avoids side channels (e.g. Dragonblood [VR19]), without requiring randomized blinding
- Fixed modulus: an order of magnitude less code

Motivation

Our initial motivation: BLS signatures [BLS01]

- Also: VRFs,

fixed-modulus arithmetic only

Why simple and constant time?

- Avoids side channels (e.g. Dragonblood [VR19]), without requiring randomized blinding
- Fixed modulus: an order of magnitude less code
- Embedded systems often have fixed-modulus hardware acceleration but *slow* generic bigint

Motivation

Our initial motivation: BLS signatures [BLS01]

- Also: VRFs, OPRFs, PAKEs, IBE, ...

Why simple and constant time?

- Avoids side channels (e.g. Dragonblood [VR19]), without requiring randomized blinding
- Fixed modulus: an order of magnitude less code
- Embedded systems often have fixed-modulus hardware acceleration but *slow* generic bigint

Why the BLS12-381 pairing-friendly elliptic curve?

- Widely used curve for ≈120-bit security level
  - ☞ Will (probably) be an IETF standard soon

Our contributions

1. "Indirect" maps via isogenies, sidestepping limitations of existing maps when $j \in \{0, 1728\}$ (Recall: pairing-friendly curves often have $j = 0$)

Our contributions

1. "Indirect" maps via isogenies, sidestepping limitations of existing maps when $j \in \{0, 1728\}$
   (Recall: pairing-friendly curves often have $j = 0$)

2. An optimization to the map of [BCIMRT10] that reduces its cost to 1 exponentiation
   ✓ On par with the fastest existing maps

Our contributions

1. "Indirect" maps via isogenies, sidestepping limitations of existing maps when $j \in \{0, 1728\}$ (Recall: pairing-friendly curves often have $j = 0$)

2. An optimization to the map of [BCIMRT10] that reduces its cost to 1 exponentiation
   ✓ On par with the fastest existing maps
   ✓ Fast impls are simple and constant time

Our contributions

1. "Indirect" maps via isogenies, sidestepping limitations of existing maps when $j \in \{0, 1728\}$
(Recall: pairing-friendly curves often have $j = 0$)

2. An optimization to the map of [BCIMRT10] that reduces its cost to 1 exponentiation
✓ On par with the fastest existing maps
✓ Fast impls are simple and constant time
✓ Applies to essentially any elliptic curve

Our contributions

1. "Indirect" maps via isogenies, sidestepping limitations of existing maps when $j \in \{0, 1728\}$ (Recall: pairing-friendly curves often have $j = 0$)

2. An optimization to the map of [BCIMRT10] that reduces its cost to 1 exponentiation
   ✓ On par with the fastest existing maps
   ✓ Fast impls are simple and constant time
   ✓ Applies to essentially any elliptic curve

3. Impl and eval of 34 hash variants for BLS12-381

Our contributions

1. "Indirect" maps via isogenies, sidestepping limitations of existing maps when $j \in \{0, 1728\}$
(Recall: pairing-friendly curves often have $j = 0$)

2. An optimization to the map of [BCIMRT10] that reduces its cost to 1 exponentiation
   - ✓ On par with the fastest existing maps
   - ✓ Fast impls are simple and constant time
   - ✓ Applies to essentially any elliptic curve

3. Impl and eval of 34 hash variants for BLS12-381
   - ✓ 1.3–2× faster than prior constant-time hashes, $\leq$ 9% slower than *non*-CT deterministic maps

Our contributions

1. "Indirect" maps via isogenies, sidestepping limitations of existing maps when $j \in \{0, 1728\}$
   (Recall: pairing-friendly curves often have $j = 0$)

2. An optimization to the map of [BCIMRT10] that reduces its cost to 1 exponentiation
   ✓ On par with the fastest existing maps
   ✓ Fast impls are simple and constant time
   ✓ Applies to essentially any elliptic curve

3. Impl and eval of 34 hash variants for BLS12-381
   ✓ 1.3–2× faster than prior constant-time hashes, $\leq$ 9% slower than *non*-CT deterministic maps
   ☞ Open-source impls in C, Rust, Python, ...

Roadmap

1. Hash functions to elliptic curves
2. Optimizing the map of [BCIMRT10]
3. Evaluation results
4. IETF standardization efforts

Notation

$H_p : \{0,1\}^\star \to \mathbb{F}_p$ and $H_q : \{0,1\}^\star \to \mathbb{F}_q$ are hash functions modeled as random oracles

Notation

$H_p : \{0,1\}^\star \to \mathbb{F}_p$ and $H_q : \{0,1\}^\star \to \mathbb{F}_q$ are hash functions modeled as random oracles, e.g.,

1. Seed a PRG with the input
2. Extract a $2\log p$-bit integer
3. Reduce mod p

Notation

$H_p : \{0,1\}^\star \to \mathbb{F}_p$ and $H_q : \{0,1\}^\star \to \mathbb{F}_q$ are hash functions modeled as random oracles

$E(\mathbb{F}_p)$ is the elliptic curve group with identity $\mathcal{O}$ and points $\{(x,y) : x,y \in \mathbb{F}_p, y^2 = x^3 + ax + b\}$
☞ additive notation, $[\alpha]P$ for scalar multiplication

Notation

$H_p : \{0,1\}^\star \to \mathbb{F}_p$ and $H_q : \{0,1\}^\star \to \mathbb{F}_q$ are hash functions modeled as random oracles

$E(\mathbb{F}_p)$ is the elliptic curve group with identity $\mathcal{O}$ and points $\{(x,y) : x, y \in \mathbb{F}_p, y^2 = x^3 + ax + b\}$
☞ additive notation, $[\alpha]P$ for scalar multiplication

$\mathbb{G} \subseteq E(\mathbb{F}_p)$ is a subgroup of prime order q.
$\#E(\mathbb{F}_p) = hq$; h is the *cofactor*.

Notation

$H_p : \{0,1\}^\star \to \mathbb{F}_p$ and $H_q : \{0,1\}^\star \to \mathbb{F}_q$ are hash functions modeled as random oracles

$E(\mathbb{F}_p)$ is the elliptic curve group with identity $\mathcal{O}$ and points $\{(x,y) : x, y \in \mathbb{F}_p, y^2 = x^3 + ax + b\}$
☞ additive notation, $[\alpha]P$ for scalar multiplication

$\mathbb{G} \subseteq E(\mathbb{F}_p)$ is a subgroup of prime order q.
$\#E(\mathbb{F}_p) = hq$; h is the *cofactor*.

BLS12-381 defines $\mathbb{G}_1 \subset E_1(\mathbb{F}_p)$, $\mathbb{G}_2 \subset E_2(\mathbb{F}_{p^2})$, $\mathbb{G}_T \subset \mathbb{F}_{p^{12}}^\times$, and $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ s.t.

$$e([\alpha]P_1, [\beta]P_2) = e(P_1, P_2)^{\alpha \cdot \beta} \qquad \alpha, \beta \in \mathbb{F}_q$$

Attempt #1: random scalar

For some distinguished point $\hat{P} \in \mathbb{G}$,

$\mathsf{HashToCurve}_{\mathsf{RS}}(\mathsf{msg})$:

$x \leftarrow H_q(\mathsf{msg})$

return $[x]\hat{P}$

Attempt #1: random scalar

For some distinguished point $\hat{P} \in \mathbb{G}$,

$\text{HashToCurve}_{\text{RS}}(\text{msg})$:

$x \leftarrow H_q(\text{msg})$

return $[x]\hat{P}$

Informally: need a point with unknown discrete log

☞ known dlog breaks security of most protocols (e.g., BLS signatures)

BLS signatures

For $H : \{0,1\}^\star \to \mathbb{G}_1,\ \hat{Q} \in \mathbb{G}_2$:

KeyGen() $\to (pk, sk)$:

$r \xleftarrow{\mathrm{R}} \mathbb{Z}_q$; return $([r]\hat{Q}, r)$

BLS signatures

For $H : \{0,1\}^\star \to \mathbb{G}_1$, $\hat{Q} \in \mathbb{G}_2$:

KeyGen() $\to (pk, sk)$:

$r \xleftarrow{\mathrm{R}} \mathbb{Z}_q$; return $([r]\hat{Q}, r)$

Sign(sk, msg) $\to$ sig:

return $[sk]H(\mathrm{msg})$

BLS signatures

For $H : \{0,1\}^\star \to \mathbb{G}_1,\ \hat{Q} \in \mathbb{G}_2$:

KeyGen() $\to (pk, sk)$:

$r \xleftarrow{R} \mathbb{Z}_q$; return $([r]\hat{Q}, r)$

Sign(sk, msg) $\to$ sig:

return $[sk]H(\text{msg})$

Verify(pk, msg, sig) $\to$ {True, False}:

$e(H(\text{msg}), pk) \stackrel{?}{=} e(\text{sig}, \hat{Q})$

BLS signatures and $\mathsf{HashToCurve}_{\mathsf{RS}}$

For $\mathsf{HashToCurve}_{\mathsf{RS}} : \{0,1\}^\star \to \mathbb{G}_1,\ \hat{Q} \in \mathbb{G}_2$:

KeyGen() $\to (pk, sk)$:

$r \xleftarrow{\mathrm{R}} \mathbb{Z}_q$; return $([r]\hat{Q}, r)$

Sign(sk, msg) $\to$ sig:

return $[sk]\mathsf{HashToCurve}_{\mathsf{RS}}(\mathsf{msg})$

Verify(pk, msg, sig) $\to$ {True, False}:

$e(\mathsf{HashToCurve}_{\mathsf{RS}}(\mathsf{msg}), pk) \stackrel{?}{=} e(\mathsf{sig}, \hat{Q})$

$$\mathsf{sig}_1 = \mathsf{Sign}(sk, \mathsf{msg}_1) = [sk \cdot H_q(\mathsf{msg}_1)]\hat{P}$$

BLS signatures and $\text{HashToCurve}_{\text{RS}}$

For $\text{HashToCurve}_{\text{RS}} : \{0,1\}^\star \to \mathbb{G}_1$, $\hat{Q} \in \mathbb{G}_2$:

KeyGen() $\to (pk, sk)$:

$r \xleftarrow{\text{R}} \mathbb{Z}_q$; return $([r]\hat{Q}, r)$

Sign$(sk, \text{msg}) \to \text{sig}$:

return $[sk]\text{HashToCurve}_{\text{RS}}(\text{msg})$

Verify$(pk, \text{msg}, \text{sig}) \to \{\text{True}, \text{False}\}$:

$e(\text{HashToCurve}_{\text{RS}}(\text{msg}), pk) \stackrel{?}{=} e(\text{sig}, \hat{Q})$

$$\text{sig}_1 = \text{Sign}(sk, \text{msg}_1) = [sk \cdot H_q(\text{msg}_1)]\hat{P}$$

☞ Trivial existential forgery:

$$\text{Sign}(sk, \text{msg}_2) = \left[H_q(\text{msg}_2) \cdot H_q(\text{msg}_1)^{-1}\right]\text{sig}_1$$

Attempt #2: hash and check

$\mathsf{HashToCurve}_{\mathsf{H\&C}}(\mathsf{msg})$:

$\mathsf{ctr} \leftarrow 0$

$y \leftarrow \bot$

while $y = \bot$:

$x \leftarrow H_p(\mathsf{ctr} \,||\, \mathsf{msg})$

$\mathsf{ctr} \leftarrow \mathsf{ctr} + 1$

$ySq \leftarrow x^3 + ax + b$

$y \leftarrow \mathsf{sqrt}(ySq)$ // $\bot$ if ySq is non-square

$P \leftarrow (x, y)$

return $[h]P$ // map to $\mathbb{G}$ via cofactor mul

Attempt #2: hash and check

$\mathsf{HashToCurve}_{\mathsf{H\&C}}(\mathsf{msg})$:

```
ctr ← 0
y ← ⊥
while y = ⊥:
    x ← H_p(ctr || msg)
    ctr ← ctr + 1
    ySq ← x^3 + ax + b
    y ← sqrt(ySq)          // ⊥ if ySq is non-square
P ← (x, y)
return [h]P                // map to 𝔾 via cofactor mul
```

Attempt #2: hash and check

$\mathsf{HashToCurve}_{\mathsf{H\&C}}(\mathsf{msg})$:

```
ctr ← 0
y ← ⊥
while y = ⊥:
    x ← H_p(ctr || msg)
    ctr ← ctr + 1
    ySq ← x^3 + ax + b
    y ← sqrt(ySq)        // ⊥ if ySq is non-square
P ← (x, y)
return [h]P              // map to 𝔾 via cofactor mul
```

☞ $E(\mathbb{F}_p) = \{(x, y) : x, y \in \mathbb{F}_p, y^2 = x^3 + ax + b\}$

Attempt #2: hash and check

$\mathsf{HashToCurve}_{\mathsf{H\&C}}(\mathsf{msg})$:

$$
\begin{aligned}
&\quad \mathsf{ctr} \leftarrow 0 \\
&\quad y \leftarrow \bot \\
&\quad \text{while } y = \bot: \\
&\quad\quad x \leftarrow H_p(\mathsf{ctr} \,||\, \mathsf{msg}) \\
&\quad\quad \mathsf{ctr} \leftarrow \mathsf{ctr} + 1 \\
&\quad\quad ySq \leftarrow x^3 + ax + b \\
&\quad\quad y \leftarrow \mathsf{sqrt}(ySq) \qquad // \bot \text{ if } ySq \text{ is non-square} \\
&\quad P \leftarrow (x, y) \\
&\quad \text{return } [h]P \qquad // \text{ map to } \mathbb{G} \text{ via cofactor mul}
\end{aligned}
$$

Attempt #2: hash and check

$\mathsf{HashToCurve}_{\mathsf{H\&C}}(\mathsf{msg})$:

- $\mathsf{ctr} \leftarrow 0$
- $y \leftarrow \bot$
- while $y = \bot$:
  - $x \leftarrow H_p(\mathsf{ctr} \,||\, \mathsf{msg})$
  - $\mathsf{ctr} \leftarrow \mathsf{ctr} + 1$
  - $ySq \leftarrow x^3 + ax + b$
  - $y \leftarrow \mathsf{sqrt}(ySq)$ // $\bot$ if ySq is non-square
- $P \leftarrow (x, y)$
- return $[h]P$ // map to $\mathbb{G}$ via cofactor mul

Not constant time; "bad" inputs are common.

Attempt #2: hash and check

HashToCurve$_{\mathsf{H\&C}}$(msg):

ctr $\leftarrow 0$

$y \leftarrow \perp$

while $y = \perp$:

  $x \leftarrow H_p(\text{ctr} \,||\, \text{msg})$

  ctr $\leftarrow$ ctr $+ 1$

  $ySq \leftarrow x^3 + ax + b$

  $y \leftarrow \text{sqrt}(ySq)$ // $\perp$ if ySq is non-square

$P \leftarrow (x, y)$

return $[h]P$ // map to $\mathbb{G}$ via cofactor mul

Not constant time; "bad" inputs are common.

Loop a fixed number of times?

Attempt #2: hash and check

$\mathsf{HashToCurve}_{\mathsf{H\&C}}(\mathsf{msg})$:

    $\mathsf{ctr} \leftarrow 0$

    $y \leftarrow \perp$

    while $y = \perp$:

        $x \leftarrow H_p(\mathsf{ctr} \,||\, \mathsf{msg})$

        $\mathsf{ctr} \leftarrow \mathsf{ctr} + 1$

        $ySq \leftarrow x^3 + ax + b$

        $y \leftarrow \mathsf{sqrt}(ySq)$ // $\perp$ if ySq is non-square

    $P \leftarrow (x, y)$

    return $[h]P$ // map to $\mathbb{G}$ via cofactor mul

✗

Not constant time; “bad” inputs are common.

✗ Loop a fixed number of times?
Slow; well-meaning “optimization” breaks CT.

Deterministic maps to elliptic curves

$M : \mathbb{F}_p \to E(\mathbb{F}_p)$, where $E : y^2 = x^3 + ax + b$ and $p > 5$:

Deterministic maps to elliptic curves

$M : \mathbb{F}_p \to E(\mathbb{F}_p)$, where $E : y^2 = x^3 + ax + b$ and $p > 5$:

Map M	**Restrictions**	**Cost**
[BF01]	$p \equiv 2 \bmod 3$, $a = 0$	1 exp

Deterministic maps to elliptic curves

$M : \mathbb{F}_p \to E(\mathbb{F}_p)$, where $E : y^2 = x^3 + ax + b$ and $p > 5$:

Map M	**Restrictions**	**Cost**
[BF01]	$p \equiv 2 \bmod 3$, $a = 0$	1 exp

$$y^2 = x^3 + b$$
$$\implies x = \sqrt[3]{y^2 - b}$$

Deterministic maps to elliptic curves

$M : \mathbb{F}_p \to E(\mathbb{F}_p)$, where $E : y^2 = x^3 + ax + b$ and $p > 5$:

Map M	**Restrictions**	**Cost**
[BF01]	$p \equiv 2 \bmod 3$, $a = 0$	1 exp
[SW06]	none	3 exp

Deterministic maps to elliptic curves

$M : \mathbb{F}_p \to E(\mathbb{F}_p)$, where $E : y^2 = x^3 + ax + b$ and $p > 5$:

Map M		**Restrictions**	**Cost**
	[BF01]	$p \equiv 2 \bmod 3, a = 0$	1 exp
	[SW06]	none	3 exp
SWU	[Ulas07]	$p \equiv 3 \bmod 4, ab \neq 0$	3 exp

Deterministic maps to elliptic curves

$M : \mathbb{F}_p \to E(\mathbb{F}_p)$, where $E : y^2 = x^3 + ax + b$ and $p > 5$:

Map M		**Restrictions**	**Cost**
	[BF01]	$p \equiv 2 \bmod 3$, $a = 0$	1 exp
	[SW06]	none	3 exp
SWU	[Ulas07]	$p \equiv 3 \bmod 4$, $ab \neq 0$	3 exp
	[Icart09]	$p \equiv 2 \bmod 3$	1 exp

Deterministic maps to elliptic curves

$M : \mathbb{F}_p \to E(\mathbb{F}_p)$, where $E : y^2 = x^3 + ax + b$ and $p > 5$:

Map M		**Restrictions**	**Cost**
	[BF01]	$p \equiv 2 \bmod 3, a = 0$	1 exp
	[SW06]	none	3 exp
SWU	[Ulas07]	$p \equiv 3 \bmod 4, ab \neq 0$	3 exp
	[Icart09]	$p \equiv 2 \bmod 3$	1 exp
S-SWU	[BCIMRT10]	$p \equiv 3 \bmod 4, ab \neq 0$	2 exp

Deterministic maps to elliptic curves

$M : \mathbb{F}_p \to E(\mathbb{F}_p)$, where $E : y^2 = x^3 + ax + b$ and $p > 5$:

Map M		**Restrictions**	**Cost**
	[BF01]	$p \equiv 2 \bmod 3$, $a = 0$	1 exp
	[SW06]	none	3 exp
SWU	[Ulas07]	$p \equiv 3 \bmod 4$, $ab \neq 0$	3 exp
	[Icart09]	$p \equiv 2 \bmod 3$	1 exp
S-SWU	[BCIMRT10]	$p \equiv 3 \bmod 4$, $ab \neq 0$	2 exp
Elligator	[BHKL13]	$b \neq 0, 2 \mid \#E(\mathbb{F}_p)$	1 exp

Deterministic maps to elliptic curves

$M : \mathbb{F}_p \to E(\mathbb{F}_p)$, where $E : y^2 = x^3 + ax + b$ and $p > 5$:

Map M		**Restrictions**	**Cost**
	[BF01]	$p \equiv 2 \bmod 3, a = 0$	1 exp
	[SW06]	none	3 exp
SWU	[Ulas07]	$p \equiv 3 \bmod 4, ab \neq 0$	3 exp
	[Icart09]	$p \equiv 2 \bmod 3$	1 exp
S-SWU	[BCIMRT10]	$p \equiv 3 \bmod 4, ab \neq 0$	2 exp
Elligator	[BHKL13]	$b \neq 0, 2 \mid \#E(\mathbb{F}_p)$	1 exp

[SS04,Ska05,FSV09,FT10a,FT10b,KLR10,CK11,Far11,FT12,FJT13,BLMP19. . .]

Deterministic maps to elliptic curves

$M : \mathbb{F}_p \to E(\mathbb{F}_p)$, where $E : y^2 = x^3 + ax + b$ and $p > 5$:

Map M		**Restrictions**	**Cost**
	[BF01]	$p \equiv 2 \bmod 3, a = 0$	1 exp
	[SW06]	none	3 exp
SWU	[Ulas07]	$p \equiv 3 \bmod 4, ab \neq 0$	3 exp
	[Icart09]	$p \equiv 2 \bmod 3$	1 exp
S-SWU	[BCIMRT10]	$p \equiv 3 \bmod 4, ab \neq 0$	2 exp
Elligator	[BHKL13]	$b \neq 0, 2 \mid \#E(\mathbb{F}_p)$	1 exp
This work		$ab \neq 0$	1 exp
		none	1^+ exp

[SS04,Ska05,FSV09,FT10a,FT10b,KLR10,CK11,Far11,FT12,FJT13,BLMP19...]

Deterministic maps to elliptic curves

$M : \mathbb{F}_p \to E(\mathbb{F}_p)$, where $E : y^2 = x^3 + ax + b$ and $p > 5$:

Map M		**Restrictions**	**Cost**
	[BF01]	$p \equiv 2 \bmod 3$, $a = 0$	1 exp
	[SW06]	none	3 exp
SWU	[Ulas07]	$p \equiv 3 \bmod 4$, $ab \neq 0$	3 exp
	[Icart09]	$p \equiv 2 \bmod 3$	1 exp
S-SWU	[BCIMRT10]	$p \equiv 3 \bmod 4$, $ab \neq 0$	2 exp
Elligator	[BHKL13]	$b \neq 0, 2 \mid \#E(\mathbb{F}_p)$	1 exp
This work		$ab \neq 0$	1 exp
		none	1^+ exp

BLS12-381: $p \equiv 1 \bmod 3$, $a = 0$, $2 \nmid \#E(\mathbb{F}_p)$

[SS04,Ska05,FSV09,FT10a,FT10b,KLR10,CK11,Far11,FT12,FJT13,BLMP19...]

Deterministic maps to elliptic curves

$M : \mathbb{F}_p \to E(\mathbb{F}_p)$, where $E : y^2 = x^3 + ax + b$ and $p > 5$:

Map M		**Restrictions**	**Cost**
	[BF01]	✗ $p \equiv 2 \bmod 3, a = 0$	1 exp
	[SW06]	none	3 exp
SWU	[Ulas07]	$p \equiv 3 \bmod 4, ab \neq 0$	3 exp
	[Icart09]	✗ $p \equiv 2 \bmod 3$	1 exp
S-SWU	[BCIMRT10]	$p \equiv 3 \bmod 4, ab \neq 0$	2 exp
Elligator	[BHKL13]	$b \neq 0, 2 \mid \#E(\mathbb{F}_p)$	1 exp
This work		$ab \neq 0$	1 exp
		none	1^+ exp

BLS12-381: $p \equiv 1 \bmod 3, \quad a = 0, \quad 2 \nmid \#E(\mathbb{F}_p)$

[SS04,Ska05,FSV09,FT10a,FT10b,KLR10,CK11,Far11,FT12,FJT13,BLMP19. . .]

Deterministic maps to elliptic curves

$M : \mathbb{F}_p \to E(\mathbb{F}_p)$, where $E : y^2 = x^3 + ax + b$ and $p > 5$:

Map M		**Restrictions**	**Cost**
	[BF01]	✗ $p \equiv 2 \bmod 3$, $a = 0$	1 exp
	[SW06]	none	3 exp
SWU	[Ulas07]	✗ $p \equiv 3 \bmod 4$, $ab \neq 0$	3 exp
	[Icart09]	✗ $p \equiv 2 \bmod 3$	1 exp
S-SWU	[BCIMRT10]	✗ $p \equiv 3 \bmod 4$, $ab \neq 0$	2 exp
Elligator	[BHKL13]	$b \neq 0, 2 \mid \#E(\mathbb{F}_p)$	1 exp
This work		$ab \neq 0$	1 exp
		none	1^+ exp

BLS12-381: $p \equiv 1 \bmod 3$, $a = 0$, $2 \nmid \#E(\mathbb{F}_p)$

[SS04,Ska05,FSV09,FT10a,FT10b,KLR10,CK11,Far11,FT12,FJT13,BLMP19...]

Deterministic maps to elliptic curves

$M : \mathbb{F}_p \to E(\mathbb{F}_p)$, where $E : y^2 = x^3 + ax + b$ and $p > 5$:

Map M		**Restrictions**	**Cost**
	[BF01]	✗ $p \equiv 2 \bmod 3, a = 0$	1 exp
	[SW06]	none	3 exp
SWU	[Ulas07]	✗ $p \equiv 3 \bmod 4, ab \neq 0$	3 exp
	[Icart09]	✗ $p \equiv 2 \bmod 3$	1 exp
S-SWU	[BCIMRT10]	✗ $p \equiv 3 \bmod 4, ab \neq 0$	2 exp
Elligator	[BHKL13]	✗ $b \neq 0, 2 \mid \#E(\mathbb{F}_p)$	1 exp
This work		$ab \neq 0$	1 exp
		none	1^+ exp

BLS12-381: $p \equiv 1 \bmod 3, \quad a = 0, \quad 2 \nmid \#E(\mathbb{F}_p)$

[SS04,Ska05,FSV09,FT10a,FT10b,KLR10,CK11,Far11,FT12,FJT13,BLMP19...]

Deterministic maps to elliptic curves

$M : \mathbb{F}_p \to E(\mathbb{F}_p)$, where $E : y^2 = x^3 + ax + b$ and $p > 5$:

Map M		**Restrictions**	**Cost**
	[BF01]	✗ $p \equiv 2 \bmod 3$, $a = 0$	1 exp
	[SW06]	✓ none	3 exp
SWU	[Ulas07]	✗ $p \equiv 3 \bmod 4$, $ab \neq 0$	3 exp
	[Icart09]	✗ $p \equiv 2 \bmod 3$	1 exp
S-SWU	[BCIMRT10]	✗ $p \equiv 3 \bmod 4$, $ab \neq 0$	2 exp
Elligator	[BHKL13]	✗ $b \neq 0, 2 \mid \#E(\mathbb{F}_p)$	1 exp
This work		$ab \neq 0$	1 exp
		none	1^+ exp

BLS12-381: $p \equiv 1 \bmod 3$, $a = 0$, $2 \nmid \#E(\mathbb{F}_p)$

[SS04,Ska05,FSV09,FT10a,FT10b,KLR10,CK11,Far11,FT12,FJT13,BLMP19...]

Deterministic maps to elliptic curves

$M : \mathbb{F}_p \to E(\mathbb{F}_p)$, where $E : y^2 = x^3 + ax + b$ and $p > 5$:

Map M		**Restrictions**	**Cost**
	[BF01]	✗ $p \equiv 2 \bmod 3, a = 0$	1 exp
	[SW06]	✓ none	3 exp
SWU	[Ulas07]	✗ $p \equiv 3 \bmod 4, ab \neq 0$	3 exp
	[Icart09]	✗ $p \equiv 2 \bmod 3$	1 exp
S-SWU	[BCIMRT10]	✗ $p \equiv 3 \bmod 4, ab \neq 0$	2 exp
Elligator	[BHKL13]	✗ $b \neq 0, 2 \mid \#E(\mathbb{F}_p)$	1 exp
This work		✗ $ab \neq 0$	1 exp
		✓ none	1^+ exp

BLS12-381: $p \equiv 1 \bmod 3, \quad a = 0, \quad 2 \nmid \#E(\mathbb{F}_p)$

[SS04,Ska05,FSV09,FT10a,FT10b,KLR10,CK11,Far11,FT12,FJT13,BLMP19...]

The Shallue–van de Woestijne map [SW06] (high level)

$$E : y^2 = f(x) = x^3 + ax + b$$

Idea #1 (Skałba): For $X_1, X_2, X_3, X_4 \neq 0$, let

$$V(\mathbb{F}_p) : f(X_1) \cdot f(X_2) \cdot f(X_3) = X_4^2$$

The Shallue–van de Woestijne map [SW06] (high level)

$$E : y^2 = f(x) = x^3 + ax + b$$

Idea #1 (Skałba): For $X_1, X_2, X_3, X_4 \neq 0$, let

$$V(\mathbb{F}_p) : f(X_1) \cdot f(X_2) \cdot f(X_3) = X_4^2$$

☞ One of $f(X_i), i \in \{1, 2, 3\}$ must be square

$\Rightarrow$ that X_i must be an x-coordinate on $E(\mathbb{F}_p)$

The Shallue–van de Woestijne map [SW06] (high level)

$$E : y^2 = f(x) = x^3 + ax + b$$

Idea #1 (Skałba): For $X_1, X_2, X_3, X_4 \neq 0$, let

$$V(\mathbb{F}_p) : f(X_1) \cdot f(X_2) \cdot f(X_3) = X_4^2$$

Idea #2: Construct a map $\mathbb{F}_p \mapsto V(\mathbb{F}_p)$, yielding polynomials $X_1(t), X_2(t), X_3(t)$.

The Shallue–van de Woestijne map [SW06] (high level)

$$E : y^2 = f(x) = x^3 + ax + b$$

Idea #1 (Skałba): For $X_1, X_2, X_3, X_4 \neq 0$, let

$$V(\mathbb{F}_p) : f(X_1) \cdot f(X_2) \cdot f(X_3) = X_4^2$$

Idea #2: Construct a map $\mathbb{F}_p \mapsto V(\mathbb{F}_p)$, yielding polynomials $X_1(t), X_2(t), X_3(t)$.

$$\mathrm{SW}(t) \triangleq \begin{cases} (X_1(t), \sqrt{f(X_1(t))}) & \text{if } f(X_1(t)) \text{ is square, else} \\ (X_2(t), \sqrt{f(X_2(t))}) & \text{if } f(X_2(t)) \text{ is square, else} \\ (X_3(t), \sqrt{f(X_3(t))}) \end{cases}$$

The Shallue–van de Woestijne map [SW06] (high level)

$$E : y^2 = f(x) = x^3 + ax + b$$

Idea #1 (Skałba): For $X_1, X_2, X_3, X_4 \neq 0$, let

$$V(\mathbb{F}_p) : f(X_1) \cdot f(X_2) \cdot f(X_3) = X_4^2$$

Idea #2: Construct a map $\mathbb{F}_p \mapsto V(\mathbb{F}_p)$, yielding polynomials $X_1(t), X_2(t), X_3(t)$.

$$\mathrm{SW}(t) \triangleq \begin{cases} (X_1(t), \sqrt{f(X_1(t))}) & \text{if } f(X_1(t)) \text{ is square, else} \\ (X_2(t), \sqrt{f(X_2(t))}) & \text{if } f(X_2(t)) \text{ is square, else} \\ (X_3(t), \sqrt{f(X_3(t))}) & \end{cases}$$

☞ constant-time cost dominated by 3 exps
(recall: Legendre symbol in $\mathbb{F}_p$ ops is 1 exp)

Hash functions from deterministic maps

Compose H_p and M in a natural way:

$\mathsf{HashToCurve}_{\mathsf{NU}}(\mathsf{msg})$:

$t \leftarrow H_p(\mathsf{msg})$ // $\{0,1\}^\star \mapsto \mathbb{F}_p$

$P \leftarrow M(t)$ // $\mathbb{F}_p \mapsto E(\mathbb{F}_p)$

return $[h]P$ // $E(\mathbb{F}_p) \mapsto \mathbb{G}$

Hash functions from deterministic maps

Compose H_p and M in a natural way:

$\mathsf{HashToCurve}_{\mathsf{NU}}(\mathsf{msg})$:

$t \leftarrow H_p(\mathsf{msg})$ // $\{0,1\}^\star \mapsto \mathbb{F}_p$

$P \leftarrow M(t)$ // $\mathbb{F}_p \mapsto E(\mathbb{F}_p)$

return $[h]P$ // $E(\mathbb{F}_p) \mapsto \mathbb{G}$

Hash functions from deterministic maps

Compose H_p and M in a natural way:

$\mathsf{HashToCurve}_{\mathsf{NU}}(\mathsf{msg})$:

$t \leftarrow H_p(\mathsf{msg})$ // $\{0,1\}^\star \mapsto \mathbb{F}_p$

$P \leftarrow M(t)$ // $\mathbb{F}_p \mapsto E(\mathbb{F}_p)$

return $[h]P$ // $E(\mathbb{F}_p) \mapsto \mathbb{G}$

Hash functions from deterministic maps

Compose H_p and M in a natural way:

$\mathsf{HashToCurve}_{\mathsf{NU}}(\mathsf{msg})$:

$t \leftarrow H_p(\mathsf{msg})$ // $\{0,1\}^\star \mapsto \mathbb{F}_p$

$P \leftarrow M(t)$ // $\mathbb{F}_p \mapsto E(\mathbb{F}_p)$

return $[h]P$ // $E(\mathbb{F}_p) \mapsto \mathbb{G}$

Hash functions from deterministic maps

Compose H_p and M in a natural way:

$\mathsf{HashToCurve}_{\mathsf{NU}}(\mathsf{msg})$:

$t \leftarrow H_p(\mathsf{msg})$ // $\{0,1\}^\star \mapsto \mathbb{F}_p$

$P \leftarrow M(t)$ // $\mathbb{F}_p \mapsto E(\mathbb{F}_p)$

return $[h]P$ // $E(\mathbb{F}_p) \mapsto \mathbb{G}$

☞ Can use a faster method for cofactor clearing:

- via endomorphisms [GLV01,SBCDK09,FKR11,BP18]
- via subgroup structure [S19 (see WB19, §5)]

Hash functions from deterministic maps

Compose H_p and M in a natural way:

$\mathrm{HashToCurve}_{\mathrm{NU}}(\mathrm{msg})$:

$t \leftarrow H_p(\mathrm{msg})$ // $\{0,1\}^\star \mapsto \mathbb{F}_p$

$P \leftarrow M(t)$ // $\mathbb{F}_p \mapsto E(\mathbb{F}_p)$

return $[h]P$ // $E(\mathbb{F}_p) \mapsto \mathbb{G}$

Possible issue: M is not a bijection: $\#E(\mathbb{F}_p) \neq p$

☞ output distribution is nonuniform

Hash functions from deterministic maps

Compose H_p and M in a natural way:

$\mathsf{HashToCurve}_{\mathsf{NU}}(\mathsf{msg})$:

$t \leftarrow H_p(\mathsf{msg})$ // $\{0,1\}^\star \mapsto \mathbb{F}_p$

$P \leftarrow M(t)$ // $\mathbb{F}_p \mapsto E(\mathbb{F}_p)$

return $[h]P$ // $E(\mathbb{F}_p) \mapsto \mathbb{G}$

Possible issue: M is not a bijection: $\#E(\mathbb{F}_p) \neq p$

☞ output distribution is nonuniform

This *could* be OK—but what if we need uniformity?

Uniform hashing from deterministic maps [BCIMRT10]

For some distinguished point $\hat{P} \in \mathbb{G}$:

$\mathsf{HashToCurve}_{\mathsf{OTP}}(\mathsf{msg})$:

$P_1 \leftarrow M(H_p(\mathsf{msg}))$

$x \leftarrow H_q(\mathsf{msg})$

$P_2 \leftarrow [x]\hat{P}$

$P \leftarrow P_1 + P_2$

return $[h]P$

Uniform hashing from deterministic maps [BCIMRT10]

For some distinguished point $\hat{P} \in \mathbb{G}$:

$\mathsf{HashToCurve}_{\mathsf{OTP}}(\mathsf{msg})$:

$P_1 \leftarrow M(H_p(\mathsf{msg}))$

$x \leftarrow H_q(\mathsf{msg})$

$P_2 \leftarrow [x]\hat{P}$

$P \leftarrow P_1 + P_2$

return $[h]P$

Uniform hashing from deterministic maps [BCIMRT10]

For some distinguished point $\hat{P} \in \mathbb{G}$:

$\mathsf{HashToCurve}_{\mathsf{OTP}}(\mathsf{msg})$:

$P_1 \leftarrow M(H_p(\mathsf{msg}))$

$x \leftarrow H_q(\mathsf{msg})$

$P_2 \leftarrow [x]\hat{P}$

$P \leftarrow P_1 + P_2$

return $[h]P$

Uniform hashing from deterministic maps [BCIMRT10]

For some distinguished point $\hat{P} \in \mathbb{G}$:

$\mathsf{HashToCurve}_{\mathsf{OTP}}(\mathsf{msg})$:

$P_1 \leftarrow M(H_p(\mathsf{msg}))$

$x \leftarrow H_q(\mathsf{msg})$

$P_2 \leftarrow [x]\hat{P}$

$P \leftarrow P_1 + P_2$

return $[h]P$

Uniform hashing from deterministic maps [BCIMRT10]

For some distinguished point $\hat{P} \in \mathbb{G}$:

$\mathsf{HashToCurve}_{\mathsf{OTP}}(\mathsf{msg})$:

$P_1 \leftarrow M(H_p(\mathsf{msg}))$

$x \leftarrow H_q(\mathsf{msg})$

$P_2 \leftarrow [x]\hat{P}$

$P \leftarrow P_1 + P_2$

return $[h]P$

☞ $[x]\hat{P}$ acts as a "one-time pad"

Uniform hashing from deterministic maps [BCIMRT10]

For some distinguished point $\hat{P} \in \mathbb{G}$:

$\mathsf{HashToCurve}_{\mathsf{OTP}}(\mathsf{msg})$:

$$P_1 \leftarrow M(H_p(\mathsf{msg}))$$
$$x \leftarrow H_q(\mathsf{msg})$$
$$P_2 \leftarrow [x]\hat{P}$$
$$P \leftarrow P_1 + P_2$$
$$\text{return } [h]P$$

- ☞ $[x]\hat{P}$ acts as a "one-time pad"
- ☞ $\mathsf{HashToCurve}_{\mathsf{OTP}}$ is *indifferentiable* from RO [MRH05]

Uniform hashing from deterministic maps [BCIMRT10]

For some distinguished point $\hat{P} \in \mathbb{G}$:

$\mathsf{HashToCurve}_{\mathsf{OTP}}(\mathsf{msg})$:

$P_1 \leftarrow M(H_p(\mathsf{msg}))$

$x \leftarrow H_q(\mathsf{msg})$

$P_2 \leftarrow [x]\hat{P}$ // ✗ expensive

$P \leftarrow P_1 + P_2$

return $[h]P$

☞ $[x]\hat{P}$ acts as a "one-time pad"

☞ $\mathsf{HashToCurve}_{\mathsf{OTP}}$ is *indifferentiable* from RO [MRH05]

Faster uniform hashing from deterministic maps

Problem: point multiplication is usually much more expensive than evaluating M.

Faster uniform hashing from deterministic maps

Problem: point multiplication is usually much more expensive than evaluating M.

Idea [BCIMRT10,FFSTV13]:

HashToCurve(msg) :

$P_1 \leftarrow M(H_p(0 \,||\, \text{msg}))$

$P_2 \leftarrow M(H_p(1 \,||\, \text{msg}))$

$P \leftarrow P_1 + P_2$

return $[h]P$

Faster uniform hashing from deterministic maps

Problem: point multiplication is usually much more expensive than evaluating M.

Idea [BCIMRT10,FFSTV13]:

HashToCurve(msg) :

$P_1 \leftarrow M(H_p(0 \,||\, \text{msg}))$

$P_2 \leftarrow M(H_p(1 \,||\, \text{msg}))$

$P \leftarrow P_1 + P_2$

return $[h]P$

Faster uniform hashing from deterministic maps

Problem: point multiplication is usually much more expensive than evaluating M.

Idea [BCIMRT10,FFSTV13]:

HashToCurve(msg) :

$$P_1 \leftarrow M(H_p(0 \,||\, \text{msg}))$$

$$P_2 \leftarrow M(H_p(1 \,||\, \text{msg}))$$

$$P \leftarrow P_1 + P_2$$

return $[h]P$

☞ Indifferentiable from RO if M is *well distributed*

✓ All of the M we've seen are well distributed.

Roadmap

1. Hash functions to elliptic curves
2. Optimizing the map of [BCIMRT10]
3. Evaluation results
4. IETF standardization efforts

The Simplified SWU map [BCIMRT10]

$$E : y^2 = f(x) = x^3 + ax + b, \qquad ab \neq 0$$

Idea: pick x s.t. $f(ux) = u^3 f(x)$.

☞ For u non-square $\in \mathbb{F}_p$, $f(x)$ or $f(ux)$ is square.

The Simplified SWU map [BCIMRT10]

$$E : y^2 = f(x) = x^3 + ax + b, \qquad ab \neq 0$$

Idea: pick x s.t. $f(ux) = u^3 f(x)$.

☞ For u non-square $\in \mathbb{F}_p$, $f(x)$ or $f(ux)$ is square.

$$u^3x^3 + aux + b = u^3(x^3 + ax + b)$$

$$\therefore \qquad x = -\frac{b}{a}\left(1 + \frac{1}{u^2 + u}\right)$$

The Simplified SWU map [BCIMRT10]

$$E : y^2 = f(x) = x^3 + ax + b, \qquad ab \neq 0$$

Idea: pick x s.t. $f(ux) = u^3 f(x)$.

☞ For u non-square $\in \mathbb{F}_p$, $f(x)$ or $f(ux)$ is square.

$$u^3x^3 + aux + b = u^3(x^3 + ax + b)$$

$$\therefore \qquad x = -\frac{b}{a}\left(1 + \frac{1}{u^2 + u}\right)$$

☞ If $p \equiv 3 \bmod 4$, $u = -t^2$ is non-square

The Simplified SWU map [BCIMRT10]

$$E : y^2 = f(x) = x^3 + ax + b, \qquad ab \neq 0$$

Idea: pick x s.t. $f(ux) = u^3 f(x)$.

☞ For u non-square $\in \mathbb{F}_p$, $f(x)$ or $f(ux)$ is square.

$$u^3x^3 + aux + b = u^3(x^3 + ax + b)$$

$$\therefore \qquad x = -\frac{b}{a}\left(1 + \frac{1}{u^2 + u}\right)$$

☞ If $p \equiv 3 \bmod 4$, $u = -t^2$ is non-square, so:

$$X_0(t) \triangleq -\frac{b}{a}\left(1 + \frac{1}{t^4 - t^2}\right) \qquad X_1(t) \triangleq -t^2 X_0(t)$$

Evaluating the S-SWU map

$$\text{S-SWU}(t) \triangleq \begin{cases} (X_0(t), \sqrt{f(X_0(t))}) & \text{if } f(X_0(t)) \text{ is square} \\ (X_1(t), \sqrt{f(X_1(t))}) & \text{otherwise} \end{cases}$$

Evaluating the S-SWU map

$$\text{S-SWU}(t) \triangleq \begin{cases} \left(X_0(t), \sqrt{f(X_0(t))}\right) & \text{if } f(X_0(t)) \text{ is square} \\ \left(X_1(t), \sqrt{f(X_1(t))}\right) & \text{otherwise} \end{cases}$$

Attempt #1 (assume $p \equiv 3 \bmod 4$):

$x_0 \leftarrow X_0(t)$

$y_0 \leftarrow f(x_0)^{\frac{p+1}{4}}$ // ✗ expensive

$x_1 \leftarrow -t^2 x_0$ // a.k.a. $X_1(t)$

$y_1 \leftarrow f(x_1)^{\frac{p+1}{4}}$ // ✗ expensive

if $y_0^2 = f(x_0)$: return (x_0, y_0)

else: return (x_1, y_1)

Evaluating the S-SWU map

$$\text{S-SWU}(t) \triangleq \begin{cases} (X_0(t), \sqrt{f(X_0(t))}) & \text{if } f(X_0(t)) \text{ is square} \\ (X_1(t), \sqrt{f(X_1(t))}) & \text{otherwise} \end{cases}$$

Attempt #1 (assume $p \equiv 3 \bmod 4$):

$x_0 \leftarrow X_0(t)$

$y_0 \leftarrow f(x_0)^{\frac{p+1}{4}}$ // ✗ expensive

$x_1 \leftarrow -t^2 x_0$ // a.k.a. $X_1(t)$

$y_1 \leftarrow f(x_1)^{\frac{p+1}{4}}$ // ✗ expensive

if $y_0^2 = f(x_0)$: return (x_0, y_0)

else: return (x_1, y_1)

Evaluating the S-SWU map

$$\text{S-SWU}(t) \triangleq \begin{cases} \left(X_0(t), \sqrt{f(X_0(t))}\right) & \text{if } f(X_0(t)) \text{ is square} \\ \left(X_1(t), \sqrt{f(X_1(t))}\right) & \text{otherwise} \end{cases}$$

Attempt #1 (assume $p \equiv 3 \bmod 4$):

$x_0 \leftarrow X_0(t)$

$y_0 \leftarrow f(x_0)^{\frac{p+1}{4}}$ // ✗ expensive

$x_1 \leftarrow -t^2 x_0$ // a.k.a. $X_1(t)$

$y_1 \leftarrow f(x_1)^{\frac{p+1}{4}}$ // ✗ expensive

if $y_0^2 = f(x_0)$: return (x_0, y_0)

else: return (x_1, y_1)

Evaluating the S-SWU map

$$\text{S-SWU}(t) \triangleq \begin{cases} (X_0(t), \sqrt{f(X_0(t))}) & \text{if } f(X_0(t)) \text{ is square} \\ (X_1(t), \sqrt{f(X_1(t))}) & \text{otherwise} \end{cases}$$

Attempt #1 (assume $p \equiv 3 \bmod 4$):

$x_0 \leftarrow X_0(t)$

$y_0 \leftarrow f(x_0)^{\frac{p+1}{4}}$ // ✗ expensive

$x_1 \leftarrow -t^2 x_0$ // a.k.a. $X_1(t)$

$y_1 \leftarrow f(x_1)^{\frac{p+1}{4}}$ // ✗ expensive

if $y_0^2 = f(x_0)$: return (x_0, y_0)

else: return (x_1, y_1)

Requires two exponentiations! Can we do better?

Eliminating an exponentiation

Recall: $f(x_1) = -t^6 f(x_0)$. So:

$$f(x_1)^{\frac{p+1}{4}} = \left(-t^6 f(x_0)\right)^{\frac{p+1}{4}}$$

Eliminating an exponentiation

Recall: $f(x_1) = -t^6 f(x_0)$. So:

$$\begin{aligned} f(x_1)^{\frac{p+1}{4}} &= \left(-t^6 f(x_0)\right)^{\frac{p+1}{4}} \\ &= t^3 \left(-f(x_0)\right)^{\frac{p+1}{4}} = t^3 \sqrt{-f(x_0)} \end{aligned}$$

Eliminating an exponentiation

Recall: $f(x_1) = -t^6 f(x_0)$. So:

$$f(x_1)^{\frac{p+1}{4}} = \left(-t^6 f(x_0)\right)^{\frac{p+1}{4}}$$
$$= t^3 \left(-f(x_0)\right)^{\frac{p+1}{4}} = t^3 \sqrt{-f(x_0)}$$

☞ We have $f(x_0)^{\frac{p+1}{4}}$. Can we use this?

Eliminating an exponentiation

Recall: $f(x_1) = -t^6 f(x_0)$. So:

$$f(x_1)^{\frac{p+1}{4}} = \left(-t^6 f(x_0)\right)^{\frac{p+1}{4}}$$
$$= t^3 \left(-f(x_0)\right)^{\frac{p+1}{4}} = t^3 \sqrt{-f(x_0)}$$

☞ We have $f(x_0)^{\frac{p+1}{4}}$. Can we use this?

$$\left(f(x_0)^{\frac{p+1}{4}}\right)^2 = f(x_0)^{\frac{p+1}{2}} = f(x_0) \cdot f(x_0)^{\frac{p-1}{2}}$$

Eliminating an exponentiation

Recall: $f(x_1) = -t^6 f(x_0)$. So:

$$f(x_1)^{\frac{p+1}{4}} = \left(-t^6 f(x_0)\right)^{\frac{p+1}{4}}$$
$$= t^3 \left(-f(x_0)\right)^{\frac{p+1}{4}} = t^3 \sqrt{-f(x_0)}$$

☞ We have $f(x_0)^{\frac{p+1}{4}}$. Can we use this?

$$\left(f(x_0)^{\frac{p+1}{4}}\right)^2 = f(x_0)^{\frac{p+1}{2}} = f(x_0) \cdot \boxed{f(x_0)^{\frac{p-1}{2}}}$$

Legendre symbol!

Eliminating an exponentiation

Recall: $f(x_1) = -t^6 f(x_0)$. So:

$$f(x_1)^{\frac{p+1}{4}} = \left(-t^6 f(x_0)\right)^{\frac{p+1}{4}}$$
$$= t^3 \left(-f(x_0)\right)^{\frac{p+1}{4}} = t^3 \sqrt{-f(x_0)}$$

☞ We have $f(x_0)^{\frac{p+1}{4}}$. Can we use this?

$$\left(f(x_0)^{\frac{p+1}{4}}\right)^2 = f(x_0)^{\frac{p+1}{2}} = f(x_0) \cdot f(x_0)^{\frac{p-1}{2}}$$
$$= -f(x_0) \qquad \text{if } f(x_0) \text{ is non-square}$$

✓ $f(x_0)^{\frac{p+1}{4}}$ is $\sqrt{-f(x_0)}$ when $f(x_0)$ is non-square!

Evaluating the S-SWU map—faster!

Attempt #2 (assume $p \equiv 3 \bmod 4$):

$x_0 \leftarrow X_0(t)$

$y_0 \leftarrow f(x_0)^{(p+1)/4}$ // ✗ expensive

$x_1 \leftarrow -t^2 x_0$ // a.k.a. $X_1(t)$

$y_1 \leftarrow t^3 y_0$ // ✓ cheap!

if $y_0^2 = f(x_0)$: return (x_0, y_0)

else: return (x_1, y_1)

Evaluating the S-SWU map—faster!

Attempt #2 (assume $p \equiv 3 \bmod 4$):

$x_0 \leftarrow X_0(t)$

$y_0 \leftarrow f(x_0)^{(p+1)/4}$ // ✗ expensive

$x_1 \leftarrow -t^2 x_0$ // a.k.a. $X_1(t)$

$y_1 \leftarrow t^3 y_0$ // ✓ cheap!

if $y_0^2 = f(x_0)$: return (x_0, y_0)

else: return (x_1, y_1)

✓ Prior work [BDLSY12] lets us avoid inversions.

Evaluating the S-SWU map—faster!

Attempt #2 (assume $p \equiv 3 \bmod 4$):

$x_0 \leftarrow X_0(t)$

$y_0 \leftarrow f(x_0)^{(p+1)/4}$ // ✗ expensive

$x_1 \leftarrow -t^2 x_0$ // a.k.a. $X_1(t)$

$y_1 \leftarrow t^3 y_0$ // ✓ cheap!

if $y_0^2 = f(x_0)$: return (x_0, y_0)

else: return (x_1, y_1)

✓ Prior work [BDLSY12] lets us avoid inversions.

✓ Straightforward to generalize to $p \equiv 1 \bmod 4$.

Generalizing: the $p \equiv 5 \bmod 8$ case

-1 is square in $\mathbb{F}_p \Rightarrow$ need $u = \xi t^2$ for ξ nonsquare.

Generalizing: the $p \equiv 5 \bmod 8$ case

-1 is square in $\mathbb{F}_p \Rightarrow$ need $u = \xi t^2$ for ξ nonsquare.

Recall Atkin's square-root trick:

$$\left(z^{\frac{p+3}{8}}\right)^2 = z \cdot \left(z^{\frac{p-1}{2}}\right)^{\frac{1}{2}}$$

Generalizing: the $p \equiv 5 \bmod 8$ case

-1 is square in $\mathbb{F}_p \Rightarrow$ need $u = \xi t^2$ for ξ nonsquare.

Recall Atkin's square-root trick:

$$\left(z^{\frac{p+3}{8}}\right)^2 = z \cdot \left(z^{\frac{p-1}{2}}\right)^{\frac{1}{2}}$$

Legendre symbol!

Generalizing: the $p \equiv 5 \bmod 8$ case

-1 is square in $\mathbb{F}_p \Rightarrow$ need $u = \xi t^2$ for ξ nonsquare.

Recall Atkin's square-root trick:

$$\left(z^{\frac{p+3}{8}}\right)^2 = z \cdot \left(z^{\frac{p-1}{2}}\right)^{\frac{1}{2}}$$

$$z^{\frac{p+3}{8}} \cdot 1^{-\frac{1}{4}} = \sqrt{z}$$

Generalizing: the $p \equiv 5 \bmod 8$ case

-1 is square in $\mathbb{F}_p \Rightarrow$ need $u = \xi t^2$ for ξ nonsquare.

Recall Atkin's square-root trick:

$$\left(z^{\frac{p+3}{8}}\right)^2 = z \cdot \left(z^{\frac{p-1}{2}}\right)^{\frac{1}{2}}$$

$$z^{\frac{p+3}{8}} \cdot 1^{-\frac{1}{4}} = \sqrt{z}$$

So we want:

$$\begin{aligned}\sqrt{f(x_1)} &= \sqrt{\xi^3 t^6 f(x_0)} \\ &= t^3 \left(\xi^3 f(x_0)\right)^{\frac{p+3}{8}} \cdot 1^{-\frac{1}{4}}\end{aligned}$$

Generalizing: the $p \equiv 5 \bmod 8$ case

-1 is square in $\mathbb{F}_p \Rightarrow$ need $u = \xi t^2$ for ξ nonsquare.

Recall Atkin's square-root trick:

$$\left(z^{\frac{p+3}{8}}\right)^2 = z \cdot \left(z^{\frac{p-1}{2}}\right)^{\frac{1}{2}}$$

$$z^{\frac{p+3}{8}} \cdot 1^{-\frac{1}{4}} = \sqrt{z}$$

So we want:

$$\sqrt{f(x_1)} = \sqrt{\xi^3 t^6 f(x_0)}$$
$$= t^3 \left(\xi^3 f(x_0)\right)^{\frac{p+3}{8}} \cdot 1^{-\frac{1}{4}}$$

☞ ξ is fixed, so we can preompute $\left(\xi^3\right)^{\frac{p+3}{8}}$

Supporting the $ab = 0$ case

Issue: S-SWU still does not work with $ab = 0$.

☞ Rules out pairing-friendly curves [BLS03,BN06,...]

Supporting the $ab = 0$ case

Issue: S-SWU still does not work with $ab = 0$.
☞ Rules out pairing-friendly curves [BLS03,BN06,...]

Idea: map to a curve E' having $ab \neq 0$ and an efficiently-computable homomorphism to E.

Supporting the $ab = 0$ case

Issue: S-SWU still does not work with $ab = 0$.
☞ Rules out pairing-friendly curves [BLS03,BN06,...]

Idea: map to a curve E' having $ab \neq 0$ and an efficiently-computable homomorphism to E.

Specifically: Find $E'(\mathbb{F}_p)$ d-isogenous to E, d small.
☞ Defines a degree $\approx d$ rational map $E'(\mathbb{F}_p) \to E(\mathbb{F}_p)$

Supporting the $ab = 0$ case

Issue: S-SWU still does not work with $ab = 0$.
☞ Rules out pairing-friendly curves [BLS03,BN06,...]

Idea: map to a curve E' having $ab \neq 0$ and an efficiently-computable homomorphism to E.

Specifically: Find $E'(\mathbb{F}_p)$ d-isogenous to E, d small.
☞ Defines a degree $\approx d$ rational map $E'(\mathbb{F}_p) \to E(\mathbb{F}_p)$

Then: S-SWU to $E'(\mathbb{F}_p)$, isogeny map to $E(\mathbb{F}_p)$.
✓ Preserves well-distributedness of S-SWU.

Roadmap

1. Hash functions to elliptic curves
2. Optimizing the map of [BCIMRT10]
3. Evaluation results
4. IETF standardization efforts

Implementation, baselines, setup, method

BLS12-381 defines $\mathbb{G}_1 \subset E_1(\mathbb{F}_p)$ and $\mathbb{G}_2 \subset E_2(\mathbb{F}_{p^2})$.

Implementation, baselines, setup, method

BLS12-381 defines $\mathbb{G}_1 \subset E_1(\mathbb{F}_p)$ and $\mathbb{G}_2 \subset E_2(\mathbb{F}_{p^2})$.

For $\mathbb{G}_1$ and $\mathbb{G}_2$, we implement:

- Maps: hash-and-check; [SW06]; this work
- Styles: full bigint; field ops only, non-CT and CT
- Hashes: non-uniform; uniform

In total: 34 hash variants, 3520 lines of C.

Implementation, baselines, setup, method

BLS12-381 defines $\mathbb{G}_1 \subset E_1(\mathbb{F}_p)$ and $\mathbb{G}_2 \subset E_2(\mathbb{F}_{p^2})$.

For $\mathbb{G}_1$ and $\mathbb{G}_2$, we implement:

- Maps: hash-and-check; [SW06]; this work
- Styles: full bigint; field ops only, non-CT and CT
- Hashes: non-uniform; uniform

In total: 34 hash variants, 3520 lines of C.

Implementation, baselines, setup, method

BLS12-381 defines $\mathbb{G}_1 \subset E_1(\mathbb{F}_p)$ and $\mathbb{G}_2 \subset E_2(\mathbb{F}_{p^2})$.

For $\mathbb{G}_1$ and $\mathbb{G}_2$, we implement:

- Maps: hash-and-check; [SW06]; this work
- Styles: full bigint; field ops only, non-CT and CT
- Hashes: non-uniform; uniform

In total: 34 hash variants, 3520 lines of C.

Setup: Xeon E3-1535M v6 (no hyperthreading or frequency scaling); Linux 5.2; GCC 9.1.0.

Implementation, baselines, setup, method

BLS12-381 defines $\mathbb{G}_1 \subset E_1(\mathbb{F}_p)$ and $\mathbb{G}_2 \subset E_2(\mathbb{F}_{p^2})$.

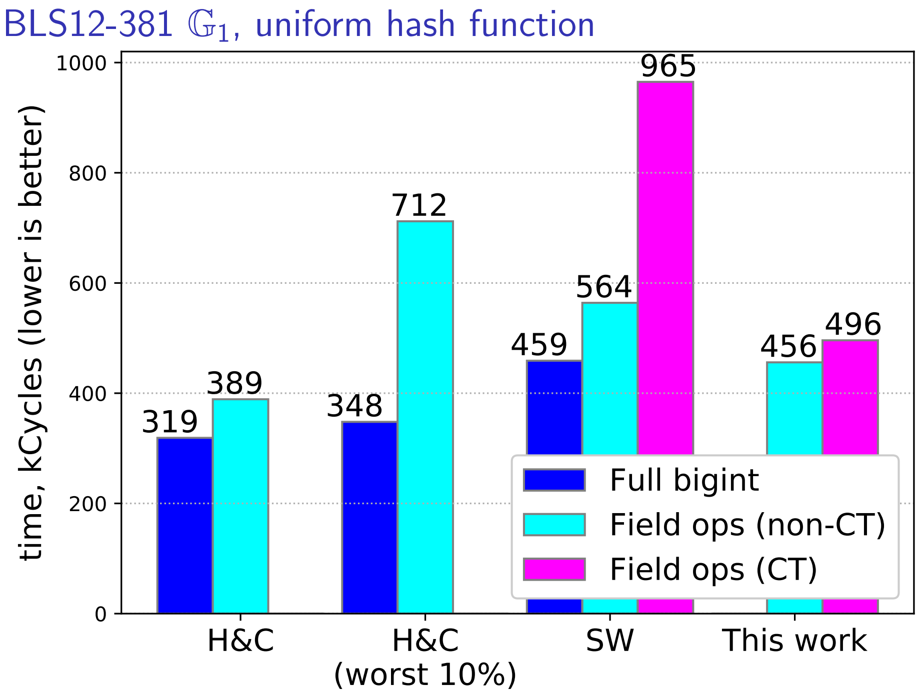
For $\mathbb{G}_1$ and $\mathbb{G}_2$, we implement:

- Maps: hash-and-check; [SW06]; this work
- Styles: full bigint; field ops only, non-CT and CT
- Hashes: non-uniform; uniform

In total: 34 hash variants, 3520 lines of C.

Setup: Xeon E3-1535M v6 (no hyperthreading or frequency scaling); Linux 5.2; GCC 9.1.0.

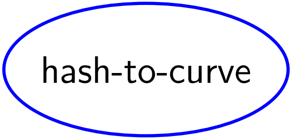
Method: run each hash 10^6 times; record #cycles.

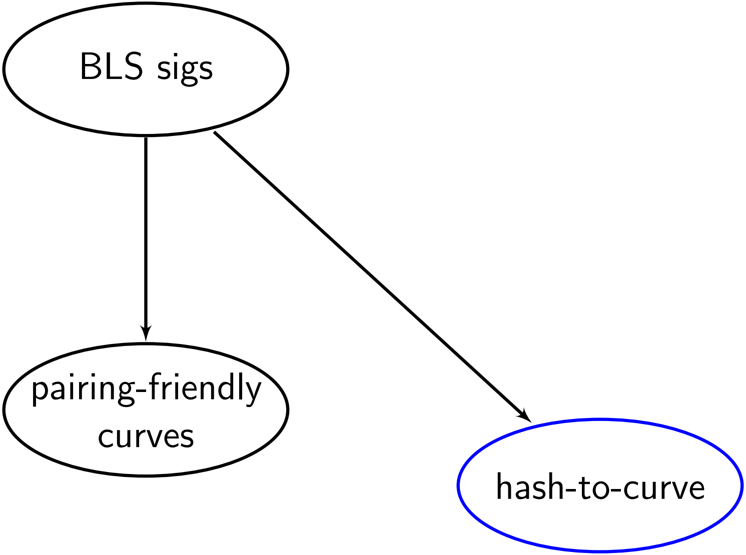


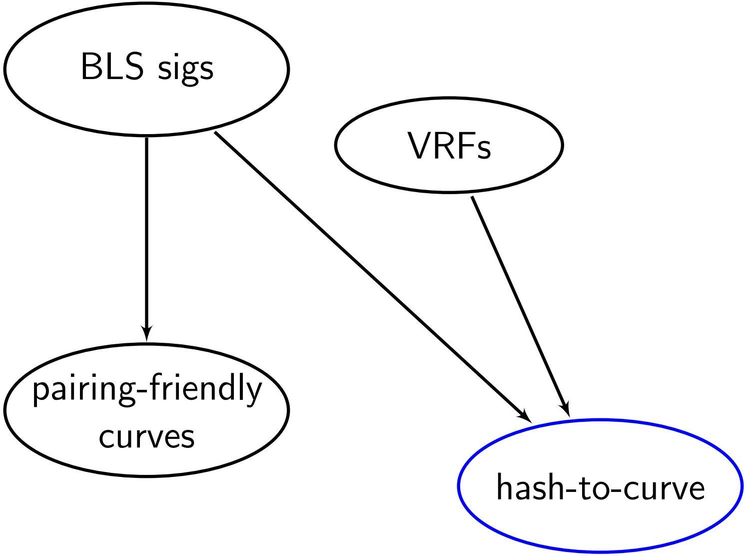
BLS12-381 $\mathbb{G}_1$, uniform hash function
time, kCycles (lower is better)
0
200
400
600
800
1000
319
389
348
712
459
564
965
456
496
H&C
H&C (worst 10%)
SW
This work
Full bigint
Field ops (non-CT)
Field ops (CT)

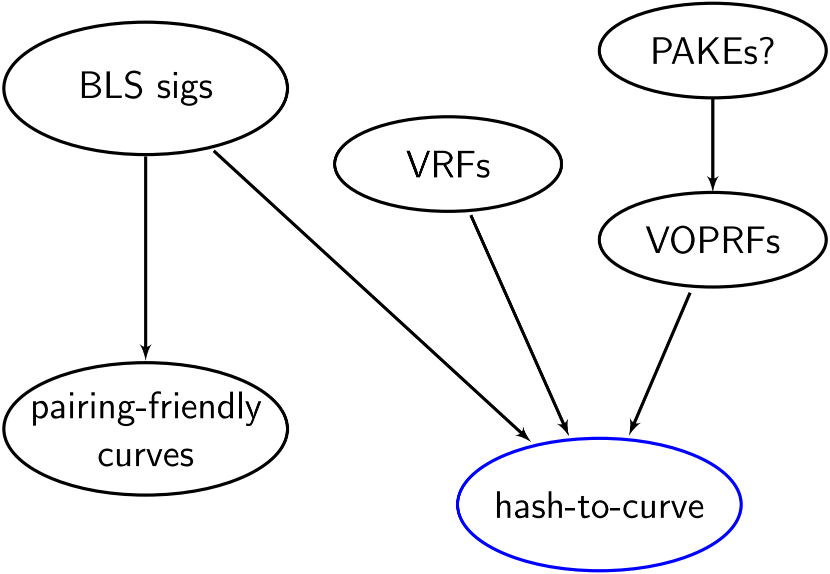
Roadmap

1. Hash functions to elliptic curves
2. Optimizing the map of [BCIMRT10]
3. Evaluation results
4. IETF standardization efforts


hash-to-curve


BLS sigs
pairing-friendly curves
hash-to-curve


BLS sigs
VRFs
pairing-friendly curves
hash-to-curve


BLS sigs
PAKEs?
VRFs
VOPRFs
pairing-friendly curves
hash-to-curve

Which maps should the IETF standardize?

$M : \mathbb{F}_p \to E(\mathbb{F}_p)$, where $E : y^2 = x^3 + ax + b$ and $p > 5$:

Map M		**Restrictions**	**Cost**
	[BF01]	$p \equiv 2 \bmod 3$, $a = 0$	1 exp
	[SW06]	none	3 exp
SWU	[Ulas07]	$p \equiv 3 \bmod 4$, $ab \neq 0$	3 exp
	[Icart09]	$p \equiv 2 \bmod 3$	1 exp
S-SWU	[BCIMRT10]	$p \equiv 3 \bmod 4$, $ab \neq 0$	2 exp
Elligator	[BHKL13]	$b \neq 0, 2 \mid \#E(\mathbb{F}_p)$	1 exp
This work		$ab \neq 0$	1 exp
		none	1^+ exp

[SS04,Ska05,FSV09,FT10a,FT10b,KLR10,CK11,Far11,FT12,FJT13,BLMP19...]

Which maps should the IETF standardize?

$M : \mathbb{F}_p \to E(\mathbb{F}_p)$, where $E : y^2 = x^3 + ax + b$ and $p > 5$:

Map M		**Restrictions**	**Cost**
	[BF01]	$p \equiv 2 \bmod 3, a = 0$	1 exp
	[SW06]	none	3 exp
~~SWU~~	~~[Ulas07]~~	~~$p \equiv 3 \bmod 4, ab \neq 0$~~	~~3 exp~~
	[Icart09]	$p \equiv 2 \bmod 3$	1 exp
S-SWU	[BCIMRT10]	$p \equiv 3 \bmod 4, ab \neq 0$	2 exp
Elligator	[BHKL13]	$b \neq 0, 2 \mid \#E(\mathbb{F}_p)$	1 exp
This work		$ab \neq 0$	1 exp
		none	1^+ exp

[SS04,Ska05,FSV09,FT10a,FT10b,KLR10,CK11,Far11,FT12,FJT13,BLMP19...]

Which maps should the IETF standardize?

$M : \mathbb{F}_p \to E(\mathbb{F}_p)$, where $E : y^2 = x^3 + ax + b$ and $p > 5$:

Map M		**Restrictions**	**Cost**
	[BF01]	$p \equiv 2 \bmod 3, a = 0$	1 exp
	[SW06]	none	3 exp
~~SWU~~	~~[Ulas07]~~	~~$p \equiv 3 \bmod 4, ab \neq 0$~~	~~3 exp~~
	~~[Icart09]~~	~~$p \equiv 2 \bmod 3$~~	~~1 exp~~
S-SWU	[BCIMRT10]	$p \equiv 3 \bmod 4, ab \neq 0$	2 exp
Elligator	[BHKL13]	$b \neq 0, 2 \mid \#E(\mathbb{F}_p)$	1 exp
This work		$ab \neq 0$	1 exp
		none	1^+ exp

[SS04,Ska05,FSV09,FT10a,FT10b,KLR10,CK11,Far11,FT12,FJT13,BLMP19. . .]

Which maps should the IETF standardize?

$M : \mathbb{F}_p \to E(\mathbb{F}_p)$, where $E : y^2 = x^3 + ax + b$ and $p > 5$:

Map M		Restrictions	Cost
	[BF01]	$p \equiv 2 \bmod 3, a = 0$	1 exp
	[SW06]	none	3 exp
~~SWU~~	~~[Ulas07]~~	~~$p \equiv 3 \bmod 4, ab \neq 0$~~	~~3 exp~~
	~~[Icart09]~~	~~$p \equiv 2 \bmod 3$~~	~~1 exp~~
~~S-SWU~~	~~[BCIMRT10]~~	~~$p \equiv 3 \bmod 4, ab \neq 0$~~	~~2 exp~~
Elligator	[BHKL13]	$b \neq 0, 2 \mid \#E(\mathbb{F}_p)$	1 exp
This work (+ tweaks to avoid infringing patents)		$ab \neq 0$	1 exp
		none	1^{+} exp

[SS04,Ska05,FSV09,FT10a,FT10b,KLR10,CK11,Far11,FT12,FJT13,BLMP19...]

Which maps should the IETF standardize?

$M : \mathbb{F}_p \to E(\mathbb{F}_p)$, where $E : y^2 = x^3 + ax + b$ and $p > 5$:

Map M		**Restrictions**	**Cost**
	[BF01]	$p \equiv 2 \bmod 3, a = 0$	1 exp
	[SW06]	none	3 exp
~~SWU~~	~~[Ulas07]~~	~~$p \equiv 3 \bmod 4, ab \neq 0$~~	~~3 exp~~
	~~[Icart09]~~	~~$p \equiv 2 \bmod 3$~~	~~1 exp~~
~~S-SWU~~	~~[BCIMRT10]~~	~~$p \equiv 3 \bmod 4, ab \neq 0$~~	~~2 exp~~
Elligator	[BHKL13]	$b \neq 0, 2 \mid \#E(\mathbb{F}_p)$	1 exp
This work (+ tweaks to avoid infringing patents)		$ab \neq 0$	1 exp
		none	1^+ exp

[SS04,Ska05,FSV09,FT10a,FT10b,KLR10,CK11,Far11,FT12,FJT13,BLMP19...]

Which maps should the IETF standardize?

$M : \mathbb{F}_p \to E(\mathbb{F}_p)$, where $E : y^2 = x^3 + ax + b$ and $p > 5$:

Map M	**Restrictions**	**Cost**
[BF01] **???**	$p \equiv 2 \bmod 3$, $a = 0$	1 exp
[SW06]	none	3 exp
~~SWU [Ulas07]~~	~~$p \equiv 3 \bmod 4$, $ab \neq 0$~~	~~3 exp~~
~~[Icart09]~~	~~$p \equiv 2 \bmod 3$~~	~~1 exp~~
~~S-SWU [BCIMRT10]~~	~~$p \equiv 3 \bmod 4$, $ab \neq 0$~~	~~2 exp~~
Elligator [BHKL13]	$b \neq 0, 2 \mid \#E(\mathbb{F}_p)$	1 exp
This work (+ tweaks to avoid infringing patents)	$ab \neq 0$	1 exp
	none	1^+ exp

☞ What about supersingular maps [BF01,BLMP19]?

[SS04,Ska05,FSV09,FT10a,FT10b,KLR10,CK11,Far11,FT12,FJT13,BLMP19...]

Recap and conclusion

Contributions:

✓ Optimizations to the map of [BCIMRT10]

✓ "Indirect" approach to expand applicability

✓ Fast impls are simple and constant time

Recap and conclusion

Contributions:

✓ Optimizations to the map of [BCIMRT10]

✓ "Indirect" approach to expand applicability

✓ Fast impls are simple and constant time

Result: hash-to-curve costs 1^+ exponentiation for essentially any prime-field elliptic curve.

Recap and conclusion

Contributions:

✓ Optimizations to the map of [BCIMRT10]

✓ "Indirect" approach to expand applicability

✓ Fast impls are simple and constant time

Result: hash-to-curve costs 1^+ exponentiation for essentially any prime-field elliptic curve.

☞ State of the art for BLS, BN, NIST, secp256k1, and other curves not covered by Elligator or Icart.

Recap and conclusion

Contributions:

✓ Optimizations to the map of [BCIMRT10]

✓ "Indirect" approach to expand applicability

✓ Fast impls are simple and constant time

Result: hash-to-curve costs 1^+ exponentiation for essentially any prime-field elliptic curve.

☞ State of the art for BLS, BN, NIST, secp256k1, and other curves not covered by Elligator or Icart.

`https://bls-hash.crypto.fyi`

`https://github.com/kwantam/bls12-381_hash`

`https://github.com/cfrg/draft-irtf-cfrg-hash-to-curve`

`rsw@cs.stanford.edu`