Hashing to Elliptic Curves

draft-irtf-cfrg-hash-to-curve

Sam Scott (sam.scott@cornell.edu)
Nick Sullivan (nick@cloudflare.com)
Christopher A. Wood (cawood@apple.com)

CFRG
IETF 102, July 2018, Montreal

Background

Goal: collate mechanisms for hashing arbitrary strings to elliptic curves

- Algorithm is used in many places (PAKEs)

Status: draft-irtf-cfrg-hash-to-curve adopted after IETF 101

- Initial version convoluted encoding, serializing, and hashing (Random Oracles)

- Detailed three algorithms with accompanying Sage implementations

Updates

- Separate encoding, serialization, and Random Oracle functionalities and sections (map2curve and hash2curve)
- Refactor algorithm recommendation table
- Add hacspec implementations of each map2curve algorithm

Encoding and ROs

map2curve: deterministically map arbitrary input to point on curve

- Algorithms include: Icart, SWU, Simple SWU, and Elligator2

Encoding and ROs

map2curve: deterministically map arbitrary input to point on curve

- Algorithms include: Icart, SWU, Simple SWU, and Elligator2

hash2curve: Random Oracle implementation

- Current hash-encode-twice due to Brier et al.:

$$\mathsf{hash2curve}(\alpha) = F(H0(\alpha)) + F(H1(\alpha))$$
$$H0(\alpha) = \mathsf{HashToBase}(0||\alpha)$$
$$H1(\alpha) = \mathsf{HashToBase}(1||\alpha)$$

- Considering replacing with hash-encode-basepoint:

$$\mathsf{hash2curve}(\alpha) = F(H0(m)) + H1(m)G$$

Recommendations

Application	Requirement	Additional Details
SPEKE [Jablon96]	Naive	H(x)*G
PAKE [BMP00]	Random Oracle	-
BLS [BLS01]	Random Oracle	-
IBE [BF01]	Random Oracle	Supersingular, pairing-friendly curve
PRF	Injective encoding	F(k, m) = k*H(m)

Algorithm selection:
Lookup requirement from table above and chosen curve from table below

Curve	Inj. Encoding	Random Oracle
P-256	Simple SWU Section 5.2.3	FFSTV(SWU)
P-384	Icart Section 5.2.1	FFSTV(Icart)
Curve25519	Elligator2 Section 5.2.4	...
Curve448	Elligator2 Section 5.2.4	...

Example

Application	Requirement	Additional Details
SPEKE [Jablon96]	Naive	H(x)*G
PAKE [BMP00]	**Random Oracle**	-
BLS [BLS01]	Random Oracle	-
IBE [BF01]	Random Oracle	Supersingular, pairing-friendly curve
PRF	Injective encoding	F(k, m) = k*H(m)

Curve	Inj. Encoding	Random Oracle
P-256	Simple SWU Section 5.2.3	**FFSTV(SWU)**
P-384	Icart Section 5.2.1	FFSTV(Icart)
Curve25519	Elligator2 Section 5.2.4	...
Curve448	Elligator2 Section 5.2.4	...

PAKEs using P-256 which require a RO should use the hash2curve construction with SWU

hacspec

Added hacspec implementations of each map2curve algorithm

- Implementations of map2curve

Open Issues

- Add support for pairing-friendly curves: https://github.com/chris-wood/draft-sullivan-cfrg-hash-to-curve/pull/20

- Write cost comparison table: https://github.com/chris-wood/draft-sullivan-cfrg-hash-to-curve/issues/5

- Write curve transformation section: https://github.com/chris-wood/draft-sullivan-cfrg-hash-to-curve/issues/3

Open Questions

- What generic hash2curve construction should we select?
- Can the document structure be improved for use?
- Is it appropriate to include hacspec in this document?

hash-to-curve - CFRG - IETF 102