Hash to curve

S. Scott, N. Sullivan, C. Wood

IETF 103 - CFRG
Nick Sullivan • 11.05.2018

Overview

Goal

Define algorithms to efficiently map from a string to a point on a curve for use in various high-level constructions (such as VRF, OPAQUE, VOPRF, etc.). Support injective and random oracle encodings.

Current Draft

draft-irtf-cfrg-hash-to-curve-02

Recent progress

- Explicit HashToBase
- Explicit recommended ciphers

Open issues

- Test vectors
- Explicit recommended ciphers

Progress - HashToBase

Explicit formula defined

- One-way hash from a string to an element of a base field of a curve
- Deterministic and constant time

Reduction of bias

- HashToBase algorithm greedily takes as many bits as possible before reducing mod p
- Reduces bias to trivial amount when appropriate hash size used

Progress - CipherSuites

Overview

- Destination Group (e.g. P256 or Curve25519)
- HashToBase algorithm
- HashToCurve algorithm (e.g. SSWU, Icart)
- (Optional) Transformation (e.g. FFSTV, cofactor clearing)

Defined

- NIST Curves (RO)
- CFRG Curves
  - Injective with cofactor clearing
  - RO with FFSTV

Open Issues

Pairing-friendly Curves

- Construction from Fouque and Tibouchi
- Ciphersuite: H2C-BN256-SHA512-FT-FFSTV
- PR under review: https://github.com/chris-wood/draft-sullivan-cfrg-hash-to-curve/pull/20

Test Vectors

- Test vectors to be defined for all ciphersuites

Open Issues

Constant-time considerations, edge cases

- SWU with p = 1 (mod 4)
- Incomplete addition law for Curve448
- A=0 case for SWU

Next steps

Complete test vectors

Independent results to be validated using Sage, Go and C implementations.

Review pairing-friendly curve algorithm

Volunteers?

Review other open issues

Constant-time considerations, edge cases.

Questions/ Discussion