draft-irtf-cfrg-cpace

STATUS IETF 109, NOVEMBER 2020

MICHEL ABDALLA, BJÖRN HAASE, JULIA HESSE

draft-irtf-cfrg-cpace
PLAN

Objectives of the editor team (Michel Abdalla, Björn Haase, Julia Hesse)

- RFC document
- Provide public code / scripts for generation of test vectors
- Provide a public reference implementation for some variants.

draft-irtf-cfrg-cpace
RFC: Identified work items

- Draft text currently refers generically to "groups" ("groups modulo negation").
  => We plan to change this such that it explicitly refers to groups on elliptic curves only.

- Currently two types of elliptic curves are considered in the draft
  - Single-coordinate Diffie-Hellman X448 and X25519 on Edwards/Montgomery curves with small co-factor
  - X-Coordinate-only Diffie-Hellman on Short-Weierstrass curves such as NIST-P256
  - We plan to add also full-group (i.e. both coordinates) support, e.g. for Ristretto25519

- Slight variations between the three alternative implementation variants, specifically regarding the properties imposed on the Map2Point algorithm.

- Explicit security analysis of all resulting slight variations considered mandatory as a first step.

`draft-irtf-cfrg-cpace`
preliminary STATUS SECURITY ANALYSIS

- Tight bounds can be established by using the approach used for TBPEKE for GAP SDH and GAP CDH problems
  - Most likely a weaker assumption set applies (Strong DH / Strong SDH problems instead of GAP CDH / GAP SDH)

- For curves with co-factor, security can be reduced to hardness of the problems on the prime-order subgroup

- Important formal requirement for the map: "For any point *G not* on the image of **`Map2Point`** there must be an efficient algorithm for finding an exponent *y* such that **`scalar_mult`**(G,*y*) *is* on the map."
  - Property provided by all currently discussed mappings from Hash2Curve and Ristretto25519 (every second exponent exponent *y* will work on average)
  - With this property the SDH problem on the subset of points produced by **`Map2Point`** can be reduced to hardness of SDH on the group (or the prime-order subgroup respectively).

- Discussion of these aspects not considered suitable for the RFC. A separate paper extensively discussing the above points is in work as a first step. (Currently the main aspect in our weekly meeting in the editor team.)

draft-irtf-cfrg-cpace
STATUS REFERENCE IMPLEMENTATIONS / TEST VECTORS

- Presently reference implementation and test vector generation for X25519/Elligator2 available
  - Separate implementations in Sagemath, Python and ANSI-C currently still held at https://github.com/BjoernMHaase/AuCPace
  - GIT-Repository under the hood of CFRG was setup
    CPace-specific code still needs to be transferred to this repository.

- Next steps: Similar reference implementation and test vector generation for Short-Weierstrass and Ristretto25519
  - At least Sagemath, preferably also generic Python and ANSI-C

Any suggestion / support regarding small self-contained libraries used for reference implementations and test-vector generation for Short-Weierstrass and Ristretto25519 implementations in ANSI-C and Python would be welcome.

draft-irtf-cfrg-cpace
SUMMARY

- We organized our team. / Unfortunately we feel slowed down significantly by the pandemic.
- Current activity: Provide tight security bounds and reduction to accurately specified assumptions for all possible the tiny implementation differences (e.g. Short-Weierstrass, X448/X25519, Ristretto25519)
- Next important step: Scripts for test vectors for all alternative implementations