Initial Review of New Elliptic Curve Multiset Hashing Proposal

John Brainard

ConsenSys - PegaSys - Sidechains Team

# Overview

A multiset hash is a hash value computed over a set of elements where each element value may have multiple instances. An Elliptic Curve multiset hash computes the hash value by mapping each element to a point on the curve, then combining the resulting points.

The basic algorithm is as follows:

1. Compute the hash value of each element in the multiset using a standard hash function
2. Map each hash value to a point on the curve
3. Combine the points to create the multiset hash

The particular scheme is defined by the choice of mapping and combining functions.

# Proposed Scheme

The proposed scheme is similar to the scheme, called ECMH, proposed by Maitin-Shepard, Tibouchi, and Aranha[1].

The only significant difference between the two schemes is the function used to map the hash values to a point on the curve. ECMH uses a deterministic mapping called Shallue van de Woestijne (SW) encoding[2]. The proposed scheme uses the simpler “Try-and-increment” method described by Icart:

Map a value to a point on the curve y2 = x3 + ax + b

“Try and increment” mapping:

1. Compute hi = hash(vi)
2. Let x = hi
3. Compute s = x3 + ax + b
4. If s is a square, go to step 7
5. Let x = x+1
6. Go to step 3
7. Let y = sqrt(x3 + ax + b)
8. Q = (x, y)
9. Return Q

# Analysis

The “try-and-increment” mapping has the advantage of simplicity over the SW encoding. One concern is that the mapping is unbounded, that is, an undetermined number of increment and test steps may be required. In practice, about half of all input values will produce a valid output after one step, one quarter will require two steps, and so on. This could lead to a side channel attack based on the execution time of the mapping. This can be mitigated by always doing a fixed number of steps before returning, but then some fraction (2-k where k is the number of steps) of values will not return a valid point.

The collision resistance demonstrated for ECMH depends only on the properties of the target elliptic curve, and not on the specifics of the value-to-point mapping, so the proposed scheme should have the same 2m/2 resistance to collisions for an m-bit hash function.

# References

[1] Maitin-Shepard, Tibouchi, and Aranha, “Elliptic Curve Multiset Hash”, <https://pdfs.semanticscholar.org/3068/d1d6275933e7f4d332a2f2cf52543a4f0615.pdf>, arxiv, Jan. 2016

[2] Shallue, A. and van de Woestijne, C. E. , “Construction of rational points on elliptic curves over finite fields”, *ANTS*, pp. 510–524, Springer 2006

[3] Icart, “How to Hash to Elliptic Curves”, <https://eprint.iacr.org/2009/226.pdf>, Crypto 2009