Call for Proposals: Hybrid Cycle of SNARK-friendly Curves

IOG

1 Executive Summary

Input Output Global (IOG), the powerhouse behind the Cardano blockchain and ecosystem, is interested in a cryptographically secure and efficient hybrid cycle of curves. While Pluto-Eris is an option to be considered, we require that the proposed curve (be it Pluto-Eris or a different curve cycle) is thoroughly analyzed for its security and efficiency properties. This call for proposals is looking for cryptographers/researchers to conduct analysis and recommend/prove (see requirements and details herein) efficient curves for use by IOG. This is an opportunity to be a part of the latest cutting-edge research into cryptography and zero-knowledge proof (ZKP) development, directly impacting the blockchain space. Successful delivery will have such a provider be considered for future research opportunities.

The deadline for submitting proposals is July 7, 2023, which will be followed by a review process and provider selection during July. Actual work is expected to start by late July and be completed no later than the end of September 2023. Proposals should include details related to the effort required and the costs that IOG will cover for the selected provider. All other details will be negotiated as per the details herein.

All communications to be directed to Vanishree Rao at vanishree.rao@iohk.io and Markulf Kohlweiss markulf.kohlweiss@iohk.io.

2 Introduction

IOG is interested in a cryptographically-secure and efficient hybrid cycle of curves. In this call for proposals, we specify the high-level motivation and the projected impact of this work, the feature and security properties of the curves and the format and acceptance criteria of work delivery.

2.1 The Motivation

IOG has incubated a protocol that aims to build a zero-knowledge proof system supporting recursion, aka proofs of proofs, based on Halo2 [hal]. Halo2 is currently based on the pasta cycle of curves with neither of the two curves being pairing friendly. This prevents the usage of efficient (succinctly verifiable) polynomial commitment schemes, such as the KZG scheme [KZG10a], that need pairing-friendly curves. This has motivated the search for a cycle of curves where at least one of the curves is pairing friendly.

While we do not exclude proposals where both curves in the cycle are pairing-friendly, we focus the remainder of the discussion on “hybrid” cycle of curves, where only one of the curves is pairing-friendly. The motivation here is that the relaxation of constraints on the curves results in more efficient curves.

An efficient hybrid cycle of curves will be a significant contribution to the blockchain space that utilizes zero-knowledge proofs. This is because, it can help Halo2, one of the most widely used zero-knowledge proof systems, to achieve constant verification complexity (w.r.t. the circuit size). A non-constant verification complexity is one of the biggest pain points in various applications, including bridges and zk rollups [bar, VOm*].

The outcome of this work is aimed to affect the critical core part of the protocol.
3 Requirements on the Hybrid Cycle of Curves

This section specifies the properties that need to be satisfied by the hybrid cycle of curves. A formal definition of a hybrid cycle of curves can be found in Appendix A.

We will present the security and feature requirements below.

Pluto/Eris is an example of such a cycle. Rather than recommending the use of a new hybrid curve, a successful delivery may recommend the usage of Pluto/Eris, but additional analysis, documentation and proof of concept deliverables to justify the curve choice always need to be included as specified below.

3.1 Security Requirements

Security level. The Pollard Rho security level for the non-pairing friendly curve needs be at least 128 bits. The STNFS cost according to [GS2019] must estimate the security level of the pairing friendly curve to be at least 128 bits.

Twist security. Both curves must have a twist security of at least 128 bits.

Sampling process. The curves need to be sampled deterministically through a script using the feature requirements and security requirements mentioned in this document; any other constraint in the script needs to be approved by IOG.

SafeCurves. Whilst all Safecurves [saf] criteria are highly desirable, it might not be possible to satisfy some of them. In that case, it is required to provide a best-possible alternative with the same overall functionality. Such exceptions will also need to be approved by IOG.

We describe some possible deviations in the following, further deviations need a formal approval from IOG to be acceptable.

- Elligator 2. If Elligator 2 [BHKL13] cannot be used, indistinguishability must be efficiently achievable through Elligator Squared [Tib14].
- Ladder support. Ladder support needn’t be satisfied.
- Embedding degree. The embedding degree criterion is not required for the pairing-friendly curve.
- Completeness. Complete formulas are desirable. An investigation of efficient scalar multiplication and multi-scalar multiplication (natively and inside proofs) is acceptable.

3.2 Feature Requirements

Efficiency. Performance of elliptic curve operations for the operations implemented here [plu] and field sizes must be comparable to or be better than Pluto-Eris [plu].

2-adicity. Both curves must have a 2-adicity of at least 32.

j-invariant. Both curves must have j-invariant = 0.

Hash to curve. It is highly desirable to have efficient hash to curve algorithms, such as the “simplified SWU” method [BCI+10] or extensions such as [WB19].

Low Hamming weight for the BN parameter \( u \). If the BN parameter \( u \) has low Hamming weight, the pairing computation can be efficient. Hence, it is desirable that the pairing-friendly curve has a low Hamming weight BN parameter.

4 Deliverables

The deliverables consist of:

1. A detailed report containing a thorough security and feature analysis, addressing every requirement mentioned above.
2. Proof of concept implementation in Rust using the Arkworks [ark] library of all group operations including $G_1, G_2, G_T$ and pairing operations for the pairing friendly curve, curve operations for $E_1$ and $E_2$ and hash to curve for both the curves. The implementation should include unit tests and performance tests with sufficient test vectors.

3. The proposal must include a bid and a description of the team executing the proposal. (We estimate that the project is executable in 5 months by an experienced elliptic curve cryptography expert.)

4.1 Acceptance Criteria

1. Any deviation from the security or feature properties must be formally approved by IOG. An example scenario includes the researcher(s) finding candidate curves with significantly better efficiency properties upon a reasonable and calculated security requirement relaxation. This approval process can occur either during the proposal submission/review process or during the proposal execution process.

2. The sampling script should enforce the feature and security requirements.

3. The report should detail the performance and security analysis.

4. The code needs to be properly documented.

5 Timeline

- Deadline for proposal submission: July 11, 2023 at 11:59PM EST
- Q&A + provider selection: July 12-21, 2023
- Contract drafting + signing: July 24-28, 2023
- Beginning of the project execution: July 31, 2023

Appendix

A Definition of Hybrid Cycle of Curves

We use the definition of 2-cycles of elliptic curves from [AHG22, Definition 1], and combine this with the existence of a pairing on one of the two curves. Specifically:

Definition 1 A hybrid cycle of curves is a pair of elliptic curves $E_1/F_{p_1} \& E_2/F_{p_2}$, such that:

- $\#(E_1/F_{p_1}) = p_2$ and $\#(E_2/F_{p_2}) = p_1$
- $E_1/F_{p_1}$ is pairing friendly.

Other definitions may be found in [BSCTV17], or [EH22, Chapter 4.3].

We follow the conventions of [GPS06]. We refer to $E_2/F_{p_2}$ as $G$ and to $E_1/F_{p_1}$ as $G_1$. For $G_1$ to be pairing friendly, there needs to exist another source group $G_2$ and a target group $G_T$ as well as an efficient, non-degenerate pairing operation $e : G_1 \times G_2 \rightarrow G_T$.

Such curves are particularly useful for compact recursive proofs, relying on polynomial commitments using the Inner Product Argument [BCC+16] over $G_1$ and $G$ for unbounded recursion, and using the pairing on $G_1$ and $G_2$ for a [KZG10b]-based polynomial commitment for compression.
References


