A Review of Soundness of Divisors-based Proofs

Brandon Goodell, Rigo Salazar, Freeman Slaughter, Luke Szramowski

Cypher Stack

17 March 2025

As with any such report, this document may contain errors, and we cannot guarantee its correctness or security. Further, no particular implementation of the construction is in the scope of this review, and we make no guarantees about the correctness, security, or suitability for intended use cases. The authors further express no endorsement of any kind. This report has not undergone any further formal or peer review.

Contents

Change Log

This document may be updated occasionally, especially if security-sensitive results come to light. We summarize such changes here.

- 17 March 2025. Initial upload to GitHub.
- 15 April 2025. Remove redundant descriptions of mathematics presented in [Bas24b], rewrite to clarify conclusions.
- 23 May 2025. Add change log and minor modifications to wording.

1 Introduction

We review "Soundness Proof for an Interactive Protocol for the Discrete Logarithm Relation" ([Bas24b]) by Alp Bassa at Veridise, which intends to formalize the protocol informally described in [Eag22] and establish its soundness. This can also be viewed as a formalization and improvement upon the gadget presented in [Kaya]. The intended application is full-chain membership proofs (FCMP++, [Kayb]) for privacy-respecting cryptocurrency protocols.

The scope of our review is restricted to [Bas24b], but information from [Eag22], [Bas24a], and [Bas24c] are necessarily discussed, along with insights obtained from personal communications with Luke "Kayaba" Parker regarding the intended use-case in Monero.

2 Executive Summary

We find that the results and proofs in [Bas24b] are plausible. However, the paper contains both minor and major issues, some of which may fall outside of its original scope. Notably, several of these issues pose nontrivial challenges to the validity of the approach presented in [Eag22], and these concerns should be carefully addressed before considering on-chain deployment.

Please note that our judgment on the parts of [Bas24b] which may require refinement are likely to differ from the opinions of other authors. We emphasize that we have no knowledge of the scope imposed on [Bas24b]. While we believe our list to be fair, we do not believe it to be exhaustive. It is additionally our belief that a thorough revision of the report addressing the issues highlighted in this document could uncover potential flaws yet undiscovered in the protocol.

3 Major and Minor Issues

3.1 Major Issues

1. Clear definitions at the beginning of a manuscript clarify writing. Defining all terms at the start of a paper helps readers understand the arguments, avoid confusion, follow the reasoning without searching for definitions elsewhere, as well as mitigates ambiguity between references with different notation. This is true even for standard definitions. [Bas24b] provides references in the literature instead, leaving it to the reader to cross-reference the proofs within against the definitions contained in other references.

2. Explanations for the objectives of the manuscript are warranted. The intended use-case of the protocol investigated in [Bas24b] is not described, and reasonable presumptions by readers are significant sources of confusion. We elaborate on this below in Section 4.1.

3. Explanations of the benefits to using the described protocol instead of other methods are necessary. We elaborate on this below in Section 4.2.

4. Since witnesses are revealed directly, the protocol appears to be trivially complete and trivially sound. Explanations of the security definitions being investigated are required.

   - It is not clear to us how the incompleteness of the protocol purported in [Bas24b] and [Eag22] can be reconciled with the ostensible trivial completeness obtained from the revealed witness.
   - We are not convinced that the notion of soundness demonstrated in this document is necessary or sufficient for intended use-cases, as the elaborate extractor which requires $13kq$ transcripts seems to be easily replaced with an extractor which reads the witness directly from the transcript.

   We elaborate on this and more below in Section 4.3.

5. [Bas24b] provides no description for the zero-knowledge context within which this protocol is intended to be deployed. We are skeptical that context can be ignored. We discuss this in Section 4.4.

6. Without a formal proof, it is possible that the extractor in [Bas24b] runs in non-polynomial time or succeeds only with negligible probability. We discuss this in Section 4.5.

7. Simplifications should be avoided if possible, even though they are expedient. We suspect that the proofs in [Bas24b] can be improved across the board by avoiding these abridgments. For example, instead of relaxing the Hasse bound to $n = O(q)$, the inequality $(\sqrt{q} - 1)^2 \leq n \leq (\sqrt{q} + 1)^2$ can be used instead, revealing important computational factors.

8. Asymptotic results are not concrete results. We discuss this further in Section 4.6.

9. The security of this protocol is not truly based on the discrete logarithm problem: given G and P, recover a such that $P = a \cdot G$. Rather, it is a *generalized* discrete logarithm problem (some authors call this the *vector form* of the discrete logarithm problem, others reserve "generalized" for a different problem altogether): given $G, B_1, \ldots, B_k$ and P, recover $a, s_1 \ldots, s_k$ such that $P = a \cdot G + s_1 \cdot B_1 + \cdots + s_k \cdot B_k$. Over $\mathbb{F}_p$, we do not expect that the generalized discrete logarithm problem can be solved faster than the standard version, but an analysis should be presented assuring the reader that even the state of the art naive algorithms have an intractable runtime.

3.2 Minor Issues

1. In the main protocol, the verifier's first action is to sample $A_0, A_1 \xleftarrow{\$} E$, where $\xleftarrow{\$}$ indicates samples taken uniformly at random. Later, in Corollary 6, the knowledge error is calculated as

$$\kappa \leq \frac{13kq - 1}{(\#E(\mathbb{F}_q))^2}$$

ie: that the protocol is $13kq$-out-of-$(\#E(\mathbb{F}_q))^2$-special-sound. The Hasse bound provides the bounds $q - 2\sqrt{q} + 1 \leq \#E(\mathbb{F}_q) \leq q + 2\sqrt{q} + 1$, so we have

$$\frac{13kq - 1}{(q + 2\sqrt{q} + 1)^2} \leq \frac{13kq - 1}{(\#E(\mathbb{F}_q))^2} \leq \frac{13kq - 1}{(q - 2\sqrt{q} + 1)^2}.$$

In fact, we can only guarantee a knowledge error $\kappa = O(\frac{13kq-1}{(q-2\sqrt{q}+1)^2})$. Of course, this is $O(q^{-1})$ as claimed by Bassa. However, the tightness of this bound and the value of (k, q) are both critical to practical, concrete security, yet not investigated. However, due to the abort constraints, the sample set cardinality is strictly smaller than $(\#E(\mathbb{F}_q))^2$. Indeed, even only demanding that the points be distinct will shrink the set slightly. This compounds the concrete worsening of κ. Assessing practical impact requires more precise analysis.

We agree informally that the challenge set size is large and very close to $(\#E(\mathbb{F}_q))^2$. We agree informally that the stated knowledge errors for the extractor built in [Bas24b] are reasonable approximations of the true values. We agree informally that the conditions causing an abort are sufficiently rare as to not present a practical problem. However, these details ought to be filled in and formalized. The total abort probability should be formally and precisely computed (and this is necessary to precisely compute the acceptance error).

2. In Sec. 2.3, the exponent is represented in "base-2" notation, which takes the form

$$a = \sum_{i=0}^{k} s_i \cdot 2^i$$

for $s_i \in \mathbb{Z}$. Even with the assumption that $0 \leq s_i < p$, this is not actually the base-2 representation. This type of representation is *highly* malleable, giving a great many distinct but equivalent representations, even if restricted to $\mathbb{F}_p$. For instance over $\mathbb{Z}$, or $\mathbb{F}_p$ for p large enough, we can represent 4 as

$1 \cdot 2^2$, or $2 \cdot 2^1$, or $4 \cdot 2^0$, resulting in vector representations of (100), (020), and (004) respectively. The point of these s_i's, which is unfortunately never stated, is that these values will be selected in order to optimize certain constraints in the R1CS presented in [Kaya]. Any future readers who attempt to naively implement the results of [Eag22] would be at a disadvantage - because the purpose of permitting such a wide spread of allowed s_i values is not clear, they may select parameters that are far from optimization, thus negating some of the computational benefit of the protocol.

3. Restrictive hypotheses placed on provers and verifiers may help clarify our problems with using n-of-m-special-soundness. Consider the binary decomposition of the discrete logarithm $\mathbf{s}$, and the point P, which are both learned by verifiers in the protocol for $\mathcal{R}_{DL}$.

   - An extractor for this protocol as-is can use $\mathbf{s}$ from the transcript to compute $a = \sum_{i=0}^{k} s_i \cdot 2^i$, and then can use a to compute $Q = a \cdot G$, and check if $Q = P$, yielding trivial soundness.
   - In the Generic Group Model, the prover has access to an oracle through which all group operations must be computed: $\mathcal{O}(G_1, G_2) = G_1 + G_2$. The transcript of the prover in the GGM reveals a (sequence of) oracle queries $(G_{i,1}, G_{i,2})$ terminating in the output of P, and this sequence of queries corresponds to an observation of a computation of an arithmetic circuit. From several such transcripts of queries, an n-of-m-special-soundness extractor may determine the discrete logarithm of P with respect to G, say that $P = b \cdot G$. Then, it may check if $a = b$ instead.

   We suspect that the difference between these methods may explain some of the gap between what seems to be trivial soundness and nontrivial n-of-m-special-soundness.

4. In [Bas24b], terminology is rather abused, and the document is not self-contained. This is unsurprising, as it is one of several white papers referencing another white paper by a distinct author. However, new terminology is introduced without explanation, like "effective divisor" and "zero divisor," leaving their intention implicit. We suspect, but are not certain of, the following collisions:

   - Bassa's zero divisors seem to be Eagen's divisors of degree zero.
   - Bassa's effective divisors seem to be Eagen's divisors with non-zero degree.

   However, a divisor is usually defined as *effective* when all coefficients in the sum are positive, whereas the context used requires a *principal divisor*, which is a *difference between effective divisors of equal degree* as in [Sil09]. Bassa's analysis, without a thorough discussion on terminology and notation, is ambiguous. Clarification would be valuable.

5. In Sec. 4, a minor modification is proposed, and it is stated that the difference is merely aesthetic. This modification is never elaborated upon, but it may actually complicate the protocol unnecessarily. The current protocol specifically checks that $D(A_i)$ is non-zero, then computes $h_i = \frac{D'(A_i)}{D(A_i)}$, which necessarily involves computing $D(A_i)^{-1}$. It seems more concise to simply compute the inverse using the Extended Euclidean Algorithm, which will then productively fail when $D(A_i) = 0$. We suggest that this be investigated a bit further before production development.

6. Other modifications of the protocol such as the one discussed in the previous point may yield moderately better efficiency. For example, it is not clear whether gains can be made by sampling two challenge points A_0, A_1 and interpolate a line, or to sample one challenge point A_0 and a slope λ. Such choices impact the choice of verification equations, which may in turn have impacts on overall protocol efficiency.

7. In Sec. 3.3, the first four conditions (i-iv) are sensible, and represented by abort conditions in the protocol in a one-to-one manner. However, condition (v) has the following issues:

   - This condition does not appear to be explicitly addressed in the abort conditions stated in the protocol.

- This condition is the only one without an analysis that computes the number of excluded cases, so it is not obvious that the number of cases covered here is negligible.
- In the same vein, Sec. 3.2 claims that "the proportion of not representable divisors will be negligible," but does not prove this.

3.3 Typos

1. Sec 3.1 has an extra q term in the Hasse bound. Correctly stated, the bound is $|n - (q+1)| \leq 2\sqrt{q}$.
2. Sec. 2.3 incorrectly states "the witness is only known to the verifier." As the vector s is the first thing communicated by the prover to the verifier, both parties possess knowledge of the witness.
3. Lemma 4, Eqn. (5) has a typo: the expression needs $L^{(j)}$.
4. Sec. 3.1 on page 2 has a typo: $D(x,y) = a(x) - y \cdot b(\underline{x})$.
5. In Lemma 3, it is never explained what $\mathbb{A}$ represents. While not strictly a typo, we believe this was left over after other material got cut.
6. Sec 3.3 has a typo in condition (v): *non-representability.*

4 Issues in More Detail

4.1 Objectives

The relation $\mathcal{R}_{DL}$ being investigated in [Bas24b] is stated to be a discrete logarithm relation, relating $a \in \mathbb{Z}$ to elliptic curve group point P whenever $P = a \cdot G$. So, it is natural to surmise that the protocol is intended to be a zero-knowledge proof of knowledge of a witness-statement pair in $\mathcal{R}_{DL}$.

This is not so. Indeed, the interactive protocol asks the prover to reveal the binary expansion of a in the first round of interaction, so the protocol is not a zero-knowledge proof of knowledge. In [Eag22], Eagen emphasizes the difference between the elliptic curve over which statements are being proven and the elliptic curve over which the proofs are being computed, but readers benefit from greater explanation.

In a personal communication with Luke "Kayaba" Parker, we were informed that this protocol is intended to demonstrate $P = a \cdot G$ without the verifier needing to compute $a \cdot G$ from a and G. That is to say, the protocol produces a proof of verifiable computation of P, and that that proving satisfaction of the arithmetic circuit for this protocol is to be done in zero-knowledge using the Bulletproofs proving system from [BBB$^+$17].

The problem at hand is sufficiently complicated that omitting this context represents a subtle but persistent source of confusion for readers.

4.2 Benefits

The first round of interaction in the protocol reveals the binary expansion of the discrete logarithm and the polynomials defining the "divisor witness." Hence, the protocol is both a trivially complete and trivially sound proving system, as extracting the binary expansion and recomputing a requires no special effort. If any verifier can extract a directly from any accepting transcript and confirm that $P = a \cdot G$ via usual scalar multiplication, what does this protocol improve upon? If this extraction is not sufficient, it is not clear why multiple transcripts with a common initial message and distinct challenges are necessary.

4.3 Security Definitions

The protocol studied in [Bas24b] is trivially sound and complete, as the witness is revealed in the first round of interaction. Because of this, the protocol cannot be zero-knowledge. Readers would therefore benefit from clarifying which security definitions the protocol proving the relation $\mathcal{R}_{DL}$ is claimed to satisfy.

In [Bas24b], all the following are mentioned, but none are defined: knowledge soundness, n-of-m-special-soundness, and witness-extended emulation. That the witness is immediately revealed, and therefore the protocol is trivially knowledge sound and complete, prompts many questions, some of which we outline here:

- Is the proof of n-of-m-special-soundness necessary?
- Are the additional transcripts used only to check that the extracted witness satisfies the desired relation?
- If the prior point holds, what is preventing the extractor from merely computing $a \cdot G$ directly?
- Are the conclusions on soundness in [Bas24b] phrased in terms of witness-extended emulation to account for the fact that the witnesses are revealed?
- If so, does the definition of witness-extended emulation capture the intended phenomena?
- It seems that the proof is intended to demonstrate n-of-m-special-soundness, but then the resulting corollary claims witness-extended emulation, and the relationship between these two definitions is not expanded upon.
- How can we reconcile the purported incompleteness from rewinding in [Bas24b] with the fact that the witness is directly revealed in every honest transcript?

4.4 Zero-Knowledge Context

Via personal communication with Luke “Kayaba” Parker, we learned that this protocol will, in practice, be executed inside a Bulletproofs [BBB$^+$17] wrapper. This is intended to make the proofs zero-knowledge. This context is not explained in [Eag22], [Bas24a], [Bas24b], or [Bas24c], so it is understandable that we missed this context.
However, we fail to see how wrapping the protocol with a Bulletproof solves all the issues presented.

- It remains to be justified that this gadget, once wrapped in a Bulletproof, will work as a proof of knowledge in real-world applications, whether zero-knowledge or otherwise.
- To the best of our knowledge, completeness is a requirement for a protocol that will end up inside a Bulletproofs wrapper. The protocol described in [Bas24b] has a non-zero completeness error, but also reveals its witnesses directly. Reconciling these facts is not trivial, and not explained sufficiently in either [Bas24b] or [Eag22].

4.5 Runtime and Acceptance Probability

Without a formal proof, it is possible that the extractor in [Bas24b] runs in non-polynomial time or succeeds only with negligible probability. Runtime and acceptance probability of the extractor constructed to establish soundness should be computed (at least asymptotically, if not more concretely) and it should be checked that the extractor behaves as expected.

Indeed, when the general forking algorithm is employed to obtain multiple transcripts for a large q, then the runtime of a $13kq$-out-of-$(\#E(\mathbb{F}_q))^2$ soundness extractor is $O(q)$ but has success probability which decreases exponentially in q. Since q is polynomial in the security parameter λ, the extractor runs in linear time, but seems to succeed only negligibly. Without runtime and acceptance probability data, the proof of soundness in [Bas24b] is not finished, even as simply a sketch.

4.6 Asymptotics and Concrete Results

The analysis in [Bas24b] seems asymptotically correct in the limit as the security parameter $\lambda \to \infty$, for a prime-valued q which is polynomial in λ. However, relying on the proofs in [Bas24b] for some fixed q without analyzing the asymptotic error may result in an incorrect impression of the true behavior. For example, Corollary 6 ends with the claim that $\kappa \sim \frac{13k}{q}$, but this may be tightened as follows:

$$\kappa \leq \frac{13kq}{(\#E(\mathbb{F}_q))^2} \leq \frac{13kq}{(q-2\sqrt{q}+1)^2} \leq \frac{13kq}{(\sqrt{q}-1)^4} = \frac{13k}{q-4\sqrt{q}-\frac{4}{\sqrt{q}}+\frac{1}{q}+6} \approx \frac{13k}{q-4\sqrt{q}}$$

While this may appear pedantic, the additive $4\sqrt{q}$ factor is non-negligible, so should be included. As an example, the constant $10^{10^{10}}$ is certainly $O(1)$, and the function $q - 10^{10^{10}}$ is certainly $O(q)$, but neglecting the constant term obscures the true behavior for a concrete choice of q. The same may be said for the completeness error:

$$\delta \leq \frac{3kq}{(\#E(\mathbb{F}_q))^2} \leq \frac{3k}{q-4\sqrt{q}-\frac{4}{\sqrt{q}}+\frac{1}{q}+6} \approx \frac{3k}{q-4\sqrt{q}}.$$

References

[Bas24a] Alp Bassa. On the Use of Logarithmic Derivatives in Eagen's Proof of Sums of Points. https://repo.getmonero.org/-/project/54/uploads/bfe9f49326a843ef1c9466e30a5d42c8/VAR_Monero_Logarithmic_Derivatives_Final.pdf, 2024. Accessed: 03-Mar-2025.

[Bas24b] Alp Bassa. Soundness Proof for an Interactive Protocol for the Discrete Logarithm Relation. https://moneroresearch.info/259, 2024. Accessed: 02-Apr-2025.

[Bas24c] Alp Bassa. Soundness Proof for Eagen's Proof of Sums of Points. https://repo.getmonero.org/-/project/54/uploads/eb1bf5b4d4855a3480c38abf895bd8e8/Veridise_Divisor_Proofs.pdf, 2024. Accessed: 03-Mar-2025.

[BBB+17] Benedikt Bünz, Jonathan Bootle, Dan Boneh, Andrew Poelstra, Pieter Wuille, and Greg Maxwell. Bulletproofs: Short Proofs for Confidential Transactions and More. Cryptology ePrint Archive, Paper 2017/1066, 2017. https://eprint.iacr.org/2017/1066.

[Eag22] Liam Eagen. Zero Knowledge Proofs of Elliptic Curve Inner Products from Principal Divisors and Weil Reciprocity. Cryptology ePrint Archive, Paper 2022/596, 2022. https://eprint.iacr.org/2022/596.

[Kaya] Kayaba. A R1CS Gadget for a 2^k-bit Scaling of a Fixed Generator in 7 Multiplicative Constraints. https://github.com/kayabaNerve/fcmp-plus-plus-paper/blob/divisor-paper/divisors.pdf. Accessed: 01 April 2025.

[Kayb] Kayaba. FCMP++. https://github.com/kayabaNerve/fcmp-plus-plus-paper/blob/divisor-paper/fcmp%2B%2B.pdf. Accessed: 03 April 2025.

[Sil09] Joseph H Silverman. *The arithmetic of elliptic curves*, volume 106. Springer, 2009.