Veridise

Notes on the R1CS Gadget for Providing Discrete Logarithm Proofs

Alp Bassa

Veridise

Eagen [3, Section 3] provides an efficient interactive proof for Elliptic Curve Inner Products (ECIP). More precisely, a protocol is introduced to provide (zero-knowledge) proofs for a collection of points summing to zero. It is immediate to see that the general ECIP can be reduced to a proof of zero sum by expressing each scalar in an appropriate basis and expanding. The idea for obtaining a proof of zero sum is to interpolate the points using a regular function on the elliptic curve, i.e., use a regular function as a witness to this fact. For a regular function, its zeroes sum to zero, and conversely any set of points (with multiplicities) summing to zero form the zeroes of a regular function of the degree given by the sum of the multiplicities. Hence the task is reduced to the proving that the witness function vanishes precisely at the given points with appropriate multiplicity.

This is done using an interactive proof, which adds one round of interaction. After the prover commits to the regular function, the verifier provides a challenge, which consists of an index 3 rational subfield of the function field of the elliptic curve, or, in geometric terms) a degree 3 map to a projective line. Rather than proving that the regular function vanishes at the points in question, the task of the prover is reduced to convincing the verifier that the pushforward (norm) of the regular function vanishes at the images of the points.

As the pushforward is univariate, its equality with the univariate function with the prescribed roots is easily checked (with high probability) by evaluating both at a random point. The random challenge provided by the verifier includes not only the index 3 subfield, but also this point of evaluation. There are some further intricacies regarding the random choices and the corresponding distribution. For details we refer to [5].

The document *FCMP++* by Luke "Kayaba" Parker [4] provides several gadgets, which are arithmetic circuits abstracting certain reusable components. The description of the circuits follow [1, 2] and generalizations. In particular, circuits are described in terms of Hadamard products and linear constraints.

The gadget subject to this note is given in [4, Section 4.4.2] and concerns the proof of knowledge of discrete logarithm. It is based on aforementioned construction by Eagen.

Given an elliptic curve E defined over $\mathbb{F}_p$ and a fixed point $G \in E$ of order q (the generator), the gadget yields a proof system for the relation

$$\{(P, s') \in E \times [0..2^{256} - 1] \mid s' \cdot G = P\}. \tag{1}$$

Here s' is an integer with $0 \leq s' < 2^{256}$. In fact rather than s' its binary representation $(s_{255} \dots s_0)$ is used. Denoting the coordinates of P by (x, y), we obtain the relation

$$\{(x, y, s_0, \dots, s_{255}) \in \mathbb{F}_p^2 \times \mathbb{Z}_2^{256} | (x, y) \in E \wedge \sum_{i=0}^{255} s_i \cdot 2^i \cdot G = (x, y)\} \tag{2}$$

(note that this notation is slightly problematic as the point at infinity is not represented as (x, y)). The s_i are in fact not restricted to be bits, so we get the relation

$$\{(x, y, s_0, \ldots, s_{255}) \in \mathbb{F}_p^2 \times \mathbb{S}^{256} | (x, y) \in E \wedge \sum_{i=0}^{255} s_i \cdot 2^i \cdot G = (x, y)\}, \tag{3}$$

wehere $\mathbb{S}$ denotes the set of scalars of the elliptic curve. Here $\sum_{i=0}^{255} s_i \cdot 2^i \cdot G = (x, y)$ is equivalent to $\sum_{i=0}^{255} s_i \cdot 2^i \cdot G - (x, y) = 0$, or equivalently $\sum_{i=0}^{255} s_i \cdot 2^i \cdot G + (x, -y) = 0$. Denoting $2^i \cdot G$ by G_i, the prover wants to show that

$$\sum_{i=0}^{255} s_i \cdot G_i + (x, -y) = 0 \tag{4}$$

for which [3] can be used. The gadget introduced in [4] closely follows [3] with this described modification.

Here a few minor issues and pointers to potential dangers we observed during the time limited study of the document:

Sign in expression in RHS of equation: There seems to be a small issue with the signs in the last two rows of the definition of $\text{DiscreteLog}_\text{G}$. In notation of [3], we have $Z = y - \lambda \cdot x$, and the RHS of Equation (1) is given by a sum over $(\mu - Z(P_i))^{-1} = (\mu - (P_i.y - \lambda \cdot P_i.x))^{-1}$. Hence in notation of [4], f corresponds to the term evaluated at $(x, -y)$, and the sum on the last line runs over evaluations of the term at $(G_i.x, G_i.y)$. The last two lines should read

$$f = \text{Inverse}(\mu - (-y - \delta \cdot x)) \tag{5}$$

$$\text{Equality}(\ldots, \sum_{i=0}^{255} (\mu - (G_i.y - \delta \cdot G_i.x))^{-1} \cdot s_i + f) \tag{6}$$

(note the sign in front of the $\delta \cdot \bullet$ terms). These and similar bugs can be avoided to a large extend by incorporating a larger set of tests in the final implementation.

Further details and context information for hash function: The protocol is rendered non-interactive using the Fiat-Shamir transform. Verifier challenges are replaced by hashes of the corresponding commitments. Some more details about this step might be useful. In particular, hashes should be fed with appropriate context information (the elliptic curve used, the generator for the cyclic group, etc.)[1] to reduce the attack surface.

Potential negative coefficients in divisor: The results in [3] are for divisor sums with positive coefficients and the corresponding witness function has only poles at infinity. The multiplicities of all other points appearing with nonzero coefficient in the divisor are implicitly assumed to be positive. In the discrete logarithm gadget applies to sums of the form $Q + \sum_i s_i \cdot G_i$ where $Q = -(x, y)$ and s_i are the bits in the binary expansion of s', with $s' \cdot G = (x, y)$. As however the s_i are not constraint to be bits, they can also be negative. This discrepancy from the original paper merits some additional caution.

[1] see for instance `https://blog.trailofbits.com/2022/04/18/the-frozen-heart-vulnerability-in-plonk/`

References

[1] Bootle, J., Cerulli, A., Chaidos, P., Groth, J., Petit, C., *Efficient Zero-Knowledge Arguments for Arithmetic Circuits in the Discrete Log Setting*. In: Fischlin, M., Coron, JS. (eds) Advances in Cryptology – EUROCRYPT 2016. EUROCRYPT 2016. Lecture Notes in Computer Science, vol 9666. Springer, Berlin, Heidelberg.

[2] Bünz, B., Bootle, J., Boneh, D., Poelstra, A., Wuille, P., Maxwell, G., *Bulletproofs: Short Proofs for Confidential Transactions and More*, 2018 IEEE Symposium on Security and Privacy (SP), San Francisco, CA, USA, 2018, pp. 315–334.

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

[4] Parker, L., *FCMP++*, May 9, 2024.

[5] Bassa, A., *Soundness Proof for Eagen's Proof of Sums of Points*, Veridise Technical Report.