MONERO RESEARCH LAB

TECHNICAL NOTE MRL-0010

Discrete logarithm equality across groups

Sarang Noether*
Monero Research Lab
December 4, 2018

Abstract

This technical note describes an algorithm used to prove knowledge of the same discrete logarithm across different groups. The scheme expresses the common value as a scalar representation of bits, and uses a set of ring signatures to prove each bit is a valid value that is the same (up to an equivalence) across both scalar groups.

1 Notation

We use the shorthand notation $\mathbb{Z}_n$ to mean the group $\mathbb{Z}/n\mathbb{Z}$. Let $\mathbb{G}$ and $\mathbb{H}$ be prime-order groups where the discrete logarithm problem is assumed to be hard: for example, `secp256k1` or the l-subgroup of `curve25519`. Let $G, G' \in \mathbb{G}$ and $H, H' \in \mathbb{H}$ be generators of their respective groups. Suppose $|G| = p$ and $|H| = q$. Let $\mathrm{H}_{\mathbb{G}} : \{0,1\}^* \to \mathbb{Z}_p$ and $\mathrm{H}_{\mathbb{H}} : \{0,1\}^* \to \mathbb{Z}_q$ be cryptographic hash functions.

Without loss of generality, assume $p \leq q$. Choose $x \in \mathbb{Z}$ such that $0 \leq x < p$. By considering the natural projections $\mathbb{Z} \to \mathbb{Z}_p$ and $\mathbb{Z} \to \mathbb{Z}_q$ with this domain restriction, there is a bijection between elements of $\mathbb{Z}_p$ and the restriction of $\mathbb{Z}_q$. Given this, we wish to prove that, given only the values xG' and xH' (and other proof elements as needed), the discrete logarithm of each is a representation of the same integer. In particular, we do not wish to reveal x to the verifier.

Since there is no meaningful map assumed between the two groups, our approach is to decompose x into bits, treating each bit as a scalar in both $\mathbb{Z}_p$ and $\mathbb{Z}_q$ using our equivalence, and generate commitments to each bit in both groups. For each bit, we will construct a Schnorr-type ring signature showing that the bit commitment is valid and the same value in each group.

This method was originally proposed publicly by Andrew Poelstra.

2 Algorithm

2.1 Prover

Given an integer $0 \leq x < p$, express in bits:

$$x = \sum_{i=0}^{n-1} b_i 2^i$$

Note that because of the equivalence discussed above, each b_i may be considered as an element of either $\mathbb{Z}_p$ or $\mathbb{Z}_q$ as needed, leading to a representation of x in each group. For each $i \in [0, n-2]$, generate random blinders $r_i \in \mathbb{Z}_p$ and $s_i \in \mathbb{Z}_q$. For $i = n-1$, set blinders

$$r_{n-1} = (2^{n-1})^{-1} \sum_{i=0}^{n-2} r_i 2^i \in \mathbb{Z}_p$$

*sarang.noether@protonmail.com

and

$$s_{n-1} = (2^{n-1})^{-1} \sum_{i=0}^{n-2} s_i 2^i \in \mathbb{Z}_q$$

to ensure that $\sum_{i=0}^{n-1} r_i 2^i = \sum_{i=1}^{n-1} s_i 2^i = 0$.

For each $i \in [0, n-1]$, use the blinders to compute two Pedersen commitments:

$$\begin{aligned} C_i^G &:= b_i G' + r_i G \in \mathbb{G} \\ C_i^H &:= b_i H' + s_i H \in \mathbb{H} \end{aligned}$$

Because of this construction, the weighted commitment sums are $\sum_{i=0}^{n-1} 2^i C_i^G = xG'$ and $\sum_{i=0}^{n-1} 2^i C_i^H = xH'$ in their respective groups.

We next construct a ring signature on each bit to show it is either 0 or 1, and that the value is the same (up to our equivalence) in both groups. Specifically, for each $i \in [0, n-1]$, we consider two cases:

Case: $b_i = 0$. Choose random $j_i \in \mathbb{Z}_p$ and $k_i \in \mathbb{Z}_q$. Set

$$\begin{aligned} e_{1,i}^G &:= \mathrm{H}_{\mathbb{G}}\left(C_i^G, C_i^H, j_i G, k_i H\right) \in \mathbb{Z}_p \\ e_{1,i}^H &:= \mathrm{H}_{\mathbb{H}}\left(C_i^G, C_i^H, j_i G, k_i H\right) \in \mathbb{Z}_q \end{aligned}$$

and choose random $a_{0,i} \in \mathbb{Z}_p$ and $b_{0,i} \in \mathbb{Z}_q$. Set

$$\begin{aligned} e_{0,i}^G &:= \mathrm{H}_{\mathbb{G}}\left(C_i^G, C_i^H, a_{0,i} G - e_{1,i}^G (C_i^G - G'), b_{0,i} H - e_{1,i}^H (C_i^H - H')\right) \in \mathbb{Z}_p \\ e_{0,i}^H &:= \mathrm{H}_{\mathbb{H}}\left(C_i^G, C_i^H, a_{0,i} G - e_{1,i}^G (C_i^G - G'), b_{0,i} H - e_{1,i}^H (C_i^H - H')\right) \in \mathbb{Z}_q \end{aligned}$$

and then define:

$$\begin{aligned} a_{1,i} &:= j_i + e_{0,i}^G r_i \in \mathbb{Z}_p \\ b_{1,i} &:= k_i + e_{0,i}^H s_i \in \mathbb{Z}_q \end{aligned}$$

Case: $b_i = 1$. Choose random $j_i \in \mathbb{Z}_p$ and $k_i \in \mathbb{Z}_q$. Set

$$\begin{aligned} e_{0,i}^G &:= \mathrm{H}_{\mathbb{G}}\left(C_i^G, C_i^H, j_i G, k_i H\right) \in \mathbb{Z}_p \\ e_{0,i}^H &:= \mathrm{H}_{\mathbb{H}}\left(C_i^G, C_i^H, j_i G, k_i H\right) \in \mathbb{Z}_q \end{aligned}$$

and choose random $a_{1,i} \in \mathbb{Z}_p$ and $b_{1,i} \in \mathbb{Z}_q$. Set

$$\begin{aligned} e_{1,i}^G &:= \mathrm{H}_{\mathbb{G}}\left(C_i^G, C_i^H, a_{1,i} G - e_{0,i}^G C_i^G, b_{1,i} H - e_{0,i}^H C_i^H\right) \in \mathbb{Z}_p \\ e_{1,i}^H &:= \mathrm{H}_{\mathbb{H}}\left(C_i^G, C_i^H, a_{1,i} G - e_{0,i}^G C_i^G, b_{1,i} H - e_{0,i}^H C_i^H\right) \in \mathbb{Z}_q \end{aligned}$$

and then define:

$$\begin{aligned} a_{0,i} &:= j_i + e_{1,i}^G r_i \in \mathbb{Z}_p \\ b_{0,i} &:= k_i + e_{1,i}^H s_i \in \mathbb{Z}_q \end{aligned}$$

The proof is the tuple $\left(xG', xH', \{C_i^G\}, \{C_i^H\}, \{e_{0,i}^G\}, \{e_{0,i}^H\}, \{a_{0,i}\}, \{a_{1,i}\}, \{b_{0,i}\}, \{b_{1,i}\}\right)$.

2.2 Verifier

Given a proof tuple, we first ensure the bit commitments faithfully represent the discrete logarithm commitments by checking that the following equations hold:

$$\begin{aligned}\sum_{i=0}^{n-1} 2^i C_i^G &= xG' \in \mathbb{G} \\ \sum_{i=0}^{n-1} 2^i C_i^H &= xH' \in \mathbb{H}\end{aligned}$$

For each $i \in [0, n-1]$, compute the following:

$$\begin{aligned}e_{1,i}^G &:= \mathrm{H}_{\mathbb{G}}\left(C_i^G, C_i^H, a_{1,i}G - e_{0,i}^G C_i^G, b_{1,i}H - e_{0,i}^H C_i^H\right) \in \mathbb{Z}_p \\ e_{1,i}^H &:= \mathrm{H}_{\mathbb{H}}\left(C_i^G, C_i^H, a_{1,i}G - e_{0,i}^G C_i^G, b_{1,i}H - e_{0,i}^H C_i^H\right) \in \mathbb{Z}_q \\ (e_{0,i}^G)' &:= \mathrm{H}_{\mathbb{G}}\left(C_i^G, C_i^H, a_{0,i}G - e_{1,i}^G (C_i^G - G'), b_{0,i}H - e_{1,i}^H (C_i^H - H')\right) \in \mathbb{Z}_p \\ (e_{0,i}^H)' &:= \mathrm{H}_{\mathbb{H}}\left(C_i^G, C_i^H, a_{0,i}G - e_{1,i}^G (C_i^G - G'), b_{0,i}H - e_{1,i}^H (C_i^H - H')\right) \in \mathbb{Z}_q\end{aligned}$$

Check that $(e_{0,i}^G)' = e_{0,i}^G$ and $(e_{0,i}^H)' = e_{0,i}^H$ from the proof tuple.

If all of these checks are successful, the verifier accepts the proof. Otherwise, it rejects the proof. The verifier is assumed to have also checked each proof tuple element to ensure it belongs to the expected group, to account for a malicious prover.