

# libzerocoin User Guide

Ian Miers, Christina Garman and Matthew Green

## 1 Introduction

This manual describes `libzerocoin`, an implementation of the cryptographic components of the Zerocoin protocol.

## 2 Using `libzerocoin`

The `libzerocoin` library is designed to integrate with a Bitcoin/Litecoin style client, and performs the base cryptographic operations necessary to integrate Zerocoin with the client. These operations include generation/verification of coins, as well as generation/verification of spend signatures. Roughly speaking, the use of Zerocoin proceeds according to the following steps:

1. **Parameter setup.** All Zerocoin clients in a deployment must share a single parameter  $N$  where  $N$  is a 2048-3072 bit modulus such that  $N = p * q$  where  $p$  and  $q$  are large safe prime numbers (i.e.,  $p = 2p' + 1$ ,  $q = 2q' + 1$  for primes  $p', q'$ ). Once  $N$  has been generated, the underlying values  $p, q, p', q'$  can and should be destroyed.

In addition to  $N$ , all clients must agree on a security level  $k$  (an integer  $\geq 80$ ), as well as a canonical value of one zerocoin (measured in the underlying currency).

2. **Coin generation.** To Mint a zerocoin, a client first generates a new coin  $c$  using operations in the `libzerocoin` library.

Once the coin is Minted, the client must now format and transmit a `ZEROCOIN_MINT` transaction to the network, using routines not present in `libzerocoin`. This transaction is similar to a normal Bitcoin/Litecoin transaction: it consists of inputs combining to the value of one zerocoin. Unlike a standard transaction, this transaction does not provide any outputs. Instead it simply embeds the Zerocoin value  $c$ .