CRYPTONOTE STANDARD 006 Nicolas van Saberhagen
Category: Main Track Seigen
Johannes Meier
Richard Lem
November 2012

CryptoNote One-Time Keys

Abstract

This document is part of the CryptoNote Standards describing a peer-to-peer anonymous payment system. It defines the exact method of achieving the unlinkability property of transactions: the anonymity of receivers. Unique keys are generated for each payment via modified Diffie-Hellman protocol [DH].

Copyright and License Notice

Table of Contents

1. Introduction 2
2. Definitions 2
3. Data Types and Accessory Functions 3
4. The Scheme 3
5. References 4

1. Introduction

CryptoNote utilizes peer-to-peer transactions to transfer asset ownership between anonymous users. The sender collects references to the money he is willing to redistribute into an array of inputs. Then, he generates a number of public keys that can be recognized by the receiver and puts them into an array of outputs, together with the corresponding sums.

Every user remains anonymous as long as his public payment address has no connection with his real identity and no information contained in transactions leads to that address. Therefore, each output's public key must be unlinkable to the receiver's address: no third party should be able to derive the address from the output key and vice versa.

CryptoNote's solution is based on one-time keys which the sender derives from random data and the receiver's address. Upon receiving a transaction, user scans all output keys and checks if he can recover the corresponding secret key. He succeeds if and only if that particular output was sent to his address.

2. Definitions

address: a textual representation of a user's public keys required to make a payment

input: a reference to an asset owned by the sender prior to the transaction

output: a new record of ownership of an asset transferred by the transaction

public key: a datum used to identify a peer for the purpose of digital signature verification

secret key: data known to a peer only, which enables him to create digital signatures under his identity

transaction: a single record of assets ownership transfer

3. Data Types and Accessory Functions

CryptoNote's signature scheme uses Curve25519 (see [CURVE]) as the underlying group. Group elements are encoded in the same way as in Ed25519 (see [ED25519]).

The hash function H is the same Keccak function that is used in CryptoNote. When the value of the hash function is interpreted as a scalar, it is converted into a little-endian integer and taken modulo l.

4. The Scheme

Output key generation:

- Sender selects a sum S that he wants to transfer to the address (A,B).

- He generates a random integer r modulo l and computes a value R = r*G. R is called tx_pubkey.

- Then he computes a one-time public key P = H(r*A || n)*G + B, where n is the index of the output in the transaction, encoded as varint (see section 3 of [CNS003]).

- The pair (S, P) is put in the transaction output. Refer to [CNS004] for more information on CryptoNote transactions.

- The value R is put in the Extra field. See [CNS005].

Secret key recovery:

- Receiver checks every output of every transaction to find if it was sent to his address.

- For every output he computes P' = H(a*R || n)*G + B.

- If P' is equal to the output key P, that output was sent to him.

- The corresponding secret key x for P (i.e. such that P = x*G) can be computed as follows: x = H(a*R || n) + b.

Where || denotes concatenation. The general scheme for output key generation and secret key recovery is provided in the figure below.


```
                +--------+                +-----------+
                | Sender |                |  Receiver |
                +----+---+                +-----+-----+
                     |                          |
                     |            +-----------+------------+
                     |            | A, a <- generate_key() |
                     |            | B, b <- generate_key() |
                     |            +-----------+------------+
                     |                          |
                     |           A, B           |
                     |<-------------------------|
                     |                          |
        +------------+-----------+              |
        | R, r <- generate_key() |              |
        | P <- H(r*A || n)*G+B   |              |
        +------------+-----------+              |
                     |                          |
                     |           R, P           |
                     |------------------------->|
                     |                          |
                     |            +-----------+------------+
                     |            | P' <- H(a*R || n)*G+B  |
                     |            | if P = P' then:        |
                     |            |     x <- H(a*R || n)+b |
                     |            +-----------+------------+
                     |                          |
                     .                          .
```

Figure 4: One-time keys in CryptoNote transactions

5. References

[CNS003] "CryptoNote Blockchain", CryptoNote Standard 003, September 2012.

[CNS004] "CryptoNote Transactions", CryptoNote Standard 004, September 2012.

[CNS005] "CryptoNote Transaction Extra Field", CryptoNote Standard 005, October 2012.

[CURVE] Bernstein, D. J., "Curve25519: new Diffie-Hellman speed records", 2006, http://cr.yp.to/ecdh/curve25519-20060209.pdf.

[DH] Diffie, W., and M. Hellman, "New Directions in Cryptography", 1976.

[ED25519] Bernstein, D. J., Duif, N., Lange, T., Schwabe, P., and B.-Y. Yang, "High-speed high-security signatures", 2011, http://ed25519.cr.yp.to/ed25519-20110926.pdf.