CRYPTONOTE STANDARD 007
Category: Main Track

Antonio M. Juarez
Oscar Norton
Neocortex
Albert Werner
CryptoNote
November 2012

CryptoNote Keys and Addresses

Abstract

This document is part of the CryptoNote Standards describing a peer-to-peer anonymous payment system. It defines various types of user keys used in CryptoNote and the way the addresses are encoded into alphanumeric strings.

Table of Contents

1. Introduction 2
2. Definitions 2
3. Address and Keys 2
4. Address Serialization 3
5. CryptoNote Base58 4
6. References 5

1. Introduction

CryptoNote Standards can refer to the term "keys" in two cases:

1) One-time private and public keys used in transactions. txout_to_key is the basic transaction type in CryptoNote. See [CNS004] for details.

2) User's permanent keys stored in personal CryptoNote wallets.

They are used for checking incoming transaction and deriving one-time private keys for ring signatures.

Each user has two pairs of (private, public) permanent keys by default. The public parts of the keys are represented as user address.

2. Definitions

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

3. Address and Keys

In order to send money, one needs two public keys corresponding to the two secret keys the receiver possesses. These keys allow him to send unlinkable payments (see [CNS002] and [CNS006] for details).

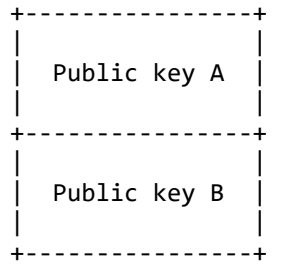

Figure 3a: Full address for unlinkable payments

When serialized to a string, these keys become a human-friendly CryptoNote address.

Spend key is a pair of secret keys corresponding to the public keys in a CryptoNote address. Spend key is required to sign transactions.

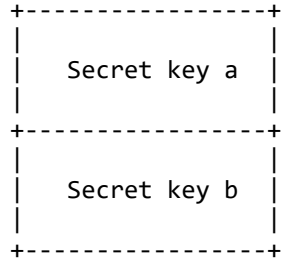

Figure 3b: Spend key

View key consists of one secret key corresponding to a public key in a CryptoNote address and one public key from the same address. View key allows its holder to identify incoming transactions, circumventing unlinkability. Said key does not allow the holder to spend any funds.

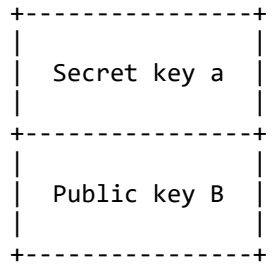

Figure 3c: View key

CryptoNote keys are Ed25519 keys encoded into 32 bytes. See [ED25519].

4. Address Serialization

CryptoNote addresses are serialized to strings using CryptoNote Base58 encoding scheme. The data encoded consists of three parts:

- public address prefix (used to distinguish the addresses of different currencies),
- a pair of public keys,

- checksum.

The pseudo-code below defines the process of generating an address:

```
Checksum = H(Varint(Prefix) || A || B)[0..3]
SerializedString = Base58(Prefix || A || B || Checksum)
```

H here is the Keccak function that is used in CryptoNote, and || denotes concatenation. For varint (variable-length encoding of integers) description see section 3 of [CNS003].

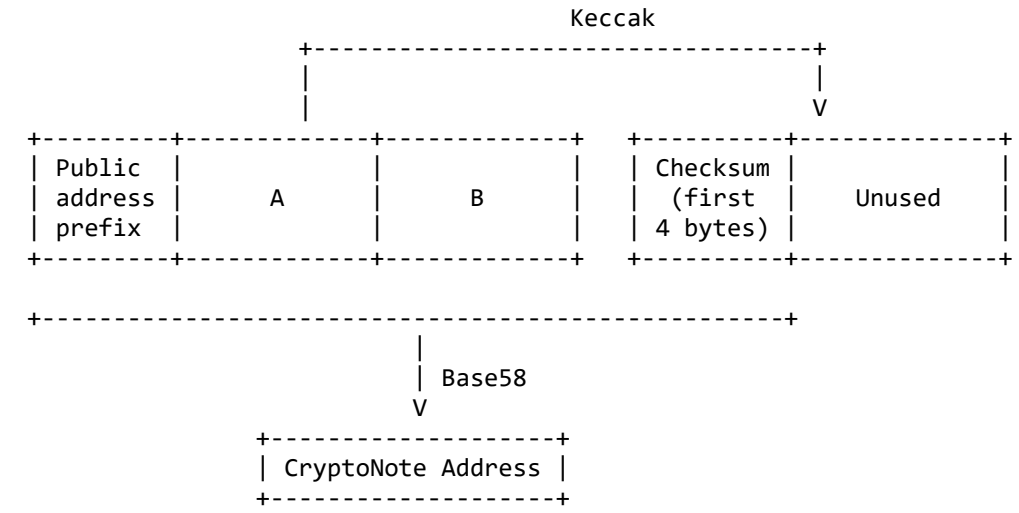

Figure 4: Address serialization algorithm

5. CryptoNote Base58

CryptoNote Base58 is a binary-to-text encoding scheme used to represent arbitrary binary data as a sequence of alphanumeric characters.

CryptoNote Base58 uses the following alphabet:

```
123456789ABCDEFGHJKLMNPQRSTUVWXYZabcdefghijkmnopqrstuvwxyz
```

The input is split into 8-byte blocks. The last block may be smaller than 8 bytes. Each block is interpreted as a big-endian integer, converted into base 58 (again, big-endian), and encoded using the alphabet shown above. The number of base-58 digits used to encode a block is the smallest number of digits sufficient to encode every

block of the same size. For example, 8-byte blocks are encoded using 11 characters.

6. References

[CNS002] "CryptoNote Signatures", CryptoNote Standard 002, May 2012.

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

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

[CNS006] "CryptoNote One-Time Keys", CryptoNote Standard 006, November 2012.

[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.