Primary- / Sub- (i ≥ 1)

OFF-CHAIN (payee's data)

Spend keys

Common key derivation method

"capped" 256 bit seed → **Mnemonic** phrase of 24 + 1 (checksum) words, among 1626 ($1626^{24} \gtrsim 2^{256}$)

H_s

- Primary Private: s_0 · G → **Primary Public:** $S_0 = s_0 G$
- Sub Public: $S_i = H_s(\text{"SubAddr"} \,|\, v_0 \,|\, i)\, G + S_0$ ← · G — $S_i = H_s(\text{"SubAddr"} \,|\, v_0 \,|\, i) + s_0$ **Sub Private**

View keys

- Primary Private: v_0 · G → **Primary Public:** $V_0 = v_0 G$
- Sub Public: $V_i = v_0 S_i$ ← · G — $v_i = v_0 s_i$ **Sub Private**

· v_0 (Sub Public) / · v_0 (Sub Private)

Actually never used: subaddresses have been designed to share v_0 usage (for blockchain-scanning performance reasons)

Addresses

- Primary: $\text{Base58}(\underbrace{0x12 \,|\, S_0 \,|\, V_0}_{\text{4 bytes-truncated Keccak256 hash}} \,|\, \text{checksum}) = \text{"4 ....."}$ [95 chars]
- Sub: $\text{Base58}(\underbrace{0x2A \,|\, S_i \,|\, V_i}_{\text{4 bytes-truncated Keccak256 hash}} \,|\, \text{checksum}) = \text{"8 ....."}$ [95 chars]

Integrated addresses

8 byte-compact paymentID, encrypted in paying transaction (vs 32 byte former one)

$\text{Base58}(\underbrace{0x13 \,|\, S_0 \,|\, V_0 \,|\, \text{payID}}_{\text{4 bytes-truncated Keccak256 hash}} \,|\, \text{checksum}) = \text{"4 ....."}$ [106 chars]

N/A because integrated addresses and sub-addresses solve somewhat the same problem

From https://monerodocs.org/public-address/integrated-address/ :

" […] **Individuals** should prefer **subaddresses** to receive payments. This is to improve privacy in certain scenarios. See article on subaddresses for details.
Businesses accepting payments in an automated way should prefer **integrated addresses**. The rationale is as follows: […] "

ON-CHAIN (by payer's initiative)

Transaction keys

- Primary Private: r · G → **Primary Public:** $R = r G$
- Sub Public: $R = r S_i$ ← · S_i — r **Sub Private**

Stealth Addresses (t ≥ 0)

Payer's POV

From payer's POV, the private key isn't known because the address we are dealing with is the transaction destination, the recipient going to be paid by the payer

- Primary Public: $X_t = H_s(r\, V_0 \,|\, t)\, G + S_0$ — $r\,(v_0 G)$ ||| $v_0\,(r G)$
- Sub Public: $X_t = H_s(r\, V_i \,|\, t)\, G + S_i$ — $r\,(v_0 S_i)$ ||| $v_0\,(r S_i)$

Identities hold true thanks to Diffie-Hellman-like connection of r, R, v_0, V_0, V_i

From payer's POV, the private key isn't known because the address we are dealing with is the transaction destination, the recipient going to be paid by the payer

Payee's POV

Used for Ring Signatures when payee will in turn become payer spending this UTXO

$x_t = H_s(v_0 \,R \,|\, t) + s_0$ · G → $X_t = H_s(v_0 \,R \,|\, t)\, G + S_0$ ← $X_t = H_s(v_0 \,R \,|\, t)\, G + S_i$ ← · G — $x_t = H_s(v_0 \,R \,|\, t) + s_i$

Used for Ring Signatures when payee will in turn become payer spending this UTXO

"Elliptic notes" ☺

Lower case letters and H_s outputs are **scalar** values. **UPPER** case letters denote **points** on Monero-chosen elliptic curve (twisted Edwards Ed25519), even if they can be represented as a single 256 bit value thanks to a technique known as compression (representation used in hashing after relevant EC algebra has been applied, in addresses, in protocol fields). So, when involving EC points, products and sums have to be intended as their elliptic curves variant (acting on a 2D discrete space), not as usual scalar ones working on "compressed points values".
$H_s(\,) = \text{sc\_reduce32}(\,\text{Keccak256}(\,)\,)$: the Keccak hash output is capped by **sc_reduce32( )** due to EC points' cyclic nature (special thanks to Koe for having pointed this out); note that the same constraint applies to the transaction private key **r** and to the cited **25-words-Mnemonic-phrase key derivation method** (as well as to any other ones).

Credits

Mastering Monero (First Edition - December 2018 / Free PDF - 18th April 2019 - SerHack and the Monero Community)
Zero to Monero: Second Edition (v2.0.0 - April 4, 2020 - Koe, Kurt M. Alonso, Sarang Noether) chapters 1, 2, 4
Review of Cryptonote White Paper (July 2014 ? - Brandon Goodell AKA Surae Noether)
How Cryptonote Addresses Are Created (luigi1111) Various topics from Monero Stack Exchange and Monerodocs

NOTE: this cheatsheet's notation is slightly different from sources', trying to fit its "at first glance recap" function