|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | | | *Cheatsheet*  *version*  *20201206*  **Primary-** | | **Sub-** (i ≥ 1) | |
| **Private** | **Public** | **Public** | **Private** |
| **OFF-CHAIN** (payee′s data) | **Spend**  **keys** | | **Common key derivation method**    “capped” 256 bit seed  **Mnemonic** phrase of  24 + 1 (checksum)  words, among 1626  (162624 ≳ 2256)  **⋅ G**  s0    Hs | **S0 =** s0 **G** | **⋅ G**  **Si = Hs**(**“SubAddr”**|v0|**i** ) **G + S0**  **⋅ v0** | s**i = Hs**(**“SubAddr”** |v0|**i** ) **+** s0  **Actually never used:** subaddresses have been designed to share v0 usage (for blockchain-scanning performance reasons)  **⋅ v0** |
| **View**  **keys** | | **⋅ G**  v0 | **V0 =** v0 **G** | **⋅ G**  **Vi =** v0 **Si** | vi **=** v0 si |
| **Addresses** | | **Base58**( **0x12** | **S0** | **V0** | **checksum** ) **= “ 4 ……….” [95 chars]**  **4 bytes-truncated Keccak256 hash** | | **Base58**( **0x2A** | **Si** | **Vi** | **checksum** ) **= “ 8 ……….” [95 chars]**  **4 bytes-truncated Keccak256 hash** | |
| **Integrated**  **addresses** | | **8 byte-compact paymentID, encrypted in paying transaction (vs 32 byte former one)**  **Base58**( **0x13** | **S0** | **V0** | **payID** | **checksum** ) **= “ 4 ……….” [106 chars]**  **4 bytes-truncated Keccak256 hash** | | **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** | | **⋅ G**  r | **R =** r **G** | **⋅ Si**  **R =** r **Si** | r |
| **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 | **Xt = Hs**( r **V0**|**t**) **G + S0**  Identities hold true thanks to Diffie-Hellman-like connection of r, R, v0, V0, Vi  r (v0 **G**)  ⫴ | **Xt = Hs**( r **Vi**|**t** ) **G + Si**  r (v0 **Si**)  ⫴ | **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  **⋅ G**  xt **= Hs**( v0 **R**|**t** ) **+** s0 | v0 (r **G**)  **Xt = Hs**( v0 **R**|**t** ) **G + S0** | v0 (r **Si**)  **⋅ G**  **Xt = Hs**( v0 **R**|**t** ) **G + Si** | **Used for Ring Signatures** when payee will  in turn become payer spending this UTXO  xt **= Hs**( v0 **R**|**t** ) **+** si |
| **“Elliptic notes” ☺** | | | **Lower** case letters and Hs 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”.  **Hs( ) = sc\_reduce32( 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**](https://masteringmonero.com/free-download.html)(First Edition - December 2018 / Free PDF - 18th April 2019 - SerHack and the Monero Community) **NOTE:** this cheatsheet’s notation is slightly different  [**Zero to Monero: Second Edition**](https://www.getmonero.org/library/Zero-to-Monero-2-0-0.pdf)(v2.0.0 - April 4, 2020 - Koe, Kurt M. Alonso, Sarang Noether) chapters 1, 2, 4 from sources’, trying to fit its “at first glance  [**Review of Cryptonote White Paper**](https://www.getmonero.org/resources/research-lab/pubs/whitepaper_review.pdf)(July 2014 ? - Brandon Goodell AKA Surae Noether) recap” function  [**How Cryptonote Addresses Are Created**](https://xmr.llcoins.net/addresstests.html) (luigi1111) Various topics from [**Monero Stack Exchange**](https://monero.stackexchange.com/) and [**Monerodocs**](https://monerodocs.org/) | | | |