Monero keys and addresses cheatsheet

		Primary-		Sub- (i ≥ 1)	
		Private	Public	Public	Private
OFF-CHAIN (payee's data)	Spend keys	**Common key derivation method** 256 bit seed < EC order ↓ **Mnemonic** phrase of 24 + 1 (checksum) words, among 1626 ($1626^{24} \gtrsim 2^{256}$) ↻ H_s s_0	$s_0 \cdot G \rightarrow S_0 = s_0 G$	$S_i = H_s(\text{"SubAddr"} \mid v_0 \mid i)\, G + S_0$	$s_i = H_s(\text{"SubAddr"} \mid v_0 \mid i) + s_0$
	View keys	v_0	$v_0 \cdot G \rightarrow V_0 = v_0 G$	$V_i = v_0 S_i$	$v_i = v_0 s_i$ **Actually never used:** subaddresses have been designed to share v_0 usage (for blockchain-scanning performance reasons)
	Addresses	$\text{Base58}(\underbrace{\text{0x12}}_{\text{4 bytes-truncated Keccak256 hash}} \mid S_0 \mid V_0 \mid \text{checksum}) = \text{"4 ....."}$ [95 chars]		$\text{Base58}(\underbrace{\text{0x2A}}_{\text{4 bytes-truncated Keccak256 hash}} \mid S_i \mid V_i \mid \text{checksum}) = \text{"8 ....."}$ [95 chars]	
	Integrated addresses	8 byte-compact paymentID, encrypted in paying transaction (vs 32 byte former one) $\text{Base58}(\underbrace{\text{0x13}}_{\text{4 bytes-truncated Keccak256 hash}} \mid S_0 \mid V_0 \mid \text{payID} \mid \text{checksum}) = \text{"4 ....."}$ [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	$r \cdot G \rightarrow R = r G$		$R = r S_i \leftarrow r \cdot S_i$	
	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	$X_t = H_s(r V_0 \mid t)\, G + S_0$ $r (v_0 G) = v_0 (r G)$	$X_t = H_s(r V_i \mid t)\, G + S_i$ $r (v_0 S_i) = v_0 (r S_i)$	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
	Stealth Addresses (t ≥ 0) — Payee's POV	**Used for Ring Signatures** when payee will in turn become payer spending this UTXO $x_t = H_s(v_0 R \mid t) + s_0$	$\cdot G \rightarrow X_t = H_s(v_0 R \mid t)\, G + S_0$	$X_t = H_s(v_0 R \mid t)\, G + S_i \leftarrow \cdot G$	**Used for Ring Signatures** when payee will in turn become payer spending this UTXO $x_t = H_s(v_0 R \mid t) + s_i$

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

"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 technic 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 worked by **sc_reduce32( )** to make it less than the EC order, as required by EdDSA (Edwards-curve Digital Signature Algorithm); note that the same constraint applies 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