Signatures

The Safe supports different types of signatures. All signatures are combined into a singlebytes and transmitted to the contract when a transaction should be executed.

Encoding

Each signature has a constant length of 65 bytes. If more data is necessary it can be appended to the end of concatenated constant data of all signatures. The position is encoded into the constant length data.

Constant part per signature:{(max) 64-bytes signature data}{1-byte signature type}

All the signatures are sorted by the signer address and concatenated.

ECDSA signature

31 > signature type > 26

To be able to have the ECDSA signature without the need of additional data we use the signature type byte to encodev .

Constant part:

{32-bytes r}{32-bytes s}{1-byte v}

r ,s andv are the required parts of the ECDSA signature to recover the signer.

eth_sign

signature

signature type > 30

To be able to useeth_sign we need to take the parametersr ,s andv from callingeth_sign and setv = v + 4

Constant part:

{32-bytes r}{32-bytes s}{1-byte v}

r ,s andv are the required parts of the ECDSA signature to recover the signer.v will be subtracted by4 to calculate the signature.

Contract signature (EIP-1271)

signature type == 0

Constant part:

{32-bytes signature verifier}{32-bytes data position}{1-byte signature type}

Signature verifier - Padded address of the contract that implements the EIP-1271 interface to verify the signature

Data position - Position of the start of the signature data (offset relative to the beginning of the signature data)

Signature type - 0

Dynamic part (solidity bytes):

{32-bytes signature length}{bytes signature data}

Signature data - Signature bytes that are verified by the signature verifier

The methodsignMessage can be used to mark a message as signed on-chain.

Pre-validated signatures

signature type == 1

Constant Part:

{32-bytes hash validator}{32-bytes ignored}{1-byte signature type}

Hash validator - Padded address of the account that pre-validated the hash that should be validated. The Safe keeps track of all hashes that have been pre-validated. This is done with amapping address to mapping of bytes32 to boolean where it's possible to set a hash as validated by a certain address (hash validator). To add an entry to this mapping useapproveHash . Also if the validator is the sender of transaction that executed the Safe transaction it'snot required to useapproveHash to add an entry to the mapping. (This can be seen in theTeam Edition tests(opens in a new tab))

Signature type - 1

Examples

Assuming that three signatures are required to confirm a transaction where one signer uses an EOA to generate a ECDSA signature, another a contract signature and the last a pre-validated signature:

We assume that the following addresses generate the following signatures:

1. 0x3
2. (EOA address) ->bde0b9f486b1960454e326375d0b1680243e031fd4fb3f070d9a3ef9871ccfd5
3. (r) +7d1a653cffb6321f889169f08e548684e005f2b0c3a6c06fba4c4a68f5e00624
4. (s) +1c
5. (v)
6. 0x1
7. (EIP-1271 validator contract address) ->0000000000000000000000000000000000000000000000000000000000000001
8. (address) +000000000000000000000000000000000000000000000000000000000000c3

9. (dynamic position) +00
10. (signature type)* The contract takes the followingbytes
11.
    - (dynamic part) for verification0000000000000000000000000000000000000000000000000000000000000000deadbeef
12. 0x2
13. (Validator address) ->0000000000000000000000000000000000000000000000000000000000000002
14. (address) +0000000000000000000000000000000000000000000000000000000000000000
15. (padding - not used) +01
16. (signature type)

The constant parts need to be sorted so that the recovered signers are sortedascending (natural order) by address (not checksummed).

The signatures bytes used forexecTransaction would therefore be the following:

"0x" +
"000000000000000000000000000000000000000000000000000000000000000100000000000000000000000000000000000000000000000000000000000000000000c300" + // encoded EIP-1271 signature
"000000000000000000000000000000000000000000000000000000000000000200000000000000000000000000000000000000000000000000000000000000000000000001" + // encoded pre-validated signature
"bde0b9f486b1960454e326375d0b1680243e031fd4fb3f070d9a3ef9871ccfd57d1a653cffb6321f889169f08e548684e005f2b0c3a6c06fba4c4a68f5e006241c" + // encoded ECDSA signature
"0000000000000000000000000000000000000000000000000000000000000008000000000000000000000000000000000000000000000000000000000000deadbeef" // length of bytes + data of bytes

Safe Guards Supported Networks