Cosigners and multi-signature verification

Providing multi-signature verification support for dApps

A guardian is a trusted party, added by the user, that acts as a cosigner/co-validator for the user's account when carrying out typical wallet operations or for recovery purposes.

For most of Argent's products e.g Web Wallet, Argent Shield etc, the guardian is usually Argent's backend. In the next section, let's take a look at how you can verify multi-signatures for accounts with an active guardian.

Verifying multi-signatures

From a dApp's end, explicit support has to be provided for verifying multi-signatures, or account owners with 2FA enabled will be unable to sign transactions. This could be done in a few lines of codes.

Given a message to sign:

```

Copy constdata:TypedData={ types:{ StarkNetDomain:[ { name:"name",type:"string"}, { name:"version",type:"felt"}, { name:"chainId",type:"felt"}, ], Airdrop:[ { name:"address",type:"felt"}, { name:"amount",type:"felt"} ], Validate:[ { name:"id",type:"felt"}, { name:"from",type:"felt"}, { name:"amount",type:"felt"}, { name:"nameGamer",type:"string"}, { name:"endDate",type:"felt"}, { name:"itemsAuthorized",type:"felt*"}, { name:"chkFunction",type:"selector"}, { name:"rootList",type:"merkletree",contains:"Airdrop"} ] }, primaryType:"Validate", domain:{ name:"myDapp", version:"1", chainId:shortString.encodeShortString("SN_GOERLI"), }, message:{ id:"0x0000004f000f", from:"0x2c94f628d125cd0e86eaefea735ba24c262b9a441728f63e5776661829a4066", amount:"400", nameGamer:"Hector26", endDate:"0x27d32a3033df4277caa9e9396100b7ca8c66a4ef8ea5f6765b91a7c17f0109c", itemsAuthorized:["0x01","0x03","0x0a","0x0e"], chkFunction:"check_authorization", rootList:[ { address:"0x69b49c2cc8b16e80e86bfc5b0614a59aa8c9b601569c7b80dde04d3f3151b79", amount:"1554785", },{ address:"0x7447084f620ba316a42c72ca5b8eefb3fe9a05ca5fe6430c65a69ecc4349b3b", amount:"2578248", },{ address:"0x3cad9a072d3cf29729ab2fad2e08972b8cfde01d4979083fb6d15e8e66f8ab1", amount:"4732581", },{ address:"0x7f14339f5d364946ae5e27eccbf60757a5c496bf45baf35ddf2ad30b583541a", amount:"913548", }, ] }, };

```

The user connects his account to sign the message, after which the signature and account address will be returned.

For a normal account without 2FA, you should get a single signature pair[r1, s1] , but for shield accounts, you should get a concatenated array of the owner and guardian signatures[r1, s1, r2, s2] , which could be verified by simply calling theisValidSignature oris_valid_signature method of the account contract:

```

Copy constcontractAccount=newContract(abi,accountAddress,provider); constmsgHash=typedData.getMessageHash(data,accountAddress); awaitcontractAccount.isValidSignature(msgHash, [signature1.r,signature1.s,signature2.r,signature2.s])

```

Previous Multicalls Next Session Keys - [early access feature]

Last updated8 days ago