# NEX Authentication Protocol implementation

## Introduction

Cure53 provided NEX with an authentication protocol in their document "NEX Authentication and Transaction Verification Protocol." Within this document, the procedure for account onboarding, login, private key storage, passphrase change are described.

An initial implementation has been completed at `neonexchange/nex-auth-protocol`. We would like the current implementation to be audited for any security issues.

## Implementation of Cryptographic Primitives

- Hashing: `scrypt-js`
- HKDF (generation of auth key and encryption key): `fution-hkdf`
- AEAD encryption/decryption: `browserify-aes` (polyfill of Node `crypto.createCipheriv()` and `crypto.createDecipheriv()`)
- Key generation: `randomBytes()` (polyfill of Node `crypto.randombytes`; also from Browserify)
- Signing, sigverif: not yet implemented; plan to use `elliptic` library
- Chain-specific key derivation: not yet implemented

## Other Implementation Notes

- "End-to-end" flows can be seen in the specs for `decryptSecretKey()` and `regenerateMnemonic()`, as these functions invoke most of the other primitives.
- Currently user ID is used as a salt whenever a salt is needed. It is expected to be a UUID.