CounterMail - protection of the private OPENPGP Key

In August 2017 we changed our private key protection. This paper describes the new private key protection.

The authentication process involves the retrieval and decryption of a private key from a Key Server. Private keys are required for all public key decryption and signing operations.
All crypto operations is done on the client side (the user's local computer).

Private Identifier

Private OPENPGP keys are indexed on the server by a value called the private Identifier. Mixing the unique userid and the passphrase together using a secure one-way hash function generates the private Identifier. Thus, the private Identifier cannot be associated with any particular username. Any data indexed by the private Identifier, including private keys, is stripped of any information that may associate it with the usernames. This means that private keys are stored anonymously. If an attacker gains complete access to the database in which the keys are stored, that attacker will be unable to determine which encrypted private key belongs to which user, thus making the process of compromising any particular private key significantly more complex.

The private Identifier is generated as follows:

A = SHA256(SHA256(OPENPGP_S2K(passphrase, username)) + userid)
Identifier = first 32 hexadecimal characters of A

userid = 16 randomly generated bytes (generated once during user registration)
username = registrated username
passphrase = users own passphrase
OPENPGP_S2K = SHA1(username+passphrase) is repeated until a total of 262144 octets have been input to the hash function

During the authentication process, the private Identifier is sent to the Key Server as a key request. The server responds by answering that no key for that value was found, or by returning one or more encrypted packages.

The encrypted package in which the private keys are stored consists of a symmetrically encrypted OPENPGP private key. The encrypted package is encrypted with the following algorithm:

aeskey = SHA256(passphrase + OPENPGP_S2K(passphrase, username) + userid)
enc_key_data = AES256_CBC_ENCRYPT(keydata, aeskey)