

CLAIMS

1. A method for restoring a distributed secret key from a backup storage, the method comprising the steps of:

- 5 – generating an original secret key and distributing the original secret key among two or more servers, each server thereby holding a share of the original secret key and none of the servers holding all shares of the original secret key,
- a first server creating a backup, the backup containing at least the share of the original secret key which is held by the first server,
- 10 – the servers refreshing the original secret key at least once, where each refresh comprises the steps of:
 - 15 – the servers generating a refreshed distributed secret key and a distributed difference between the previous secret key and the refreshed secret key, each server holding a share of the difference, and the difference constituting a sharing of zero, and the servers discarding their shares of the previous secret key,
 - each server generating a secret version of its share of the difference and sharing the secret version with at least some of the other servers, and
 - 20 – each server generating an accumulated secret version of the shares of the difference received from the other servers based on the received secret versions of the shares of the difference and previous accumulated secret versions,
- the first server restoring its share of the original secret key from the backup and requesting the accumulated secret version of its share of the difference from the other servers,
- 25 – at least some of the other servers providing the accumulated secret version of the first server's share of the difference to the first server, and
- the first server restoring its share of the latest refreshed secret key from the received accumulated secret version and the restored share of the original secret key.

2. A method according to claim 1, wherein the step of each server generating a secret version of its share of the difference is performed by means of an additively homomorphic scheme, and wherein the step of each server generating an accumulated secret version comprises each server adding the received secret versions of the shares of the difference to
5 the respective previous accumulated secret versions, and discarding the previous accumulated secret versions.
3. A method according to claim 2, wherein the additive homomorphic scheme is a public-key encryption scheme, and wherein the step of the first server creating a backup comprises storing a private decryption key as part of the backup.
- 10 4. A method according to claim 2 or 3, wherein the step of each server generating a secret version of its share of the difference comprises each server creating a sharing of its share of the difference and distributing the shares among at least some of the other servers.
5. A method according to any of the preceding claims, wherein each refresh comprises the steps of:
- 15 – the servers creating a sharing of zero, and each server adding its share of the sharing of zero to its share of previous secret key, thereby creating the refreshed secret key being distributed among the servers, and discarding their shares of the previous secret key, the sharing of zero constituting the distributed difference, and
- each server generating a secret version of its share of the sharing of zero and sharing
20 the secret version with at least some of the other servers.
6. A method according to any of the preceding claims, further comprising the step of storing the backup in an offline storage.
7. A method according to any of the preceding claims, wherein each step of refreshing the secret key further comprises the step of each server verifying correctness of the received
25 secret versions of the shares of the difference by verifying a zero-knowledge proof.
8. A method according to any of the preceding claims, further comprising the step of the first server verifying correctness of the received accumulated secret version by verifying a zero-knowledge proof.

9. A method according to any of the preceding claims, wherein the distributed secret key is a secret signature key.
10. A method according to any of claims 1-8, wherein the distributed secret key is a secret encryption key.
- 5 11. A method according to any of the preceding claims, further comprising the step of the first server creating a new backup, and repeating the steps of the servers refreshing the original secret key at least once, and wherein the step of the first server restoring its share of the original secret key comprises restoring a share of the secret key stored with the new backup.
- 10 12. A method according to claim 11, wherein each refresh further comprises the step of the first server generating an accumulated secret version of its share of the difference and storing the accumulated secret version at the first server, wherein the secret version of the share of the difference forms part of the new backup, and wherein the step of the first server restoring its share of the latest refreshed secret key is performed based on the restored
- 15 share of the secret key forming part of the new backup, the secret version of the accumulated difference forming part of the new backup, and the accumulated secret version received from at least some of the other servers.