A private-key backup strategy through Shamir's Secret Sharing algorithm, distributed Secret Keepers, and Secure QR Codes (SQRC).

A submission to Lykke Wallet's Private key distributed backup contest

by José Aguinaga

Abstract

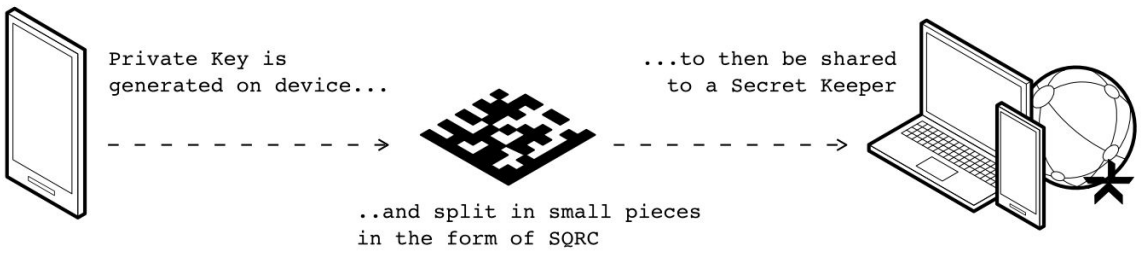
Shamir's Secret Sharing algorithm allows to create a secret (e.g. a private key) which can be split to a number of parties, isolating the responsibility of storing the entirety of the secret on a single entity. By distributing the split secrets in the form of Secure QR Codes (SQRC) to a number of parties that have no relationship between themselves, one can leverage on a reconciliation strategy to retrieve the individual pieces if needed, as part of a backup mechanism. Since the secrets are distributed in the form of SQRC, transmission and retrieval can be done through mobile devices, while still ensuring the entrusted parties learn nothing from storing their share of the secret.

Introduction

During Lykke's mobile application onboarding, a private key is generated locally to ensure the security and integrity of the digital wallet. This private key can be backed up through BIP39, in the form of twelve mnemonic words the user has to safeguard in case of losing access to his device or account. However, this gives entrust the user with the sole responsibility of safekeeping his account, which in return gives a poor user experience.

An alternative backup strategy relies on distributing the responsibility of storing of the key to trusted or semi-trusted parties. This, in return, give us the problem that now the involved parties have access to the key, and can use it to their own advantage. Can we distribute a private key, in a secure way, without exposing the content of the key by itself?

The backup strategy introduced in this document, uses **Shamir's Secret Sharing Algorithm** and **Secure QR Codes (SQRC)**, in order distribute the private key to semi-trusted parties (known in this document as **Secret Keepers**), while protecting the content of the key or the distributed pieces of the key.



With a private key split locally within their devices, users can then pick between a range of three possible Secret Keepers to distribute their split private key: first, **an individual, or series of individuals outside the Lykke's Network** to whom a SQRC is sent by e-mail or SMS through the user's mobile device. Second, **an anonymised randomly selected user or users within the Lykke's Network** that are given another SQRC, and requested to keep the keys in their form in exchange of a small amount of Lykke LKK. Finally, **Lykke AG** itself, through a local secure storage solution. For the last two parts, a time interval is required in order to ensure the second SQRC is never stored at the same time the third SQRC is given to Lykke AG.

It's important to describe that through Shamir's Secret Sharing algorithm, not all the pieces are required to restore the key, so a **threshold** can be chosen. Thus, each user can choose how much "secret" he or she shares in the SQRC shared to each Secret Keeper selected, in order to recover his private key.

How to make a distributed backup

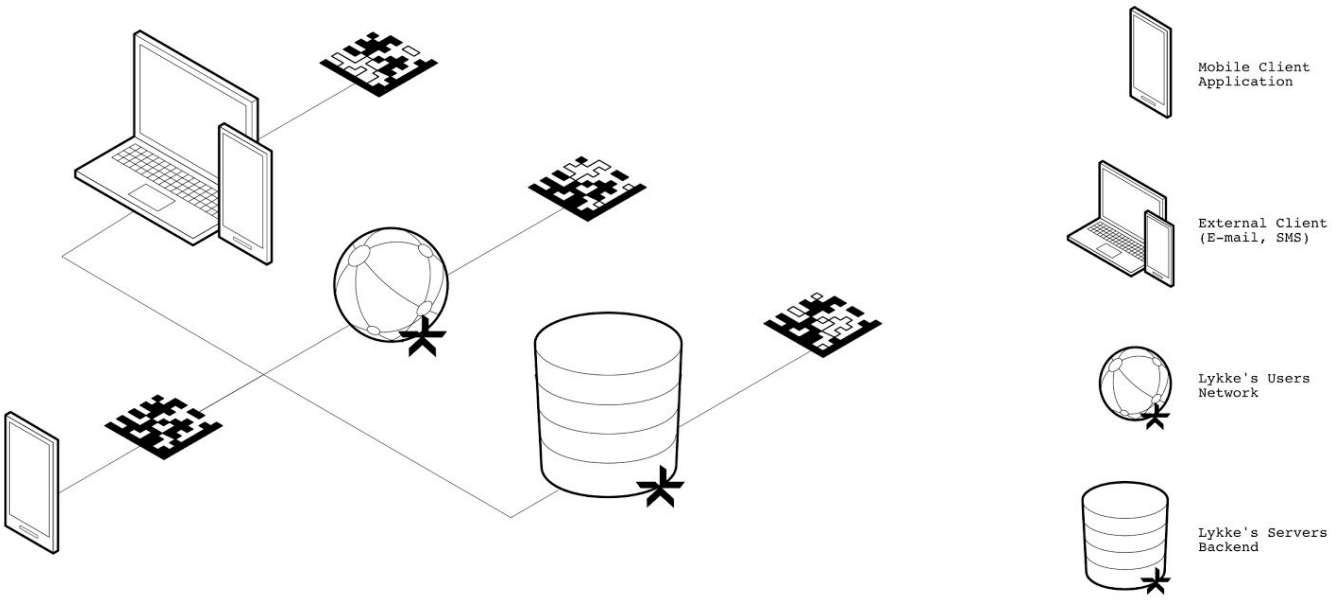


Figure 1. A secret key is split in different SQRCs, and issued to different members within our system. Each SQRC has a percentage of the total key, and in this case, any combination of two pieces are required to restore the private key.

A **Secret Keeper** is an actor within a system that holds a confidential piece of a secret, while still learning nothing from the secret itself. In its pure form, it's just an entity that works as a temporal storage solution.

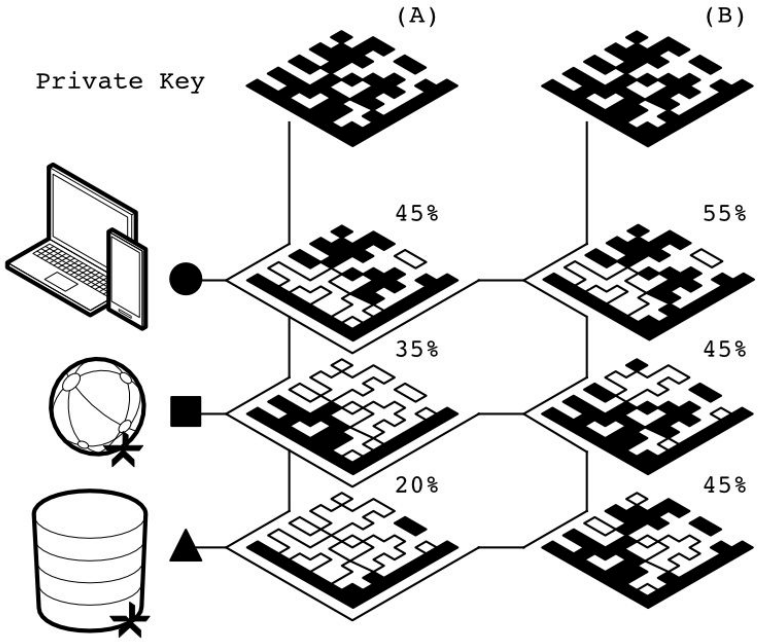
The most important part of distributing a secret, is to identify the actors that can become Secret Keepers. The proposed solution uses a 3 actors model for sharing the responsibilities:

- **External Party.** Any party that is outside of the Lykke's network. Ideally, we are talking about a close individual (or individuals) that can be reached through Email or SMS, and can be verified through similar channels.

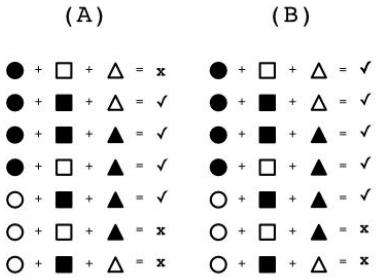
- **Lykke's Users Network.** Any user(s) chosen at random within Lykke's network. Ideally, we are looking for active, involved users within the Network.

- **Lykke's Servers.** Any server within Lykke AG. These servers should work only as storage application (i.e. database).

These actors will storage a **percentage** of our key, in the form of a **SQRC** divided through a client base implementation of Shamir's Secret Sharing algorithm (see **Fig. 1)**. In Fig. 1 image, each actor was given roughly 45% of the key, which would require at least two SQRC. There are other possible combinations, where for instance, one Secret Keeper has more than 45% and only its SQRC it's needed to restore the private key (see **Fig. 2**).



Split parts with different percentages allows scenarios where multiple combinations provide enough acceptance criteria to restore the key when ownership is >50%.



How to recover the key from a backup step by step

To showcase the recovery of the backup key, we will take an imaginary use case that will cover the backup procedure, and then the recovery of the key from the given backup.

Martin Uhl decides to backup his Lykke's account private key. Since the key can be described as a 15 words mnemonic (as described on BIP39), it looks something like this:

```
arena lake gate local shop process skate glow already
bulb crucial trend spin ripple combine
```

Now, instead of writing each of these keys and having to safekeep all of the words, Martin opts for doing a distributed backup. The Lykke applications prompts Martin to pick to whom he wants to provide how much percentage of the key. He picks to requiring 50% of the keys to restore the key, and then giving 50% of the key to a series of friends and/or family, while giving other Lykke network users 30%, and Lykke AG 20%. Behind the scenes, the Lykke mobile application now generates the following list of mnemonics:

*Shamir's Secret Split key implementation through mnemonics, see **seedsplit**.*

- divorce dynamic furnace shoulder cart jazz devote grant razor valley heavy swamp switch limb gain rug electric

- divorce roast energy sunny bulk foil marine develop squeeze jewel lawn yellow twelve apology hamster pool scene

- divorce series inside off fantasy undo peace risk change vanish man voice input cricket umbrella mention whale

- dizzy bring credit action ripple involve upset burden glue camera stairs bacon satisfy movie corn electric gasp

- dizzy excuse evoke say corn trophy ancient reason exist law seat holiday youth fork panther silver man

- dizzy question tomorrow enrich misery next calm rubber oven essence coyote poem wheat coyote garment obvious food

- dizzy shrimp update measure powder smoke gap pioneer glass peace robust copy naive name measure work ocean

- doctor act turkey own happy summer pumpkin hybrid wisdom come error service random hold follow success wise

- doctor joke tell grief apart laugh alert unveil fee country that depart toward zebra hamster only february

- doctor merit wrist fly frog mountain invest nut wrestle beef lecture moon divide debate arena athlete engine

Where the blue ones represent the ones he will share with his family, the green ones the ones shared with Lykke users, and the red ones with Lykke AG itself. (For the blue ones, he can then decide how many of those send to each friend). Having only five of those can restore the original key.

The new mnemonics are then converted into SQRC, and sent to each party. To retrieve the key, Martin has to do one of the two steps:

1. *Reach his friend or family to send him the SQRC back, to which the he will scan with the Lykke mobile app. The Lykke app is embedded with the ability to convert the SQRC to the original mnemonic list, and thus, his key (see Anexus A).*

2. *Reach Lykke through the app. The Lykke app will proceed to verify the user information to then send the SQRC back from the Lykke servers, while reaching the linked Lykke users that hold Martin's SQRC, and request an exchange for them. After returning them to Martin, the moment he has enough SQRC, he proceed to restore his key.*

Current identified issues and how to address them

For first actor

- SMS/E-mail with SQRC get deleted.
    - Possible solution: Review each SQRC every X amount of time.
- Confirmation on receival of SQRC is unknown.
    - Possible solution: Receivers of the SQRC need to confirm with a link included in the SMS or email.
- Wants to use the key him/herself.
    - Possible solution: Since the key is encrypted, there's no way to allowing the usage of the key.

For second actor

- Lykke users lose other's people's SQRCs by losing their own devices.
    - Possible solution: Review and rotate each SQRC every X amount of time.
- Lykke users can't return the SQRC (e.g. exchange request expire, user goes offline)
    - Possible solution: Set up an expiration note on a multi signed transaction for the colored coin that holds the SQRC to avoid blockage. On expiration, the SQRC is returned to the Lykke server. Only use active users.
- Exploiting the system (i.e. forcing Lykke to pay up by requesting SQRC)
    - Possible solution: Rotate ownership of split keys, only pay up after retaining keys for X time instead of by request. Make users take on the cost of retrieving their backups.

For third actor

- Lykke's servers/databases are compromised
    - Possible solution: Never give more than 45% of SQRC pieces to Lykke AG. Ensure no metadata used to encrypt the SQRC is inside Lykke servers.

General issues

- The time to back-up the key depends in how fast the Lykke server's can find suitable back-up providers, before being able to store a SQRC themselves.
- Notification of a retrieval for the first actor is done outside the Lykke system. A retrieval can be prompted by any X party, and the first actor would need to confirm that the request was valid (i.e. confirm with the user that he or she was indeed requesting for the backup).

ANEXUS A: How to implement SQRC

SQRC (Secure QR Codes) are registered trademarks of DENSO WAVE INCORPORATED in Japan and in other countries.

DESCRIPTION

In their form, SQRC are only normal QR codes that have been encrypted in a way that no useful information can be retrieved. For instance, the Japanese government stamps information about yourself in your passport when you visit the country. Only QR readers that have been “embedded” with the private key paired with the public key used to generate the QR code, can read the contents of the QR. Hence, this is a SQRC.

How could Lykke create SQRC? It already has a series of meta-data information it could use in order to generate (and regenerate) a key for encrypting the split keys of the secret key:

1. Email (Email is verified during onboarding)
2. Password + Passphrase (Required on onboarding)
3. Phone (Phone is verified during onboarding)
4. Pin (Required for wallet)

Using a PBKDF2 implementation with sensible defaults (i.e. 100,000 iterations, 256-bit length key, and SHA-256), we can make input these metadata information in order to generate a key that will be used to encrypt the contents of the split keys.

POSSIBLE CONCERNS

Is this secure? Relying on metadata to generate a private key is the only to have a sensible user experience, instead of making him or her remember yet another password, to which would take us to square one.

What happens if the metadata gets leaked? In case the metadata is retrieved by an attacker, the hacker would still need to be able to get the pieces of the split keys in order to restore the private key.

Doesn’t Lykke holds the metadata? Yes, although ideally we can pick other information to generate the PBKDF2 not even Lykke can return. Or, we can make Lykke hold this information, but never hold any SQRC pieces itself, and just allow Lykke users to hold those.

ANEXUS B: Document verification (@jjperezaguinaga)

-----BEGIN PGP SIGNED MESSAGE-----
Hash: SHA512

A private-key backup strategy through Shamir's Secret Sharing algorithm, distributed Secret Keepers, and Secure QR Codes (SQRC).

A submission to Lykke Wallet’s Private key distributed backup contest by José Aguinaga

Abstract
Shamir's Secret Sharing algorithm allows to create a secret (e.g. a private key) which can be split to a number of parties, isolating the responsibility of storing the entirety of the secret on a single entity. By distributing the split secrets in the form of Secure QR Codes (SQRC) to a number of parties that have no relationship between themselves, one can leverage on a reconciliation strategy to retrieve the individual pieces if needed, as part of a backup mechanism. Since the secrets are distributed in the form of SQRC, transmission and retrieval can be done through mobile devices, while still ensuring the entrusted parties learn nothing from storing their share of the secret.
-----BEGIN PGP SIGNATURE-----
Version: Keybase OpenPGP v2.0.68
Comment: https://keybase.io/crypto

wsBcBAABCgAGBQJY53DKAAoJEEcNV2F2U20cXHwH/AIT6Jo8i/uuJf4Jl+gJrLVa
f1c7jP4LGLbt/CUyY1MTJcLqkj/1EZux8D9VBmGmHM/TdH8q8enkZgr97FJg0L7E
uQ3/U/GmGU+a9DyUBYkt9iqVrvPKi7eiHu+Y54skeg29MNhbLFiTfce+6c1OeSMU
Wfv/E7Kl0asWRzwIQp0efbfWy5z5R6QFjXUVOedGe7bKmlpTs3X9YLlyOK5SNjwW
bUy/kj0o+kJDj9LbFvZ527quI1qiEWJ0uOzmbUUC76GpqIIFp4xEtblJi3m3hxo
aMElm4Nk42t3vkp1skV1fMVqeHdf2JmheLwgrrK1gzFYboQLeX/RFSOWPfKvZ2w=
=07iP
-----END PGP SIGNATURE-----

LATEST NEWS

The following describe the latest headlines of today (04.07.2017) as per Reuters.com.

- Samsung tips best quarterly profit in over three years as chips soar
- U.S. fires missiles at Assad airbase; Russia denounces 'aggression'
- Global stocks off lows, oil rallies after U.S. missile strike on Syria