Keybase

Yuchen Yang
11/21/19

Outline

Security model and keybase

Potential attacks and defenses

- identity
- merkle tree and blockchain
- keys

KBFS

```
Starting via systemd...
run_keybase: Success!
```

Security model

End-user devices are trusted; keybase servers are untrusted

Attacks:

1. Server DDOS'ed
2. Server compromised; attacker corrupts server-side code and keys to send bad data to clients
3. Server compromised; attacker distributes corrupted client-side code

What does Keybase do

1. **Identity proofs**: "I am Joe on Keybase and MrJoe on Twitter"
2. **Follower statements**: "I am Joe on Keybase and I just looked at Chris's identity"
3. **Key ownership**: "I am Joe on Keybase and here's my public key"
4. **Revocations**: "I take back what I said earlier"

Joe posts a signed statement both on Twitter and Keybase

Joe also signs the hash or Joe's previous signature.

Each user owns a signature chain

https://keybase.io/docs/server_security

Attacks and Defenses

Breaking into user accounts (Keynote, twitter, ...)

Compromised server

Breaking into user devices

Identity proof

a public post

Following (Identity verification)

1. request for user info from keybase server

```
{
  "keybase_username": "maria",
  "public_key":      "---- BEGIN PGP PUBLIC KEY...",
  "twitter_username": "@maria2929",
  "twitter_proof":    "https://twitter.com/maria2929/2423423423"
  ...... all accounts verified
}
```

Following

1. request for user info
2. client verify identity proofs
3. human review

Following

1. request for user info
2. client verify identity proofs
3. human review

Repeating step 2 & 3 is horrible!!!

Solution

Signed snapshot when following.

Client continue computer review.

Older follower statements are better.

If a snapshot is the same with the previous one, nothing changes during the period of time.

6 months ago I confirm...
6 months ago I confirm...
6 months ago I confirm...
maria
5 months ago I confirm...
5 months ago I confirm...
5 months ago I confirm...

Web of trust

No

more confidence in the age of the account

Attacks and Defenses

Breaking into user accounts (Keynote, twitter, ...)

Compromised server

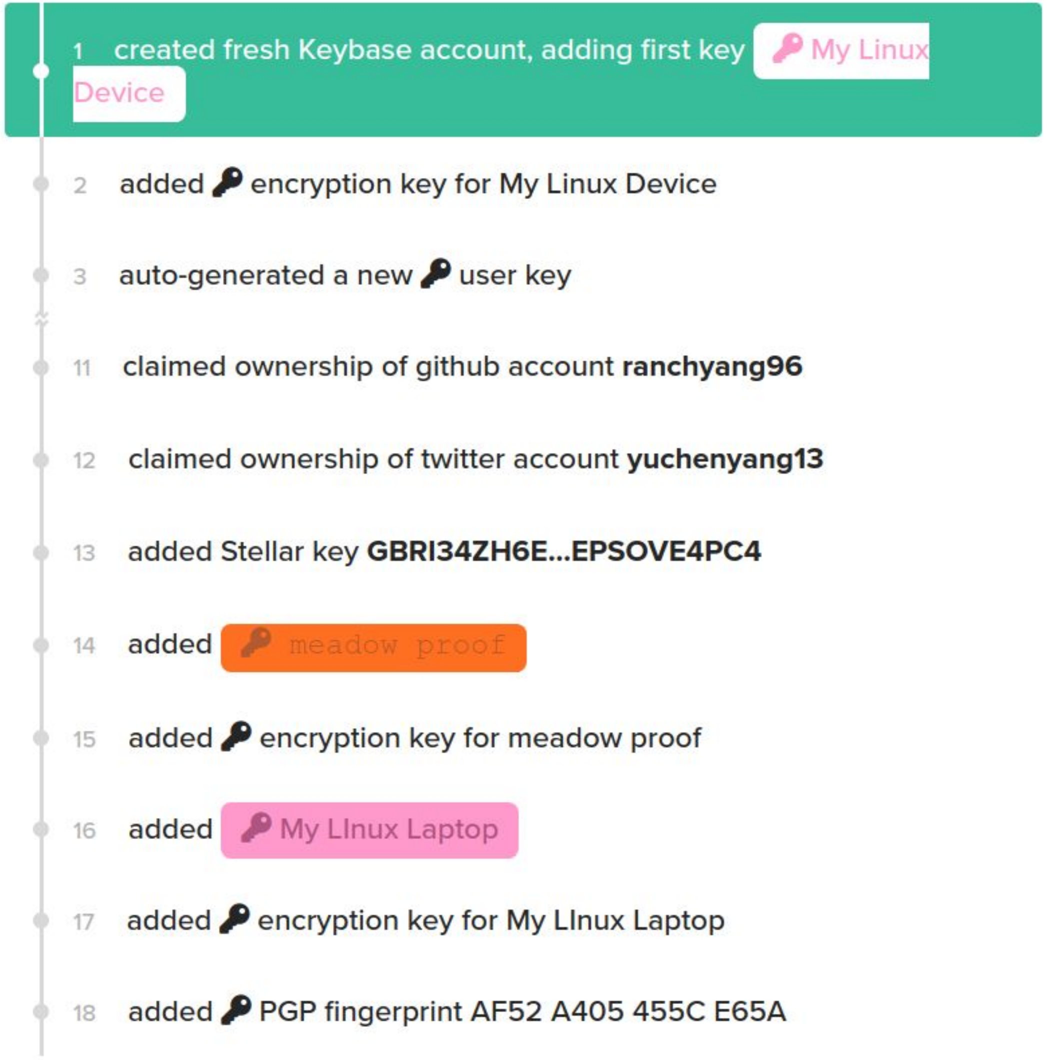
Breaking into user devices

Compromised server

Attacks:

1. **Server DDOS'ed**
2. **Server compromised; attacker corrupts server-side code and keys to send bad data to clients**
3. Server compromised; attacker distributes corrupted client-side code

Sigchain



1 created fresh Keybase account, adding first key My Linux Device

2 added encryption key for My Linux Device

3 auto-generated a new user key

11 claimed ownership of github account **ranchyang96**

12 claimed ownership of twitter account **yuchenyang13**

13 added Stellar key **GBRI34ZH6E...EPSOVE4PC4**

14 added `meadow proof`

15 added encryption key for meadow proof

16 added My LInux Laptop

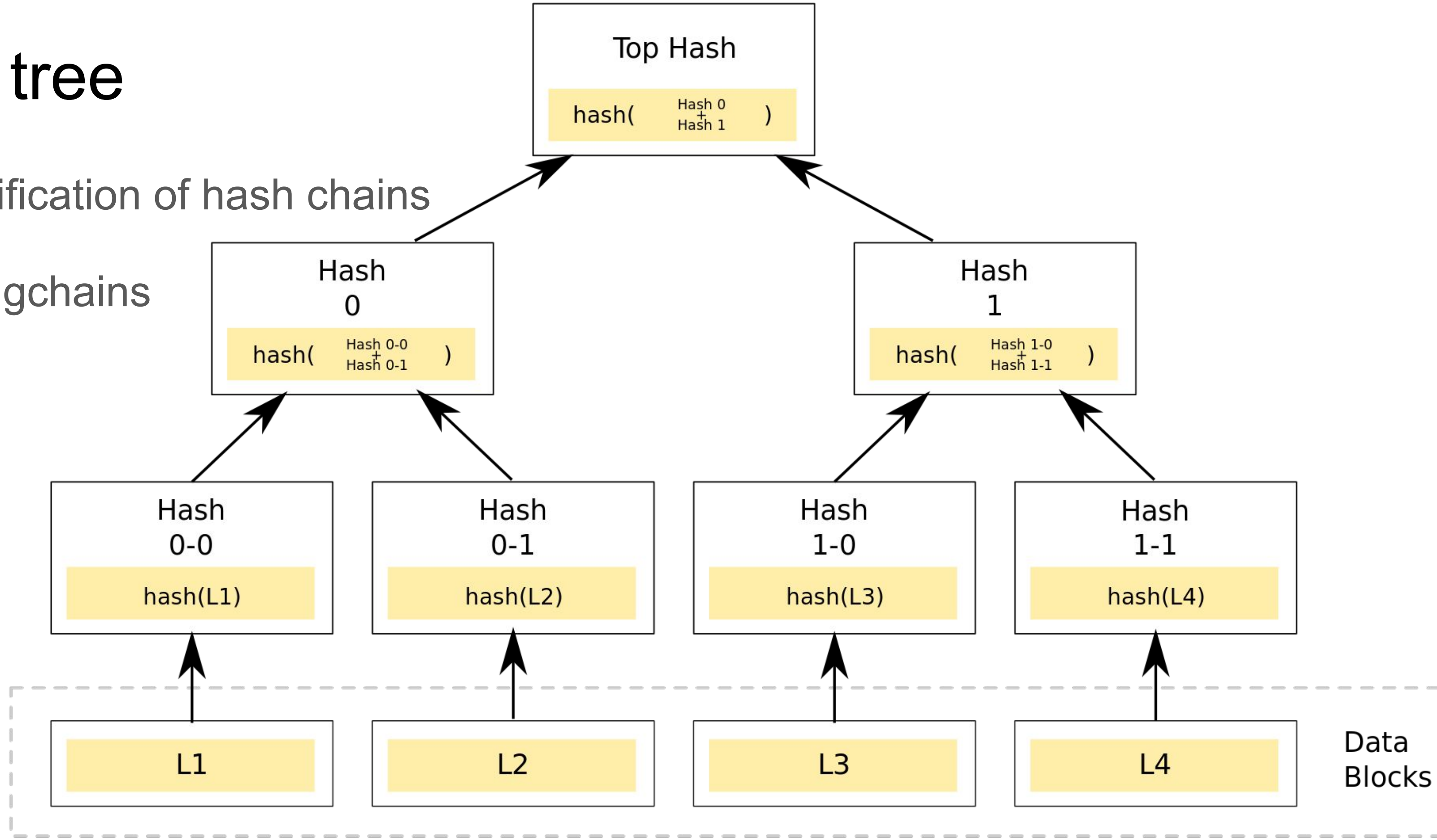
17 added encryption key for My LInux Laptop

18 added PGP fingerprint AF52 A405 455C E65A

Merkle tree

Enable verification of hash chains

Cover all sigchains



Writing to Bitcoin Blockchain

Announcements verifiably signed by Keybase

Merkle roots hashed into the Bitcoin blockchain

Compromised server

Attacks:

1. Server DDOS'ed
2. Server compromised; attacker corrupts server-side code and keys to send bad data to clients
3. **Server compromised; attacker distributes corrupted client-side code**

Client code integrity

OpenAPI for any other client developers

all updates signed

update mechanism provided to download clients without HTTPS trust

Attacks and Defenses

Breaking into user accounts (Keynote, twitter, ...)

Compromised server

Breaking into user devices

Keys (Device security)

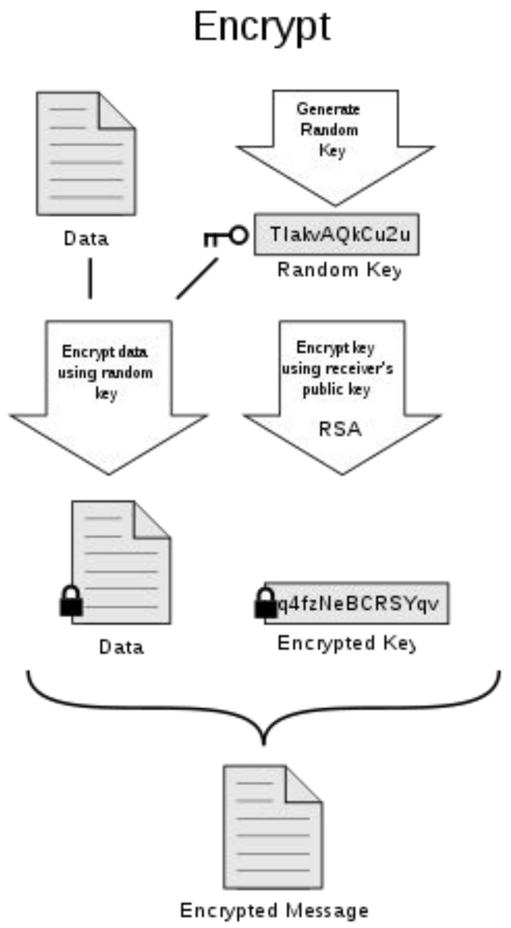
PGP key

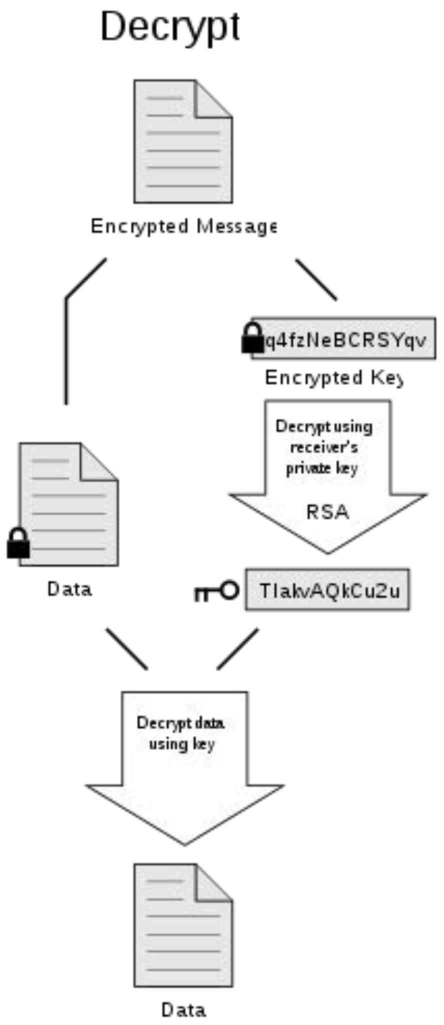
add a new device

lost my devices

local key security

PGP





https://en.wikipedia.org/wiki/Pretty_Good_Privacy#PGP_fingerprint

Per-user key

PGP public key visible to everyone

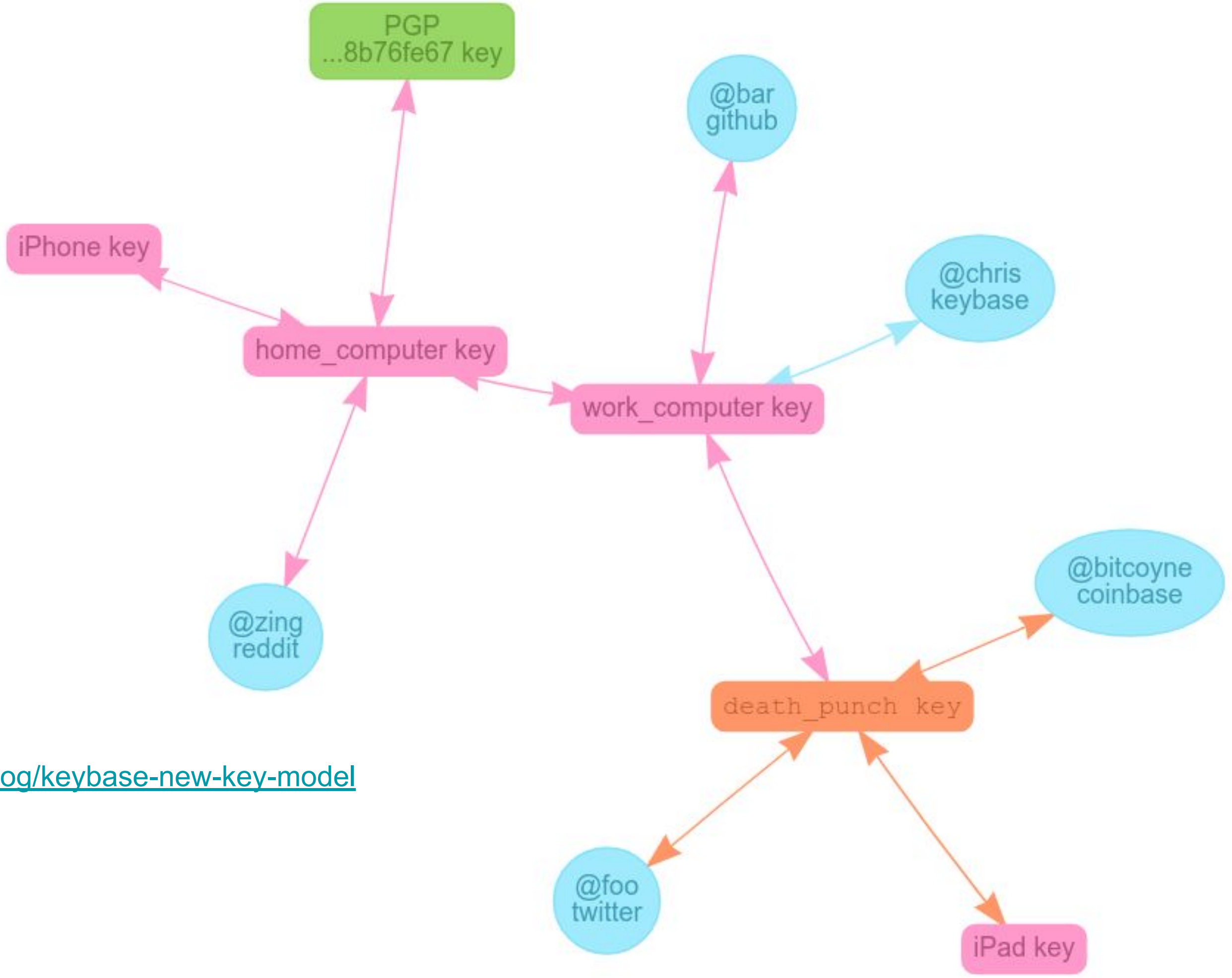
private key encrypted with all device key/paper key

New device (X provision Y)

Establish a session between X and Y:

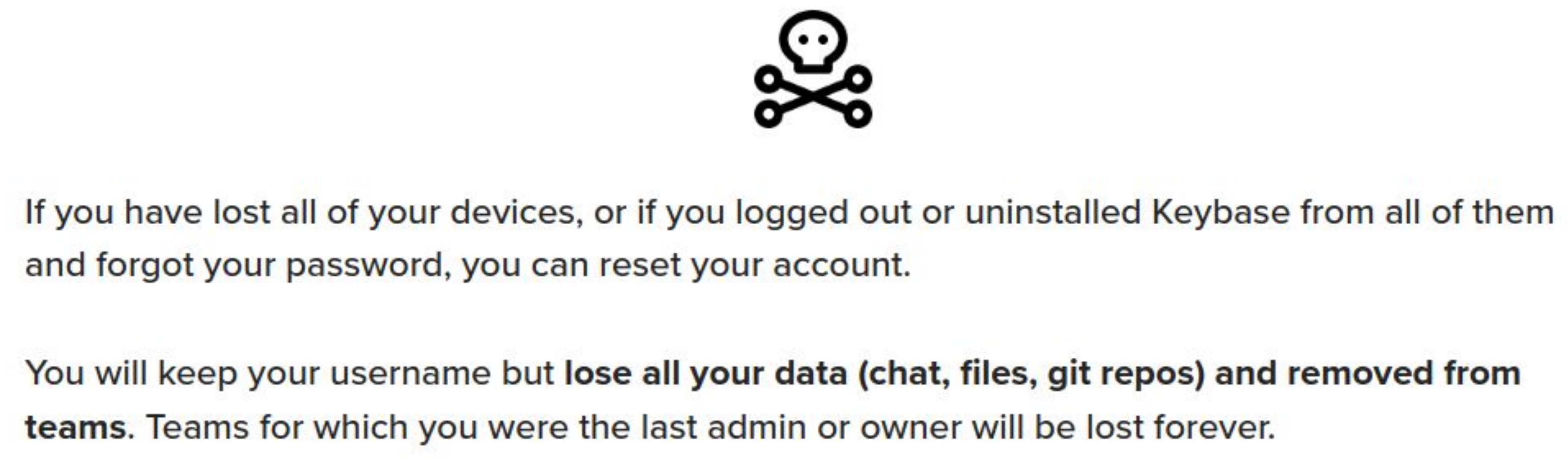
1. Generate word string W
2. secret S by running scrypt on W
3. session identifier I by hashing S
4. both X and Y send messages to the server with identifier I_x and I_y
5. enter S_y on X
6. encrypted transport with encrypted keys

https://keybase.io/docs/crypto/key-exchange



https://keybase.io/blog/keybase-new-key-model

Device lost



If you have lost all of your devices, or if you logged out or uninstalled Keybase from all of them and forgot your password, you can reset your account.

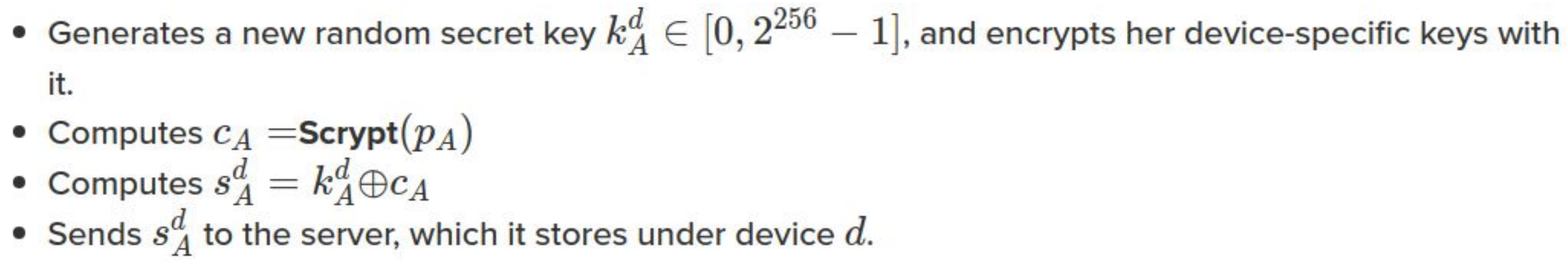
You will keep your username but **lose all your data (chat, files, git repos) and removed from teams**. Teams for which you were the last admin or owner will be lost forever.

reconfirm identity necessary

https://keybase.io/account/reset_me

Local key security

Key generation



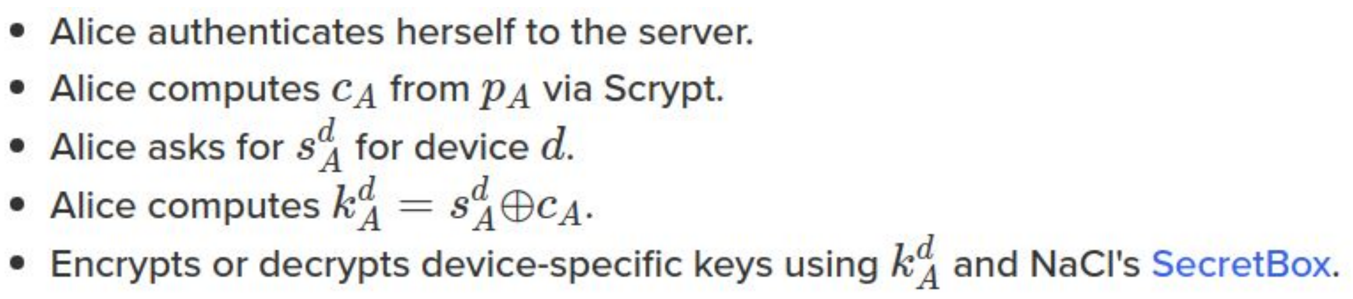
- Generates a new random secret key $k_A^d \in [0, 2^{256} - 1]$, and encrypts her device-specific keys with it.
- Computes $c_A =$ **Scrypt**(p_A)
- Computes $s_A^d = k_A^d \oplus c_A$
- Sends s_A^d to the server, which it stores under device d.

Scrypt is a password-based key derivation function.

Designed to be computationally intensive, so that it takes hundreds of milliseconds to compute.

Local key security

Encrypting and Decrypting



- Alice authenticates herself to the server.
- Alice computes c_A from p_A via Scrypt.
- Alice asks for s_A^d for device d.
- Alice computes $k_A^d = s_A^d \oplus c_A$.
- Encrypts or decrypts device-specific keys using k_A^d and NaCl's SecretBox.

Local key storage

store in local OS’s keychain

use the hardware integration which provides guarantees

password changes need calculation of c_A differences.

https://keybase.io/docs/crypto/local-key-security

KBFS

filesystem mounted to /keybase in unix-like machines

public files: /public/writers (/public/alice,bob,...)

private files: /private/writers#readers (/private/alice#bob,charlie)

team: /team/teamname (team/team.subteam.subsubteam)

team administration by adding signatures to a chain

KBFS Summary

Signing(sibling) keys and encryption(sub) keys for private personal directories

For private group files, key files are generated along with metadata, writer keys, user keys, contents

For public files, only signature used

one global merkle tree for private files, one for public files

https://keybase.io/docs/crypto/kbfs