Passwords in Peer-to-Peer

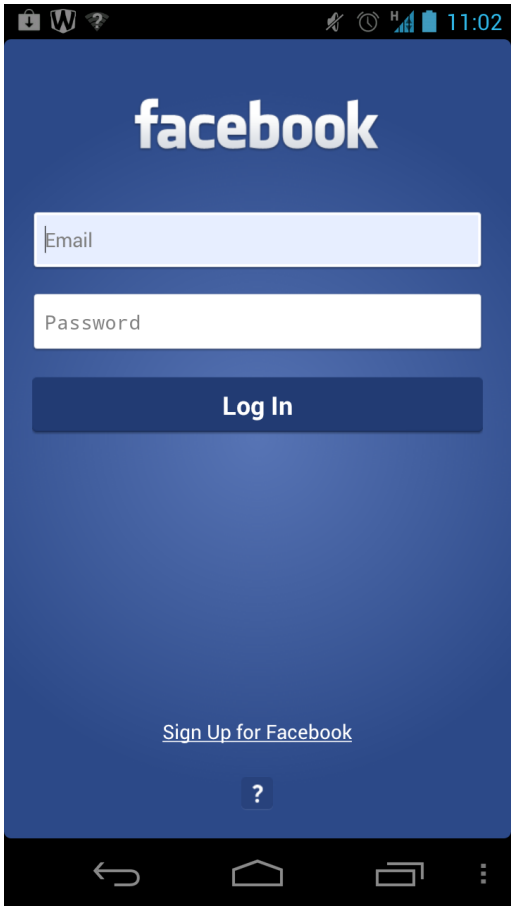
Gunnar Kreitz, Oleksandr Bodriagov, Benjamin Greschbach,
Guillermo Rodríguez-Cano, and Sonja Buchegger

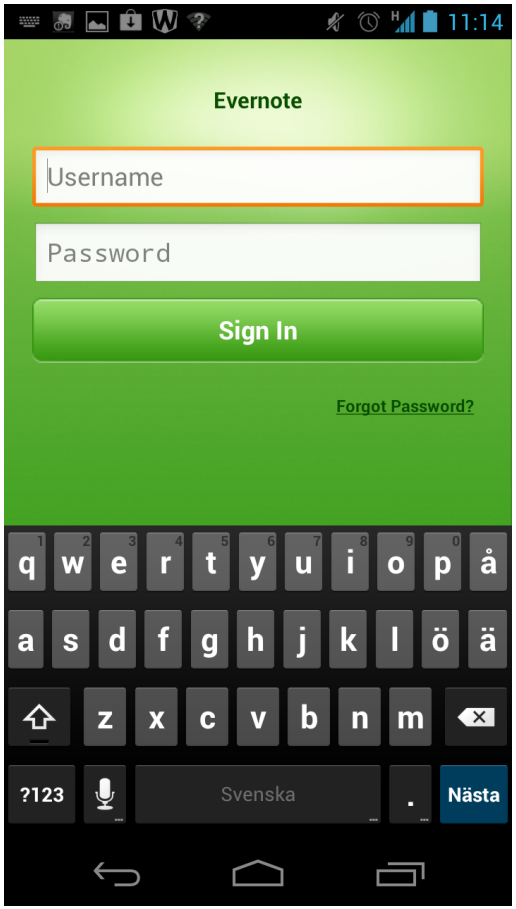
KTH – Royal Institute of Technology

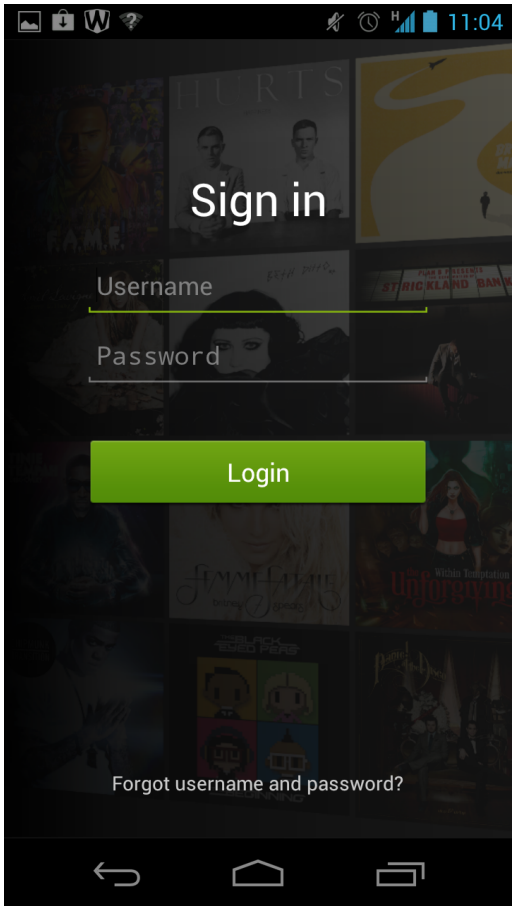
IEEE P2P'12, September 4 2012

KTH Computer Science and Communication

Modern End-User Services





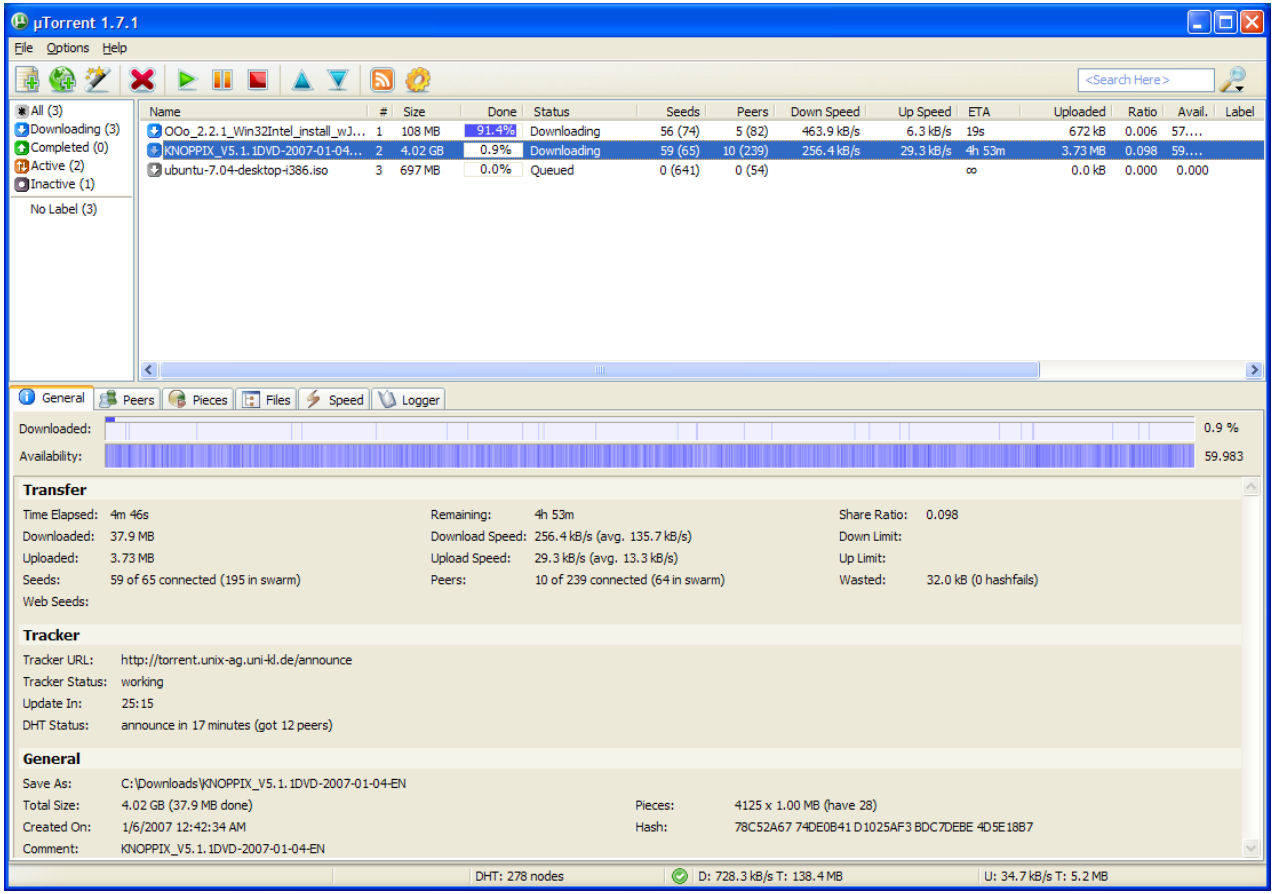


KTH Computer Science and Communication

Logging in to P2P?

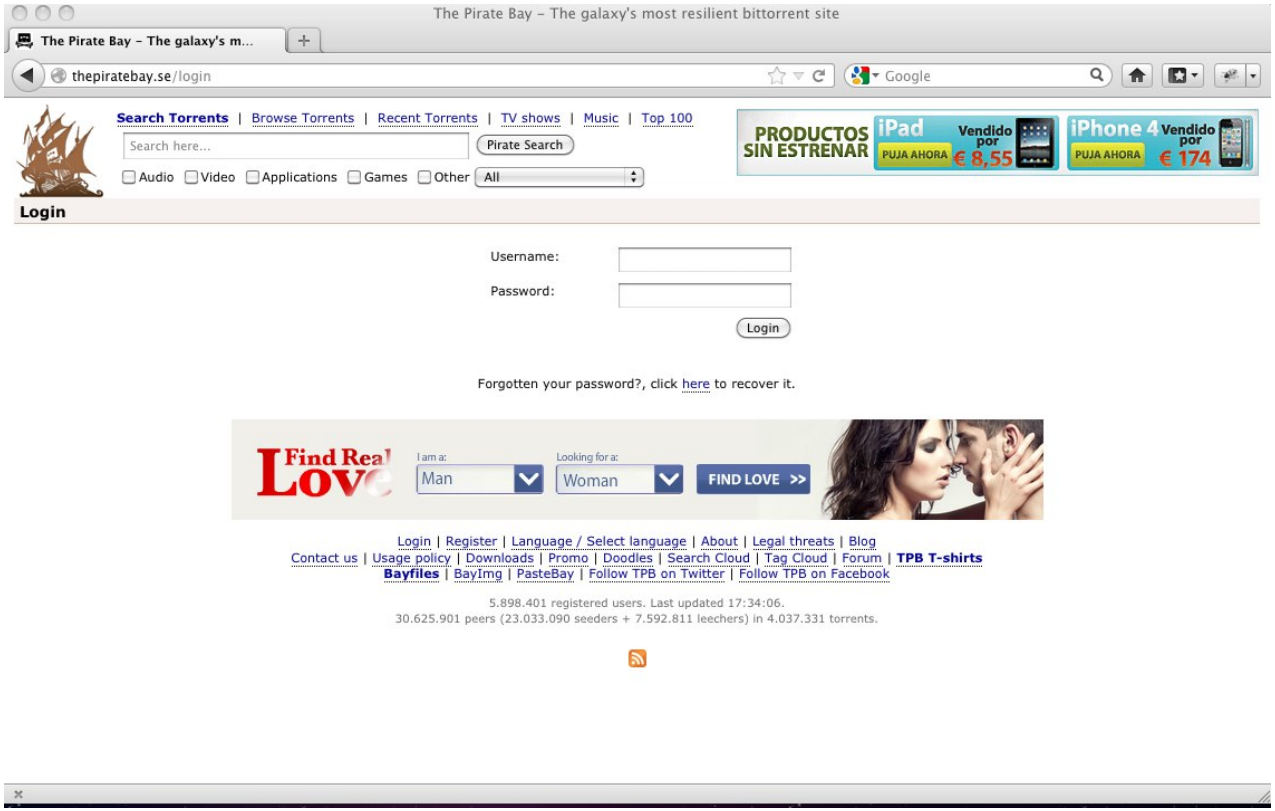
What was the last P2P application you logged in to?

The Easy Case



KTH Computer Science and Communication

Sometimes we Want Authentication



KTH VETENSKAP OCH KONST
KTH Computer Science and Communication

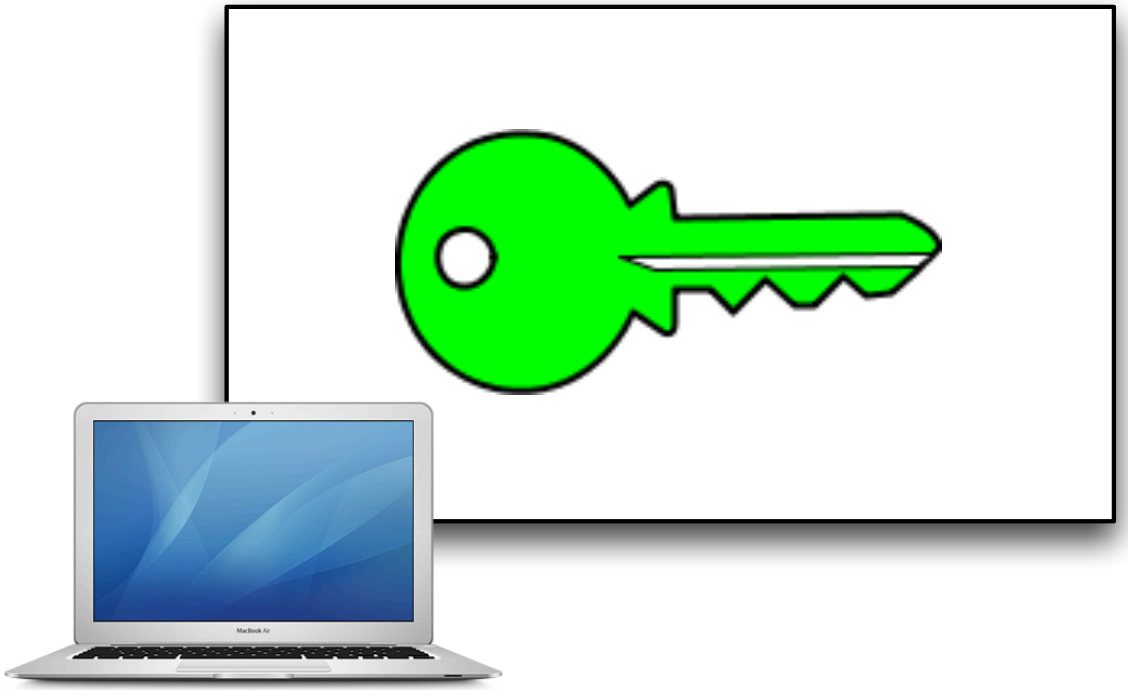
Standard P2P Solution

- Generate a cryptographic key
- Whoever knows key is authenticated as user
- On a single machine it Just Works^{TM}

Standard P2P Solution

- Generate a cryptographic key
- Whoever knows key is authenticated as user
- On a single machine it Just Works^{TM}
- . . . but, needs to be copied to smartphone.

Standard P2P Solution Depicted

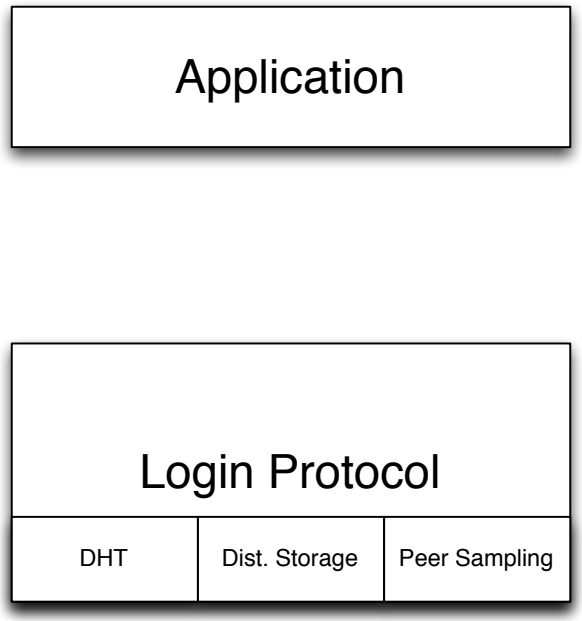


KTH VETENSKAP OCH KONST
KTH Computer Science and Communication

Problem Crops up in P2P Backup

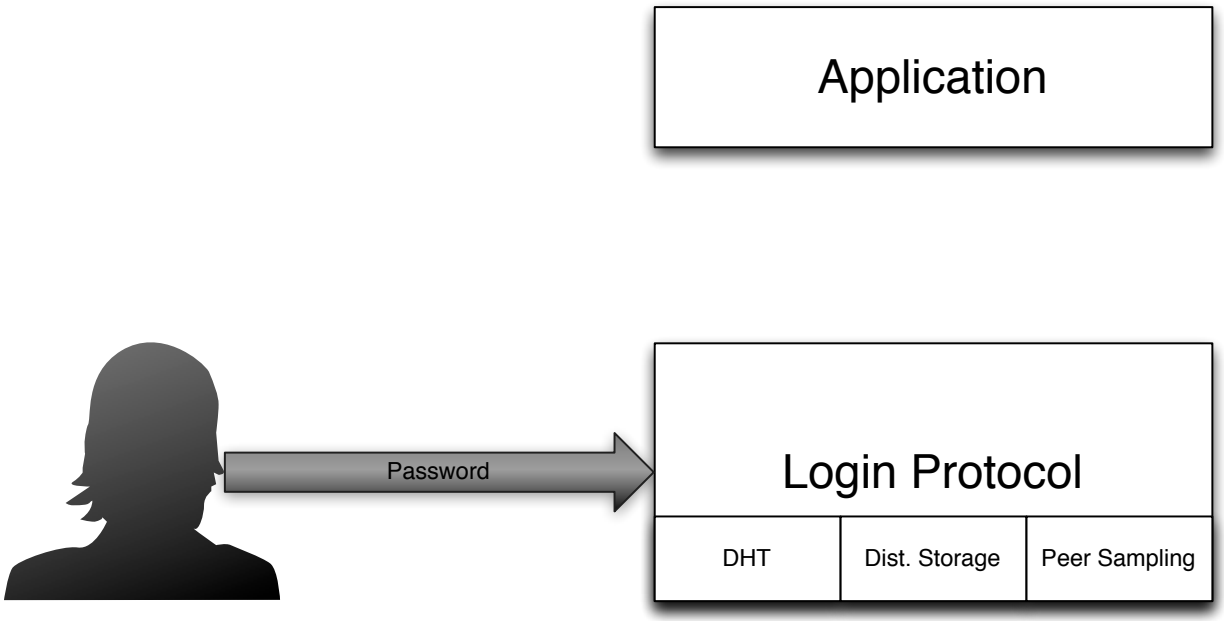
- Default solution fails spectacularly for backup
- Some punt problem to user: back up the key separately
- Some derive master key from a password

Target Scenario



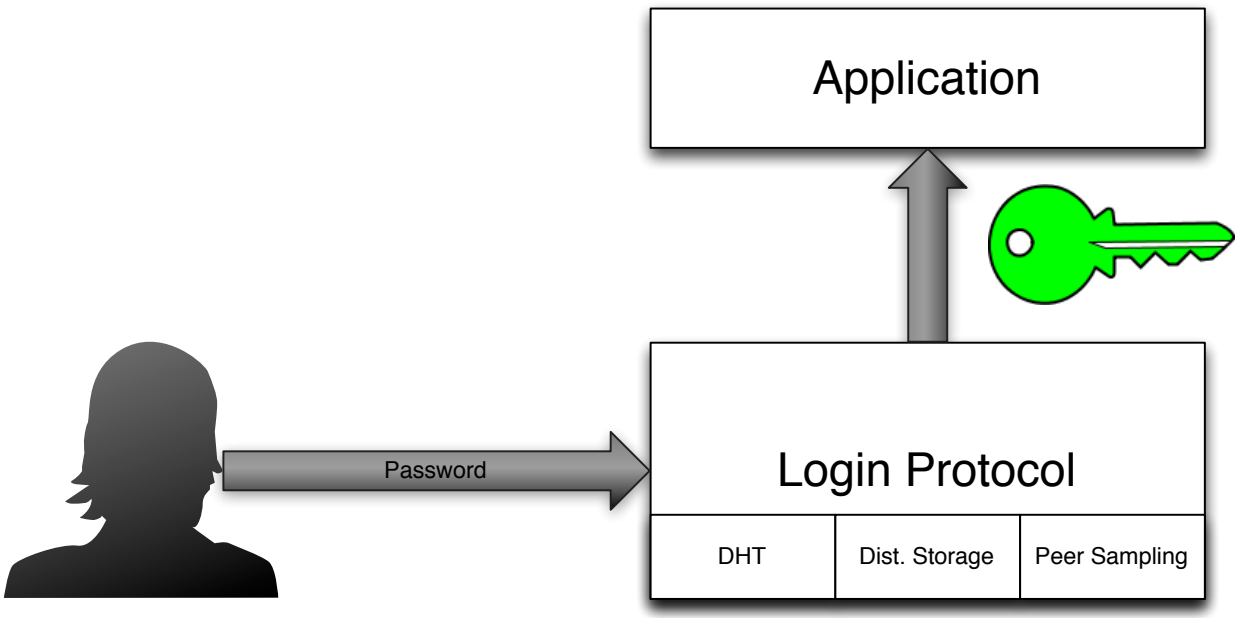
KTH Computer Science and Communication

Target Scenario



KTH Computer Science and Communication

Target Scenario

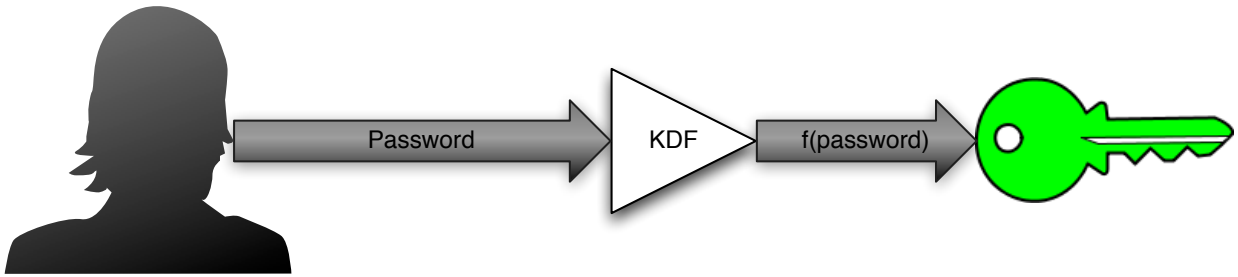


KTH Computer Science and Communication

The Basic Solution

- Basic solution: derive key from user-entered password
- Cryptographic key-derivation function (PBKDF-2, bcrypt)

Basic Solution Depicted



KTH Computer Science and Communication

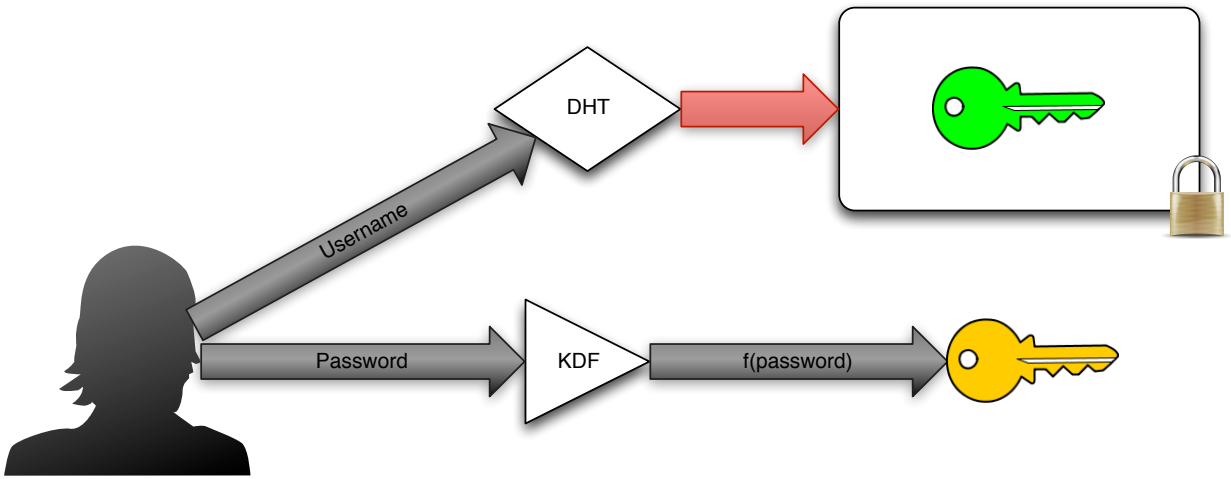
Adding some flexibility

- Tying key we use to password can be limiting
- Add a *key store* where keys we need are stored
- Key store is encrypted with key derived from password

Locating the key store

- How does one find the key store?
- Lookup service (DHT) mapping username to storage location
- Needs to be write-once to prevent DoS

Flexible Basic Solution Depicted



KTH Computer Science and Communication

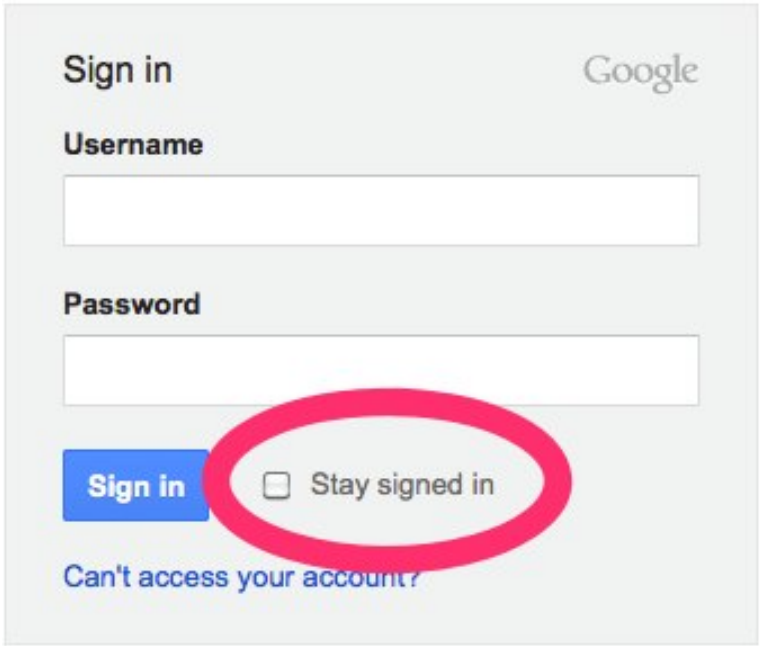
Why not just use a Cloud?

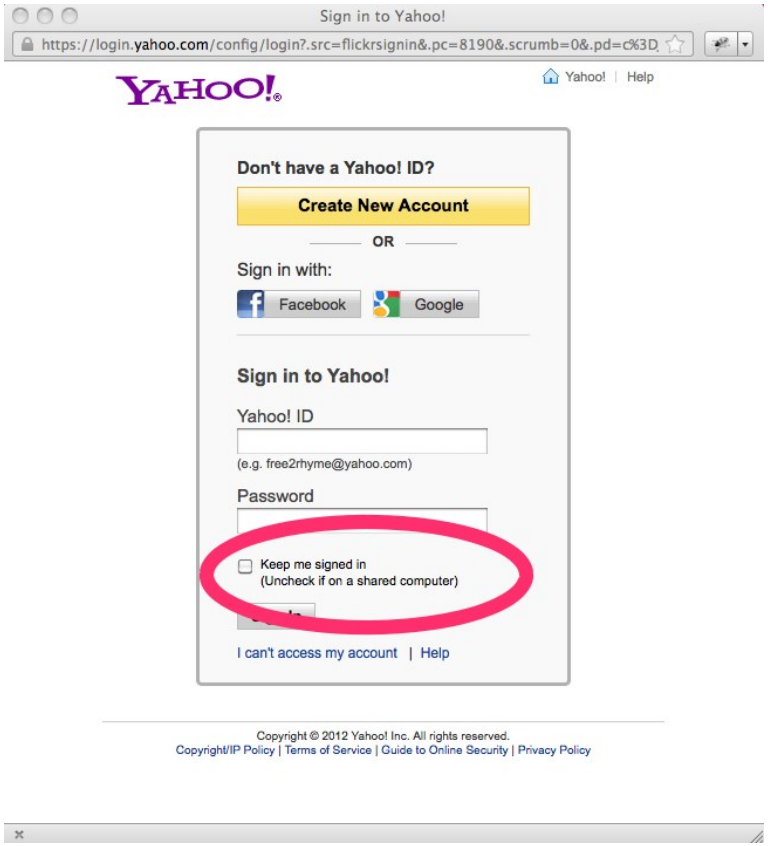
- Censorship resistance
- Administrative cost
- Requires trust in cloud provider

Why not just use a Cloud?

- Censorship resistance
- Administrative cost
- Requires trust in cloud provider
- Our protocol can be deployed on cloud storage

Remember Me





KTH Computer Science and Communication

Requirements

- Password must not be recoverable from device
- Must be able to revoke stored credentials on other devices

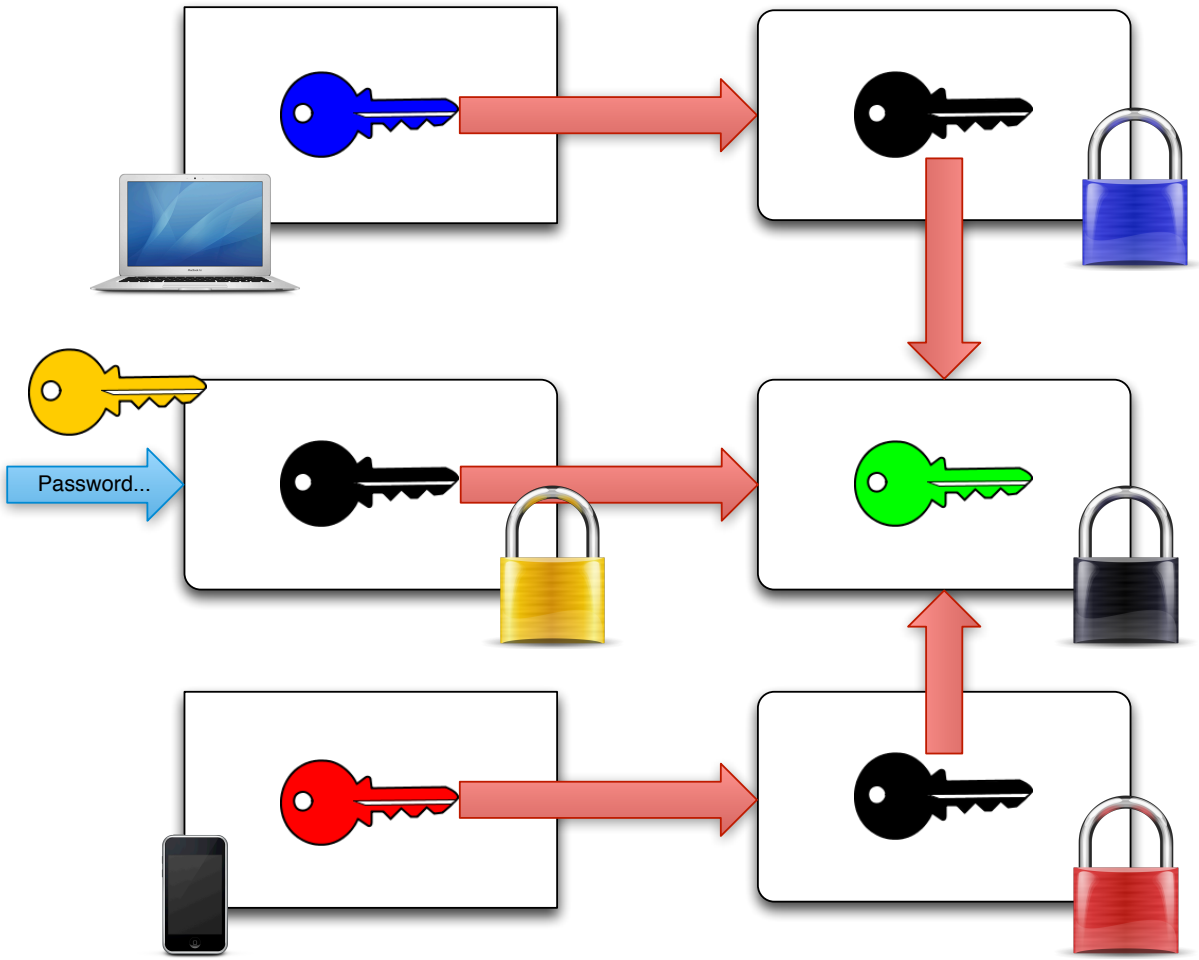
Design

- All problems in computer science can be solved by another level of indirection

Design

- All problems in computer science can be solved by another level of indirection
- For each device, a device login information file in distributed storage
- File contains encrypted copy of key store key
- Device stores key to decrypt its device login file

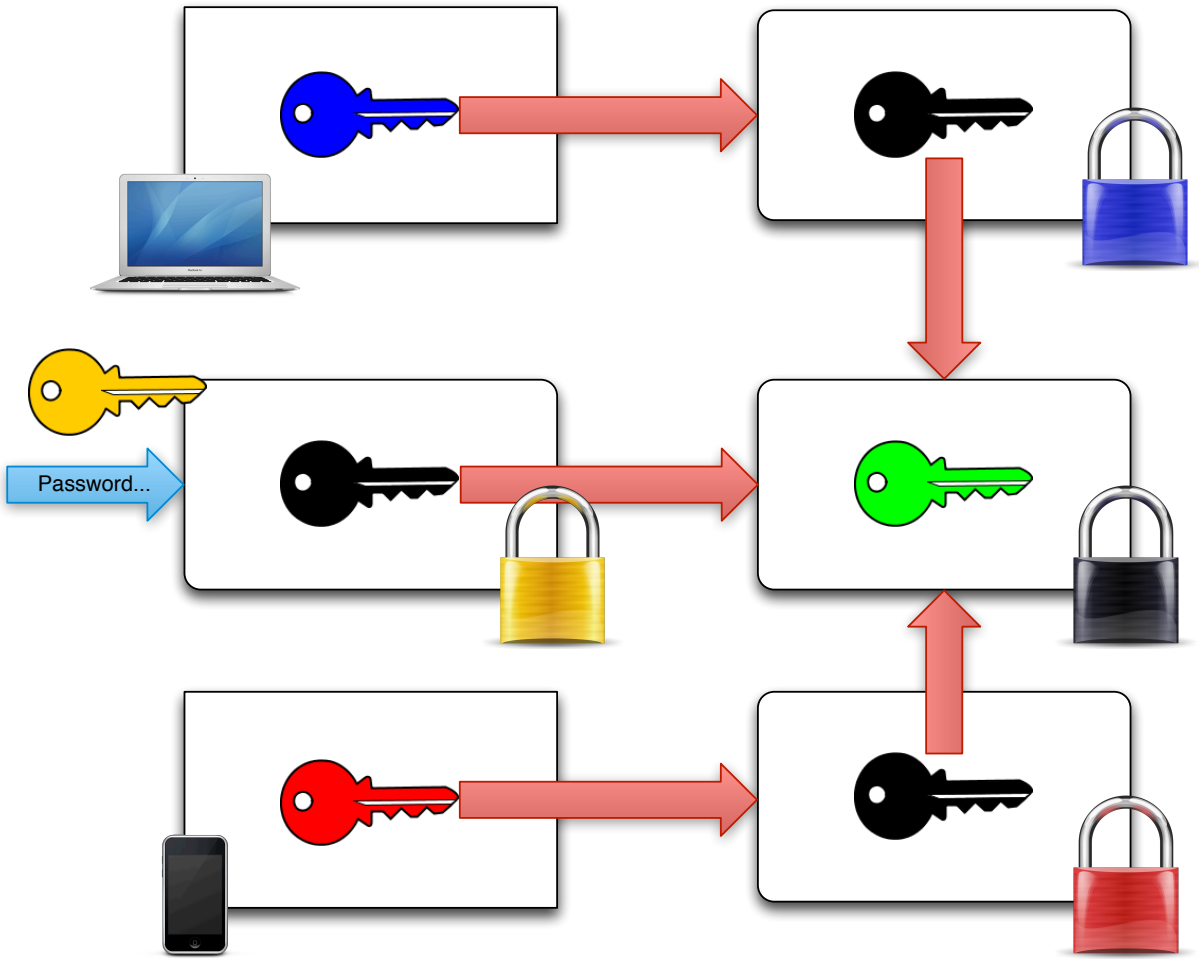
Simplified Remember Me in Pictures



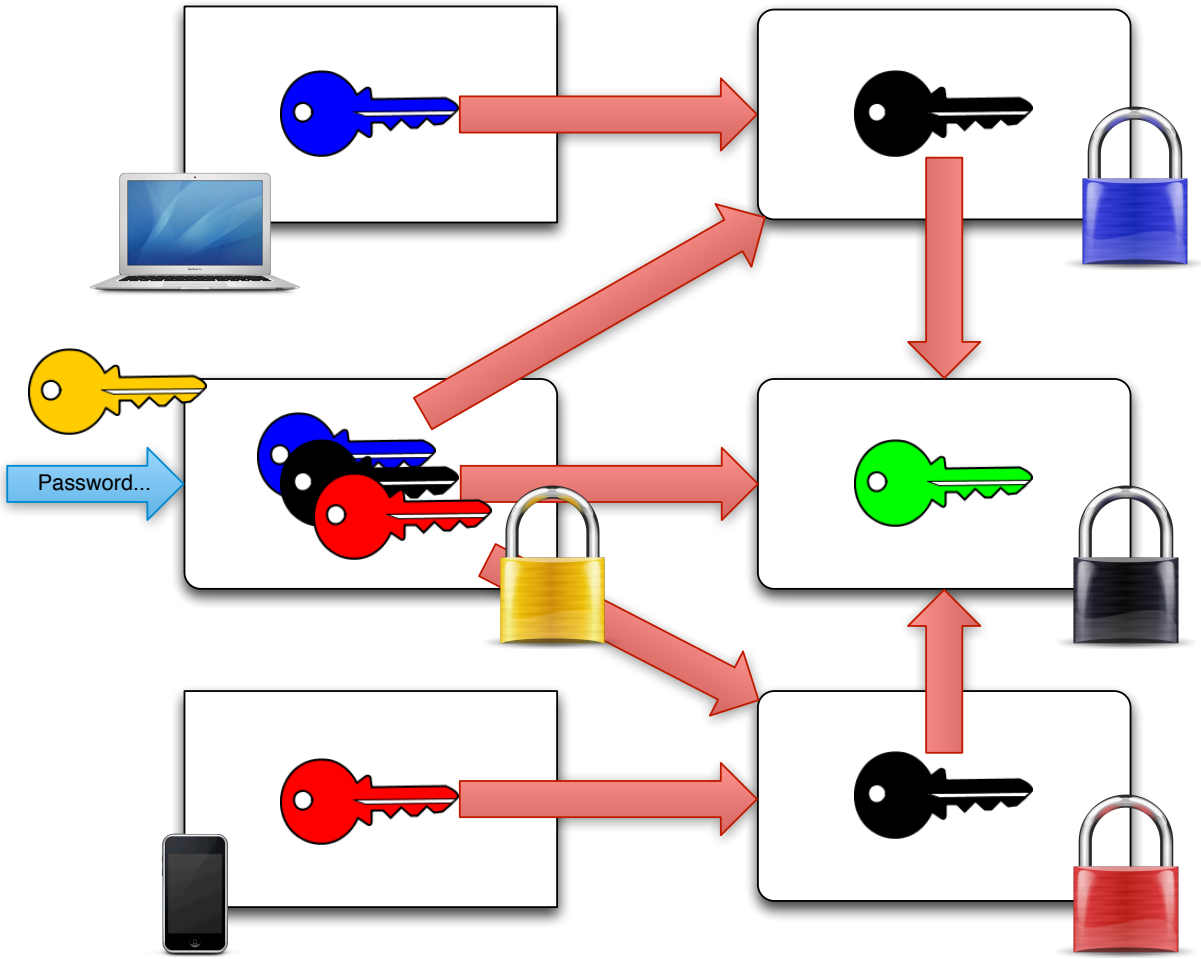
Revokation

- To support revokation, devices' keys are also escrowed
- Key store key is replaced
- All non-revoked devices' login information files are updated
- Application keys from key store must also be updated!

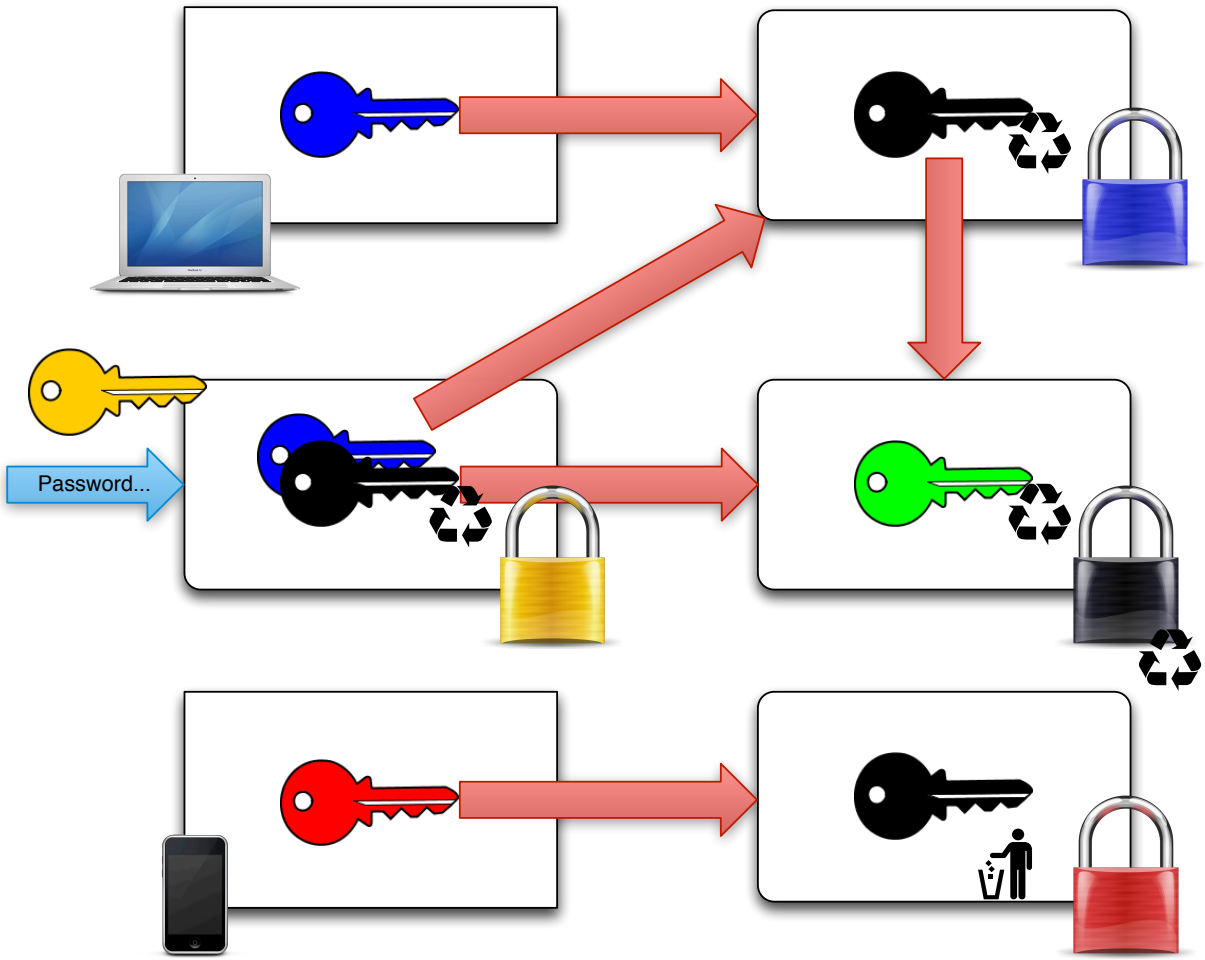
Remember Me in Pictures



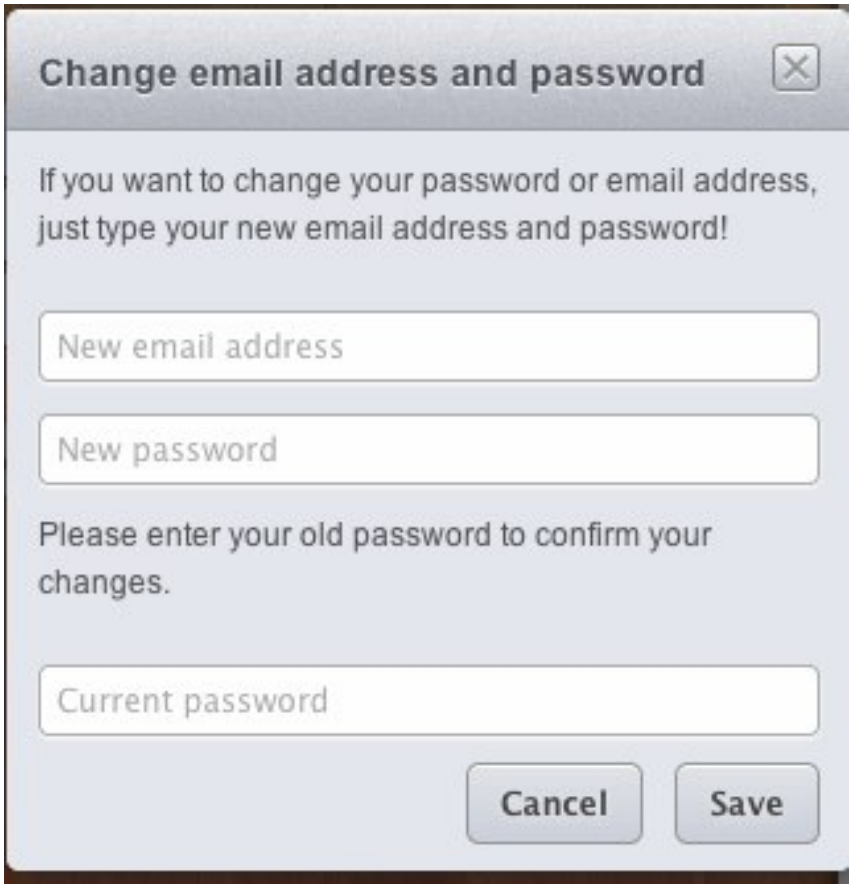
Remember Me in Pictures



Remember Me in Pictures



Password Change



Requirements

- User must know old password to change it

Authentication for distributed storage

- How to authenticate for the distributed storage (Catch-22)?

Authentication for distributed storage

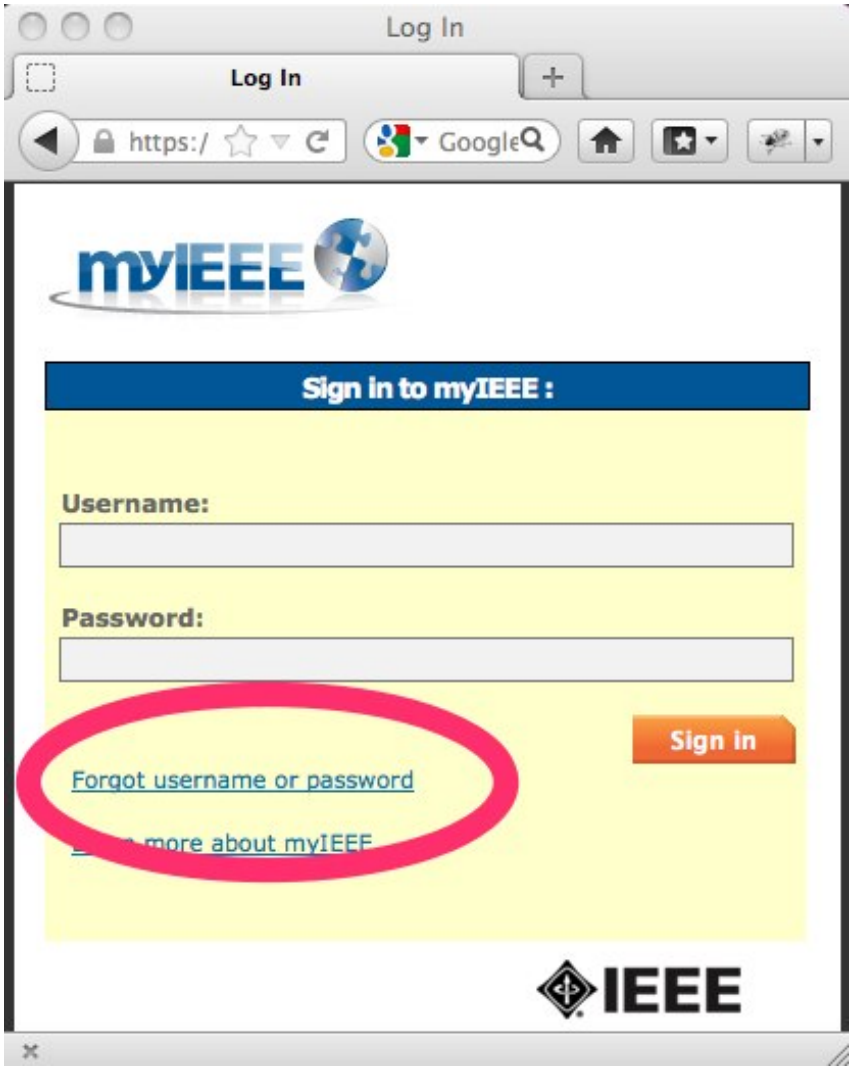
- How to authenticate for the distributed storage (Catch-22)?
- With cryptographic keys, of course! (It's a normal P2P protocol)
- All files are world-readable, only writing requires key
- Write-permission for most files from key store
- Write-permission for login info only if one can decrypt login info

KTH Computer Science and Communication

Design

- Using password, retrieve key needed to update login info
- Generate new key for key store
- Store new key store
- Update login info to point to new key store
- Update/remove device login information files

Forgotten Passwords



Requirements

- Should not reveal secret information to others
- Based on e-mail and/or security questions

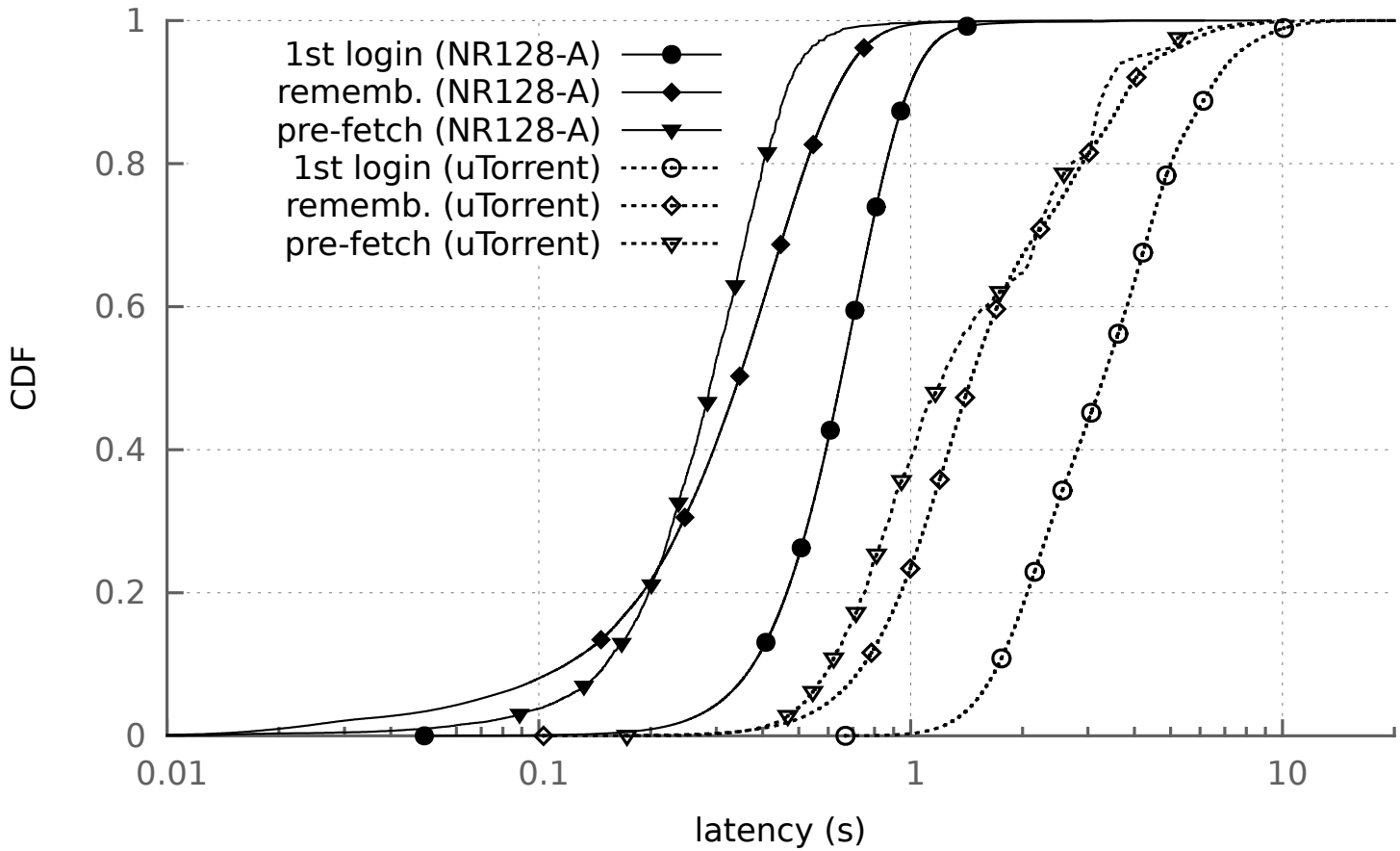
Design

- Two separate schemes for e-mail and questions
- Building on secret sharing
- For questions, shares are encrypted with key derived from answers
- For e-mail, shares are spread among random peers
- Using cryptographic commitments to only reveal e-mail address when actually used

Evaluation Setting

- Simulation using timing from Bittorrent DHT
- DHT timing by Jiménez et al (P2P'11)
- Excludes crypto computation (negligible aside from KDF)

Evaluation Results



KTH Computer Science and Communication

Future Work

- Security proofs
- Prevent offline guessing attacks
- Implementation

Summary

- Fully distributed protocol supporting main features of password-based login
- Including “Remember me”, password change, password reset
- Room for future improvements

KTH Computer Science and Communication