A Formal Treatment of End-to-End Encrypted Cloud Storage

Matilda Backendal[1], Hannah Davis[2], Felix Günther[3], Miro Haller[4], Kenny Paterson[1]

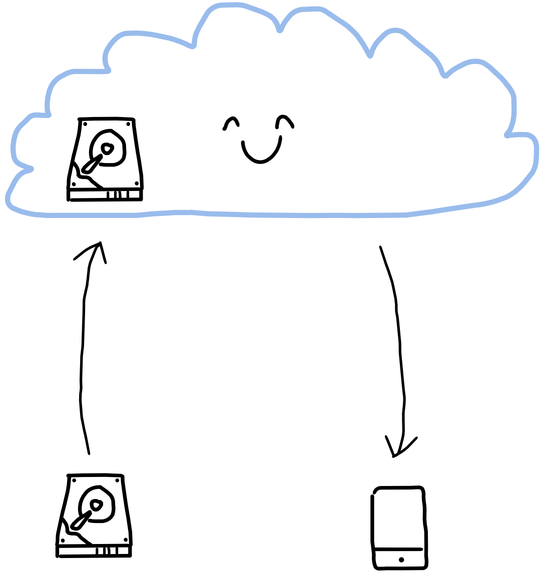
[1]ETH Zurich , [2]Seagate Technology, [3]IBM Research Zurich, [4]UC San Diego

Berkeley Security Seminar, September 4, 2024

Cloud Storage

Benefits:

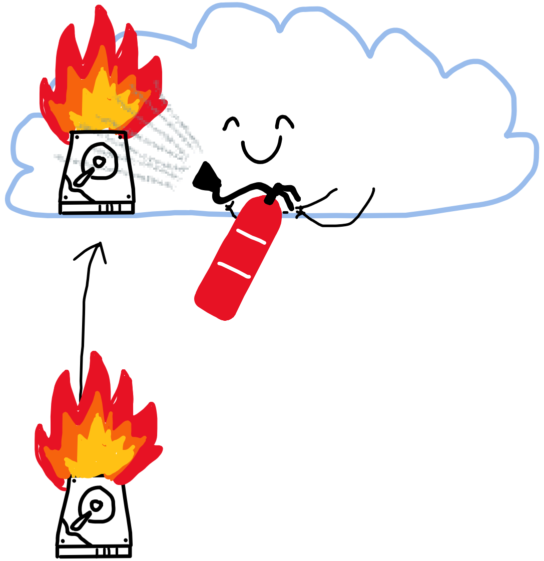
- + Availability



Cloud Storage

Benefits:

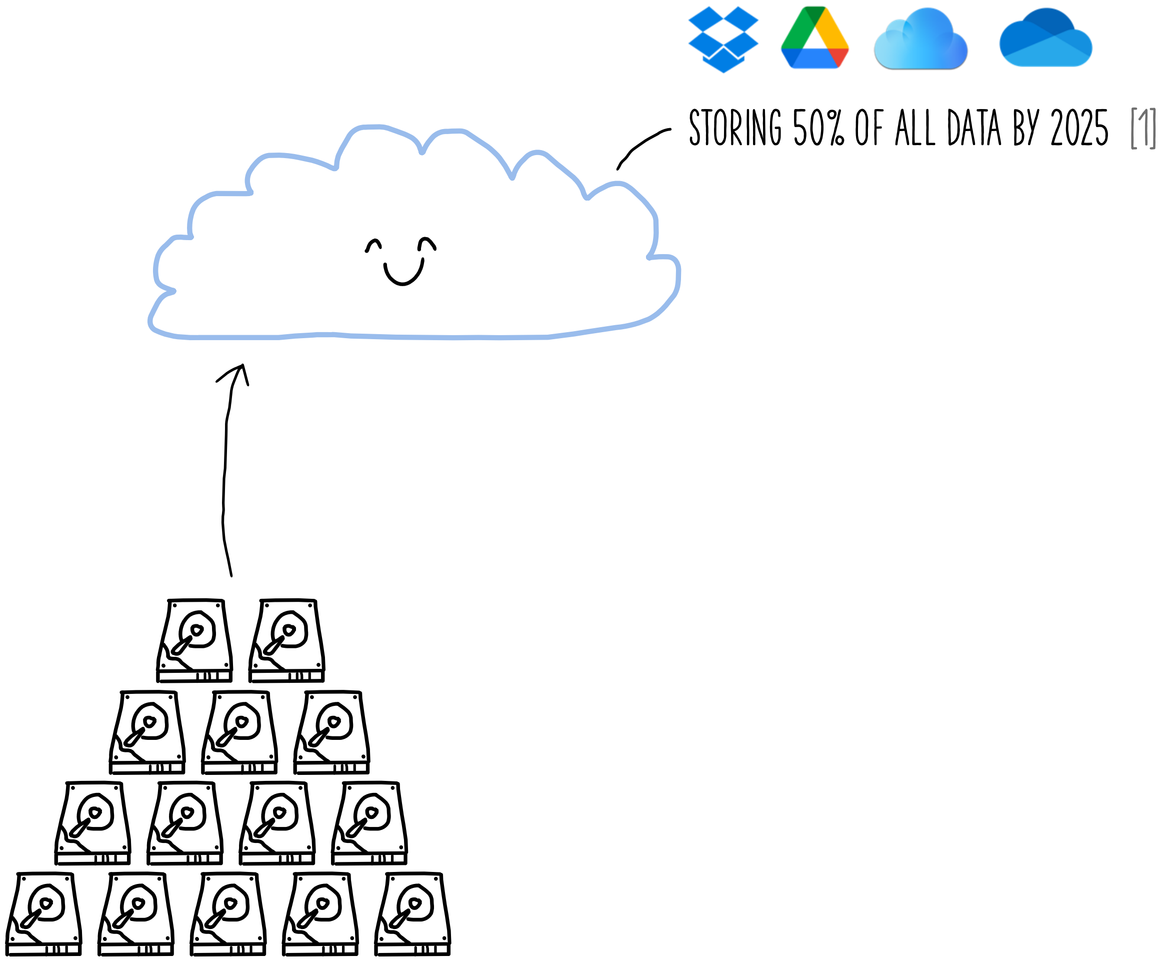
- + Availability
- + Redundancy



Cloud Storage

Benefits:

- + Availability
- + Redundancy
- + Scalability



[1] https://cybersecurityventures.com/the-world-will-store-200-zettabytes-of-data-by-2025/ (Sausalito, Calif., Feb. 1, 2024)

Cloud Storage

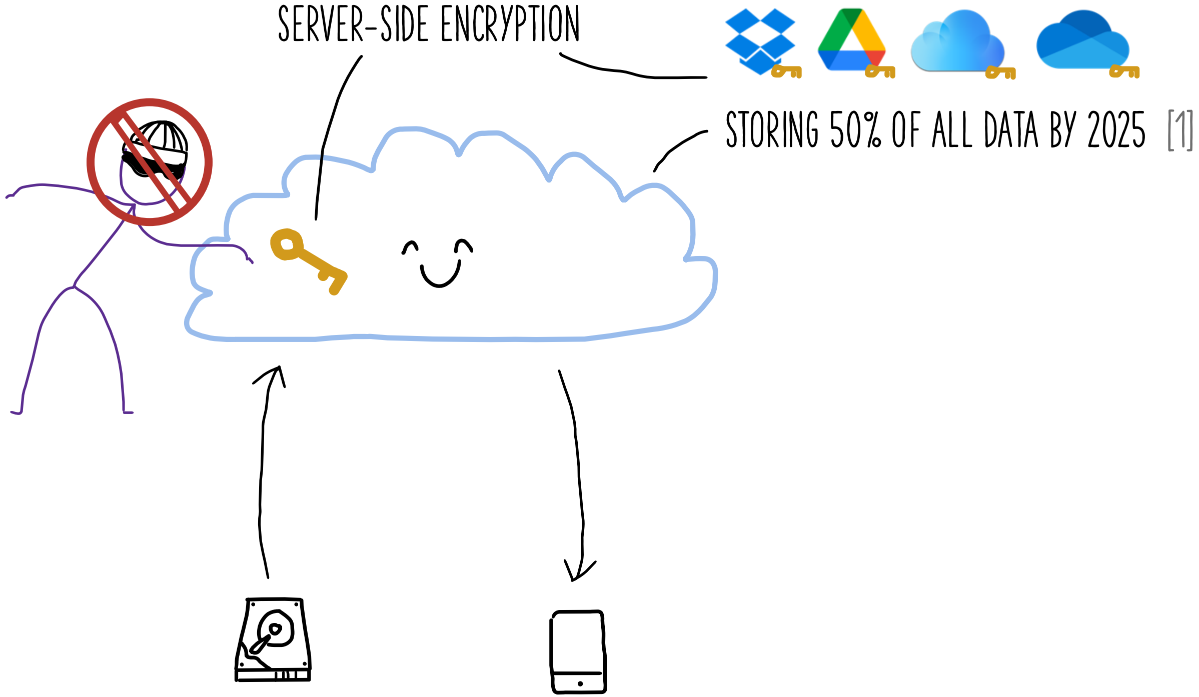
Benefits:

- + Availability
- + Redundancy
- + Scalability

Concerns:

- - Data leaks to third party

=> SERVER-SIDE ENCRYPTION



[1] https://cybersecurityventures.com/the-world-will-store-200-zettabytes-of-data-by-2025/ (Sausalito, Calif., Feb. 1, 2024)

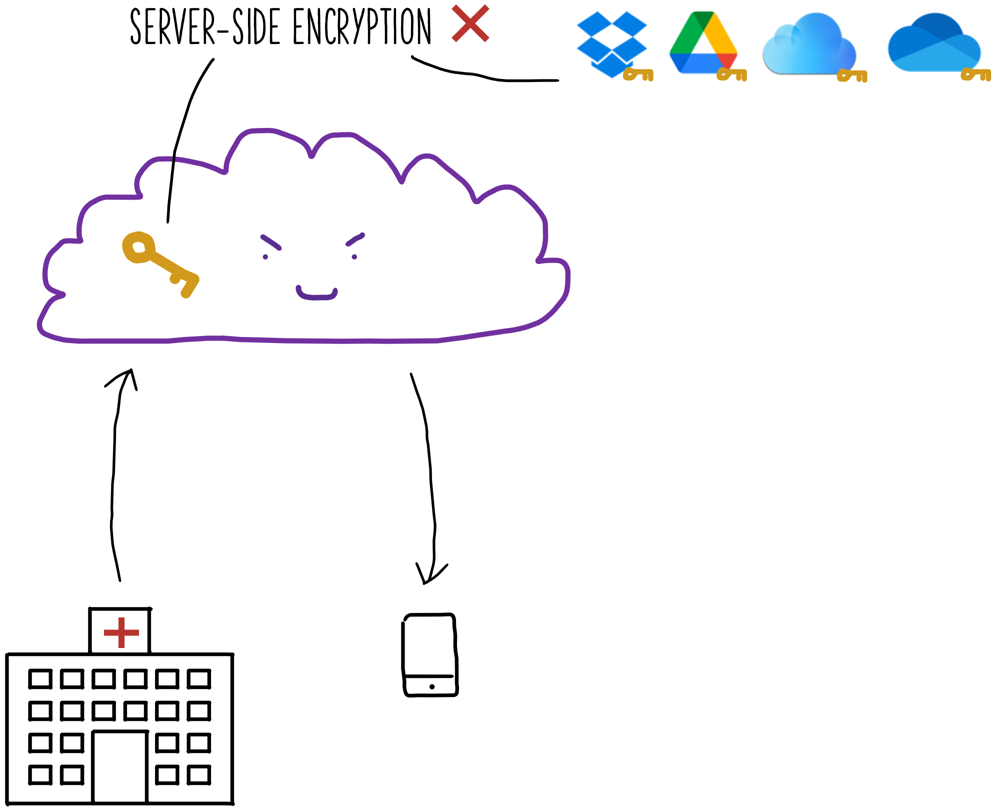
Cloud Storage

Benefits:

+ Availability
+ Redundancy
+ Scalability

Concerns:

- Data leaks to third party
  => SERVER-SIDE ENCRYPTION
- Malicious server
  => END-TO-END ENCRYPTION



https://www.hipaajournal.com/healthcare-cloud-usage-grows-but-protecting-phi-can-be-a-challenge/

Cloud Storage

Benefits:

+ Availability
+ Redundancy
+ Scalability

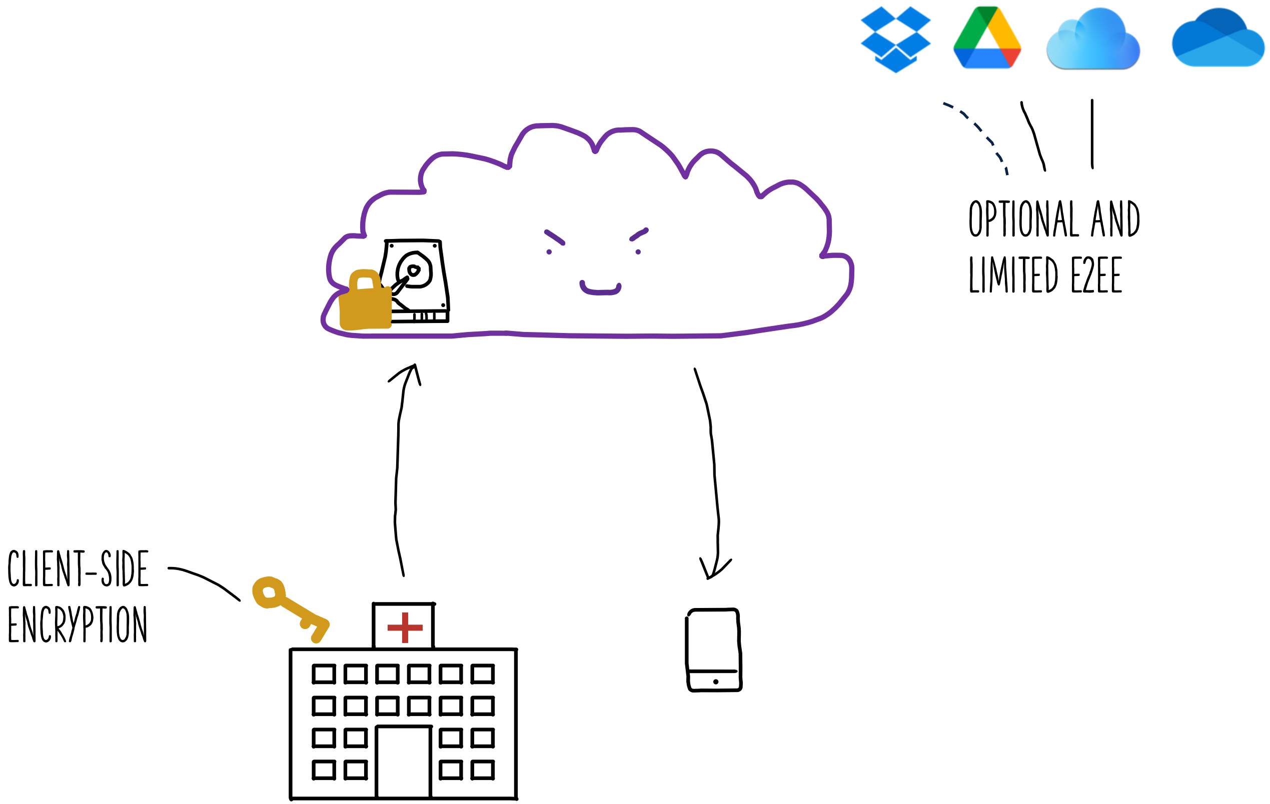
Concerns:

- Data leaks to third party

  => SERVER-SIDE ENCRYPTION

- Malicious server

  => END-TO-END ENCRYPTION



https://www.hipaajournal.com/healthcare-cloud-usage-grows-but-protecting-phi-can-be-a-challenge/

E2EE Cloud Storage

INSECURE!

"WITH **MEGA**, YOU CONTROL THE ENCRYPTION"

300 MILLION USERS

MEGA

[SP:BHP23]
[EC:AHMP23]

INSECURE!

AMNESTY INTERNATIONAL,
THE GERMAN FEDERAL GOVERNMENT
& ETH

"ULTIMATE SECURITY"

Nextcloud

[EuroSP:ABCP23]

INSECURE!

"EXCEPTIONALLY PRIVATE CLOUD"

sync.com

pCloud

"EUROPE'S MOST SECURE CLOUD STORAGE"

"THE STRONGEST ENCRYPTED CLOUD STORAGE IN THE WORLD"

icedrive

[CCS:TH24]

Seafile™

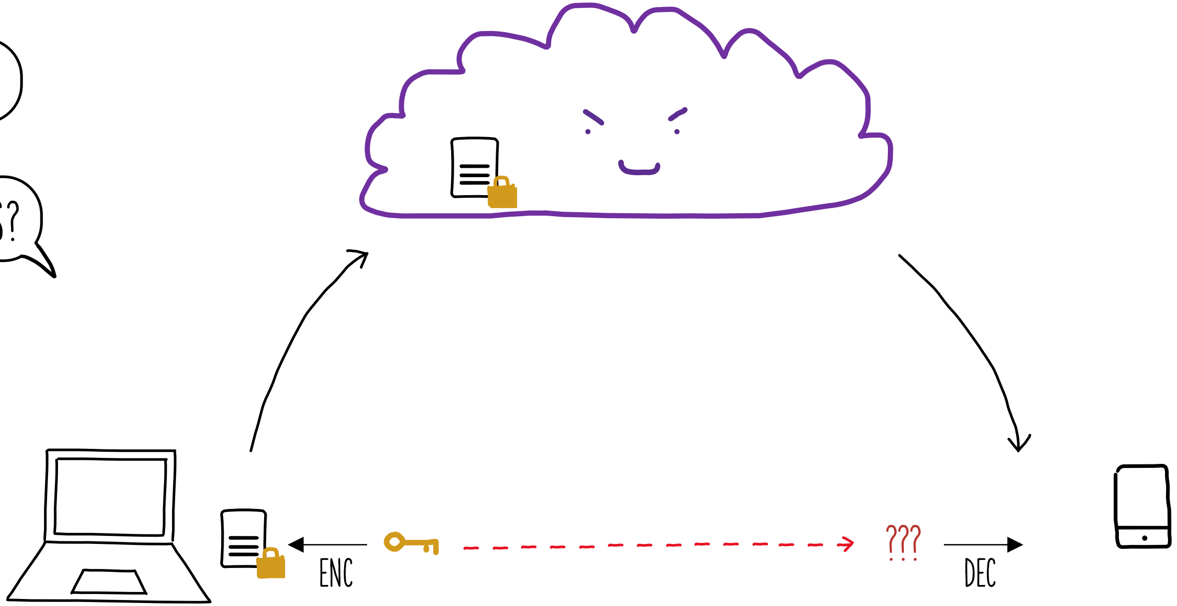
"SUPPORTS CLIENT-SIDE END-TO-END ENCRYPTION"

Why Is It Hard?

JUST USE YOUR FAVORITE AEAD SCHEME FOR CLIENT-SIDE ENCRYPTION!

HOW DO YOU TRANSFER KEYS BETWEEN DEVICES?

1	key distribution

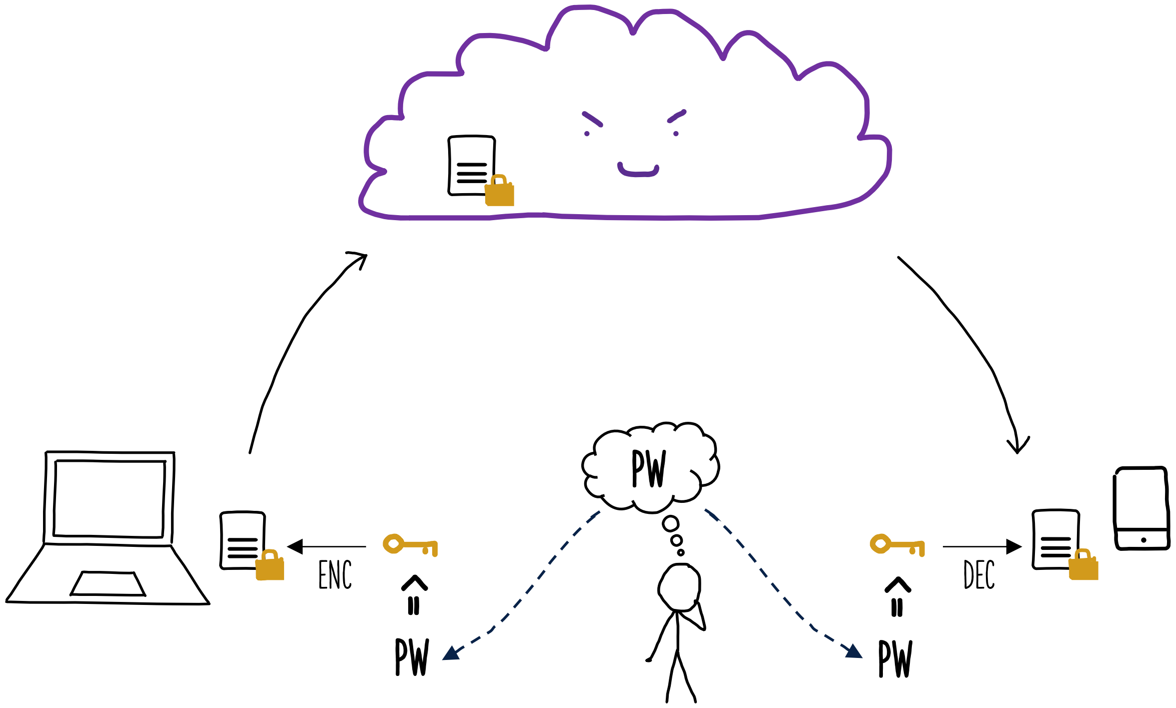


Why Is It Hard?

DERIVE KEYS FROM THE PASSWORD!

1 key distribution

2 password-based security



Why Is It Hard?

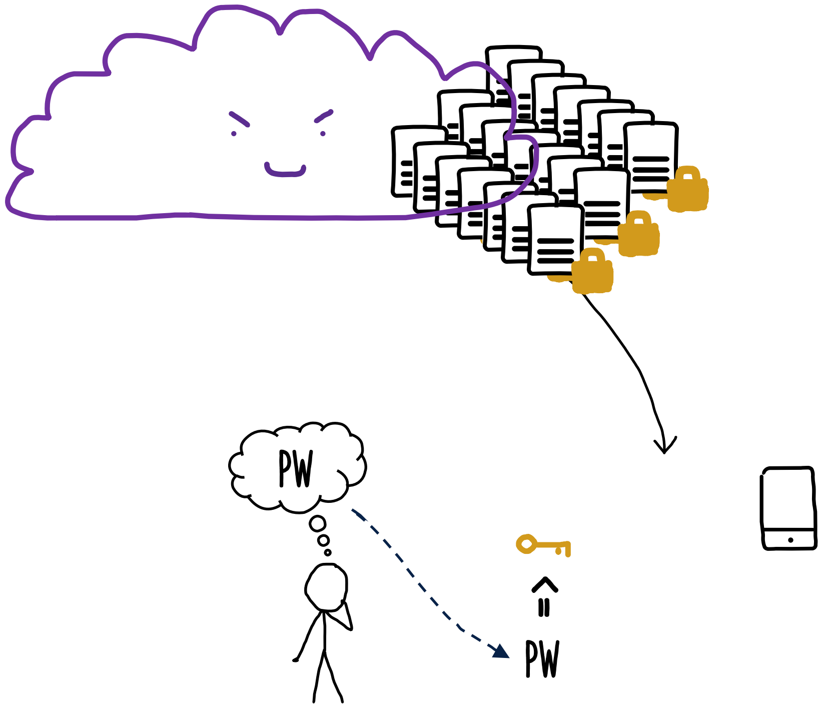
DERIVE KEYS FROM THE PASSWORD!

WHAT IF THE PASSWORD CHANGES?

1 key distribution

2 password-based security

PROBLEM 1: PW CHANGE



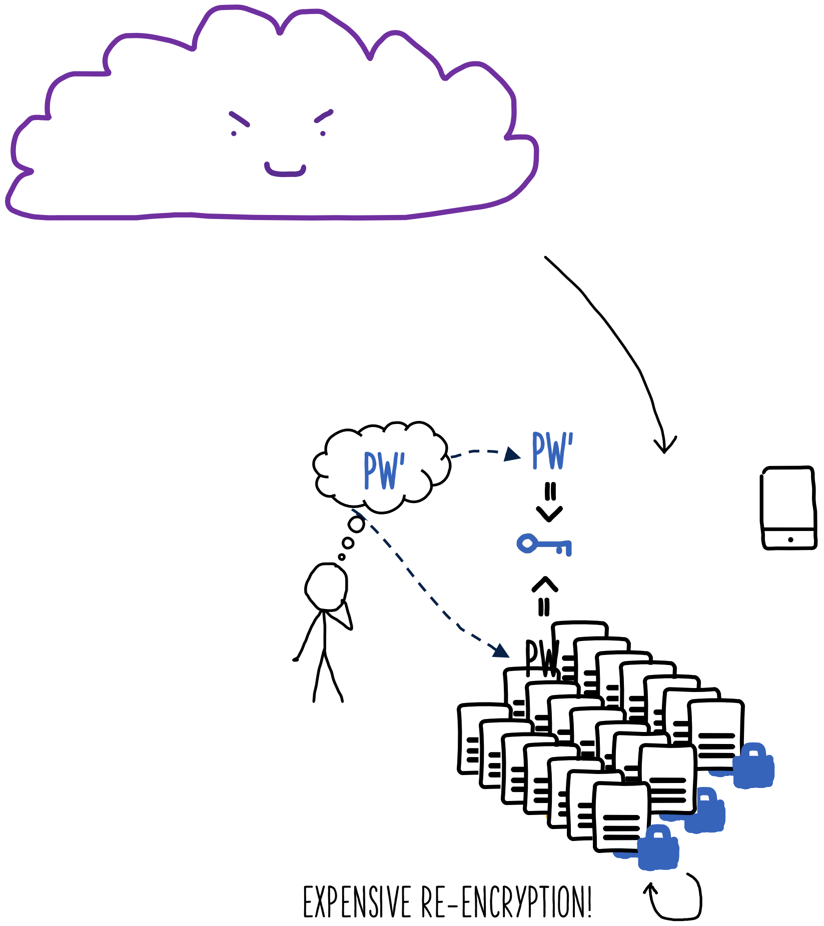
Why Is It Hard?

DERIVE KEYS FROM THE PASSWORD!

WHAT IF THE PASSWORD CHANGES?

1. key distribution
2. password-based security

PROBLEM 1: PW CHANGE



Why Is It Hard?

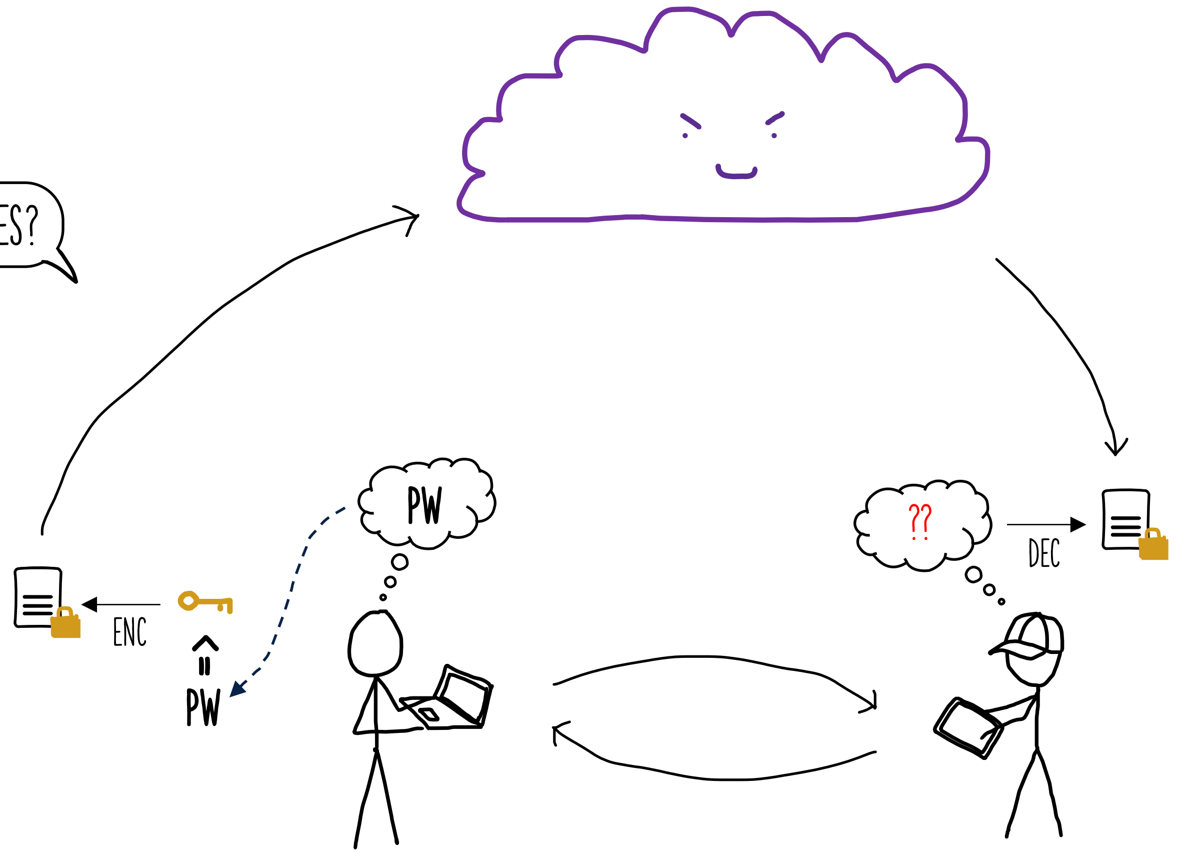
DERIVE KEYS FROM THE PASSWORD!

HOW DO YOU SHARE FILES?

1. key distribution
2. password-based security
3. file sharing

PROBLEM 1: PW CHANGE

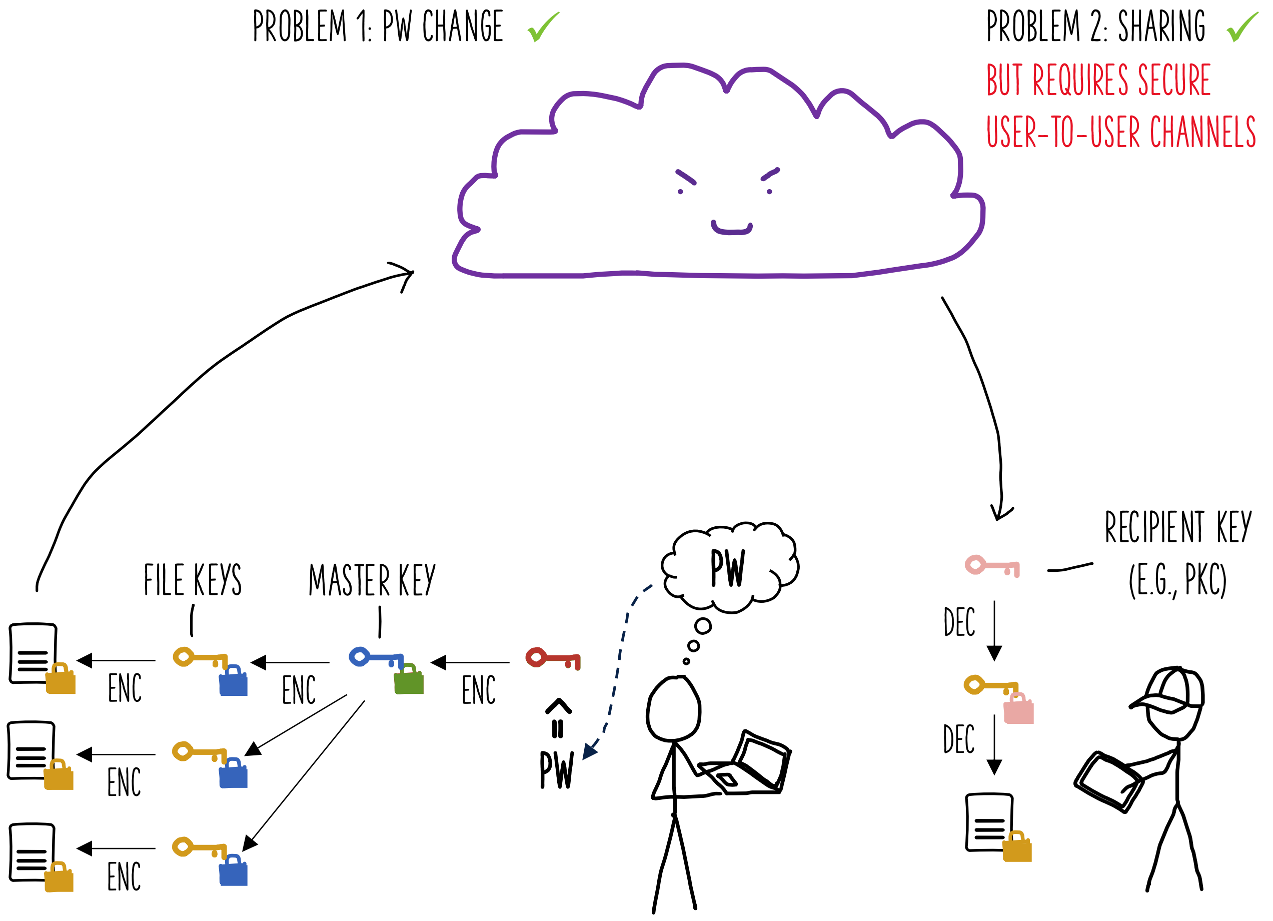
PROBLEM 2: SHARING



Why Is It Hard?

BUILD A KEY HIERARCHY!

1	key distribution
2	password-based security
3	file sharing



Why Is It Hard?

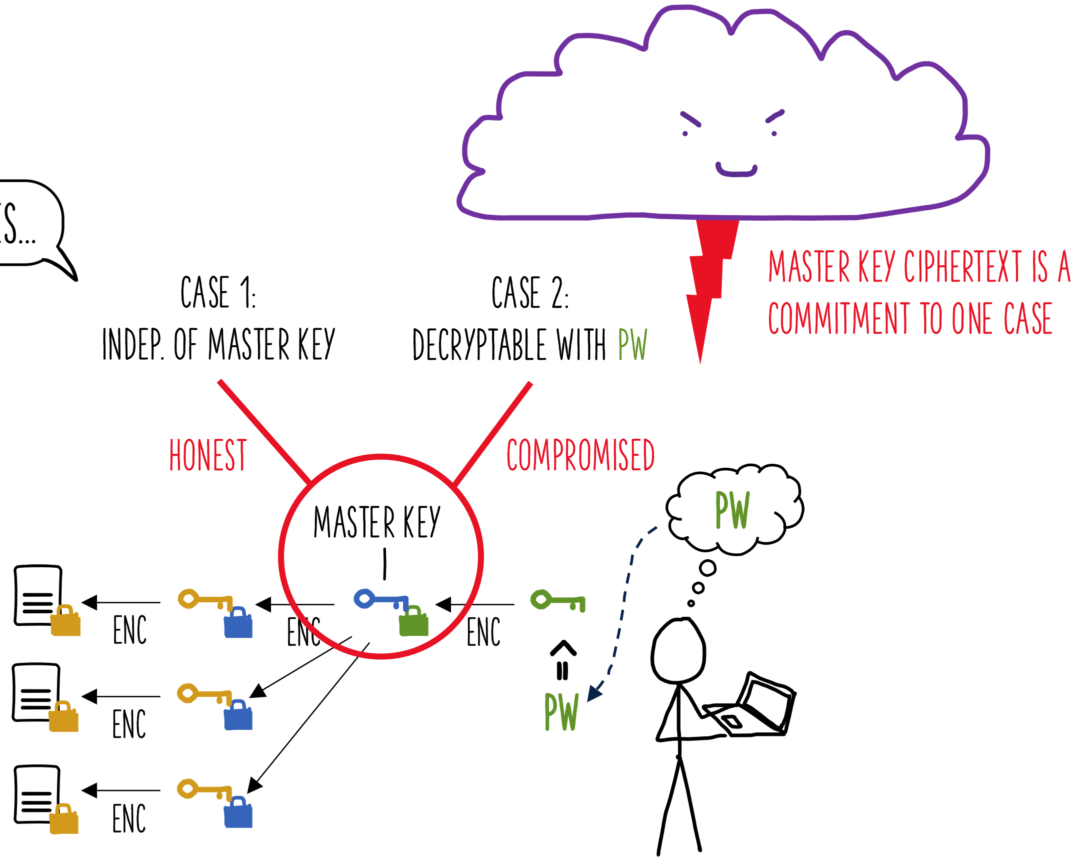
BUILD A KEY HIERARCHY!

WORKS, BUT LEADS TO COMMITMENT ISSUES...

1 key distribution

2 password-based security

3 file sharing



Why Is It Hard?

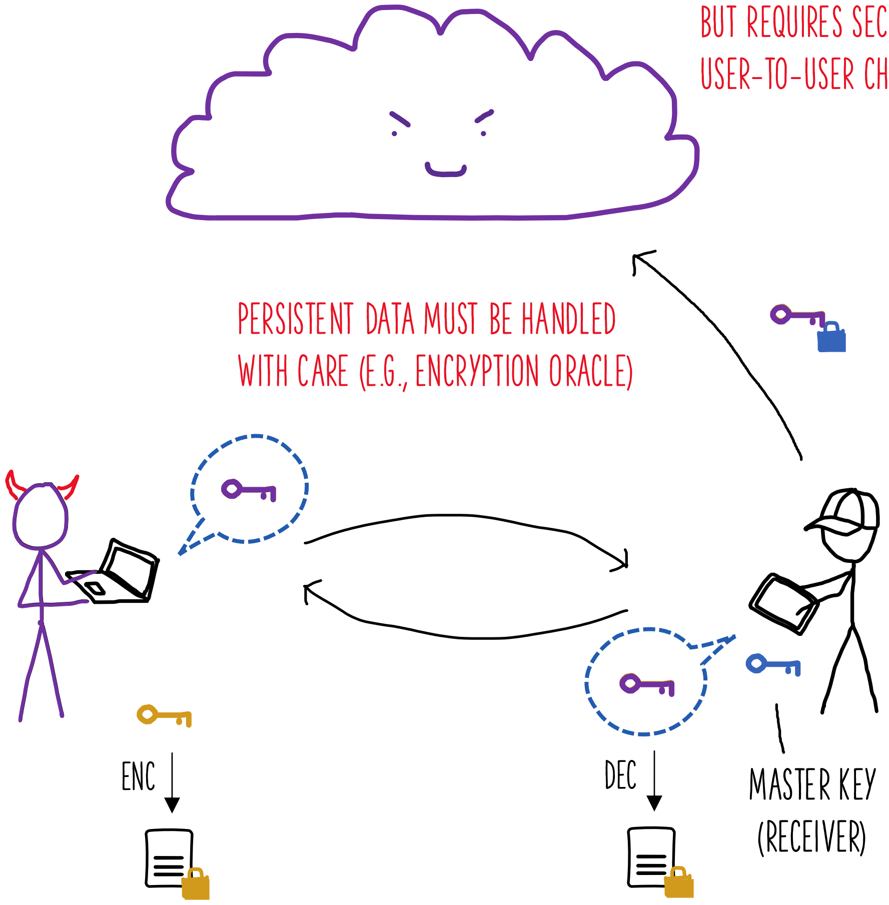
USE SECURE MESSAGING TECHNIQUES!

HOW TO PROTECT DATA AT REST?

1	key distribution
2	password-based security
3	file sharing
4	persistent data

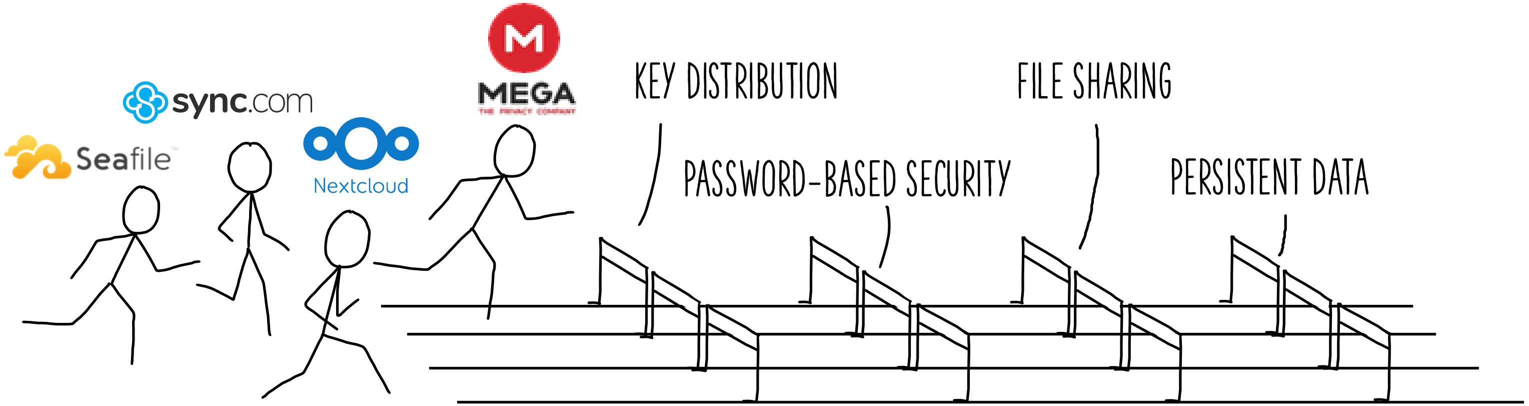
PROBLEM 2: SHARING ✓

BUT REQUIRES SECURE USER-TO-USER CHANNELS

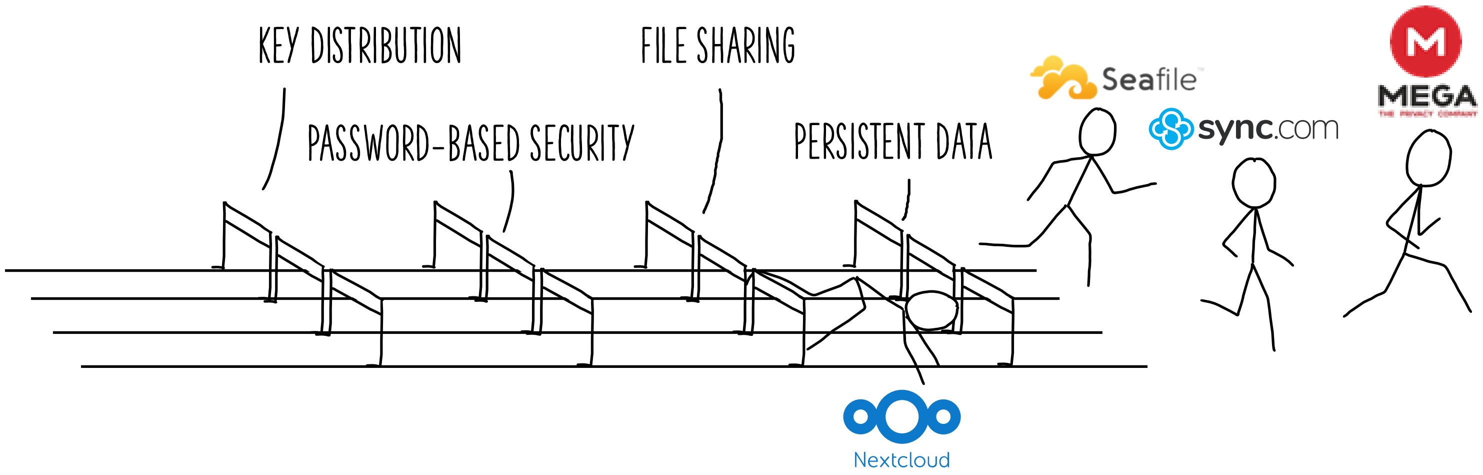


PERSISTENT DATA MUST BE HANDLED WITH CARE (E.G., ENCRYPTION ORACLE)

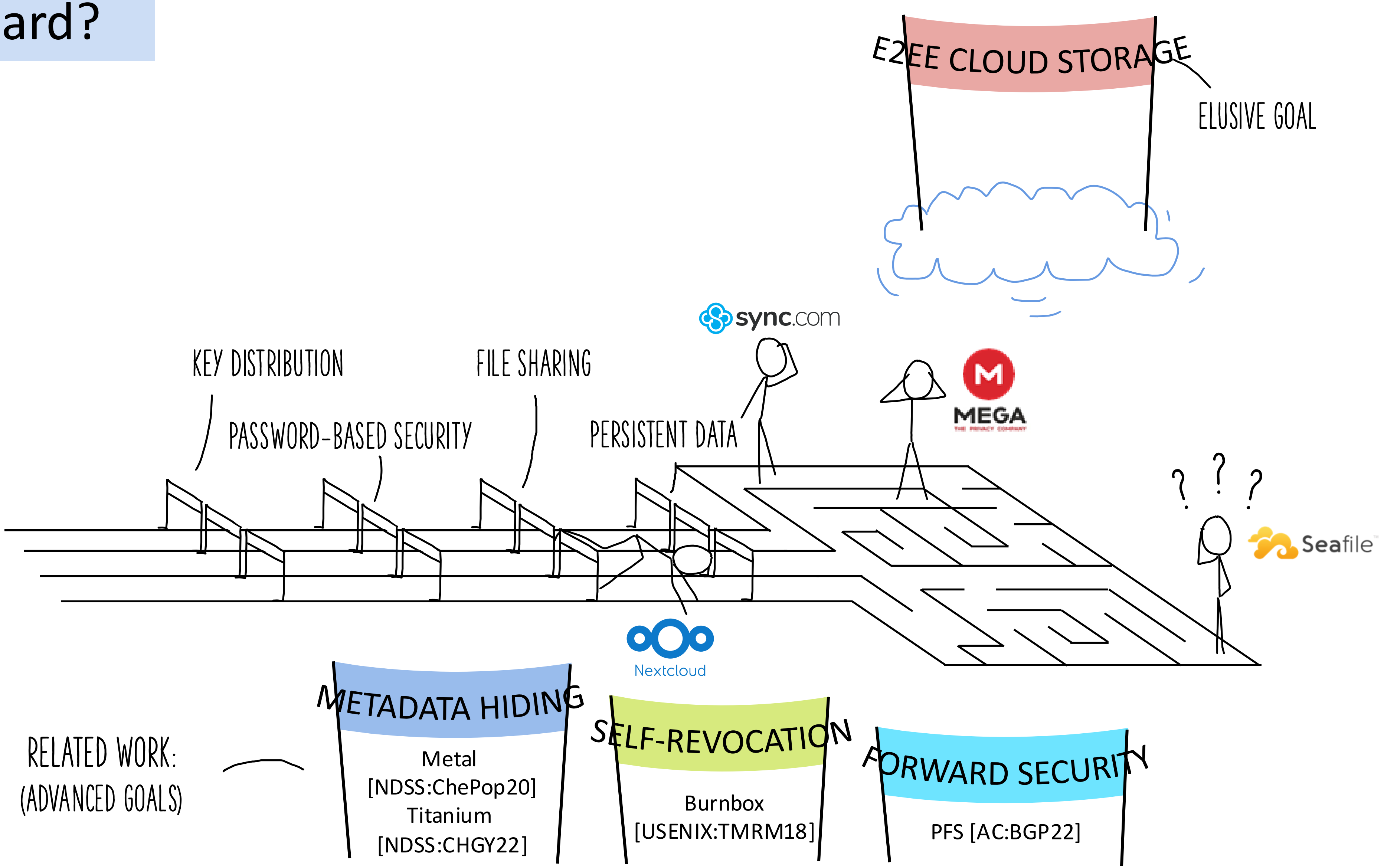
Why Is It Hard?



Why Is It Hard?



Why Is It Hard?



RELATED WORK:
(ADVANCED GOALS)

METADATA HIDING

Metal
[NDSS:ChePop20]
Titanium
[NDSS:CHGY22]

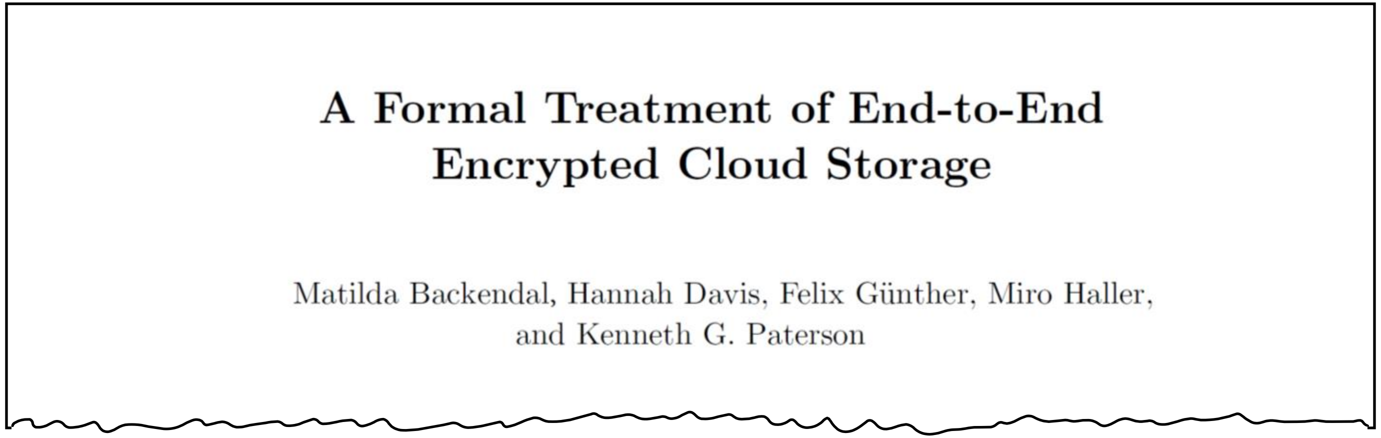
SELF-REVOCATION

Burnbox
[USENIX:TMRM18]

FORWARD SECURITY

PFS [AC:BGP22]

Contributions



A Formal Treatment of End-to-End Encrypted Cloud Storage

Matilda Backendal, Hannah Davis, Felix Günther, Miro Haller,
and Kenneth G. Paterson

1 Formal Model

- Syntax
- Security games

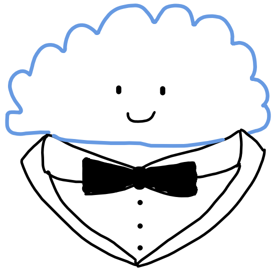
2 Construction

- CSS (Cloud Storage Scheme)
- Security proofs

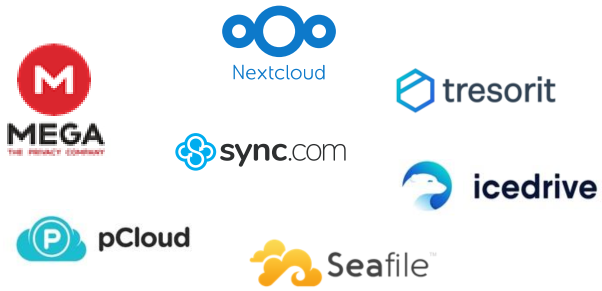
1. Formalizing E2EE Cloud Storage

Formalizing E2EE Cloud Storage

Model Goals



ALL MODELS ARE WRONG,
BUT SOME ARE USEFUL!

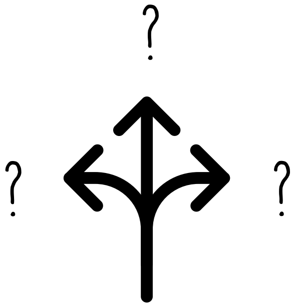


Capture existing systems

1 Expressive

Capture *real-world* systems

2 Faithful



Capture future systems

3 Generic

Syntax

WHAT MAKES A CLOUD STORAGE A CLOUD STORAGE?

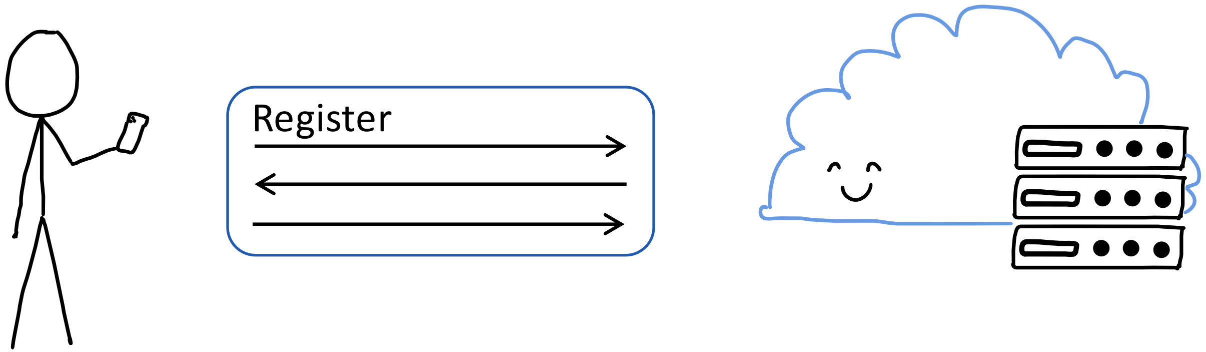
1 EXPRESSIVE ✓

Core Functionality

- Register (create account)
- Authenticate (log in)

- Put (upload a file)
- Update (modify content)
- Get (download)
- Share
- Accept (receive share)

INTERACTIVE PROTOCOLS



Syntax

HOW DO WE MAKE THE MODEL USEFUL?

1 EXPRESSIVE ✓

Core Functionality

- Register (create account)
- Authenticate (log in)

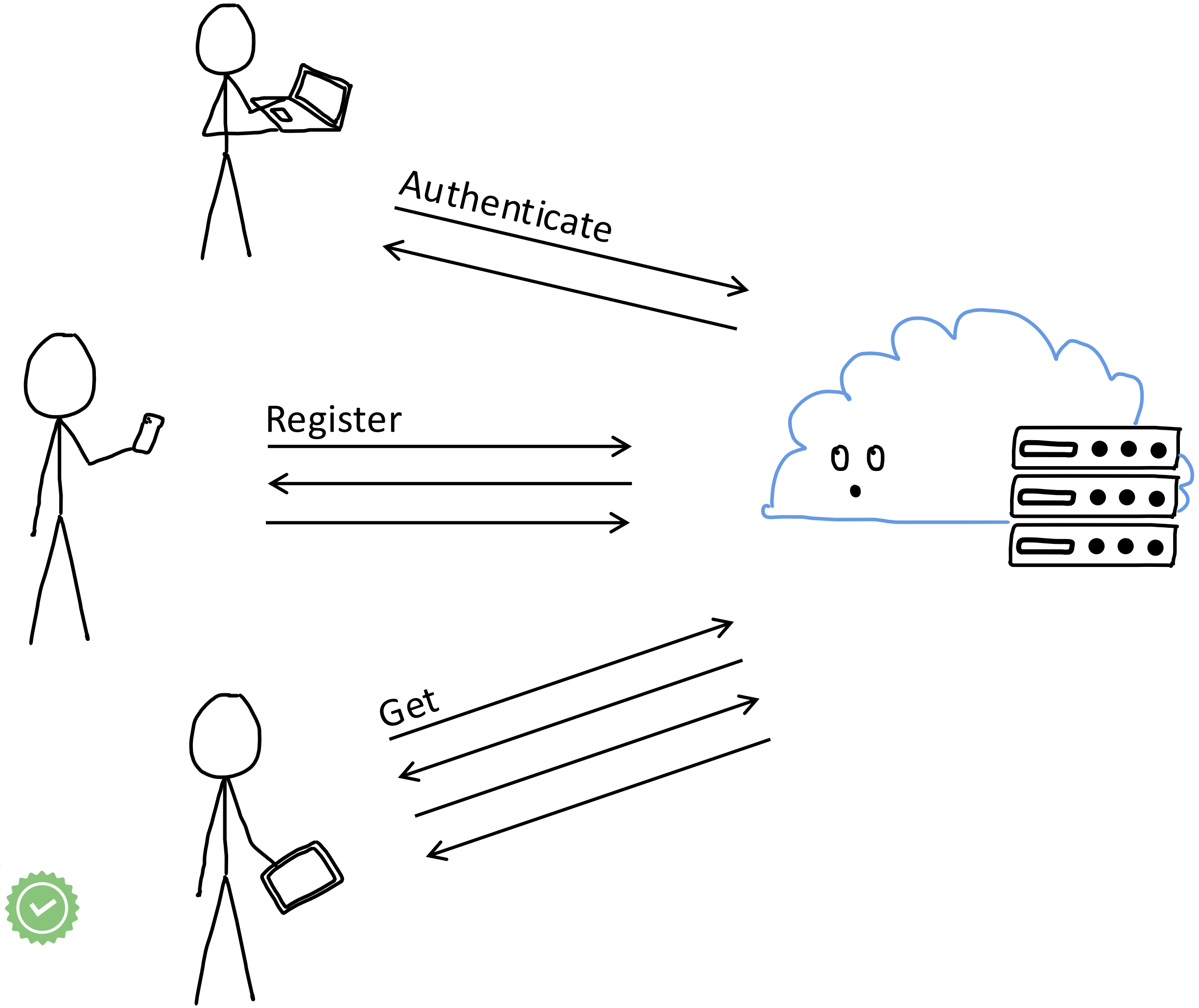
- Put (upload a file)
- Update (modify content)
- Get (download)
- Share
- Accept (receive share)

INTERACTIVE PROTOCOLS

Model Choices

- Arbitrary interleaving

2 FAITHFUL ✓



Syntax

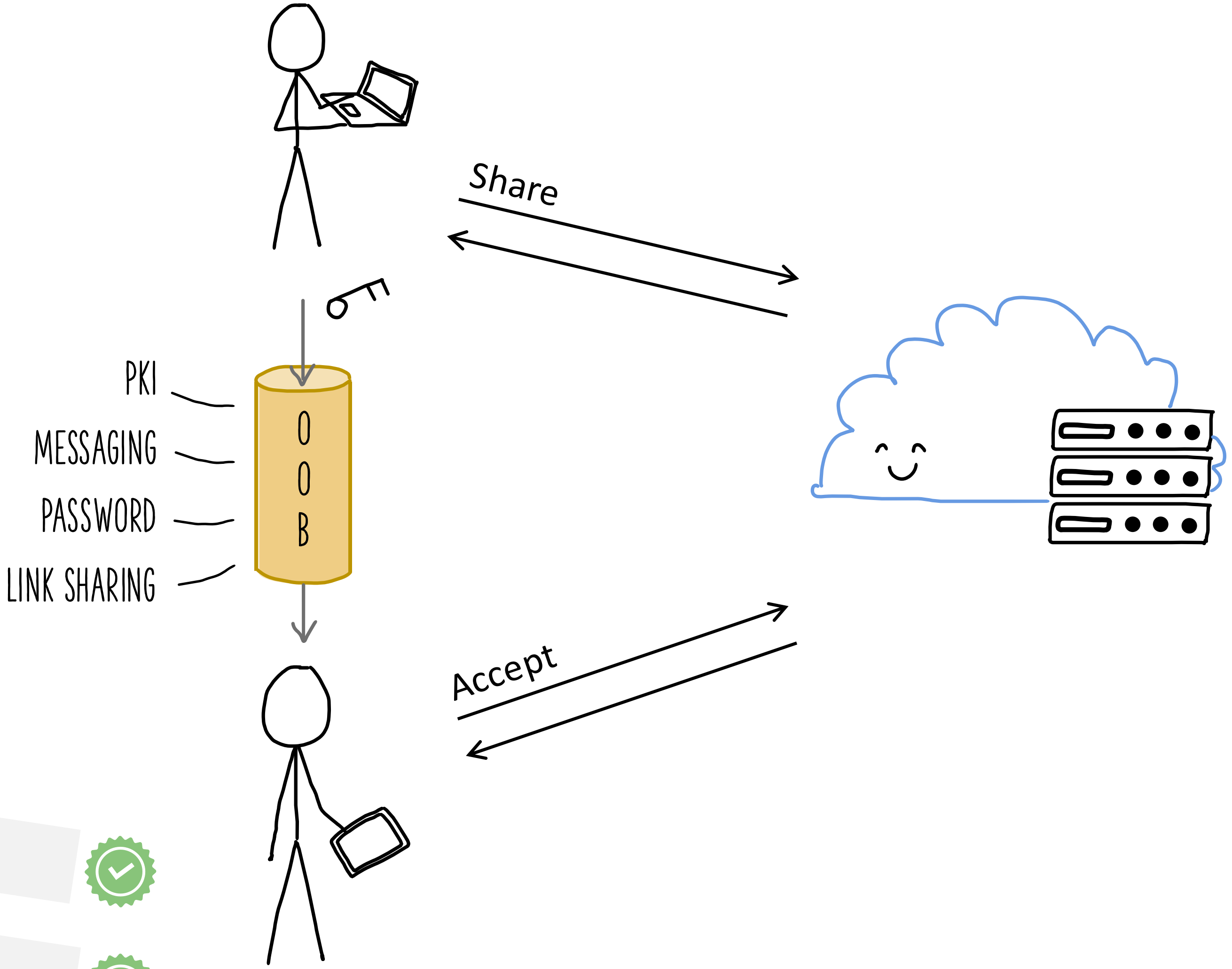
HOW DO WE MAKE THE MODEL USEFUL?

1 EXPRESSIVE ✓

Core Functionality

- Register (create account)
- Authenticate (log in)
- Put (upload a file)
- Update (modify content)
- Get (download)
- Share
- Accept (receive share)

INTERACTIVE PROTOCOLS



Model Choices

- Arbitrary interleaving
- Abstract OOB channel for sharing

2 FAITHFUL ✓

3 GENERIC ✓

Syntax

HOW DO WE MAKE THE MODEL USEFUL?

1 EXPRESSIVE ✓

Core Functionality

- Register (create account)
- Authenticate (log in)

- Put (upload a file)
- Update (modify content)
- Get (download)
- Share
- Accept (receive share)

INTERACTIVE PROTOCOLS

OFTEN NOT CONSIDERED IN RELATED WORK

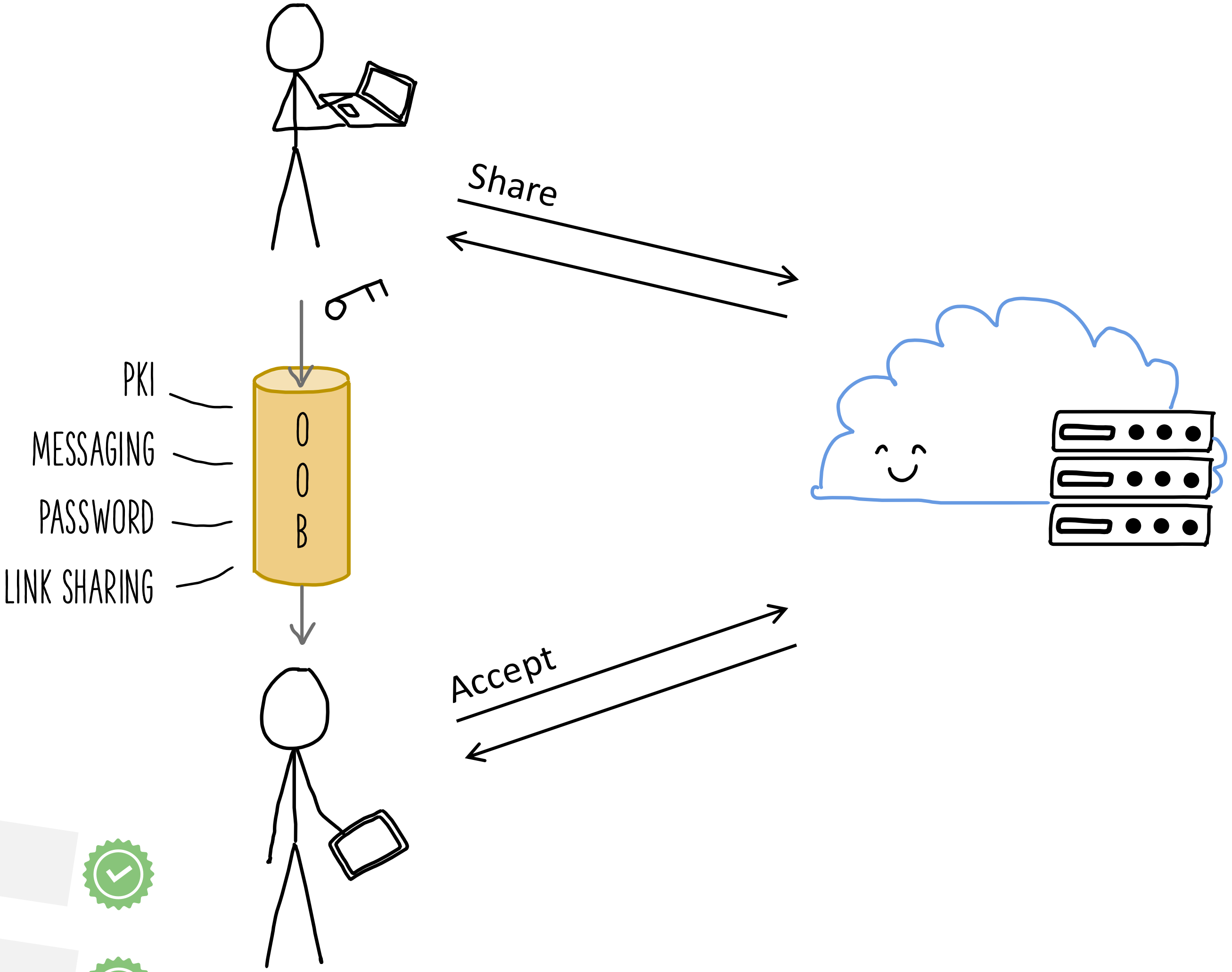
Model Choices

- Arbitrary interleaving

- Abstract OOB channel for sharing

2 FAITHFUL ✓

3 GENERIC ✓



Security Notions

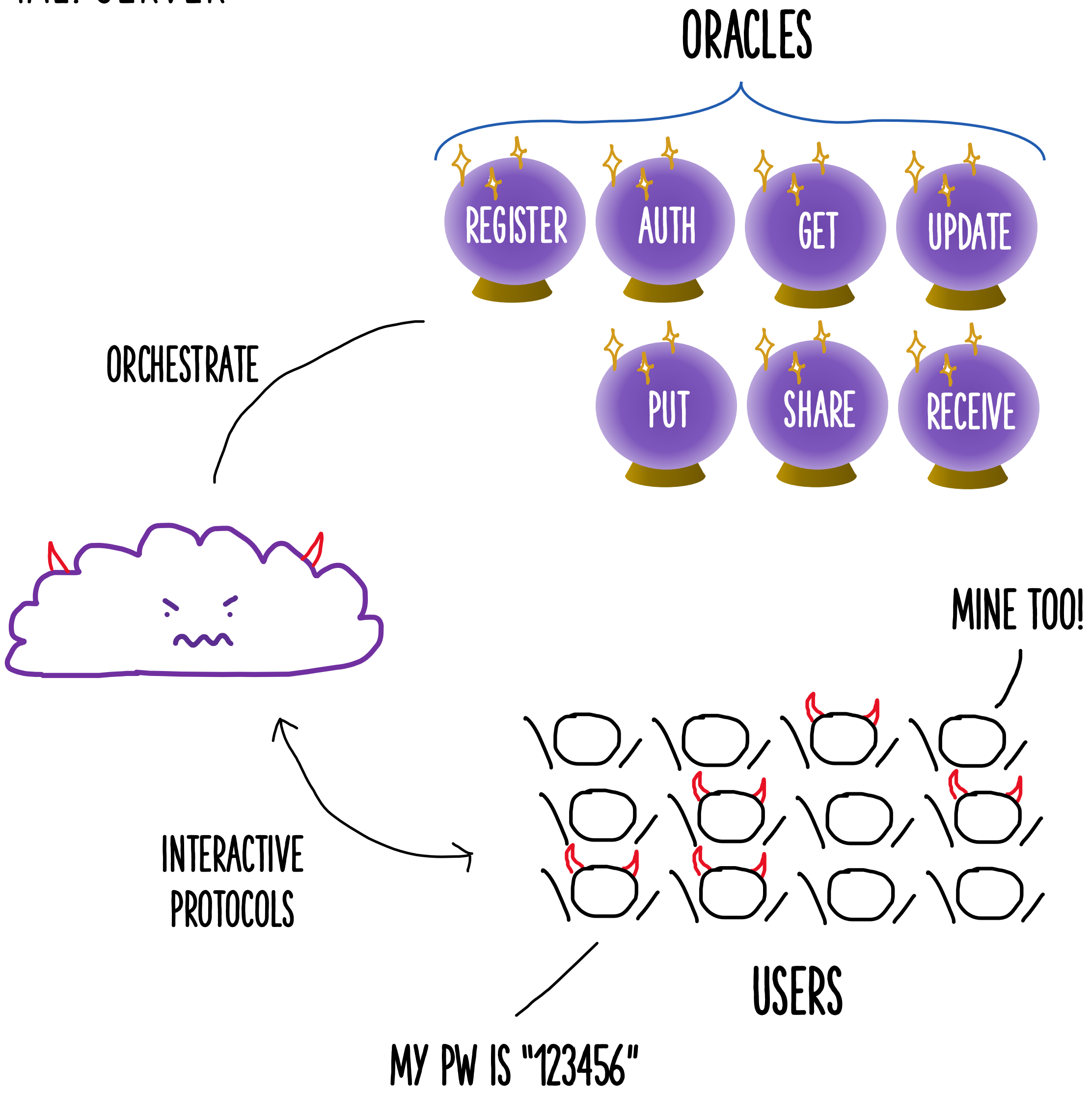
CLIENT-TO-CLIENT (C2C): MAL. SERVER

Threat model:

- Malicious cloud provider
- Full control over network & operations

Game mechanics:

- Correlated passwords
- Adversary can
  - Compromise users (adaptive/selective)
  - Control users (via oracles)
  - Control server (directly)

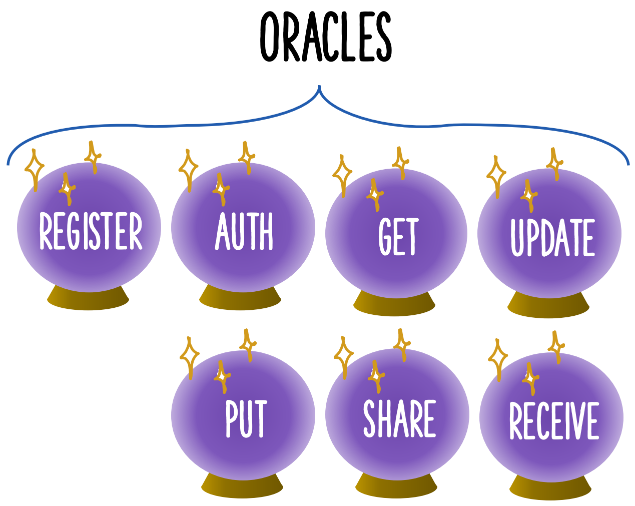


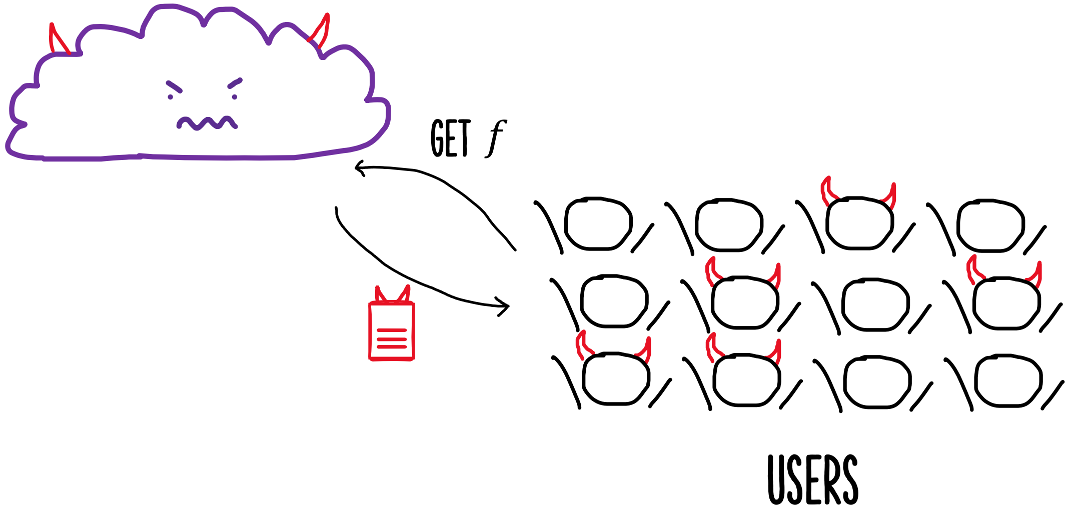
Security Notions

CLIENT-TO-CLIENT (C2C): MAL. SERVER

Integrity:

- Adversary simulates interaction
- Wins if it can, for an honest user,
  1. inject a file, or
  2. modify a file.





Security Notions

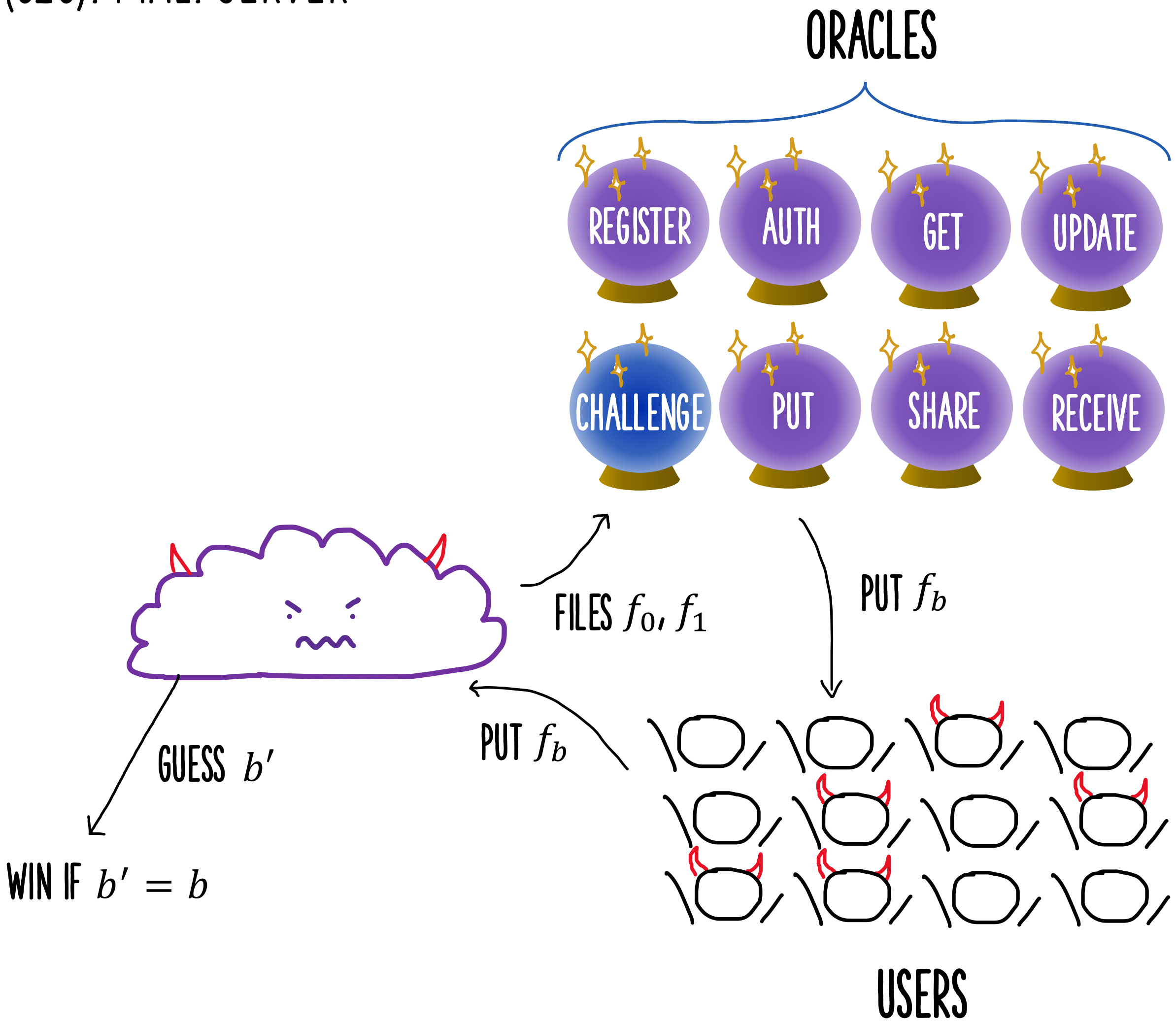
CLIENT-TO-CLIENT (C2C): MAL. SERVER

Integrity:

- Adversary simulates interaction
- Wins if it can, for an honest user,
  1. inject a file, or
  2. modify a file.

Confidentiality:

- Additional challenge oracle
  - Submit two files f_0, f_1
  - File f_b is uploaded
  - Guess bit b



Security Notions: Considerations

Integrity:

- Adversary simulates interaction
- Wins if it can, for an honest user,
  1. inject a file, or
  2. modify a file.

NOT INT-CTXT

Confidentiality:

- Additional challenge oracle
  - Submit two files f_0, f_1
  - File f_b is uploaded
  - Guess bit b

NOT IND-CCA

1 No generic ciphertexts

↳ ALLOWS GENERIC SYNTAX

2 Adaptive & selective compromises

↳ AVOIDS COMMITMENT ISSUES

3 UC vs. game-based notions

↳ UC SECURE CHANNEL TECHNIQUES DO NOT APPLY

Security Notions

CLIENT-TO-SERVER (C2S): MAL. CLIENT [ONGOING WORK]

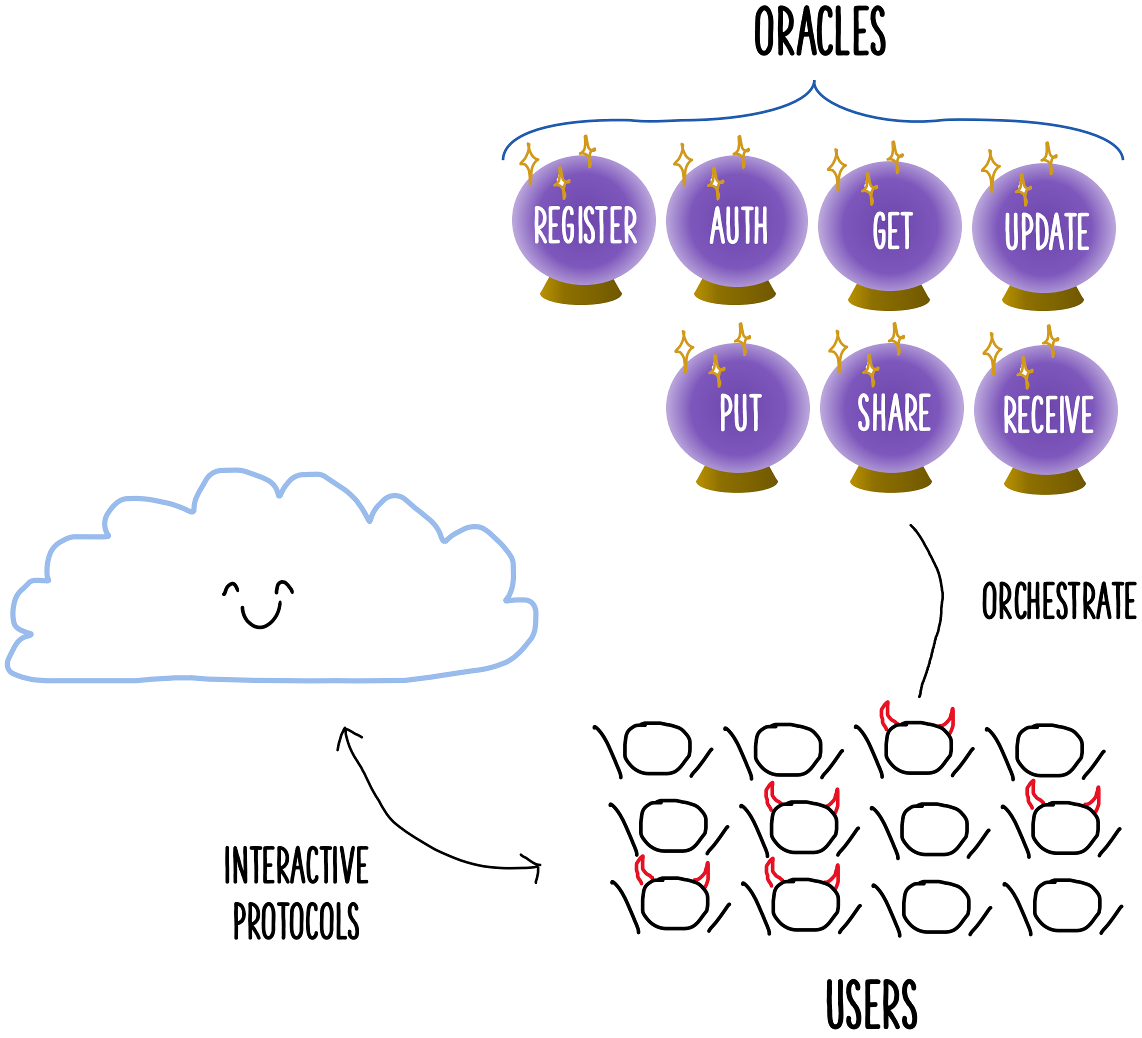
Threat model:

- Honest server
- Malicious clients
- Adversary controls honest user operations

INFEASIBLE IN C2C!

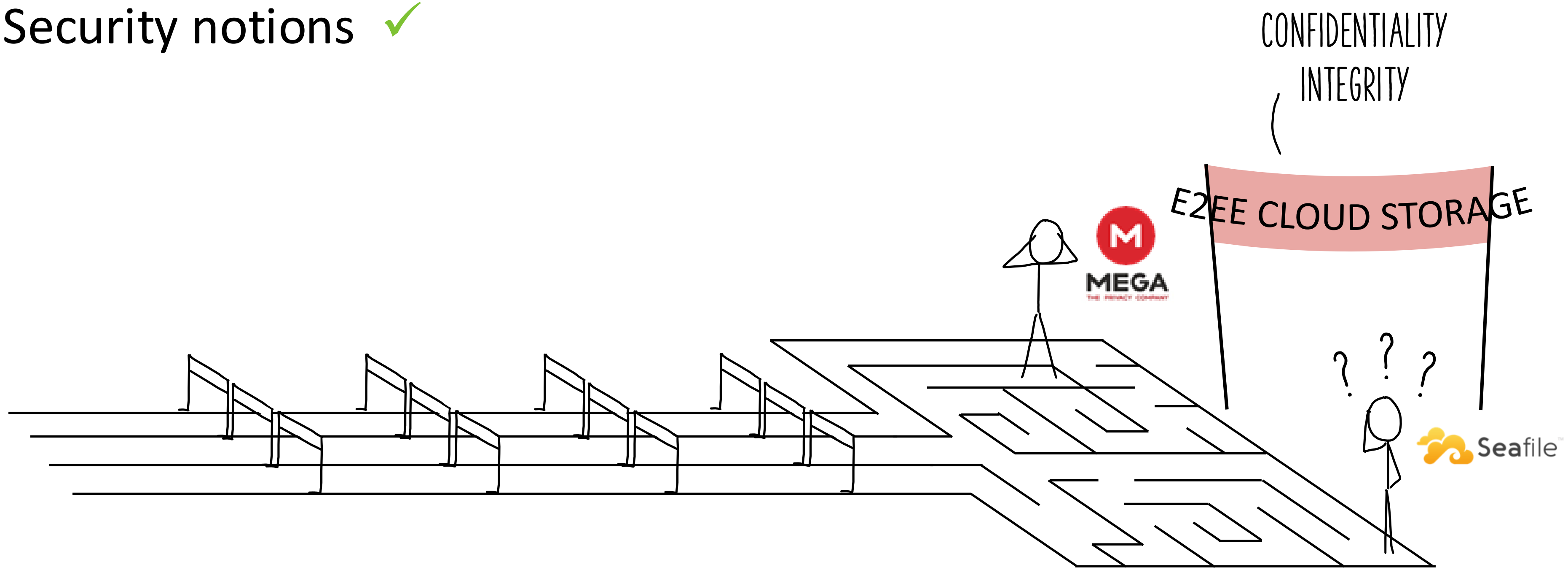
Additional goals:

- Authentication & authorization
- No offline dictionary attacks on pw
- Availability for honest user files



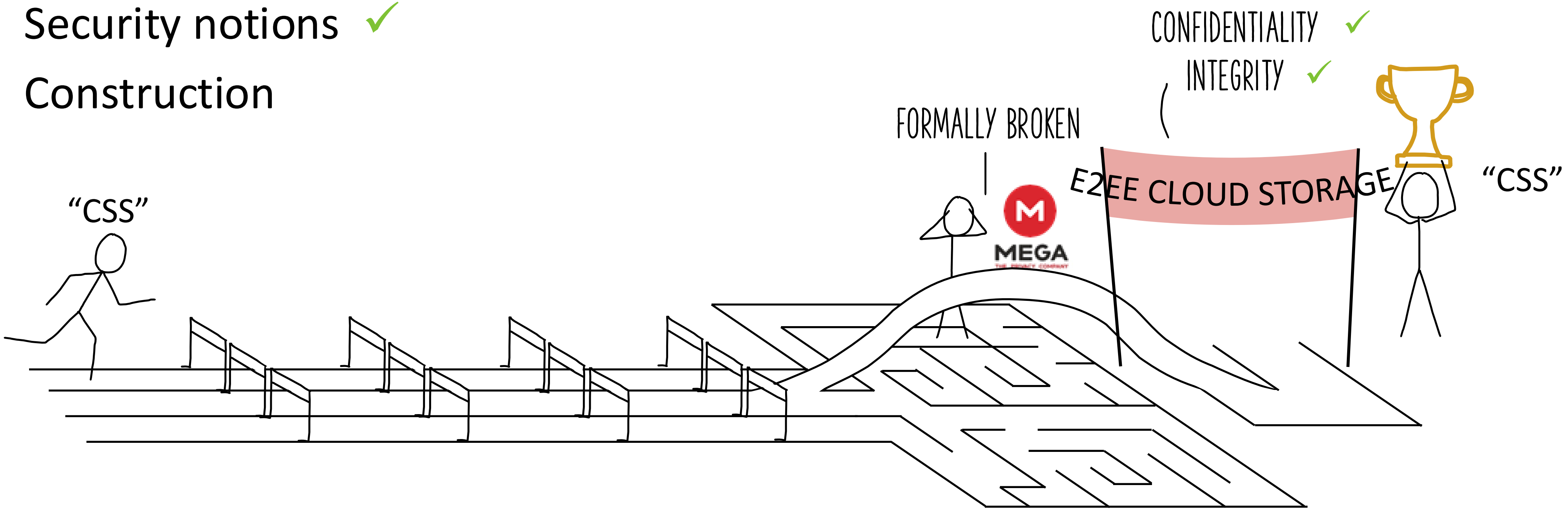
Are We Done?

- Syntax ✓
- Security notions ✓



Are We Done?

- Syntax ✓
- Security notions ✓
- Construction



2. Constructing E2EE Cloud Storage

CSS (Cloud Storage Scheme)

Building Blocks

OPRF

PRF

AEAD

MAC

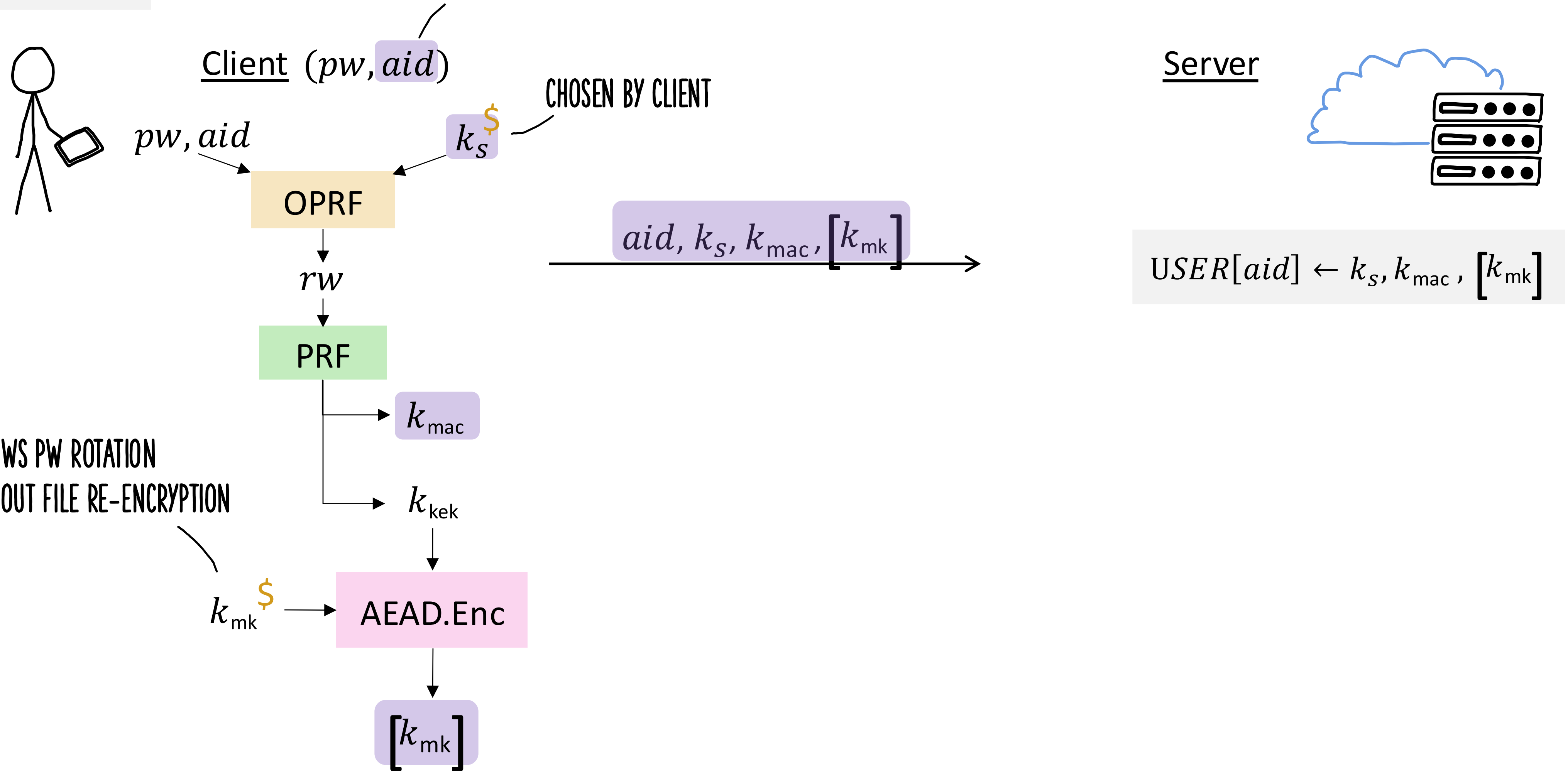
2HashDH
HANDLES PWS => ROM

ONLY NEEDED FOR
C2S SECURITY

CSS (Cloud Storage Scheme)

*SIMPLIFIED

Registration



GLOBALLY UNIQUE

Client (pw, aid)

CHOSEN BY CLIENT

pw, aid

$k_s \overset{\$}{}$

OPRF

rw

PRF

k_{mac}

k_{kek}

ALLOWS PW ROTATION
WITHOUT FILE RE-ENCRYPTION

$k_{\text{mk}} \overset{\$}{}$

AEAD.Enc

$[k_{\text{mk}}]$

$aid, k_s, k_{\text{mac}}, [k_{\text{mk}}]$

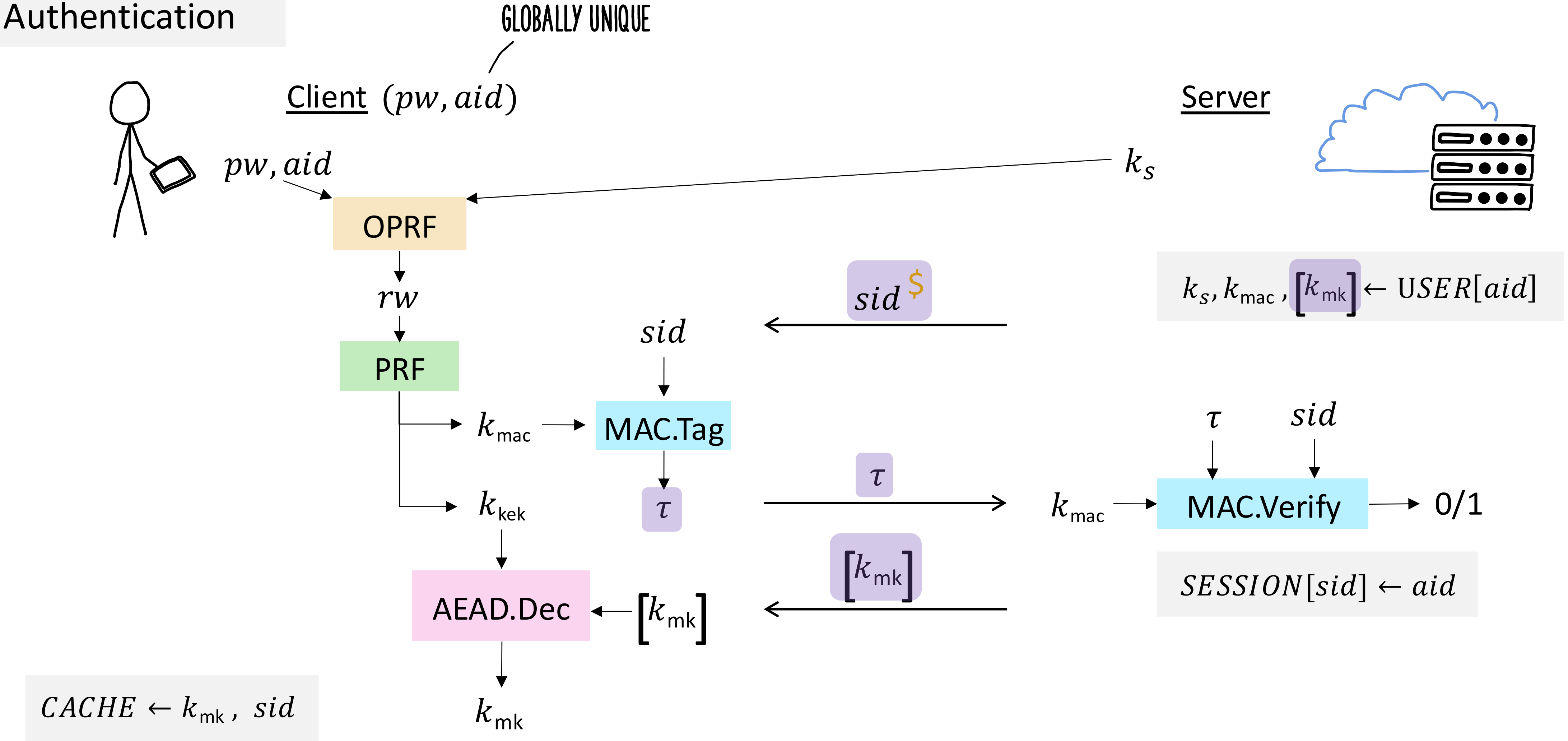
Server

$USER[aid] \leftarrow k_s, k_{\text{mac}}, [k_{\text{mk}}]$

CSS (Cloud Storage Scheme)

*SIMPLIFIED

Authentication



$k_s, k_{\text{mac}}, [k_{\text{mk}}] \leftarrow USER[aid]$

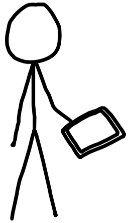
$SESSION[sid] \leftarrow aid$

$CACHE \leftarrow k_{\text{mk}},\ sid$

CSS (Cloud Storage Scheme)

*SIMPLIFIED

Put

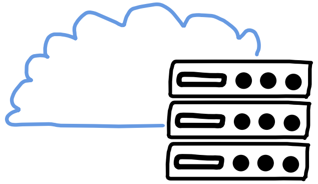


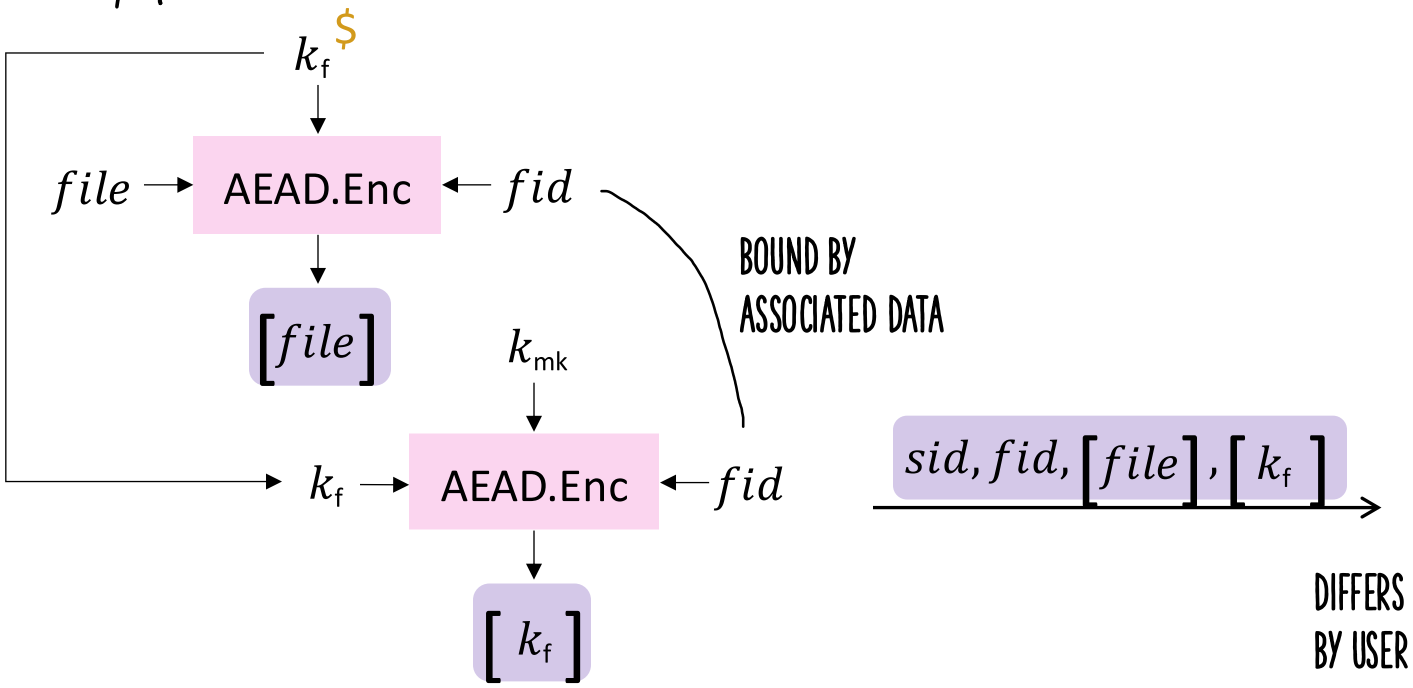
GLOBALLY UNIQUE

Client $(file, fid)$

$k_{mk}, sid \leftarrow CACHE$

Server





BOUND BY ASSOCIATED DATA

DIFFERS BY USER

$aid \leftarrow SESSION[sid]$

$FILE[fid] \leftarrow [file]$

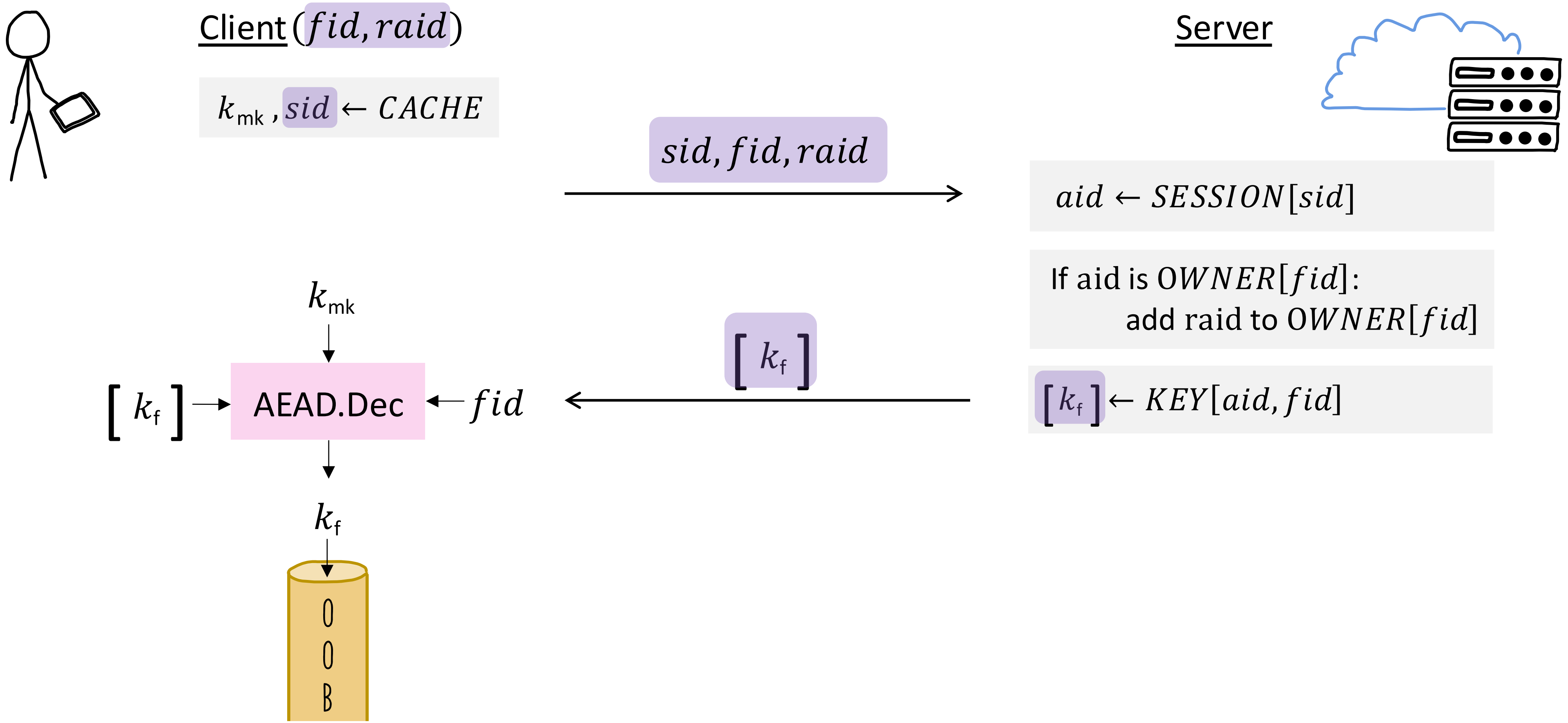
$OWNER[fid] \leftarrow \{aid\}$

$KEY[aid, fid] \leftarrow [k_f]$

CSS (Cloud Storage Scheme)

*SIMPLIFIED

Share



Client $(fid, raid)$

$k_{\text{mk}}, sid \leftarrow CACHE$

$sid, fid, raid$

Server

$aid \leftarrow SESSION[sid]$

If aid is $OWNER[fid]$:
    add raid to $OWNER[fid]$

$[k_{\text{f}}] \leftarrow KEY[aid, fid]$

$[k_{\text{f}}]$

k_{mk}

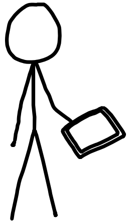
$[k_{\text{f}}]$ → AEAD.Dec ← fid

k_{f}

CSS (Cloud Storage Scheme)

*SIMPLIFIED

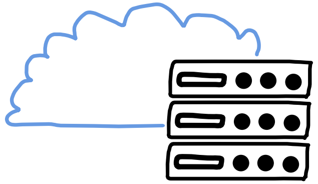
Accept

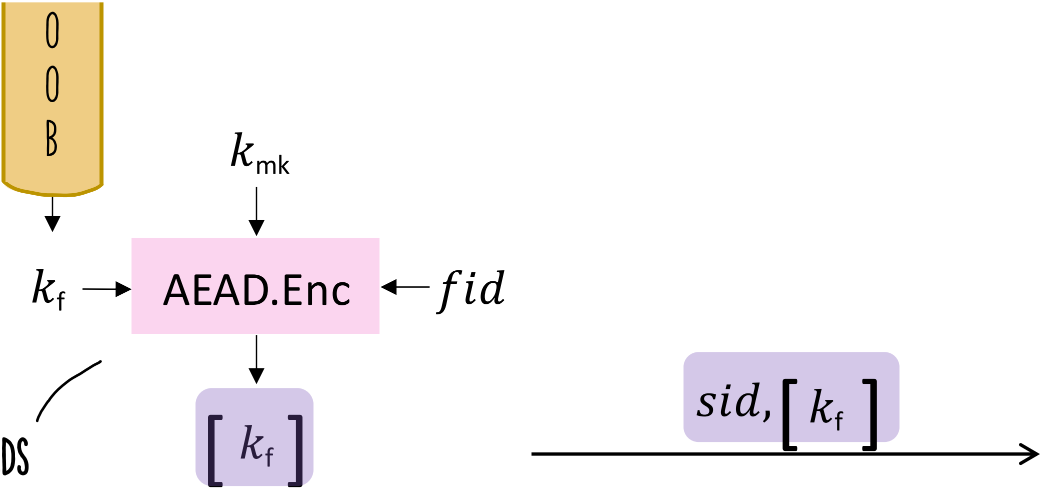


Client(fid)

$k_{\text{mk}}, sid \leftarrow CACHE$

Server





RE-ENCRYPTION AVOIDS PUBLIC KEY OPERATIONS

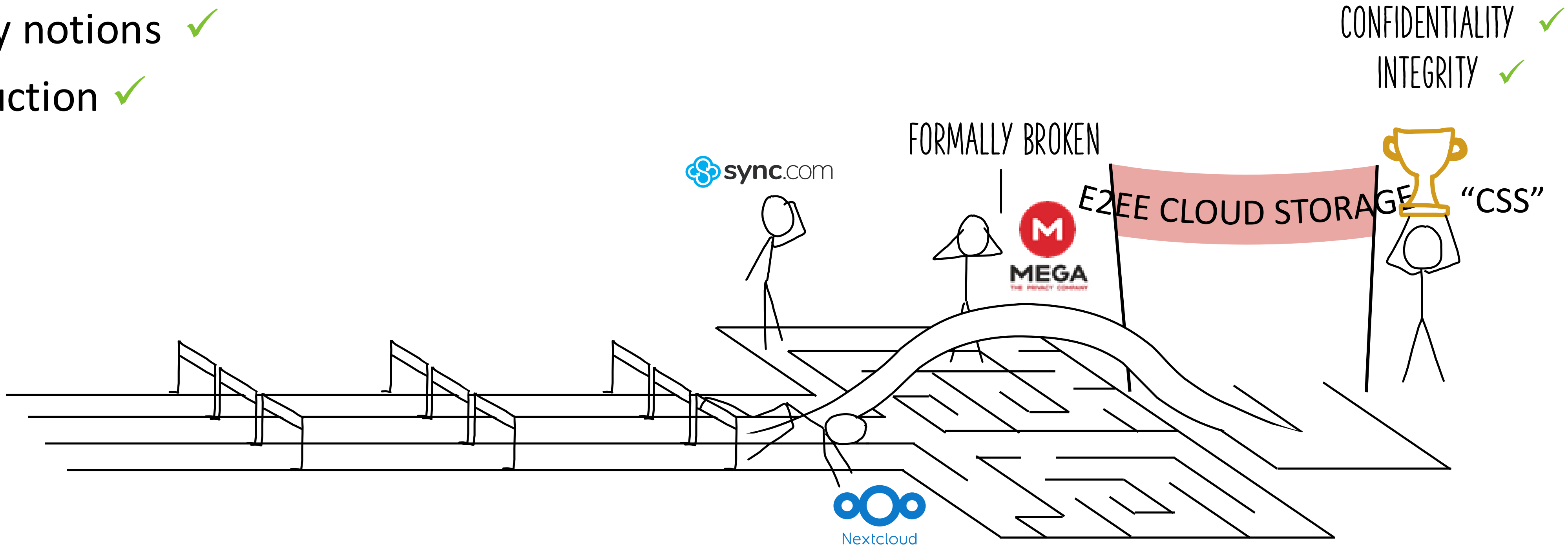
$aid \leftarrow SESSION[sid]$

If aid is $OWNER[fid]$:

$KEY[aid, fid] \leftarrow [k_{\text{f}}]$

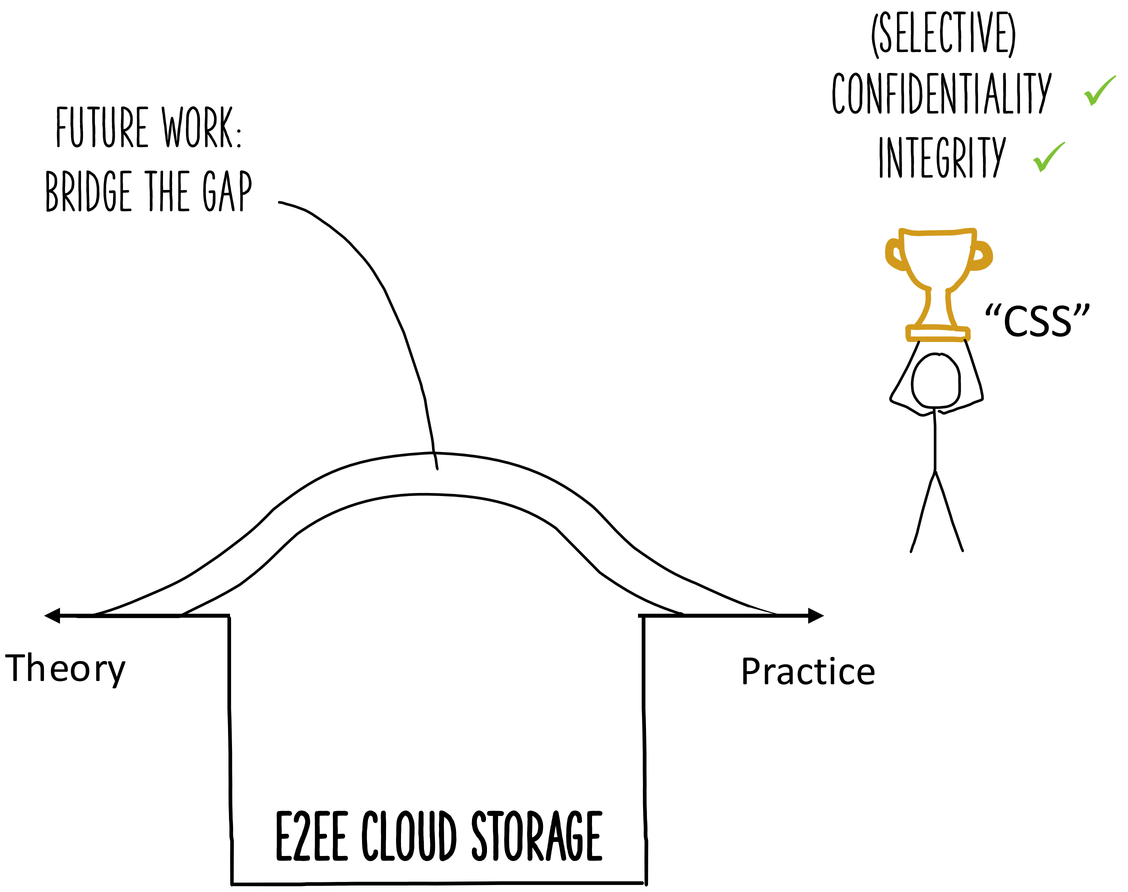
Are We Done?

- Syntax ✓
- Security notions ✓
- Construction ✓



Are We Done?

- Syntax ✓
- Security notions ✓
- Construction ✓

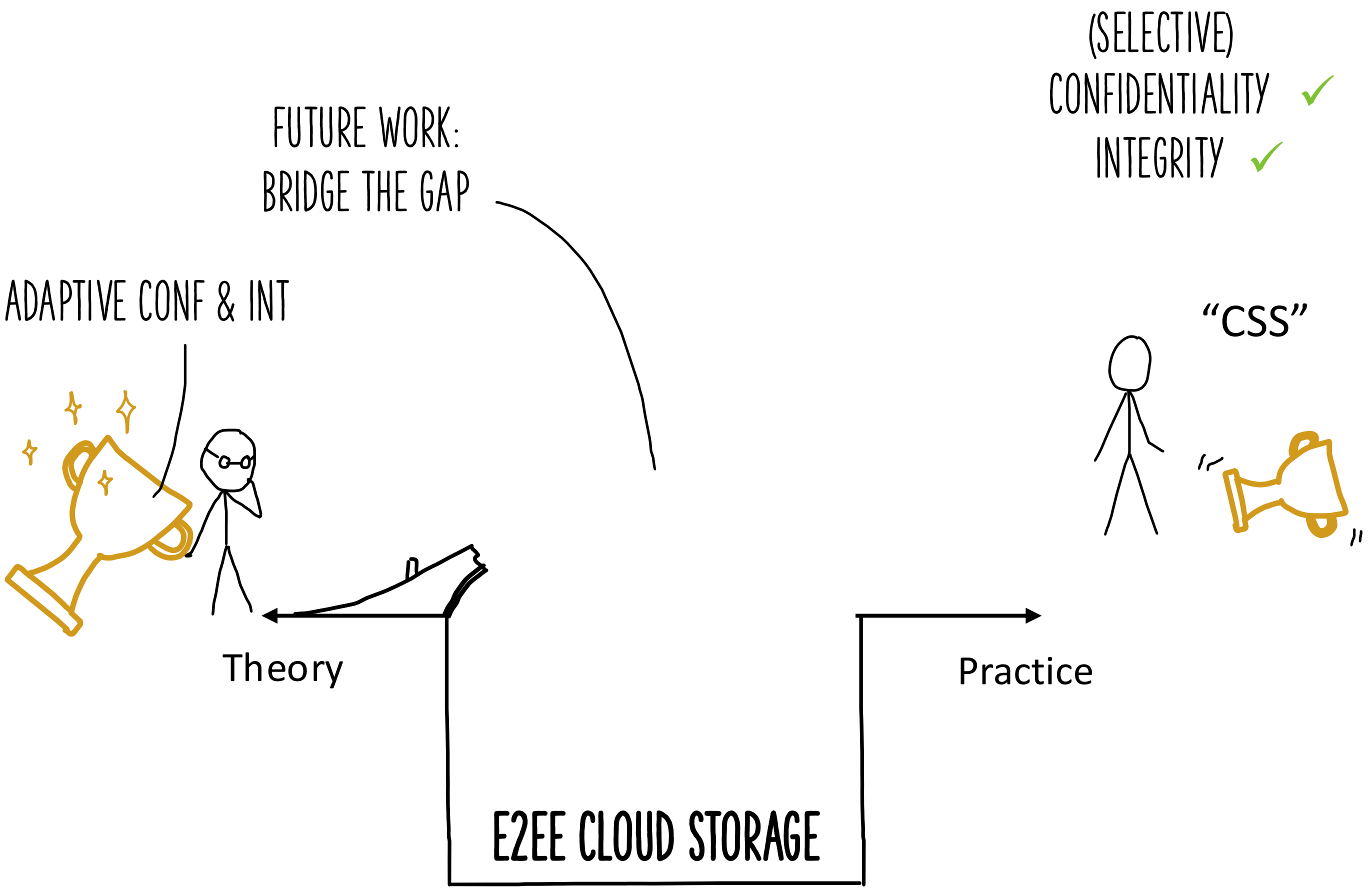


Are We Done?

- Syntax ✓
- Security notions ✓
- Construction ✓

Still missing:

- Adaptive security proof

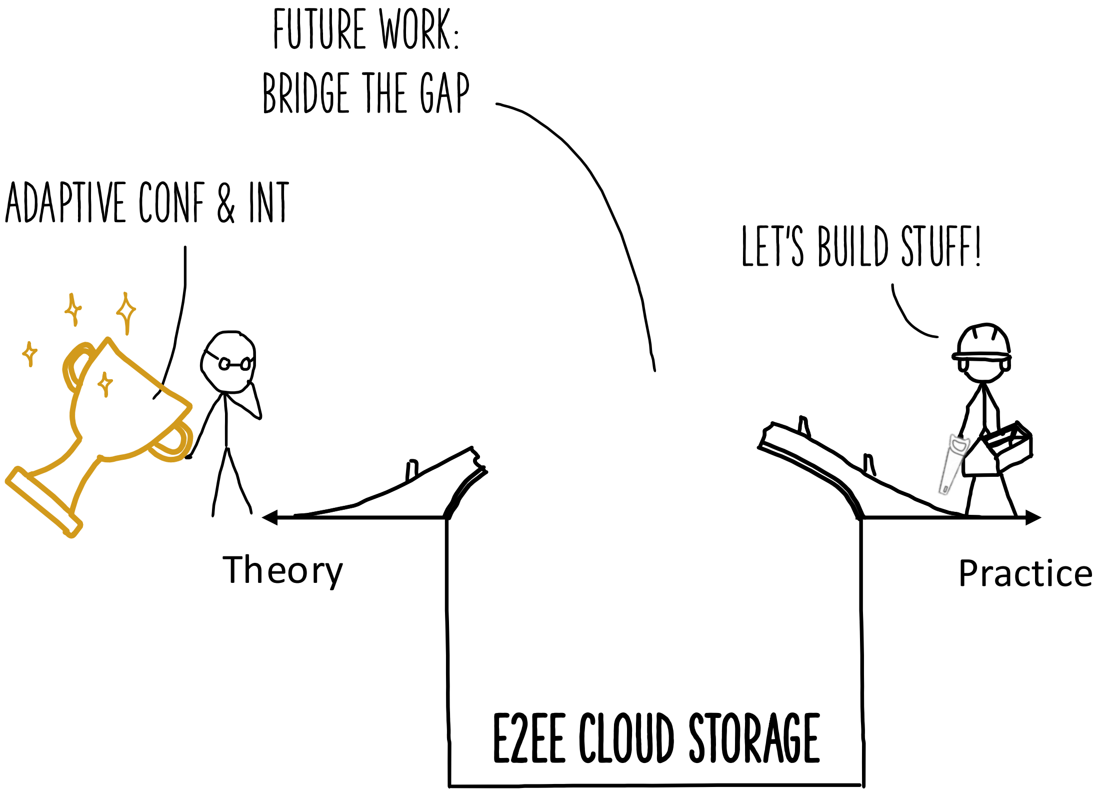


Are We Done?

- Syntax ✓
- Security notions ✓
- Construction ✓

Still missing:

- Adaptive security proof
- Implementation
- Feedback, model extensions, …



A Formal Treatment of End-to-End Encrypted Cloud Storage

Matilda Backendal, Hannah Davis, Felix Günther, Miro Haller, Kenny Paterson

mbackendal@inf.ethz.ch mhaller@ucsd.edu

eprint.iacr.org/2024/989