



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

---

Matilda Backendal<sup>1</sup>, Hannah Davis<sup>2</sup>, Felix Günther<sup>3</sup>, Miro Haller<sup>4</sup>, Kenny Paterson<sup>1</sup>

<sup>1</sup>ETH Zurich, <sup>2</sup>Seagate Technology, <sup>3</sup>IBM Research Zurich, <sup>4</sup>UC San Diego

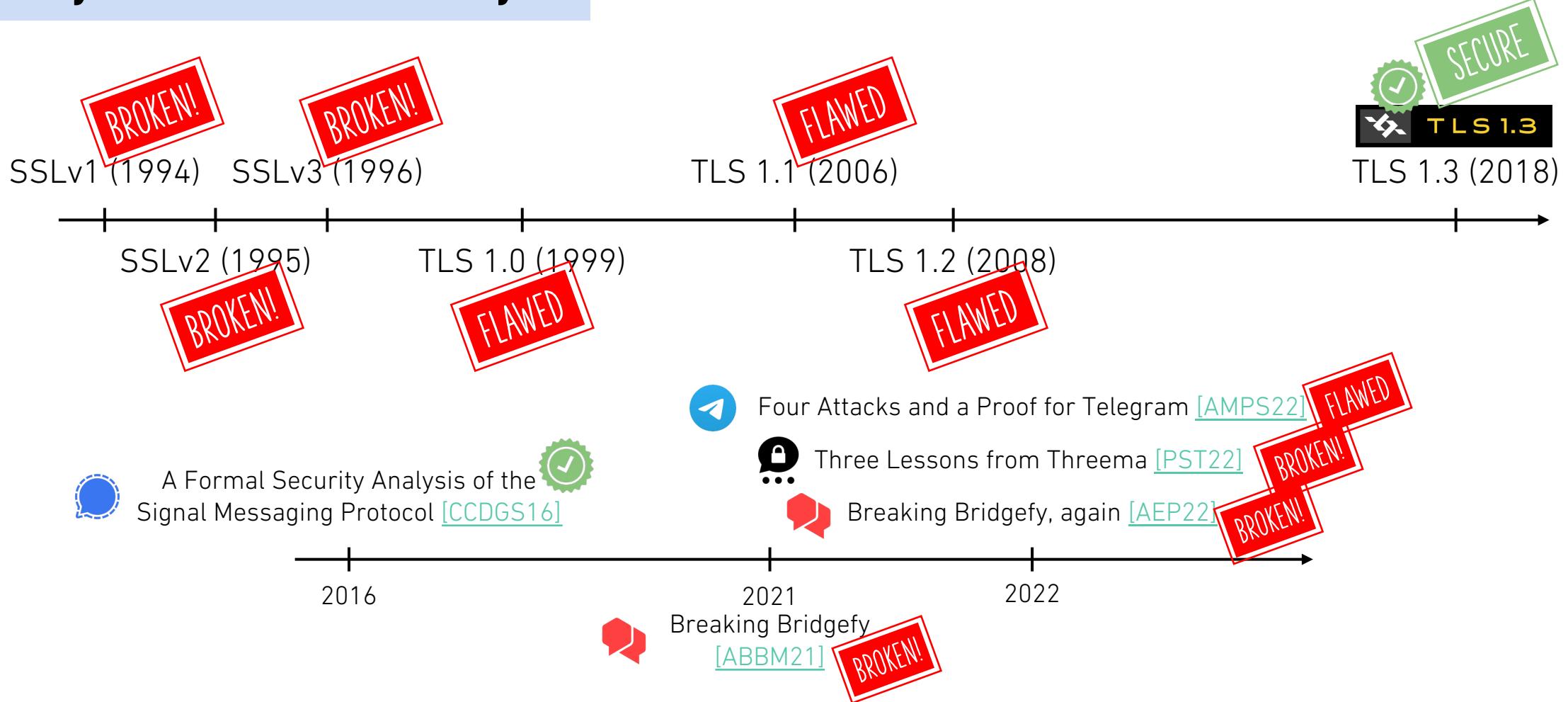


**Privacy is a fundamental human right. It's also one of our core values. Which is why we design our products and services to protect it. That's the kind of innovation we believe in.**

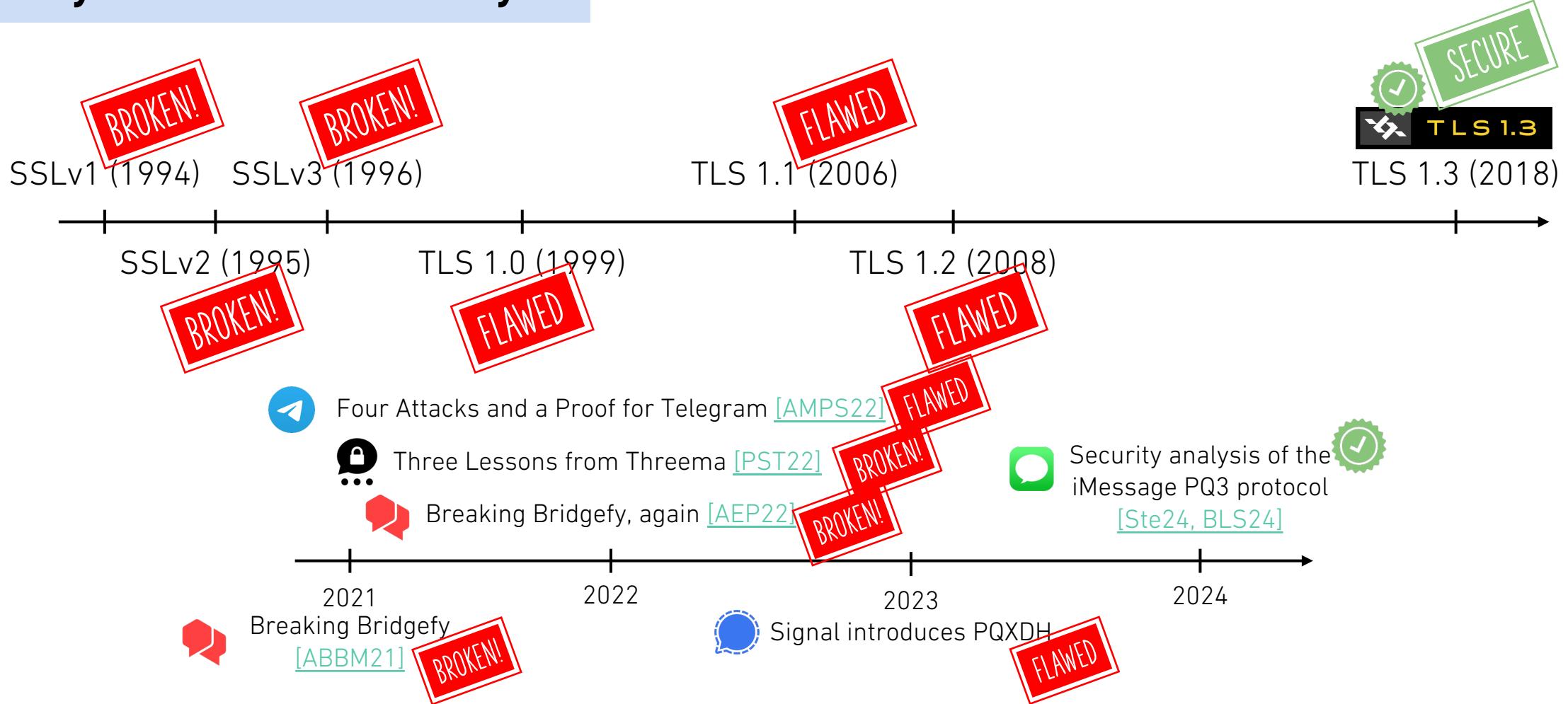
Screenshot from <https://www.apple.com/privacy/>

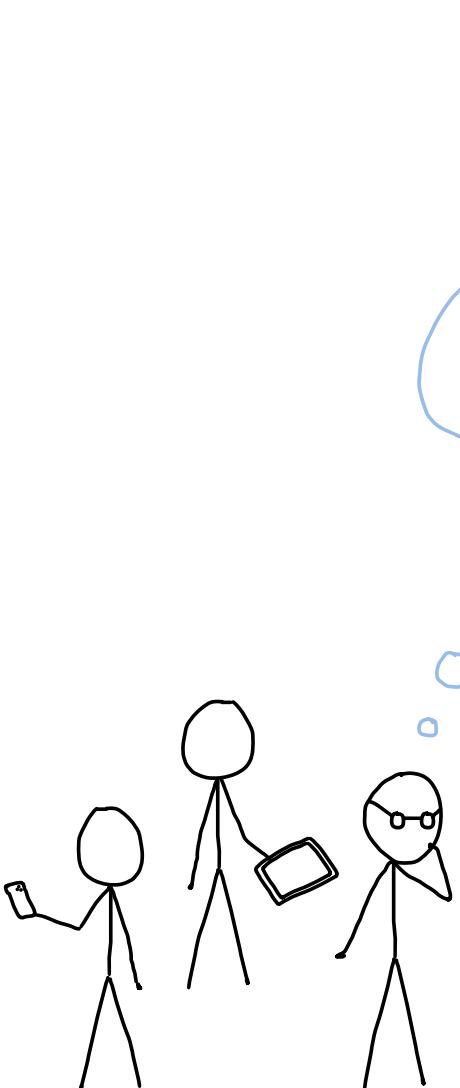
# Why E2E Security?

# Why Provable Security?



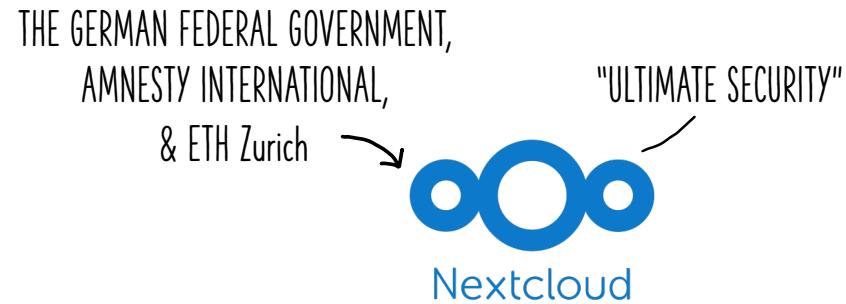
# Why Provable Security?





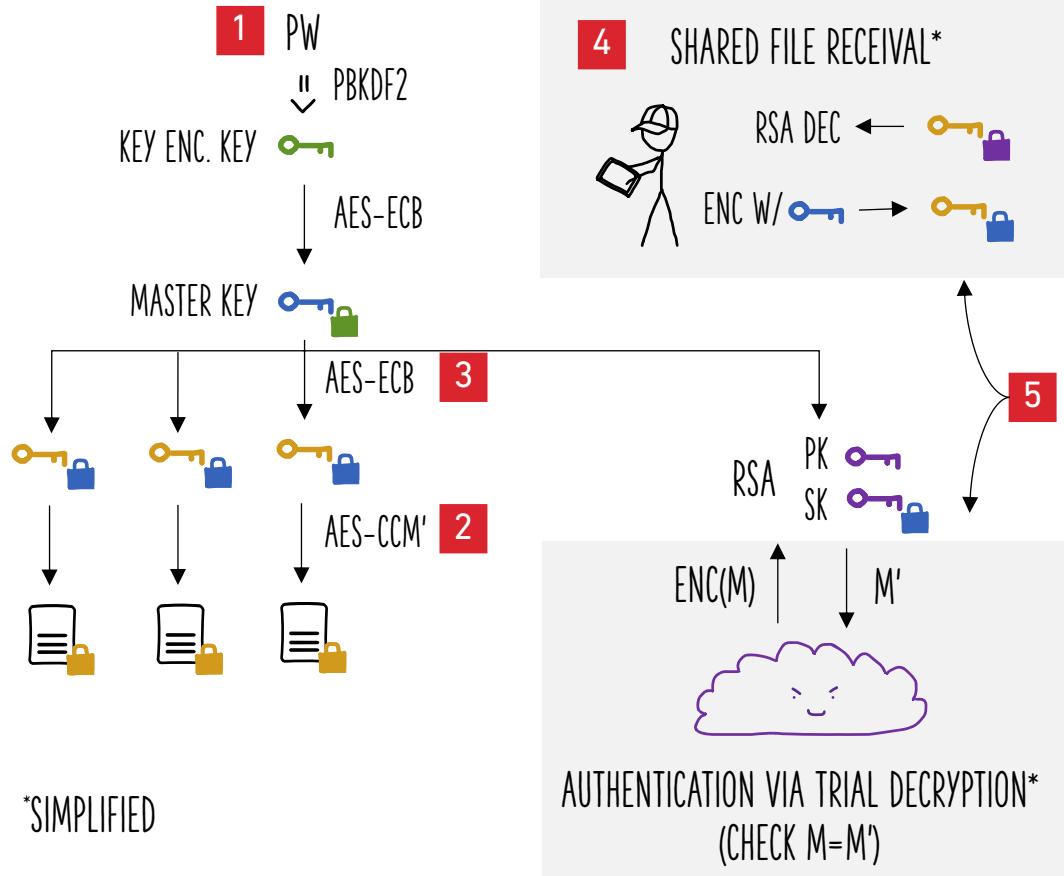
# Is iCloud with Advanced Data Protection E2E Secure?

# E2EE Cloud Storage Providers



# MEGA Case Study

## MEGA's key hierarchy\*



## MEGA's challenges

ASIDE: GETTING AWAY FROM PW REQUIRES ADDITIONAL ASSUMPTIONS (E.G., TRUSTED KEY STORAGE ON iPhone)

- 1 Stateless clients → SECURITY DEPENDS ON PW STRENGTH
- 2 File re-encryption → REPLACING AES-CCM > 180 DAYS
- 3 Ciphertext integrity → ENABLES ATTACKS IN [1, 2]
- 4 File sharing → RSA SECRET KEY DECRYPTION [2]
- 5 Key reuse → FILE KEY DECRYPTION [1]

[1] Matilda Backendal, Miro Haller\* and Kenneth G. Paterson. (2023). "MEGA: Malleable Encryption Goes Awry" IEEE S&P 2023.

[2] Martin R. Albrecht, Miro Haller, Lenka Mareková\*, Kenneth G. Paterson. (2023). "Caveat Implementor! Key Recovery Attacks on MEGA" Eurocrypt 2023.

# E2EE Cloud Storage Providers

"WITH MEGA, YOU  
CONTROL THE ENCRYPTION"



300 MILLION USERS



[SP:BHP23]  
[EC:AHMP23]

AMNESTY INTERNATIONAL,  
THE GERMAN FEDERAL GOVERNMENT

& ETH



"ULTIMATE SECURITY"

[EuroSP:ABCP23]

"FREE, ENCRYPTED, AND SECURE CLOUD STORAGE.  
YOUR PRIVACY, SECURED BY MATH"



NOT PROVABLY SECURE

"EXCEPTIONALLY PRIVATE CLOUD"



"THE STRONGEST ENCRYPTED  
CLOUD STORAGE IN THE WORLD"

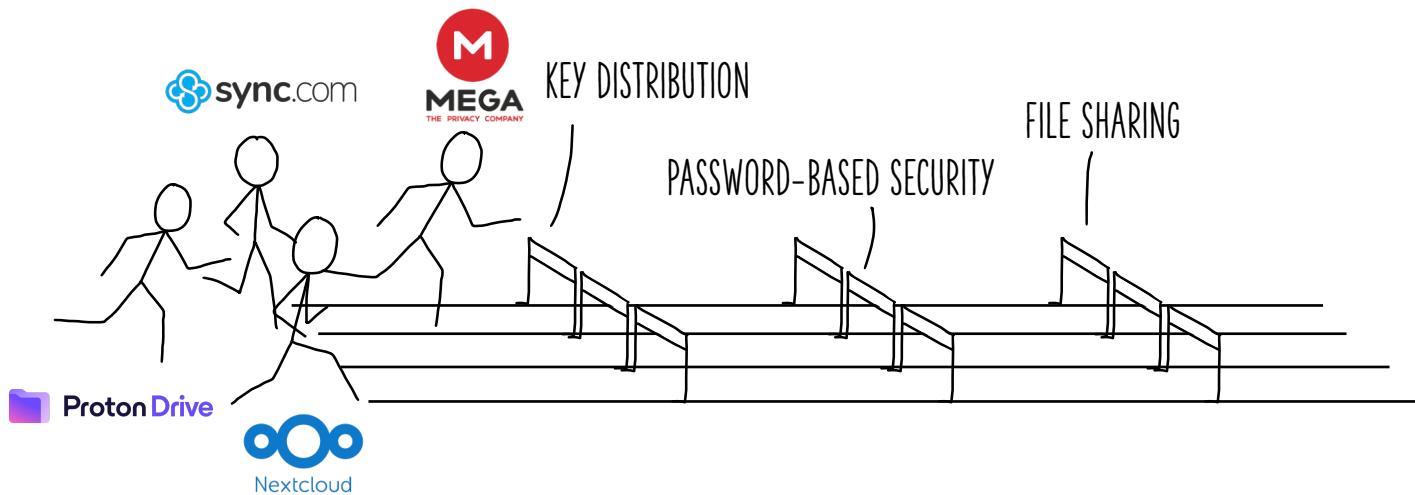


"EUROPE'S MOST SECURE CLOUD STORAGE"

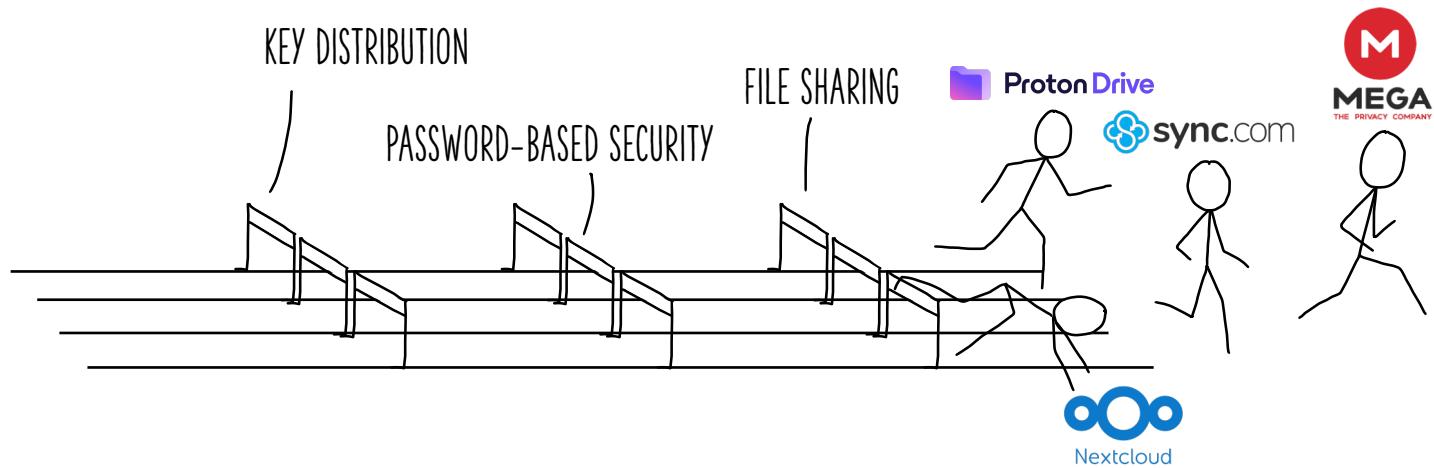


"SUPPORTS CLIENT-SIDE  
END-TO-END ENCRYPTION"

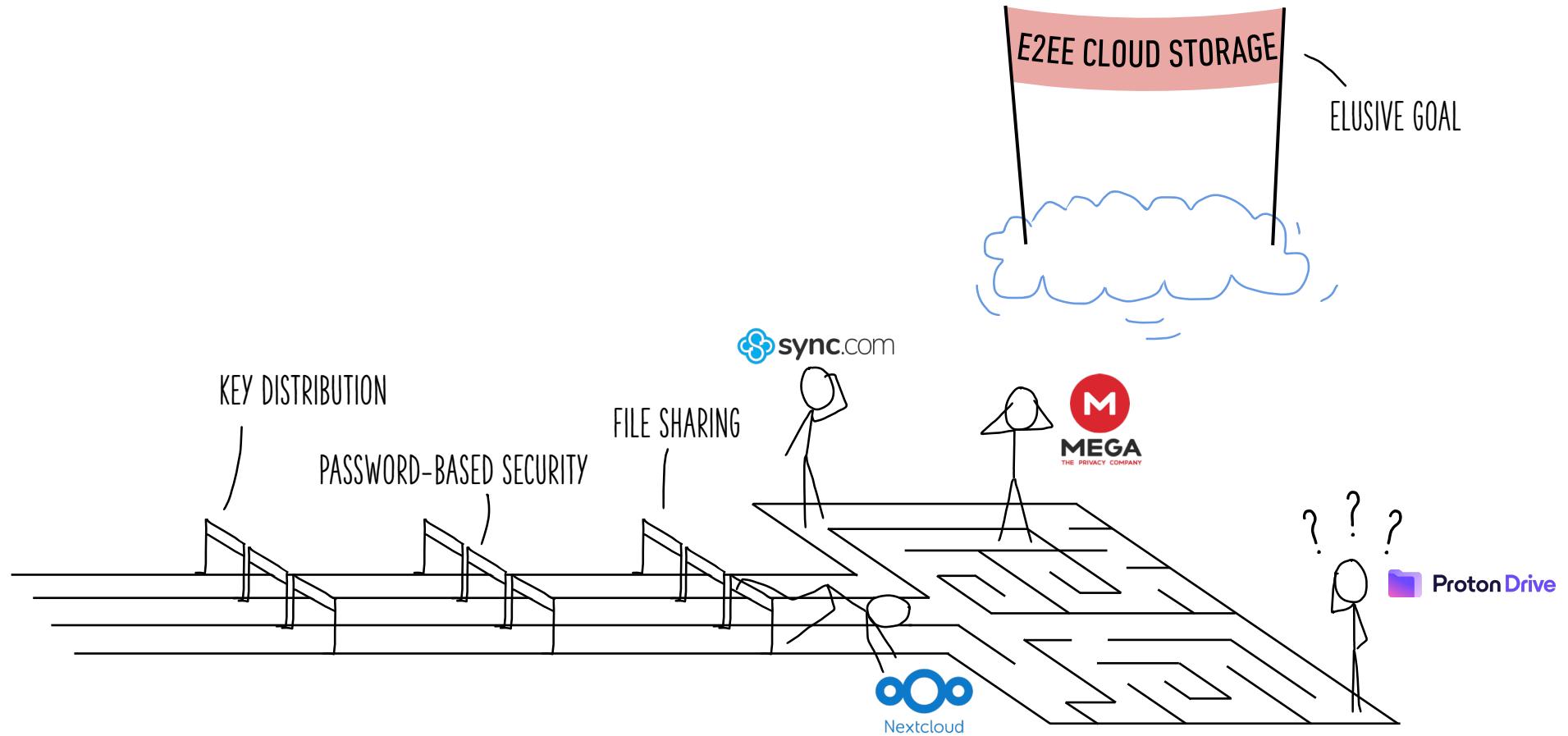
# Why Is It Hard?



# Why Is It Hard?



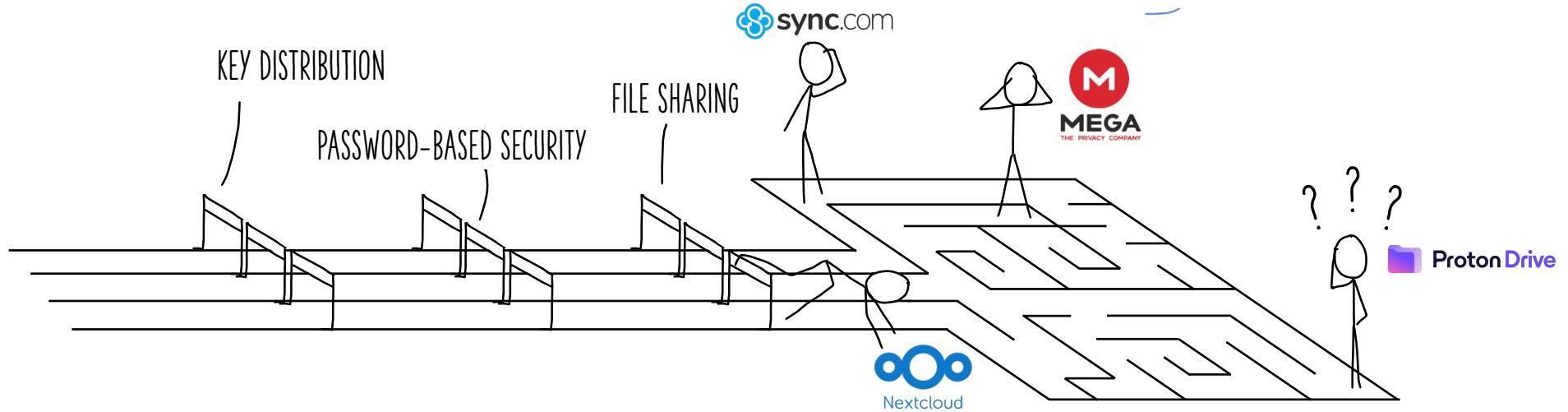
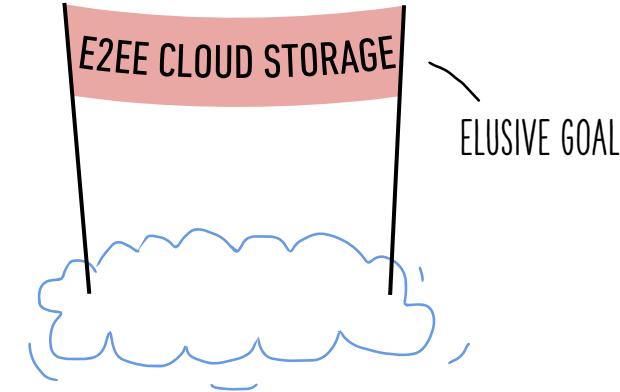
# Why Is It Hard?



# Our Work

## Formal Model for E2EE Cloud Storage

- Core functionality  
→ Syntax & correctness
- Security notions
- Provably secure protocol



# 1. Formalizing E2EE Cloud Storage



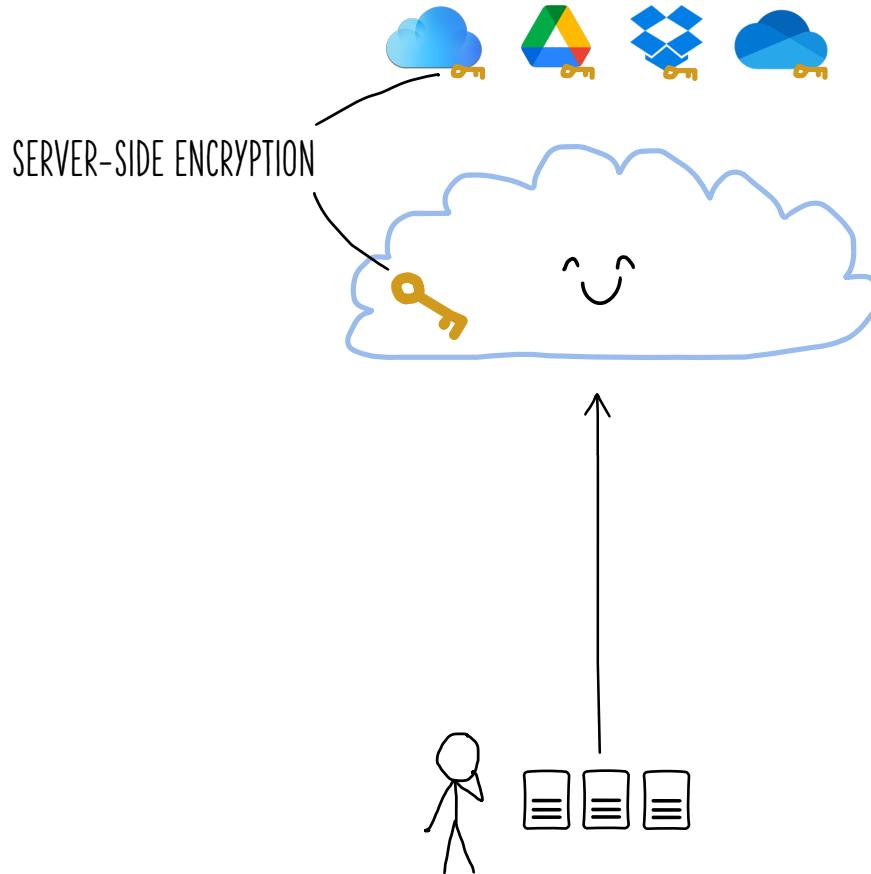
# Formalizing E2EE Cloud Storage

## Goal:

- Secure data at rest
- ...with maximal functionality

## Methods:

- Server-side encryption
  - + Plaintext access -> features
  - Plaintext access -> less privacy



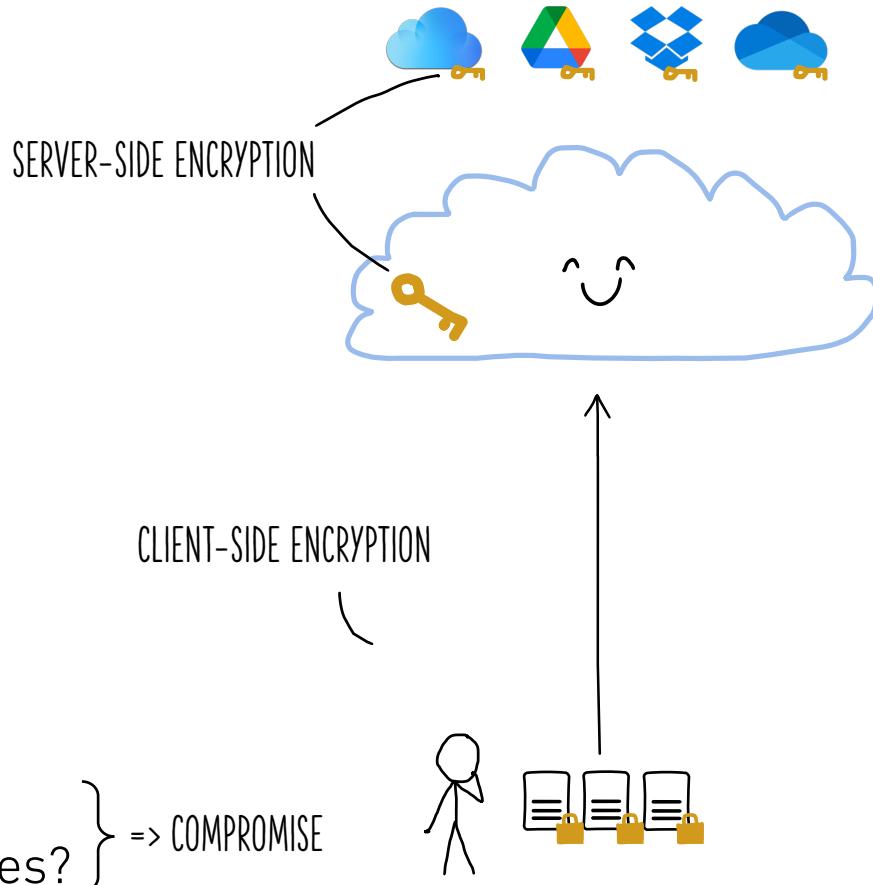
# Formalizing E2EE Cloud Storage

## Goal:

- Secure data at rest
- ...with maximal functionality
- ...and strong privacy

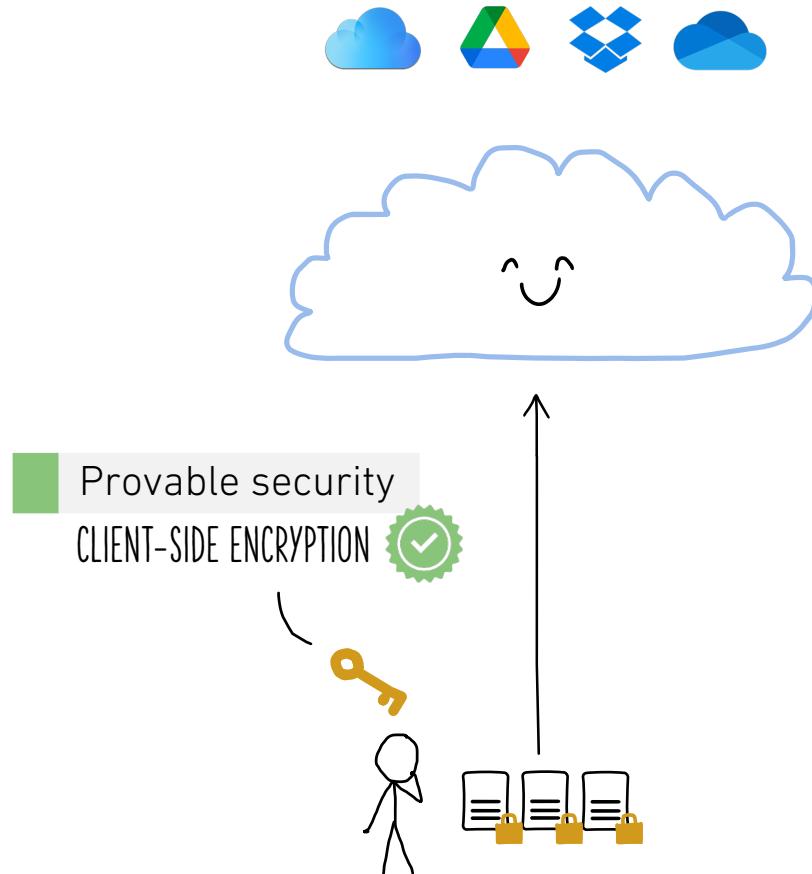
## Methods:

- Server-side encryption
  - + Plaintext access -> features
  - Plaintext access -> less privacy
- End-to-end encryption
  - + No plaintext access -> privacy
  - No plaintext access -> less features?



# Formalizing E2EE Cloud Storage

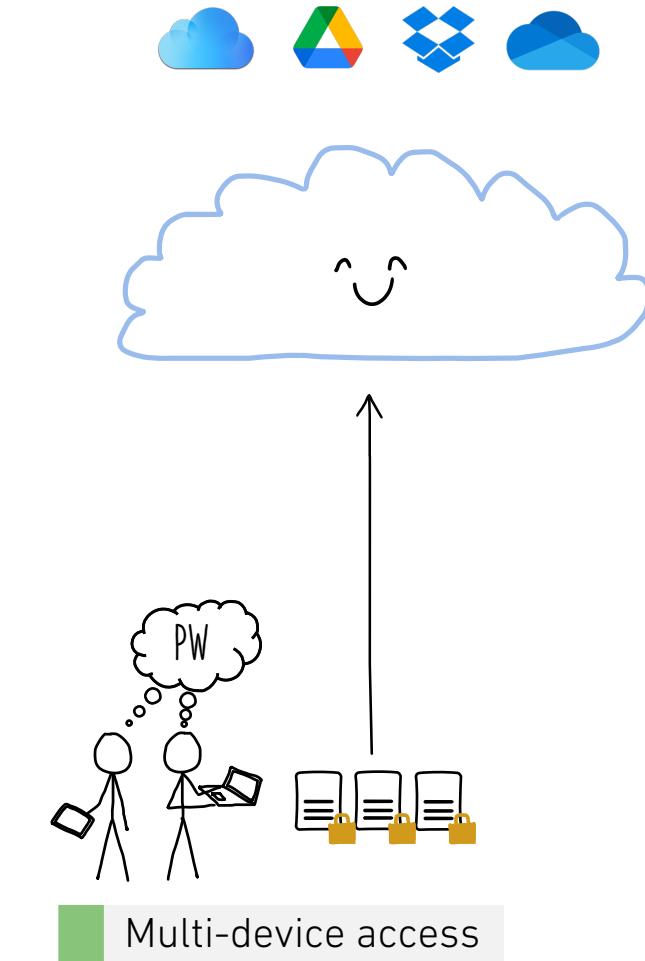
In scope:



# Formalizing E2EE Cloud Storage

In scope:

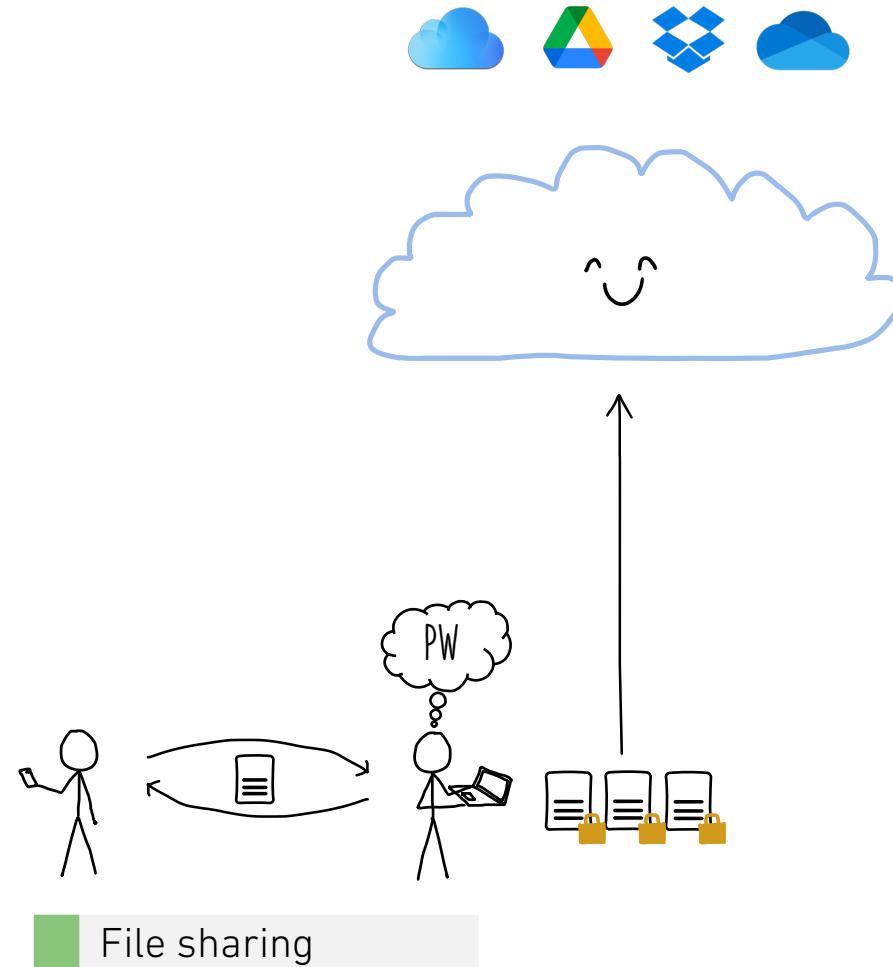
Provable security



# Formalizing E2EE Cloud Storage

In scope:

- Provable security
- Multi-device access

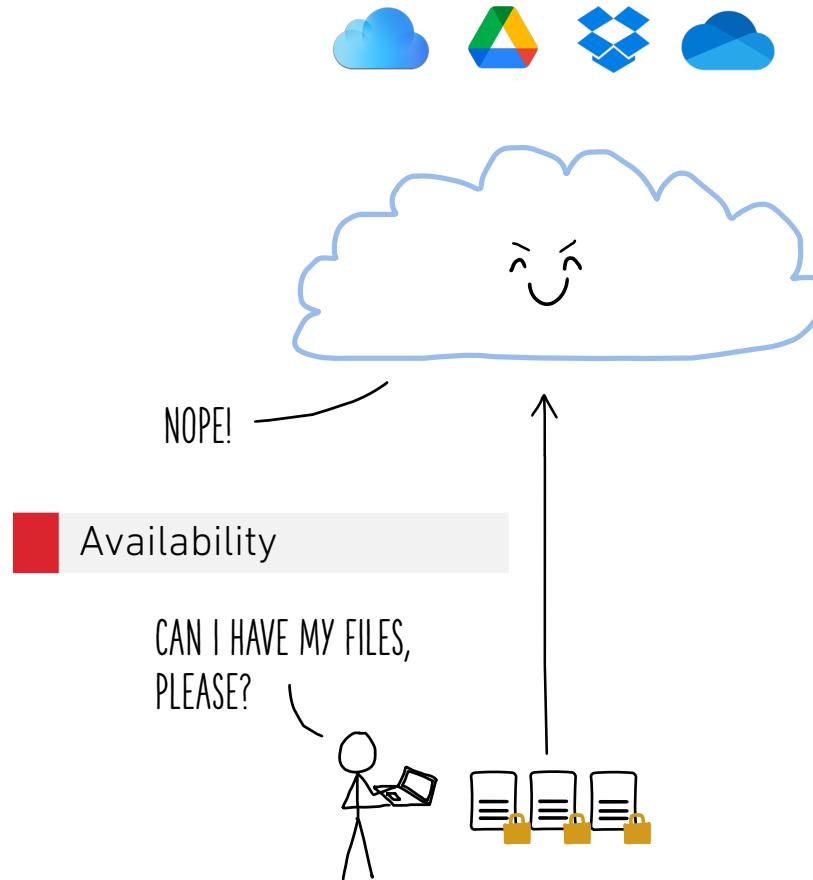


# Formalizing E2EE Cloud Storage

In scope:

- Provable security
- Multi-device access
- File sharing

Out of scope:



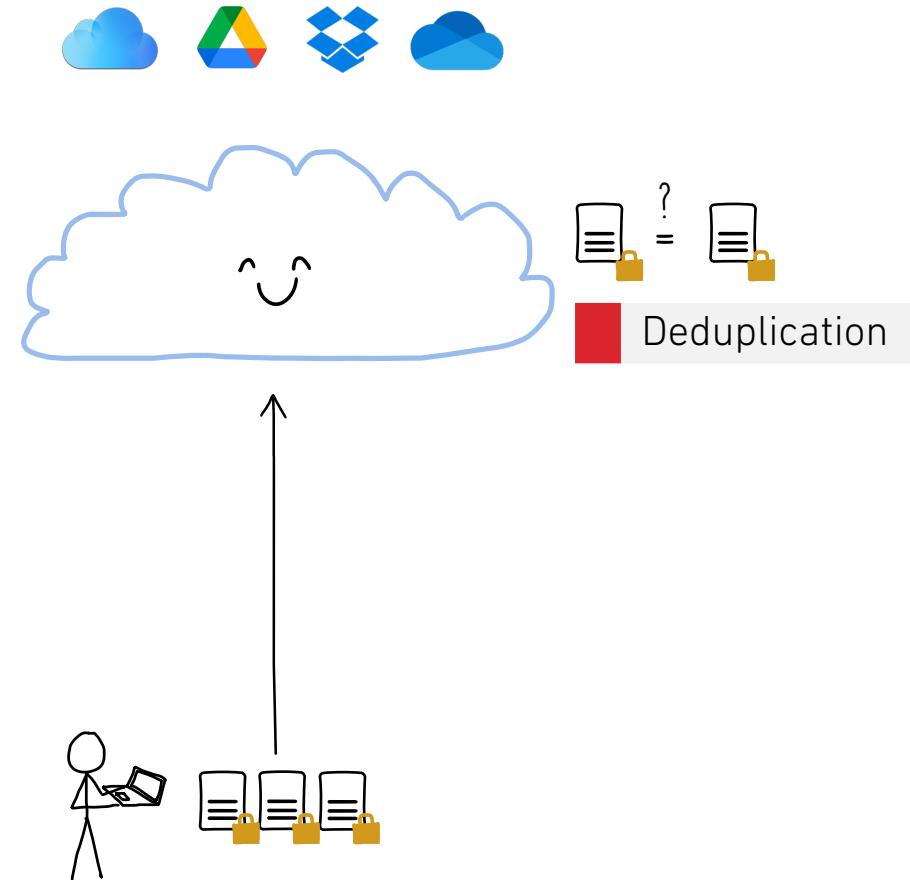
# Formalizing E2EE Cloud Storage

In scope:

- Provable security
- Multi-device access
- File sharing

Out of scope:

- Availability



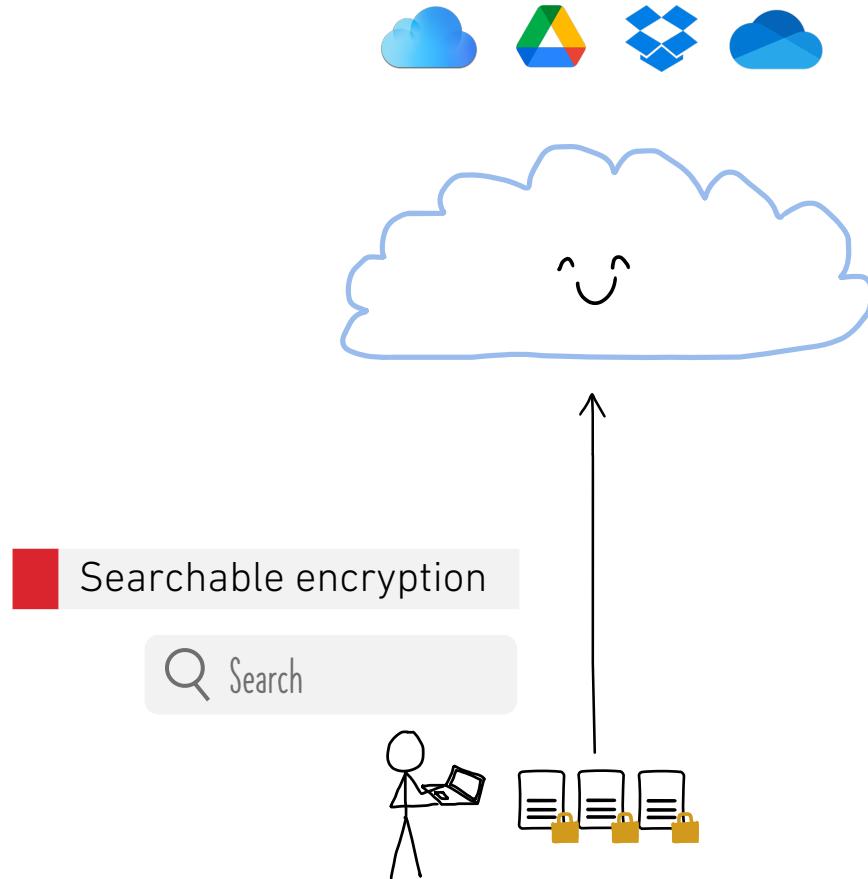
# Formalizing E2EE Cloud Storage

## In scope:

- Provable security
- Multi-device access
- File sharing

## Out of scope:

- Availability
- Deduplication



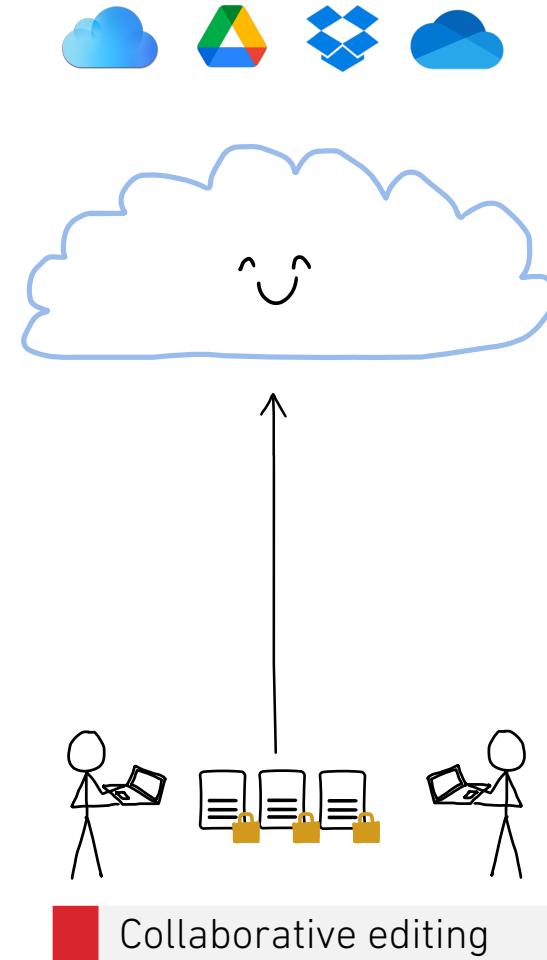
# Formalizing E2EE Cloud Storage

## In scope:

- Provable security
- Multi-device access
- File sharing

## Out of scope:

- Availability
- Deduplication
- Searchable encryption



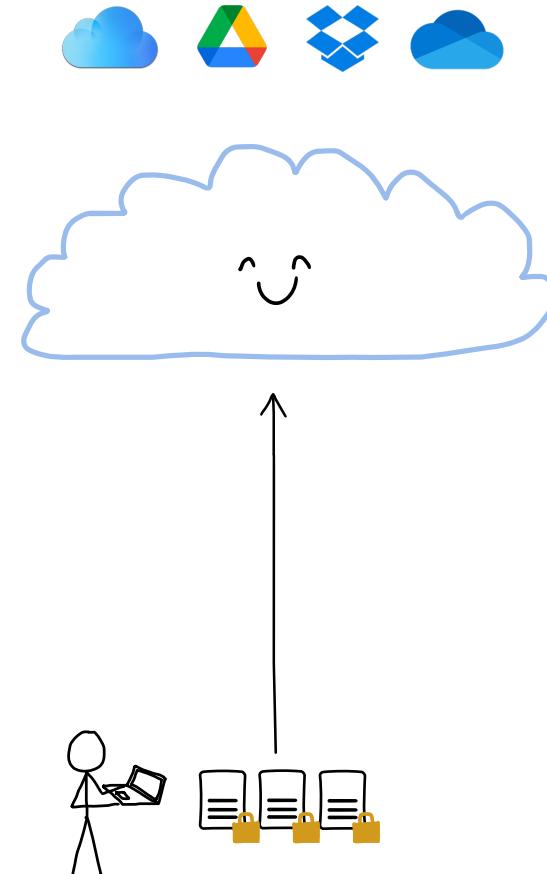
# Formalizing E2EE Cloud Storage

## In scope:

- Provable security
- Multi-device access
- File sharing

## Out of scope:

- Availability
- Deduplication
- Searchable encryption
- Collaborative editing
- Advanced Security
  - Metadata & access pattern hiding
  - Revocable access
  - Forward secrecy
  - ...



# Formalizing E2EE Cloud Storage

## Model Goals



Proton Drive



icedrive



Capture existing systems

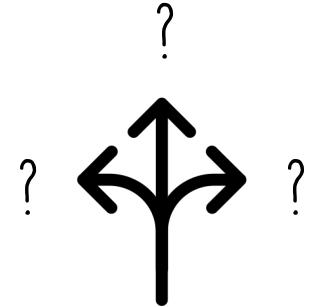
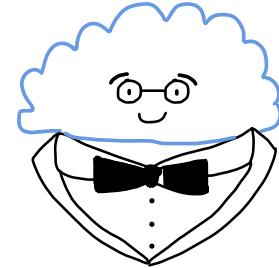
1 Expressive



Capture *real-world* systems

2 Faithful

ALL MODELS ARE WRONG,  
BUT SOME ARE USEFUL!



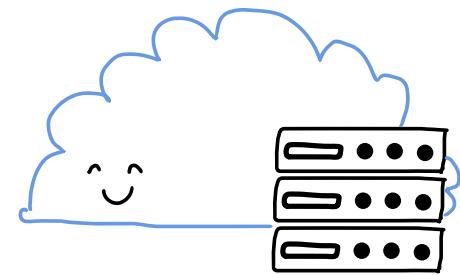
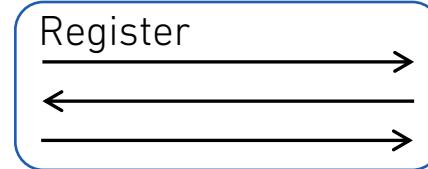
Capture future systems

3 Generic

## Core Functionality

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

INTERACTIVE  
PROTOCOLS



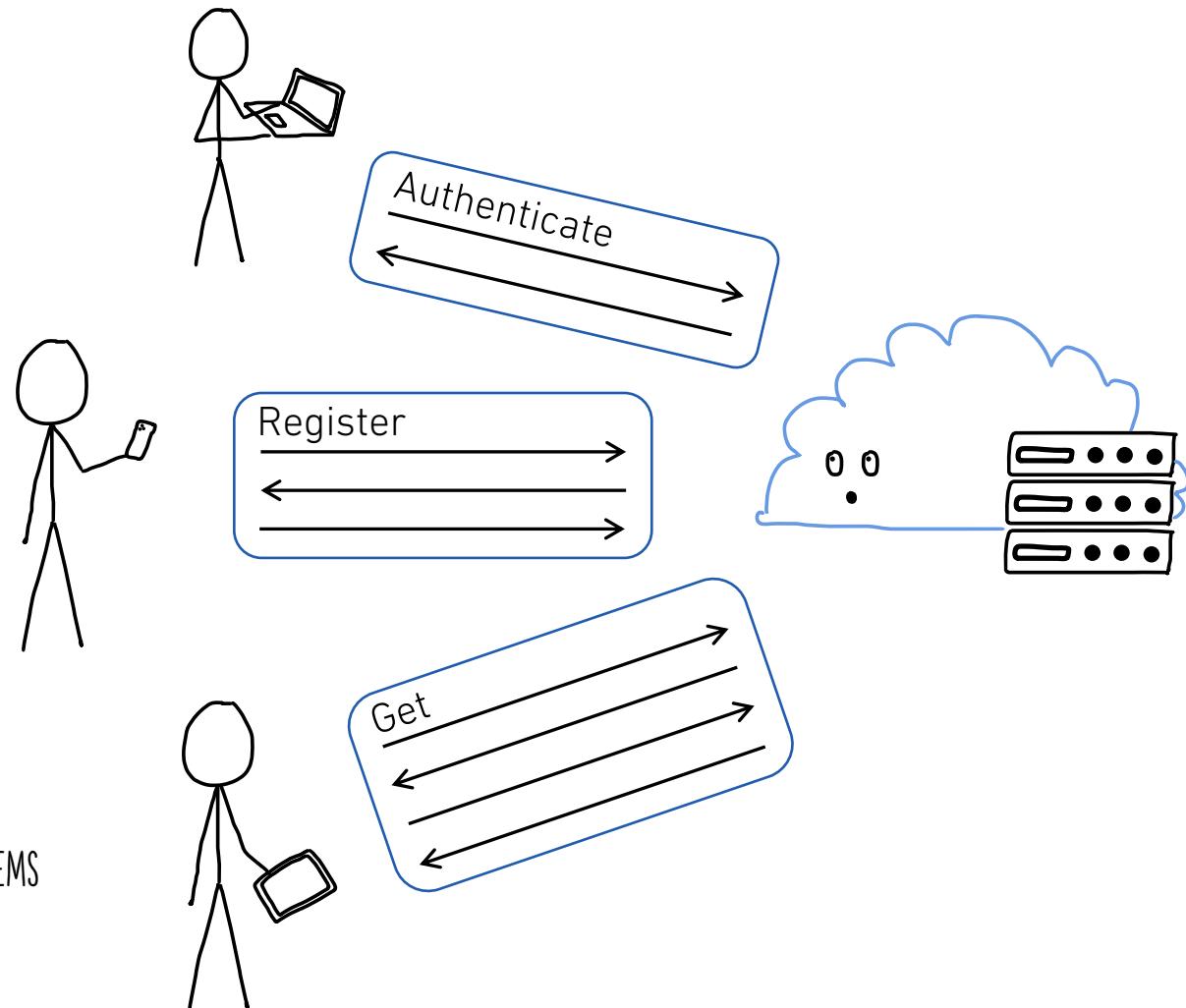
# Syntax

## HOW DO WE MAKE THE MODEL USEFUL?

### 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

- Non-atomic operations → FAITHFUL TO REAL-WORLD SYSTEMS

# Syntax

# HOW DO WE MAKE THE MODEL USEFUL?

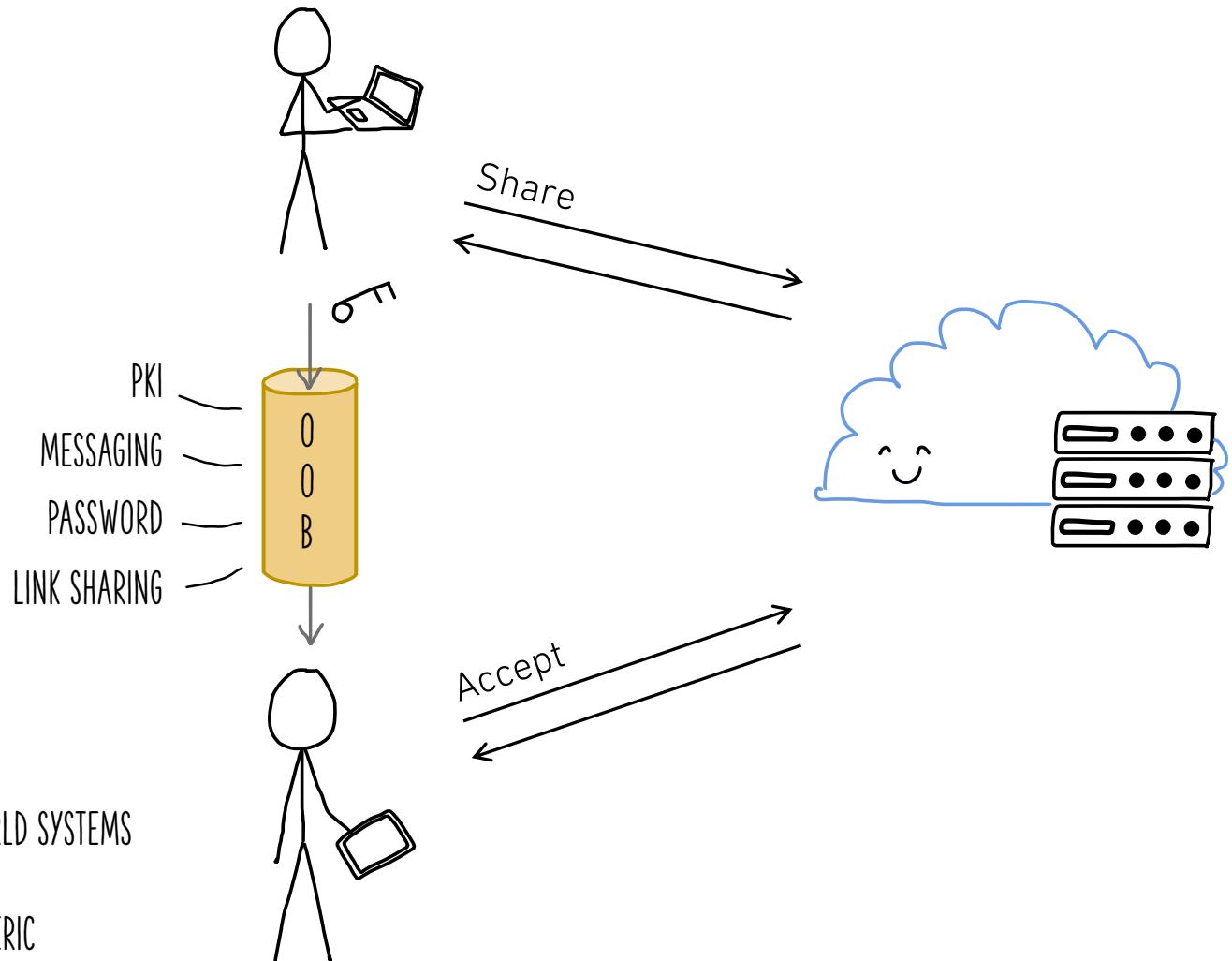
## Core Functionality

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

INTERACTIVE  
PROTOCOLS

## Model Choices

- Non-atomic operations → FAITHFUL TO REAL-WORLD SYSTEMS
- Abstract OOB channel for sharing → GENERIC

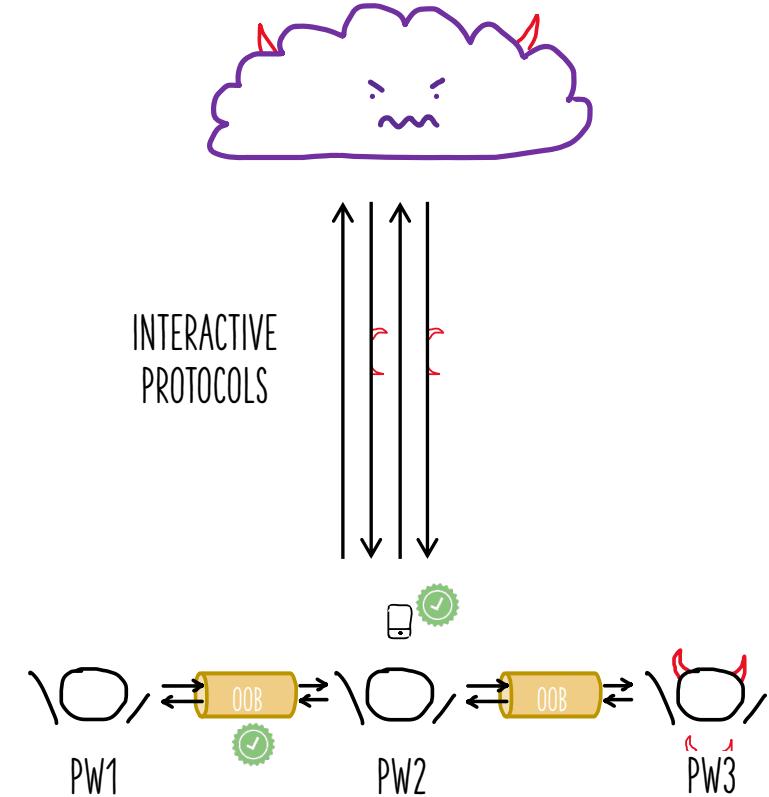


### Threat model:

- Malicious cloud provider
- Trusted OOB-channels between honest users
- Trusted client code

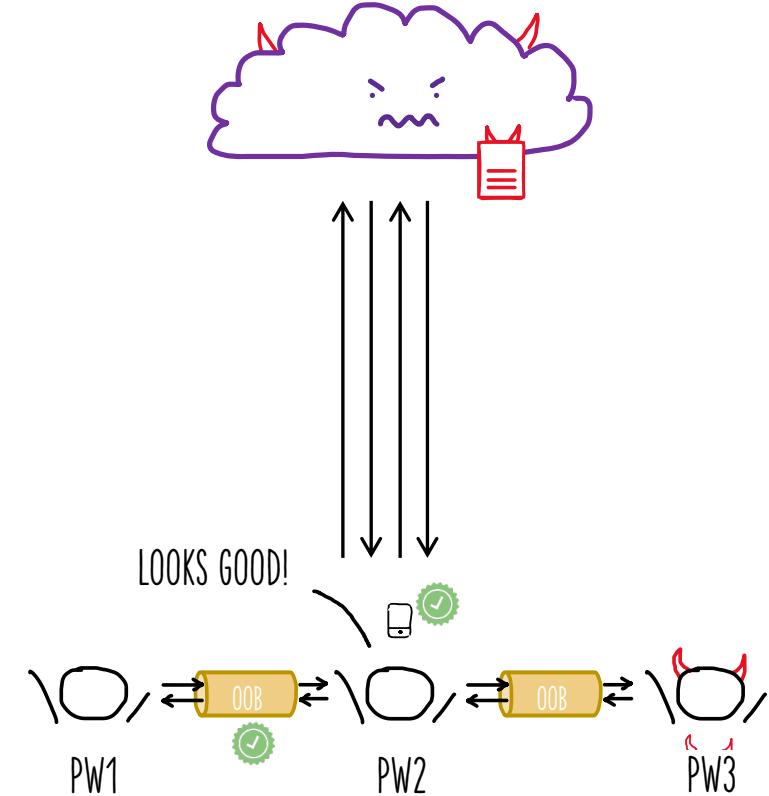
### Adversary capabilities:

- Control client protocol steps (which & when)
- Specify server responses
- Guess honest user passwords
- Compromise users (adaptive/selective)



### Integrity:

- Wins if adversary can, for an honest user,
  1. inject a file, or
  2. modify a file.



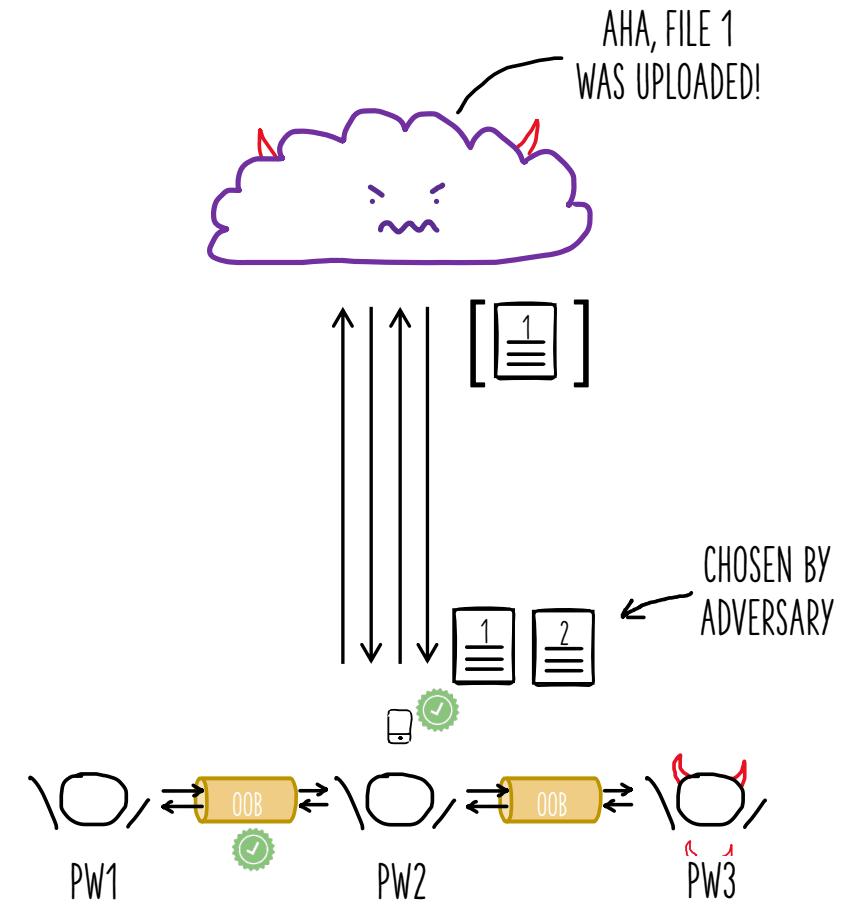
### Integrity:

- Wins if adversary can, for an honest user,
  1. inject a file, or
  2. modify a file.

### Confidentiality:

- Wins if adversary can, for an honest user,
  - learn any information and distinguish files

IND-CPA-style game

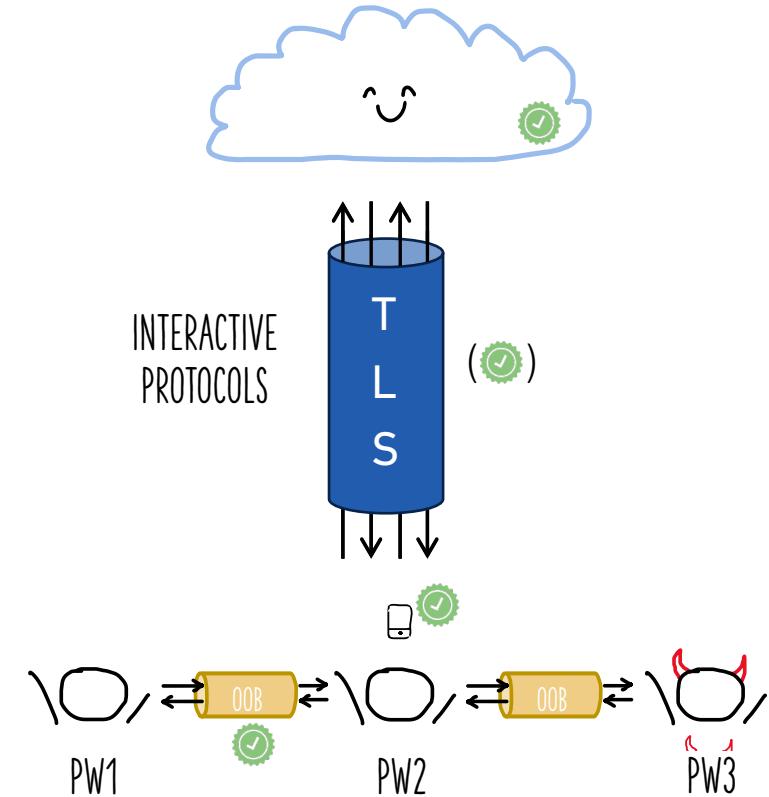


### Threat model:

- ~~Malicious~~ honest cloud provider, malicious clients
- Trusted OOB-channels between honest users
- Trusted client code
- + Trusted client-to-server channels?

### Adversary capabilities:

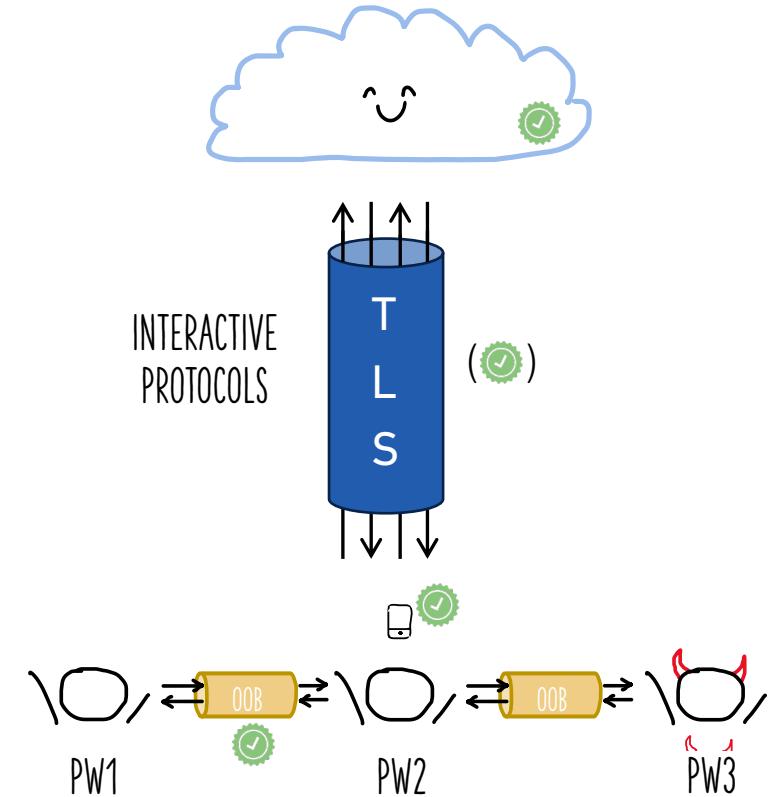
- Control client protocol steps (which & when)
- ~~Specify server responses~~
- Guess honest user passwords
- Compromise users (adaptive/selective)



### Additional goals:

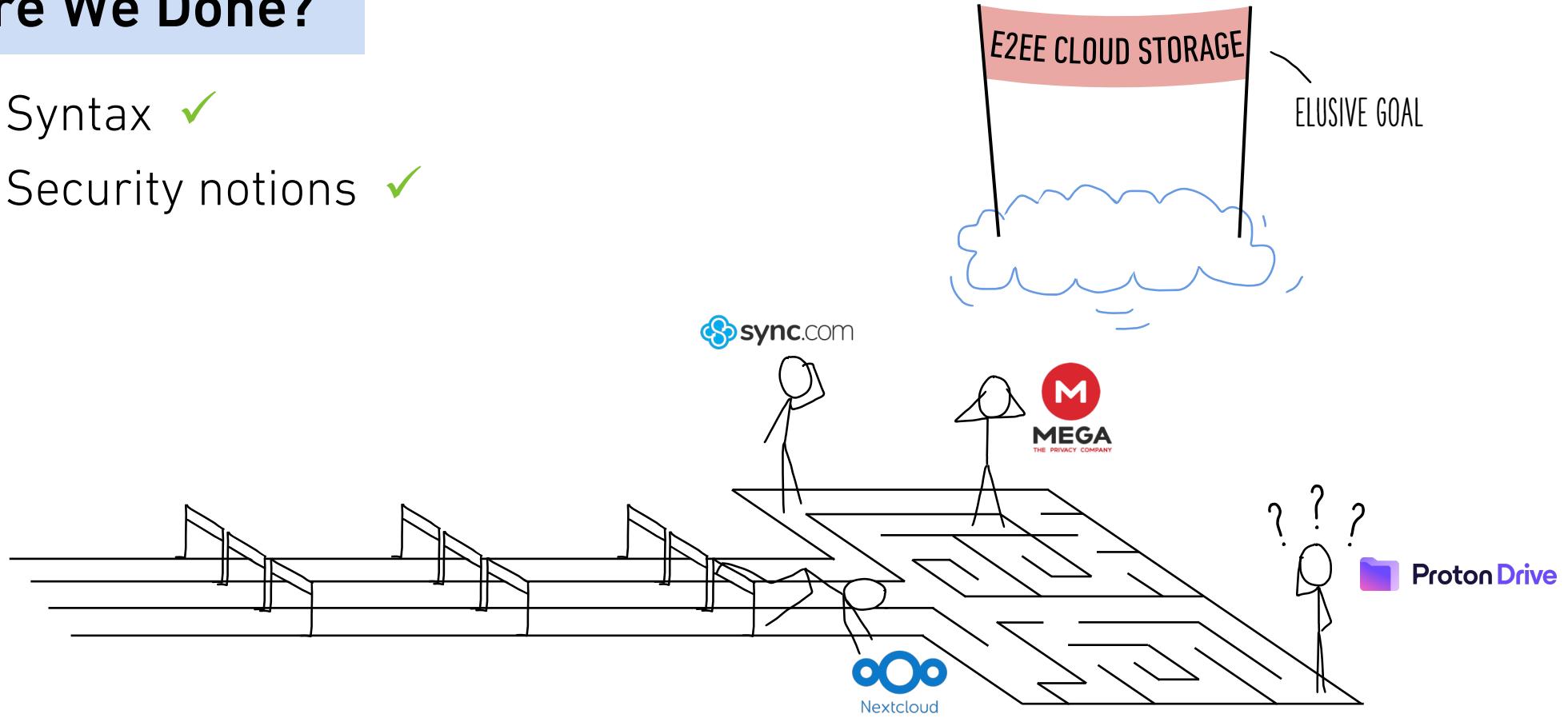
- Authentication & authorization
- No offline dictionary attacks on pw
- Availability for honest user files

INFEASIBLE IN MALICIOUS SERVER SETTING!



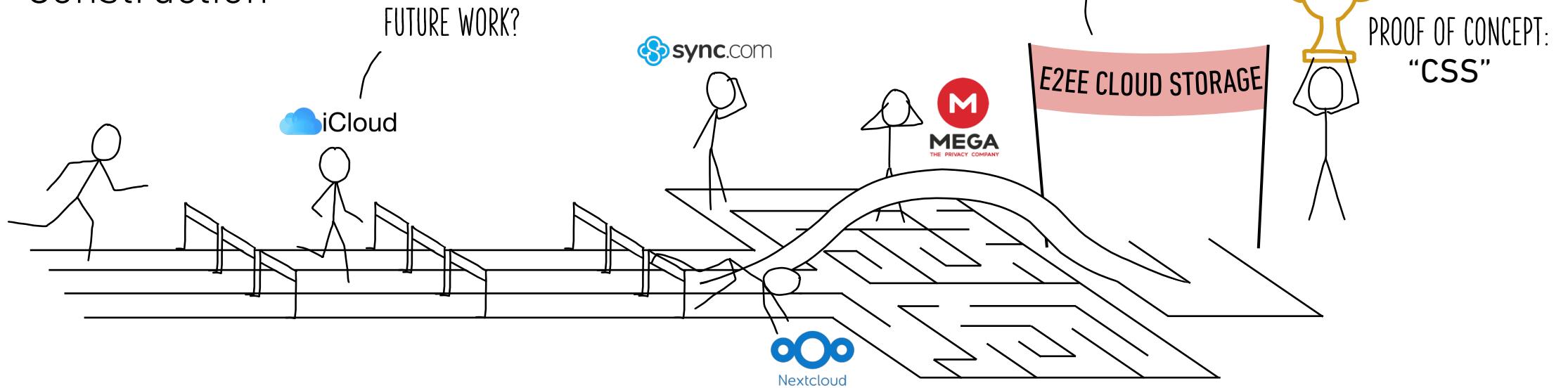
# Are We Done?

- Syntax ✓
- Security notions ✓



# Are We Done?

- Syntax ✓
- Security notions ✓
- Construction

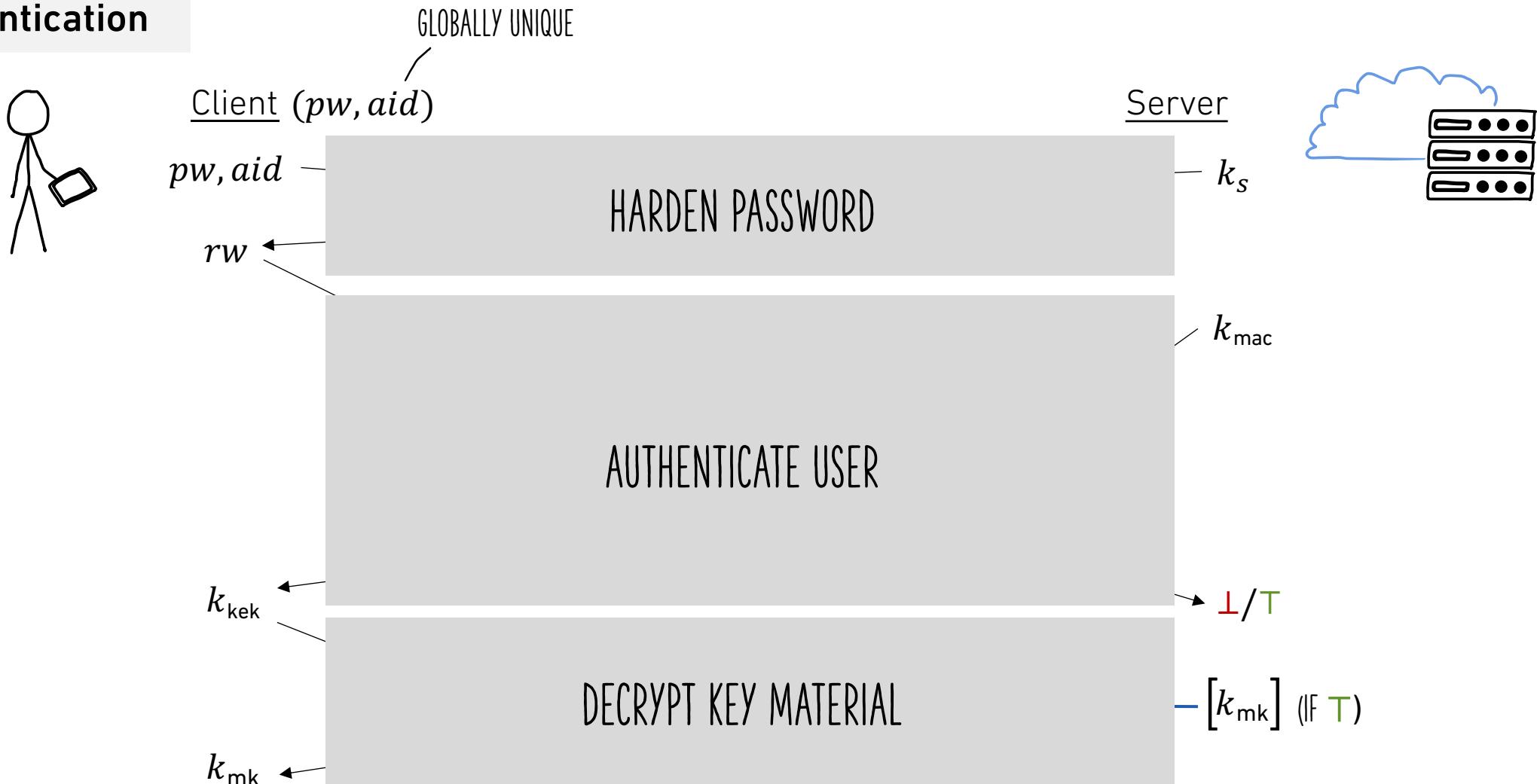


## **2. Constructing E2EE Cloud Storage**



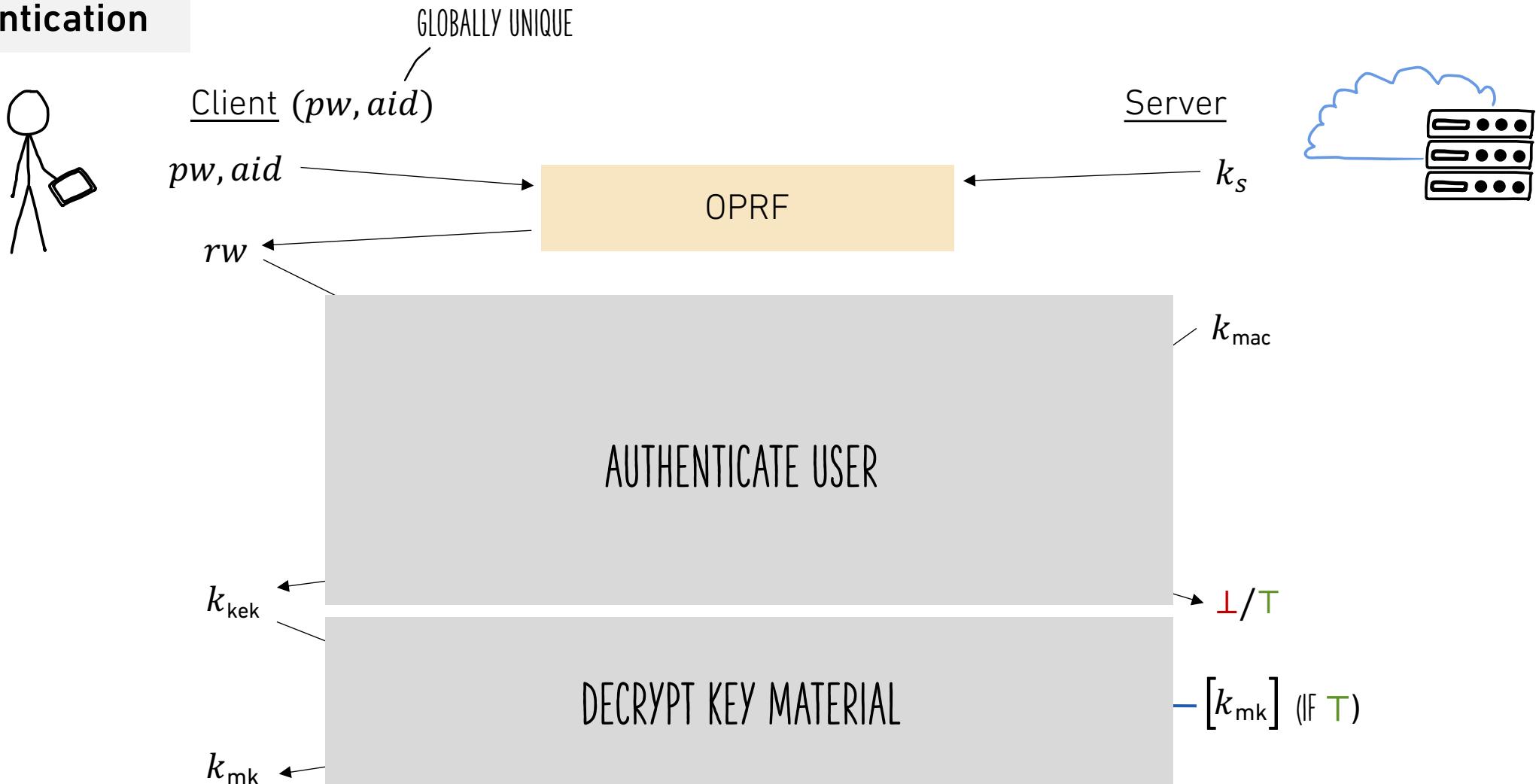
# CSS (Cloud Storage Scheme)

## Authentication



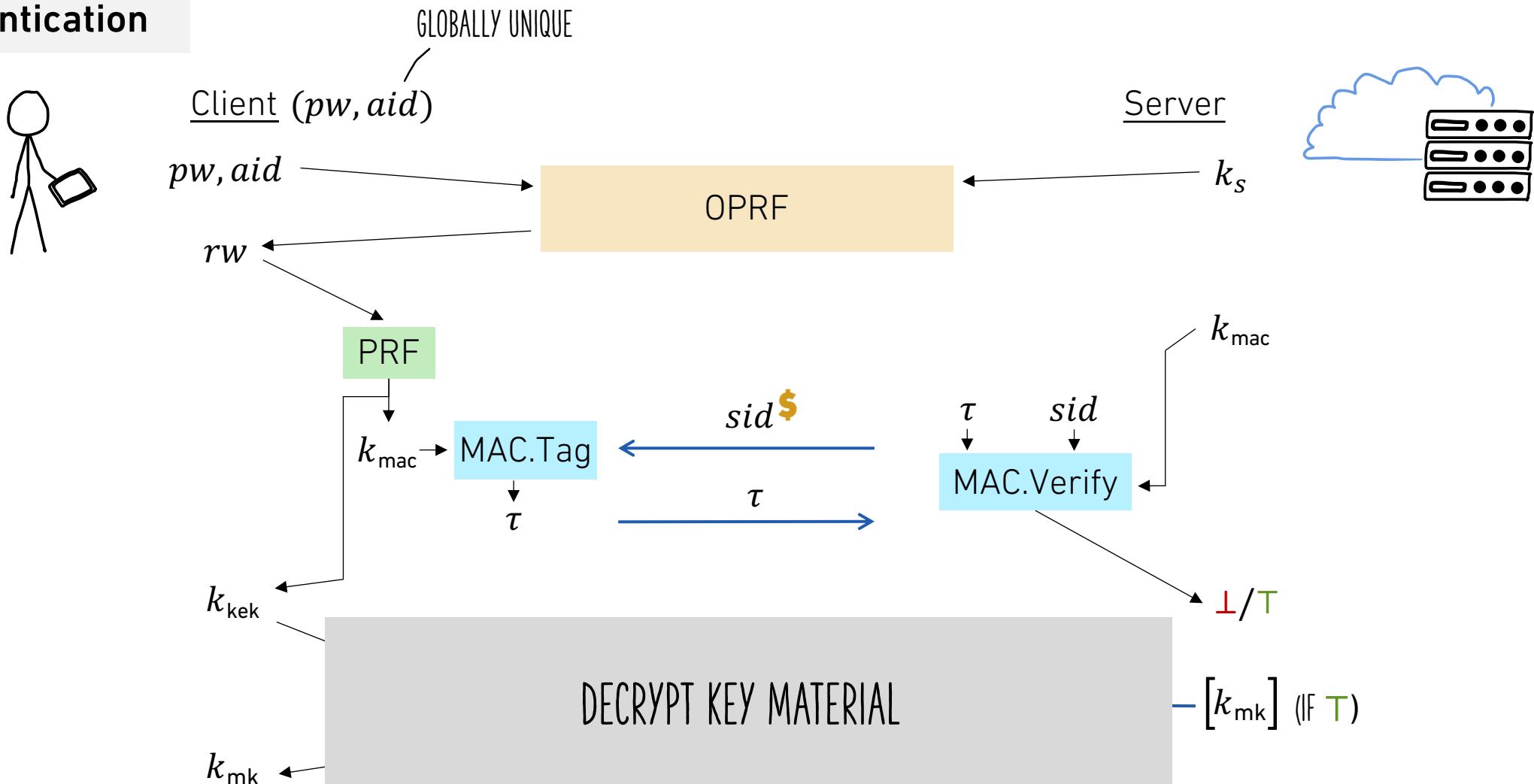
# CSS (Cloud Storage Scheme)

## Authentication



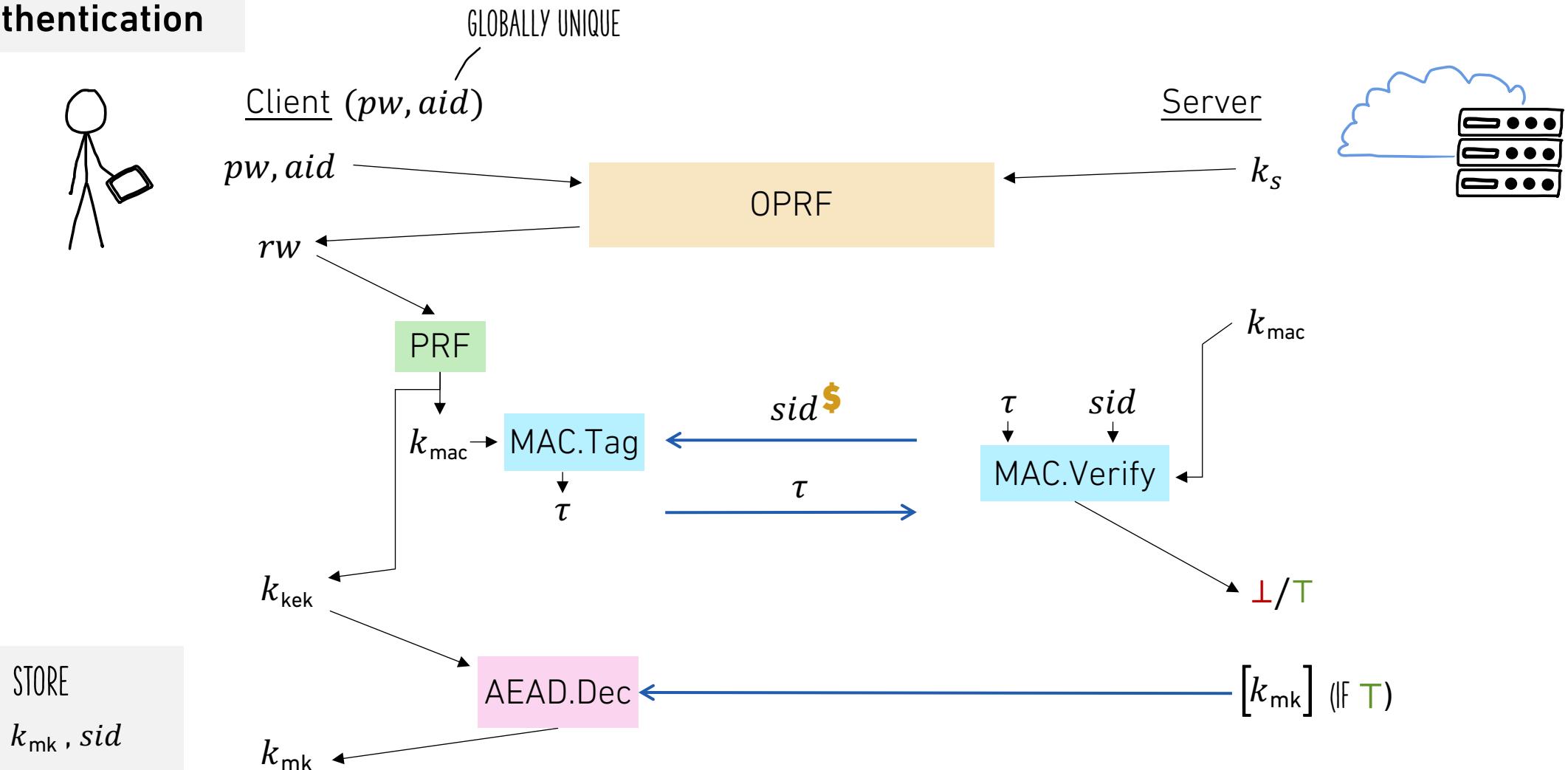
# CSS (Cloud Storage Scheme)

## Authentication



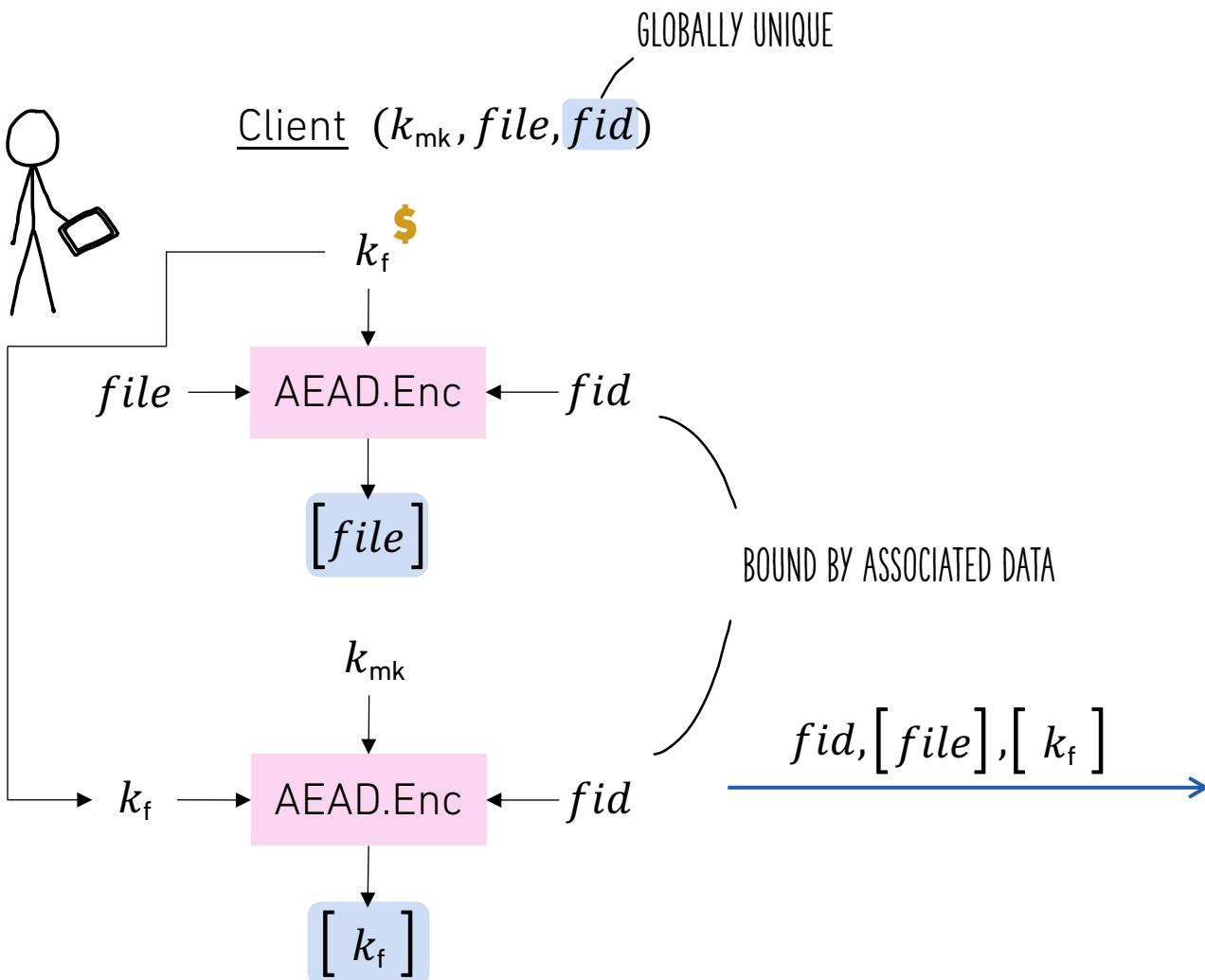
# CSS (Cloud Storage Scheme)

## Authentication

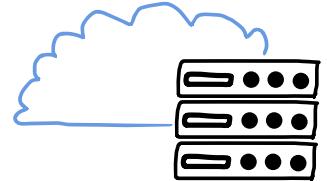


# CSS (Cloud Storage Scheme)

Put



Server

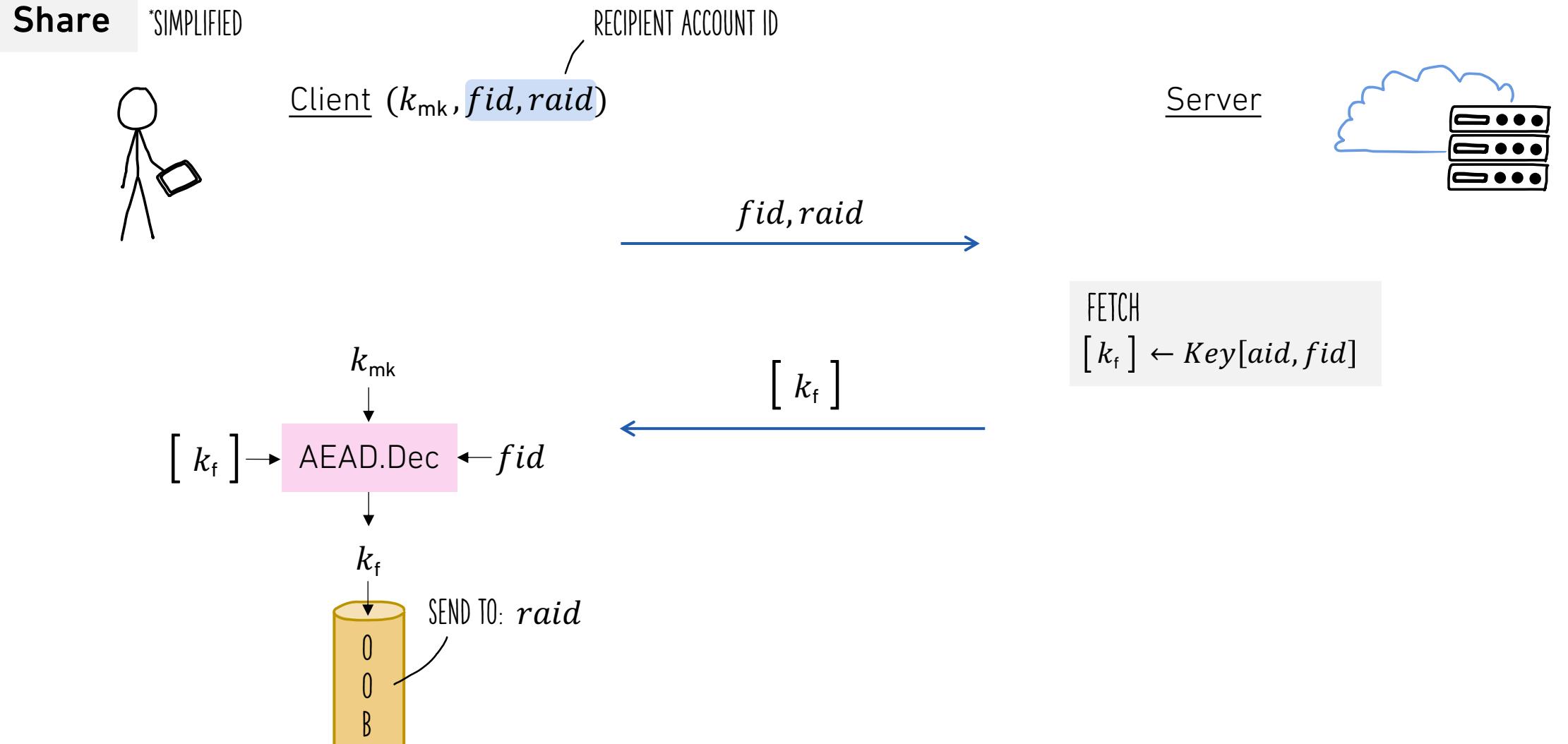


STORE

$File[fid] \leftarrow [file]$

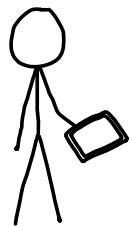
$Key[aid, fid] \leftarrow [k_f]$

# CSS (Cloud Storage Scheme)

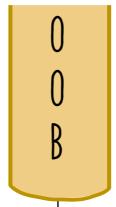


# CSS (Cloud Storage Scheme)

Accept \*Simplified



Client ( $k_{mk}, fid$ )



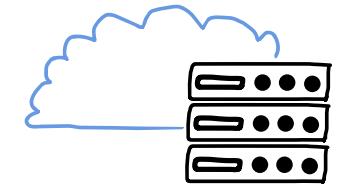
$k_{mk}$

$k_f \rightarrow \text{AEAD.Enc} \leftarrow fid$

[  $k_f$  ]

$fid, [ k_f ]$

Server



STORE

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

# The Future of E2EE Cloud Storage

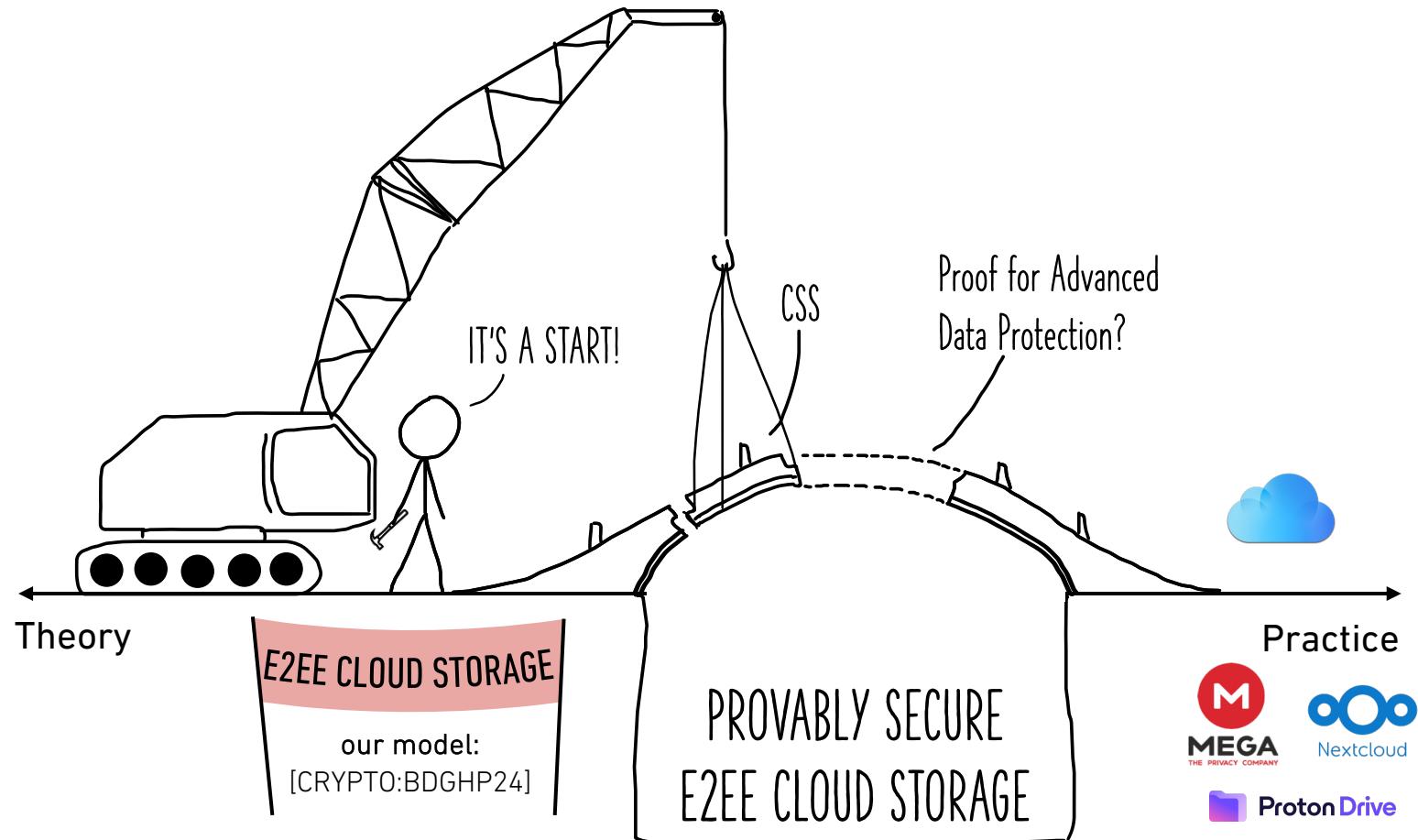
## The good news

- Confidentiality
- Integrity
- Interactive protocols

FUTURE WORK

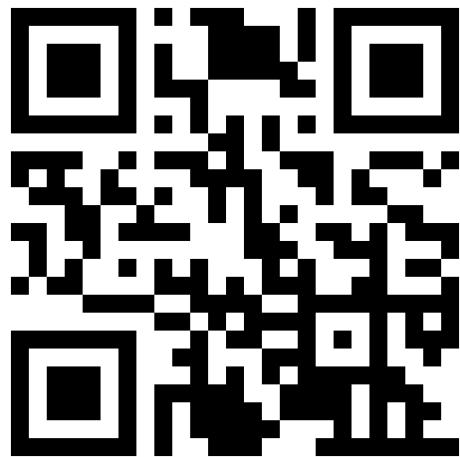
## The bad news

- PW-based key hierarchy
- Mandatory identity provider
- Functionality match?
- Adaptive security proof



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

**Matilda Backendal**, Hannah Davis, Felix Günther, **Miro Haller**, Kenny Paterson  
[mbackendal@inf.ethz.ch](mailto:mbackendal@inf.ethz.ch)      [mhaller@ucsd.edu](mailto:mhaller@ucsd.edu)



[eprint.iacr.org/2024/989](https://eprint.iacr.org/2024/989)

