Why **E2EE Cloud Storage** is hard

Lessons Learnt from

MEGA
Malleable Encryption Goes Awry

Cryptanalysis of the MEGA Cloud Storage
by **Miro Haller**, **Matilda Backendal** & Kenny Paterson

APPLIED CRYPTO GROUP
ETH zürich
UC San Diego

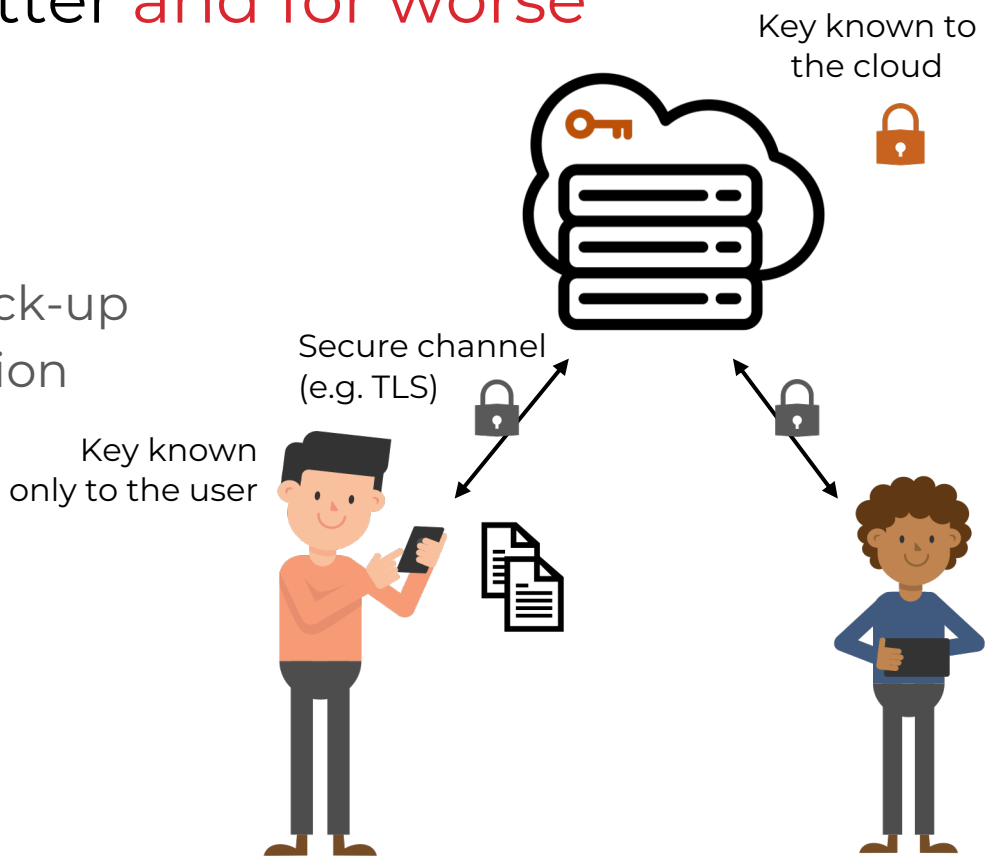
Cloud storage, for better and for worse

Advantages

+ Outsource storage
+ Easy file access and back-up
+ Sharing and collaboration

and disadvantages...

- Privacy



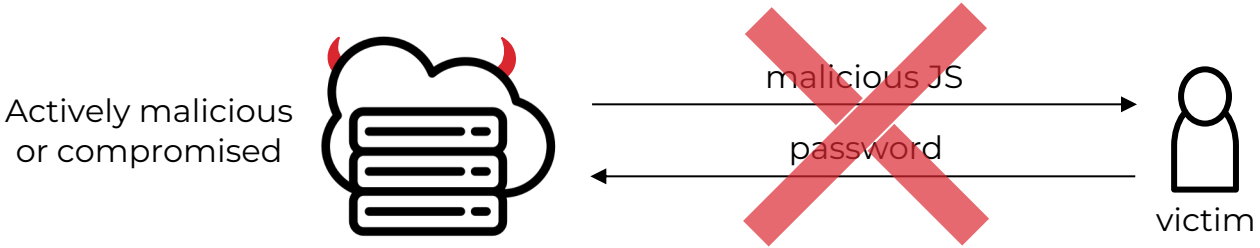
End-to-End Encryption: Why do we care?

- Without E2EE: Cloud provider can
  - ...read sensitive files
  - ...perform analytics and serve targeted advertising
  - ...be hacked by malicious external actors
- With E2EE: Even a malicious cloud *cannot*
  - ...access user data
  - ...modify user files

Image from: https://dayoftheshirt.com/shirts/93714/have-your-cake-and-eat-it-too-teeturtle

Threat model and scope

- Security goals:
  - Confidentiality and integrity



- Out of scope:
  - Availability
  - User anonymity
  - Targeted dictionary attacks on user password
  - Serving malicious JavaScript

Consumer cloud storage

Provider	Active users
Google Drive	> 1 billion
OneDrive	0.5 – 1 billion
iCloud	> 850 million
Dropbox	>700 million

Sources:

Google Drive (2018): https://techcrunch.com/2018/07/25/google-drive-will-hit-a-billion-users-this-week/?guccounter=1

OneDrive (2015, 2022): https://www.computerworld.com/article/3003140/microsofts-onedrive-changes-follow-the-money.html, https://news.microsoft.com/bythenumbers/en/give

iCloud (2018): https://www.cnbc.com/2018/02/11/apple-could-sell-icloud-for-the-enterprise-barclays-says.html

Dropbox (2022): https://dropbox.gcs-web.com/news-releases/news-release-details/dropbox-announces-second-quarter-fiscal-2022-results

Consumer cloud storage lacks privacy

Provider	Active users	E2EE
Google Drive	> 1 billion	✗
OneDrive	0.5 – 1 billion	✗
iCloud	> 850 million	✗
Dropbox	>700 million	✗

Sources:

Google Drive (2018): https://techcrunch.com/2018/07/25/google-drive-will-hit-a-billion-users-this-week/?guccounter=1

OneDrive (2015, 2022): https://www.computerworld.com/article/3003140/microsofts-onedrive-changes-follow-the-money.html, https://news.microsoft.com/bythenumbers/en/give

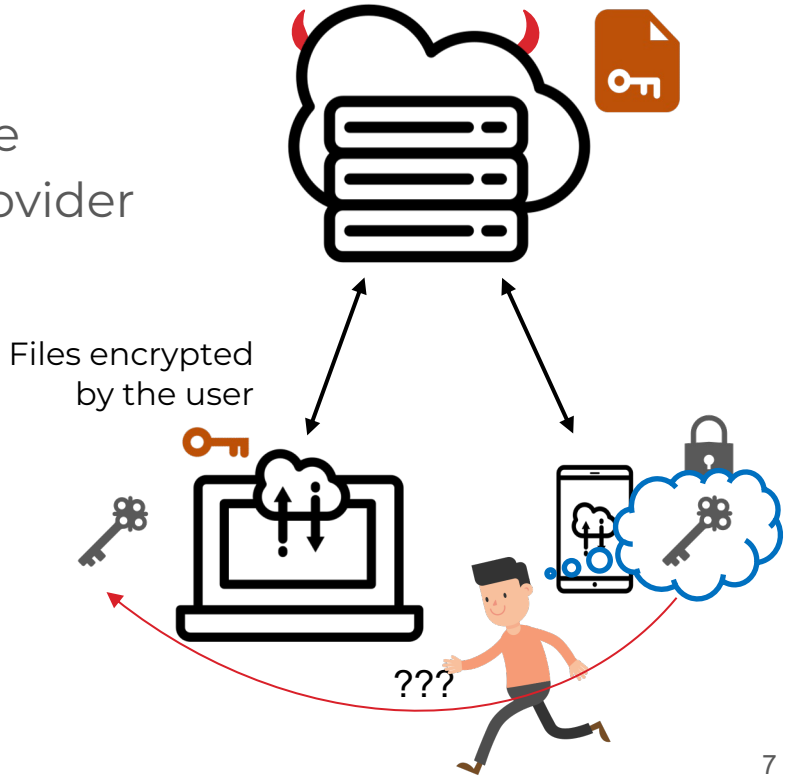
iCloud (2018): https://www.cnbc.com/2018/02/11/apple-could-sell-icloud-for-the-enterprise-barclays-says.html

Dropbox (2022): https://dropbox.gcs-web.com/news-releases/news-release-details/dropbox-announces-second-quarter-fiscal-2022-results

Client-side encryption: How hard can it be?

Problem #1: key management

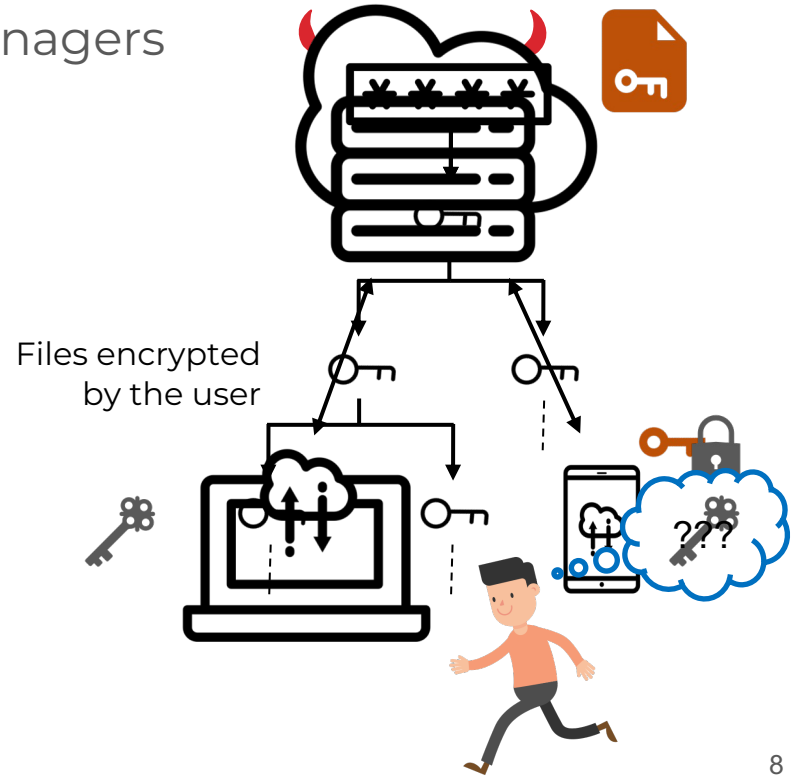
- Want: data available from any device
- Challenge: transfer via untrusted provider
- Solution: encrypt the keys



Client-side encryption: How hard can it be?

Problem #2: users are not good key managers

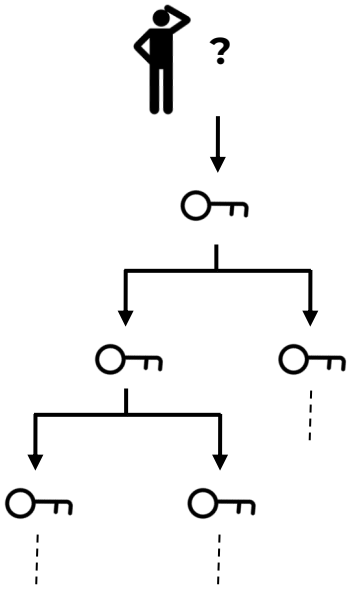
- Solution: passwords



Client-side encryption: How hard can it be?

Problem #2: users are not good key managers

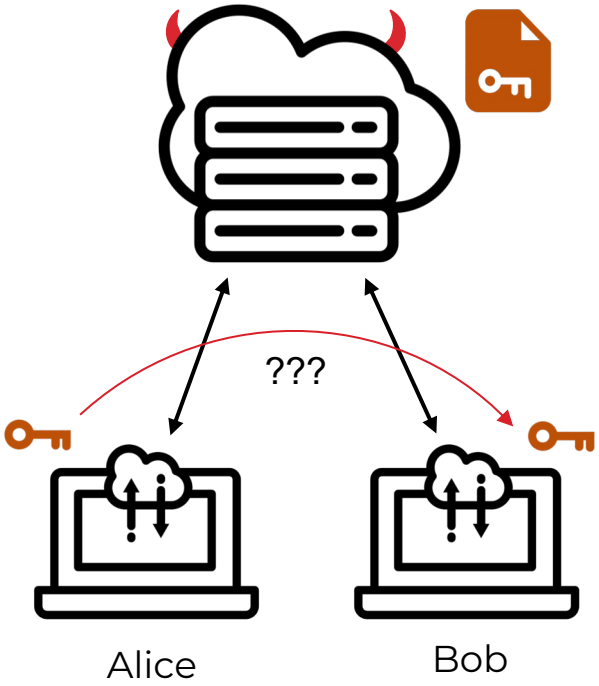
- Solution: passwords
- Challenge: passwords!
  - Users are not good password managers either…
  - Forgotten password ⇒ lost access
  - Password leak/compromise ⇒ key recovery



Client-side encryption: How hard can it be?

Problem #3: sharing encrypted files

- Want: keys shared across users
- Challenge: establishing trusted channel



MEGA's Design

Who is MEGA?

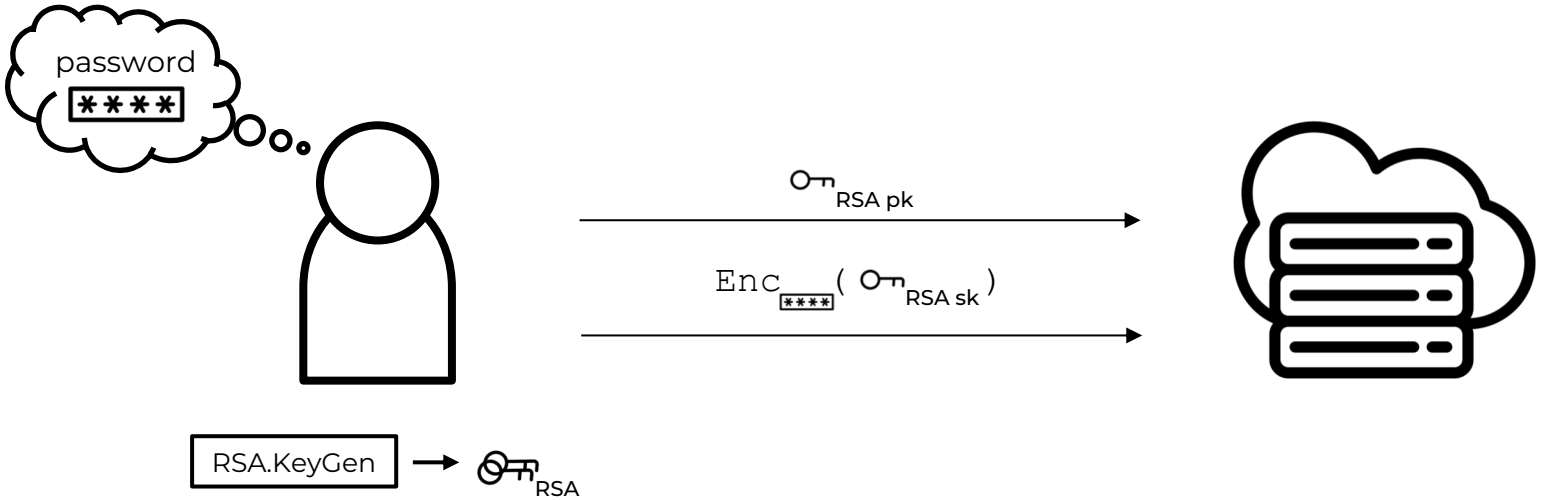
"MEGA does not have access to your password or your data."
https://mega.io/security

The largest E2EE cloud storage service

- 10+ million daily active users
- 270+ million accounts
- 130+ billion files
- 1000+ PB of stored data

Setup*

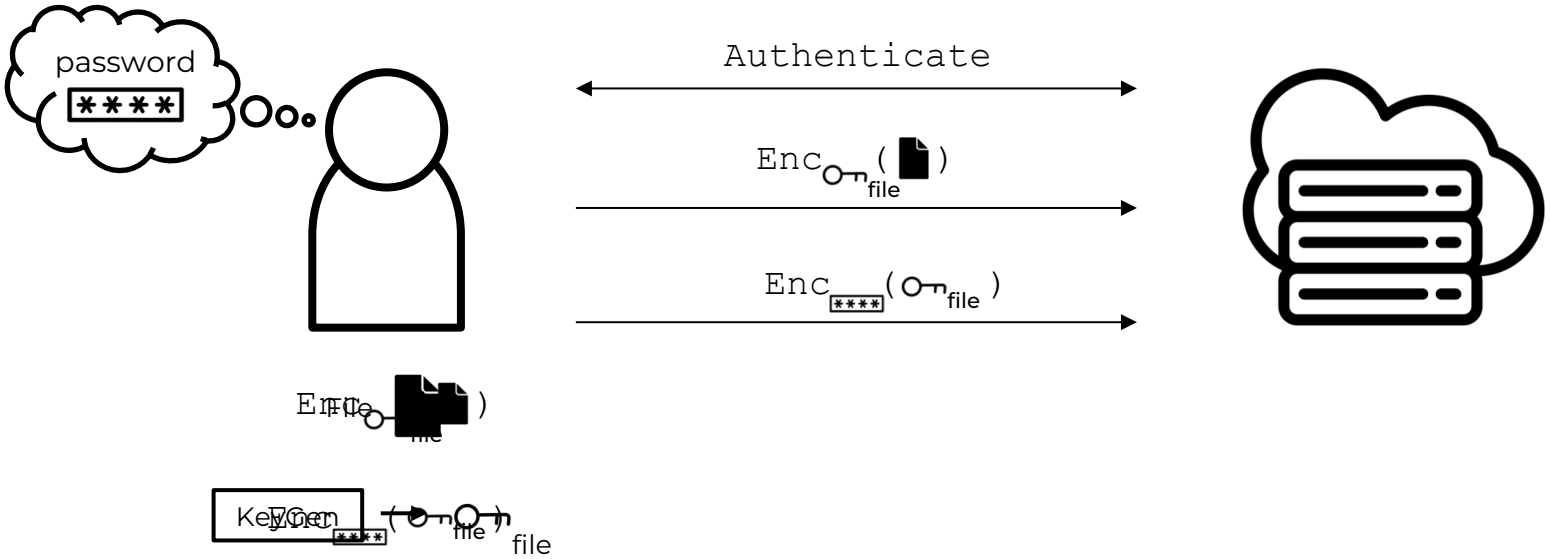
Client uploads encrypted RSA secret key to the cloud to set up authentication.



*strongly simplified

File upload*

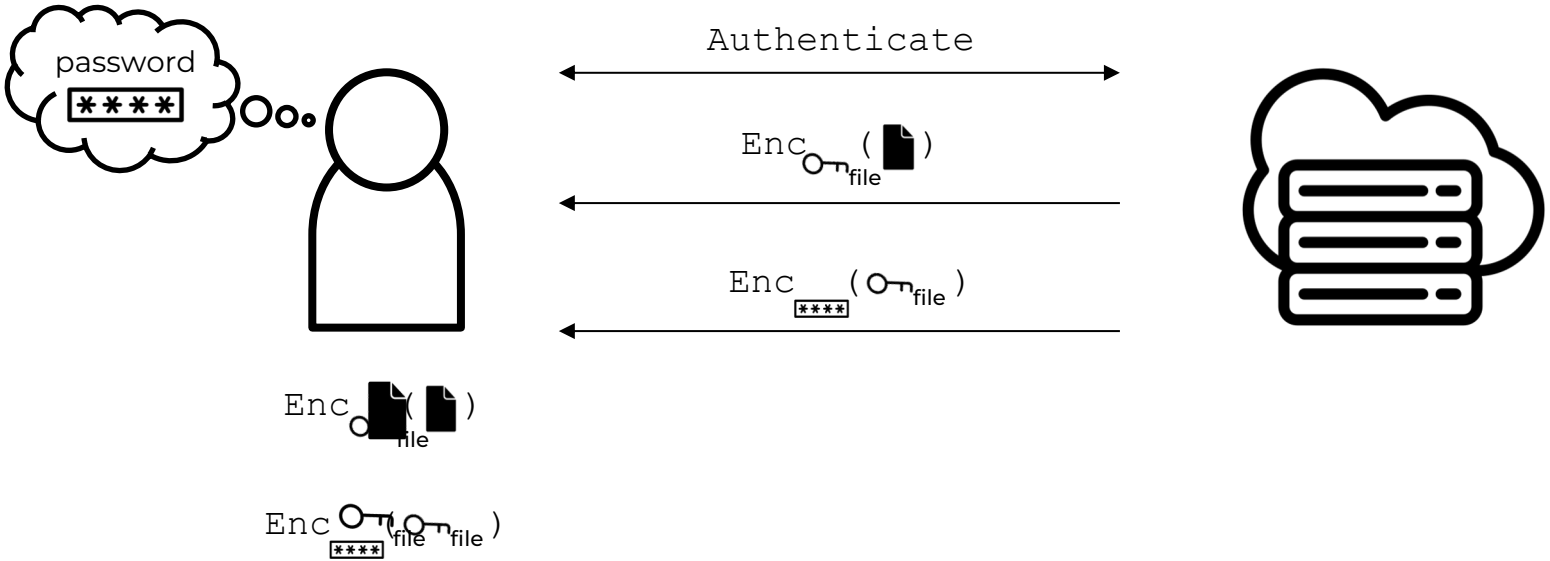
The user locally generates a file encryption key and uploads the encrypted file and encrypted file key to the cloud.



*strongly simplified

File download*

The user retrieves the encrypted file and encrypted key material. They recover the file key using the password and decrypt the file.

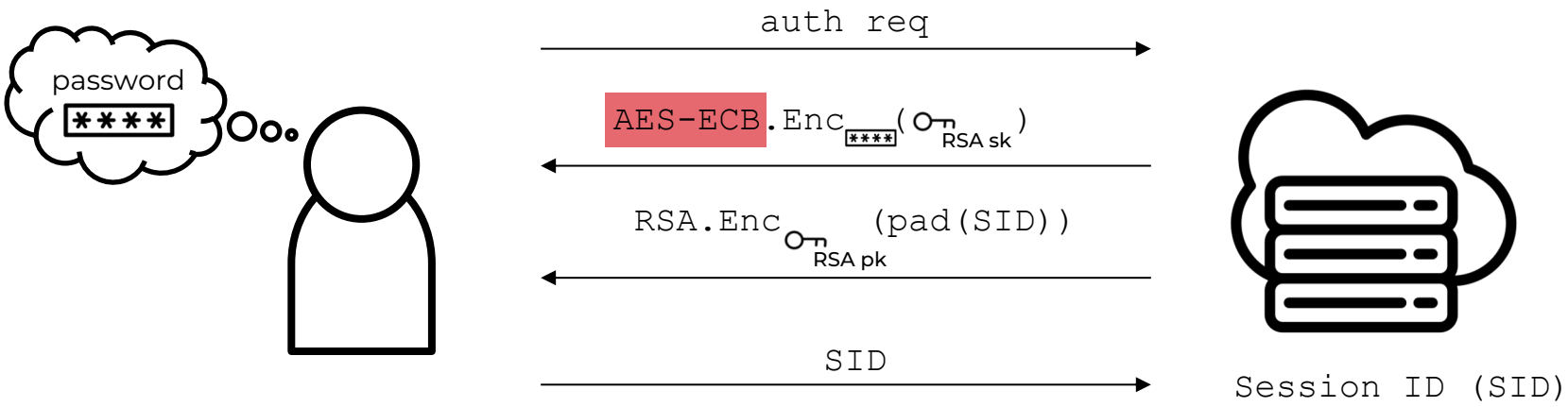


*strongly simplified

Cryptanalysis of **MEGA**

Attack 1: RSA key recovery*

MEGA's user authentication:

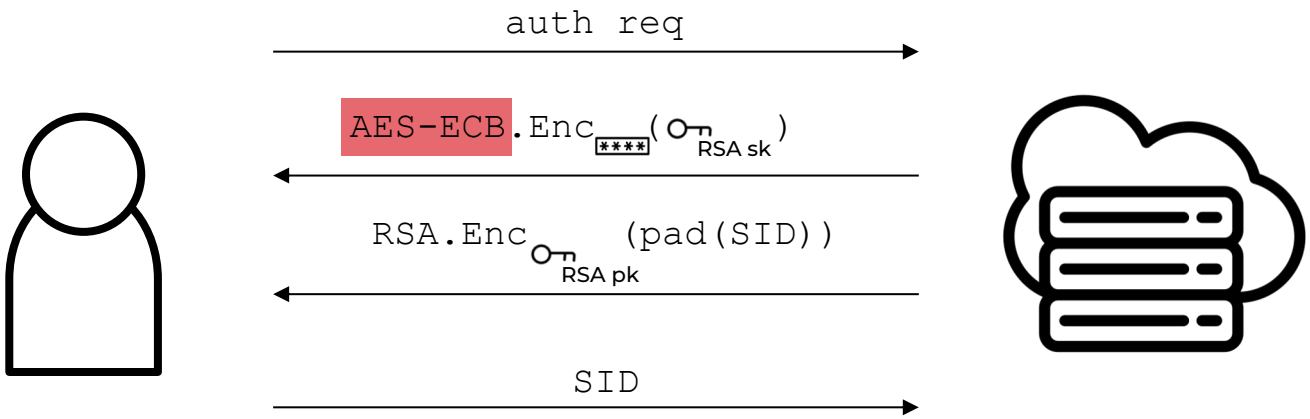


AES-ECB is clearly a bad choice.

*strongly simplified

Attack 1: RSA key recovery*

MEGA's user authentication:

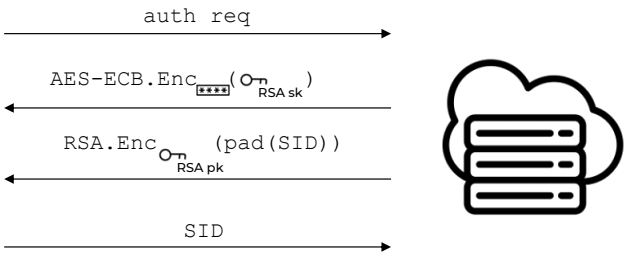


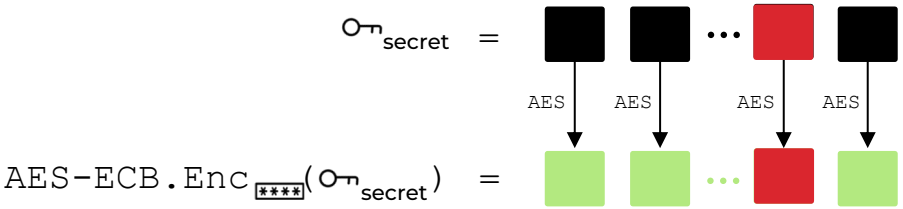
AES-ECB is clearly a bad choice.

*strongly simplified

Attack 1: RSA key recovery*

MEGA's user authentication:



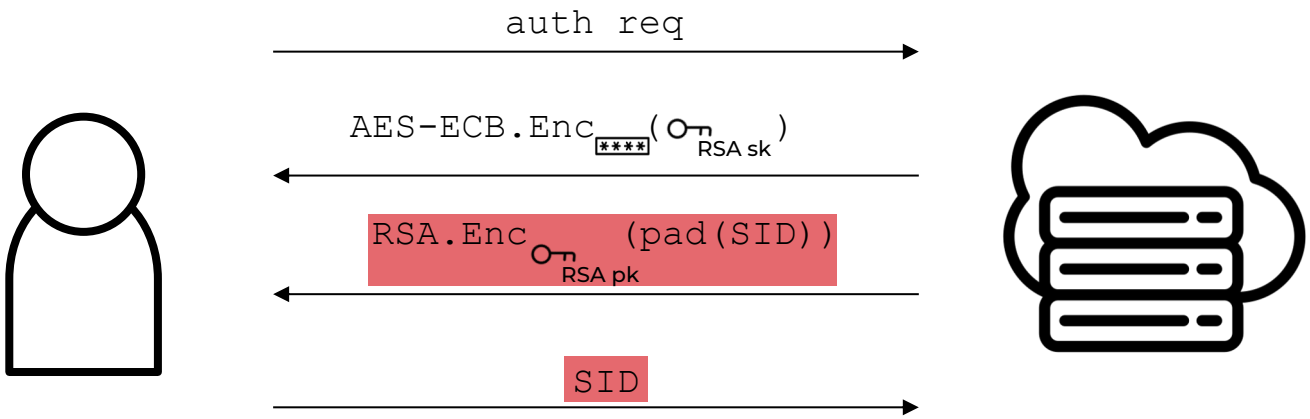


AES-ECB is clearly a bad choice.

*strongly simplified

Attack 1: RSA key recovery*

MEGA's user authentication:

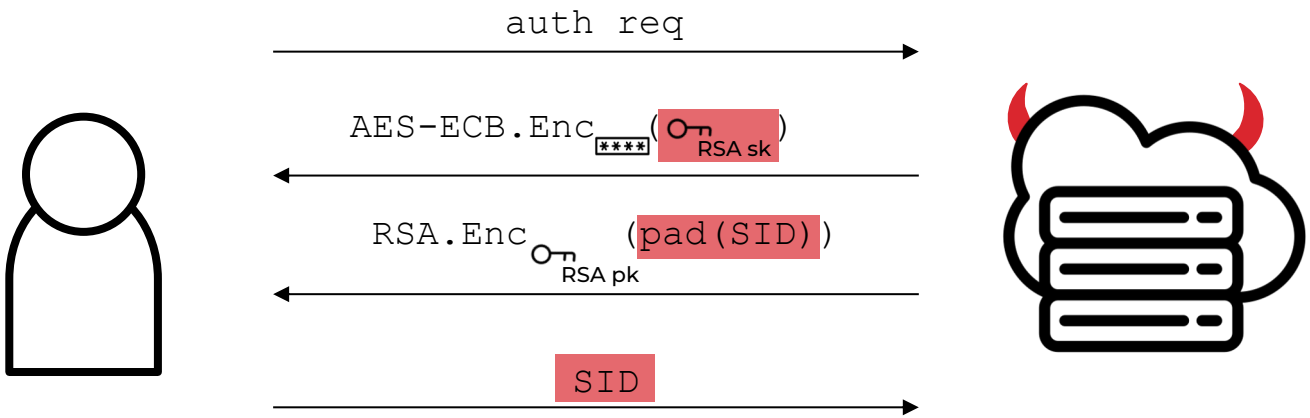


Partial decryption oracle for chosen SID!

*strongly simplified

Attack 1: RSA key recovery*

MEGA's user authentication:



malleability of the secret key

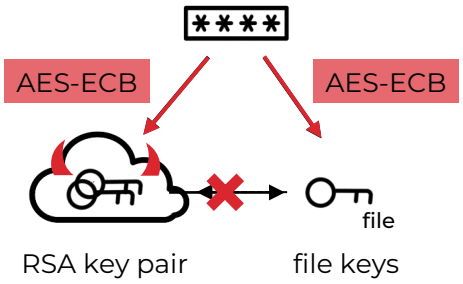
+ information from partial decryption oracle

= binary search for RSA secret key

*strongly simplified

Attack 1: RSA key recovery

- Impact:
  - RSA key recovery in 512 logins*
- Enables attack 2: file key recovery
  - Adversary knows RSA key from Attack 1
  - File keys are also encrypted with AES-ECB

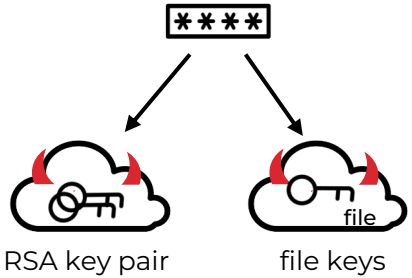


Cut and paste AES-ECB ciphertext blocks from file key to RSA secret key ciphertext.

*Attacks only get better over time:
6 logins: Ryan, Heninger. PKC '23.; 2 logins: Albrecht, Haller, Mareková, Paterson. Eurocrypt '23.

Attack 2: file key recovery

- Impact:
  - Cheap AES-ECB decryption oracle
  - This allows compromise of file (and other) keys
  - And hence decryption of all user data!



Attacks

- Attack 1: RSA key recovery
  - Malleable secret key + oracle
- Attack 2: file key recovery
  - Cut and paste AES ctxt blocks
- Attack 3: integrity attack
  - File forgery under the “zero key”
- Attack 4: framing attack
  - Like attack 3, but not detectable
- Attack 5: Bleichenbacher
  - Adapted to MEGA’s RSA padding

Mistakes to avoid

- No AE for key encryption
- Missing key separation
- Rolling your own crypto
- No cryptographic agility

Lessons from MEGA

- Aim for E2EE
- Have a bug bounty program
- Collaborate during disclosure
- Full mitigation impossible
  - Re-encryption requires > 185 days
- Recovery from compromise?

Mistakes to avoid

- No AE for key encryption
- Missing key separation
- Rolling your own crypto
- No cryptographic agility
- No post-compromise security

How To E2EE Cloud Storage

Goals and challenges for E2EE cloud storage

- Ideal properties
  - Cryptographic agility
  - Modularity
  - Basic features: multi-device access, file sharing
  - Advanced features: post-compromise security, forward security
- Challenges
  - Device support → key management
  - Users handle keys, or passwords
  - Key exchange between users
  - Post-compromise and forward security for persistent data
- Malicious storage provider: a strong threat model
  - Today: cryptographic design from a malicious provider!

Looking ahead

- Standardization effort...
  - ...involving various stakeholders
  - ...to design a well-analysed and practical E2EE cloud storage system

- How do we interest providers?
  - Economic incentives: features, integration
  - Political incentives: data privacy laws

Thank you!
Questions?

Paper: "**MEGA**: Malleable Encryption Goes Awry"

Website:
mega-awry.io

Attacks PoC:
github.com/MEGA-Awry

Additional references:

Icons from the Noun Project by: arif fauzi hakim, M Yudi Maulana, alrigel, Oh Rian, rukanicon, Тимур Минвалеев, Ami Ho, juli, Andrew Doane, Eucalyp, Symbolon, Adrien Coquet, Rediffusion