# Cloud Storage Systems

## From Bad Practice to Practical Attacks

Master's thesis of **Miro Haller**
advised by Prof. Dr. Kenny Paterson, Matilda Backendal


# Popular cloud providers

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

# Popular cloud providers lack privacy

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

# E2EE cloud storage: why do we care?

- Cloud Storage
  - Outsource storage
  - Easy file sharing and backup
  - Collaboration
- Without E2EE
  - Cloud provider has direct access to user data

The benefits of outsourced storage,
with the security guarantees of local storage!

# MEGA: the biggest E2EE cloud provider


- Fifth biggest cloud storage service
  - 270+ million accounts
  - 1000+ PB of stored data
- Provide E2EE
  - “MEGA does not have access to your password or your data.” [1]

[1] https://mega.io/security

# Cryptanalysis of MEGA

- Surely, MEGA's system has been well analyzed!
  - Around since 2013
  - Open-source clients
  - Attractive bug bounty program

# Cryptanalysis of MEGA

- Surely, MEGA's system has been well analyzed!
  - Around since 2013
  - Open-source clients
  - Attractive bug bounty program
- Well…
  - 5 attacks with which MEGA can recover:
    - User key material
    - Decrypt all files
    - Inject arbitrary files
  - Feasible in practice and improved by follow-up work [1]

[1] Ryan-Heninger, eprint 2022/914

# How did we break MEGA's cryptography?

- From bad practice...
  - No authenticated encryption
  - Used AES-ECB to encrypt keys
  - Insufficient key separation

# How did we break MEGA's cryptography?

- From bad practice...
  - No authenticated encryption
  - Used AES-ECB to encrypt keys
  - Insufficient key separation

**1 week**

# How did we break MEGA's cryptography?

- From bad practice...
  - No authenticated encryption
  - Used AES-ECB to encrypt keys
  - Insufficient key separation
- ... to broken in practice
  - Tamper with encrypted user key material
  - Observe client behavior to extract information about secret keys
  - Abuse unintended interactions to break confidentiality of other parts

**1 week**

**3 months**

# Thank you!
# Questions?

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

Website:
mega-awry.io

Miro Haller:
mirohaller.com

# References

Sources for user statistics:


- 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
- Mega (2022): https://mega.nz/about

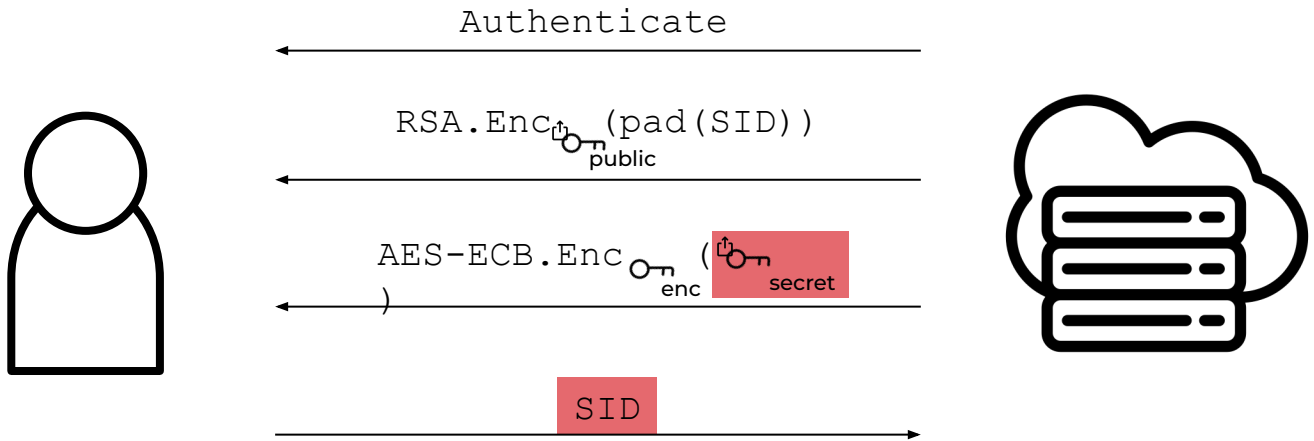
# Backup Slides

# 5 attacks on MEGA

- Attack 1: RSA key recovery
  - Modify the encrypted RSA key sent during authentication
  - Observe client behavior on the garbled key
  - Binary search for prime factor
- Attack 2: file key recovery
  - Insert AES-ECB ciphertext blocks of file key into the known RSA secret key ciphertext
- Attack 3: integrity attack
  - Create an encryption of the zero file key for any user and forge file encryption
- Attack 4: framing attack
  - Like the integrity attack, but for random file keys
- Attack 5: Bleichenbacher
  - Bleichenbacher's RSA decryption attack, adapted to MEGA's custom RSA padding

# Attack 1: RSA key recovery

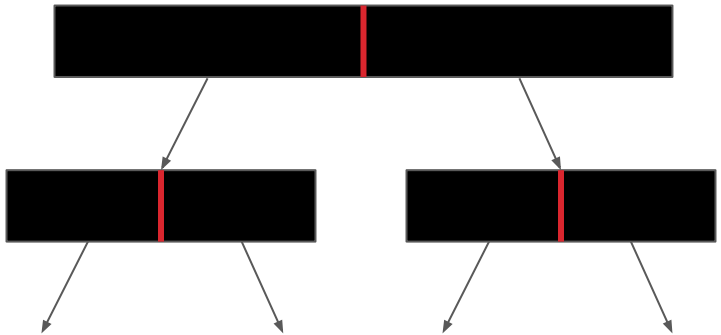
MEGA's user authentication:




- We can modify the key and use the decryption oracle to observe the client's behavior on the garbled key

# Attack 1: RSA key recovery

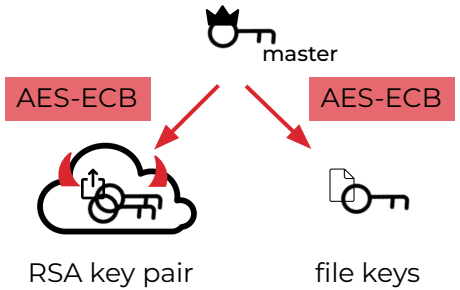
- Construct an oracle that is true if $m < q$ and false if $m \geq q$
- Binary search for q
  - 1023 queries to recover 1024-bit q value
  - 683 queries (512 in theory): lattice-based optimization
  - 6 queries: Ryan-Heninger, eprint 2022/914



# Attack 2: file key recovery

- Recall: the RSA sk is encrypted with AES-ECB
- File keys are also encrypted with AES-ECB

💡 We can insert AES-ECB ciphertext blocks into the known RSA secret key!



# MEGA's Key Hierarchy