The Future of Passwords

A tour of usability, failures, and cryptography

Julia Hesse, Research Staff Member, Foundational Cryptography

NSS-SocialSec 2023

What a wonderful world

Facility access

Mobile payments

Supermarket checkout

Efficiency & usability – This is what makes the sale.

Increasingly powerful digital identities → Overwhelmed users

Authentication tech of today

Biometrics

WHAT YOU ARE

face, fingerprint, iris
heartrate, gait, veins

✚ usability

✚ device independent

▬ trust in TEE & sensor sec

▬ not resettable

▬ immature crypto

Passwords

WHAT YOU KNOW

password, PIN, mnemonic,
pattern, security question

✚ device independent

▬ prone to weak choices

▬ trust in pw manager

▬ immature deployments

Keys/Devices

WHAT YOU OWN

hardware token (YubiKey)
cryptographic keys (FIDO)

✚ strong security

▬ device dependent

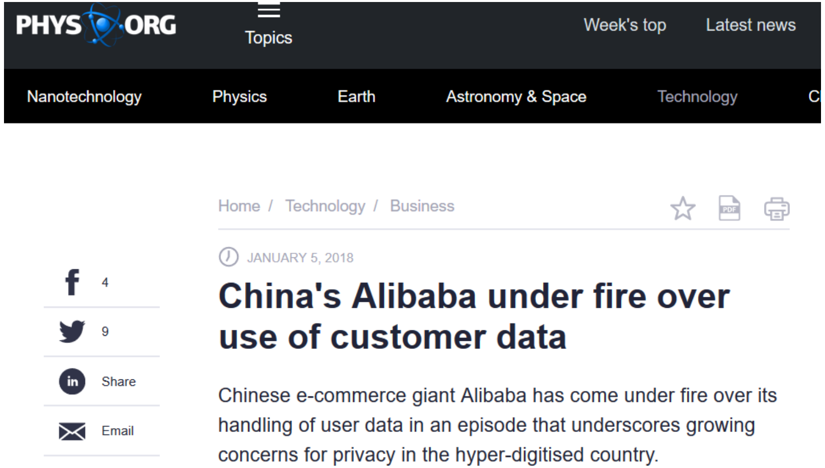
▬ trust in TEE

▬ prone to loss

Some predictions

1 In the next 5-10 years, most of our biometric data will get stolen through breaches.

- Massive databases containing information that uniquely identifies all individuals
- More and more providers will roll back to resettable authentication means

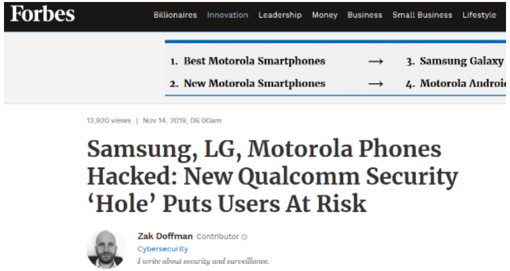


Home / Technology / Business

JANUARY 5, 2018

China's Alibaba under fire over use of customer data

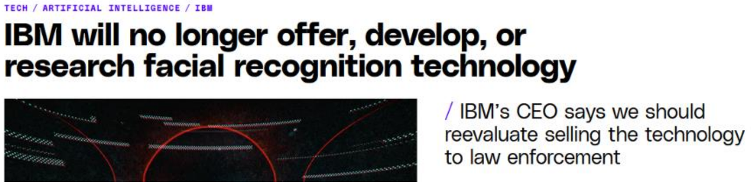
Chinese e-commerce giant Alibaba has come under fire over its handling of user data in an episode that underscores growing concerns for privacy in the hyper-digitised country.



13,920 views | Nov 14, 2019, 06:00am

Samsung, LG, Motorola Phones Hacked: New Qualcomm Security 'Hole' Puts Users At Risk

Zak Doffman Contributor
Cybersecurity
I write about security and surveillance.



TECH / ARTIFICIAL INTELLIGENCE / IBM

IBM will no longer offer, develop, or research facial recognition technology

/ IBM's CEO says we should reevaluate selling the technology to law enforcement

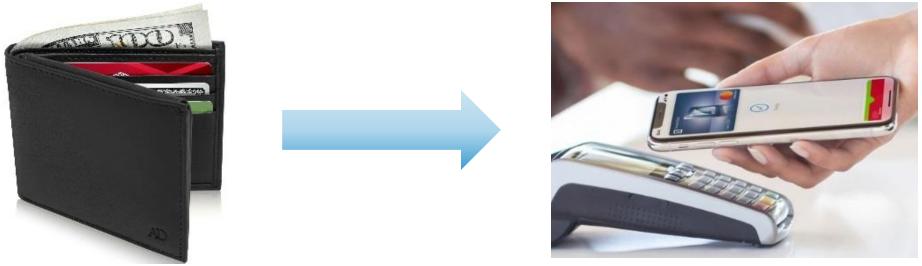
Some predictions

1 In the next 5-10 years, most of our biometric data will get stolen through breaches.

- Massive databases containing information that uniquely identifies all individuals
- More and more providers will roll back to resettable authentication means

2 We will always need a device-independent authentication method

- Otherwise, phone access = bank access & wallet access & social media access &…



Some predictions

1 In the next 5-10 years, most of our biometric data will get stolen through breaches.

- Massive databases containing information that uniquely identifies all individuals
- More and more providers will roll back to resettable authentication means

2 We will always need a device-independent authentication method

- Phone access = bank access & wallet access & social media access &…

3 Passwords will not disappear any time soon.

$$1 \wedge 2 \rightarrow 3$$

Password Authentication

The good, the bad, the ugly

Passwords – Some stats

PASSWORD

The most common password

70

Average number of passwords per citizen, Western Europe

30%

of US citizens use the same password for all their accounts

99.000.000.000.000.000.000.000 yrs

Time to crack a 12 random digit/word/character password

Passwords – Main issues

- Tempting to choose weak passwords or to reuse passwords

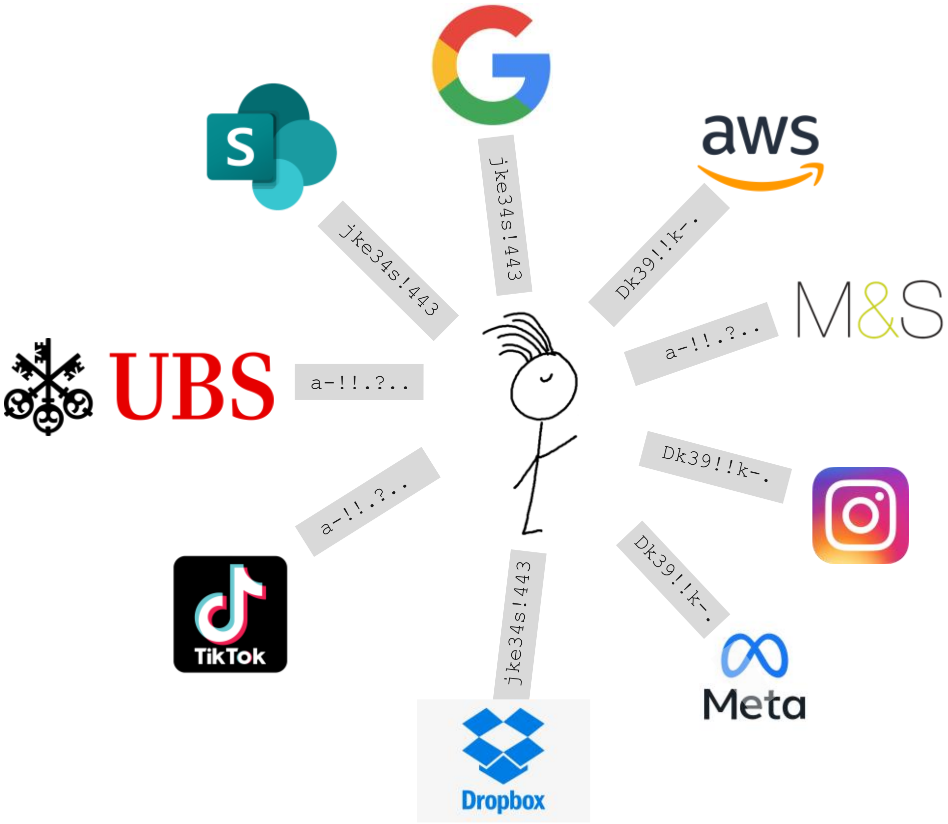
- Secure password for each account

  = Trust in password managers

- Insecure deployments of password authentication

focus of this talk

Under the hood: ~~cryptography~~

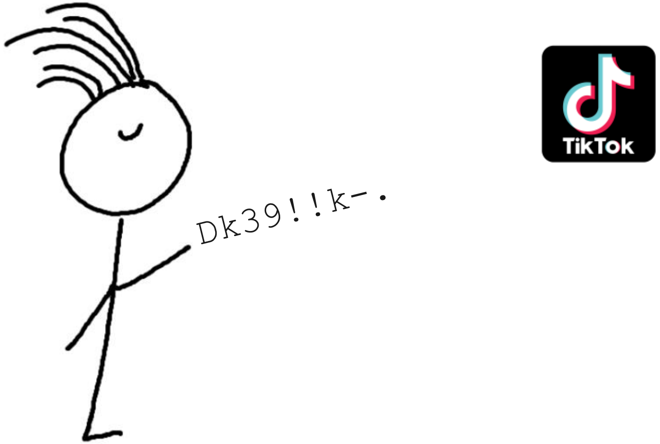


1. Establish a TLS connection to your provider

2. Send your cleartext password to the provider

3. Provider hashes and compares against his database of hashed passwords

The absence of cryptography



Dk39!!k-.

The absence of cryptography

UBS
Dk39!!k-.
TikTok

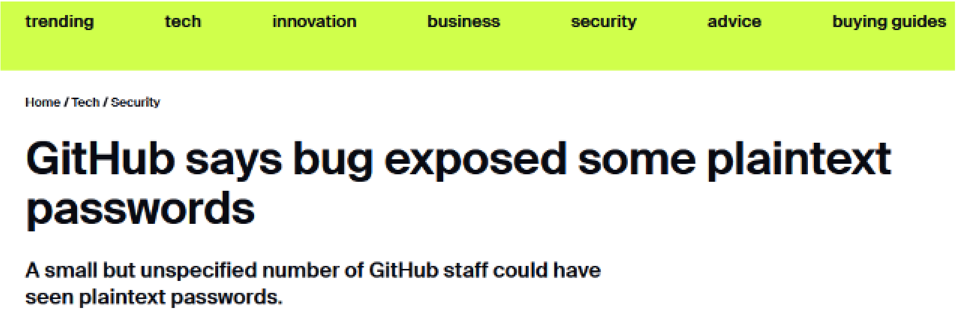
Ok, I think I just told my bank password to the guys at TikTok...

Cleartext password handling

- “Password-over-TLS” puts **full trust** in our providers’
  - Implementations, hardware, admins
  - Security measures against hackers, physical site protection
  - Contractors, third-party software

ONE of them fails: password leaked

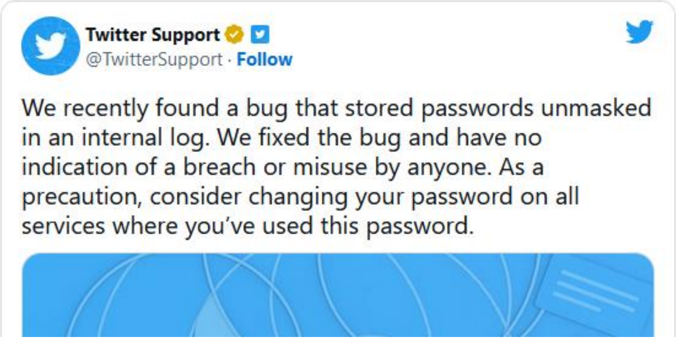
Cleartext password handling – A selection of breaches



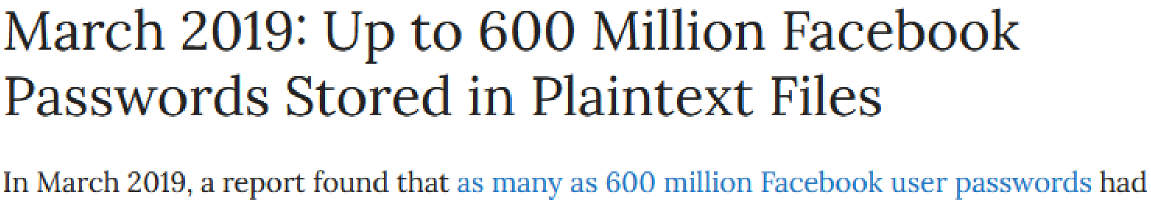
Home / Tech / Security

GitHub says bug exposed some plaintext passwords

A small but unspecified number of GitHub staff could have seen plaintext passwords.

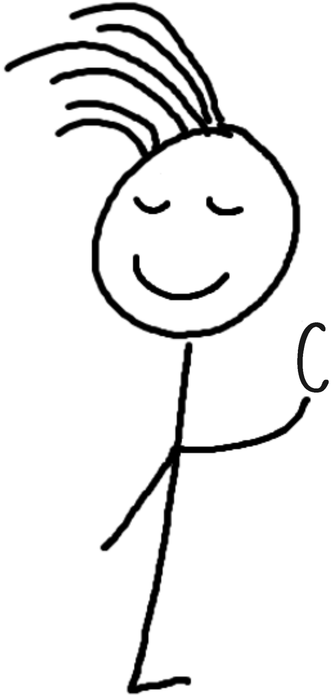


We recently found a bug that stored passwords unmasked in an internal log. We fixed the bug and have no indication of a breach or misuse by anyone. As a precaution, consider changing your password on all services where you've used this password.



March 2019: Up to 600 Million Facebook Passwords Stored in Plaintext Files

In March 2019, a report found that as many as 600 million Facebook user passwords had



CRYPTOGRAPHY TO THE RESCUE!

Cryptographic tools to stop leaking passwords

- Zero Knowledge Proofs
  - Prove knowledge of preimage of a hash without revealing preimage

Cryptographic tools to stop leaking passwords

- Zero Knowledge Proofs
  - Prove knowledge of preimage of a hash without revealing preimage

- Multi-Party Computation
  - Perform hashing of password and comparison with database without revealing the inputs to the computation

Cryptographic tools to stop leaking passwords

- Zero Knowledge Proofs
  - Prove knowledge of preimage of a hash without revealing preimage

- Multi-Party Computation
  - Perform hashing of password and comparison with database without revealing the inputs to the computation

around since 30+ years

- Password-based cryptography (e.g., key exchange from passwords, password-protected secret sharing)

as fast as a classical (DH) key exchange

So why do passwords still leak?

- Cleartext password transmission is tradition and deployed everywhere – hard to change
- Patent issues
- Cryptographers often do not communicate their findings well
- Missing specifications
- Only few (good) open-source implementations

Password authentication, built right.

A progress report

Quick guide to secure password authentication

- Never store passwords unprotected on servers ✓
- Only the user device sees/handles/caches the user’s cleartext password ✗
- Provide brute-force protection whenever users can choose weak passwords ✗

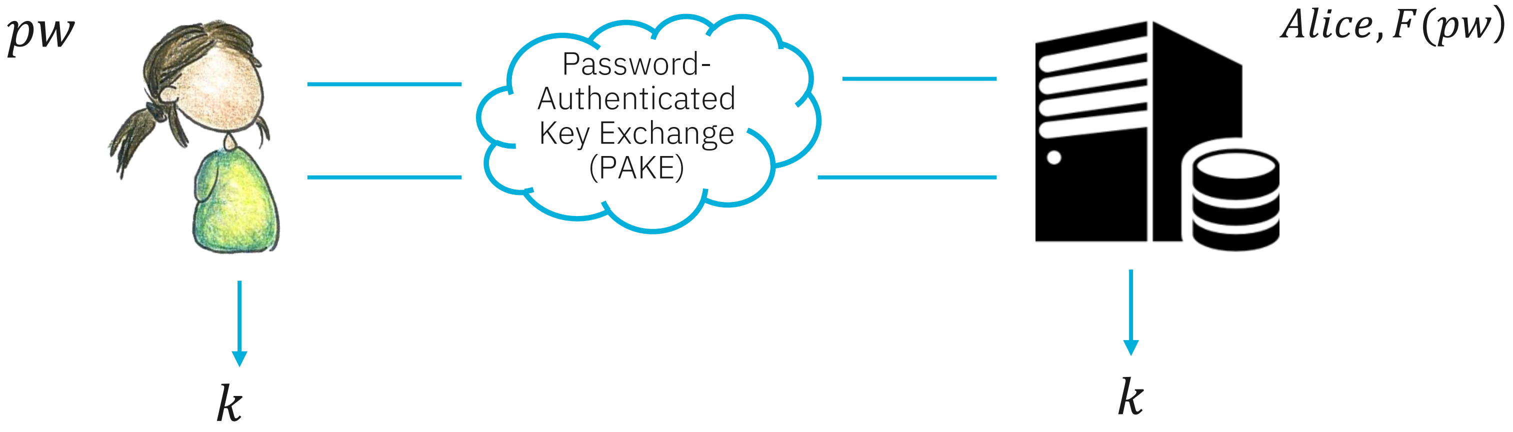
Upcoming topics

- A peek into a cryptographer’s toolbox
  - Password-authenticated key exchange (PAKE)
  - Oblivious pseudorandom functions (OPRFs)
- TLS-OPAQUE - The password button for TLS
- WhatsApp’s E2EE chat backup protocol

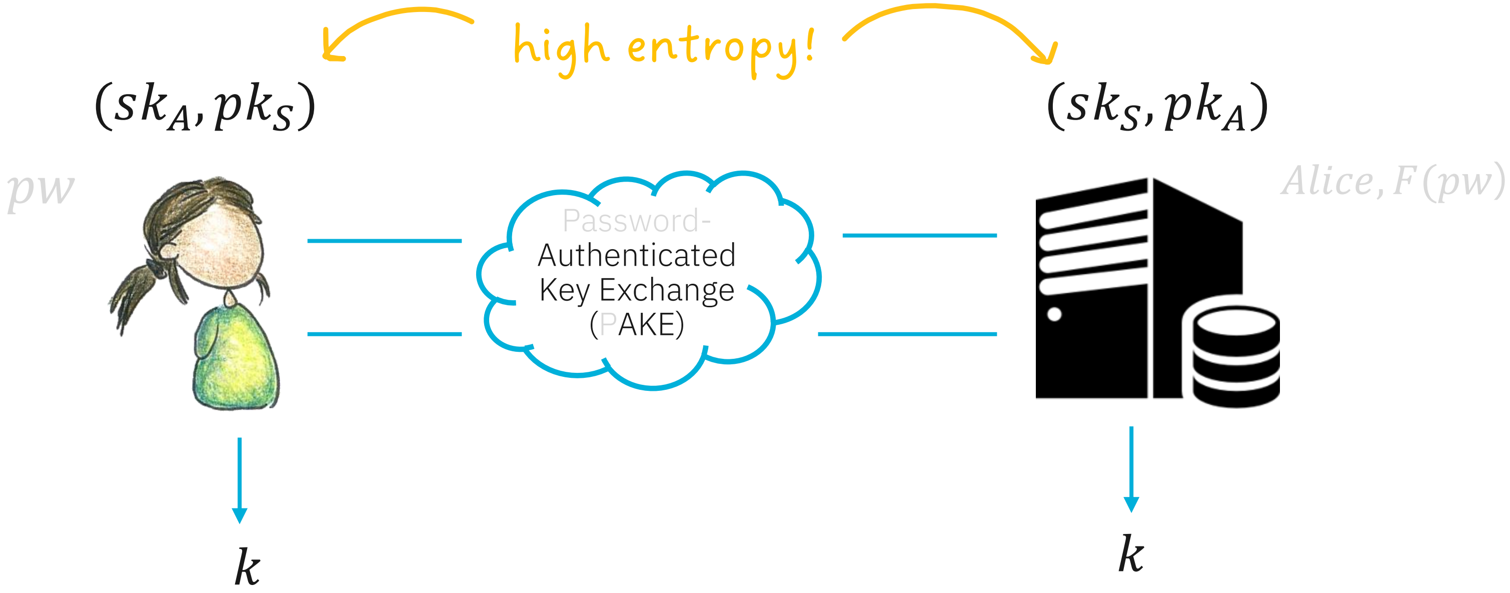
Password-authenticated key exchange

From shared passwords to shared keys

Password-Authenticated Key Exchange (PAKE)

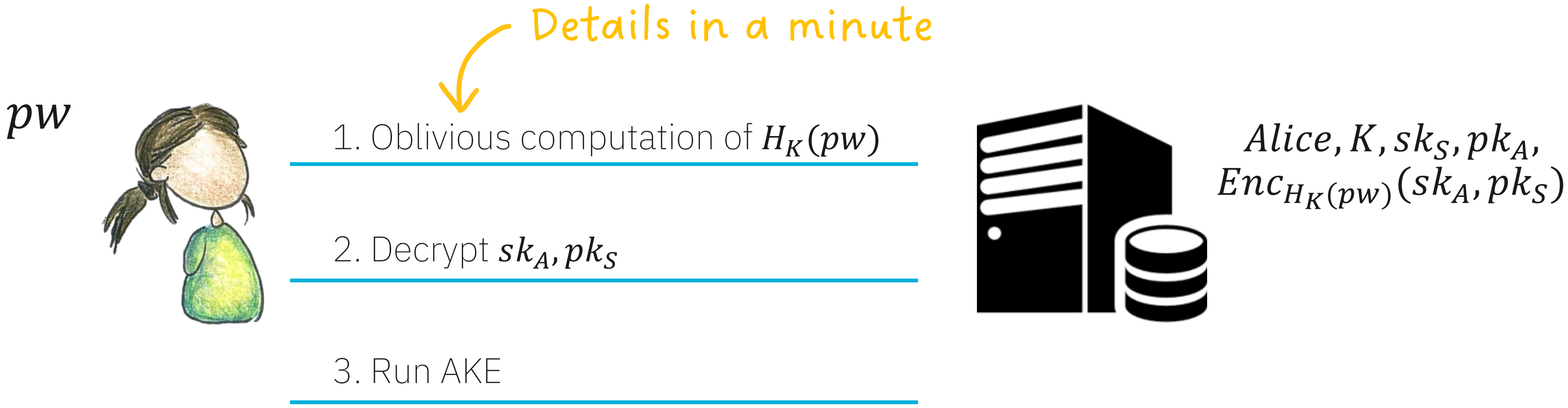


Password-Authenticated Key Exchange (PAKE)



e.g., Diffie-Hellman, HMQV, TLS 1.3 Handshake

Password-Authenticated Key Exchange (PAKE)



Details in a minute

pw

1. Oblivious computation of $H_K(pw)$

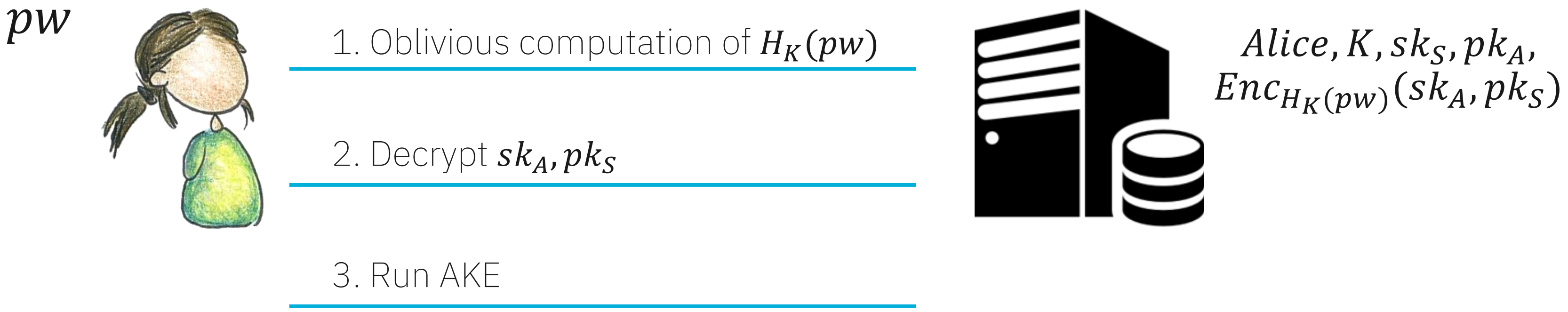
2. Decrypt sk_A, pk_S

3. Run AKE

$Alice, K, sk_S, pk_A,$
$Enc_{H_K(pw)}(sk_A, pk_S)$

High level idea: store Alice's pw-encrypted AKE keys on server

Password-Authenticated Key Exchange (PAKE)



pw

1. Oblivious computation of $H_K(pw)$

2. Decrypt sk_A, pk_S

3. Run AKE

$Alice, K, sk_S, pk_A,$
$Enc_{H_K(pw)}(sk_A, pk_S)$

OPAQUE (Jarecki et al, Eurocrypt 2018)

Password-Authenticated Key Exchange (PAKE)

- PAKE allows to turn shared passwords into shared keys
- Immediately yields password authentication: just add key confirmation
  - Server does not see pw in the clear
  - Client cannot run brute-force dictionary attack
- 2-3 move protocols, speed of 1-3x DH key exchange
- Patent on DH-style PAKEs - https://patentimages.storage.googleapis.com/63/1f/fc/24e3941e5b6c8d/EP1248408A2.pdf

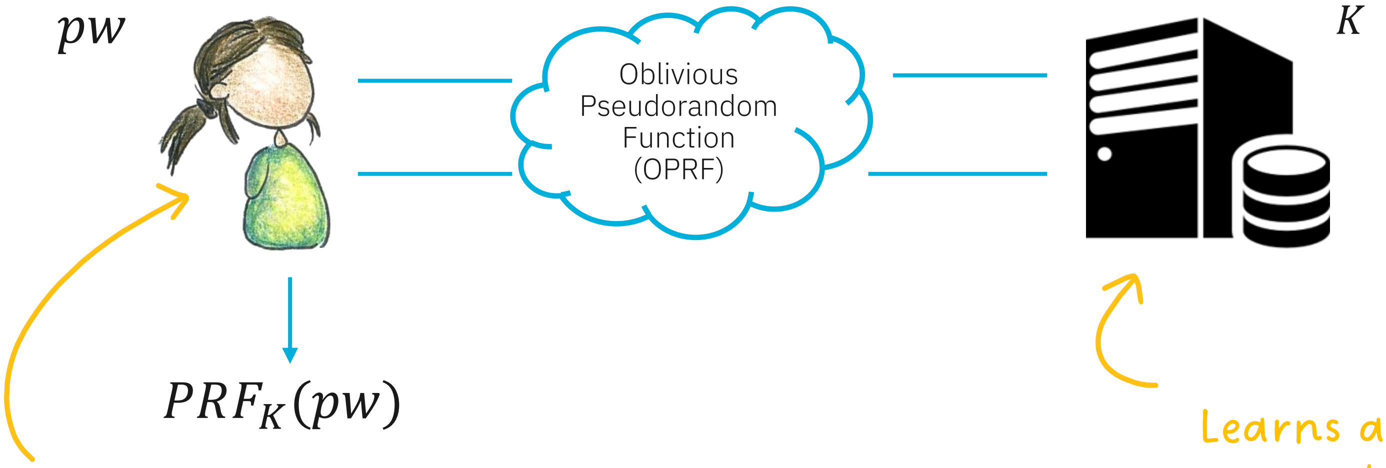
30+ yrs of research

SoK: Password-Authenticated Key Exchange [HO22] https://ia.cr/2021/1492

Oblivious pseudorandom functions

Putting a rate limit on password hashing

Oblivious pseudorandom function (OPRF)



pw

K

Oblivious Pseudorandom Function (OPRF)

$PRF_K(pw)$

Learns absolutely nothing

Does not learn anything else about K

Using OPRFs for password protection

SoK: Oblivious Pseudorandom Functions [CHL22]
https://ia.cr/2022/302

15+ yrs of research

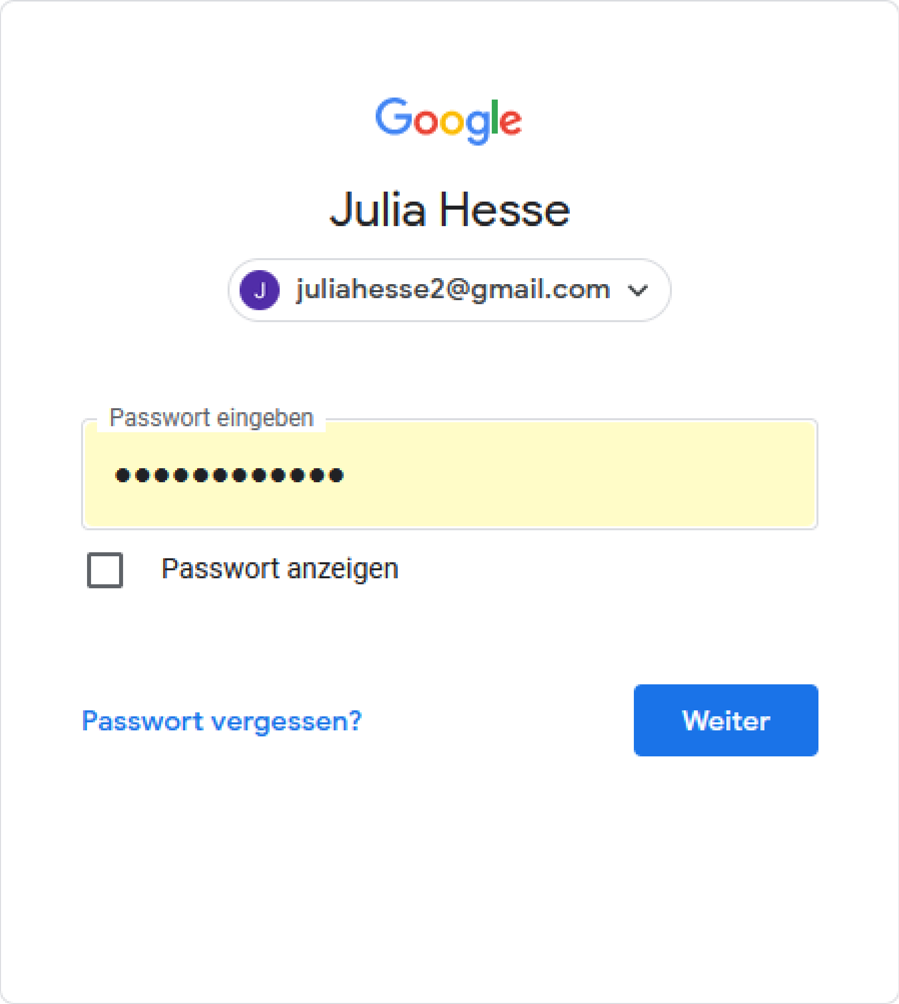
- OPRF = 2-party computation of keyed hash function
- Server holds the PRF key
  - Server can **rate-limit password hashing**
  - Brute-force dictionary attack requires the PRF key
- PRF keys are per user
  - PRF key is essentially a secret hash seed
  - Prevents **precomputation attacks**, e.g., in OPAQUE

Ready?

Let's use these tools to protect our passwords!

TLS-OPAQUE

The password button on TLS channels



Google

Julia Hesse

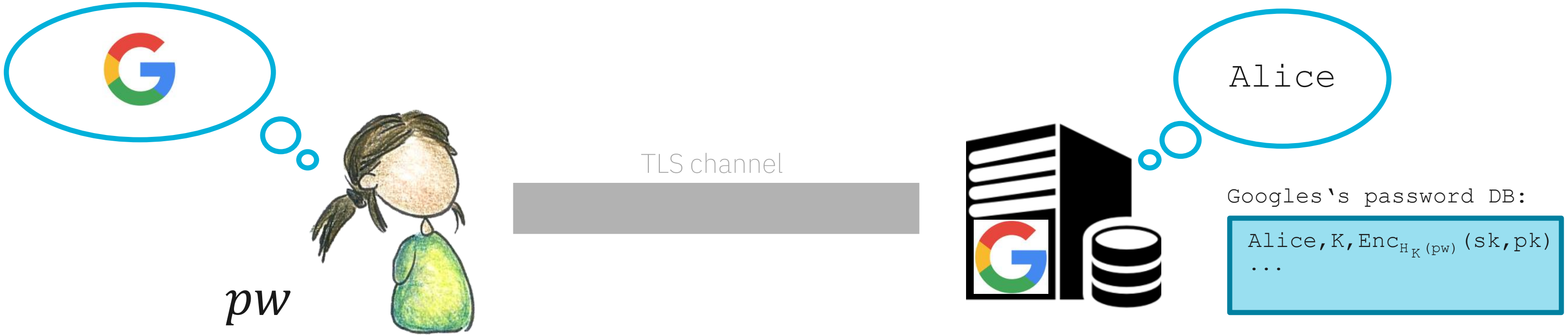
juliahesse2@gmail.com

Passwort eingeben

●●●●●●●●●●●●

□ Passwort anzeigen

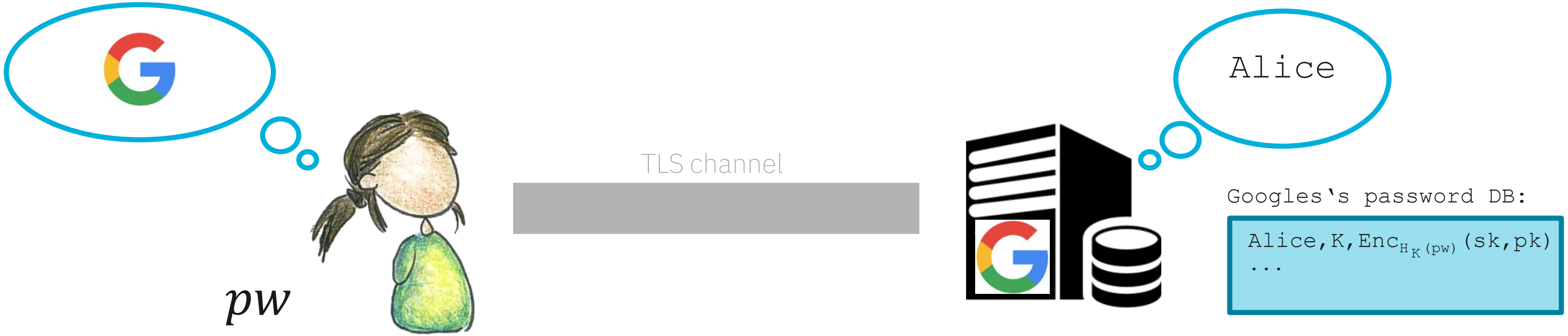
TLS-OPAQUE



TLS-OPAQUE offers post-handshake password authentication

IETF draft: https://datatracker.ietf.org/doc/html/draft-sullivan-tls-opaque-01

Post-handshake password authentication



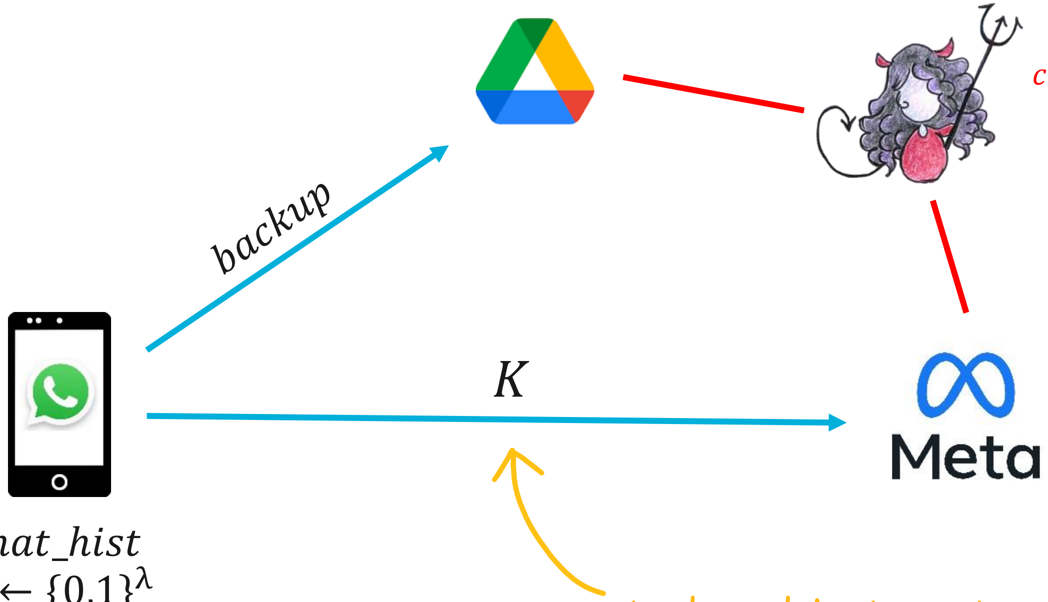
- Uses OPAQUE's encrypted AKE keys (seen before)
- Uses 2HashDH OPRF (seen before) to rate-limit password hashing
  - One password guess per active attack
- Google **never sees or computes with cleartext password** of Alice

Password-Authenticated TLS via OPAQUE and Post-Handshake Authentication [HJKW23] https://ia.cr/2023/220

WhatsApp’s E2EE chat backups

…or why a subpoena against Mark Zuckerberg is useless these days

Chat history backups before 2021



Google Drive

No E2EE if Google & Meta collude:

$chat\_hist \leftarrow Dec_K(backup)$

$backup$

K

My phone

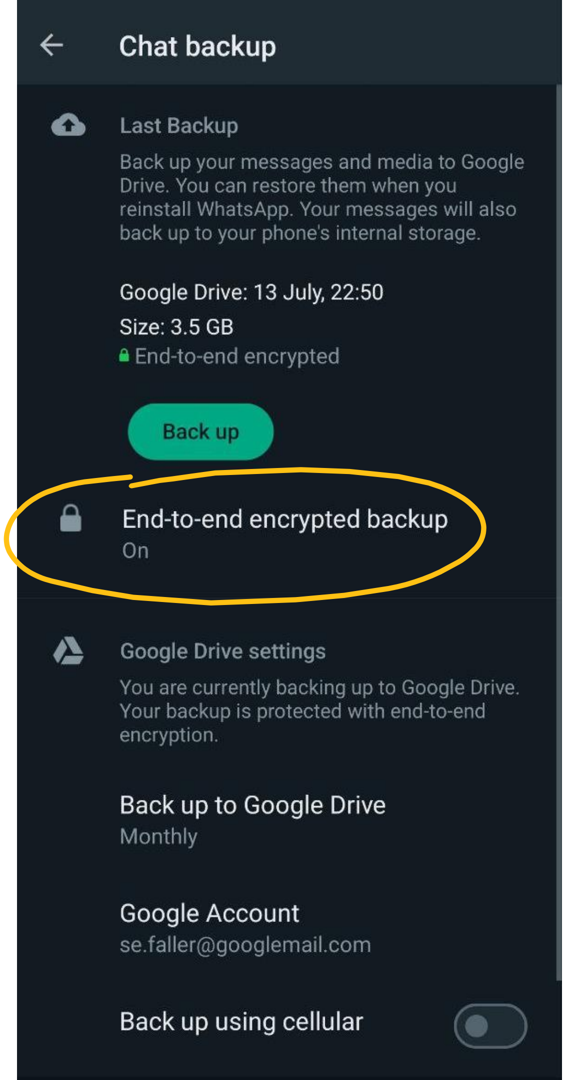
That company owning WhatsApp

$chat\_hist$

$K \leftarrow \{0,1\}^{\lambda}$

$backup \leftarrow Enc_K(chat\_hist)$

to be able to retrieve K in case I lose my phone



Last Backup

Back up your messages and media to Google Drive. You can restore them when you reinstall WhatsApp. Your messages will also back up to your phone's internal storage.

Google Drive: 13 July, 22:50

Size: 3.5 GB

End-to-end encrypted

End-to-end encrypted backup
On

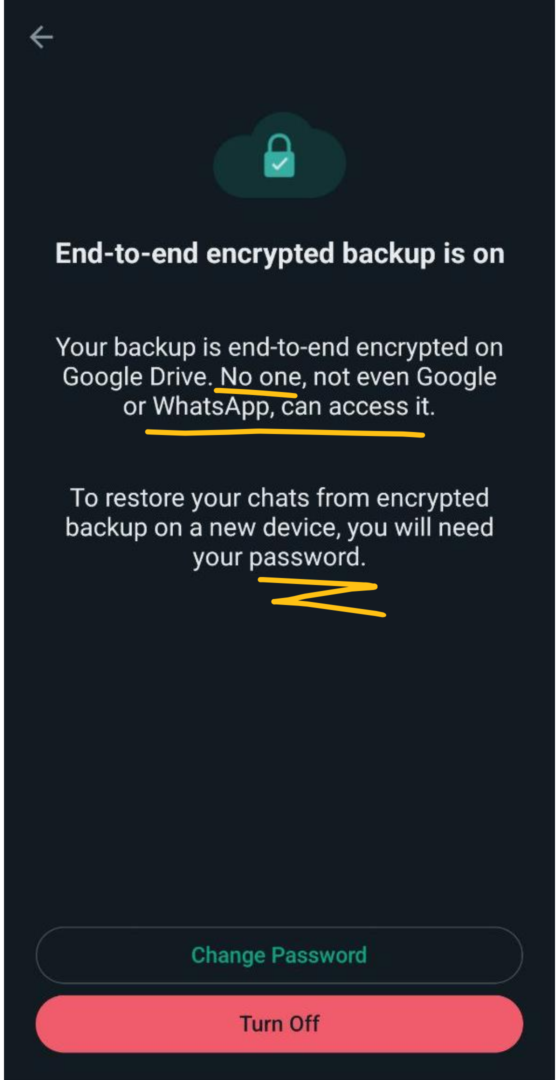
Google Drive settings

You are currently backing up to Google Drive. Your backup is protected with end-to-end encryption.

Back up to Google Drive
Monthly

Google Account
se.faller@googlemail.com

Back up using cellular ○

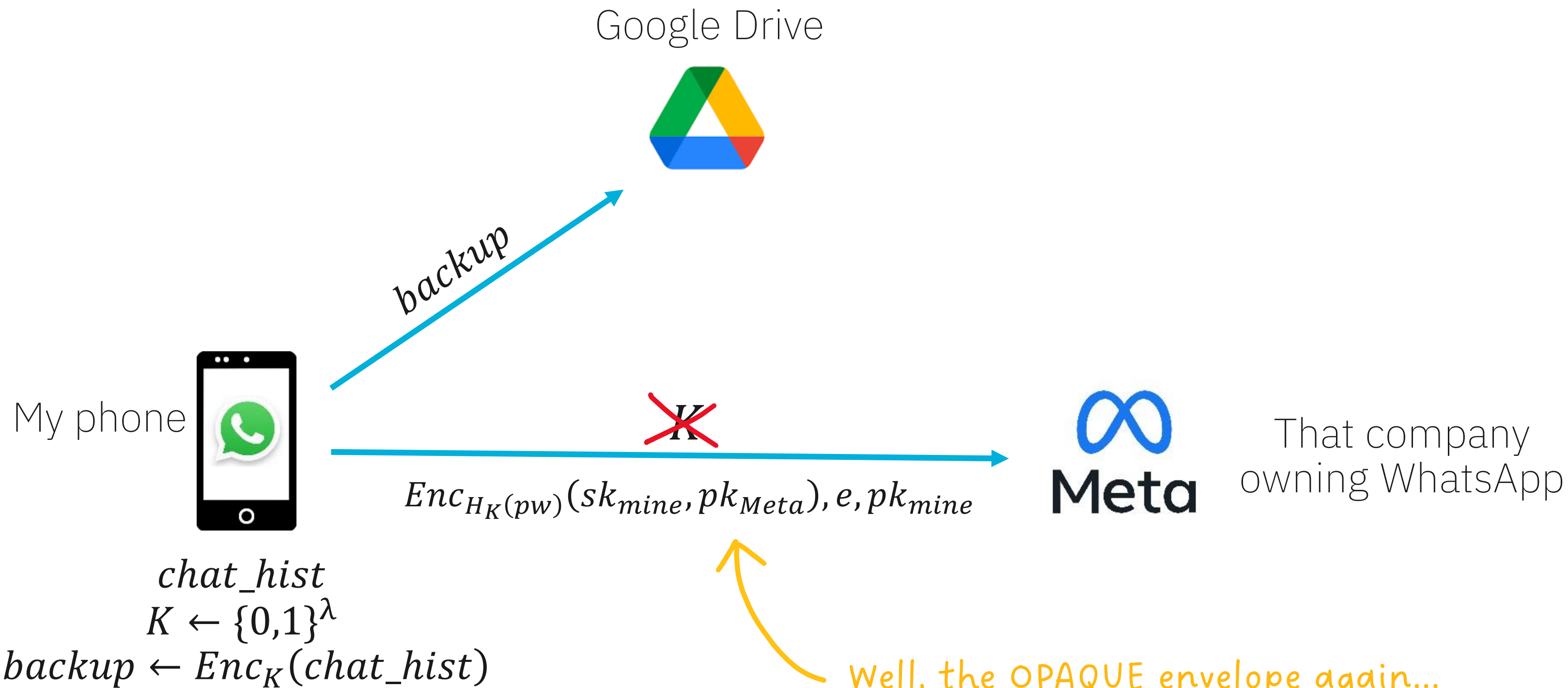


End-to-end encrypted backup is on

Your backup is end-to-end encrypted on Google Drive. No one, not even Google or WhatsApp, can access it.

To restore your chats from encrypted backup on a new device, you will need your password.

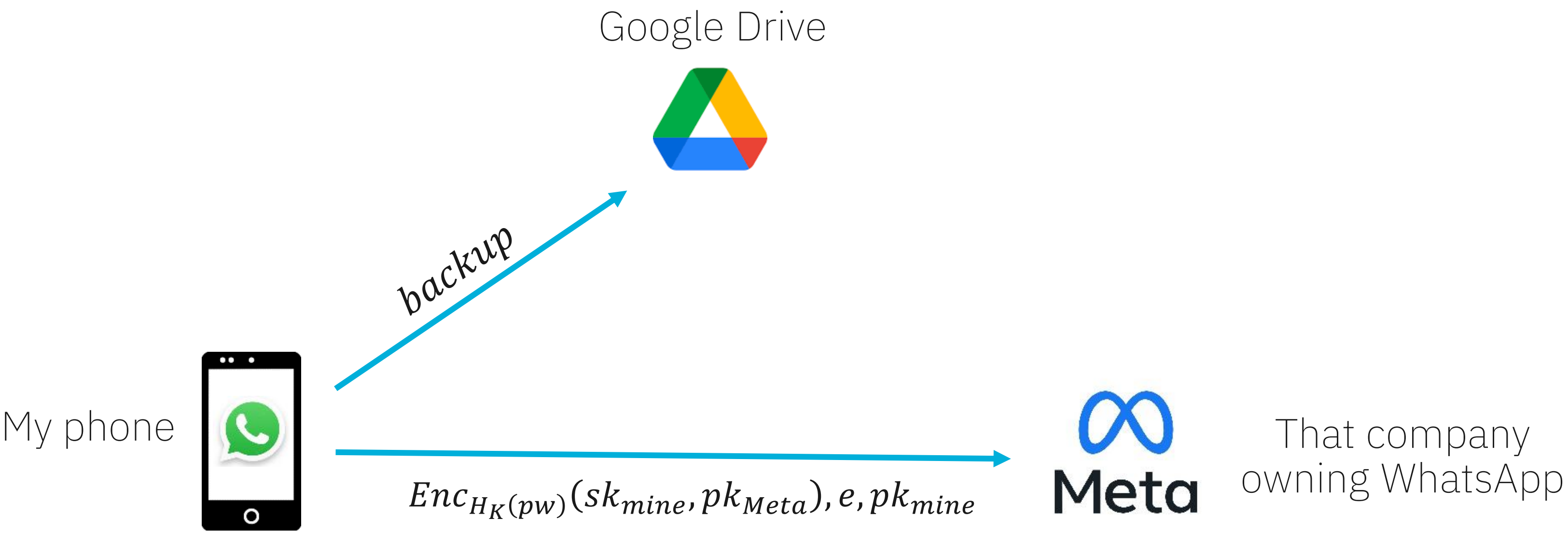
2021: E2EE chat backups in WhatsApp


Google Drive
$backup$
My phone
$\cancel{K}$
$Enc_{H_K(pw)}(sk_{mine}, pk_{Meta}), e, pk_{mine}$
Meta
That company owning WhatsApp

$chat\_hist$
$K \leftarrow \{0,1\}^{\lambda}$
$backup \leftarrow Enc_K(chat\_hist)$
$e \leftarrow Enc_{H_{K_{oprf}}(pw)}(K)$

Well, the OPAQUE envelope again...

2021: E2EE chat backups in WhatsApp



Google Drive

$backup$

My phone

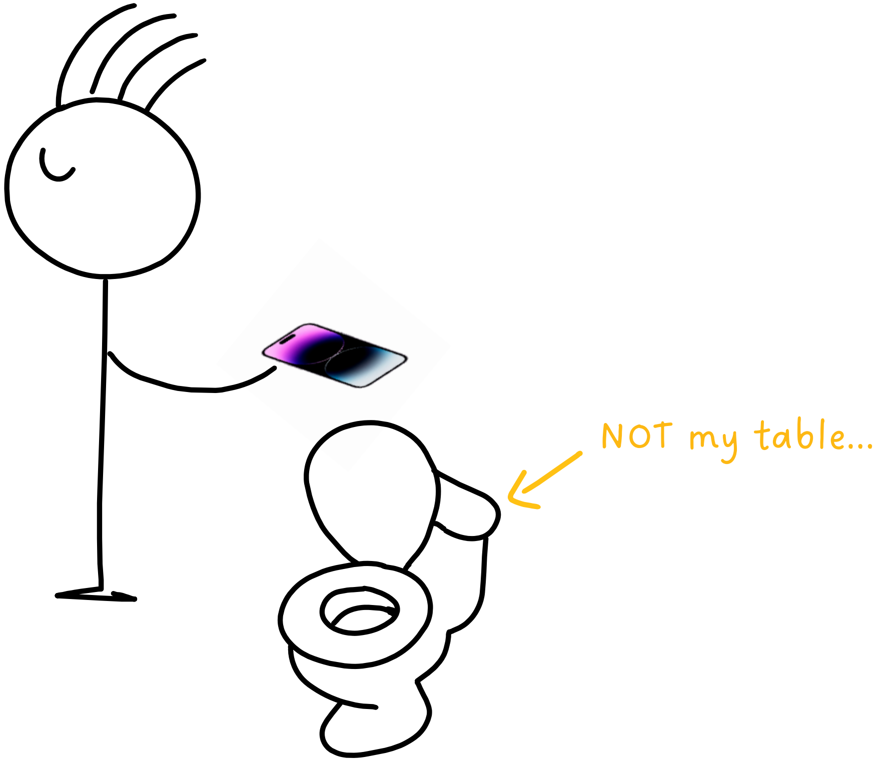
$Enc_{H_K(pw)}(sk_{mine}, pk_{Meta}), e, pk_{mine}$

That company owning WhatsApp

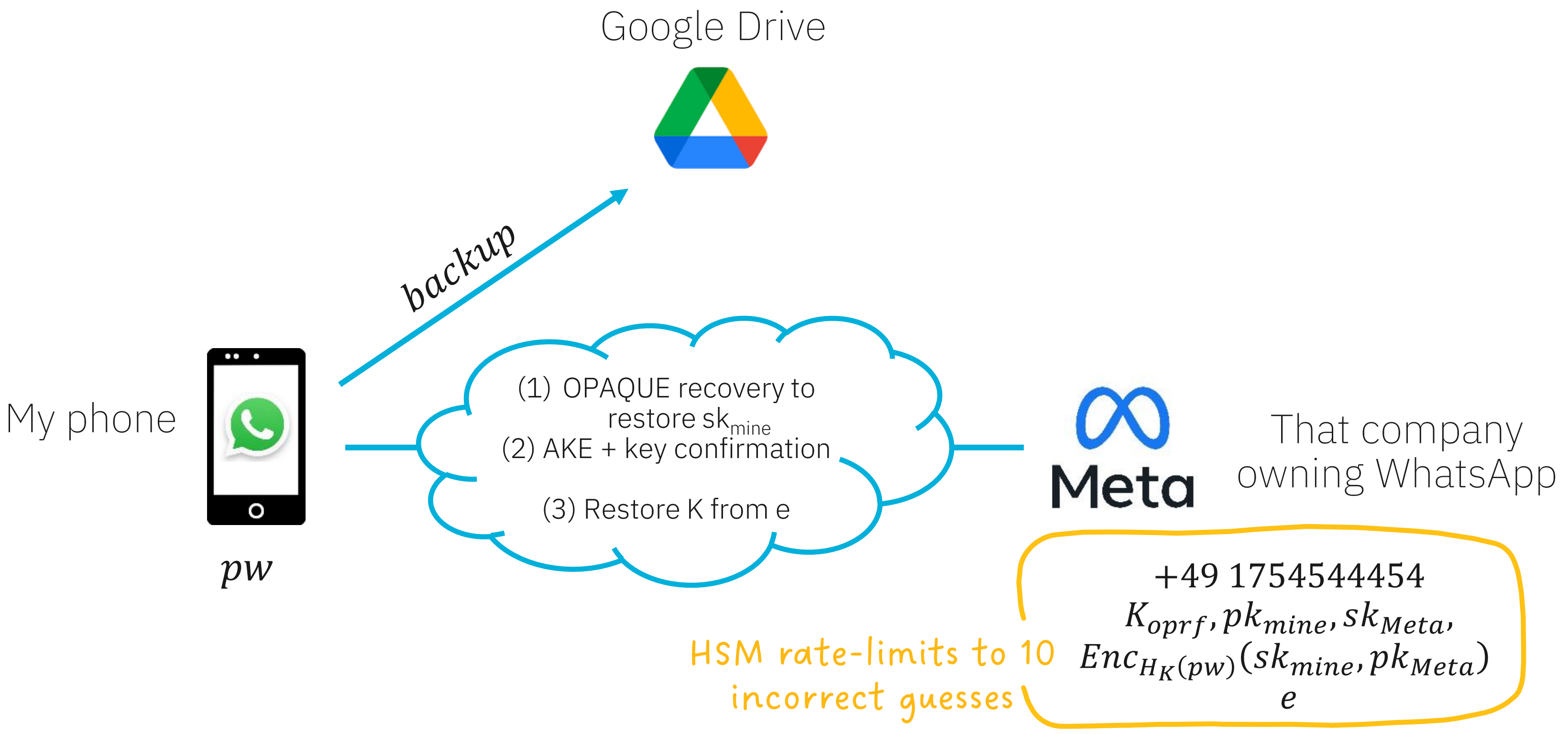
$chat\_hist$
$K \leftarrow \{0,1\}^{\lambda}$
$backup \leftarrow Enc_K(chat\_hist)$
$e \leftarrow Enc_{H_{K_{oprf}}(pw)}(K)$

+49 1754544454
$K_{oprf}, pk_{mine}, sk_{Meta},$
$Enc_{H_K(pw)}(sk_{mine}, pk_{Meta})$
e

Assume disaster happens



2021: E2EE chat backups in WhatsApp



Google Drive

$backup$

My phone

pw

(1) OPAQUE recovery to restore sk$_{mine}$

(2) AKE + key confirmation

(3) Restore K from e

Meta

That company owning WhatsApp

+49 1754544454

$K_{oprf}, pk_{mine}, sk_{Meta},$

$Enc_{H_K(pw)}(sk_{mine}, pk_{Meta})$

e

HSM rate-limits to 10 incorrect guesses

Security Analysis of the WhatsApp End-to-End-Encrypted Backup Protocool [DFGHHHJ23] https://ia.cr/2023/843

All you need to know on one slide

Passwords are going to be around for a while

We have the cryptographic tools to protect them from breaches

- Matt Greene’s blogpost on PAKE
  https://blog.cryptographyengineering.com/2018/10/19/lets-talk-about-pake/
- Meta’s OPRF and OPAQUE implementations
  https://github.com/facebook/voprf https://github.com/facebook/opaque-ke
- Get involved in writing specs
  https://www.irtf.org/mailman/listinfo/cfrg
- Want a challenge? Nothing yet post-quantum...

drawn by Giorgia Azzurra Marson