DS 593: Privacy in Practice

Systems for Privacy

Instructor: Alishah Chator

News?

Password reuse is rampant: nearly half of observed user logins are compromised

2025-03-17


Radwa Radwan


Sabina Zejnilovic

5 min read

https://blog.cloudflare.com/password-reuse-rampant-half-user-logins-compromised/

Last time

- Introduction to Cryptography
- Differential Privacy
- Other Mechanisms for Privacy

Today

- How do we use these building blocks in real systems?

What exactly do PETS buy us?

- We started this course thinking about legal and social mechanisms for privacy
  - What do our technical tools give us in the best case?
  - In the worst case?

What exactly do PETS buy us?

- We started this course thinking about legal and social mechanisms for privacy
  - What do our technical tools give us in the best case?
  - In the worst case?
- **It’s about codifying trust and reducing attack surface**

Privacy Concerns?

Systems for Authentication

- One of the most fundamental problems is how to control access to our devices
- Generally speaking, once someone has access to a device all bets are off
- Many different approaches, some better than others

Password-based Authentication

- Leverages a user selected word or phrase for authentication
  - Pro: often human understandable
- Checks to see if the provided password matches
- How do we determine the security of the system?
- How might we build this?

Most common passwords 2022

1. password
4,929,113 uses

2. 123456
1,523,537

3. 123456789
413,056

4. guest
376,417

5. qwerty
309,679

6. 12345678
284,946

7. 111111
229,047

8. 12345
188,602

9. col123456
140,505

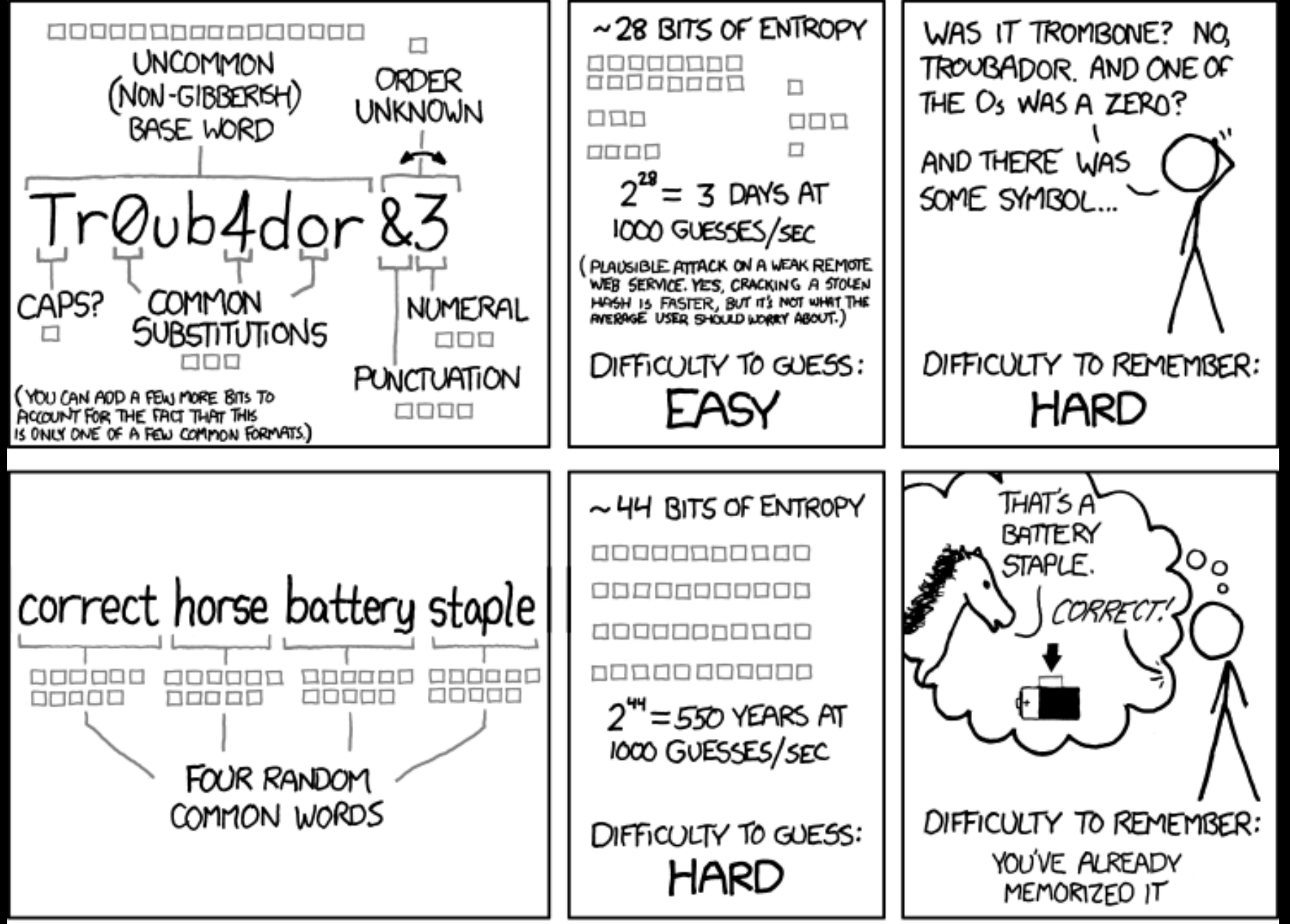
10. 123123
127,762

TIME IT TAKES A HACKER TO BRUTE FORCE YOUR PASSWORD IN 2023

Number of Characters	Numbers Only	Lowercase Letters	Upper and Lowercase Letters	Numbers, Upper and Lowercase Letters	Numbers, Upper and Lowercase Letters, Symbols
4	Instantly	Instantly	Instantly	Instantly	Instantly
5	Instantly	Instantly	Instantly	Instantly	Instantly
6	Instantly	Instantly	Instantly	Instantly	Instantly
7	Instantly	Instantly	1 sec	2 secs	4 secs
8	Instantly	Instantly	28 secs	2 mins	5 mins
9	Instantly	3 secs	24 mins	2 hours	6 hours
10	Instantly	1 min	21 hours	5 days	2 weeks
11	Instantly	32 mins	1 month	10 months	3 years
12	1 sec	14 hours	6 years	53 years	226 years
13	5 secs	2 weeks	332 years	3k years	15k years
14	52 secs	1 year	17k years	202k years	1m years
15	9 mins	27 years	898k years	12m years	77m years
16	1 hour	713 years	46m years	779m years	5bn years
17	14 hours	18k years	2bn years	48bn years	380bn years
18	6 days	481k years	126bn years	2tn years	26tn years

HIVE SYSTEMS

› Learn how we made this table at hivesystems.io/password



THROUGH 20 YEARS OF EFFORT, WE'VE SUCCESSFULLY TRAINED EVERYONE TO USE PASSWORDS THAT ARE HARD FOR HUMANS TO REMEMBER, BUT EASY FOR COMPUTERS TO GUESS.

Building Password Systems

- Simplest (Worst approach, unfortunately has been done in practice): Save password to file, check if passwords match
- Better: Hash the password and save that, when given a password hash it and then check the hashes
  - Use special purpose hash function like bcrypt
- Even better: Hash the password and a random *salt,* and use that for checking
- Best: Use something stronger than a password (not as user-friendly), or at least use a password manager to generate strong passwords

An Example of Hashing a Salted Password

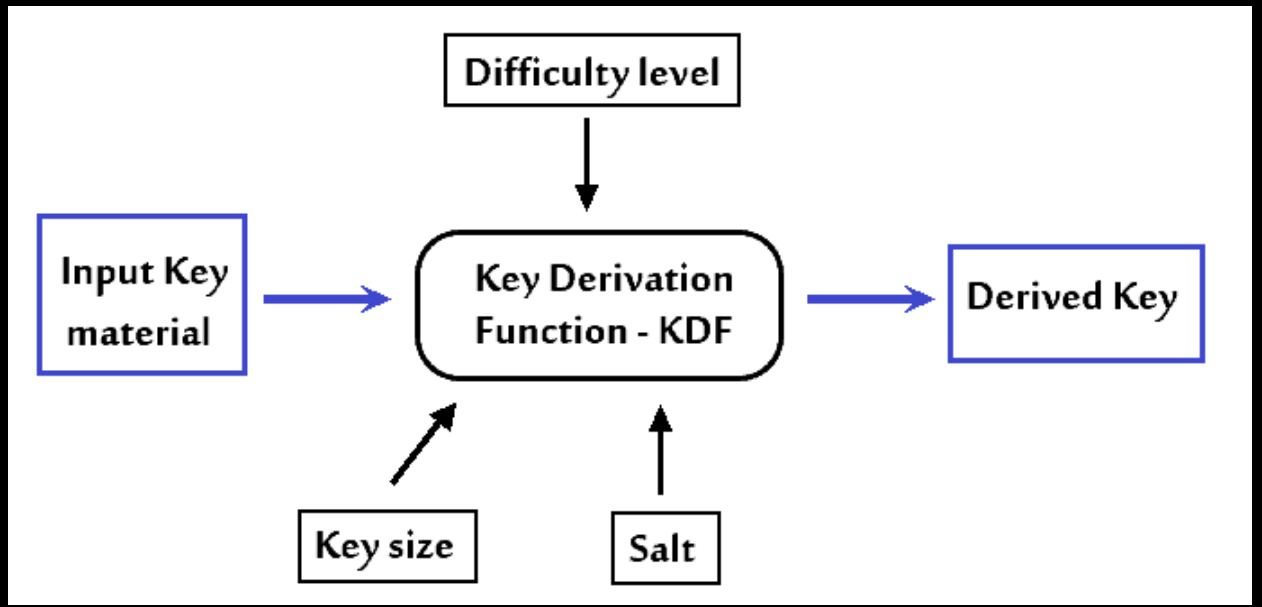
Password (Original Input):	h@ckPr00f
Salt (Unique Integer):	H13qY7YPCDJULY7l
Password + Salt Combo:	h@ckPr00fH13qY7YPCDJULY7l
Password Hash (Output):	$argon2id$v=19$m=16,t=2,p=1$SDEzcVk3WVBDREpVTFk3bA$59cGxSHPjlD4mrJYFxMFUQ

Password Recovery

- Many systems that use passwords offer some mechanism to regain access to a system if the password is lost/forgotten
- By providing some additional information, you can generate a new password
- What does this capability imply?

Password-based Key Derivation Function

- Compromise between the user-friendliness of passwords and the strong guarantees of cryptographic keys
- Generally involves using a slow hash function many times
- Security?

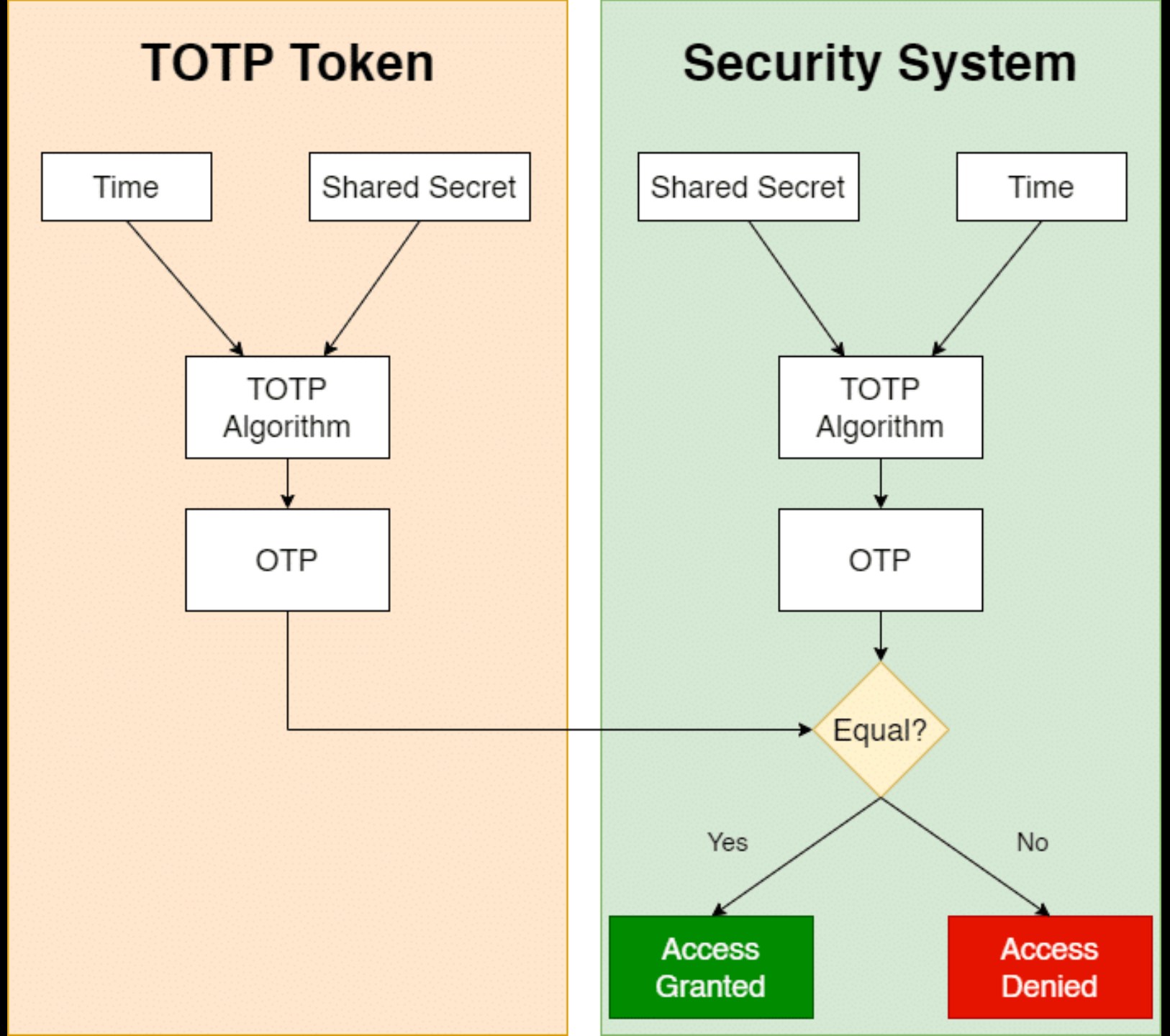


One-time Passcodes (OTP)

- An OTP is a (short) random string intended for a single authentication

- Comes in different flavors
  - HMAC-based OTP
  - Time-based OTP

- Involves a shared secret that is used to generate OTPs and then the OTPs are compared


TOTP Token
Time
Shared Secret
TOTP Algorithm
OTP
Security System
Shared Secret
Time
TOTP Algorithm
OTP
Equal?
Yes
No
Access Granted
Access Denied

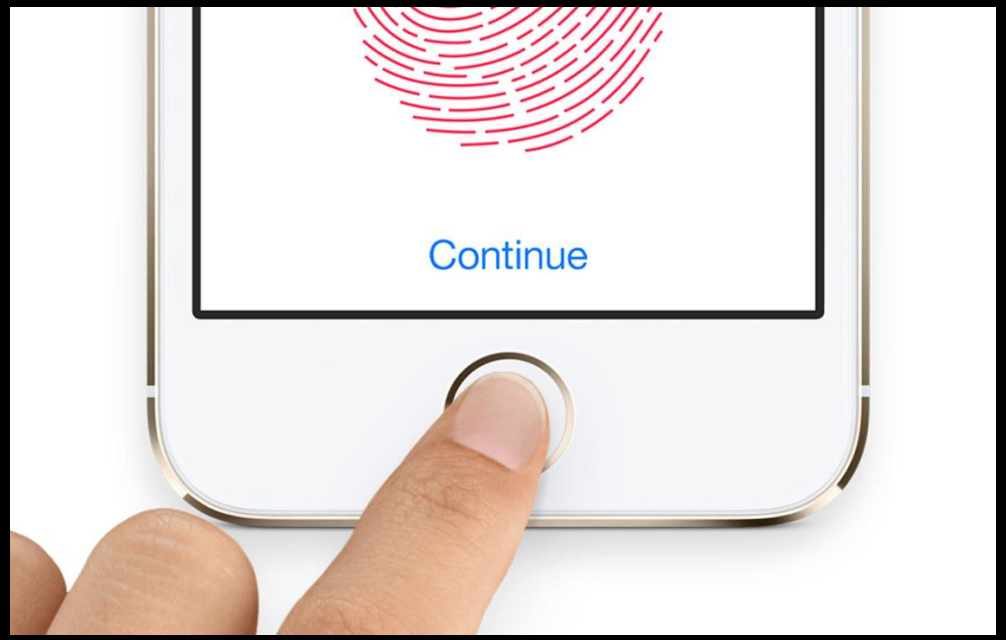
Multi-Factor Authentication (2FA/MFA)

- An MFA system adds robustness by adding additional steps to authentication beyond a password
- Each factor should be using different channels
- Types of factors:
  - Something you know: Passwords, keys, identifying info
  - Something you have: Phone, yubikey, hardware token, smartcard
  - Something you are: Fingerprint, Face, Voice
- Different factors are usually used to produce OTPs

MFA Security Considerations

- Some factors are easier to corrupt
  - Ex: text or calls to your phone
- 2FA recovery is much more complex
- Still vulnerable to social engineering

Biometrics

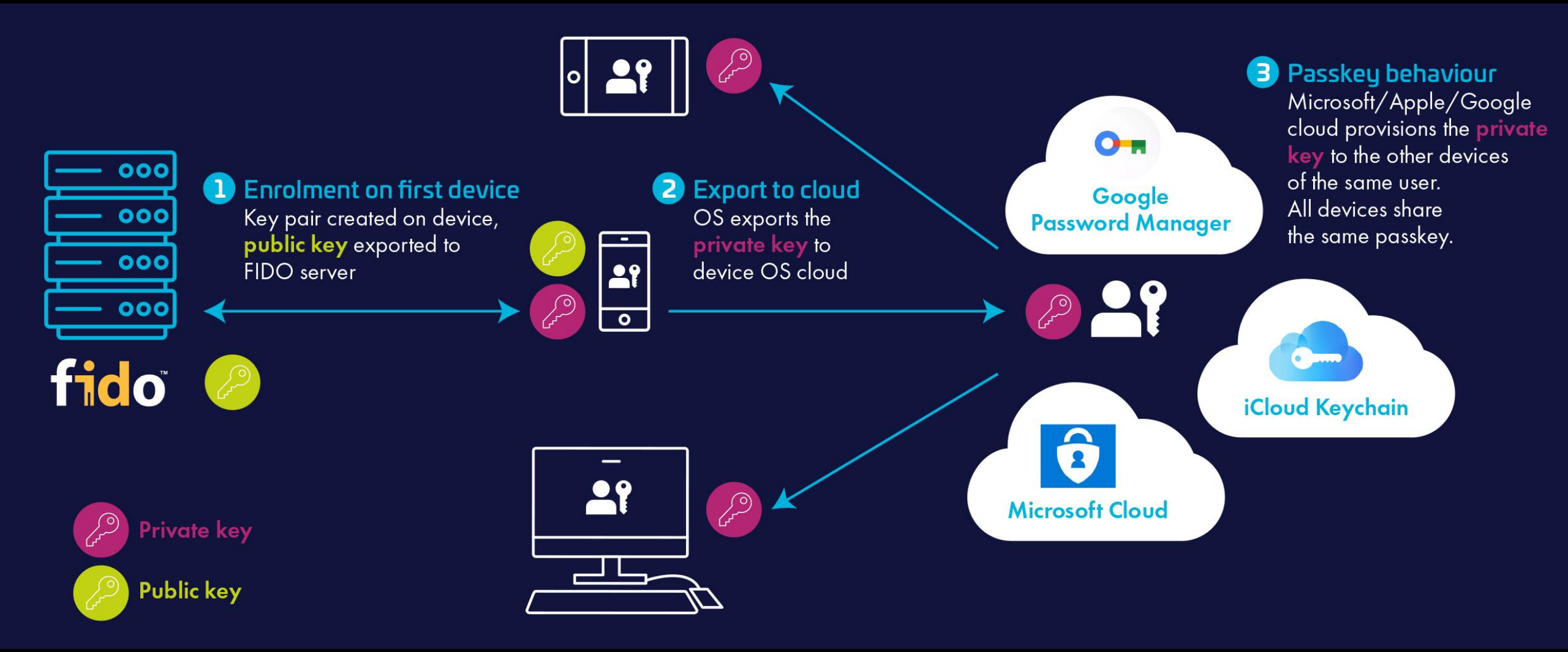


Biometrics

- Biometric authentication in principle is the same as what we have seen
- Takes an input, hashes it, and then compares
- Main difference: the type of hash function
  - What properties might we need?

Passkeys

- Passwords suck, how do we get rid of them?
- Ideally we use cryptographic keys everywhere but how do we store these safely without needing passwords?
- What if we secure the keys with biometrics or other easy to use factors?


1 Enrolment on first device
Key pair created on device, public key exported to FIDO server
2 Export to cloud
OS exports the private key to device OS cloud
3 Passkey behaviour
Microsoft/Apple/Google cloud provisions the private key to the other devices of the same user.
All devices share the same passkey.
Google Password Manager
iCloud Keychain
Microsoft Cloud
fido
Private key
Public key

Passkey considerations

- Why weren’t we doing this from the start?

- Suppose we use passkeys everywhere, what might be an expected consequence of this?
  - Hint: 5A

Trusted Platform Module (TPM)

- A special purpose hardware chip intended for storing cryptographic material
- Designed in a way that is hardened and tamper-resistant
- Now widespread in modern computers and phones

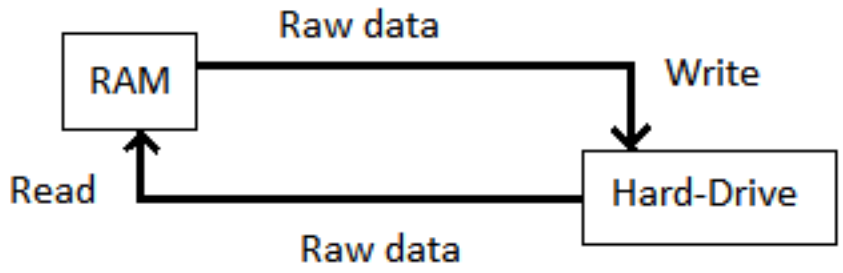
Limitations of Password-based authentication

- What is our password check actually doing?

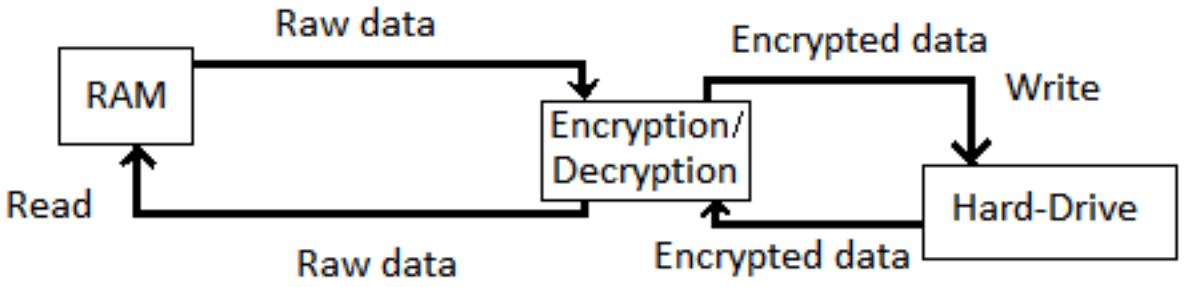
Full Disk Encryption

- Access to the unauthenticated system without knowing the password should leak nothing
- Leverages the presences of a TPM
- Full Disk Encryption achieves this by:
  - Encrypting the entire contents of your system under a key
  - On boot, user must supply the key to render the system usable
  - As needed during use, the OS will decrypt data as it is moved to memory and used

Without encryption:



With encryption:



File-system Encryption

- Full disk encryption provides no protections for a logged in system
- For even greater security, individual files can also be encrypted such that each file independently must be decrypted before being usable
- Still not perfect

Next Time

Privacy and Security on the Internet