

Authentication Part 2

CMSC 23200, Spring 2025, Lecture 16

Grant Ho

University of Chicago, 05/15/2025
(Slides adapted from Blase Ur)

Logistics

- Assignment 5 due tonight (Thur, May 15) by 11:59pm
- Assignment 3 grades posted
 - Regrade requests until Thurs, May 22
- Final Exam on Wed, May 28 @ 10am (BOTH SECTIONS)
 - Closed notes
 - Practice exams uploaded early next week on Ed

Outline

- Recap: Password Cracking Goal & Overview
- Password Cracking Methods: Markov Models
- Practical Authentication Issues
- Password Alternatives / Add-ons

Offline Attack (Password Database Cracking)

Attacker compromises database (e.g., via SQL injection)

- `hash("Blase's password") =`

`$2a$04$iHdEgkI681VdDMc3f7edau9phRwORvhYjqWAIb7hb4B5uFJO1g4zi`

`$` = delimiter

`2a` = bcrypt

`04` = 2^4 iterations (cost)

`iHdEgkI681VdDMc3f7edau` = 16 bytes of salt (radix-64 encoded)

`9phRwORvhYjqWAIb7hb4B5uFJO1g4zi` = 24 bytes of hash output (radix-64 encoded)

- Attacker makes guesses (from most likely/probable to the least) and hashes those guesses
- Finds match → try on other sites
 - Password **reuse** is a core problem

Password Cracking



80d561388725fa74f2d03cd16e1d687c



1. $h("123456") = e10adc3949ba59abbe56e057f20f883e$
2. $h("password") = 5f4dcc3b5aa765d61d8327deb882cf99$
3. $h("monkey") = d0763edaa9d9bd2a9516280e9044d885$
4. $h("letmein") = 0d107d09f5bbe40cade3de5c71e9e9b7$
5. $h("p@ssw0rd") = 0f359740bd1cda994f8b55330c86d845$
6. $h("Chic4go") = \mathbf{80d561388725fa74f2d03cd16e1d687c}$



Some Key Password-Cracking Approaches

- Brute force
- Wordlist
- Mangle
 - Hash
- Markov
- Probabilistic
- Deep learning
- In practice

GOAL:

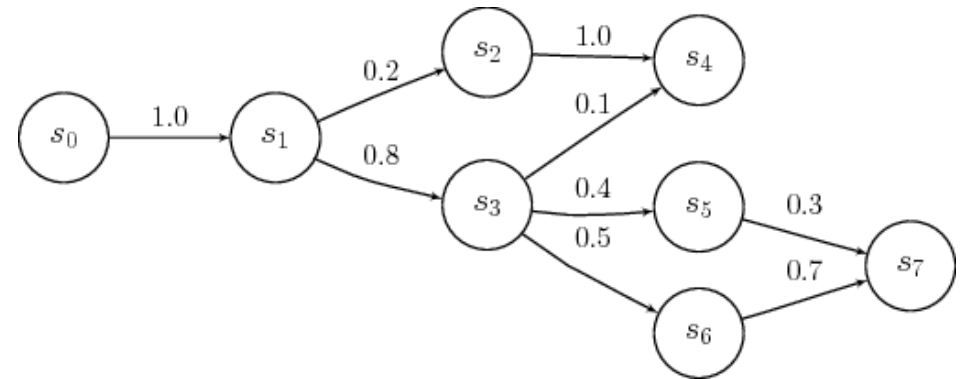
- Generate a high-probability list of password guesses, such that guesses likely correspond to real passwords
(i.e., learn real passwords from breached database as efficiently / quickly as possible)

Outline

- Recap: Password Cracking Goal & Overview
- Password Cracking Methods: Markov Models
- Practical Authentication Issues
- Password Alternatives / Add-ons

Markov Models

- Predicts future characters from previous (n-gram)
- Approach requires training data:
 - Passwords
 - Dictionaries
- Smoothing is critical
 - Enables model to handle unseen char combinations



Guessing / Generating Passwords

passw  o or maybe 0 or O or ...

Guessing / Generating Passwords

passw



Next char is:

A: 3%

B: 1%

C: 0.6%

...

O: 55%

...

Z: 0.01%

0: 20%

1: ...

Markov Models: Training

chic4gooo

2-gram model (1 character of context):

[start] → c (1.0)

4 → g (1.0)

c → h (0.5), 4 (0.5)

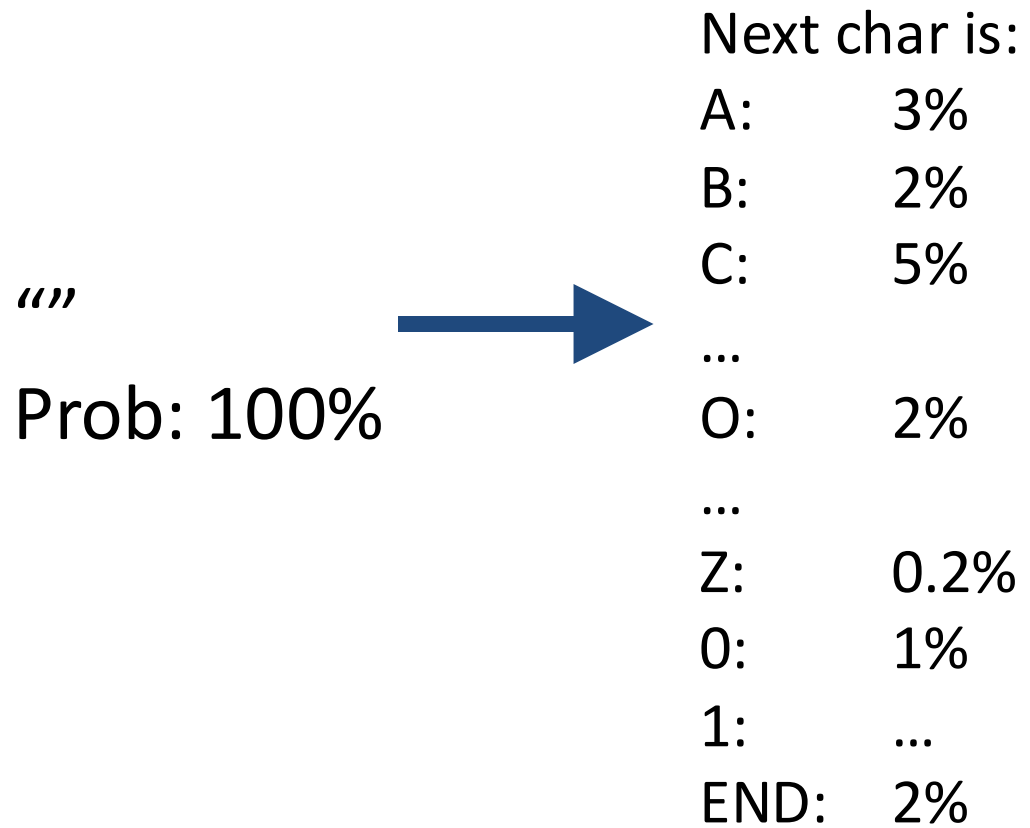
g → o (1.0)

h → i (1.0)

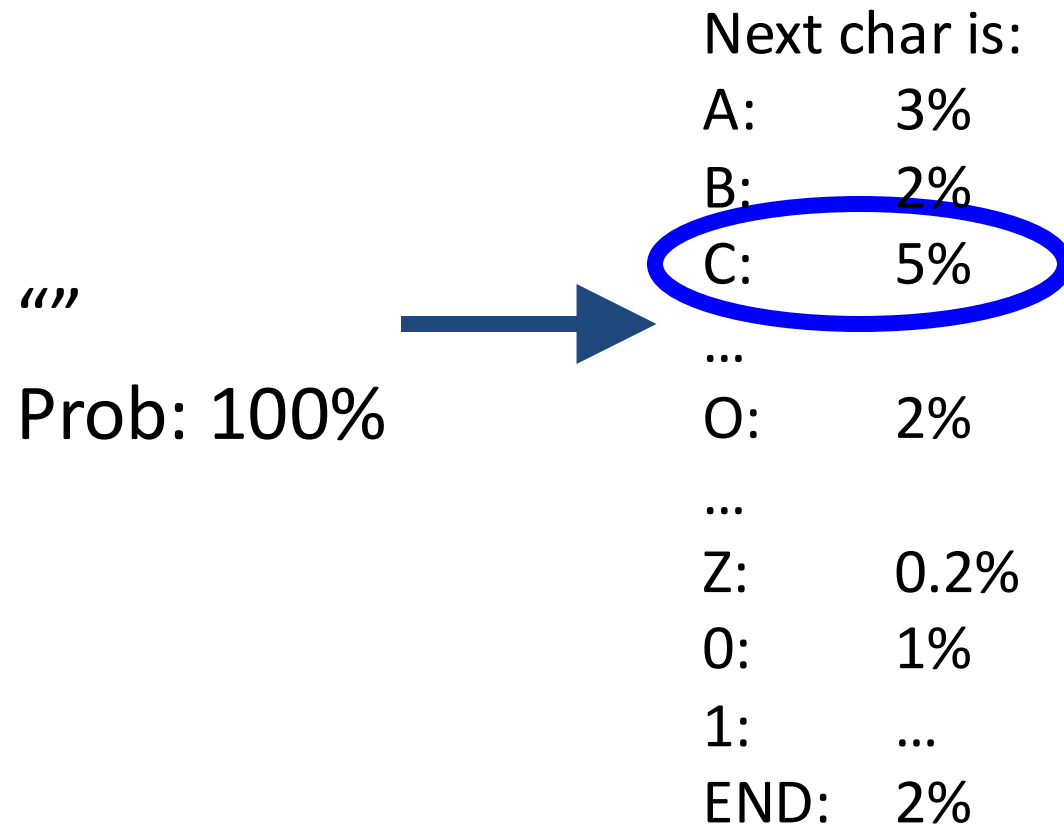
i → c (1.0)

o → o (0.67) [end] (0.33)

Guessing / Generating Passwords



Guessing / Generating Passwords



Guessing / Generating Passwords

“C”

Prob: 5%



Guessing / Generating Passwords

“C”

Prob: 5%



Next char is:

A: 10%

B: 1%

C: 4%

...

O: 8%

...

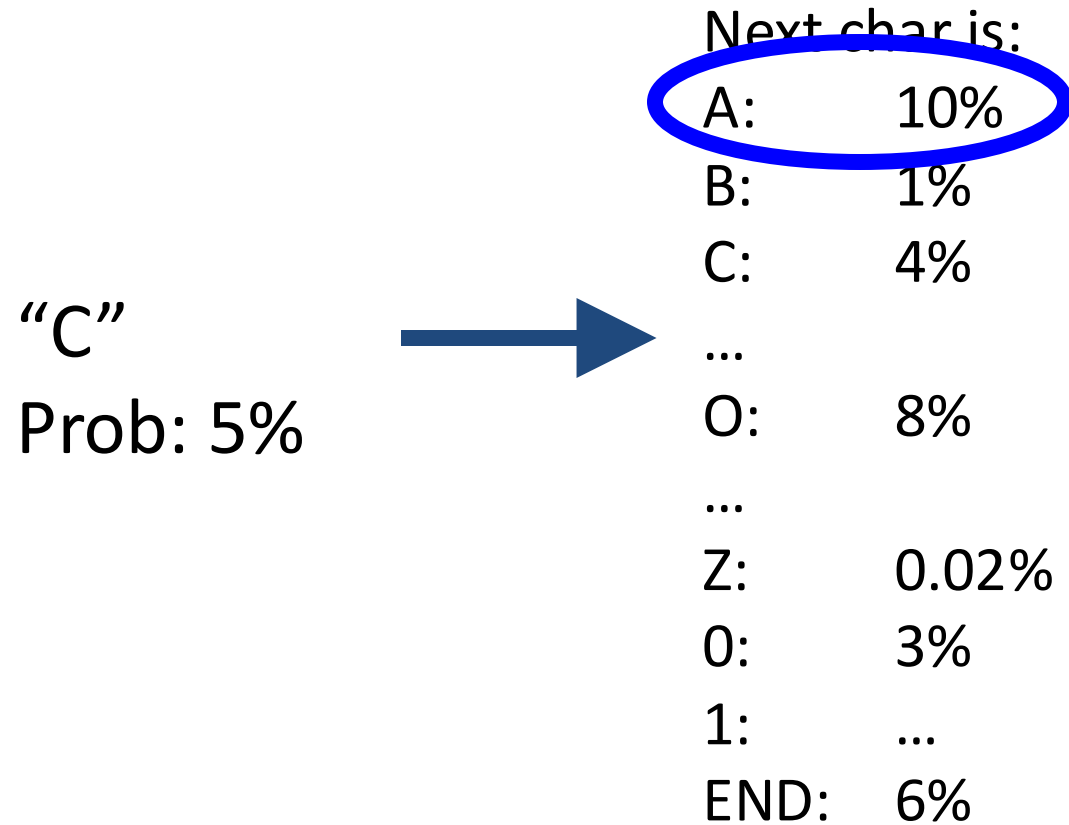
Z: 0.02%

0: 3%

1: ...

END: 6%

Guessing / Generating Passwords



Guessing / Generating Passwords

“CA”

Prob: 0.5%



Next char is:

A: 3%

B: 10%

C: 7%

...

O: 1%

...


Z: 0.03%

0: 2%

1: ...

END: 12%

Guessing / Generating Passwords

"CAB" Prob: 0.05%		Next char is:	
		A:	3%
		B:	10%
		C:	7%
		...	
		O:	1%
		...	
		Z:	0.03%
		0:	2%
		1:	...
		END:	3%

Guessing / Generating Passwords

“CAB”

Prob: 0.05%



Next char is:

A: 4%

B: 3%

C: 1%

...

O: 2%

...


Z: 0.01%

0: 4%

1: ...

END: 12%

Guessing / Generating Passwords

"CAB" Prob: 0.05%		Next char is:	
		A:	4%
		B:	3%
		C:	1%
		...	
		O:	2%
		...	
		Z:	0.01%
		0:	4%
		1:	...
		END:	12%

Guessing / Generating Passwords

“CAB”

Prob: 0.006%

Guessing / Generating Passwords

CAB - 0.006%

CAC - 0.0042%

ADD1 - 0.002%

CODE - 0.0013%

...

Professionals (“Pros”): Password Cracking

- Proprietary wordlists and configurations
 - Also use automated tools like Markov models
 - Manually tuned & interactive updates/tuning during attack
- For example: KoreLogic
 - Password audits for Fortune 500 companies
 - Run DEF CON “Crack Me If You Can”

KoreLogic
SECURITY



How different are “Pro” results?

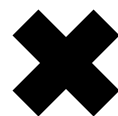
4 password sets

```
password
iloveyou
team0123
...
```

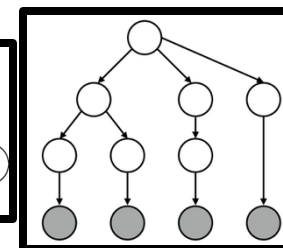
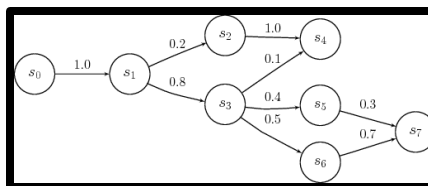
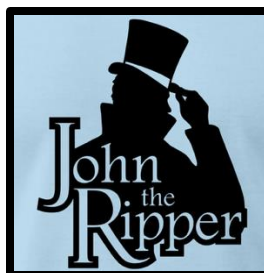
```
passwordpassword
1234567812345678
!1@2#3$4%5^6&7*8
...
```

```
Pa$$w0rd
iLov3you!
1QaZ2W@x
...
```

```
pa$$word1234
12345678asDF
!q1q!q1q!q1q
...
```

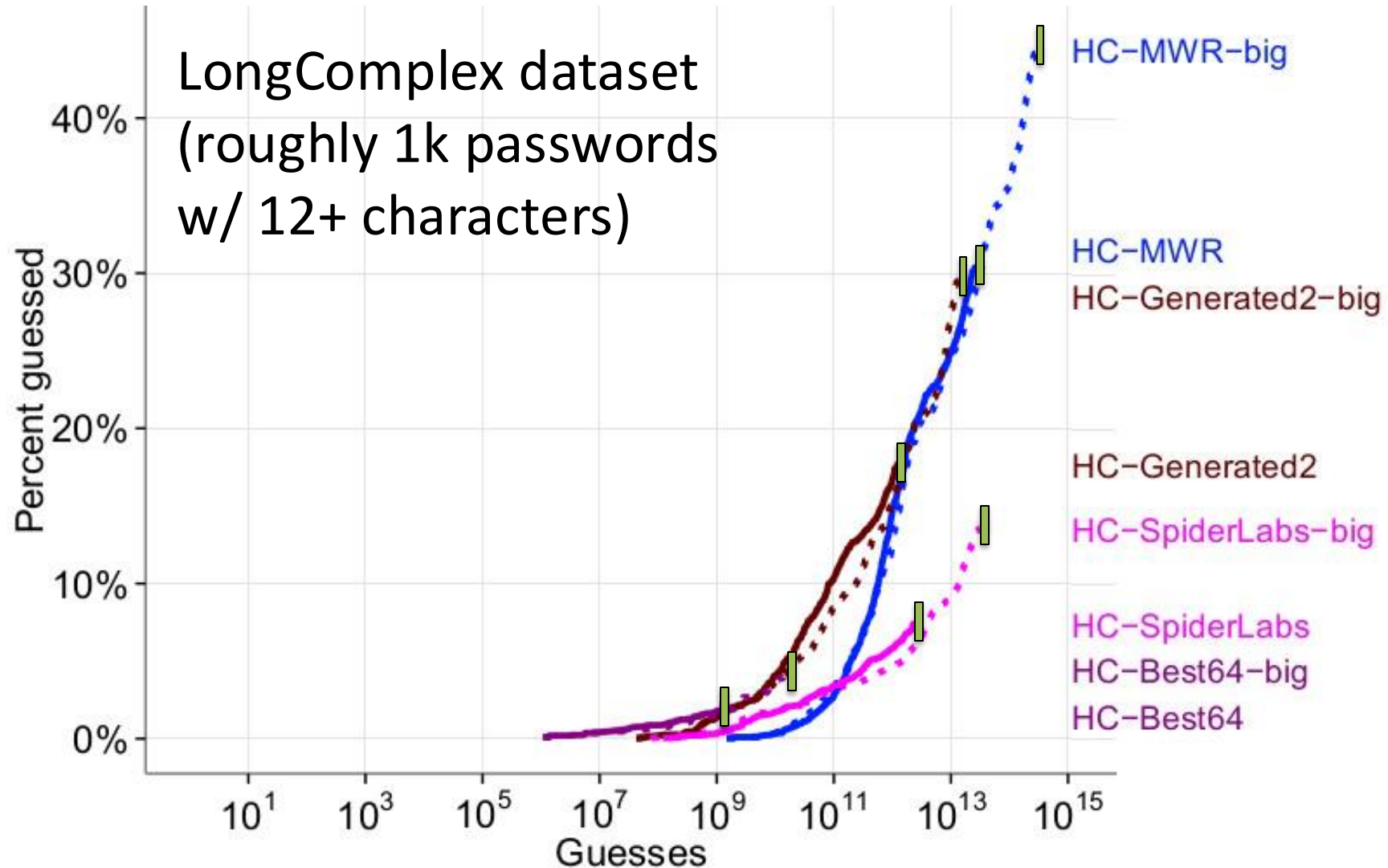


5 approaches

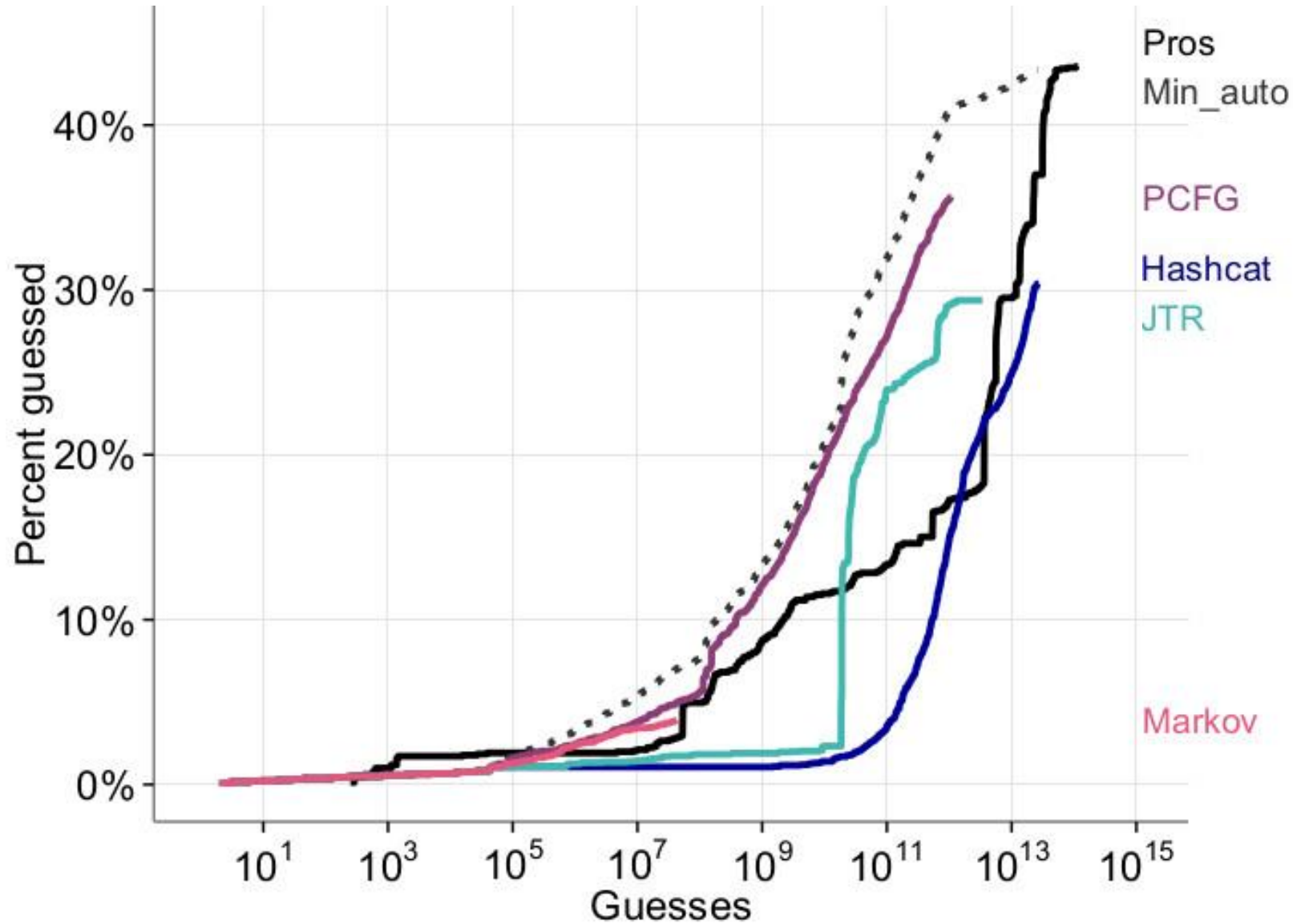


Blase Ur, Sean M. Segreti, Lujo Bauer, Nicolas Christin, Lorrie Faith Cranor, Saranga Komanduri, Darya Kurilova, Michelle L. Mazurek, William Melicher, Richard Shay. Measuring Real-World Accuracies and Biases in Modeling Password Guessability. In *Proc. USENIX Security Symposium*, 2015.

Configuration Is Crucial



Comparison for Complex Passwords



Per-Password Highly Impacted

P@ssw0rd!

Per-Password Highly Impacted

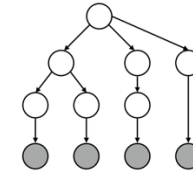
- JTR guess # 801



P@ssw0rd!

Per-Password Highly Impacted

- JTR guess # 801
- Not guessed in 10^{14} PCFG guesses



P@ssw0rd!

Outline

- Recap: Password Cracking Goal & Overview
- Password Cracking Methods: Markov Models
- Practical Authentication Issues
- Password Alternatives / Add-ons

Authentication in Practice: Password Reuse 😞

Password Reuse is Very Common

The Hacker News

Subscribe – Get Latest News

Home

Data Breaches

Cyber Attacks

Vulnerabilities

Webinars

Expert Insights

Contact



Facebook CEO Zuckerberg's Twitter, Pinterest accounts Hacked! And the Password was...

Jun 06, 2016 Mohit Kumar



— Trending News



Google Rolls Out On-Device AI Protections to Detect Scams in Chrome and Android



Security Tools Alone Don't Protect You – Control Effectiveness Does



Reevaluating SSEs: A Technical Gap Analysis of Last-Mile Protection



Fake Security Plugin on WordPress Enables Remote Admin Access for Attackers



SonicWall Patches 3 Flaws in SMA

Monitoring the Underground Economy


Listing

trdealmgn4uvm42g.onion/listing/3600

Welcome back, [redacted] 0 0 0 BTC 0.0000 Home My RealDeal Support Logout

TheRealDeal All I want to order ... Go

Home / Information and Fraud / Databases / LinkedIn 167M



LinkedIn 167M

By [peace_of_mind](#) (100.0%) Level 1 (14)

0 5.0000 / BTC 5.0000

In stock.

Postage Option

Escrow Yes, escrow by RealDeal is available.

Class Digital

Ships From Worldwide

Qty: 0

Buy It Now

Favorite Question

[BEST PRODUCTS](#)[REVIEWS](#)[NEWS](#)[VIDEO](#)[HOW TO](#)[SMART HOME](#)[CARS](#)[DEALS](#)[JOIN / SIGN IN](#)

SECURITY

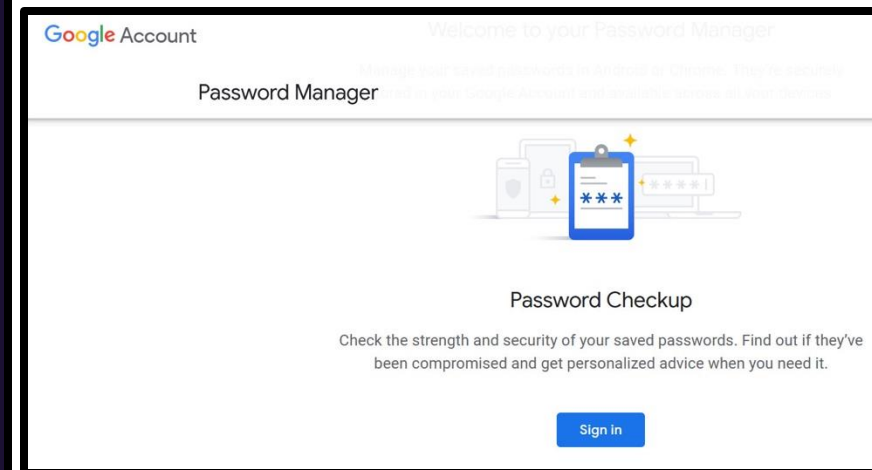
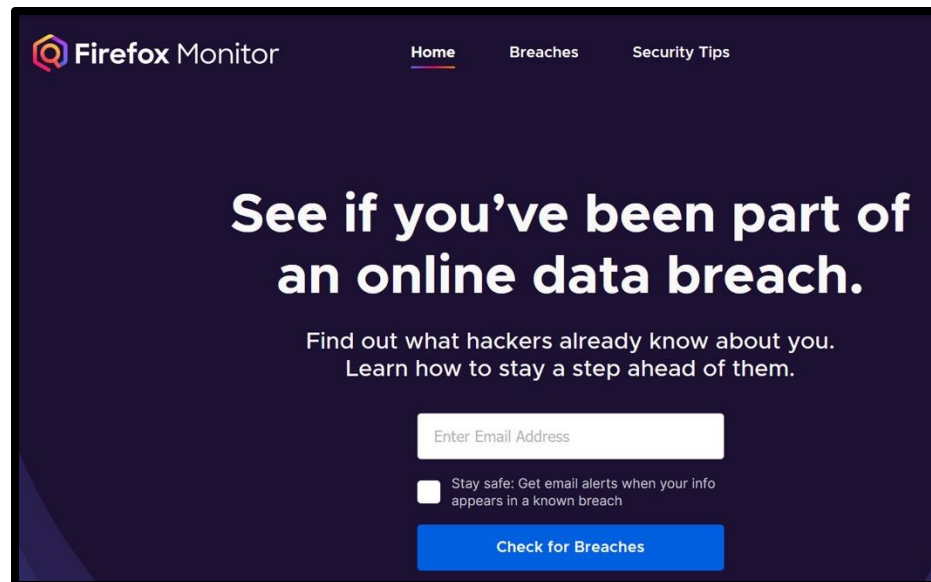
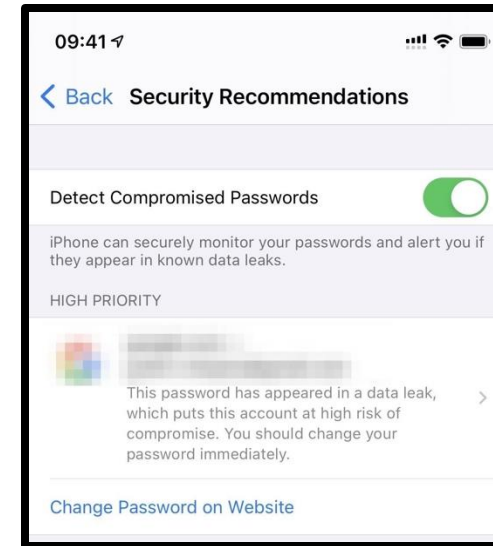
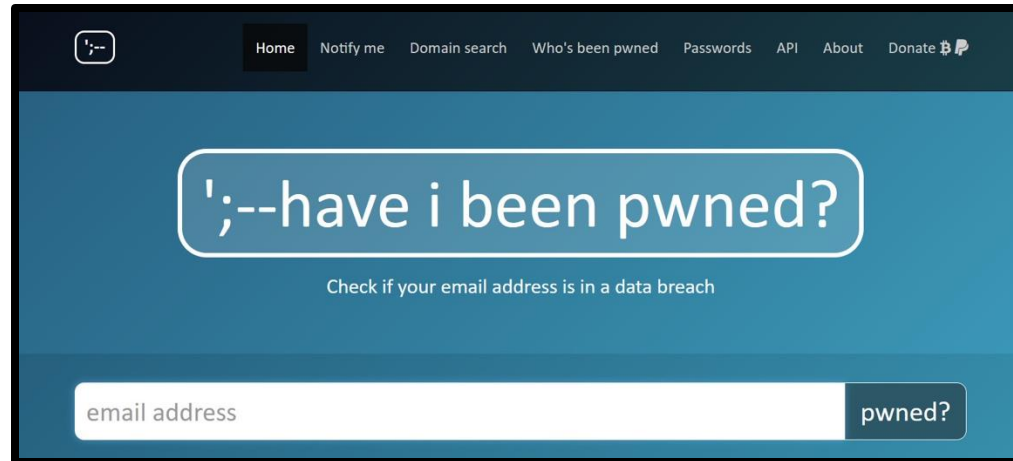
Facebook buys black market passwords to keep your account safe

The company's security chief says account safety is about more than just building secure software.

BY KATIE COLLINS | NOVEMBER 9, 2016 12:56 PM PST



Checking for Compromised Credentials



Password Managers

- Use one master password & have the password manager randomly generate + autofill passwords for every website
- Need to trust password manager service (software, sometimes service's web servers) and your single master password
 - Often still a good idea + best practice



bitwarden



1Password

Authentication in Practice: I Forgot My Password

Password/Account Reset: Big Challenge!

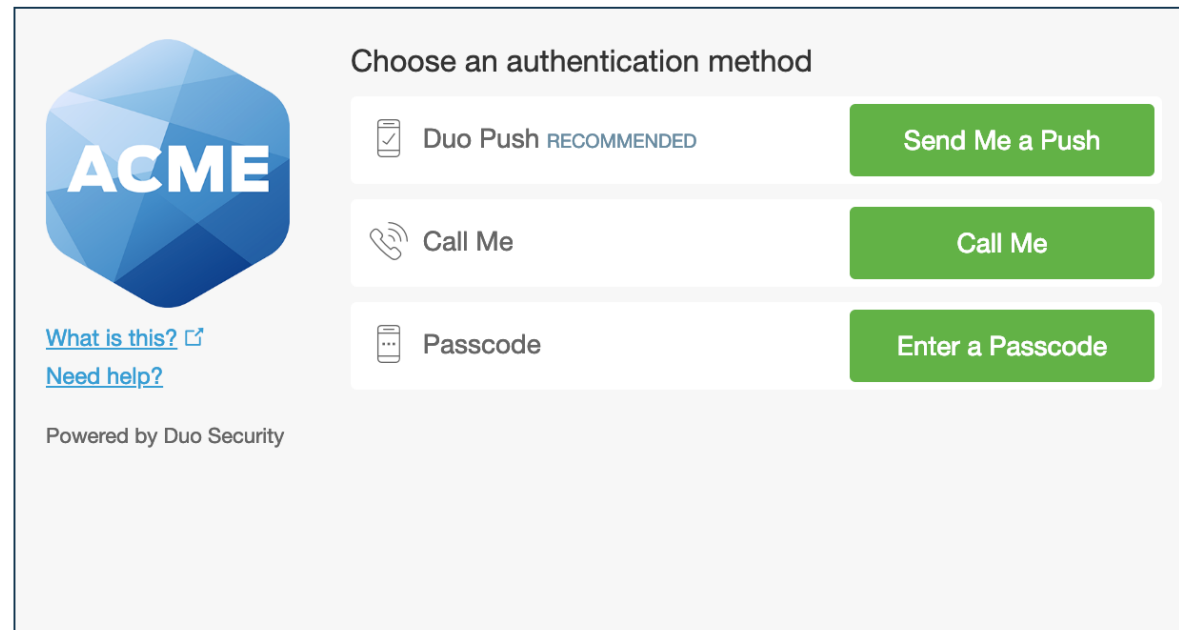
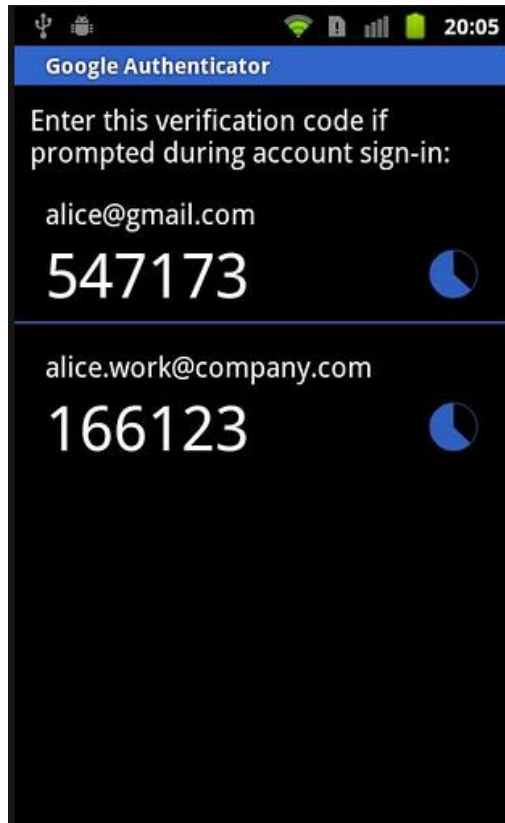
Help! I forgot my password!

- (No backup)
- Send an email?
- Security questions?
- In-person verification?
- Other steps?

Outline

- Recap: Password Cracking Goal & Overview
- Password Cracking Methods: Markov Models
- Practical Authentication Issues
- Password Alternatives / Add-ons

Two-Factor Auth



Hardware 2FA: Physical Tokens

- Codes based on a cryptographic key & challenge-response
 - User interaction (e.g., pushing button triggers device to sign/verify the challenge)



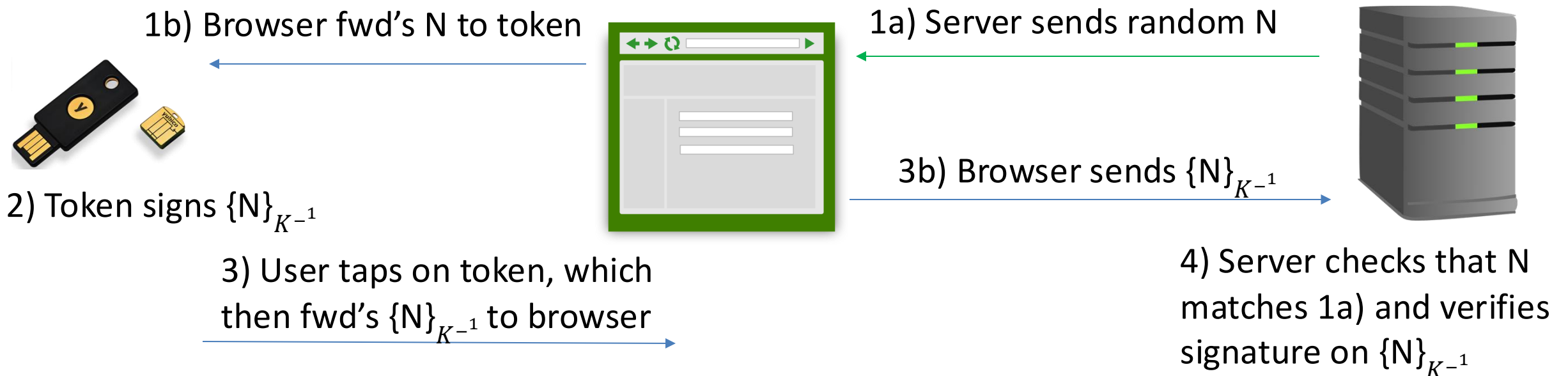
U2F: Hardware 2FA Protocol

- Hardware 2FA token has a public & private key pair embedded in device

A. Setup

- Alice's browser gets K = 2FA token's public key and sends K to server
- Server stores (username, K) in its 2FA database

B. Authentication



Adding Phishing Resistance

1b) Browser fwd's N to token

AND it includes **D** = domain of actual webpage in browser



2) Token signs $\{N, \mathbf{D}\}_{K^{-1}}$

3) User taps on token, which then fwd's $\{N, \mathbf{D}\}_{K^{-1}}$ to browser



1a) Server sends random number N

3b) Browser sends $\{N, \mathbf{D}\}_{K^{-1}}$



4) Server checks:

- **D** matches its domain
- N matches what it sent
- Valid signature on $\{N, \mathbf{D}\}_{K^{-1}}$

Phishing Attack Now Fails!

- During phishing attack, browser will be at website w/ domain $D' = \text{gmai1.com}$, instead of real domain $D = \text{gmail.com}$



Phishing Attack Now Fails!

- During phishing attack, browser will be at website w/ domain $D' = \text{gmail1.com}$, instead of real domain $D = \text{gmail.com}$

1b) Browser fwd's N to token

AND it includes $D' = \text{domain of actual webpage in browser}$



2) Token signs $\{N, D'\}_{K^{-1}}$

3) User taps on token, which then fwd's $\{N, D'\}_{K^{-1}}$ to browser



1a) Gmail sends random number N

3b) Browser sends $\{N, D'\}_{K^{-1}}$



4) Gmail checks:

- N matches what it sent
- Valid signature on $\{N, D'\}_{K^{-1}}$
- But D' doesn't match its domain!**

Passwordless FIDO2

- **Goal:** Authenticate on the web using public-key crypto directly, instead of using passwords (e.g., with U2F hardware tokens)
- Originally intended to be implemented in specialized hardware (e.g., 2FA tokens)
 - But now allows for other authenticators like TouchID

FIDO2 BRINGS SIMPLER, STRONGER AUTHENTICATION TO WEB BROWSERS



FIDO AUTHENTICATION: THE NEW GOLD STANDARD



Protects against phishing, man-in-the-middle and attacks using stolen credentials



Log in with a single gesture – HASSLE FREE!



Already supported in market by top online services

Passkeys

Goal: Make FIDO2 / WebAuthn more usable by syncing the private key across devices

- See: <https://developers.google.com/identity/passkeys>
- Example of Google's changing approach over the years:

Our Passwordless journey

Passkeys bring us much closer to the passwordless future we've been mapping out for over a decade.

2008	2011	2012	2013	2014	2017	2019	2023
Launched Google Password Manager for easier and safer sign-ins.	Enabled 2-Step Verification (2SV) for Google accounts.	Introduced phishing-resistant security key for Google employees.	Joined the FIDO Alliance to drive open standards for a passwordless world.	Expanded phishing-resistant security keys for everyone.	Introduced Advanced Protection Program (APP) for high-risk users.	Extended our FIDO support in Android for passwordless re-auth across websites.	Enabled passkeys for Google Accounts, Workspace customers and 3rd party partners on Chrome and Android.

Modern Password / Auth Recommendations

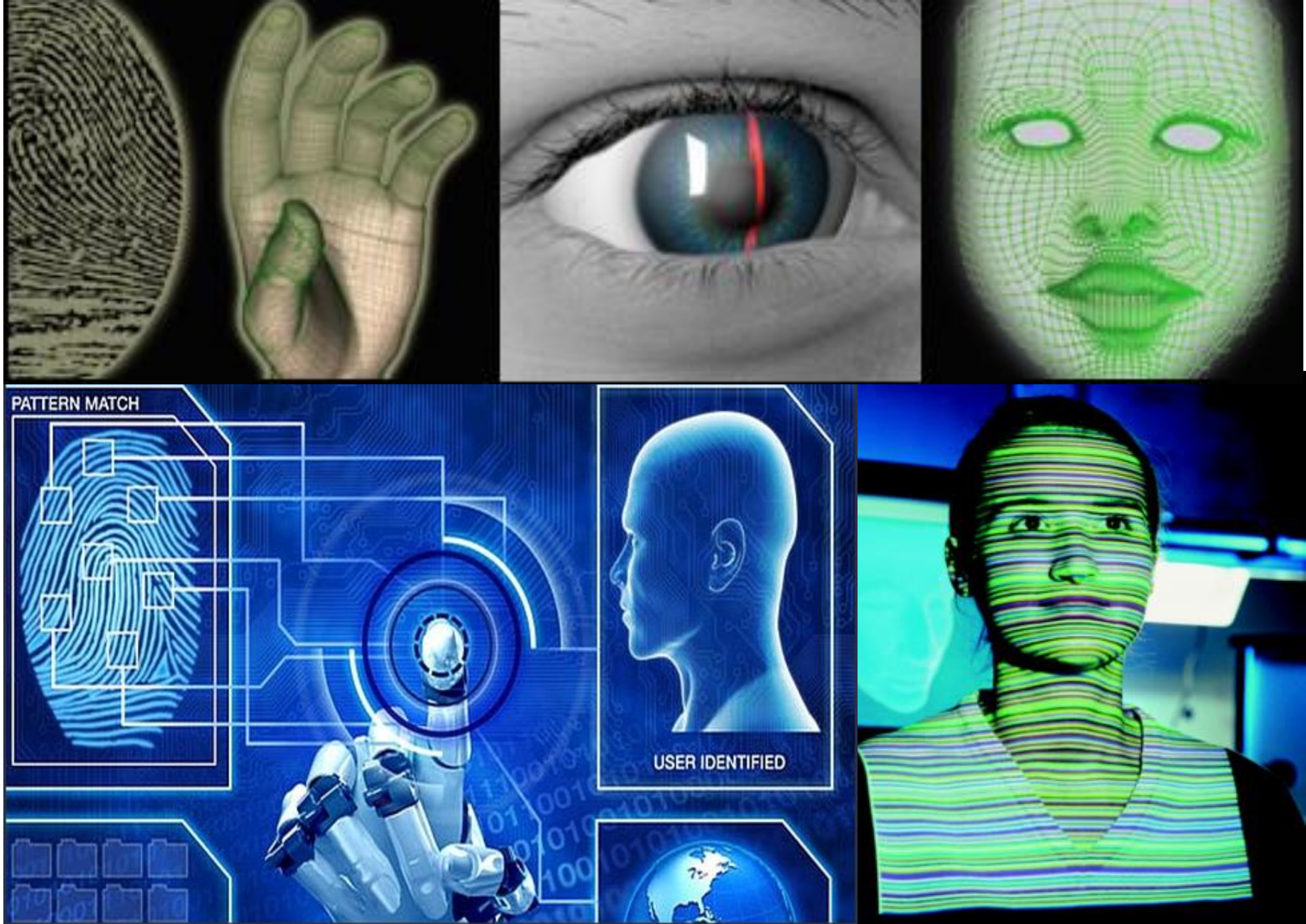
What can websites do to strengthen authentication?

- Rate-limit authentication attempts
- Minimum password length should be at least 8 characters
- Maximum password length should be at most 64 characters
 - Do not allow unlimited length, to prevent denial-of-service
- Promptly check passwords vs. known breach datasets
- Encourage/require use of two-factor authentication (consider password-less FIDO2)

What about
Biometrics?



•Images fair use from wordpress.com and kaspersky.com, as well as Creative Commons from matsuyuki on Flickr



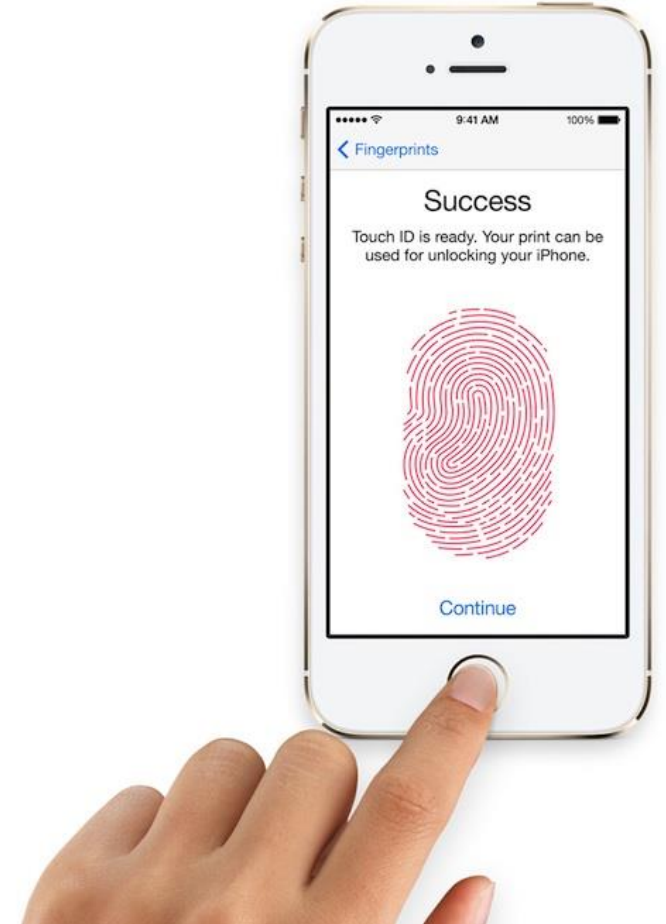
Images fair use from fbi.gov, ifsecglobal.com, and siemens.com

Biometrics

- Physical
 - Fingerprint
 - Iris scans or retina scans
 - Face recognition
 - Finger/hand geometry
- Behavioral
 - The way you type
- Mixed / Hybrid
 - Voice or speech recognition
 - Many others

Smartphone Biometrics

- Purpose is to reduce the number of times a user must enter their password
- Falls back to the password
- Some facial recognition systems can be tricked by a photo
- Some fingerprint recognition systems can be tricked by a gummy mold



Biometrics Authentication

- During “enrollment”:
 - Device extracts a set of features from biometric input
 - ML model trained on this set of features
- During authentication (“test time”)
 - Features extracted from new biometric input
 - ML model used to classify whether new input is “close enough” to target user
- “ML model” & classification could just be similarity/distance between enrollment input & authentication input

Practical Challenges for Biometrics

- Immutable (can't be changed easily)
- But biometrics can inadvertently change over time (e.g., injury), sensitive to environment changes, etc.
- High equipment costs (client-side)
- Non-secret and potentially easy to forge
- Potentially sensitive data