13. Authentication and Access Control

Blase Ur and David Cash
February 10th, 2020
CMSC 23200 / 33250

THE UNIVERSITY OF CHICAGO

Who Am I?

- David Cash
  - Distinguished cryptographer
  - Fan of rare plants
  - All-around good guy

Or Am I?

How (and why) do we authenticate users?

Authentication Abstractly

- Verify that **people** or **things** (e.g., a server) are who they claim to be
- Authentication ≠ Authorization
  - *Authorization* is deciding whether an entity should have access to a given resource
- Access control lists / policies
- Terminology:
  - **Principal:** the legitimate owner of an identity
  - **Claimant:** entity trying to be authenticated

How We Authenticate (1/2)

How We Authenticate (1/2)

- Something you know
  - Password
  - PIN (Personal Identification Number)
- Something you have
  - Smart card
  - Private key (of a public-private key pair)
  - Phone (running particular software)
- Something you are
  - Biometrics (e.g., iris or fingerprint)

How We Authenticate (2/2)

- Somewhere you are
  - Location-limited channels
- Someone you know (social authentication)
  - Someone vouches for you
  - You can identify people you should know
- Some system vouches for you
  - Single sign-on (e.g., UChicago shib)
  - PKI Certificate Authorities


12345
password
123456789
iloveyou
princess
rockyou
1234567
12345678
abc123
nicole
daniel
babygirl
monkey
lovely
jessica
654321
michael
ashley
qwerty
111111
iloveu
000000
michelle
tigger
sunshine
chocolate
password1
soccer
anthony
friends
butterfly
purple
angel
jordan
liverpool
justin
loveme
football
123123
basketball
superman
pretty
secret
angels
jennifer
hannah
playboy
666666
jasmine
tinkerbell
andrea
carlos
eminem
flower
samantha
hottie
cookie
bubbles
dragon
shadow
summer
melissa
matthew
andrew
chelsea
barbie
danielle
elizabeth
brandon
poohbear
softball
whatever
lovers
spongebob
forever
robert
rangers
thomas
steven
william
christian
carolina
snoopy
sakura
princess1
patrick
pokemon
heaven
hello
mickey
jesus
computer
orange
rainbow
charlie
america
beautiful
poohbear
sweety
jonathan
miguel
taylor
dancer
blink182
monica
madison
killer
tweety
naruto
junior
destiny
rachel
joseph
cheese
mother
sandra
chicken
batman
shorty
george
victoria
brittany
pepper
thomas
alejandra
mariposa

Why Are Passwords So Prevalent?

Why Are Passwords So Prevalent?

- Easy to use
- Easy to deploy
- Nothing to carry
- No “silver-bullet” alternative

Attacks on Passwords Are Common

LinkedIn

SONY

Adobe

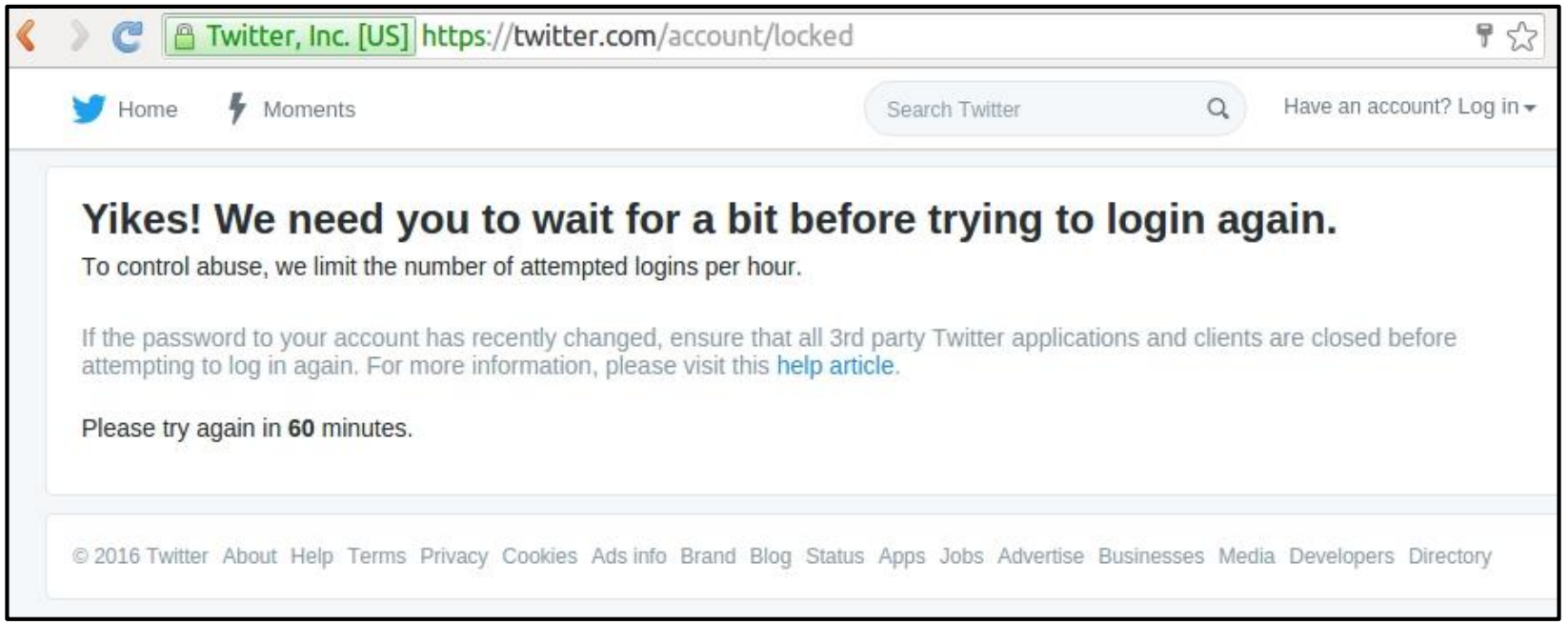
GAWKER

Attacks Against Passwords

- Online attack
  - Try passwords on a live system
  - Usually rate-limited

Attacks Against Passwords

- Online attack
  - Try passwords on a live system
  - Usually rate-limited



Attacks Against Passwords

- Online attack
  - Try passwords on a live system
  - Usually rate-limited
- Offline attack
  - Try to guess passwords from the password store / password database

Some Breached Companies

LinkedIn

Adobe

SONY

Domino's Pizza

GAWKER

000webhost.com
better than paid hosting

YAHOO!

STRATFOR
GLOBAL INTELLIGENCE

Attacks Against Passwords

- Online attack
  - Try passwords on a live system
  - Usually rate-limited
- Offline attack
  - Try to guess passwords from the password store / password database
- Phishing attack

Attacks Against Passwords

- Online attack
  - Try passwords on a live system
  - Usually rate-limited
- Offline attack
  - Try to guess passwords from the password store / password database
- Phishing attack
- Shoulder surfing

Attacks Against Passwords

- Online attack
  - Try passwords on a live system
  - Usually rate-limited
- Offline attack
  - Try to guess passwords from the password store / password database
- Phishing attack
- Shoulder surfing
- Attack password-protected file / device

Storing Passwords

- **Hash** and **salt** passwords
- Hash function: one-way function
  - Traditionally designed for efficiency (e.g., MD5)
  - Password-specific hash functions (e.g., bcrypt, scrypt, PBKDF2)

Storing Passwords

- Salt: random string assigned per-user
  - Combine the password with the salt, then hash it
  - Stored alongside the hashed password
  - Prevents the use of rainbow tables

Data-Driven Statistical Attacks

- (2009) 32 million passwords: 

- (2016) 117 million passwords: 

- (2017) 3 billion passwords: 

- Total: > 5 billions of passwords stolen from > 300 services

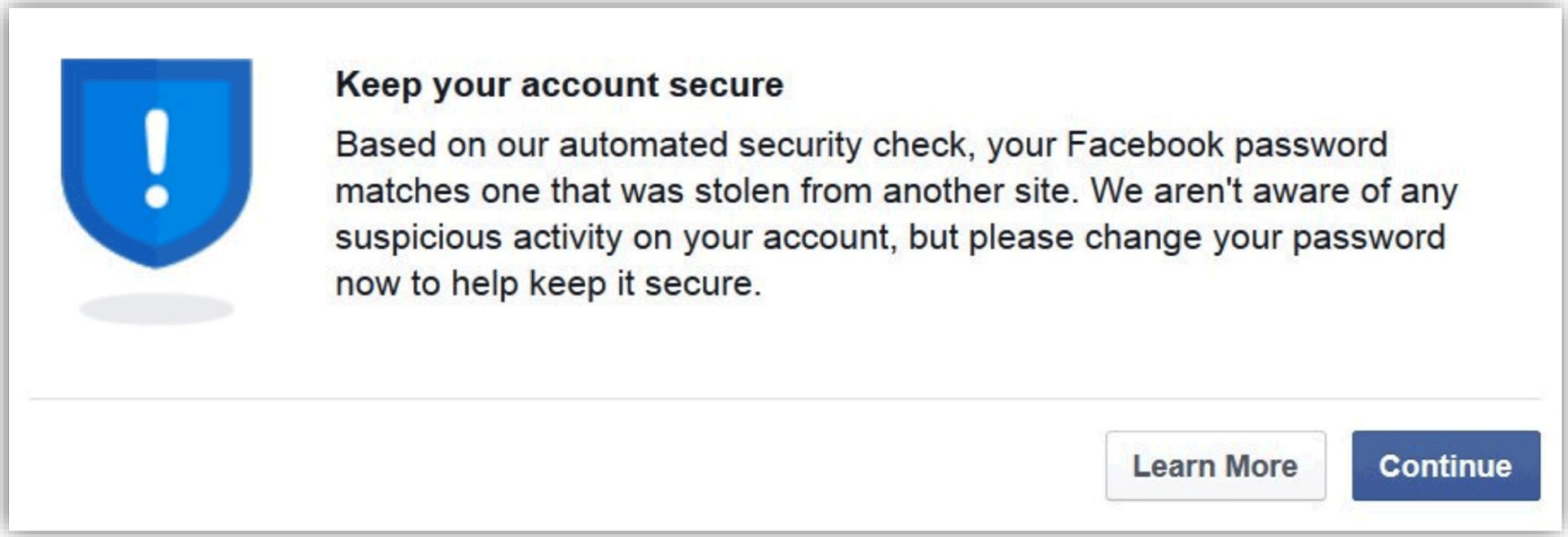
Offline Attack

- Attacker compromises database
  - hash(“Blase”) =

`$2a$04$iHdEgkI681VdDMc3f7edau9phRwORvhYjqWAIb7hb4B5uFJO1g4zi`

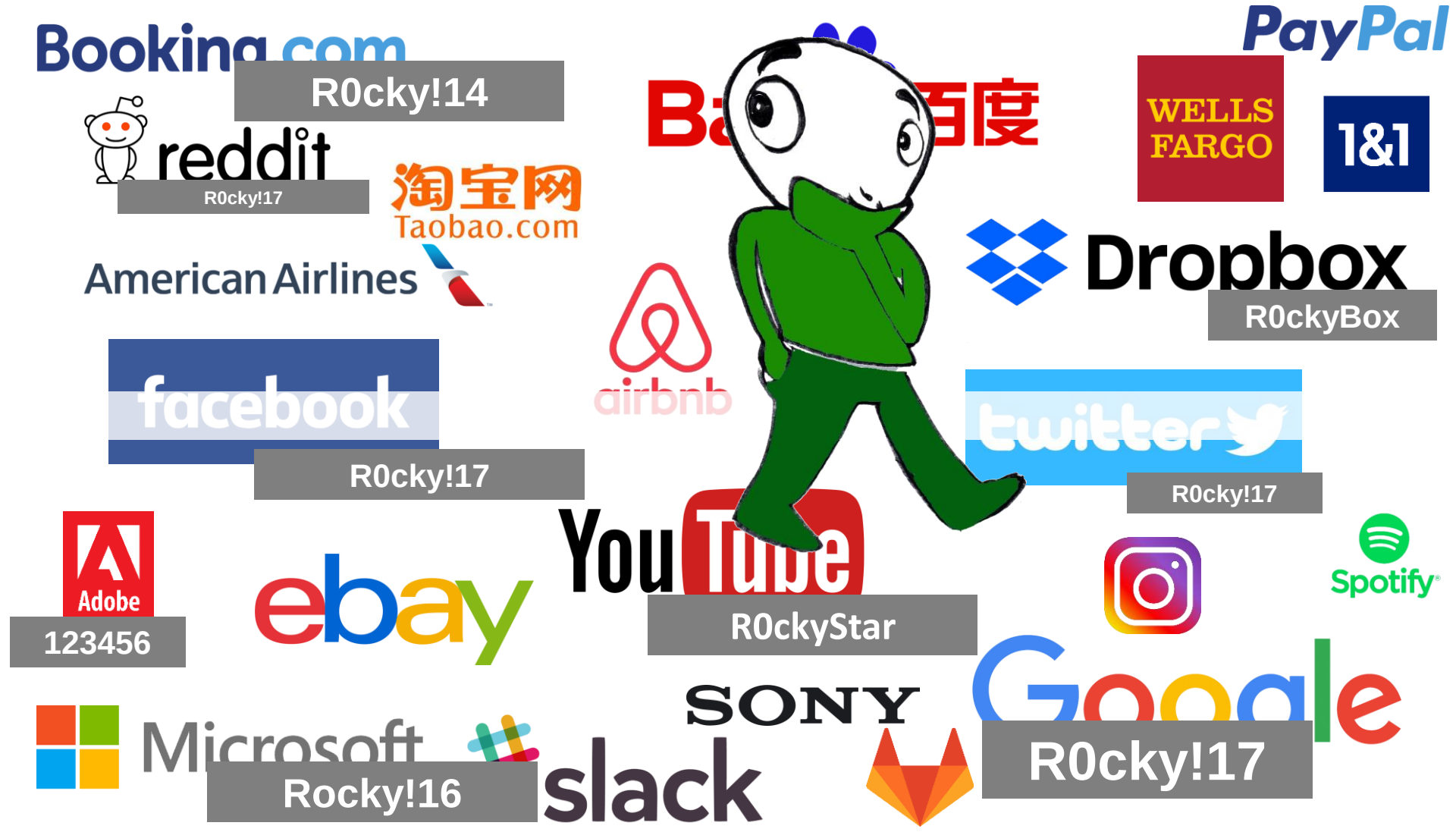
- Attacker makes and hashes guesses
- Finds match → try on other sites
  - Password **reuse** is a core problem

Password Reuse-Based Attacks



Maximilian Golla, Miranda Wei, Juliette Hainline, Lydia Filipe, Markus Dürmuth, Elissa Redmiles, Blase Ur. "What was that site doing with my Facebook Password?" Designing Password-Reuse Notifications. In *Proc. CCS*, 2018.

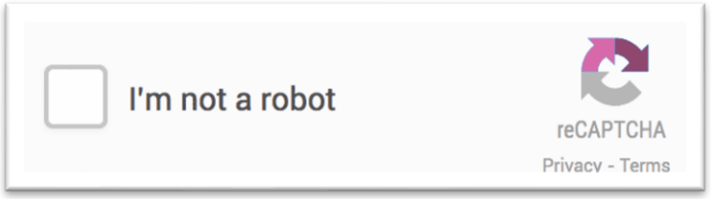
People Reuse Passwords



Memory-Hard Hash Function

Email	Argon2i Hash of Password
…	…
jim@mail.com	$argon2i$v=19$m=4096,…
…	…

Rate-Limiting Guessing



Password Strength Meter

Username

Password

acmccs18

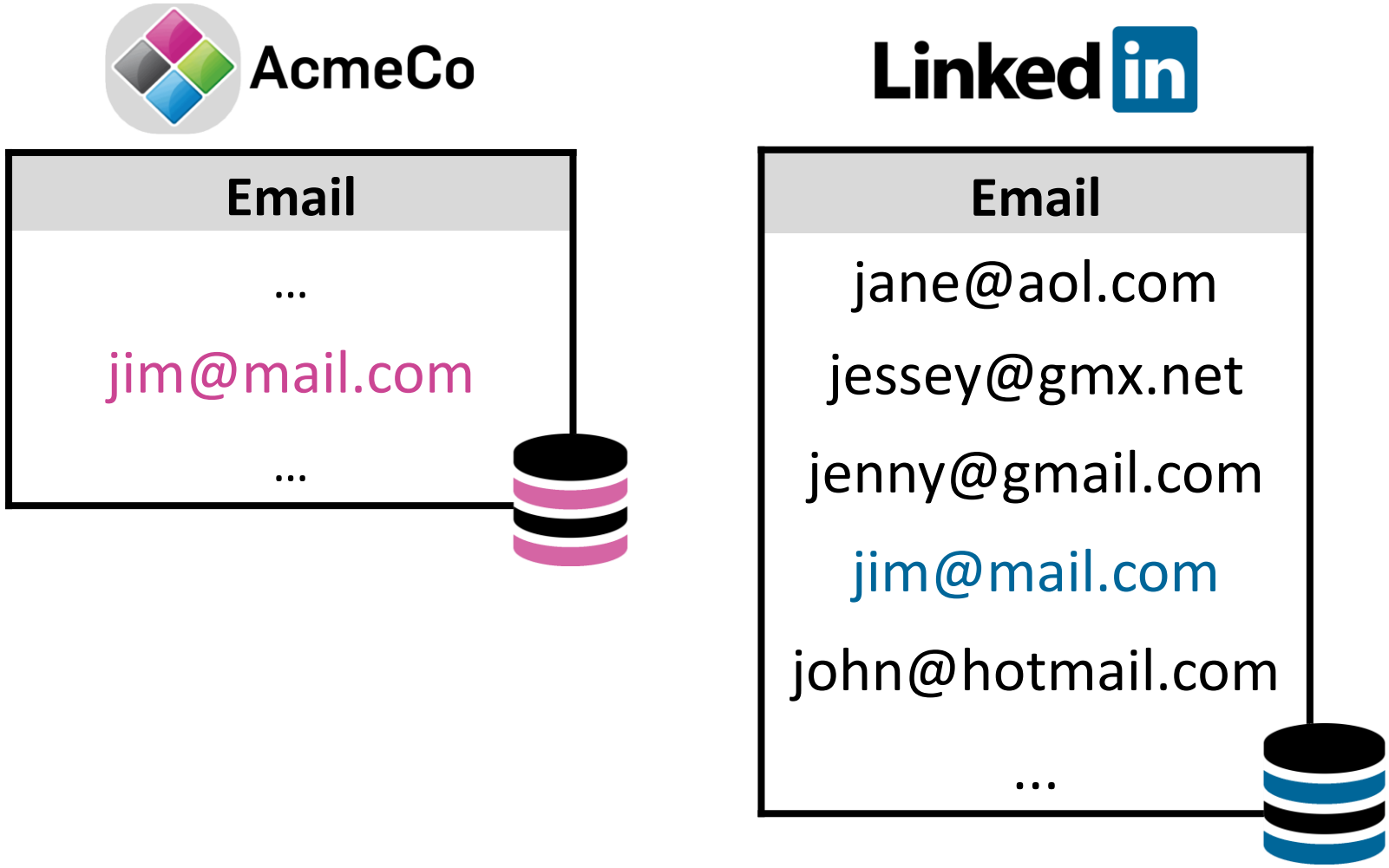
Show Password & Detailed Feedback

Your password could be better.

- Consider inserting digits into the middle, not just at the end (Why?)
- Make your password longer than 8 characters (Why?)
- Consider using 1 or more symbols (Why?)

A better choice: \a#D18cmccs

How to make strong passwords


AcmeCo
Email
...
jim@mail.com
...
LinkedIn
Email
jane@aol.com
jessey@gmx.net
jenny@gmail.com
jim@mail.com
john@hotmail.com
...

LinkedIn

Email	SHA-1 Hash of Password
jane@aol.com	7c4a8d09ca3762af61e595209
jessey@gmx.net	5baa61e4c9b93f3f0682250b6
jenny@gmail.com	7c222fb2927d828af22f59213
jim@mail.com	ba93664a90285b9ff18a7a081
john@hotmail.com	b1b3773a05c0ed0176787a4f1
...	...

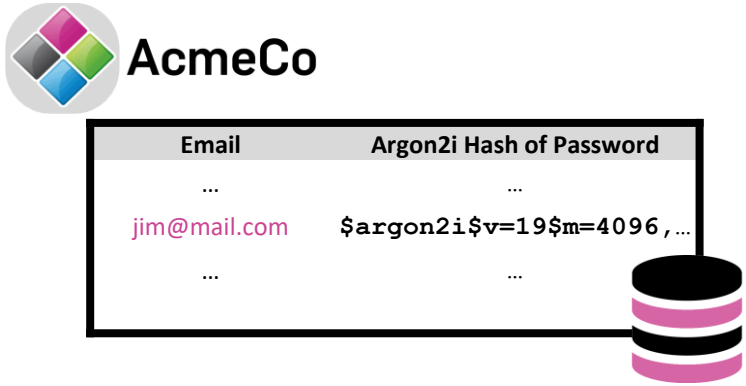
Crack All The Things!

```
Bash
$> hashcat -m 100 -a0 $TARGET $DICT
123456
Password
R0cky!17
Football!17
CanadaRocks!
```

LinkedIn

Email	Cracked SHA-1 Hashes
jane@aol.com	123456
jessey@gmx.net	5baa61e4c9b93f3f0682250b6
jenny@gmail.com	Canada4ever
jim@mail.com	R0cky!17
john@hotmail.com	HikingGuy89
…	…

Dead On Arrival



Dead On Arrival

AcmeCo

Email	Argon2i Hash of Password
…	…
jim@mail.com	`$argon2i$v=19$m=4096,…`
…	…

LinkedIn

Email	Cracked SHA-1 Hashes
jane@aol.com	`123456`
jessey@gmx.net	`5baa61e4c9b93f3f0682250b6`
jenny@gmail.com	`Canada4ever`
jim@mail.com	`R0cky!17`
john@hotmail.com	`HikingGuy89`
…	…

Dead On Arrival

AcmeCo

Email	Cracked
…	…
jim@mail.com	R0cky!17
…	…

1 guess is enough!

LinkedIn

Email	Cracked SHA-1 Hashes
jane@aol.com	123456
jessey@gmx.net	5baa61e4c9b93f3f0682250b6
jenny@gmail.com	Canada4ever
jim@mail.com	R0cky!17
john@hotmail.com	HikingGuy89
…	…

LILY HAY NEWMAN SECURITY 10.03.17 07:29 PM

SO, UH, THAT BILLION-ACCOUNT YAHOO BREACH WAS ACTUALLY 3 BILLION

Anatomy of a password disaster: Adobe's gian

Adobe,

ars TECHNICA BIZ & IT TECH SCIENCE POLICY CARS GAMING & CULTURE

RISK ASSESSMENT —

How LinkedIn's password sloppiness hurts us all

ure dumps.

guardian

all sections

';--have i been pwned?

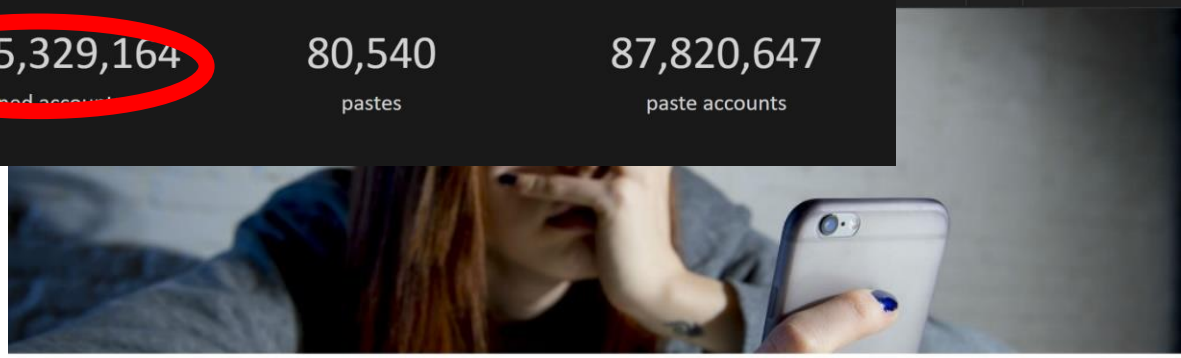
Check if you have an account that has been compromised in a data breach

314	5,555,329,164	80,540	87,820,647
owned websites	owned accounts	pastes	paste accounts

Facebook

Facebook sa[illegible] had personal data stolen in recent breach

Hackers were able to access name, birthdate and other data in nearly half of the 30 million accounts that were affected

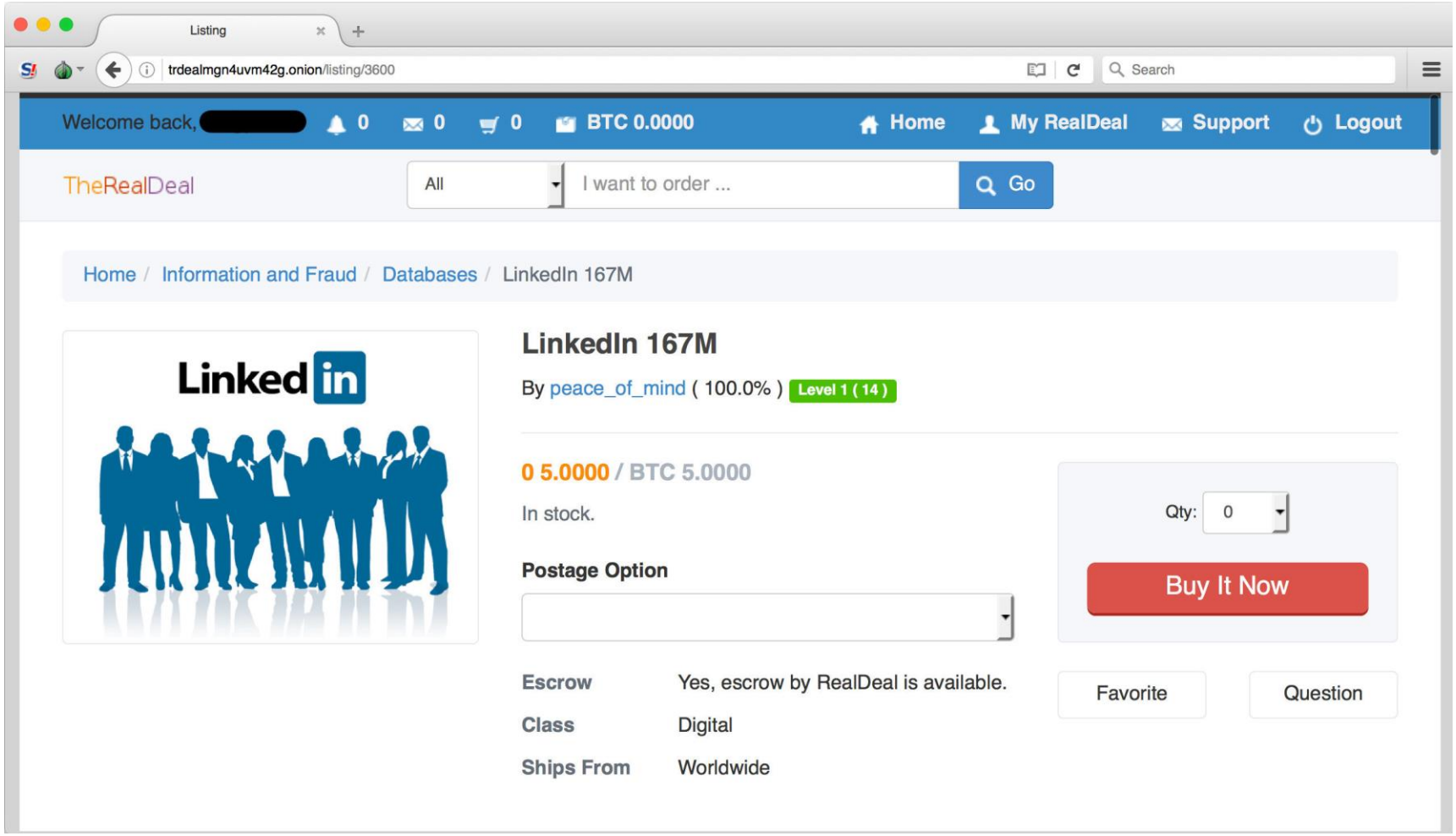


You Can Now Look Up Your Terrible 2006 MySpace Password

June 29, 2016 // 11:35 AM EST

Written by
LORENZO FRANCESCHI-BICCHIERAI
STAFF WRITER

Monitoring the Black Market



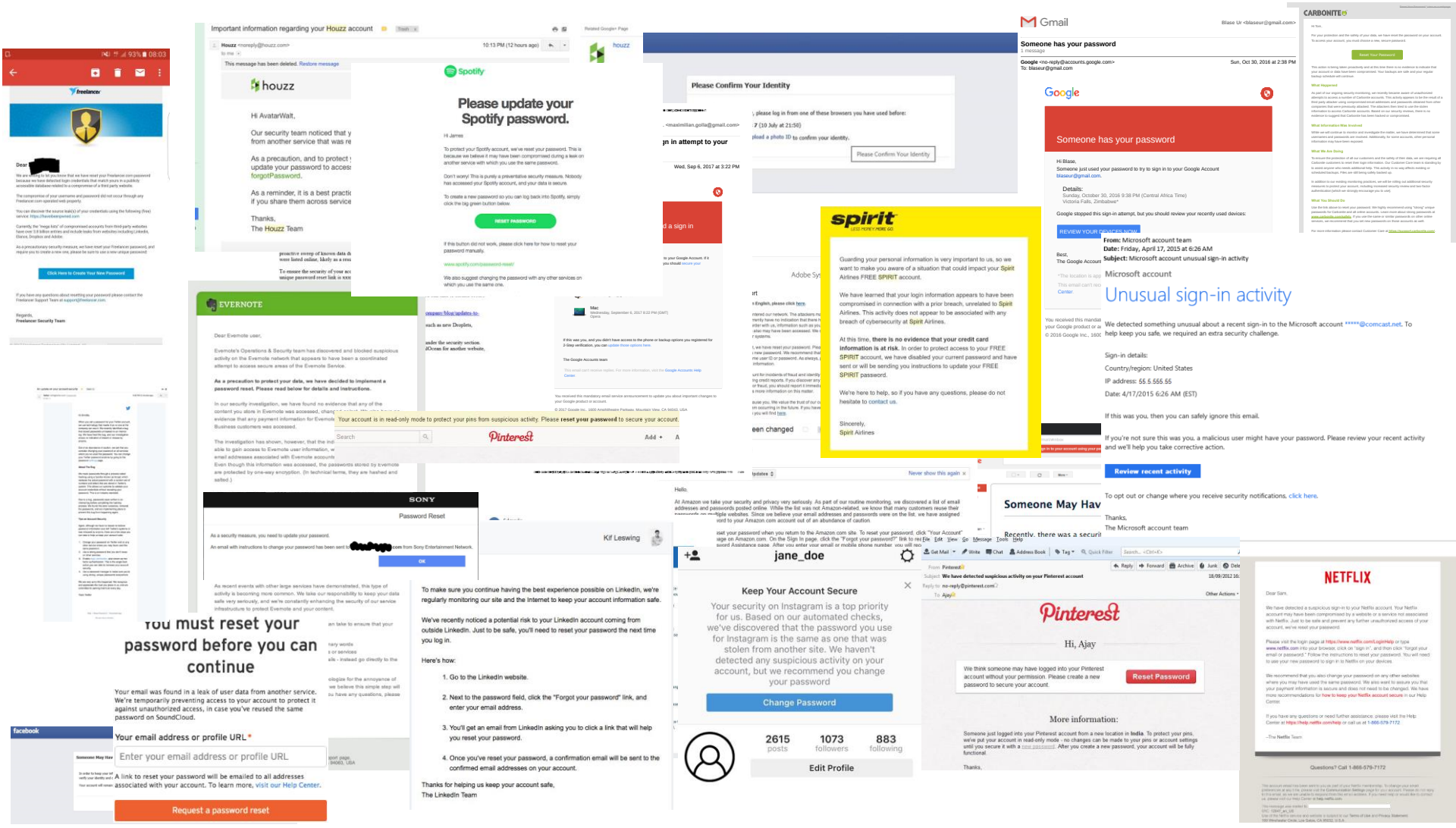
SECURITY

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

Password-Reuse Notifications



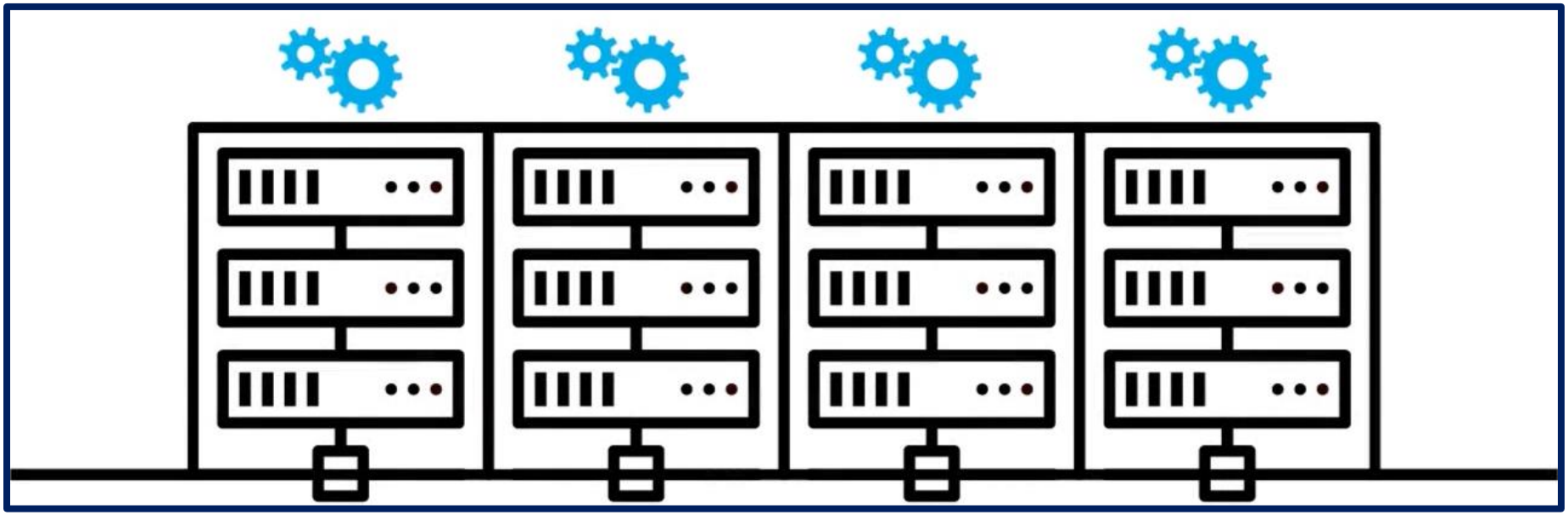
Understanding Users' Password Behaviors

Some Ways to Understand Users

- Retrospective analysis of user-created passwords 

- Large-scale online studies
- Examine real passwords
- Qualitative studies

Password Cracking



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.

Password-Strength Metrics

- Statistical approaches
  - Traditionally: Shannon entropy
  - Recently: α-guesswork
- Disadvantages for researchers
  - Usually no per-password estimates
  - Huge sample required
  - Not real-world attacks

Parameterized Guessability

- How many guesses a particular cracking algorithm with particular training data would take to guess a password

j@mesb0nd007!

Guess # 366,163,847,194

n(c$JZX!zKc^bIAX^N

Guess # past cutoff

Questions About Guessability

1) How does guessability used in research compare to an attack by professionals?
2) Would substituting another cracking approach impact research results?

Approach

password iloveyou teamo123 …	Pa$$w0rd iLov3you! 1QaZ2W@x …	passwordpassword 12345678123456 78 !1@2#3$4%5^6&7*8 …	pa$$word1234 12345678asDF !q1q!q1q!q1q …

4 password sets

5 password-cracking approaches

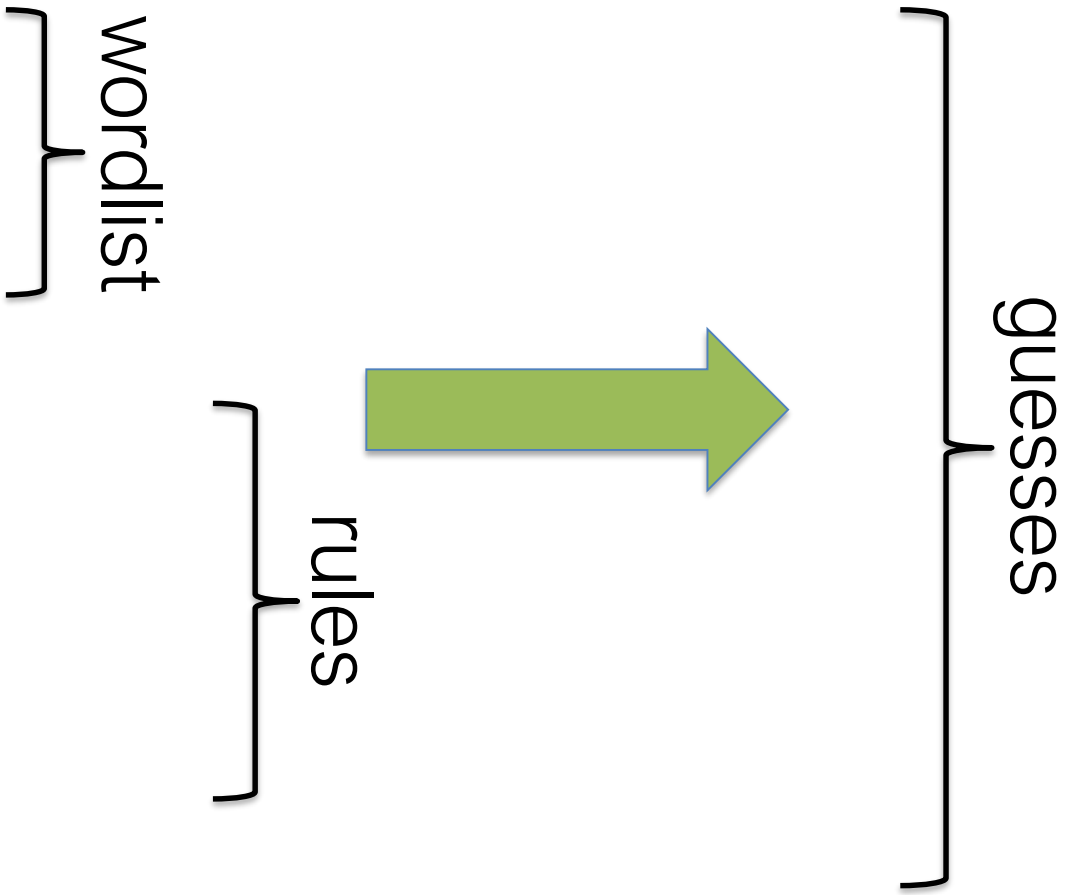
Five Cracking Approaches

- John the Ripper
- Hashcat
- Markov models
- Probabilistic Context-Free Grammar
- Professionals

John the Ripper

- Guesses variants of input wordlist
- Wordlist mode requires:
  - Wordlist (passwords and dictionary entries)
  - Mangling rules
- Speed: Fast
  - 10^{13} guesses
- “JTR”

John the Ripper



John the Ripper

usenix

security

wordlist

rules

guesses

John the Ripper

usenix

security

wordlist

[ ]

[add 1 at end]

[change e to 3]

rules

guesses

John the Ripper

usenix

security

wordlist

[ ]

[add 1 at end]

[change e to 3]

rules

usenix

security

usenix1

security1

us3nix

s3curity

guesses

John the Ripper

usenix

security

wordlist

[ ]

[add 1 at end]

[change e to 3]

rules

usenix

security

usenix1

security1

us3nix

s3curity

guesses

John the Ripper

usenix
security

wordlist

[ ]
[add 1 at end]
[change e to 3]

rules

usenix
security
usenix1
security1
us3nix
s3curity

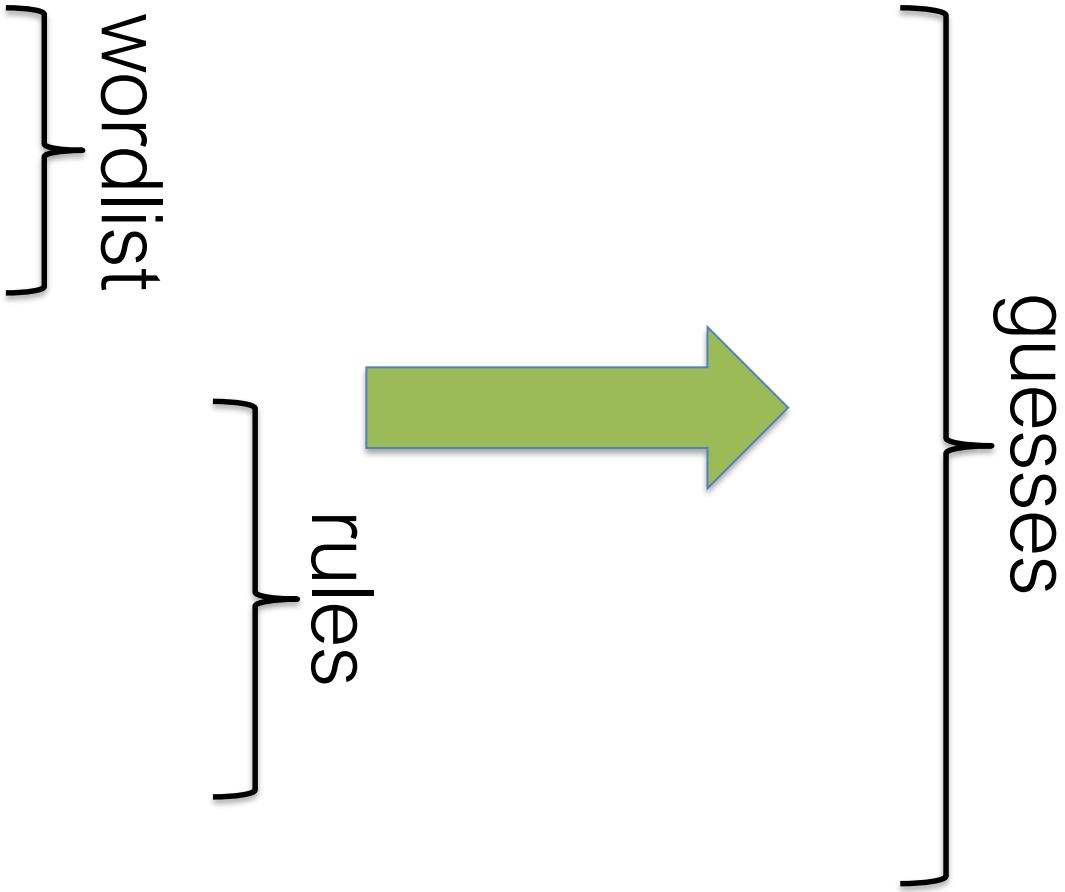
guesses

Hashcat

- Guesses variants of input wordlist
- Wordlist mode requires:
  - Wordlist (passwords and dictionary entries)
  - Mangling rules
- Speed: Fast
  - 10^{13} guesses

Hashcat


hashcat
advanced
password
recovery


wordlist
rules
guesses

Hashcat

usenix

security

wordlist

[ ]

[add 1 at end]

[change e to 3]

rules

guesses

Hashcat

usenix

security

wordlist

[ ]

[add 1 at end]

[change e to 3]

rules

usenix

usenix1

us3nix

security

security1

s3curity

guesses

Hashcat

usenix

security

wordlist

[ ]

[add 1 at end]

[change e to 3]

rules

usenix

usenix1

us3nix

security

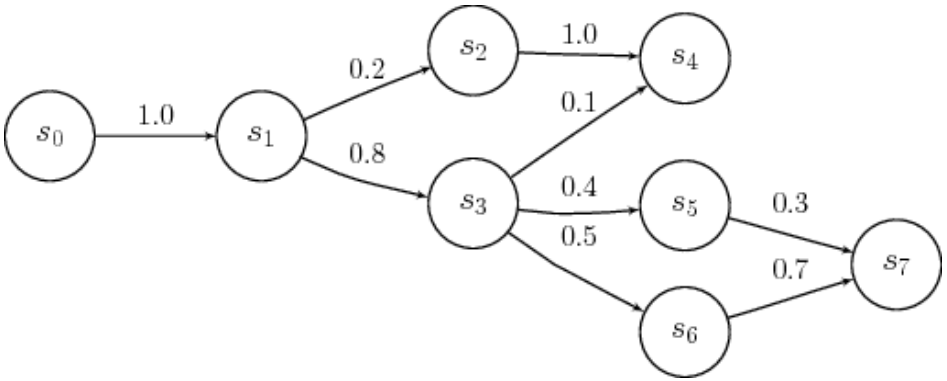
security1

s3curity

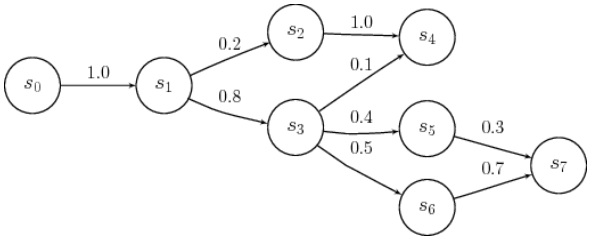
guesses

Markov Models

- Predicts future characters from previous
- Approach requires weighted data:
  - Passwords
  - Dictionaries
- Ma et al. IEEE S&P 2014
- Speed: Slow
  - 10^{10} guesses

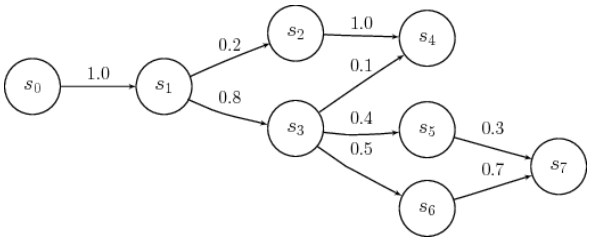


Markov Models



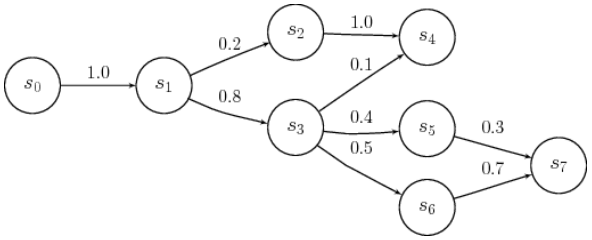
usenixsecurity

Markov Models



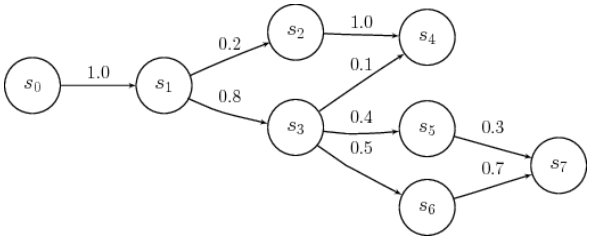
usenixsecurity

Markov Models



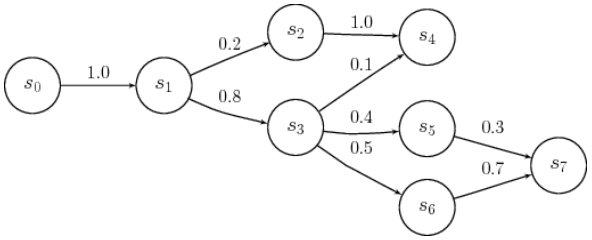
usenixsecurity

Markov Models



usenixsecurity

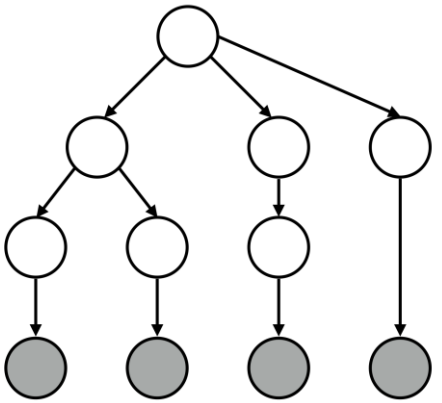
Markov Models



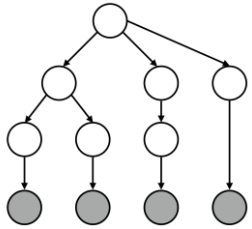
usenixsecurity

Probabilistic Context-Free Grammar

- Generate password grammar
  - Structures
  - Terminals
- Kelley et al. IEEE S&P 2012
  - Based on Weir et al. IEEE S&P 2009
- Speed: ~~Slow~~ Medium
  - 10^{14} guesses
- “PCFG”



PCFG



passwordpassword

password123

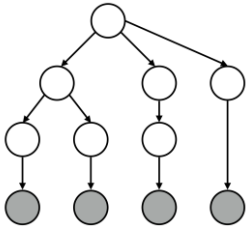
usenix3

5ecurity

iloveyou

nirvana123

PCFG



passwordpassword

password123

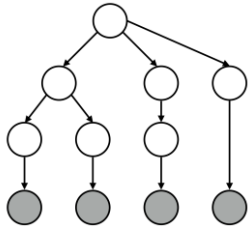
usenix3

5ecurity

iloveyou

nirvana123

PCFG



passwordpassword

password123

usenix3

5ecurity

iloveyou

nirvana123

Professionals (“Pros”)

- Contracted KoreLogic
  - Password audits for Fortune 500 companies
  - Run DEF CON “Crack Me If You Can”
- Proprietary wordlists and configurations
  - 10^{14} guesses
  - Manually tuned, updated

KoreLogic SECURITY

Approach

4 password sets

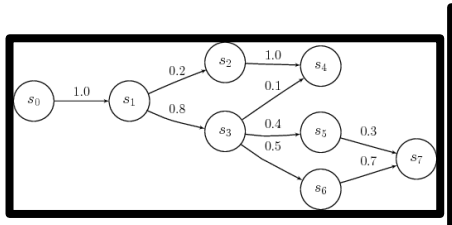
```
password
iloveyou
teamo123
…
```

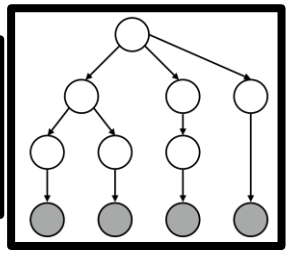
```
passwordpassword
12345678123456 78
!1@2#3$4%5^6&7*8
…
```

```
Pa$$w0rd
iLov3you!
1QaZ2W@x
…
```

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

✖ 5 approaches



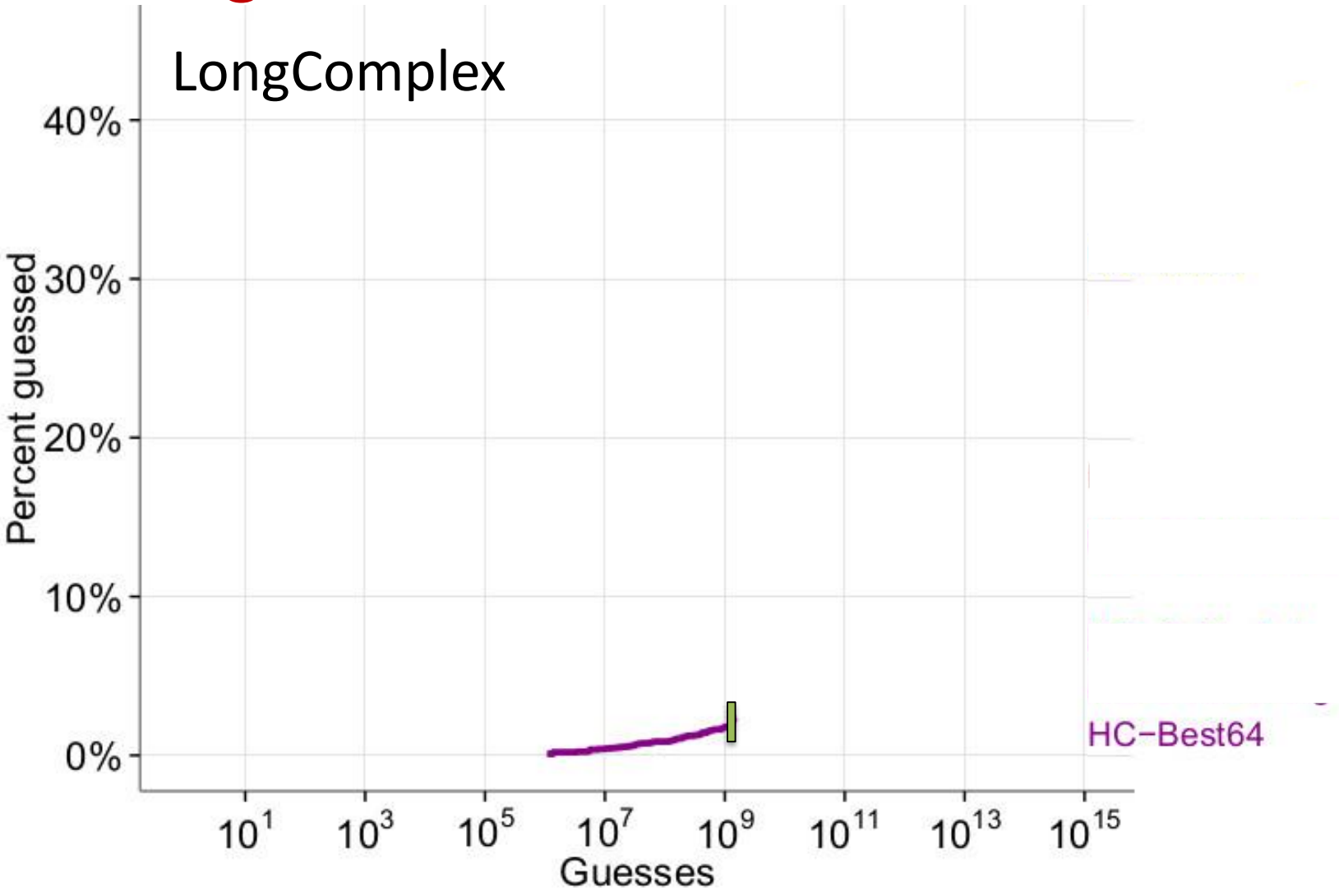


KoreLogic SECURITY

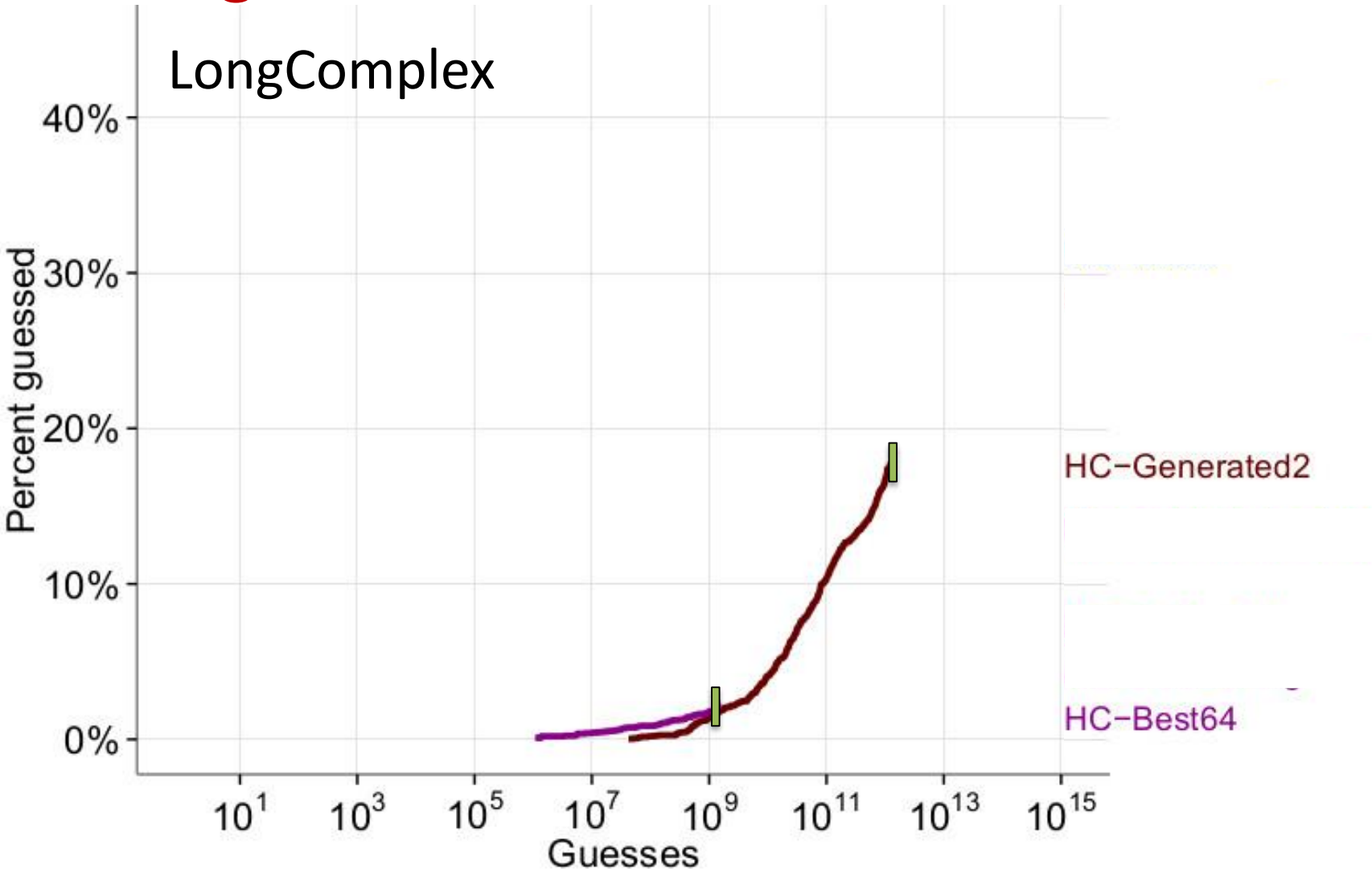
Outline of Results

- Importance of Configuration
- Comparison of Approaches
- Impact on Research Analyses

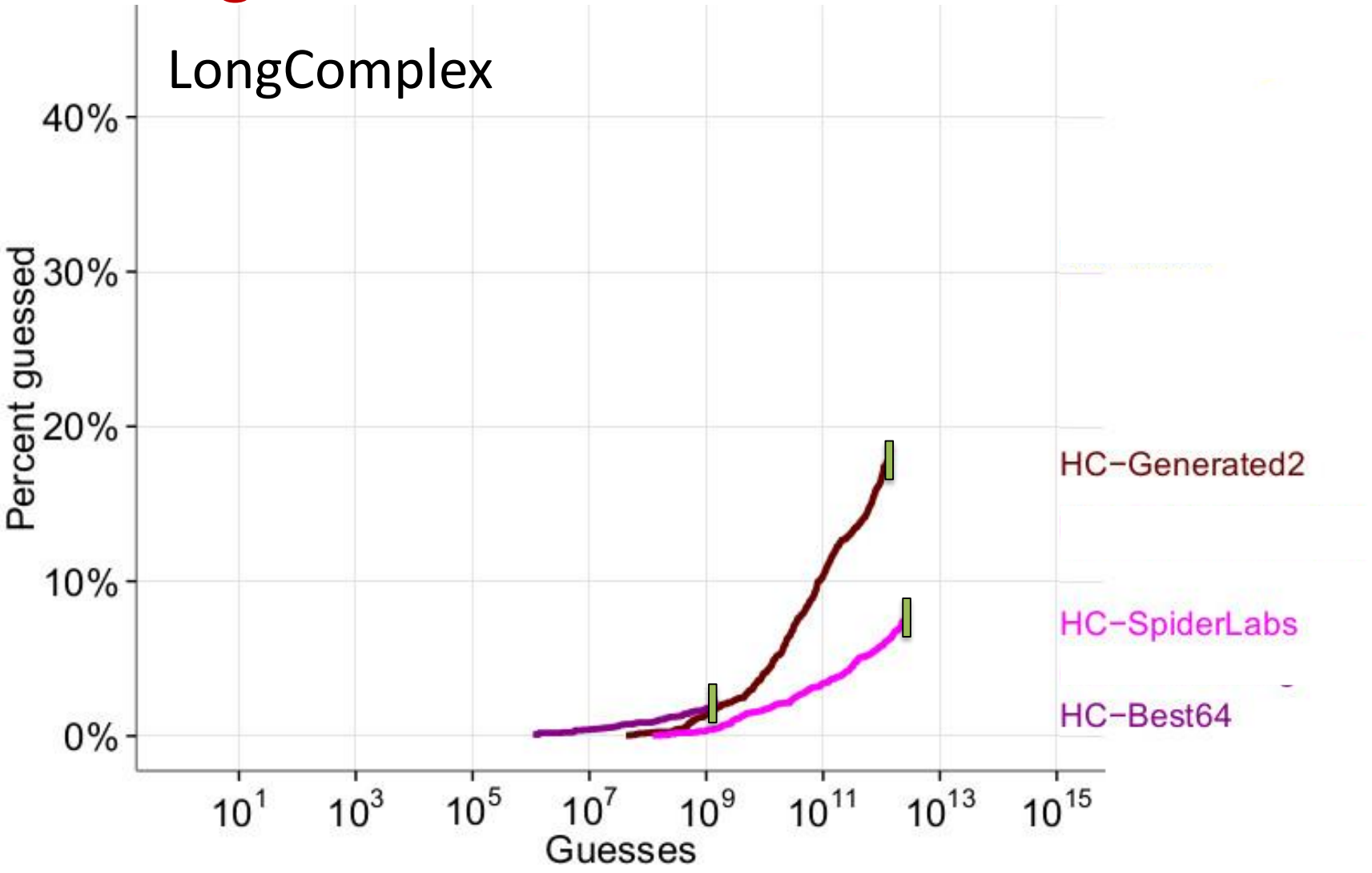
Configuration Is Crucial



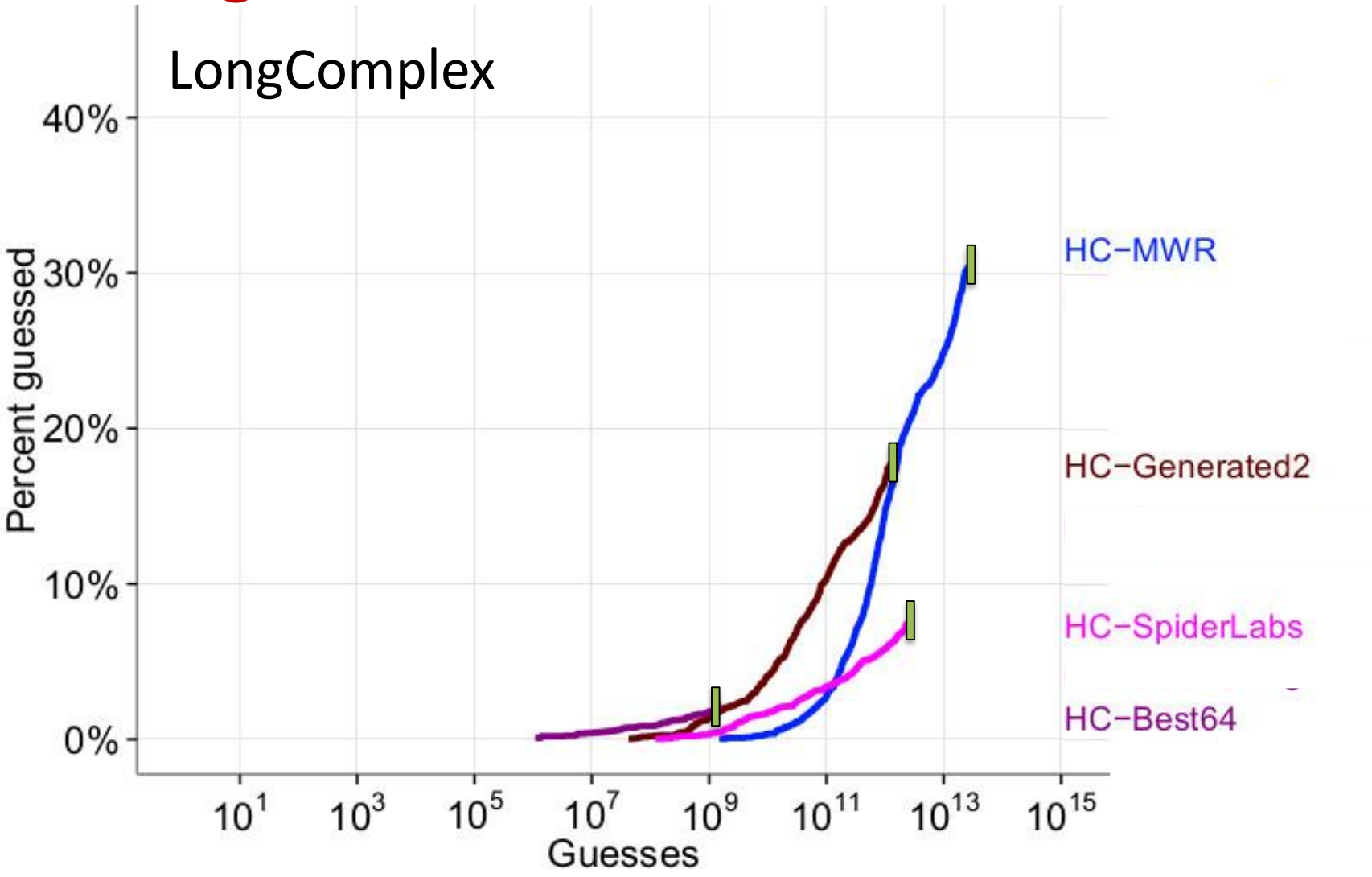
Configuration Is Crucial



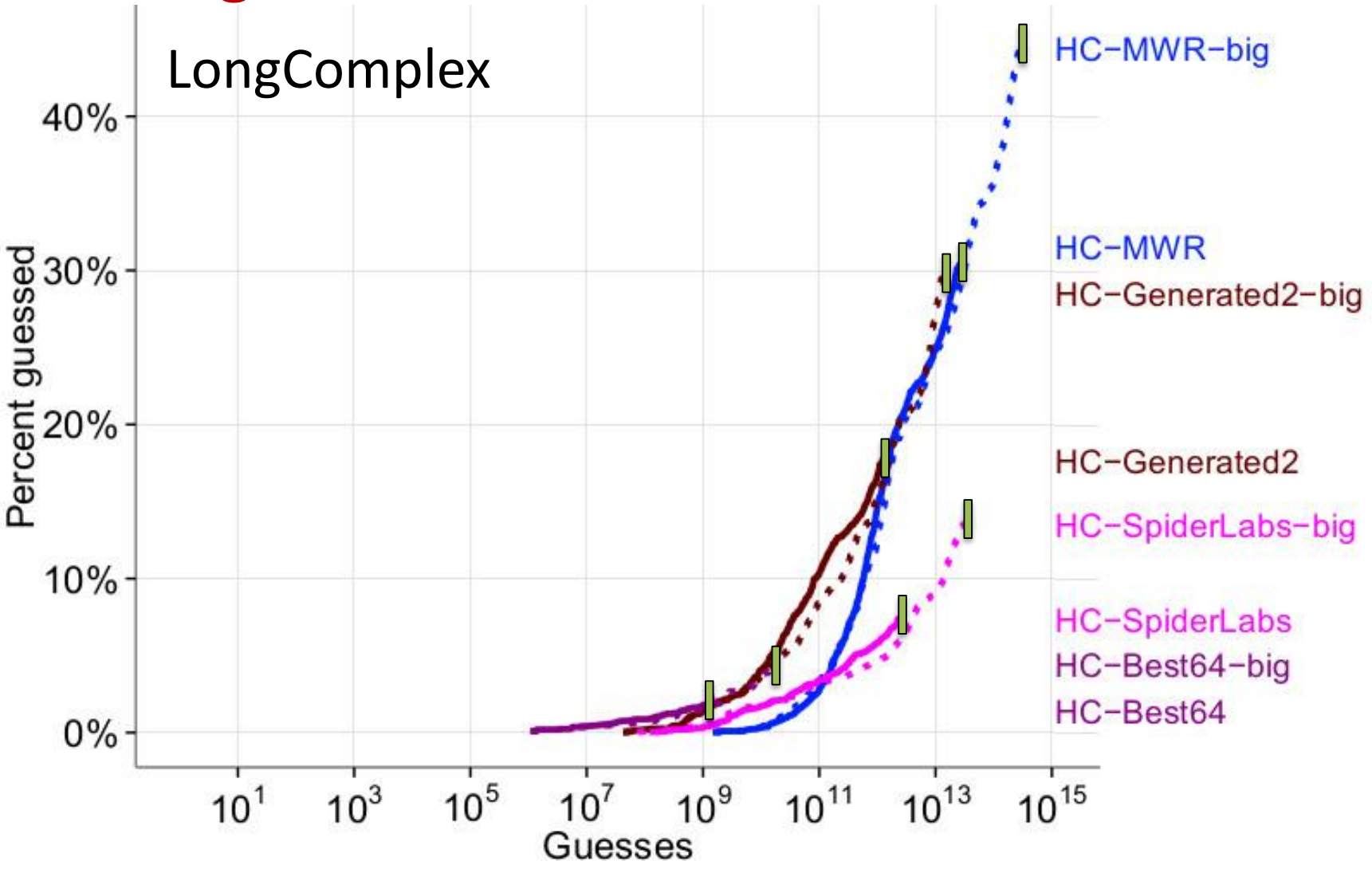
Configuration Is Crucial



Configuration Is Crucial



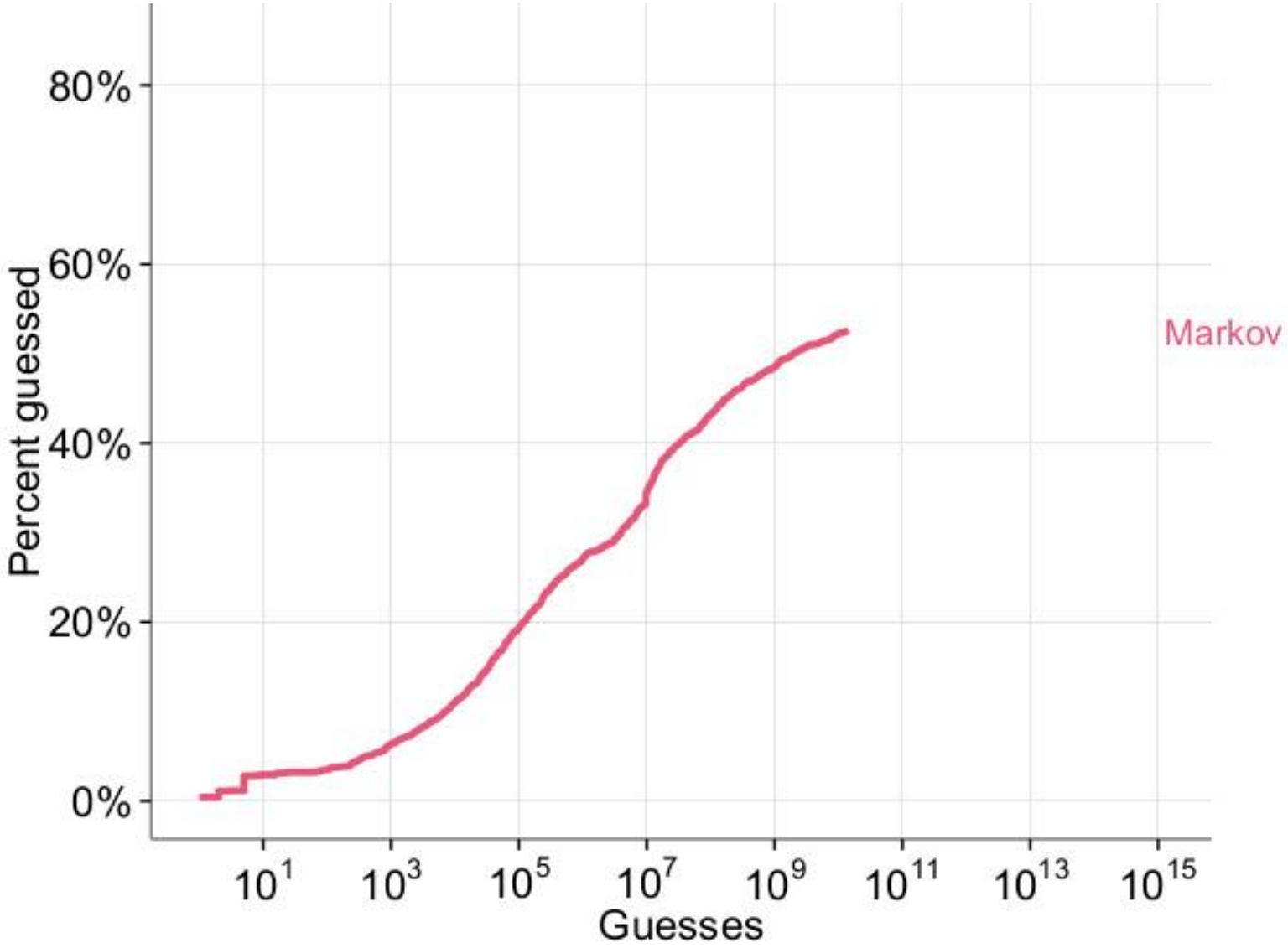
Configuration Is Crucial



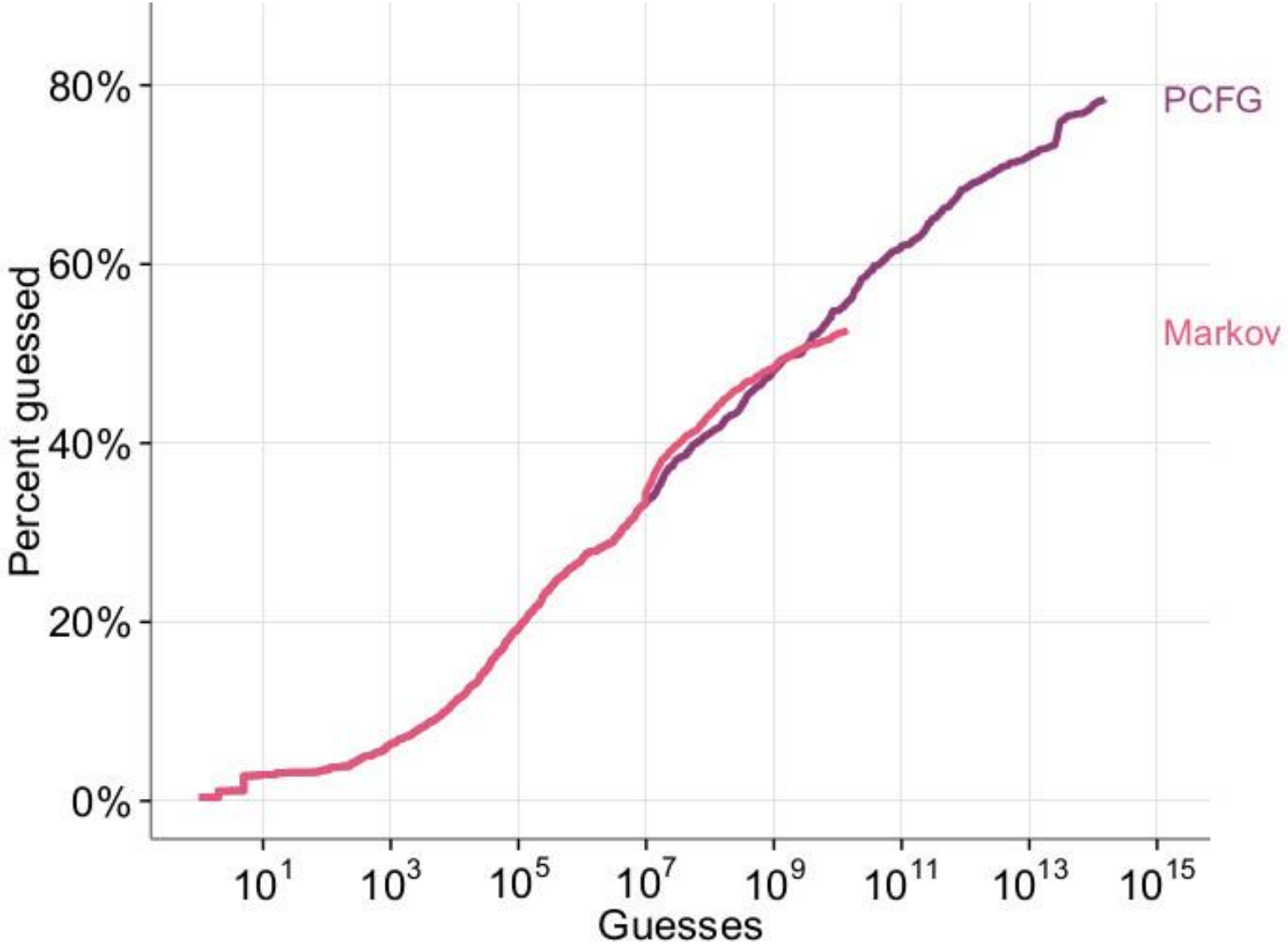
Outline of Results

- Importance of Configuration
- Comparison of Approaches
- Impact on Research Analyses

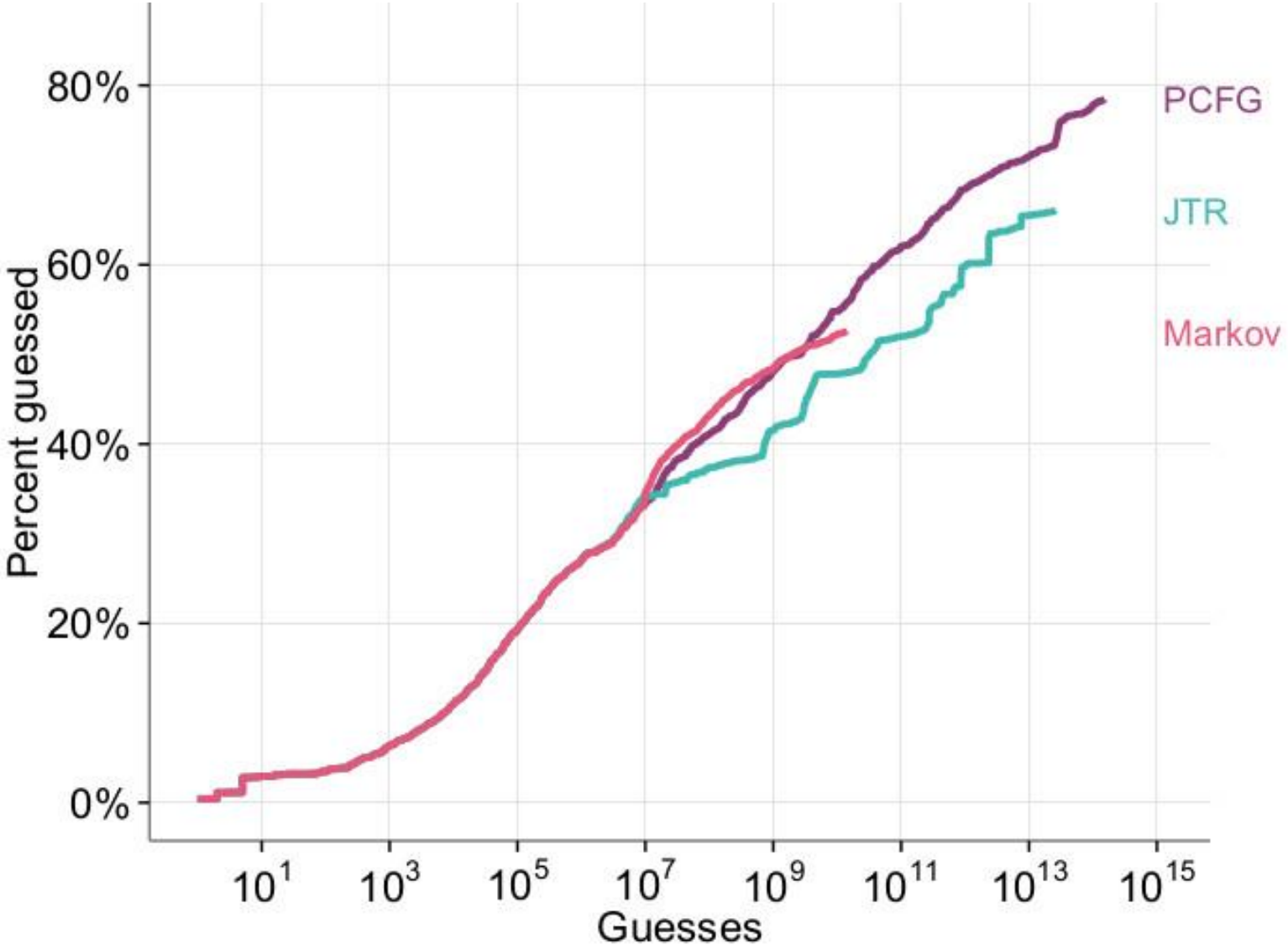
Comparison for Basic Passwords



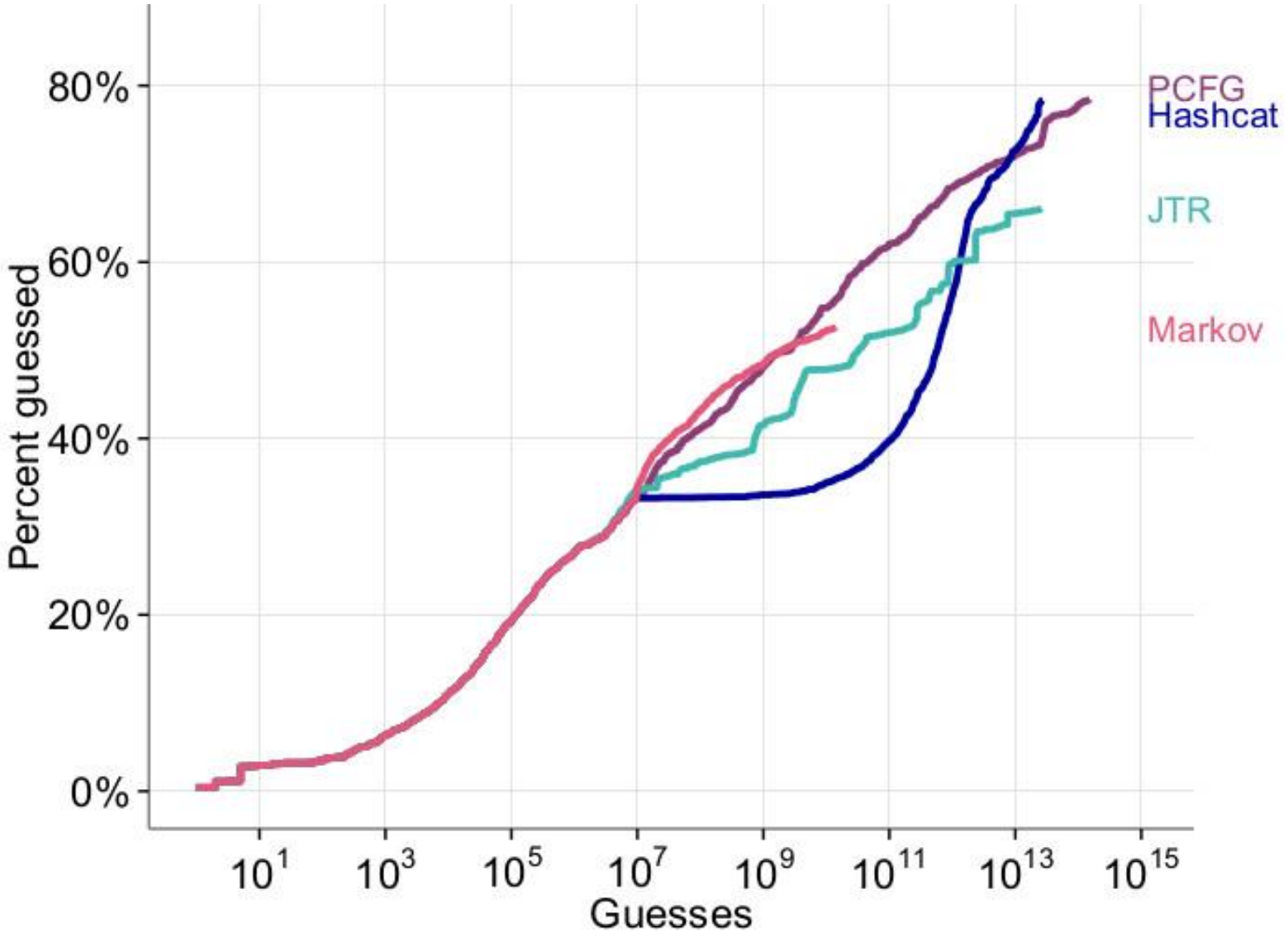
Comparison for Basic Passwords



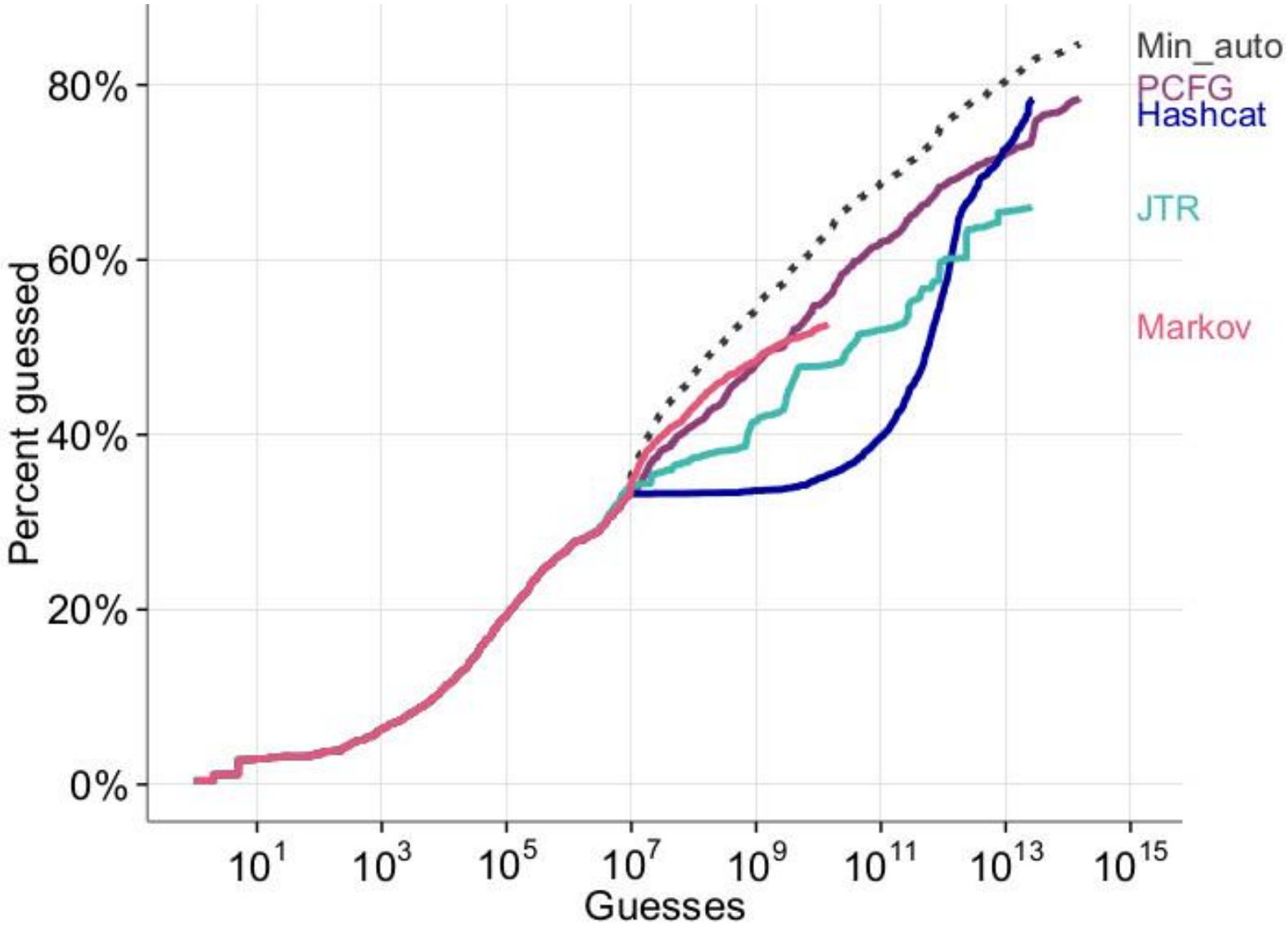
Comparison for Basic Passwords



Comparison for Basic Passwords

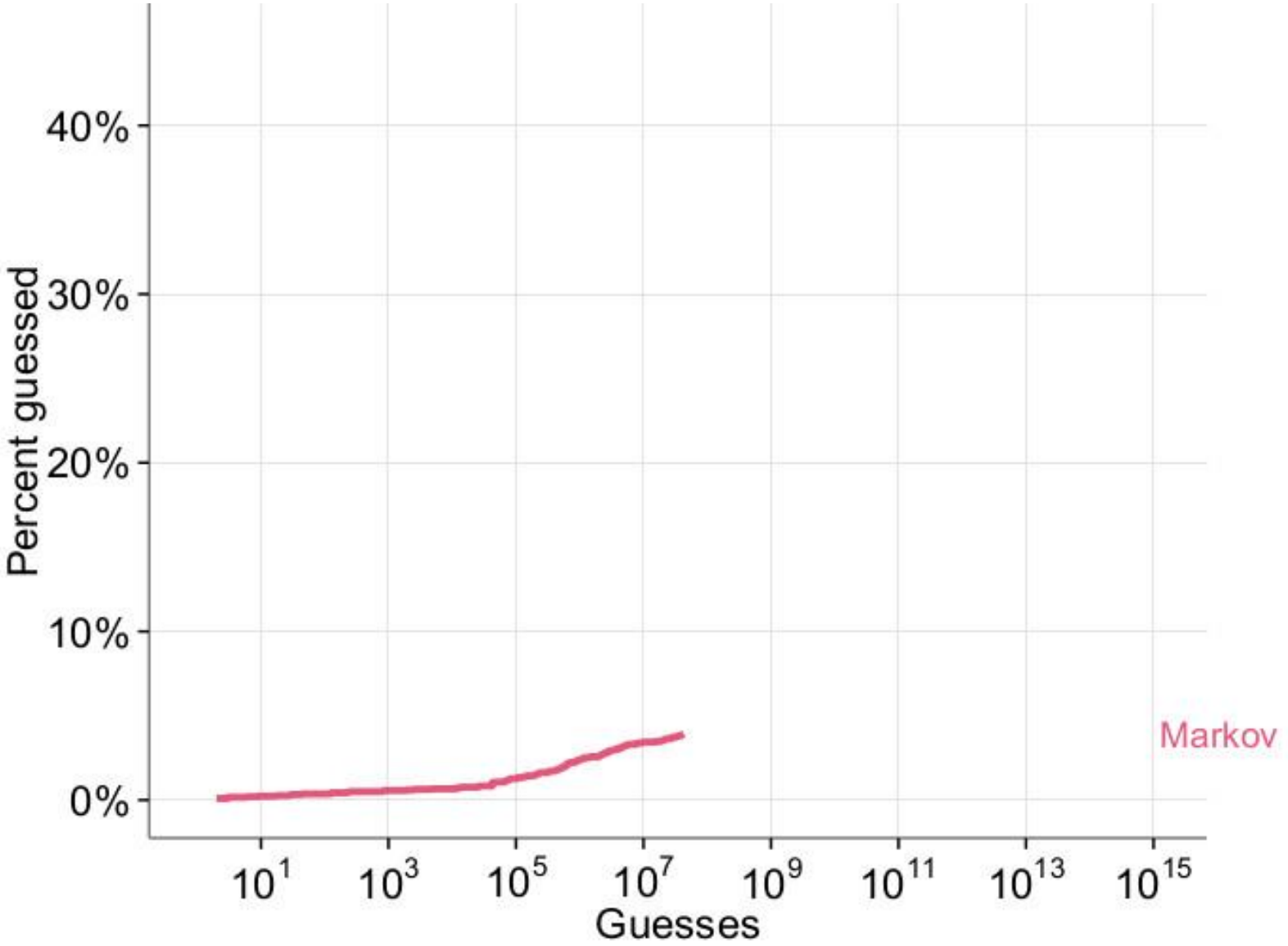


Comparison for Basic Passwords

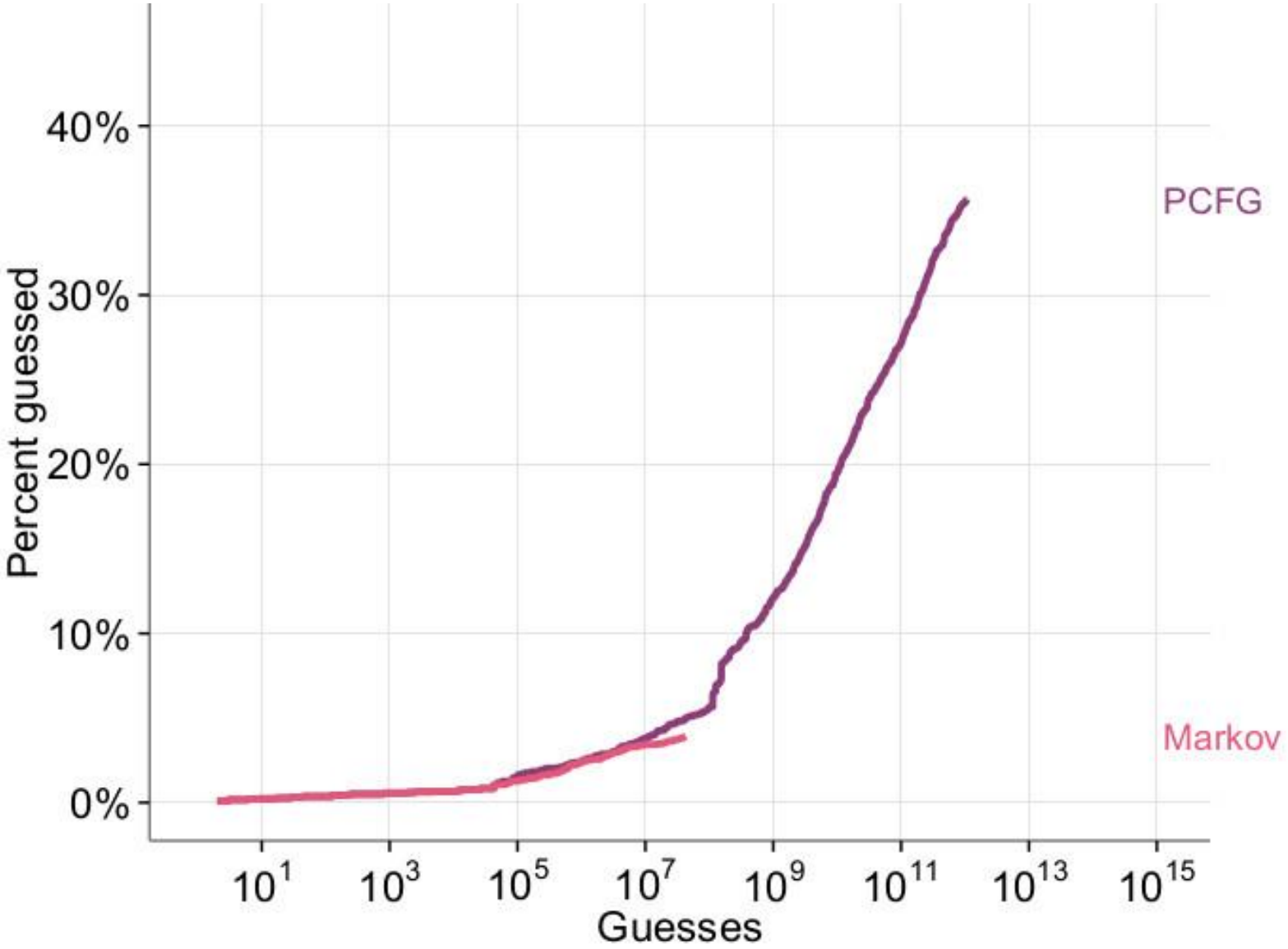


Comparison for Complex Passwords

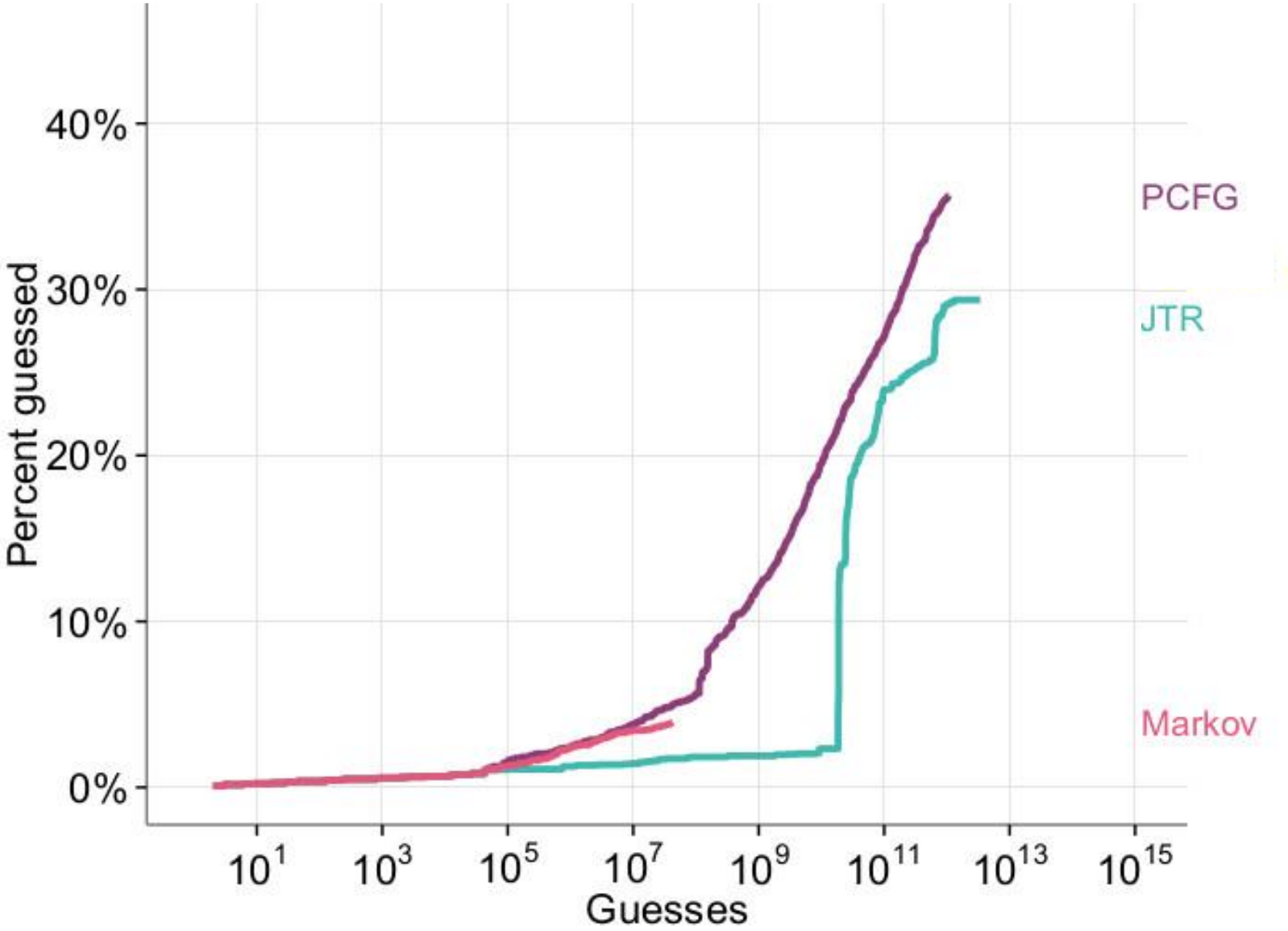
Comparison for Complex Passwords



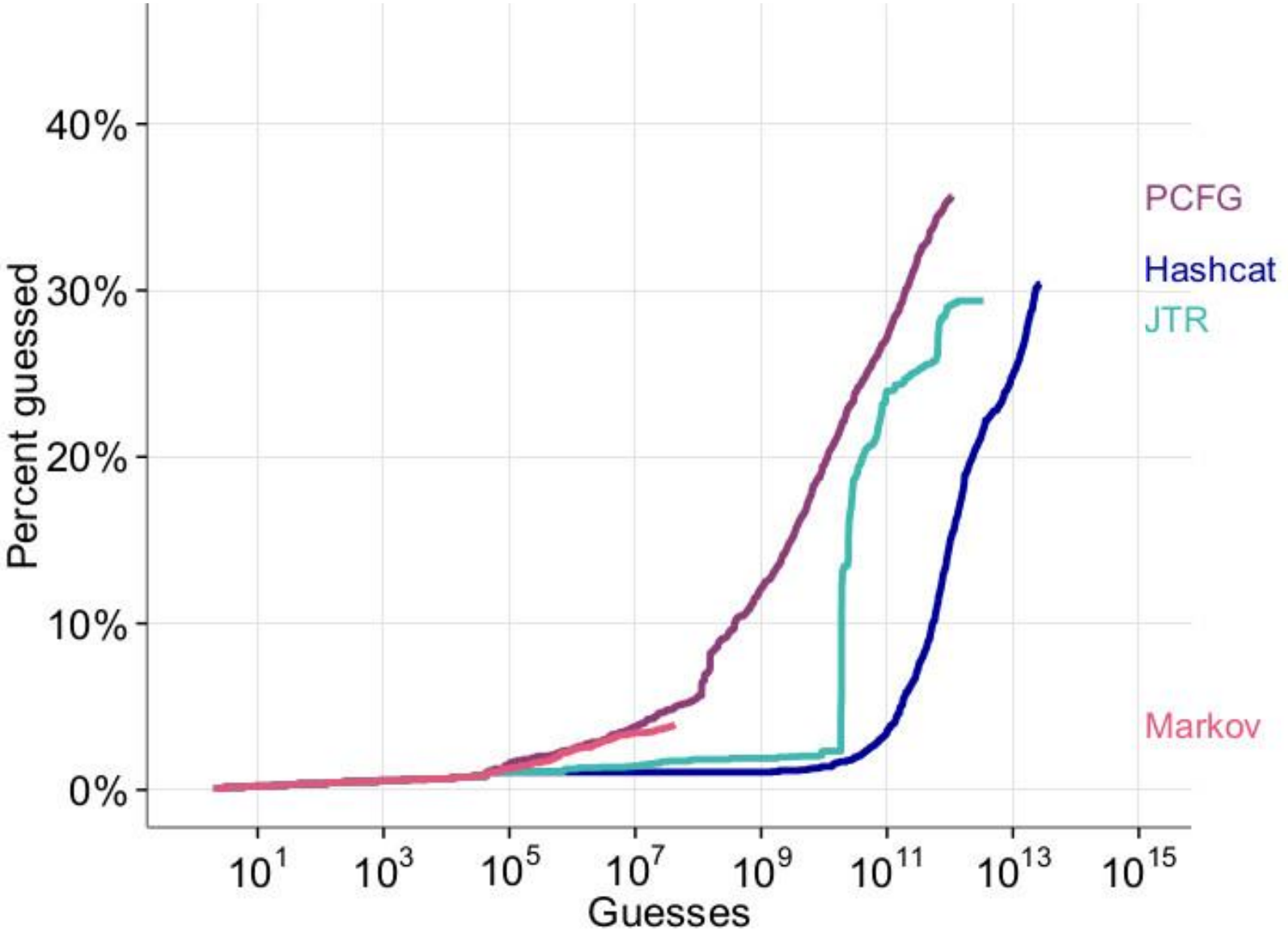
Comparison for Complex Passwords



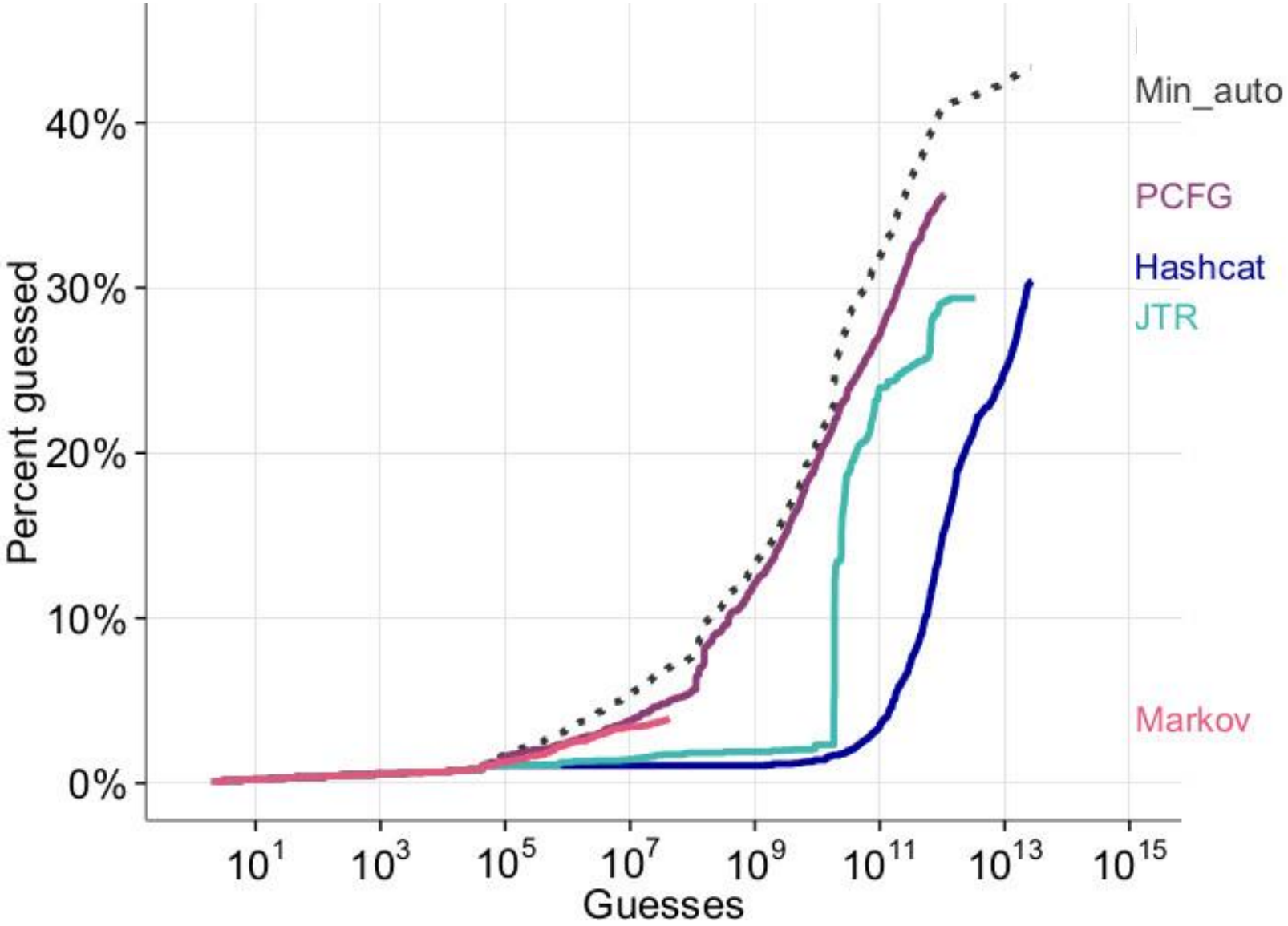
Comparison for Complex Passwords



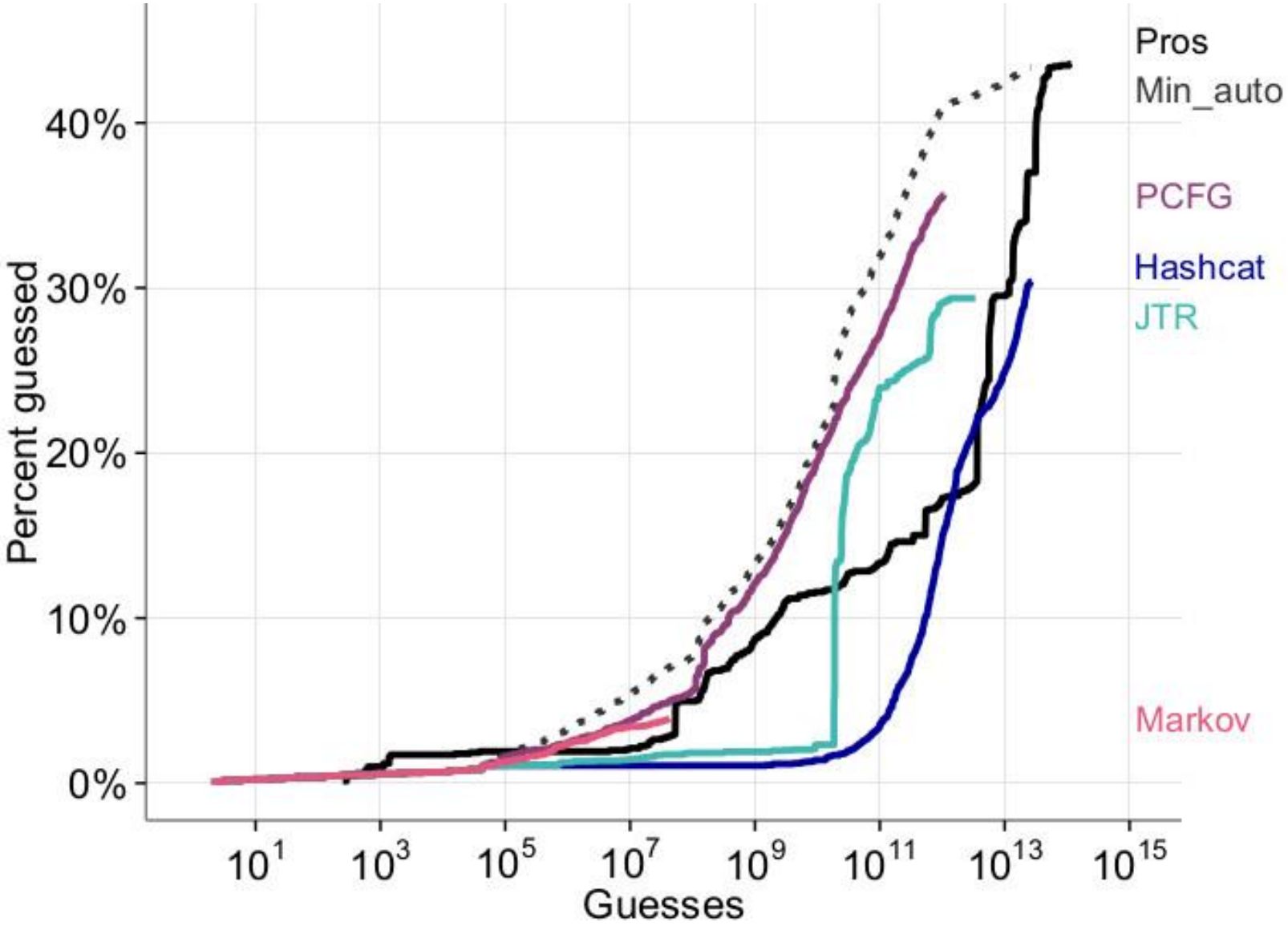
Comparison for Complex Passwords



Comparison for Complex Passwords



Comparison for Complex Passwords



Per-Password Highly Impacted

P@ssw0rd!

Per-Password Highly Impacted

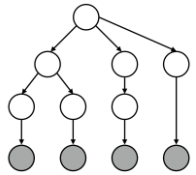
- JTR guess # 801 


```
P@ssw0rd!
```

Per-Password Highly Impacted

- JTR guess # 801

- Not guessed in 10^{14} PCFG guesses

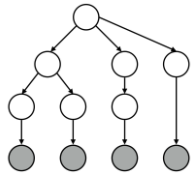


P@ssw0rd!

Per-Password Highly Impacted

- JTR guess # 801

- Not guessed in 10^{14} PCFG guesses



P@ssw0rd!

How Do We Help Users Make Better Passwords?

Problem 1: Bad Advice

Carnegie Mellon University

Password Requirements

Must Contain

- At least 8-characters.
- At least one uppercase alphabetic character (e.g., A-Z).
- At least one lowercase alphabetic character (e.g., a-z).
- At least one number (e.g., 0-9).
- At least one special character (e.g., []~!@#$%^&*()?<>./_-+=).

Cannot Contain

- Known information (i.e., first name, last name, Andrew userID, date of birth, 9-digit Carnegie Mellon ID number, SSN, job title).
- Four or more occurrences of the same character (e.g., aaaa, 2222, a123a345a678a).*
- A word that is found in a standard dictionary.* (after removing non-alpha characters).

****This requirement does not apply to Andrew account passwords that are more than 19 characters in length (e.g., passphrase).***

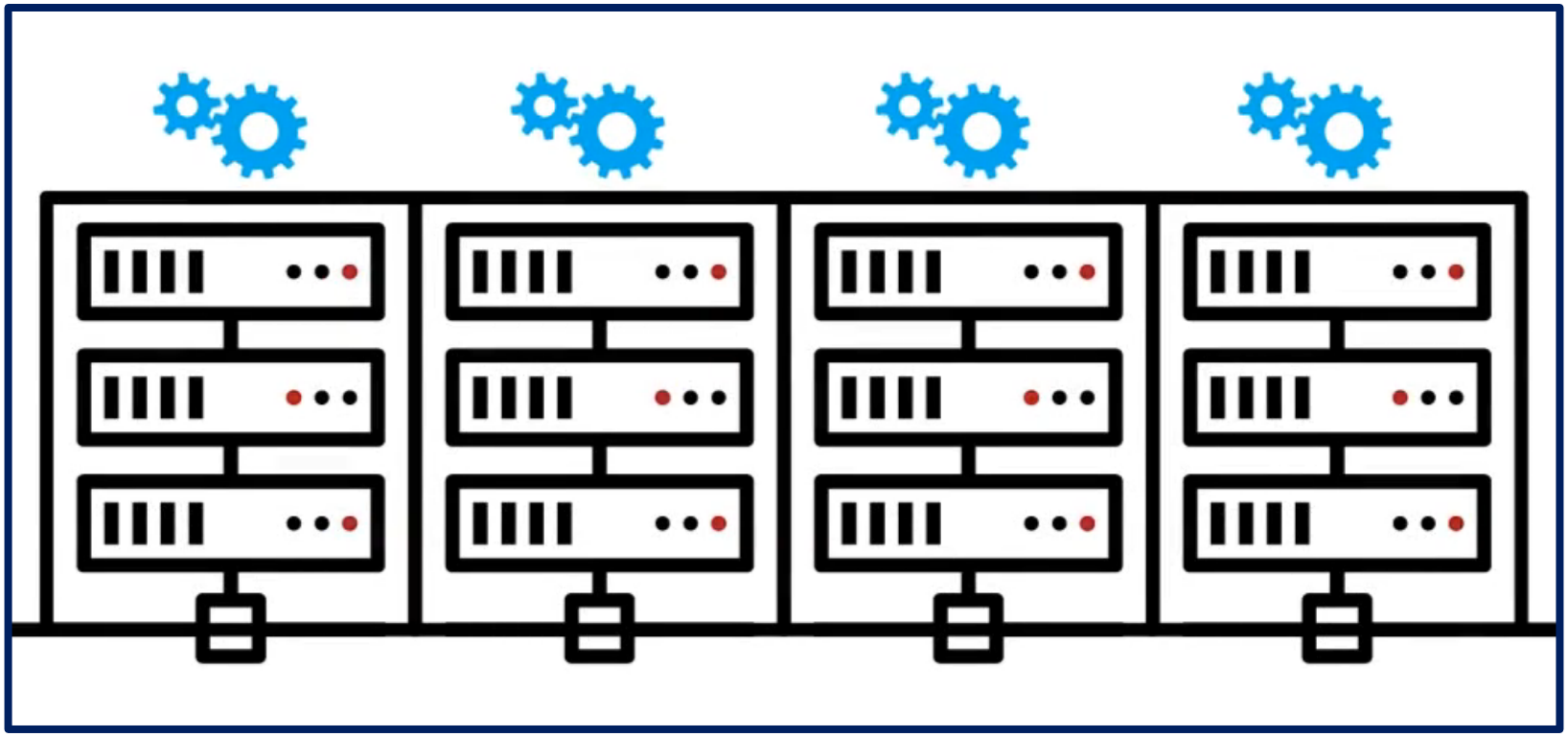
Additional Policies

- Last five passwords cannot be used.
- Cannot be changed more than four times in a day.

Problem 2: Inaccurate Feedback

Problem 3: Unhelpful Feedback

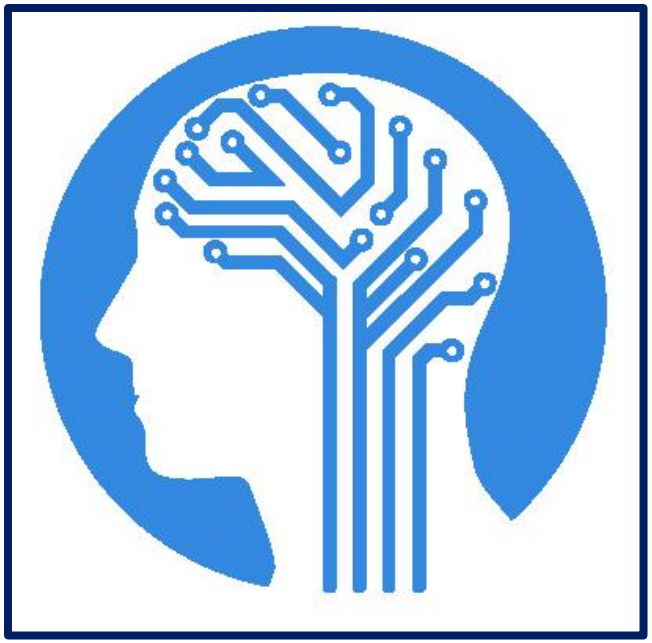
Better Password Scoring



William Melicher, Blase Ur, Sean M. Segreti, Saranga Komanduri, Lujo Bauer, Nicolas Christin, Lorrie Faith Cranor. Fast, Lean, and Accurate: Modeling Password Guessability Using Neural Networks. In *Proc. USENIX Security Symposium*, 2016.

Better Password Scoring

- Real-time feedback
- Runs entirely client-side
- Accurately models password guessability



Recurrent Neural Networks (RNNs)

LSTM Architecture

Generating Passwords

Generating Passwords

`passw`  o or maybe 0 or O or ...

Generating Passwords

`passw` 

Next char is:

A: 3%

B: 1%

C: 0.6%

…

O: 55%

…

Z: 0.01%

0: 20%

1: …

Generating Passwords

“”

Prob: 100%

Next char is:

A:	3%
B:	2%
C:	5%
…	
O:	2%
…	
Z:	0.2%
0:	1%
1:	…
END:	2%

Generating Passwords

“”

Prob: 100%

Next char is:

A:	3%
B:	2%
C:	5%
…	
O:	2%
…	
Z:	0.2%
0:	1%
1:	…
END:	2%

Generating Passwords

“C”

Prob: 5%

Generating Passwords

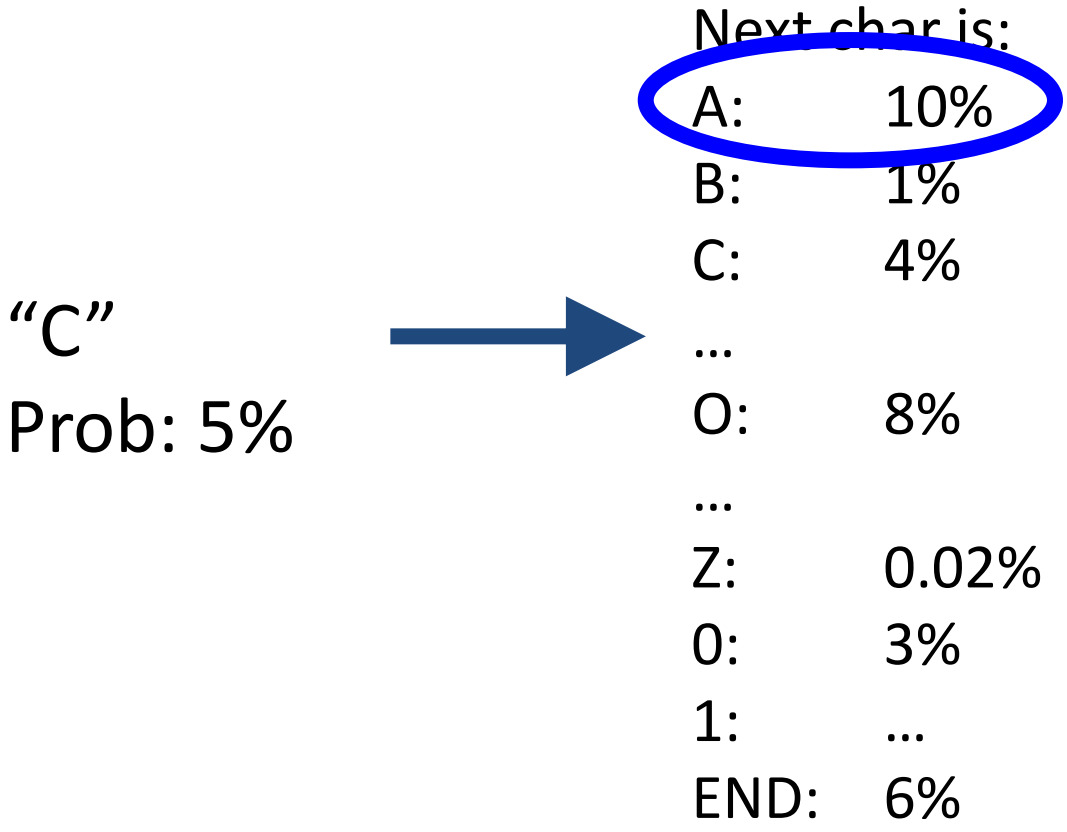
“C”

Prob: 5%

Next char is:

A:	10%
B:	1%
C:	4%
…	
O:	8%
…	
Z:	0.02%
0:	3%
1:	…
END:	6%

Generating Passwords



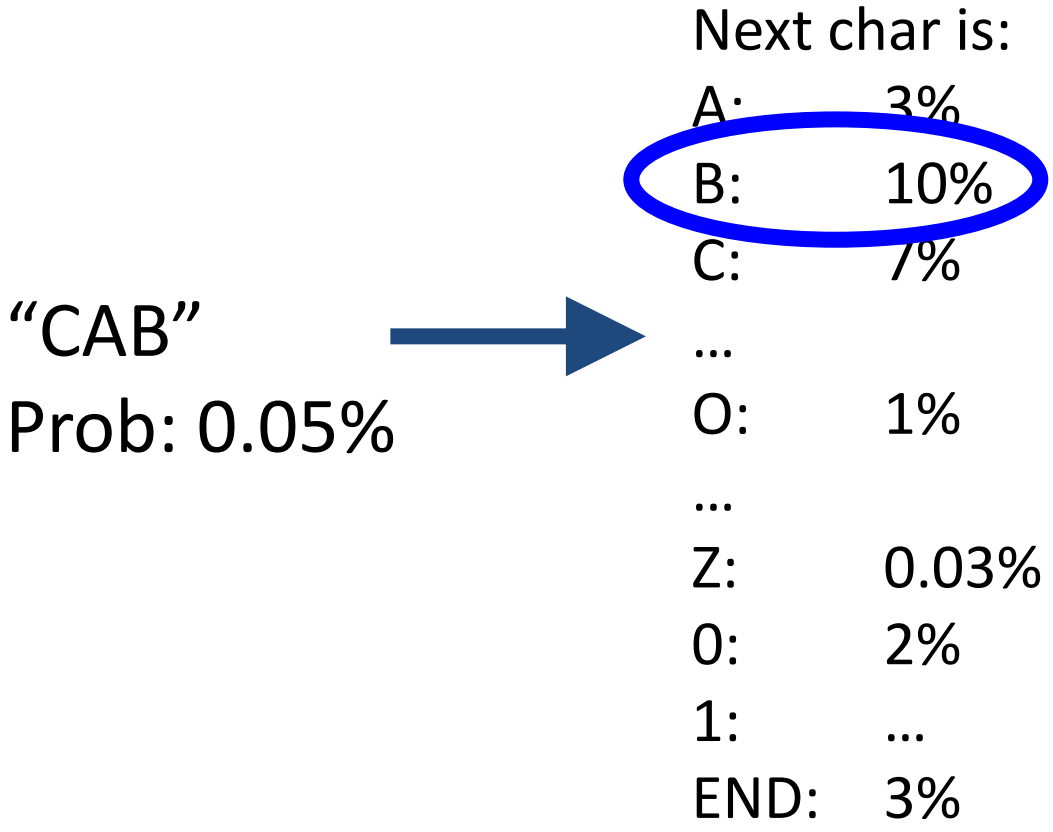
Generating Passwords

"CA"
Prob: 0.5%

Next char is:

A:	3%
B:	10%
C:	7%
…	
O:	1%
…	
Z:	0.03%
0:	2%
1:	…
END:	12%

Generating Passwords



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%

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%

Generating Passwords

“CAB”

Prob: 0.006%

Descending Probability Order

CAB - 0.006%
CAC - 0.0042%
ADD1 - 0.002%
CODE - 0.0013%
…

Design Space

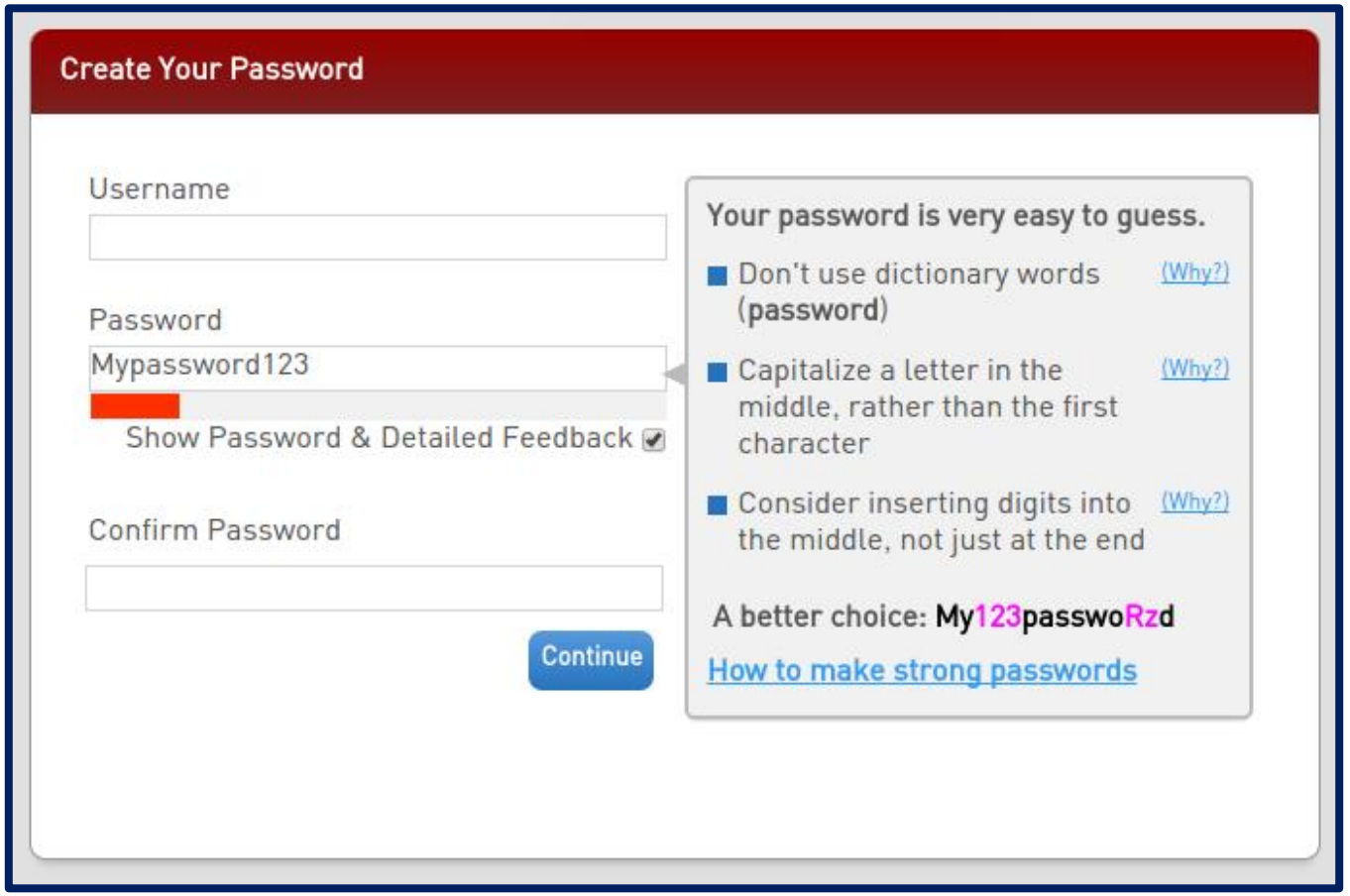
- Model size: 3mb (browser) vs. 60mb (GPU)
- Transference learning
  - Novel password-composition policies
- Training data
  - Natural language
- (Many others)

Key Results

- Neural networks produce better guesses than previous methods
- Larger model not a major advantage
- Browser implementation in Javascript

Intelligibility (Explanations)

Building a Data-Driven Meter



Blase Ur, Felicia Alfieri, Maung Aung, Lujo Bauer, Nicolas Christin, Jessica Colnago, Lorrie Faith Cranor, Henry Dixon, Pardis Emami Naeini, Hana Habib, Noah Johnson, William Melicher. Development and Evaluation of a Data-Driven Password Meter. In *Proc. CHI*, 2017.

We designed & tested a meter with:

1) Principled strength estimates

2) Data-driven feedback to users

We designed & tested a meter with:

1) Principled strength estimates (RNN)

2) Data-driven feedback to users

We designed & tested a meter with:

1) Principled strength estimates

2) Data-driven feedback to users

Provide Intelligible Explanations

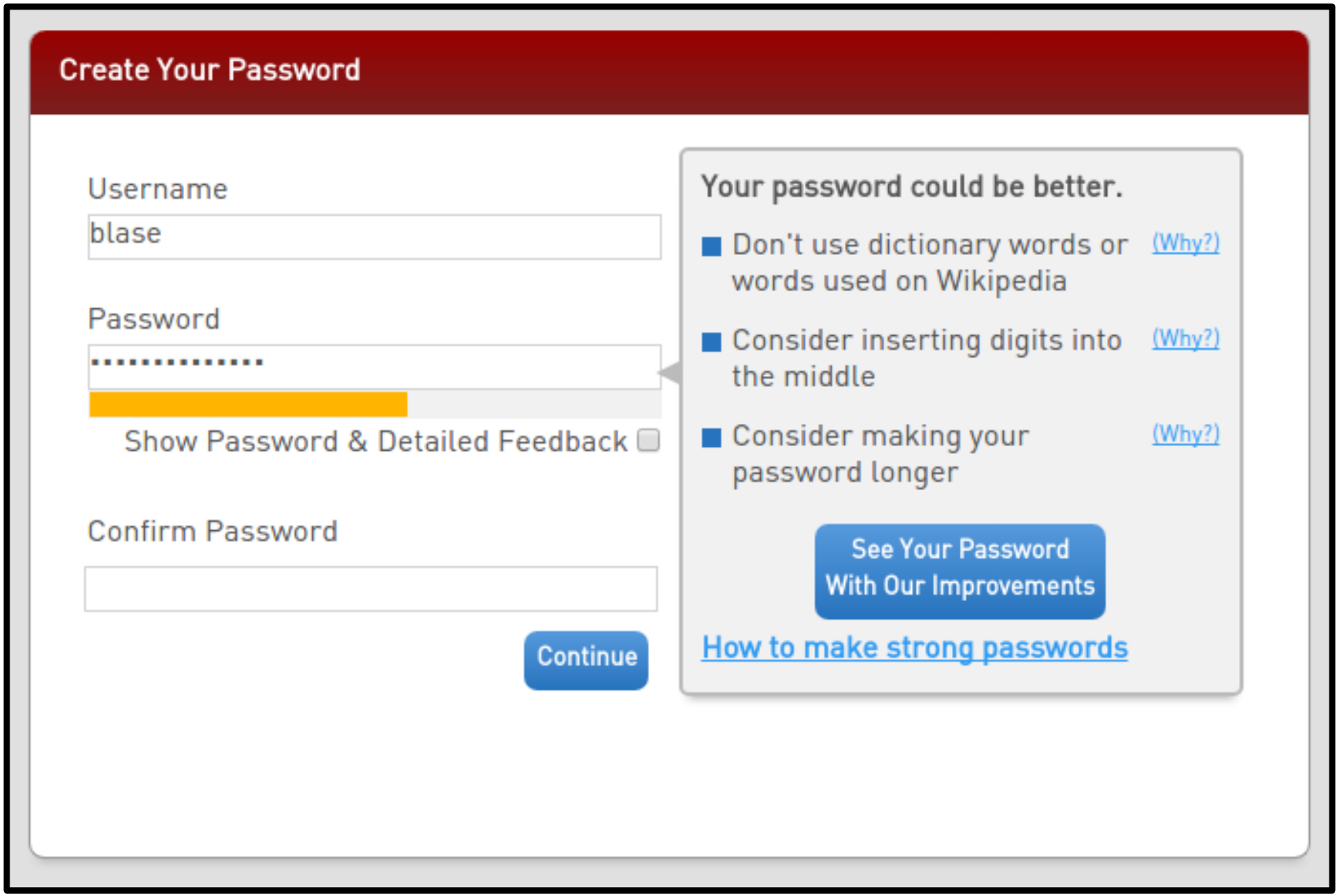
`Unic0rns`

Don't use simple transformations of words or phrases (**unicorns** → **Unic0rns**)

Capitalize a letter in the middle, rather than the first character

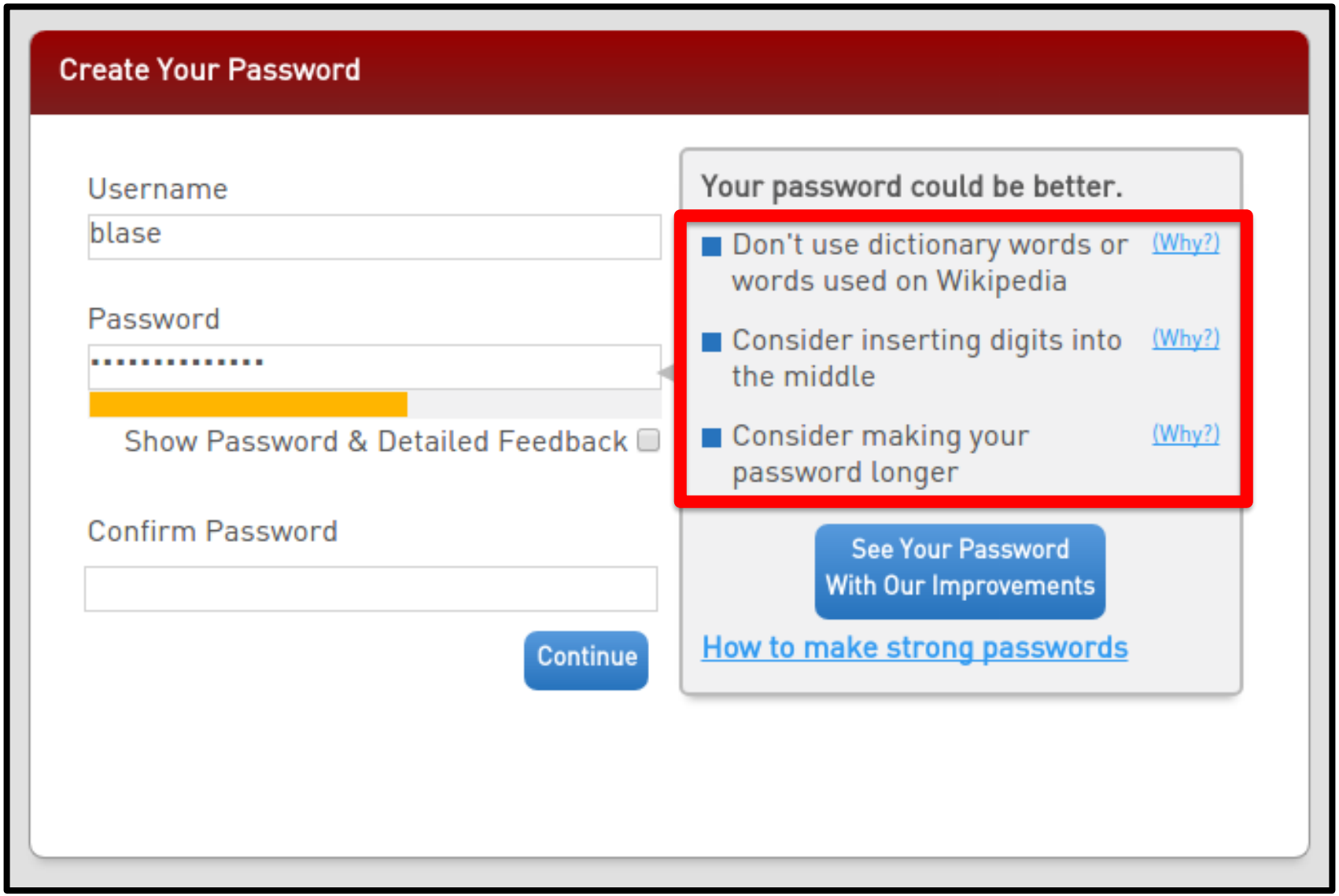
- 21 characteristics
- Weightings determined with regression

After Requirements Are Met…

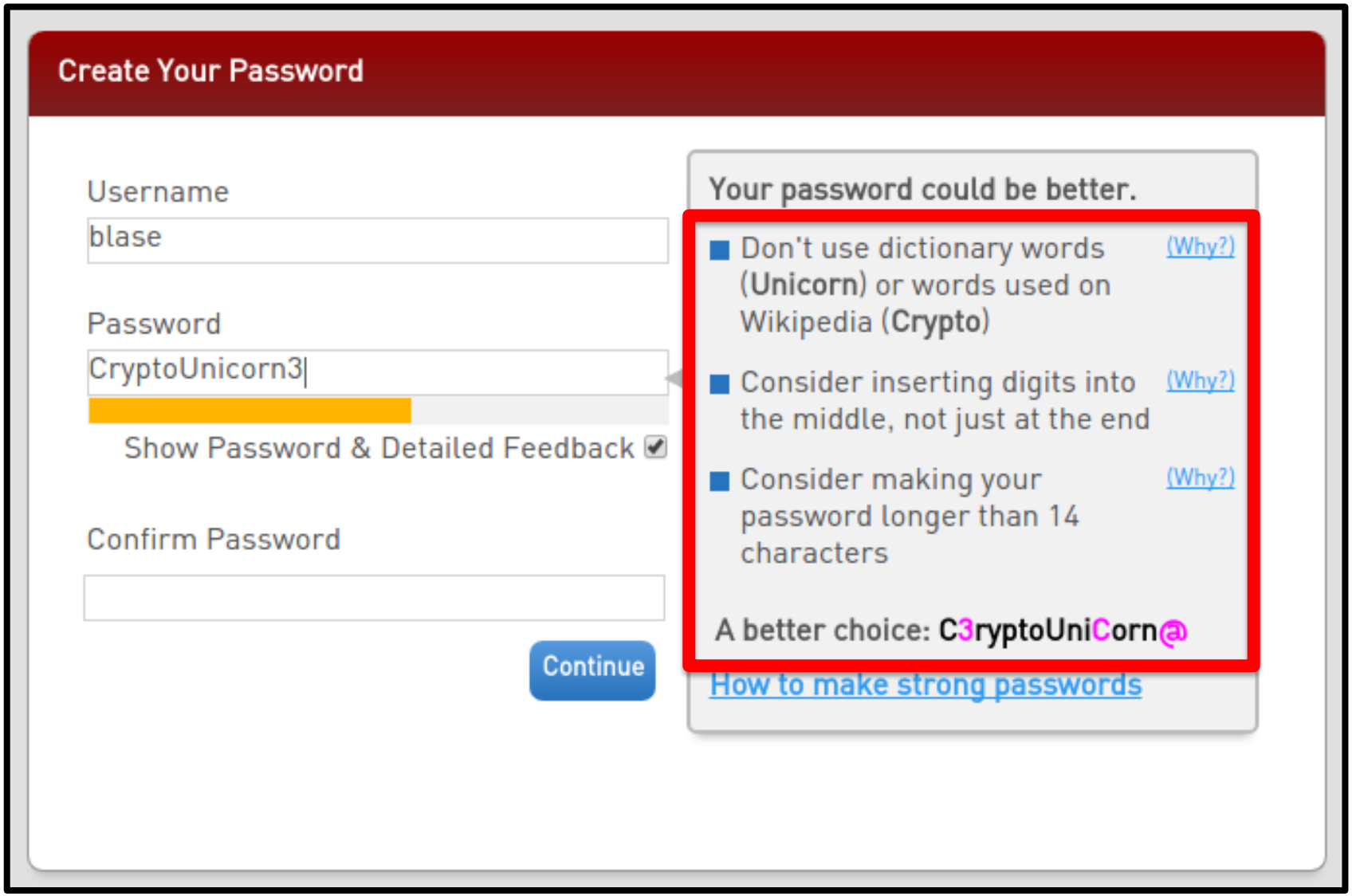


…Displays Score Visually

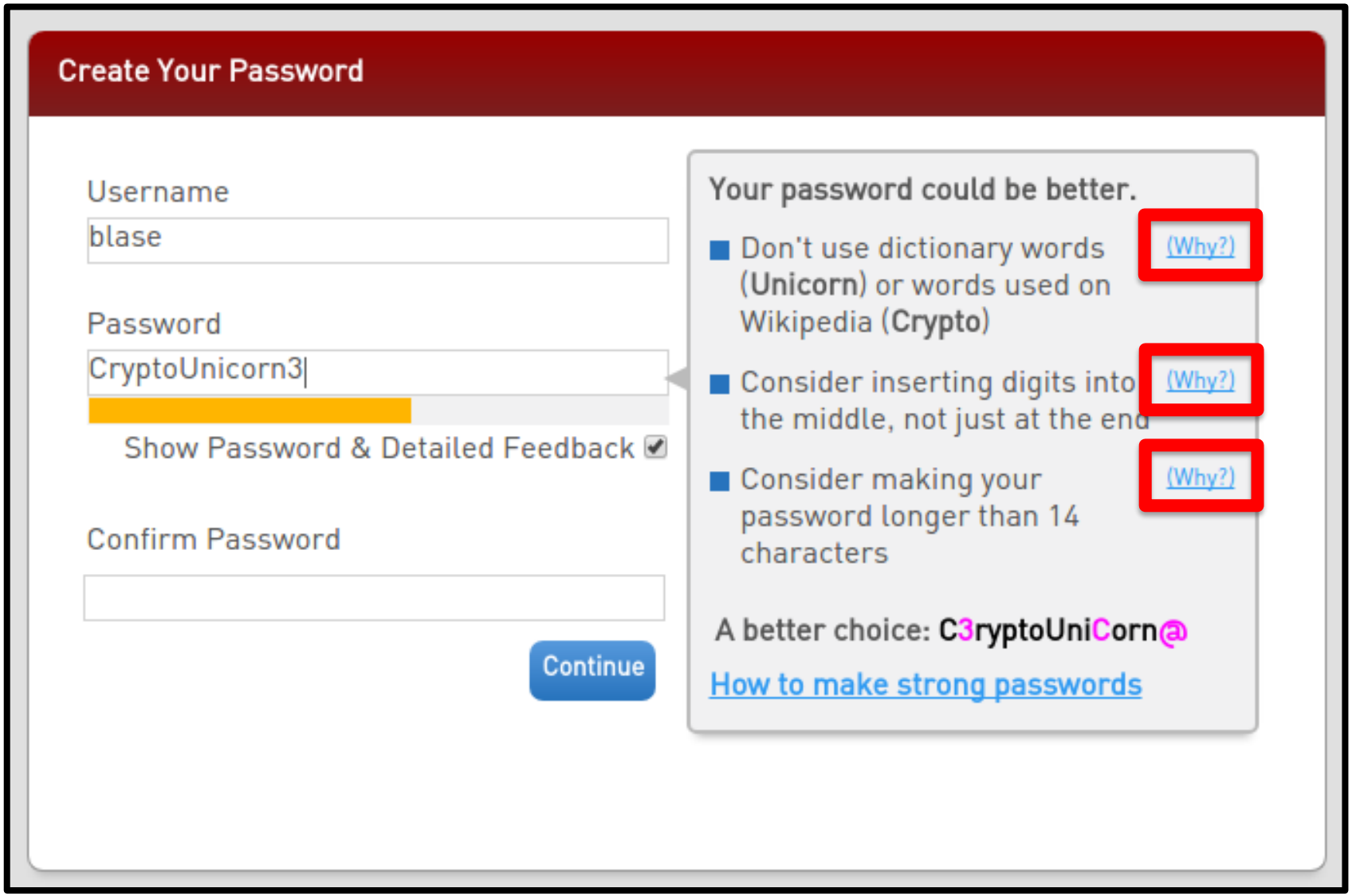
…Provides Text Feedback



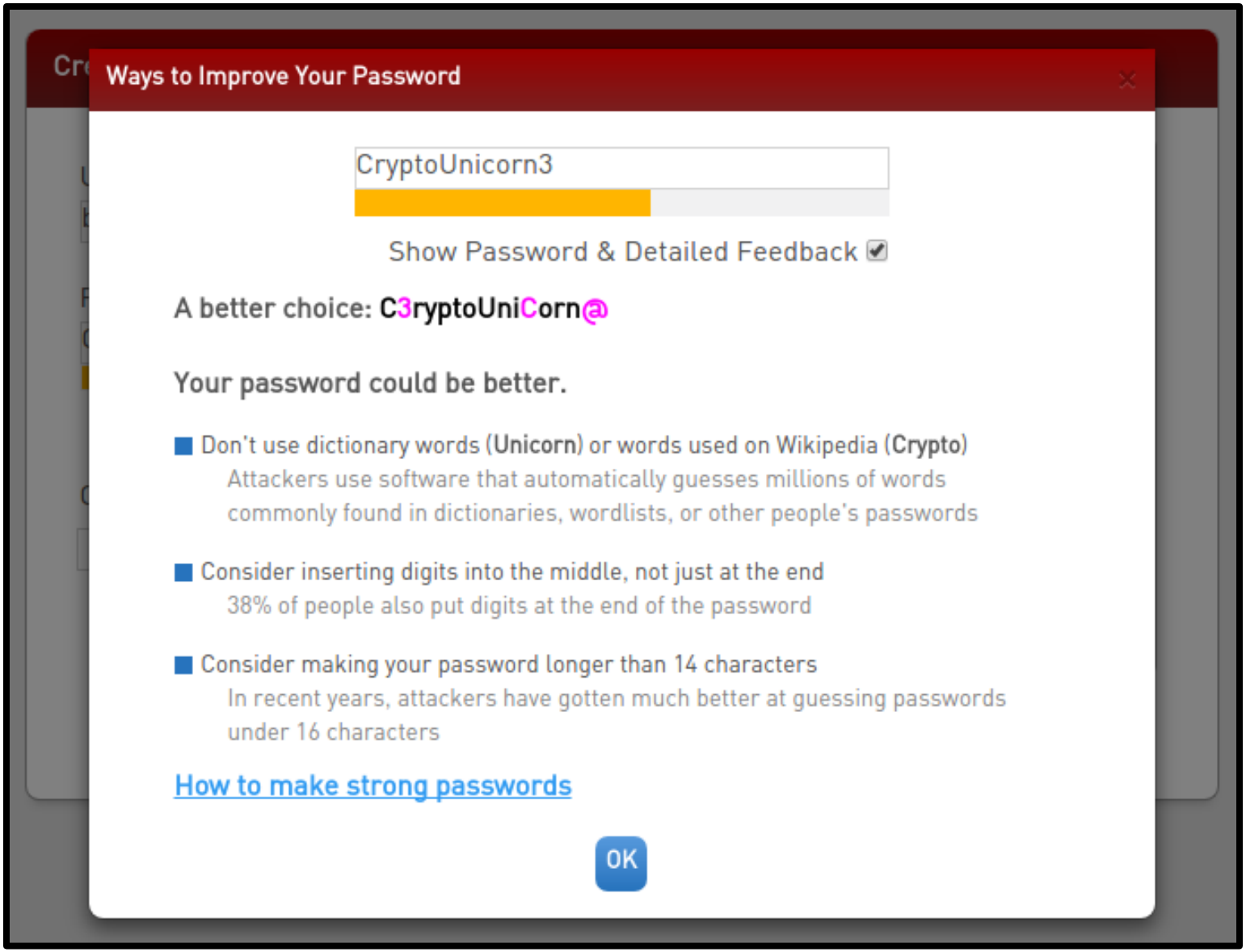
…Gives Detail (Password Shown)



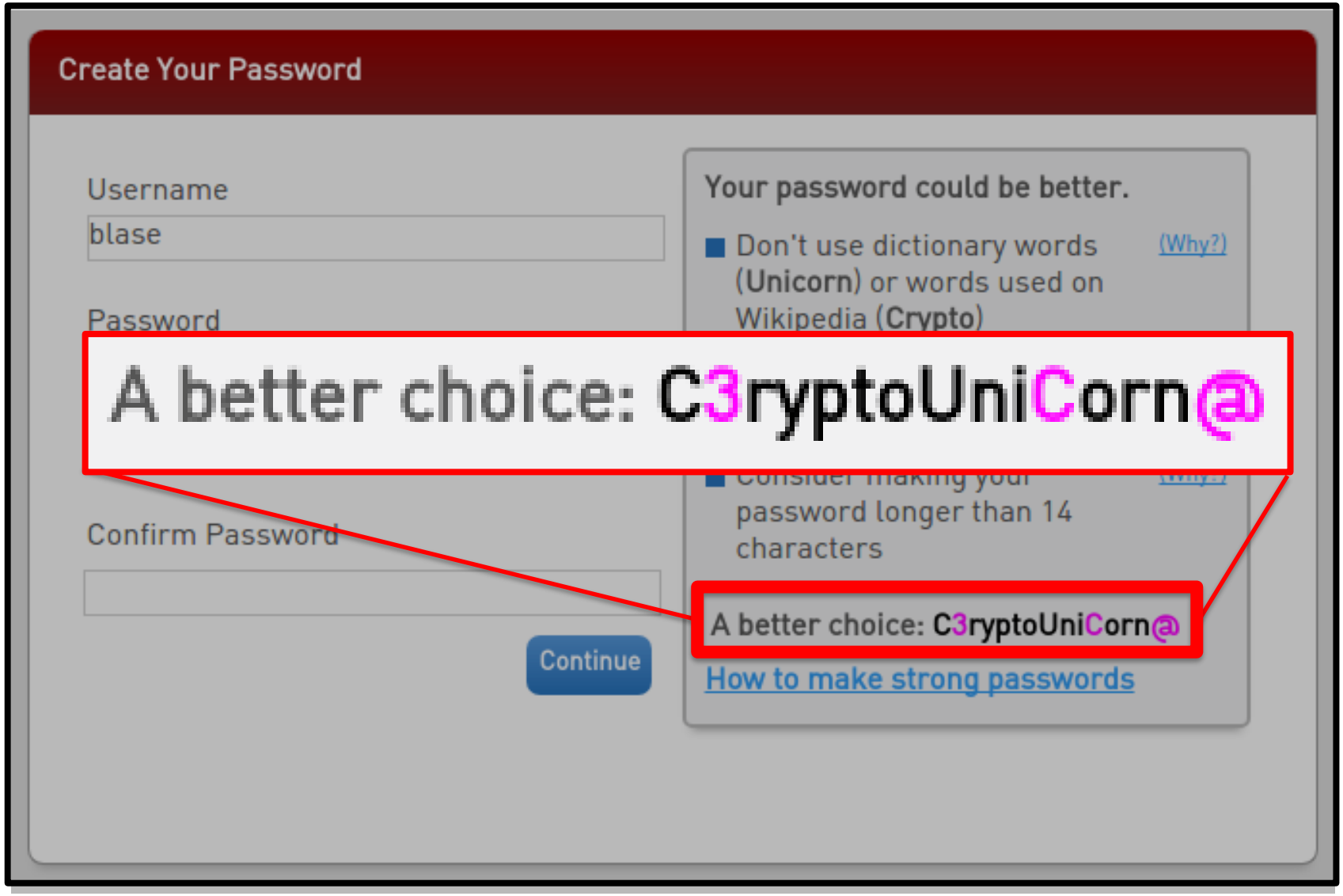
…Offers Explanations



Explanations Shown in Modal



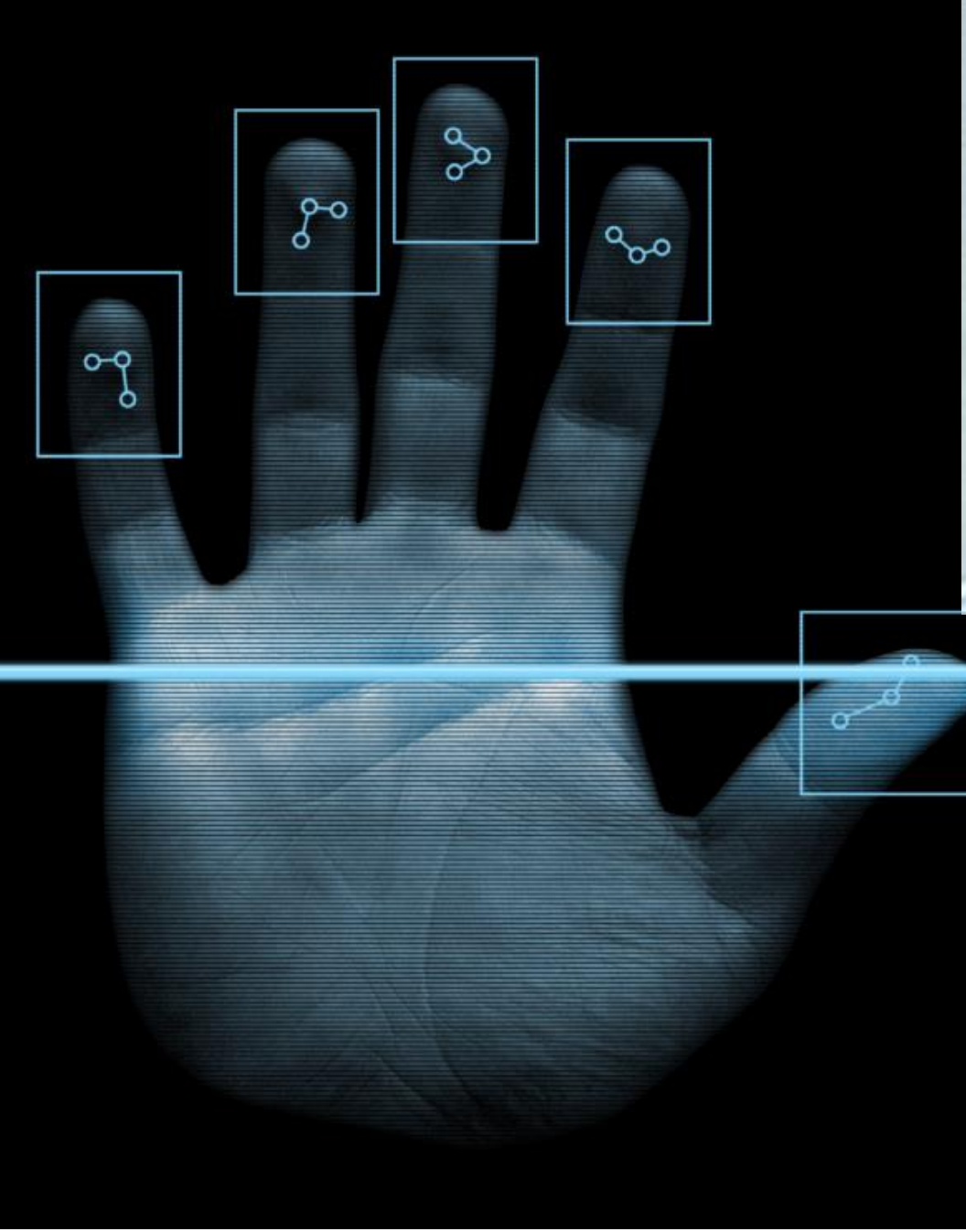
Standard Feedback

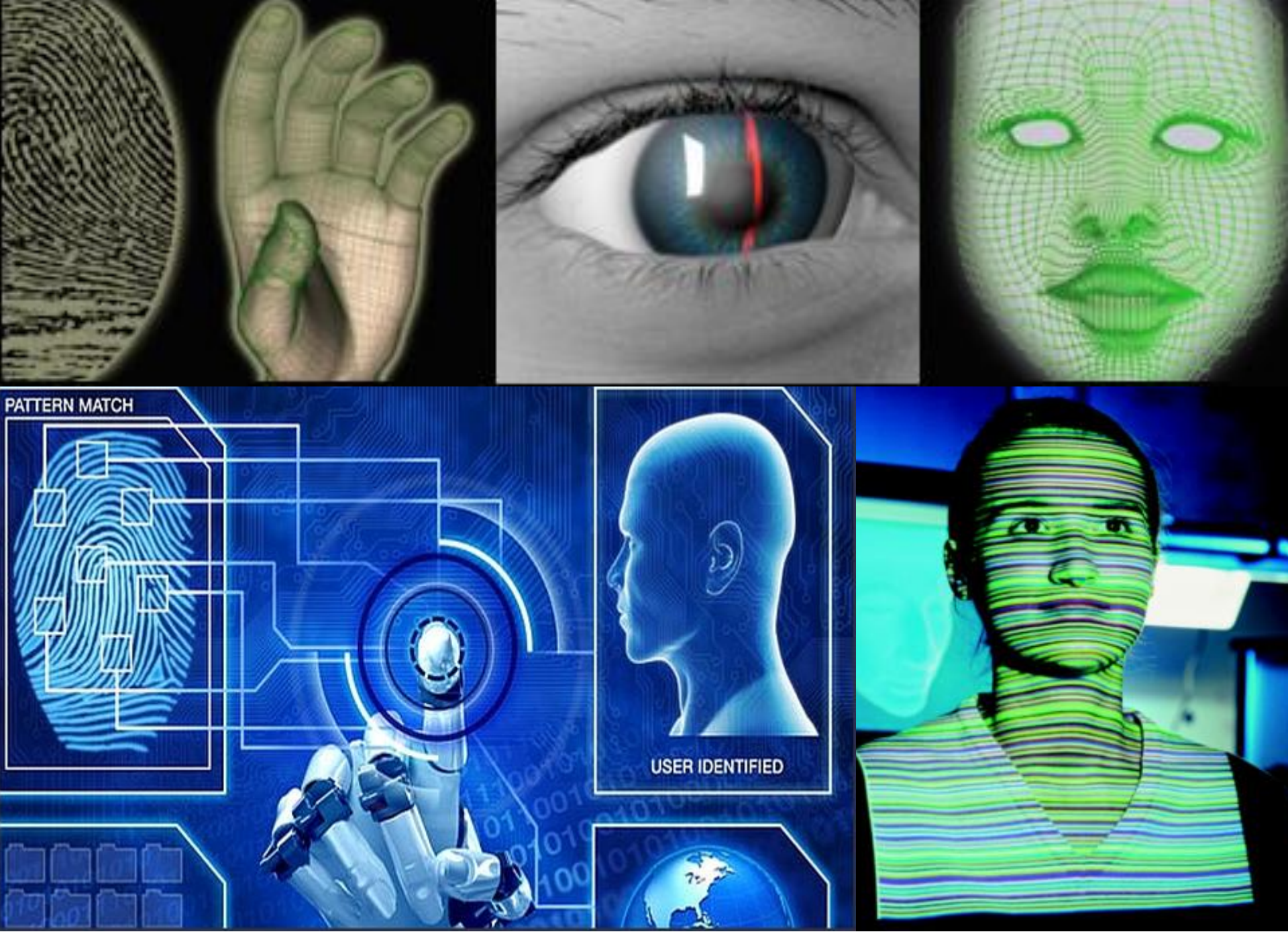


What about Biometrics?

mages on previous slide fair use from androidcentral.com and businessinsider.com. Photo above fair use from abcnews.com

nages 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

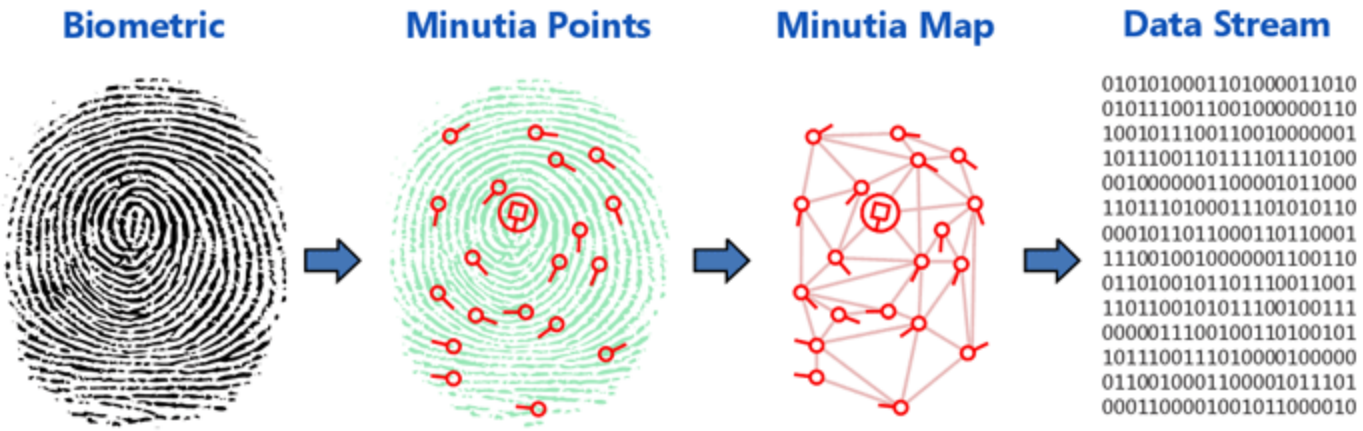
Biometrics

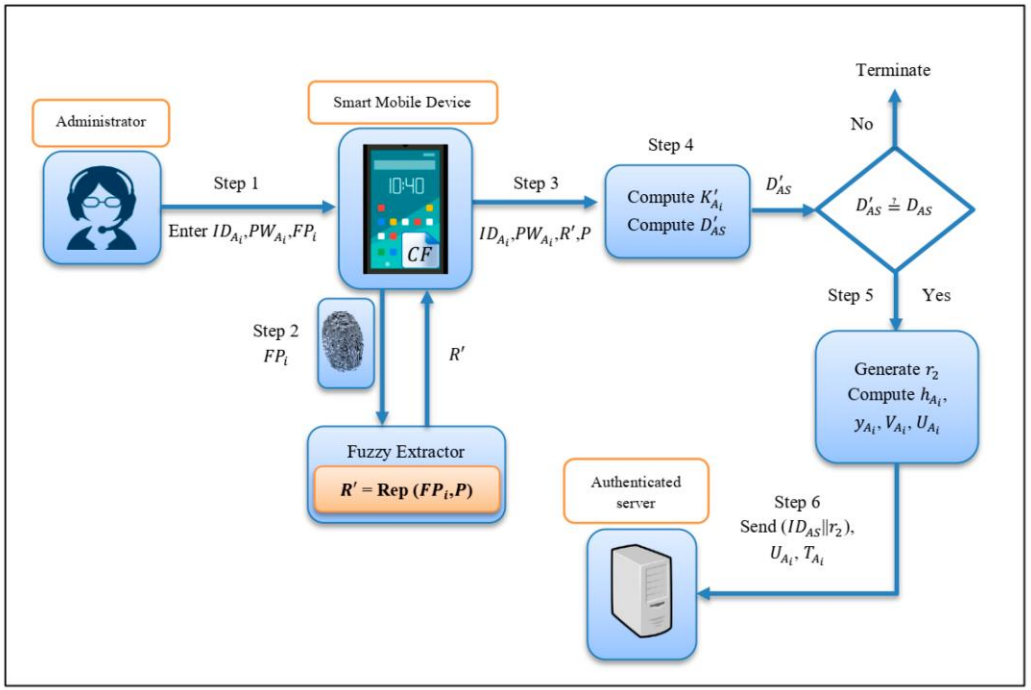
- Fingerprint
- Iris scans or retina scans
- Face recognition
- Finger/hand geometry
- Voice or speech recognition
- The way you type
- (Many others)

Practical Challenges for Biometrics

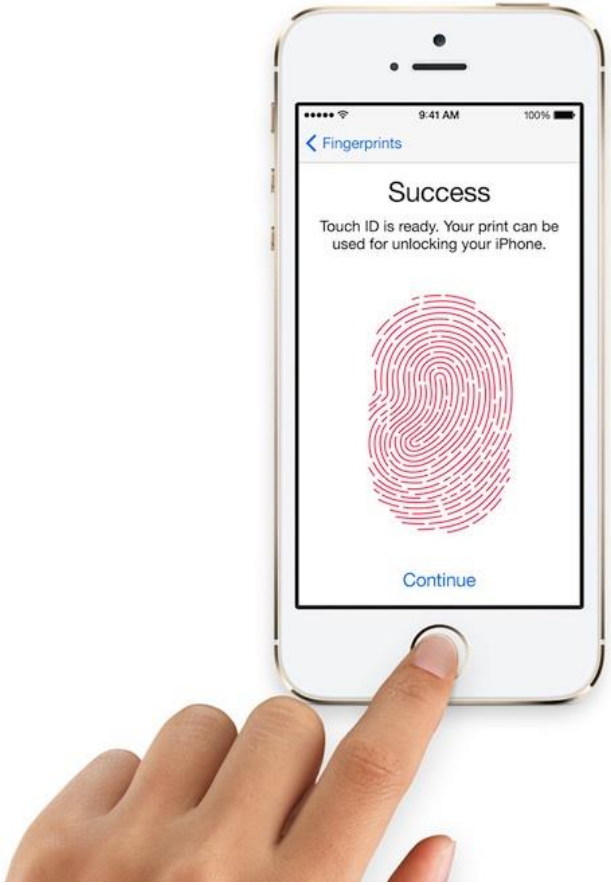
- Immutable (can’t be changed)
- Potentially sensitive data
- High equipment costs
- Sensitive to changes in the environment
- Biometrics can change over time

Storing Biometrics: Templates





iPhone 5S Touch ID



Android 4.0 Face Unlock

•Images fair use from androidcentral.com, creativebits.org, and businessinsider.com.

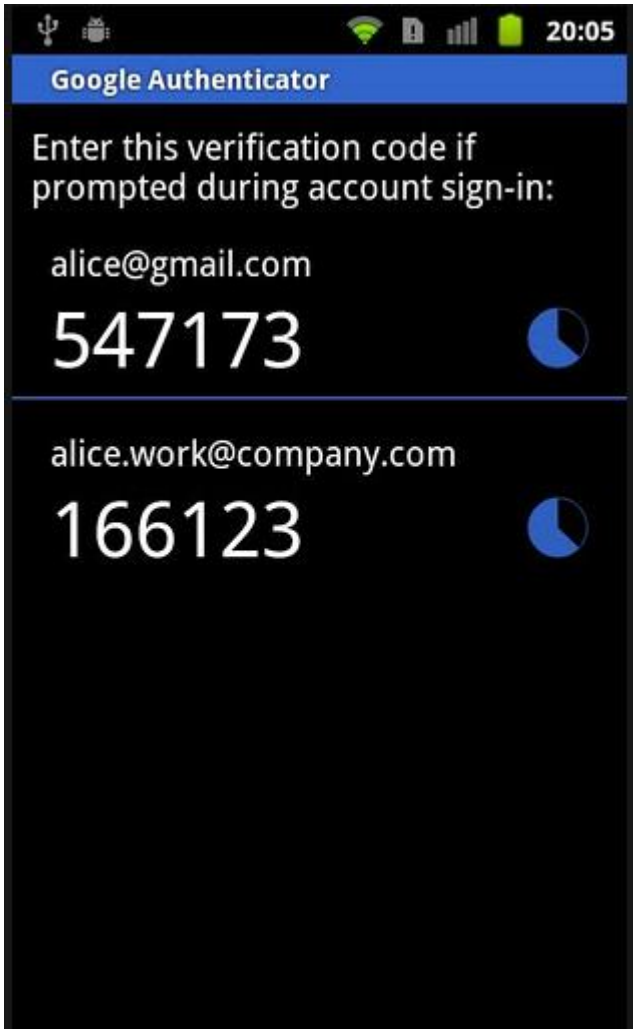
Smartphone Biometrics

- Purpose is to reduce the number of times a user must enter their password
- Falls back to the password
- Face recognition can be tricked by a photo
- Fingerprint recognition can be tricked by a gummy mold
- Users find fingerprint unlock convenient, but do not particularly like face unlock

Practical Authentication

Single Sign-On

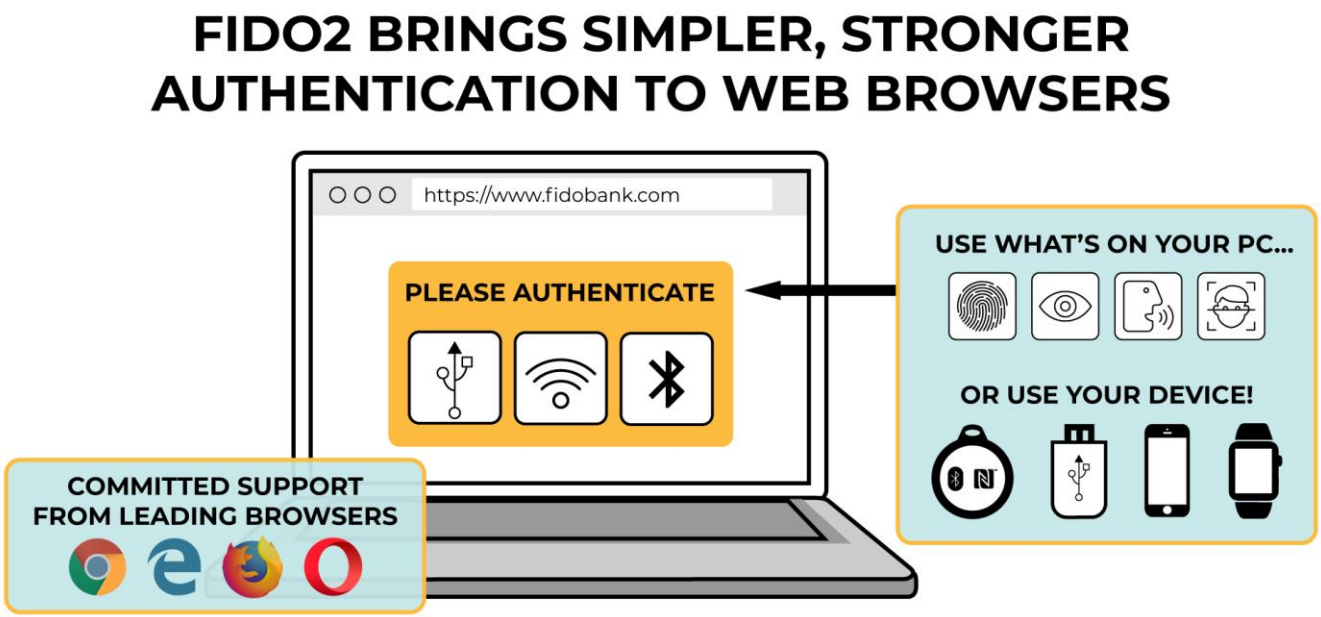
Two-Factor Auth



Physical Tokens / Smart Cards

- Codes based on a cryptographic key
  - Token manufacturer also knows the key
- What if there is a breach?

WebAuthn (2019)



Resetting Accounts

- I forgot my password!
- Send an email?
- Security questions?
- In-person verification?
- Other steps?
- (No backup)

Password Managers

- Trust all passwords to a single master password
  - Also trust software

1Password

Conclusions

- Authentication is really hard!
  - Hard for system administrators
  - Hard for users
- Unfortunately, authentication is necessary