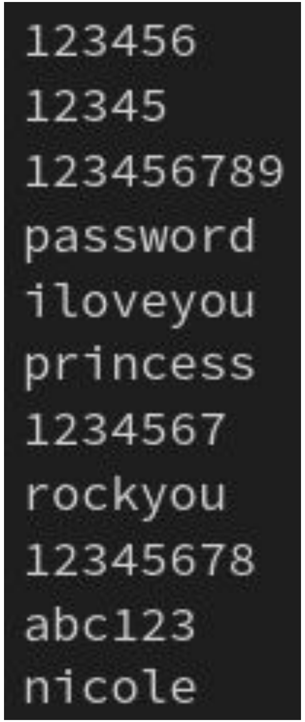
tipsy: how to correct password typos securely

Philippe Partarrieu
philippe.partarrieu@os3.nl

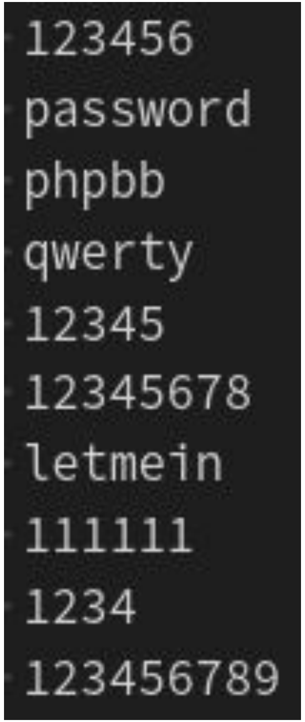
Prof. Zeno Geradts
geradts@uva.nl

Problems with passwords: short & easy



```
123456
12345
123456789
password
iloveyou
princess
1234567
rockyou
12345678
abc123
nicole
```

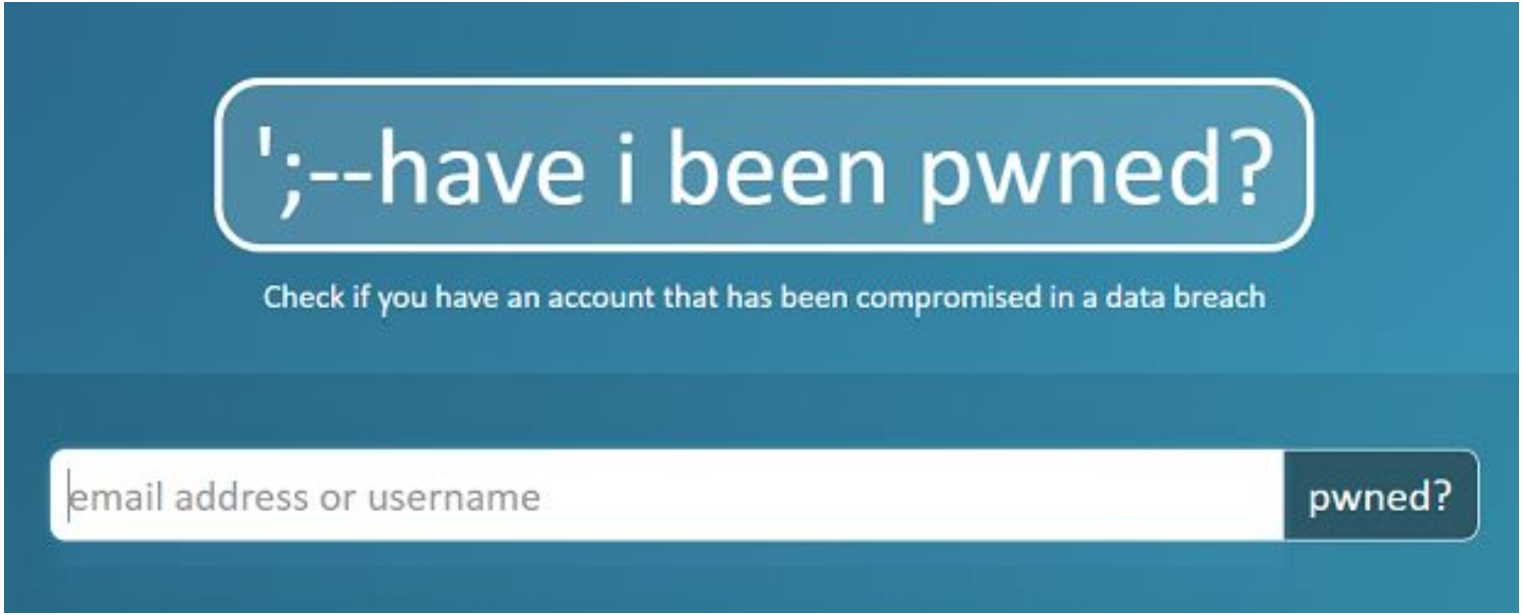
rockyou 10 most frequent passwords



```
123456
password
phpbb
qwerty
12345
12345678
letmein
111111
1234
123456789
```

phpBB 10 most frequent passwords

Problems with passwords: reuse



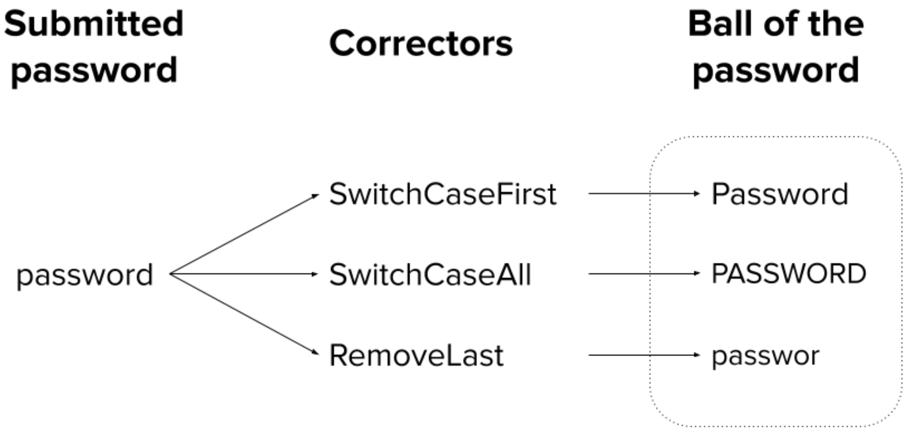
';--have i been pwned?

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

Solution: password managers

Do secure typo-tolerant password authentication schemes exist?

Some lingo: what's a *ball*?

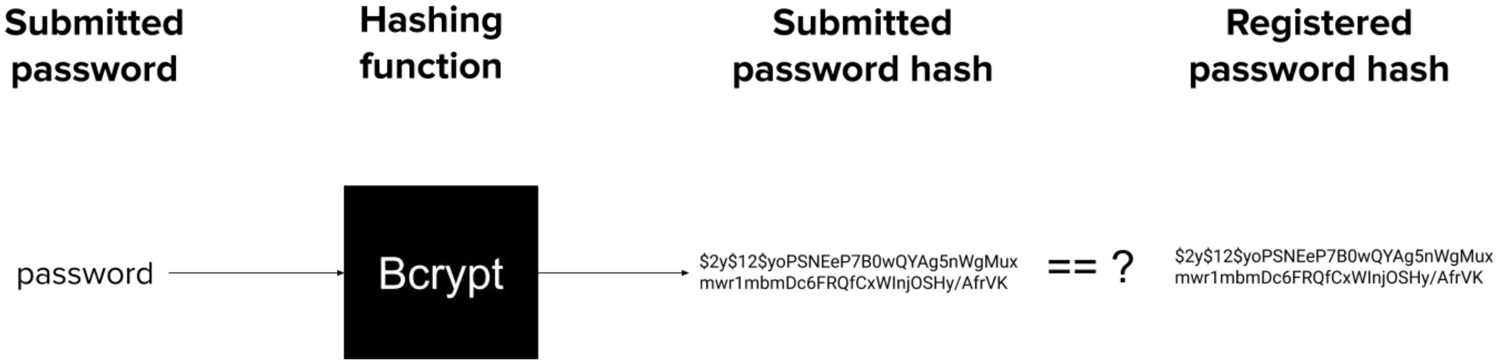


Some lingo: what’s a *checker*?

Noun

1. A password checker compares two optionally salted hashes

Some lingo: what’s a *exact* checker?



Submitted password

Hashing function

Submitted password hash

Registered password hash

password

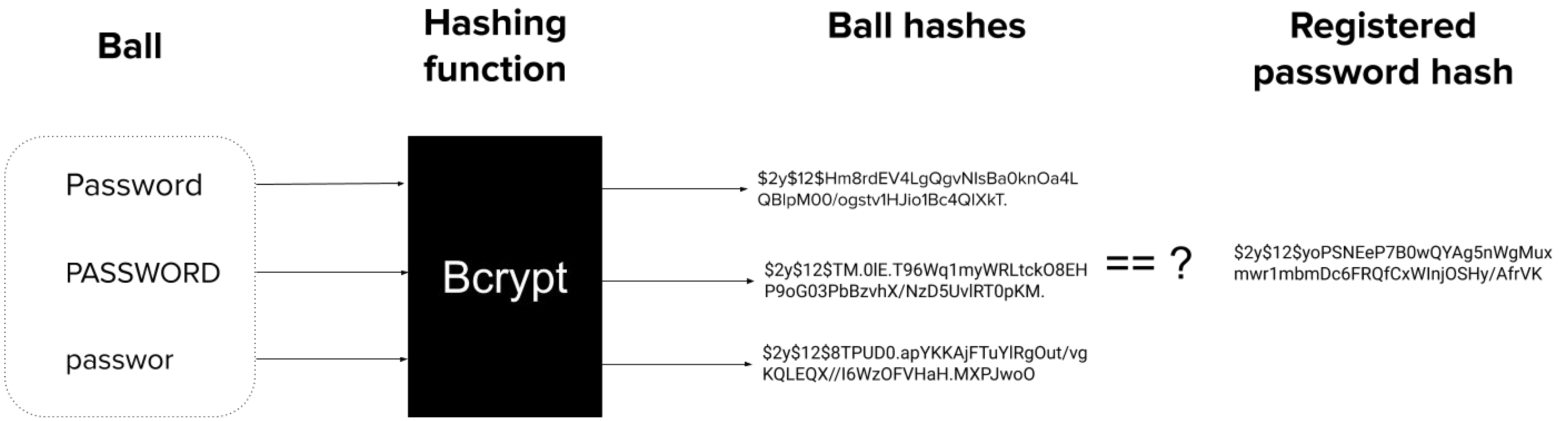
Bcrypt

$2y$12$yoPSNEeP7B0wQYAg5nWgMux
mwr1mbmDc6FRQfCxWInjOSHy/AfrVK

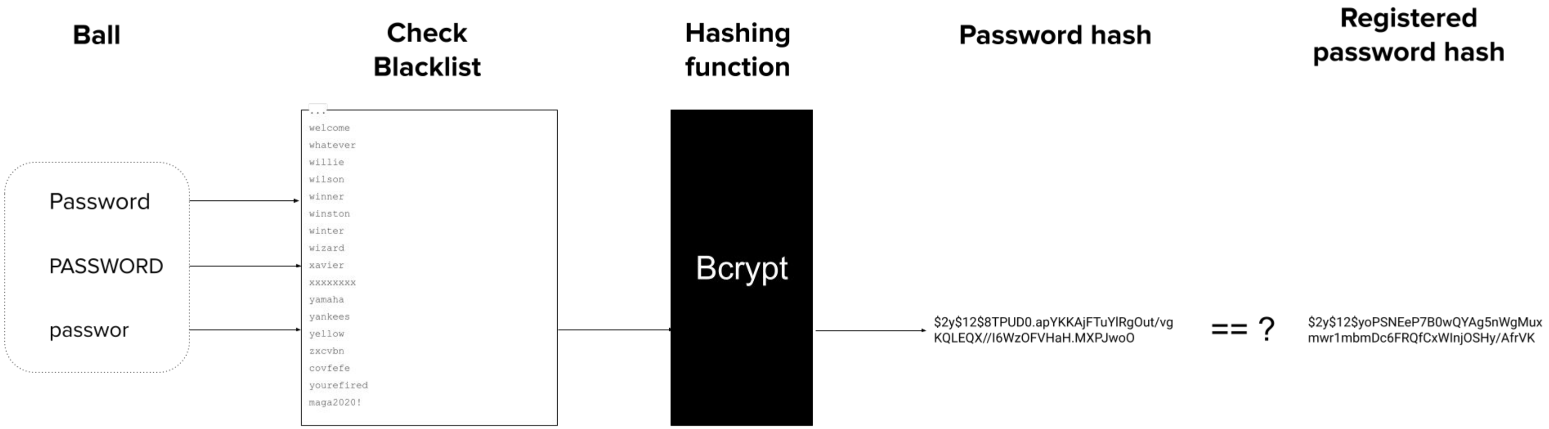
== ?

$2y$12$yoPSNEeP7B0wQYAg5nWgMux
mwr1mbmDc6FRQfCxWInjOSHy/AfrVK

Some lingo: what’s an *always* checker?


Ball
Hashing function
Ball hashes
Registered password hash
Password
PASSWORD
passwor
Bcrypt
$2y$12$Hm8rdEV4LgQgvNIsBa0knOa4LQBlpM00/ogstv1HJio1Bc4QIXkT.
$2y$12$TM.0lE.T96Wq1myWRLtckO8EHP9oG03PbBzvhX/NzD5UvlRT0pKM.
$2y$12$8TPUD0.apYKKAjFTuYlRgOut/vgKQLEQX//I6WzOFVHaH.MXPJwoO
== ?
$2y$12$yoPSNEeP7B0wQYAg5nWgMuxmwr1mbmDc6FRQfCxWInjOSHy/AfrVK

Some lingo: what’s a *blacklist* checker?



Ball

Password

PASSWORD

passwor

Check Blacklist

```
...
welcome
whatever
willie
wilson
winner
winston
winter
wizard
xavier
xxxxxxxx
yamaha
yankees
yellow
zxcvbn
covfefe
yourefired
maga2020!
```

Hashing function

Bcrypt

Password hash

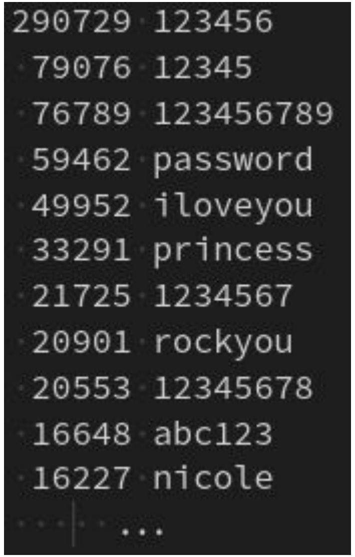
$2y$12$8TPUD0.apYKKAjFTuYlRgOut/vgKQLEQX//I6WzOFVHaH.MXPJwoO

== ?

Registered password hash

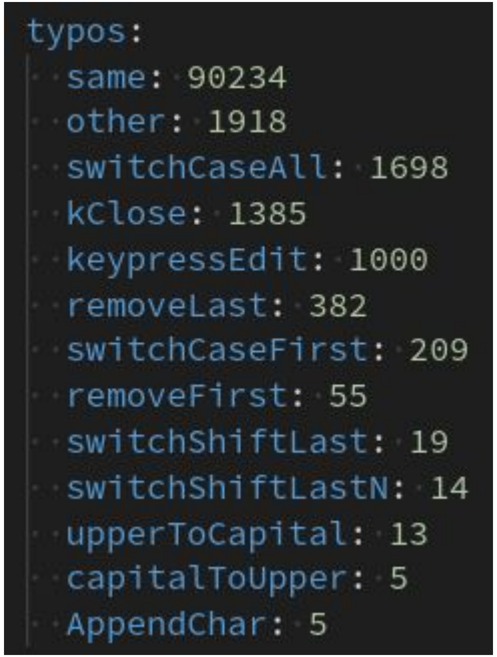
$2y$12$yoPSNEeP7B0wQYAg5nWgMuxmwr1mbmDc6FRQfCxWInjOSHy/AfrVK

Some lingo: what’s an *approximately optimal* checker?



```
290729 123456
 79076 12345
 76789 123456789
 59462 password
 49952 iloveyou
 33291 princess
 21725 1234567
 20901 rockyou
 20553 12345678
 16648 abc123
 16227 nicole
       ...
```

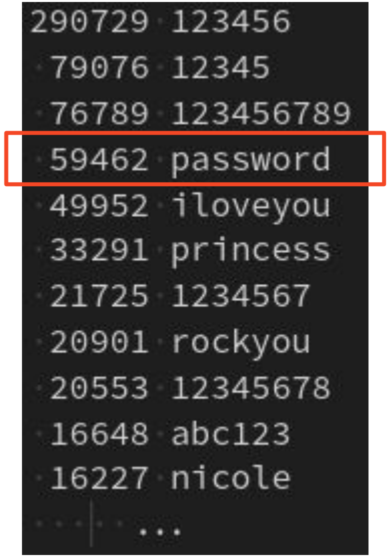
Password distribution estimation using rockyou leak



```
typos:
  same: 90234
  other: 1918
  switchCaseAll: 1698
  kClose: 1385
  keypressEdit: 1000
  removeLast: 382
  switchCaseFirst: 209
  removeFirst: 55
  switchShiftLast: 19
  switchShiftLastN: 14
  upperToCapital: 13
  capitalToUpper: 5
  AppendChar: 5
```

Typo distribution estimation using research from Chatterjee et al.

Aside: what’s the probability of a password?

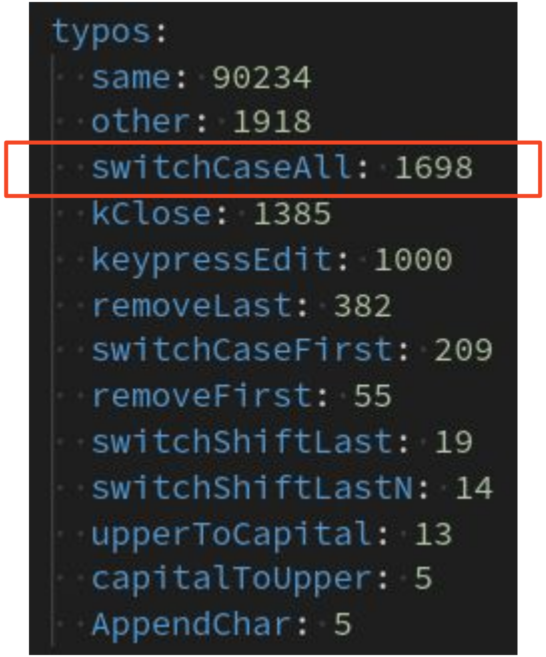


```
290729 123456
 79076 12345
 76789 123456789
 59462 password
 49952 iloveyou
 33291 princess
 21725 1234567
 20901 rockyou
 20553 12345678
 16648 abc123
 16227 nicole
       ...
```

Password probability distribution estimation using rockyou leak

probability = password frequency / total number of passwords
= 59462 / 15879595
= 0.00374455393
≃ 0.3%

Aside: what's the probability of a typo?

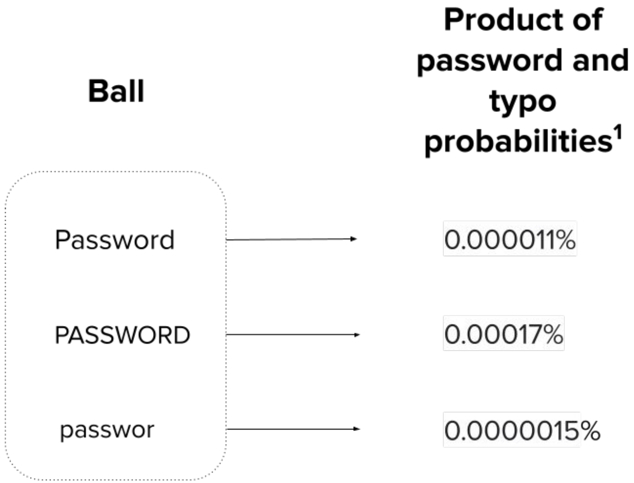


```
typos:
  same: 90234
  other: 1918
  switchCaseAll: 1698
  kClose: 1385
  keypressEdit: 1000
  removeLast: 382
  switchCaseFirst: 209
  removeFirst: 55
  switchShiftLast: 19
  switchShiftLastN: 14
  upperToCapital: 13
  capitalToUpper: 5
  AppendChar: 5
```

Typo distribution estimation using research from Chatterjee et al.

probability = typo frequency / total number of typos
= 1698 / 96963
= 0.01751183441
≃ 1.8%

Some lingo: what’s an *approximately optimal* checker?



Ball

Product of password and typo probabilities[1]

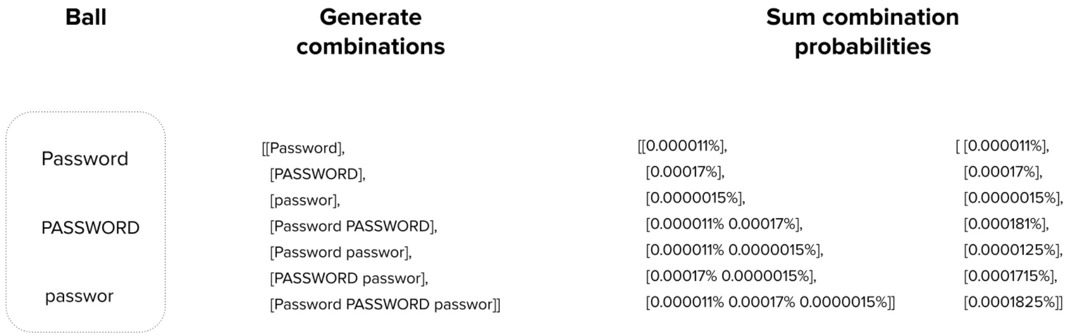
Password → 0.000011%

PASSWORD → 0.00017%

passwor → 0.0000015%

[1] password probability * typo probability * 100

Some lingo: what’s an *approximately optimal* checker?



Ball

Password

PASSWORD

passwor

Generate combinations

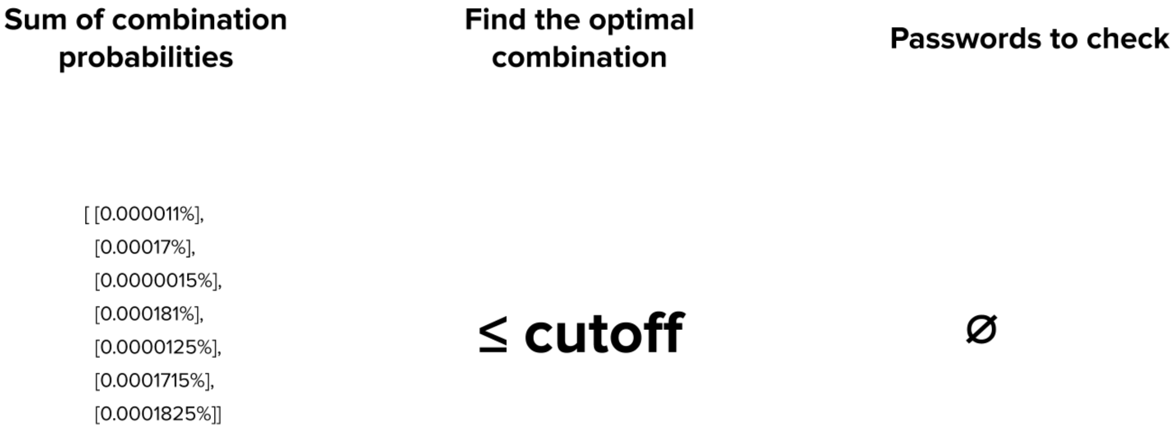
[[Password],
[PASSWORD],
[passwor],
[Password PASSWORD],
[Password passwor],
[PASSWORD passwor],
[Password PASSWORD passwor]]

Sum combination probabilities

[[0.000011%],
[0.00017%],
[0.0000015%],
[0.000011% 0.00017%],
[0.000011% 0.0000015%],
[0.00017% 0.0000015%],
[0.000011% 0.00017% 0.0000015%]]

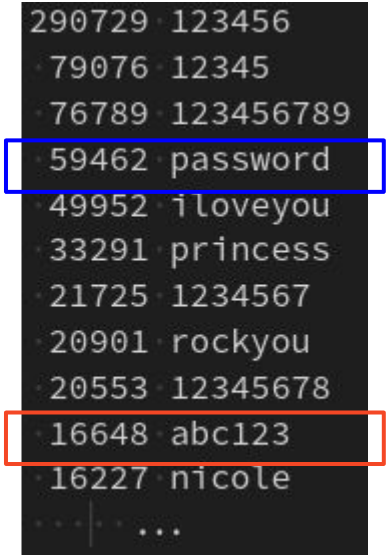
[[0.000011%],
[0.00017%],
[0.0000015%],
[0.000181%],
[0.0000125%],
[0.0001715%],
[0.0001825%]]

Some lingo: what’s an *approximately optimal* checker?



Sum of combination probabilities	Find the optimal combination	Passwords to check
[[0.000011%], [0.00017%], [0.0000015%], [0.000181%], [0.0000125%], [0.0001715%], [0.0001825%]]	≤ cutoff	∅

Aside: how do we find the cutoff?



```
290729 123456
  79076 12345
  76789 123456789
  59462 password
  49952 iloveyou
  33291 princess
  21725 1234567
  20901 rockyou
  20553 12345678
  16648 abc123
  16227 nicole
        ...
```

Password distribution estimation using rockyou leak

cutoff = probability of qth most probable password
  - probability of the submitted password
 = 0.1% - 0.3%
 = - 0.2%

How can we compare the security of the different checkers?

Experiment Design: calculating security loss

Intuitively we think that using typo-tolerance will increase the probability of success of the optimal online attack by a factor of **c**, where c is the number of correctors

This intuition is true iif the set of registered passwords is uniform

Experiment Design: calculating security loss

Intuitively we think that using typo-tolerance will increase the probability of success of the optimal online attack by a factor of **c**, where c is the number of correctors

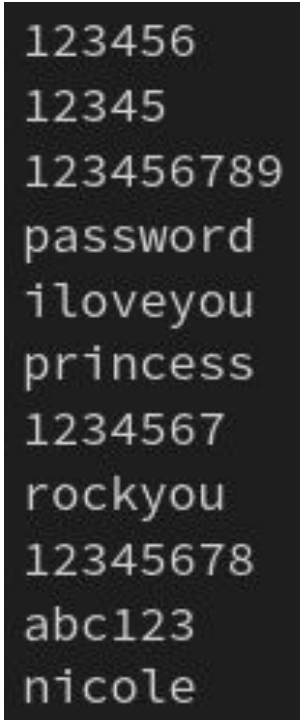
This intuition is true iif the set of registered passwords ~~is uniform~~

Experiment Design: evaluating the security loss

There exists two kinds of attackers:

- Estimating attackers (real attackers) do not have knowledge about the password distribution. They use custom wordlists to tweak password generation algorithms such as PCFGs
- Exact knowledge attackers know the exact distribution of the registered passwords

Experiment Design: exact knowledge attackers



```
123456
12345
123456789
password
iloveyou
princess
1234567
rockyou
12345678
abc123
nicole
```

Naive attack consists in submitting the most-probable passwords from the distribution

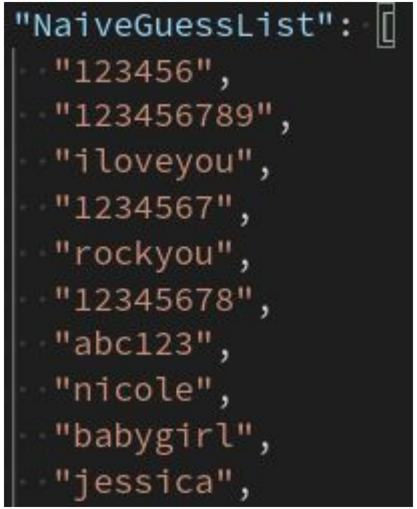
Experiment Design: exact knowledge attackers

Maximum coverage problem

> “As input you are given several sets and a number k. The sets may have some elements in common. You must select at most k of these sets such that the maximum number of elements are covered, i.e. the union of the selected sets has maximal size.”

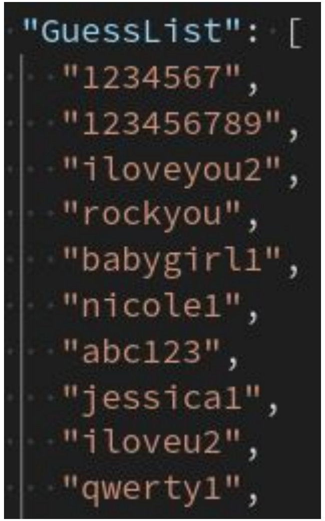
Experiment Design: calculating security loss

For the Always checker with q = 1000 and 3 correctors, using RockYou



```
"NaiveGuessList": [
    "123456",
    "123456789",
    "iloveyou",
    "1234567",
    "rockyou",
    "12345678",
    "abc123",
    "nicole",
    "babygirl",
    "jessica",
```

Extract of the best 1000 guesses against an exact checker

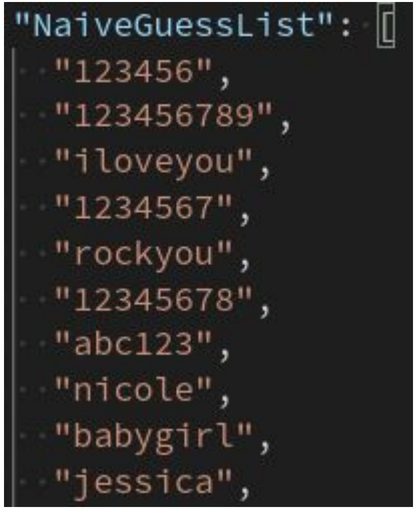


```
"GuessList": [
    "1234567",
    "123456789",
    "iloveyou2",
    "rockyou",
    "babygirl1",
    "nicole1",
    "abc123",
    "jessica1",
    "iloveu2",
    "qwerty1",
```

Extract of the best 1000 guesses against the always checker

Experiment Design: calculating security loss

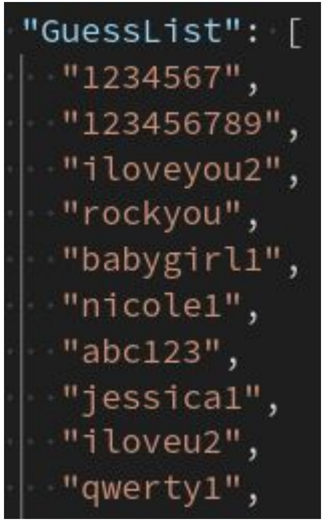
For the Always checker with q = 1000 and 3 correctors, using RockYou



```
"NaiveGuessList": [
  "123456",
  "123456789",
  "iloveyou",
  "1234567",
  "rockyou",
  "12345678",
  "abc123",
  "nicole",
  "babygirl",
  "jessica",
```

$\lambda_q = 0.19$

Extract of the best 1000 guesses against an exact checker



```
"GuessList": [
  "1234567",
  "123456789",
  "iloveyou2",
  "rockyou",
  "babygirl1",
  "nicole1",
  "abc123",
  "jessica1",
  "iloveu2",
  "qwerty1",
```

$\lambda^{greedy}_q = 0.21$

Extract of the best 1000 guesses against the always checker

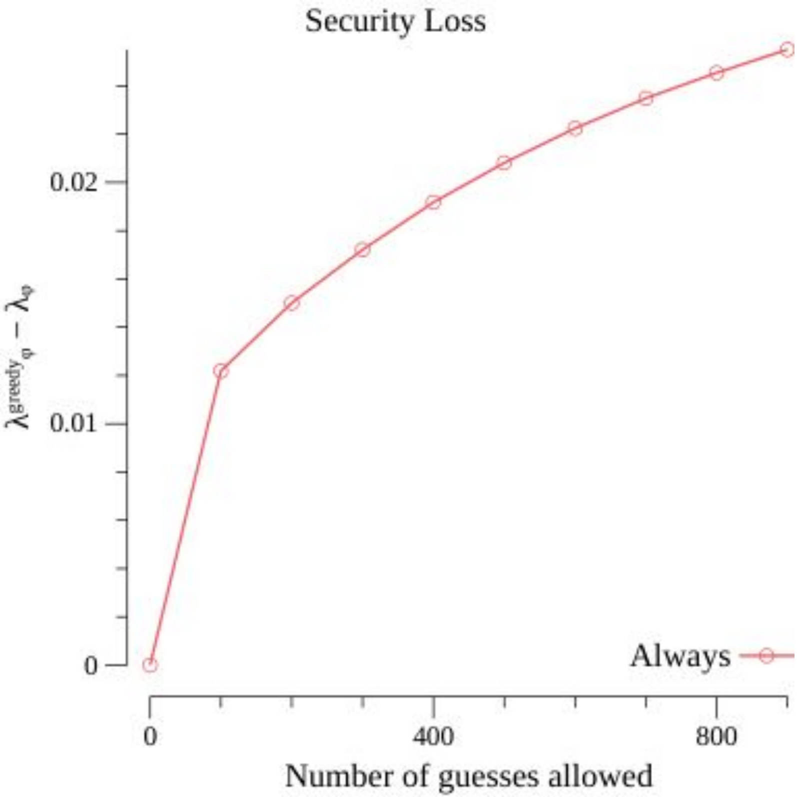
Experiment Design: calculating security loss

For the Always checker with q = 1000 and 3 correctors, using RockYou

$$\begin{aligned} \lambda^{greedy}_q - \lambda_q &= 0.21 - 0.19 \\ &= 0.02 \\ &= 2\% \end{aligned}$$

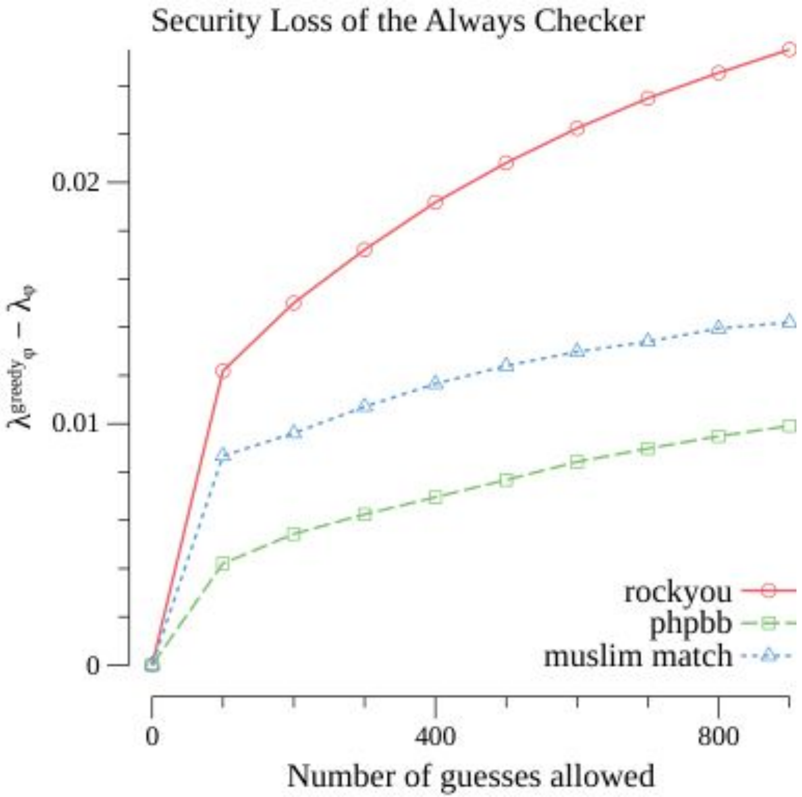
Results: security loss for RockYou

For the Always checker and 3 correctors



Results: security loss across datasets

For the Always checker and 3 correctors



Results: security loss as a %

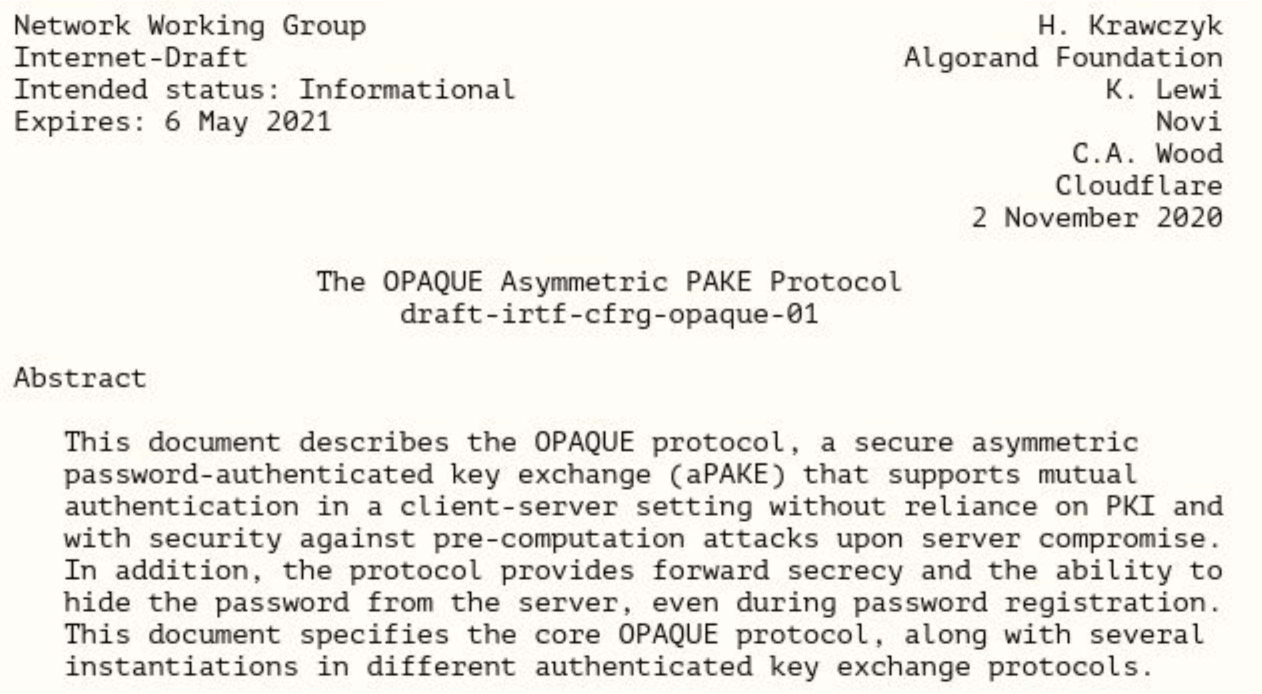
Using 3 correctors, for exact knowledge attackers

Attacker password distribution	q = 10				q = 100				q = 1000			
	AI	BI	AO	Ex	AI	BI	AO	Ex	AI	BI	AO	Ex
rockyou	0.3	0.1		**3.4**	0.8	0.3		**7.5**	2.5	1.2		**19**
phpbb	0.2	0.06		**2.8**	0.3	0.1		**5.5**	0.9	0.7		**12**
muslim match	0.4	0.09		**5.7**	0.6	0.5		**11**	1.4	1.2		**20**

Conclusion

- Typo correction with minimal security loss is possible
- We can take this idea further and do personalised typo correction
- Ideally we should all use password managers

Future work: OPAQUE



```
Network Working Group                                        H. Krawczyk
Internet-Draft                                       Algorand Foundation
Intended status: Informational                                  K. Lewi
Expires: 6 May 2021                                                Novi
                                                               C.A. Wood
                                                              Cloudflare
                                                         2 November 2020
```

The OPAQUE Asymmetric PAKE Protocol
draft-irtf-cfrg-opaque-01

Abstract

This document describes the OPAQUE protocol, a secure asymmetric password-authenticated key exchange (aPAKE) that supports mutual authentication in a client-server setting without reliance on PKI and with security against pre-computation attacks upon server compromise. In addition, the protocol provides forward secrecy and the ability to hide the password from the server, even during password registration. This document specifies the core OPAQUE protocol, along with several instantiations in different authenticated key exchange protocols.

Thanks for listening! 🍻

Source code: https://github.com/ppartarr/tipsy

Security and Network Engineering
https://os3.nl

University of Amsterdam
https://uva.nl