tipsy: how to correct password typos securely

Philippe Partarrieu
philippe.partarrieu@os3.nl

Prof. Zeno Geradts
geradts@uva.nl
Problems with passwords: short & easy

rockyou 10 most frequent passwords

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

phpBB 10 most frequent passwords

- 123456
- password
- phpbb
- qwerty
- 12345
- 12345678
- letmein
- 111111
- 1234
- 123456789
Problems with passwords: reuse
Solution: password managers
Do secure typo-tolerant password authentication schemes exist?
Some lingo: what’s a *ball*?

```
Submitted password  Correctors  Ball of the password

password

- SwitchCaseFirst
- SwitchCaseAll
- RemoveLast

- Password
- PASSWORD
- password
```
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 (Bcrypt) → Submitted password hash

Password hash: S2yS12SyoPSNEnF7B0wQYa5nWgLmUx mwr1mbmDc6FRQfCw1iO3Hy/AfrVK

Registered password hash: S2yS12SyoPSNEnF7B0wQYa5nWgLmUx mwr1mbmDc6FRQfCw1iO3Hy/AfrVK

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

<table>
<thead>
<tr>
<th>Ball</th>
<th>Hashing function</th>
<th>Ball hashes</th>
<th>Registered password hash</th>
</tr>
</thead>
<tbody>
<tr>
<td>Password</td>
<td>Bcrypt</td>
<td>$2yS12$Hy8rdEV4LgQqvN8sBa0knOe4LGBlpMO0/ogstvfHJio18c4O1XXIF.</td>
<td>$s2yS12$SyoPSNEeP7b0wYAg5nWgMux1nrwr1mbrmDc6FR0CxD8IjOSHy/ArFVK</td>
</tr>
<tr>
<td>PASSWORD</td>
<td></td>
<td>$2yS12$STM.01E.T9K6Wg1myWRLtcK08EHp9c0q3ppbZvX/NzD5UvIRT6pKM.</td>
<td></td>
</tr>
<tr>
<td>password</td>
<td></td>
<td>$2yS12$S8TPUD0.apYKKajFTuYIRgOut/vgKQLEQX/16WzOFVHaH.MXPJwoO</td>
<td></td>
</tr>
</tbody>
</table>
Some lingo: what’s a blacklist checker?

```
... welcome willie elise winston walter xenos yassina yellow          coretex youarefixed
...
```

Check Blacklist

Hashing function

$2y$12$8TFjDE.4oYKKKjFTLr6Rq0u/x/gKQLEQX/9kWyXoFVHah.JKPUjw0D$

Registered password hash

Password hash

$2y$12$ycoPSNNeTP78gKcYApG6WjMnxmwr1mbn0DcFkQFctWiyQSHy/AfVY$
Some lingo: what’s an *approximately optimal* checker?

**Password distribution estimation using rockyou leak**

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

**Typo distribution estimation using research from Chatterjee et al.**

```
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```
Aside: what’s the probability of a password?

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
```

Probability = typo frequency / total number of typos
= 1698 / 96963
= 0.01751183441
≈ 1.8%

Typo distribution estimation using research from Chatterjee et al.
Some lingo: what’s an *approximately optimal* checker?

<table>
<thead>
<tr>
<th>Ball</th>
<th>Product of password and typo probabilities¹</th>
</tr>
</thead>
<tbody>
<tr>
<td>Password</td>
<td>0.000011%</td>
</tr>
<tr>
<td>PASSWORD</td>
<td>0.00017%</td>
</tr>
<tr>
<td>passwor</td>
<td>0.0000015%</td>
</tr>
</tbody>
</table>

¹ password probability * typo probability * 100
**Some lingo:** what’s an *approximately optimal* checker?

<table>
<thead>
<tr>
<th>Ball</th>
<th>Generate combinations</th>
<th>Sum combination probabilities</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>[Password], [PASSWORD], [password], [Password PASSWORD], [Password password], [PASSWORD password], [Password PASSWORD password]</td>
<td>[0.000011%], [0.000017%], [0.0000015%], [0.0000011%]</td>
</tr>
</tbody>
</table>
**Some lingo:** what’s an *approximately optimal* checker?

<table>
<thead>
<tr>
<th>Sum of combination probabilities</th>
<th>Find the optimal combination</th>
<th>Passwords to check</th>
</tr>
</thead>
<tbody>
<tr>
<td>[0.000001%], [0.00017%], [0.0000015%], [0.0000181%], [0.0000125%], [0.0001715%], [0.00001825%]]</td>
<td>≤ cutoff</td>
<td>Ø</td>
</tr>
</tbody>
</table>
Aside: how do we find the cutoff?

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 if and only if 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 if 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

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

Extract of the best 1000 guesses against an exact checker

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

$\lambda_q = 0.19$  

$\lambda_{\text{greedy}}^q = 0.21$

Extract of the best 1000 guesses against an exact checker

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

$$\lambda^{\text{greedy}}_q - \lambda_q = 0.21 - 0.19$$
$$= 0.02$$
$$= 2\%$$
Results: security loss for RockYou

For the Always checker and 3 correctors

![Security Loss Chart](image-url)
Results: security loss across datasets

For the Always checker and 3 correctors
Results: security loss as a %

Using 3 correctors, for exact knowledge attackers

<table>
<thead>
<tr>
<th>Attacker password distribution</th>
<th>q = 10</th>
<th>q = 100</th>
<th>q = 1000</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Al</td>
<td>BI</td>
<td>AO</td>
</tr>
<tr>
<td>rockyou</td>
<td>0.3</td>
<td>0.1</td>
<td>3.4</td>
</tr>
<tr>
<td>phpbb</td>
<td>0.2</td>
<td>0.06</td>
<td>2.8</td>
</tr>
<tr>
<td>muslim match</td>
<td>0.4</td>
<td>0.09</td>
<td>5.7</td>
</tr>
</tbody>
</table>
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
Internet-Draft
Intended status: Informational
Expires: 6 May 2021

H. Krawczyk
Algorand Foundation

K. Lewi
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](https://github.com/ppartarr/tipsy)

---

Security and Network Engineering
https://os3.nl

University of Amsterdam
https://uva.nl