Digit Semantics based Optimization for Practical Password Cracking Tools

Haodong Zhang, Chuanwang Wang, Wenqiang Ruan, Junjie Zhang, Ming Xu, Weili Han

Presenter: Haodong Zhang

Laboratory for Data Analytics and Security, Fudan University
Shanghai Key Laboratory of Data Science, Fudan University
Introduction

Textual passwords

One of the most widely used authentication schemes at present
- Low cost
- Friendly usage

Users lean to make password meaningful by employing semantic patterns in order to facilitate memorization and input.

Semantics represented with digits (digit semantics)  Date, Phone, Postcode …

- Largely missed in most studies on password semantics.
- Limited in one/two types of digit semantics or the length of digit string

The lack of a comprehensive analysis of digit semantics in passwords.
- No applications on the practical password cracking tools.

The lack of the combination of digit semantics and practical password cracking tools
Introduction

Our Work

The lack of a comprehensive analysis of digit semantics in passwords.

The lack of the combination of digit semantics and practical password cracking tools

- The digit semantics extraction tool and a large-scale comprehensive analysis of digit semantics in the passwords from the real world.

- Password cracking optimization based on digit semantics: new operations on the level of digit semantics and the digit semantics mangling rules constructed from them.
**Background**

**Rule-based Attacks**

```
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th>hash</th>
</tr>
</thead>
<tbody>
<tr>
<td>alice</td>
<td>Append!</td>
<td>alice!</td>
<td>3da87</td>
</tr>
<tr>
<td>(word)</td>
<td>(rule)</td>
<td>(guess)</td>
<td>......</td>
</tr>
</tbody>
</table>
```

**Wordlist** : leaked passwords (plaintext), words from dictionaries, etc.

**Rule set** : mangling rules, which indicate the operations to be done on the word

**Target file**: leaked passwords which are protected by hash algorithms

“wordlist mode” in JtR (rule-major order)

“rule-based attacks” in Hashcat (word-major order)

* Note that JtR and Hashcat order guesses differently
## Background

### Language of Mangling Rules

- Written in a specific language
- Consists of one or more operations
- Parsed left to right.

<table>
<thead>
<tr>
<th>Operation</th>
<th>Description</th>
<th>Example Rule</th>
<th>Input Word</th>
<th>Output Word</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Lowercase all letters</td>
<td>1</td>
<td>p@ssW0rd</td>
<td>p@ssw0rd</td>
</tr>
<tr>
<td>$X</td>
<td>Append character X to end</td>
<td>$1</td>
<td>p@ssW0rd</td>
<td>p@ssW0rd1</td>
</tr>
<tr>
<td>sXY</td>
<td>Replace all instances of X with Y</td>
<td>ss$</td>
<td>p@ssW0rd</td>
<td>p@$W0rd</td>
</tr>
<tr>
<td>&lt;N</td>
<td>Reject plains if their length is greater than N</td>
<td>&lt;G</td>
<td></td>
<td></td>
</tr>
<tr>
<td>!X</td>
<td>Reject plains which contain char X</td>
<td>!z</td>
<td></td>
<td></td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

52 operations in JtR; 55 operations in Hashcat (32 operation in common)
Content

- Introduction
- Background

- Digit Semantics in Password
  - Extraction Tool
  - Empirical Analysis

- Optimization
  - Design & Enforcement
  - Evaluation

- Conclusion
Digit Semantics

Common Digit Patterns
- Repeat: 1111
- Continuation: 1234
- Leap: 1357
- Repeat+: 121212
- Palindrome: 1235789

Information Represented with Digits
- Phone: 110, 139xxxxxxxx
- MathConstant: 31415
- Date: 1997
- Postcode: 200433
- Idiom: 520

Combination of Single Tags
- Combination: 123520

Step A
- Digit Strings (Segment)
  - Common Digit Patterns

Step B
- Combination of Single Tags

Step C
- Subword

Use $S_i$ as Dictionary

unigram-language-model-based word segmentation method (ULM)
## Digit Semantics

### Empirical Analysis

- Rich digit semantics in both English passwords (XATO & Rockyou) and Chinese passwords (Dodonew & Youku).
- The important role of Date.
- Differences in the distributions of Common Digit Patterns, Postcode, Phone, Idiom, Combination.

<table>
<thead>
<tr>
<th>Tags</th>
<th>Dodonew in segs</th>
<th>Dodonew in passwords</th>
<th>Youku in segs</th>
<th>Youku in passwords</th>
<th>XATO in segs</th>
<th>XATO in passwords</th>
<th>Rockyou in segs</th>
<th>Rockyou in passwords</th>
</tr>
</thead>
<tbody>
<tr>
<td>Repeat</td>
<td>2.32%</td>
<td>1.86%</td>
<td>0.92%</td>
<td>0.80%</td>
<td>3.21%</td>
<td>1.16%</td>
<td>2.50%</td>
<td>0.74%</td>
</tr>
<tr>
<td>Continuation</td>
<td>8.56%</td>
<td>6.82%</td>
<td>2.45%</td>
<td>2.11%</td>
<td>8.36%</td>
<td>3.03%</td>
<td>12.20%</td>
<td>3.60%</td>
</tr>
<tr>
<td>Leap</td>
<td>0.32%</td>
<td>0.25%</td>
<td>0.36%</td>
<td>0.30%</td>
<td>0.46%</td>
<td>0.16%</td>
<td>0.61%</td>
<td>0.18%</td>
</tr>
<tr>
<td>Repeat+</td>
<td>1.87%</td>
<td>1.50%</td>
<td>1.04%</td>
<td>0.92%</td>
<td>3.54%</td>
<td>1.30%</td>
<td>2.65%</td>
<td>0.79%</td>
</tr>
<tr>
<td>Palindrome</td>
<td>1.06%</td>
<td>0.85%</td>
<td>0.82%</td>
<td>0.73%</td>
<td>2.17%</td>
<td>0.79%</td>
<td>2.33%</td>
<td>0.69%</td>
</tr>
<tr>
<td>Numpad</td>
<td>4.03%</td>
<td>3.23%</td>
<td>3.30%</td>
<td>2.91%</td>
<td>3.55%</td>
<td>1.30%</td>
<td>3.42%</td>
<td>1.01%</td>
</tr>
<tr>
<td>Total Above</td>
<td>18.16%</td>
<td>14.51%</td>
<td>8.89%</td>
<td>7.77%</td>
<td>21.29%</td>
<td>7.73%</td>
<td>23.71%</td>
<td>7.01%</td>
</tr>
<tr>
<td>Phone</td>
<td>4.27%</td>
<td>3.43%</td>
<td>10.62%</td>
<td>9.41%</td>
<td>0.81%</td>
<td>0.30%</td>
<td>5.35%</td>
<td>1.59%</td>
</tr>
<tr>
<td>MathConstant</td>
<td>0.12%</td>
<td>0.09%</td>
<td>0.11%</td>
<td>0.09%</td>
<td>0.16%</td>
<td>0.06%</td>
<td>0.15%</td>
<td>0.05%</td>
</tr>
<tr>
<td>Date</td>
<td>21.19%</td>
<td>17.01%</td>
<td>19.52%</td>
<td>17.22%</td>
<td>42.92%</td>
<td>15.79%</td>
<td>32.06%</td>
<td>9.50%</td>
</tr>
<tr>
<td>Postcode</td>
<td>5.41%</td>
<td>4.35%</td>
<td>4.47%</td>
<td>3.96%</td>
<td>7.36%</td>
<td>2.79%</td>
<td>8.70%</td>
<td>2.58%</td>
</tr>
<tr>
<td>Idiom</td>
<td>5.05%</td>
<td>4.03%</td>
<td>3.04%</td>
<td>2.65%</td>
<td>1.10%</td>
<td>0.40%</td>
<td>1.08%</td>
<td>0.32%</td>
</tr>
<tr>
<td>Total Above</td>
<td>51.05%</td>
<td>40.83%</td>
<td>44.03%</td>
<td>38.60%</td>
<td>68.34%</td>
<td>25.04%</td>
<td>64.34%</td>
<td>19.02%</td>
</tr>
<tr>
<td>Combination</td>
<td>16.86%</td>
<td>13.55%</td>
<td>22.94%</td>
<td>20.36%</td>
<td>6.62%</td>
<td>2.44%</td>
<td>10.56%</td>
<td>3.14%</td>
</tr>
<tr>
<td>Total Above</td>
<td>67.91%</td>
<td>54.37%</td>
<td>66.97%</td>
<td>58.94%</td>
<td>74.96%</td>
<td>27.47%</td>
<td>74.90%</td>
<td>22.15%</td>
</tr>
</tbody>
</table>
Digit Semantics

Distribution of Location

POS_ALL, POS_START, and POS_END can describe almost all tagged segments (over 94.08%)
Digit Semantics

Distribution of Length

- The length of most tagged segments (over 99.30%) is distributed below 12.
- Segments with even length are significantly more than those with odd length.
Optimization

Design & Enforcement

Digit Semantics Operations

Tag_Trans \( B \text{ tag pos p1 p2} \)
- Tags that are highly structured and easy to deform
- To transform matched segments according to the specific format.

- Repeat, Continuation, Leap \( 111 \Rightarrow 1111, 11111, \ldots \)
- Repeat+, Palindrome \( 123 \Rightarrow 12321, 123321 \)
- Date \( 1997 \ (YYYY) \Rightarrow 9701, 9702, \ldots \ (YYMM) \)

Tag_Replace \( F \text{ tag pos p1 p2} \)
- All tags
- To replace the matched segment of a certain tag with a dictionary \( 1997 \Rightarrow 111, 8888, \ldots \)

Digit Semantics Rules

B9214 To transform \( B \) a date string \( 9 \) matching YYYY \( 1 \) at the end of a word \( 2 \) into date strings matching YYMM \( 4 \)

B9214 $1
**Optimization**

**Design & Enforcement**

---

**Kernel process of modified Hashcat**

The gray part represents the original process without modification

- Detection Module
- Processing Module
- Running Logic

---

**Diagram Description:**

- **Kernel Entrance**
  - No operation left?
    - Yes: Next step
    - No: The operation is $B/E$?
      - Yes: apply_rules
      - No: Traverse the operation in the current rule

- **Detection Module**
  - apply_rules
  - Intermediate Result
  - Index of word, hash and rule: $Extra_1$ and $Extra_2$

- **Processing Module**
  - Hashing and Matching
  - Inner Functions for Original Operations
  - Inner Functions for Digit Semantics Operations
Evaluation

Rule Sets & Data Sets

Rule Sets:

Digits (1,974 rules)
Tag_Trans 1,740 rules
Tag_Replacer 234 rules

SpiderLabs (5,146 rules)
Best64 (77 rules)
T0XIC (4,085 rules)
Generated2 (65,117 rules)
Random[1] (15,085 rules)
HR_n (n represents the rule count)

Evaluation Sets:

UUU9 (Chinese) 2,209,915 (Training) 551,689 (Testing)
Neopets (English) 2,115,419 (Training) 528,953 (Testing)

* Filter out the passwords that do not contain a segment with more than 2 digits in evaluation sets.

Wordlist:

Dodonew (Chinese) 10,119,695
XATO (English) 5,189,384

* Deduplicated and reordered by frequency.

Evaluation

JtR: Rule Order

Mix_Digits  SpiderLabs + (Random - 1974 rules) + Digits
Mix_Compare SpiderLabs + Random
Mix_Base    SpiderLabs

Reordered iteratively in descending order of *success density* (*Hit Count / Guess Count*)
Evaluation

JtR: Cracking Results

Dodonew-UUU9
(Chinese Passwords)

XATO-Neopets
(English Passwords)
Evaluation

Hashcat

- A significant increase when cracking both Chinese and English passwords under each existent rule set
- A promising result when adding the top 100 digit semantics rules
Evaluation

Hashcat

Dodonew-UUU9

Digits_100 vs HR_10000 (similar amount of extra guesses)

Digits_100 vs HR_100000 (guesses of one more order of magnitude)

<table>
<thead>
<tr>
<th>Wordlist</th>
<th># Word</th>
<th>Target Set</th>
<th>Rule Set</th>
<th>Extra Guesses</th>
<th>Improvement in Each Built-in Rule Set</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Best64</td>
<td>T0XIC</td>
<td>Generated2</td>
</tr>
<tr>
<td>Dodonew</td>
<td>10,119,695</td>
<td>UUU9</td>
<td>Digits_100</td>
<td>$1.17 \times 10^{11}$</td>
<td>146.78%</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Digits</td>
<td>$4.78 \times 10^{11}$</td>
<td>154.09%</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>HR_10000</td>
<td>$1.00 \times 10^{11}$</td>
<td>92.59%</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>HR_100000</td>
<td>$1.01 \times 10^{12}$</td>
<td>136.09%</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>HR_500000</td>
<td>$5.05 \times 10^{12}$</td>
<td>160.74%</td>
</tr>
<tr>
<td>XATO</td>
<td>5,189,384</td>
<td>Neopets</td>
<td>Digits_100</td>
<td>$1.81 \times 10^{10}$</td>
<td>92.24%</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Digits</td>
<td>$1.15 \times 10^{11}$</td>
<td>98.77%</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>HR_10000</td>
<td>$5.19 \times 10^{10}$</td>
<td>61.46%</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>HR_100000</td>
<td>$5.19 \times 10^{11}$</td>
<td>96.66%</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>HR_500000</td>
<td>$2.59 \times 10^{12}$</td>
<td>117.17%</td>
</tr>
</tbody>
</table>
Conclusion

- The digit semantics extraction tool and a large-scale comprehensive analysis of digit semantics in the passwords from the real world.

- Password cracking optimization based on digit semantics: new operations on the level of digit semantics and the digit semantics mangling rules constructed from them.
Q & A