Improving the Systematic Generation of Secure and Memorable Passphrases by MASCARA

Kousshik Raj (17CS30022)
Supervisor:- Dr. Mainack Mondal

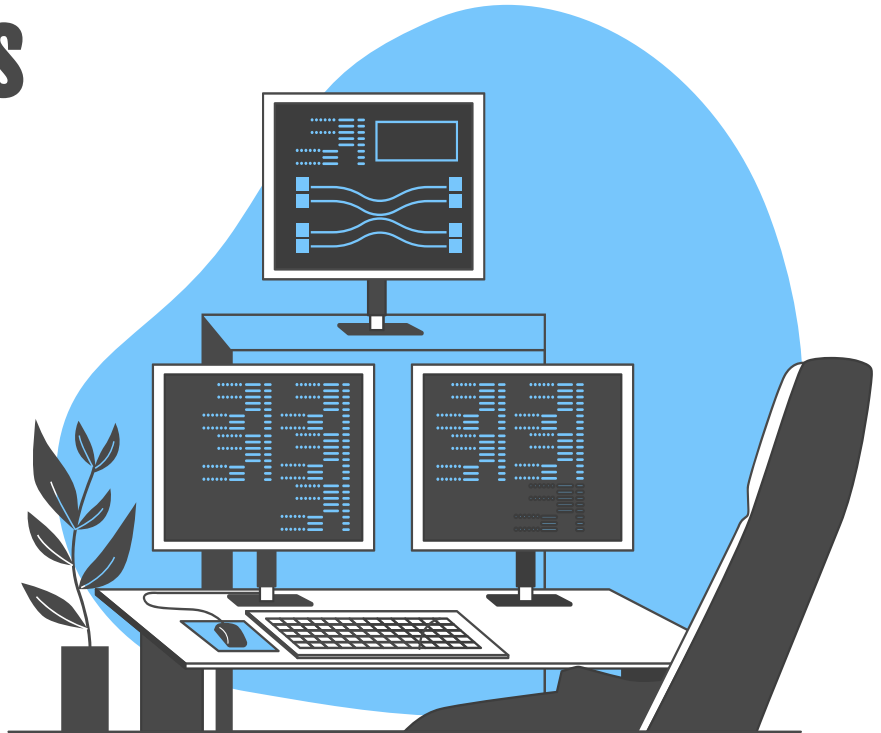
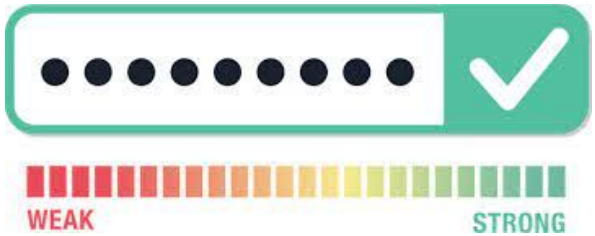


Table of Contents

01

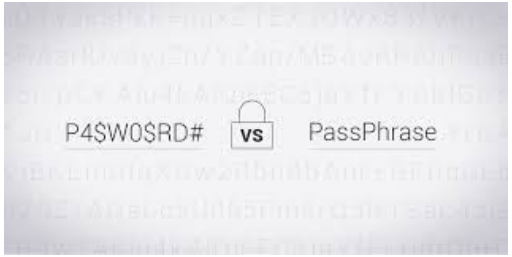
Introduction

Understanding the Situation



Passwords Everywhere

Passwords are the most common authentication mechanism, but they are not memorable and securable often



Passphrase as Alternative

We can use passphrase as alternative, either as passwords or context to generate passwords. Easy to remember, hard to guess

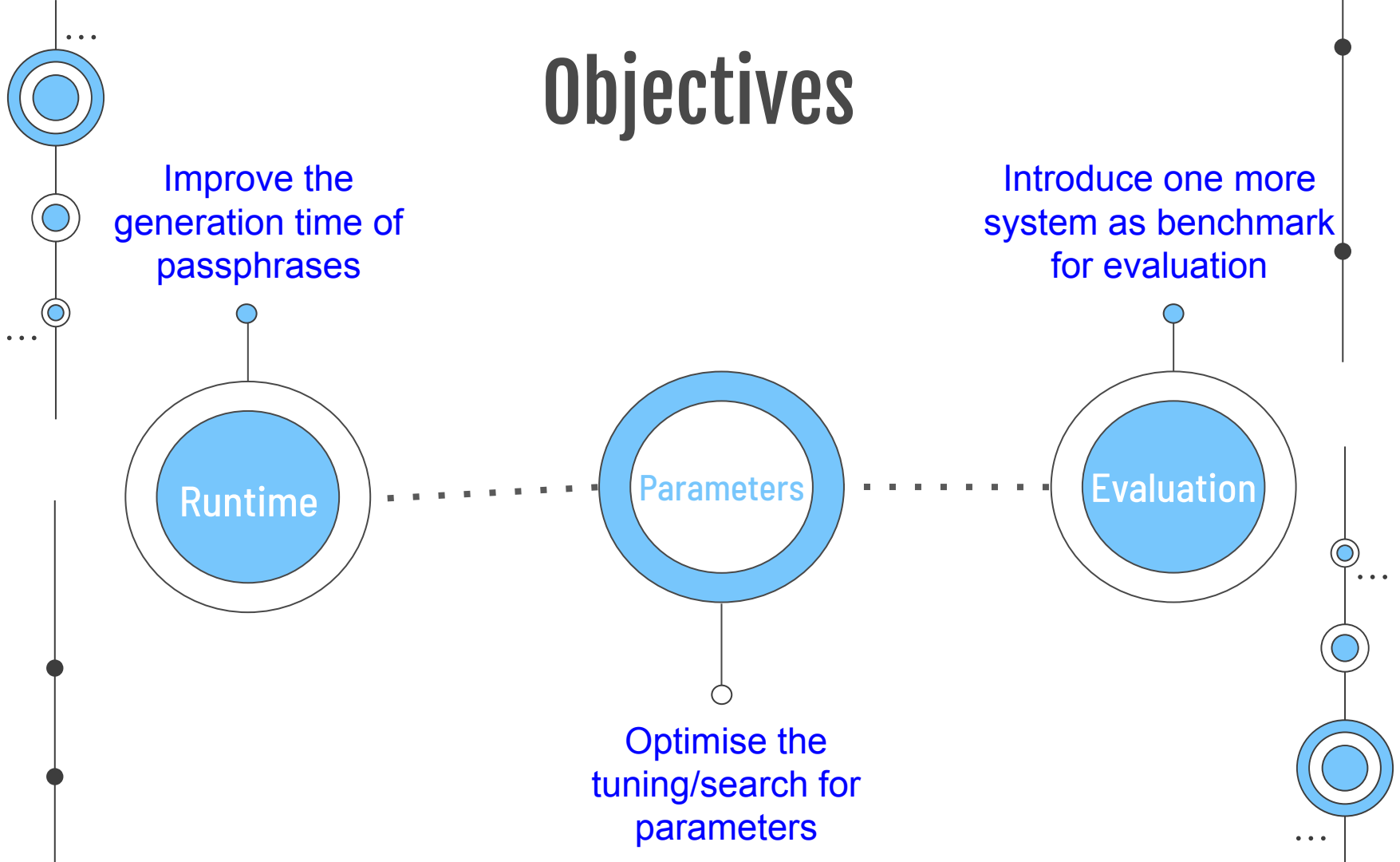
Current Approaches

User: Easy to remember and easy to guess

Machine: Hard to guess and hard to remember

Mascara: Achieves a tradeoff between remembering and guessing

Objectives


Improve the generation time of passphrases
Runtime
Parameters
Optimise the tuning/search for parameters
Introduce one more system as benchmark for evaluation
Evaluation

02

Background

Generating Passphrases

...

- Usually selecting random words from a wordlist - Diceware
- Other variations available - better wordlist, introduce rules for adjacent words
- Studies show that these passphrases are difficult to remember.
- Users choose wordlist - less secure

...

...

Measuring Security and Memorability

...

- Increasing entropy of passwords makes them secure. Same for passphrases.
- Till now security measured through entropy or user study. Mascara provides a new framework.
- Previously, memorability correlated with frequency of passwords, login duration, etc. using user studies.
- Some works have proposed using Character Error Rate (CER) as measure of memorability. Mascara builds upon this.

... ...

Threat Model

...

- For security estimations, we need to know about the scenario surrounding the attempt.
- In this work, the full knowledge of the algorithm, wordlist is assumed. No control/knowledge over the random components.
- Adversarial Guessing has been proposed previously, which is used in Mascara and this work.
- The number of guesses (guessrank) needed to arrive at the passphrase, given the probability of distribution. Just lower bound.

... ...

02

MASCARA

Training Markov Model

...

- The training data is extracted from dump of top 5% recent Wikipedia articles.
- The data was cleaned and normalised.
- Resultant corpus of 3.3~million sentences with 4,55,614 unique words.
- Sentences modified to start with <s> and end with <e>
- From this, unigram $L_1 = (\log(w_c / W))$ and bigram $(L_2 = \log(ww'_c / WW'))$ probabilities are calculated.

...

...

Guessability and Memorability

...

- CER measure used to quantify memorability. It denotes the average rate of error per character. Using previous works and adding new attributes, equation is

$$CER(s) = c_1.L_1(s) + c_2.L_2(s) + c_3.std(s)$$

- L_1 and L_2 are log sum of prob. of unigrams and bigrams. Std is the standard deviation of word lengths.

- Guessrank can be calculated from probability of generation of passphrase $s = w_1 w_2 \dots w_n$ (w_1 = <s> and w_n = <e>)

- Probability of passphrase is product of probabilities of individual bigrams

... ...

Constrained Generation

...

- At each stage of the intermediate generation, the overall CER and guessrank is estimated for overall passphrase.

- **CER** - $S(w_1...w_i) = c_1.L_1(w_i) + c_2.L_2(w_{i-1}w_i) + c_3.std(w_1...w_i)$ || **Guessrank** - $L_2(w_{i-1}w_i)$

- Limit $S(w_1...w_i) <= \theta_1$ and $L_2(w_{i-1}w_i) <= \theta_2$ at every step.

- The threshold provides control over the final Guessrank and CER.

...

...

Final Algorithm

...

- Mascara generates the passphrase incrementally, starting with <s>.
- At every step, the constraints has to be satisfied till the desired length of passphrase is generated.
- Algorithm is greedy and not optimal, but helps to make intermediate decisions.
- Choice of θ_1 and θ_2 is important. In Mascara, after manual testing, set to 80% of min L_2 and 0.5, respectively.

...

...

Final Algorithm

...

GetFirstWord($\mathcal{M}$) :

$W \leftarrow \mathcal{M}.\mathsf{next}(\texttt{<s>}) \setminus \mathsf{B}$

$w \leftarrow_{L_1} W$

return w

MascaraGen($l, \mathcal{M}$):

$w_1 \leftarrow \mathsf{GetFirstWord}(\mathcal{M})$

$i \leftarrow 2$

while $i \leq l$ do

  $W' \leftarrow \mathcal{M}.\mathsf{next}(w_{i-1})$

  if $i \in \{1, l\}$ then $W' \leftarrow W' \setminus B$ /* Remove stopwords */

  if $i \geq 2$ then /* CER and Guess rank constraint */

    $W' \leftarrow \{w \in W' \mid S(w_1 \dots w) \leq \theta_1 \text{ and } L_2(w_{i-1}, w) \leq \theta_2\}$

  if $i = l$ then /* Ends in a end symbol */

    $W' \leftarrow \{w \in W' \mid \texttt{<e>} \in \mathcal{M}.\mathsf{next}(\mathsf{w})\}$

  $w_i \leftarrow_{\mathsf{s}} W'$

  if $w_i = \perp$ then

    $W \leftarrow \mathcal{M}.\mathsf{next}(\texttt{<s>}) \setminus \mathsf{B}$ /* No passphrase found; restart */

    $w \leftarrow_{L_1} W$

  else $i \leftarrow i + 1$

return $w_1 \dots w_l$

03

PROPOSED WORK

Improving Runtime: Before

...

- Memorability is associated with meaningfulness. So meaningful passphrases might be more memorable.
- So, Mascara uses *PhraseMachine* to eliminate passphrases that are not linguistically sound.
- Problem is that the fraction of such sentences is very less. Many passphrases are almost meaningful.
- So generation time scales up dramatically.

...

...

Improving Runtime: After

...

- Assumption: Memorable sentence need not be meaningful.
- Also, using *PhraseMachine* does not improve CER much (our memorability measure). Decreases Guessrank.
- So, remove *PhraseMachine* and add some minor heuristics.
- If still need more meaningful passphrases, go for *intermediate* heuristics.

...

...

Optimising Parameter Tuning: Before

...

- θ values are very important for the system. Everything hinges on it. How does Mascara estimate it?

- θ_2's range is first estimated as between some percentage of the minimum L_2. The relation between θ_2 and θ_1 and the upper bound on CER gives us range for θ_1.

- Every combination is then evaluated by qualitatively evaluating passphrases, as well as CER and Guessrank measures.

- Not scalable and might not be accurate due to bias of human intervention.

...

...

Optimising Parameter Tuning: After

...

- Main problem is manual intervention. Need some autonomous evaluation.
- Introduce new metric *U*: average of the ratio of log Guessrank and CER, with bottom 1% Guessrank and top 1% CER are assigned 0 and ∞
- Set up the range for θ and decide step and perform grid search. Rank the various systems based on *U*.
- Take top k systems and do a manual evaluation.

...

...

Introducing New Benchmark: MMAP

...

- Evaluation is important. Highlights advantages and disadvantages.
- Mascara considered traditional approaches. But none of them target memorability.
- MMAP, a popular tool from the internet, is used for the purpose. Incorporated into the evaluation system.
- Define the algorithm, memorability and guessability.

...

...

MMAP: Algorithm

- It tries to generate meaningful sentences, and thereby memorable.
- Has a hardcoded dictionary of 20000 handpicked diverse words, segregated into nouns, verbs, adverbs, adjectives, etc.
- 27 Linguistic templates that follow sentence structure are also encoded. Templates are certain combinations of the classes.
- A template is chosen randomly, and classes in it are replaced by words.
- Final generated sentence is modified using articles, conjunctions, etc. based on certain rules.

MMAP: Guessrank and Memorability

...

- Only the Guessrank matters, not the approach. Previous approach is usable here, but attacker has better choice.
- Every template has finite set of passphrases. We estimate Guessrank by enumerating the templates and exhausting the passphrases.
- Precalculate the number of passphrases in each template and remember the template source for generated passphrase.
- No real motivation to change the CER calculation. CERs from different approaches are not comparable.

...

...

04

EVALUATING THE SYSTEM

Phrase Mahine: Runtime Impact

...

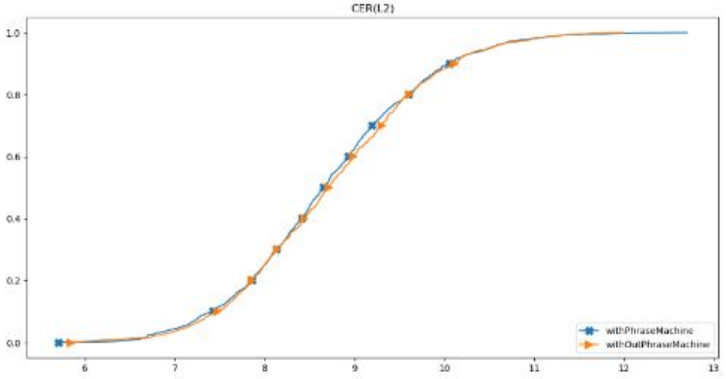
Length	Runtime$_a$ (in s)	Rejection$_a$	Runtime$_b$ (in s)	Rejection$_b$
4	0.08	4.34	8.7	68.4
5	0.09	5.85	18.2	91.8
6	0.10	7.24	28.1	112.4
7	0.13	11.62	43.1	201.7

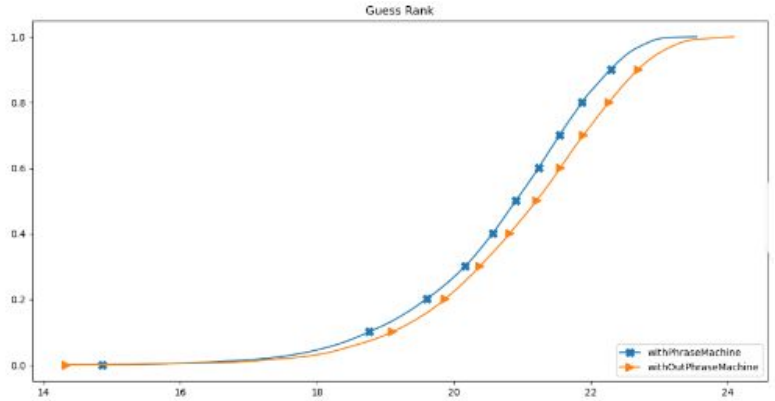
- Table lists average runtime and rejection count over 100 passphrases before and after removing *PhraseMachine* across multiple lengths.

- As length increases, the improvement after the change increases significantly.

... ...

Phrase Mahine: CER & Guessrank Impact





- If we had lost CER and Guessrank for runtime, would not be worth it.
- But no real change in CER and slight improvement in Guessrank.

Evaluation of Passphrases: Benchmarks

- **User:** Several passphrases used by users retrieved from parsing previous password leaks.
- **Diceware:** Generating passphrases by concatenating random words from the Wiki Dataset.
- **Markov:** Passphrases generated using bigram Markov model trained over the Wiki Dataset.
- **MMAP:** Passphrases generated using the MMAP algorithm discussed.
- **Mascara:** Passphrases generated by the improved Mascara system.

Evaluation of Passphrases: Sample Passphrases

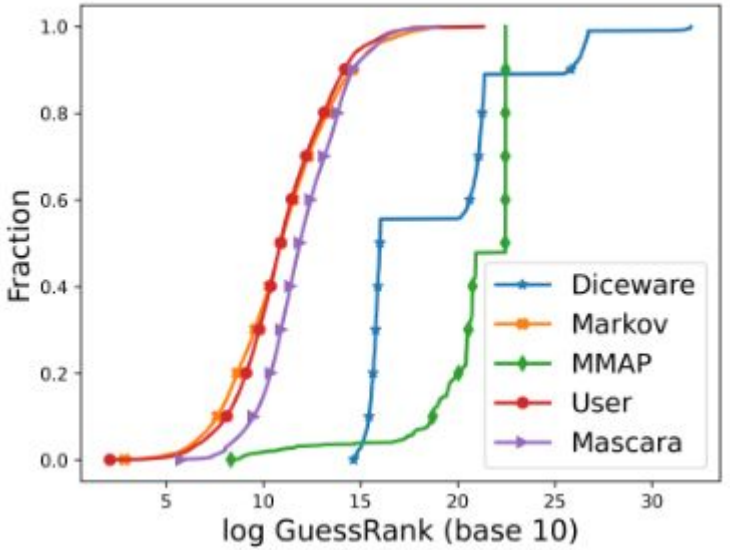
Type	Samples
Diceware	valedictory silliest illeism niglichen dreamscape manchuria dervish verbally whinging ferries portmore permanency downcast epilogue guarding richemont feeney stoppers scalable deuteronomic kudo
UserPP	little bitty pretty one dont forget the password just another happy ending ride with the wind hot and sexy chicken wing
Markov	murrays situation spores have protractable and the problem standard kit during ultraviolet signals beamed there improvements achieved worldwide with coitus are shipwrecked man
MMAP	a flea will republish your gladiator how does its 1 shire milk a jack the frothy thing faxed the aphorism the rivalry was misspelling prior to our eyeballs can Carole wrestle the gaiter
Mascara	woolwich the fleet including queen many local fiscal rights lee sung before god jackal with prayer requests drakes book nearly too quickly

...

...

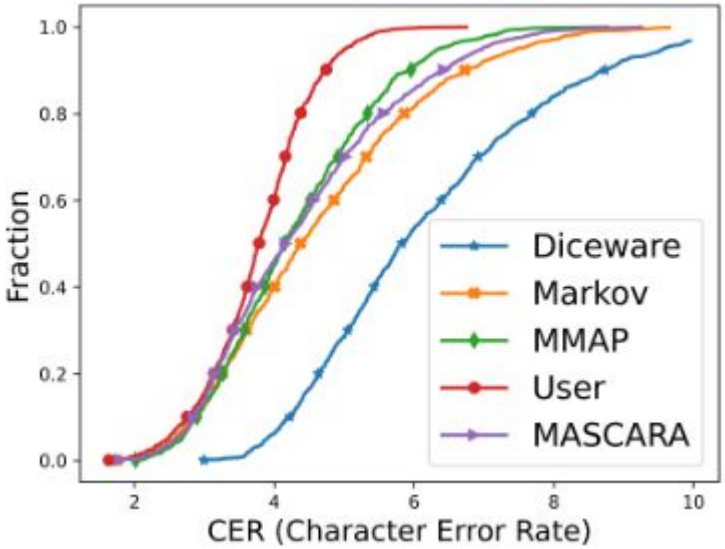
Evaluation of Passphrases: Guessrank

- User and Markov have the worst Guessrank
- Diceware and MMAP have very high Guessranks.
- Mascara provides modest improvement
- MMAP loses its advantage if lengths are increased.



Evaluation of Passphrases: CER

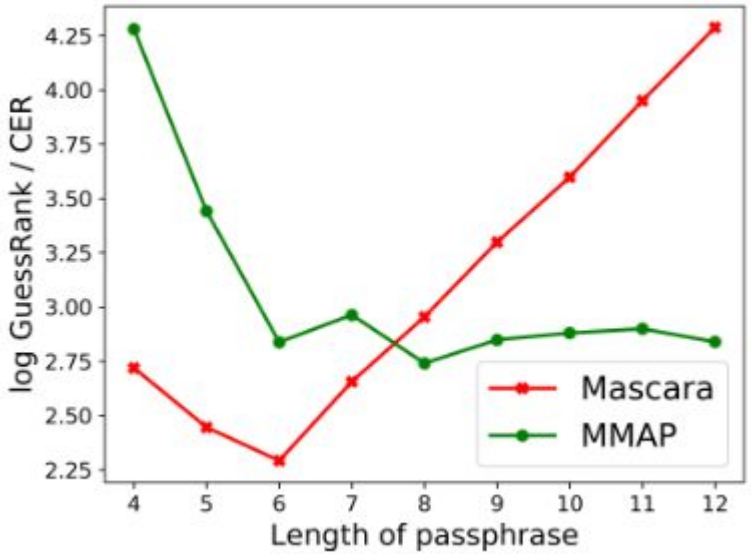
- User has best CER.
- Diceware has worst.
- Others intermediate and nearly same.
- MMAP has a slight lead. The lead difference might increase if passphrase lengths increase.



Evaluation of Passphrases: MMAP vs Mascara

...

- Is MMAP better? For small lengths, yes.
- For larger lengths, guess rank saturates, whereas Mascara's keep increasing.
- For larger lengths, CERs almost converge.
- Mascara better for larger passphrases (> 6).



...

...

05

CONCLUSION

Conclusion

...

- Passphrases are important alternative for passwords, but no proper approach to generate memorable and secure passphrases.
- Mascara uses a memorability framework and constrained Markov generation process to maintain balance between memorability and security.
- Mascara not perfect and several changes were introduced to improve it.
- Runtime was improved by removing redundancy, searching for parameters was optimised and a new approach was introduced into the evaluation.

...

...

Future Works

...

- Find better training corpus. Current corpus has a huge tail in terms of bigrams
- Introduce more powerful intermediate heuristics to make the generated passphrase meaningful like PoS guided *nextWord*
- Try experimenting with trigrams, but have to be careful, as noise and overhead are easily introduced.
- CER not exactly a perfect measure to capture memorability, as typing error and memorability are not very well related.

...

...

Thanks!