Cracking password tendencies: additive vs subtractive changes in password creation

Lawrence Liu (LWL@mit.edu)
9.66 Final Project

Abstract

When asked to strengthen passwords, users often add characters, mirroring a broader cognitive bias toward additive changes. We study password editing as a constrained concept-editing model to fit human survey results (n=100, 400 total password edits). We find that additive biases dominate in unconstrained settings and while deletions, which often provide good security-memory tradeoff, are underutilized. We develop a utility based cognitive model that attempts to capture the importance of security, memory, and rationality.

Introduction

When tasked with strengthening a password, people often add information in predictable patterns, such as numbers, special characters, or extra words (Mazurek et al., 2013). This is often a result of password box prompts - “Your password is too weak,” “Your password is too short,” “Your password must contain a number or special character.” After all, it is known that the password length is the primary factor in characterizing password strength (Kelley et al., 2012). Thus, tech savvy users are likely biased to be additive when prompted to edit their passwords, which mirrors the cognitive bias that humans tend to perform additive rather than subtractive updates when modifying complex concepts even when subtraction would lead to better outcomes (Adams, Converse, Hales, & Klotz, 2021).

From a cognitive perspective, password modification extends beyond the realm of security and becomes a classic constrained concept-editing problem. With a base password in mind, users often need to transform the password such that it fits website specific constraints in length and complexity as well as self imposed constraints of memorability and coherence. While not explicitly tested in Adams et al., it may be reasonable †o hypothesize that people tend to prefer additive additions in password editing. However, it is important to note that interrupting predictable structure, like keyboard walks (”qwerty”, ”asdf”, ”12345”) and full words, through use of deletions can increase strength. Thus, this begs an interesting question of how humans intuitively modify passwords under various prompts and constraints. Our goal is to gain insight into how people approach these transformations by fitting these competing constraints to human data.

This domain provides a natural, real-world concept-manipulation setting where humans update an internal generative model under constraints. A few factors makes this an incredibly rich domain for computational cognitive science research. Firstly, security improvements can be measured quantitatively, independently of the users (entropy, pattern removal, presence of common words). Additionally, memorability and effort impose cognitive loads on the user. Finally, as mentioned before, users should be biased towards additive complexity even when it may not always be the best choice.

In this paper, we will formalize password modification and investigate how users modify passwords under various scenarios and constraints. By collecting data from survey respondents, we aim to first show that results from Adams et al. extend to the password domain. Then, using a utility based optimization process, we will fit a model that quantifies password modification selection strategies. We aim to provide insight into how users naturally approach the task of password modification unconstrained, and how users adjust their approaches with constraints.

Literature Review

Previous literature on password security has shown that human generated passwords exhibit regular patterns, shaped by a desire for usability over randomness and security. Large scale empirical studies on password security have shown that humans prefer using predictable patterns like common words, keyboard walks, and incremental changes (Florencio & Herley, 2007; Gaw & Felten, 2006). Although password length is a large factor in guess-ability (Kelley et al., 2012), later work has shown that internal structure and common edits play an equally important role (Ur et al., 2015). Leveraging these large scale studies, Wheeler developed a low budget password strength evaluator zxcvbn(Wheeler, 2016). The model leverages common trends in password structure to approximate and rank password likelihoods. This open source model is still widely utilized online to quickly determine password strength. These findings indicate that password security, a field that should be largely governed by information theory and probability, is heavily impacted by human cognition in practice.

Separately, there is a growing field of cognitive science that investigates how humans modify the world around them. Adams et al. has shown that across numerous domains, people systematically prefer additive changes over subtractive changes, even when the subtractive changes requires fewer actions/effort. Although additive bias is seemingly universal part of human nature, recent work has shown that culture, task, and age all impact the bias, markedly demonstrating that the bias is influenced by both nature and nurture (Juvrud, Myers, & Nyström, 2024). While the bias has been studied in abstract ahd physical tasks, its role pass word modification, a real world potentially high stakes cognitive domain, has received little attention.

To model how participants approach password modification, we draw on the framework that human behavior can be modeled as a utility-based choice. Systems of discrete choices and softmax decision rules have been widely used for decades to model tradeoffs between rewards, cognitive load, and preferences (Luce et al., 1959; McFadden, 1972). More recent work has developed the idea of resource-rational modeling which has formalized how pressures like time constraints and task constraints can increase apparent rationality ((Lieder & Griffiths, 2020). These models provide a great foundation for the problem of password modification under constraints while balancing the tradeoffs between security and memorability.

Methods

The high level layout of this project can be split into three sections: the data collection, exploratory data analysis, and cognitive model development. The data collection and exploratory data analysis were used to produce many of the same charts that Adams et al. produced on tasks. The cognitive model aimed to quantify the biases and cognitive loads on participants.

Data Collection

Survey Setup All survey frames can be found in the Appendix A. Participants were recruited through a one-time mass email sent to MIT Dormspam, a collection of opt-in mailing lists for students living in MIT undergraduate dorms. Participation was completely voluntary, with no compensation provided offered. Upon accessing the survey, all participants were provided the following as the motivation for data collection:

```
This study investigates how people modify
passwords when asked to strengthen them.
You'll work with sample passwords and modify
them according to specific instructions.
```

All participants were able to withdraw from the study at any time without penalty. No data was recorded until they completely finished the entirety of the survey. Additionally, the participants were provided with the following disclosure.

- Do NOT use any of your own passwords in this study
- This form is completely anonymous
- All data will be deleted after the final project is completed
- You will be asked to modify 4 different passwords under different conditions. Each task will ask you to retype the modified password to ensure memorability

After consenting to the survey, participants completed a series of 4 password modification tasks in a fixed order. To ensure that participants were not simply mashing keys or choosing arbitrary strings of characters, participants were informed that they would need to retype the password on the following page without the ability to go back.

Tasks The data collection included four tasks presented in the same set order for each participant. In each task, the format was very similar - participants were asked to ”strengthen” a password by adding, removing, or changing any characters. Tasks 1 and 2 shared the same set of 2 base passwords (”Password12345”, ”MITPassword!”), and participants received one of the two options for **both** tasks 1 and 2. Tasks 3 and 4 shared the same set of 2 base passwords (”ninepointsixsixzero”, ”iloveyou123456789”) that were randomly assigned in a uniformly random order such that each participant saw each of these base passwords once.

1. Task 1 asked users to strengthen the password. This question imposed no constraints on edit type, length of password, similarity of modified password to original, or time allowed to make changes.
2. Task 2 gave users the same password as Task 1 with the same goal. However, this question only allowed users to use deletions. There were no other constraints.
3. Task 3 gave users a new password and asked users to strengthen the password. This question was effectively the same as task 1, except the passwords came from a different set of base passwords.
4. Task 4 gave users a password and gave users exactly 20 seconds to strengthen the password. A countdown timer was clearly visible on the screen as the users worked. An extra five seconds were added at the end if the user had not submitted to ensure that they were not cut off mid keystroke.

Exploratory Data Analysis

Parameterizing the Data After collecting responses from participants, we ended up with n=100 responses to the survey and a total of 400 password modifications. Because the password edits spanned from simple deletions to complete rewrites with strange characters (and one user who pasted the entirety of ”The Bee Movie” script as a password), it was necessary to parameterize the password edits into a structured representation to lower dimensionality. To begin, all passwords were truncated to 72 characters, the maximum allowed by zxcvbn for evaluating password strength. 72 characters is sufficiently long that there is no password guesser that can guess it within a century. 2 passwords were truncated to 72 characters.

We formalized password transformations using a hierarchy of 3 conceptual levels: coarse edit operations, semantic transformation, and strategy level classifications.

1. Level 1: Coarse edit operations. At this level, we simply classified changes as coarse grained changes to the structure of the password, including changes in password character length, edit distance metrics, and counts of characters added, deleted, switched, and left unchanged. Each of the 400 passwords modified were assigned a dominant strategy based on the edit that was most common. Level 1 features purely measure changes in the password characters, without consider semantics or strategies.

2. Level 2: Semantic features. At this level, we identified commonly occurring password features that are important to both security and cognition. These features included the addition or deletion of numbers, special characters, leetspeak substitutions (for example substituting ’a’ with ’@’ or ’S’ with ’$’), letter runs, repeated characters, sequences, and keyboard runs. Together, these features characterize the edits beyond just variants of edit distance.

3. Level 3: Strategy Classification. At this level, we attempted to place password changes into one or more common modification strategies, including appending a suffix, prepending a prefix, inserting or deleting characters in the middle, interrupting substrings, or complete rewrites if the similarity between original and new password was very low. These were encoded as binary variables and variables encoding the total number of strategies employed. This was not a comprehensive list of all strategies that could be employed, but covered common ones. This level went beyond types of edits to capture behavioral patterns.

Using these statistics, we were able to easily recreate many of the main charts and figures from the Adams et al. paper, as seen in the Results section.

Cognitive Model

We developed a utility based cognitive model to understand strategy selection and the impact of various constraints. Using maximum likelihood estimation on the collected behavioral data, we fit the parameters governing the model to explain security-memory tradeoff and the impact of constraints.

Looking at the exploratory data analysis, we saw that the top 6 strategies utilized were target deletion, leetspeak substitution, appending suffix, inserting characters in the middle of the password, complete rewrites, and other (collapsed category of more rare strategies). We wrote the following utility function for each strategy:

$$U(s|c) = \alpha_K \cdot K(s) - \alpha_M \cdot M(s) + \beta_s + \varepsilon$$

where ’s’ denotes the strategy, ’c’ is the task specific context, α_K is the importance of security improvement, α_M is the cost of memorizing the new password, β_s is the strategy specific preference, and ε is stochastic noise.

$K(s)$ is the security of the resulting password, measured using the open source password security analysis package zxcvbn. The values of $K(s)$ are all positive numbers, typically less than 20, that measure the log 10 values of approximately how many guesses a decent password cracker should take to guess the password. The M(s) is the edit distance between the old and new password, measured in number of steps. As the number of changes increase, the memory burden should also theoretically increase, decreasing the utility of the strategy.

Strategy decision was made by a softmax over the 6 strategies, where

$$P(s|c,\theta) = \frac{e^{(\tau U(s|c))}}{\sum_{s'} e^{(\tau U(s'|c))}}$$

The variable τ is the temperature, and it can be thought of as a rationality parameter. As τ increases, the softmax becomes more deterministic, which is considered more rational since the participant is more likely to choose the highest utility option. Lower τ makes the distribution more uniform, decreasing rationality as the participant is more likely to pick lower utility strategies.

We estimated the values of these parameters using Maximum likelihood estimation over θ, with the log likelihood denoted as

$$\mathbb{L}(\theta|D) = \sum_i P(s_i|c_i,\theta)$$

where $\theta = [\alpha_K, \alpha_M, \beta_1, ..., \beta_6, \varepsilon]$

This likelihood function was then optimized over 500 maximum iterations with the bounds $\alpha_K, \alpha_M \in [0.01, 10], \beta_s \in [-5, 5], \tau \in [0.1, 10]$, and initializing $\alpha_K = \alpha_M = 0.5$, $\beta_s = log(\frac{count(s)}{\sum_{s'} count(s')})$. Note that in task 2, β_s for everything except for ”targeted_deletion” and ”other” were set to -∞ since we only allowed the users to delete characters or do nothing; this forces softmax to only assign non zero probabilities to these two strategies. Since the model was developed with the data in mind (ie choosing the top 6 most common edits), we did k-fold cross validation to check if parameters were overfitting to the data.

Results

Exploratory Data Analysis Results

Firstly, after running all 400 password modifications through the 3 levels of transformations, we plotted the primary level 1 coarse edit operations distribution for each task in Figure 1.

As we can see in task 1, the additive changes were the strongest, taking up 50% of all dominant edits, while subtractive edits were at 8%. When unprompted, participants greatly favored additive edits to subtractive edits. Interestingly, while task 3 was as unconstrained as task 1, the proportion of subtractive edits greatly increased to over 35% while additive edits decreased to 36%. This may have indicated that priming participants by only allowing deletions in task 2 increased the likelihood they choose to consider deletions in task 3.

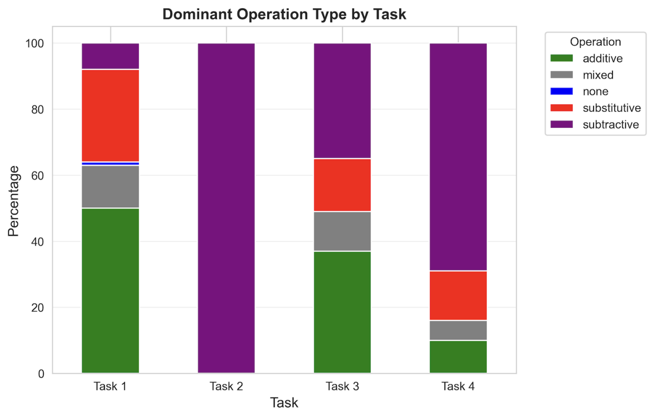


Figure 1: Distribution of coarse edit operations by task. Note that task 2 only allowed subtractive edits.

Additionally, as was mentioned previously, deletions may increase security even if they the length of the password. We can see this effect in Figure 2. The security of the deletions in red lie in the same realm of range of security as many of the complete rewrites and leetspeak substitutions while carrying lower memory burden.

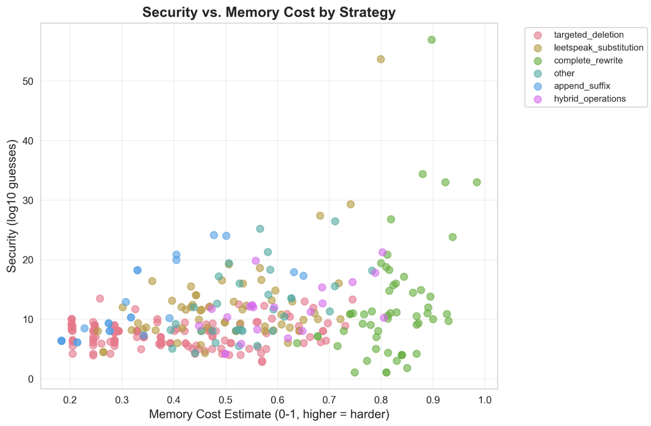


Figure 2: Security vs Memory cost tradeoffs of various password edits

These plots demonstrated that Adams et al.'s additive update bias persists in the password domain, and that passwords made shorter through deletions can still be more secure than passwords edited through other additive changes.

Cognitive Model Results

We then fit the cognitive model to the data by task. Since each task had 100 responses, with 5-fold cross validation, the train was 80 samples and the validation was 20 samples each time. All optimization methods converged within 500 iterations, with the final results displayed in Table 1.

From the parameters in Table 1, we see that the negative log likelihood is high for both task 1 and 3, while marginally lower for task 4. The negative log likelihood is naturally low for task 2 since there were only two possible options (deletion or no change) and most chose deletion. We also see that the security and memory coefficients did not change at all in any of the tasks; this could be due to the factors having weak predictive power as well as the betas being able to absorb arbitrary values.

However, we see interesting variation in the τ rationality parameter across tasks. In task 1 and 3, τ is very low, meaning that the participants were essentially choosing primary edit strategies without much consideration of their utility. Logically, the rationality in task 2 should be quite high since the participants didn't a choice in what they could do. Interestingly, the rationality in task 4 is also high. It seems that when the participants were time constrained, they defaulted to being more rational, significantly more so than when unconstrained as in tasks 1 and 3.

Looking at the strategy preference betas, we can see that in the unconstrained non-primed case (task 1), the targeted deletions had a negative beta. After going through a task that only allowed deletions, the targeted deletion beta flipped to positive in task 3. Additionally, it's interesting to see that all betas were negative in task 4, alongside the higher τ. This implies that participants tended to avoid large changes and prioritized rationality when faced with time constraints.

Finally, let's consider the cross validation results. Since there were originally 6 strategies, a baseline out of sample accuracy is $\frac{1}{6}$. Tasks 1, 2, and 4 achieved much higher accuracies. Task 3 out of sample accuracy being quite low as well as task 1 standard deviation being nearly twice as high as task 2 and 4 indicates that this model struggles to fit unconstrained tasks.

Discussion

Limitations and Future Work

Before discussing some of the potential takeaways from this project, we must address the limitations to frame the takeaways. Firstly, without constraints, the password space is too large to search or fit directly. Labeling each password a single primary edit strategy glosses over a lot of the nuance in the password edits. There is also certain uncertainty in the labeling process; classification for items like leetspeak and keyboard runs are done with common substitutions and runs, respectively. The list is certainly not comprehensive. However, with only 100 samples per task, using a limited set of parameters with common definitions may be the best we could do on the front of reparameterization. Fitting a model by considering character to character changes or more factors would have been computationally infeasible or caused significant overfitting. Additionally, basing the cognitive model off of the exploratory data analysis is, in a sense,

Table 1: Fitted Model Parameters by Task

Parameter	Task 1 (Baseline)	Task 2 (Deletions)	Task 3 (Baseline)	Task 4 (Time Constrained)
N	100	100	100	100
$-\mathbb{L}(\theta)$	162.1	32.5	166.4	148.7
α_K	0.5	0.5	0.5	0.5
α_M	0.5	0.5	0.5	0.5
τ (rationality)	0.242	3.666	0.148	1.849
Strategy Preferences (β_s)				
Targeted deletion	-2.83	1.188	1.63	-1.18
Leetspeak	1.93	$-\infty$	2.79	-1.51
Complete Rewrite	-0.719	$-\infty$	4.912	-1.588
Append Suffix	0.957	$-\infty$	-5.00	-2.43
Insert Middle	0.034	$-\infty$	-3.958	-2.805
Other	4.797	0.589	4.912	-1.793
Cross Validation Results				
Out of sample accuracy	0.380 ± 0.121	0.900 ± 0.063	0.190 ± 0.080	0.400 ± 0.071

circular model building. We try to mitigate some of these concerns by doing k-fold cross validation to get a sense of how well the model would actually perform out of sample. However, optimally, we would develop a model completely independently of the data.

With these limitations in mind, we can summarize some of the findings from this study.

- Adding time pressure in task 4 increased rationality of participants and suppressed all strategies. Looking more closely, we can see that the more complex and less defined strategies, complete rewrite and other, decreased from 4.912 in task 3 to -1.588 and -1.793 in task 4, respectively. These negative deltas were the largest decreases out of the 6 strategies. These changes imply that under time constraints, participants stay more rational and try to choose simpler changes.

- We did not get to see any changes in α_K and α_M, weakly indicating that the security and memorability tradeoffs were not as significant as the strategy choice. However, there could have been other reasons these values did not change; to start, the models were less than 50% accurate across tasks 1, 3, and 4. There may not have been enough signal to distinguish the alphas in the first place.

- While task 1 showed a strong additive bias and task 3 showed a strong subtractive bias, there are numerous explanations why this may have been the case. The priming from task 2 may have put the idea of subtractive changes into the front of participants' minds as they entered task 3. Additionally, task 3 and 4 base passwords were longer to being with; average character length of task 1 and 2 passwords were 12.5 characters while average character length of task 3 and 4 passwords were 18 characters.

Future work could aim to address many of these limitations. For one, the utility functions could be designed to resolve some of the identifiably issues, either by using multiplicative utilities or adding strategy specific memory and security tradeoffs. Collecting more data could significantly aid in avoiding overfitting when the model contains more features. Future work could also attempt to completely separate the train and test sets.

Conclusion

Through this research, we demonstrated that the additive bias explained in Adams et al. also exist in the password domain. We tried to create a utility based model to model the password edit choices, but faced challenges re parameterizing the human dataset. Ultimately, it was seen that rationality increases and the use of simple strategies increase with edit and time constraints.

Code

All code can be accessed on github at

```
https://github.com/Bookmaster9/
password_editing_additive_bias
```

References

Adams, G. S., Converse, B. A., Hales, A. H., & Klotz, L. E. (2021). People systematically overlook subtractive changes. *Nature*, *592*(7853), 258–261.

Florencio, D., & Herley, C. (2007). A large-scale study of web password habits. In *Proceedings of the 16th international conference on world wide web* (pp. 657–666).

Gaw, S., & Felten, E. W. (2006). Password management strategies for online accounts. In *Proceedings of the second symposium on usable privacy and security* (pp. 44–55).

Juvrud, J., Myers, L., & Nyström, P. (2024). People overlook subtractive changes differently depending on age, culture, and task. *Scientific Reports*, *14*(1), 1086.

Kelley, P. G., Komanduri, S., Mazurek, M. L., Shay, R., Vidas, T., Bauer, L., … Lopez, J. (2012). Guess again (and again and again): Measuring password strength by simulating password-cracking algorithms. In *2012 ieee symposium on security and privacy* (pp. 523–537).

Lieder, F., & Griffiths, T. L. (2020). Resource-rational analysis: Understanding human cognition as the optimal use of limited computational resources. *Behavioral and brain sciences*, *43*, e1.

Luce, R. D., et al. (1959). *Individual choice behavior* (Vol. 4). Wiley New York.

Mazurek, M. L., Komanduri, S., Vidas, T., Bauer, L., Christin, N., Cranor, L. F., … Ur, B. (2013). Measuring password guessability for an entire university. In *Proceedings of the 2013 acm sigsac conference on computer & communications security* (pp. 173–186).

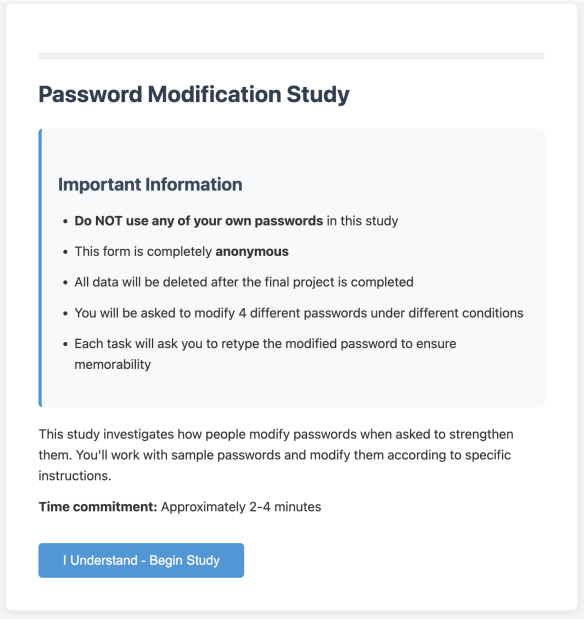
McFadden, D. (1972). Conditional logit analysis of qualitative choice behavior.

Ur, B., Noma, F., Bees, J., Segreti, S. M., Shay, R., Bauer, L., … Cranor, L. F. (2015). ” i added’!’at the end to make it secure”: Observing password creation in the lab. In *Eleventh symposium on usable privacy and security (soups 2015)* (pp. 123–140).

Wheeler, D. L. (2016). zxcvbn:{Low-Budget} password strength estimation. In *25th usenix security symposium (usenix security 16)* (pp. 157–173).

Appendix

A. Survey Frames



Password Modification Study

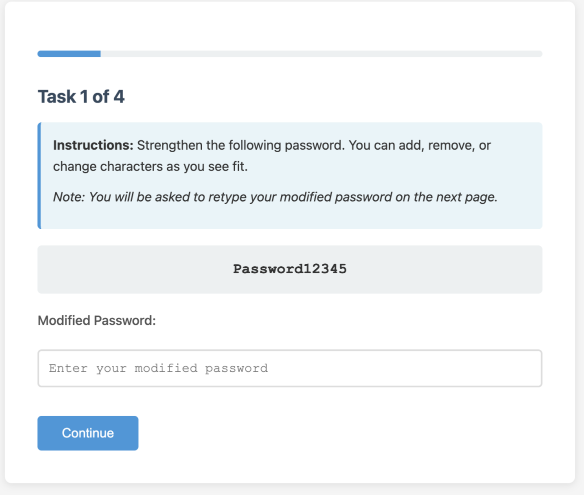
Important Information

- **Do NOT use any of your own passwords** in this study
- This form is completely **anonymous**
- All data will be deleted after the final project is completed
- You will be asked to modify 4 different passwords under different conditions
- Each task will ask you to retype the modified password to ensure memorability

This study investigates how people modify passwords when asked to strengthen them. You'll work with sample passwords and modify them according to specific instructions.

Time commitment: Approximately 2-4 minutes

Figure 3: Survey Page 1 (Disclosure page)



Task 1 of 4

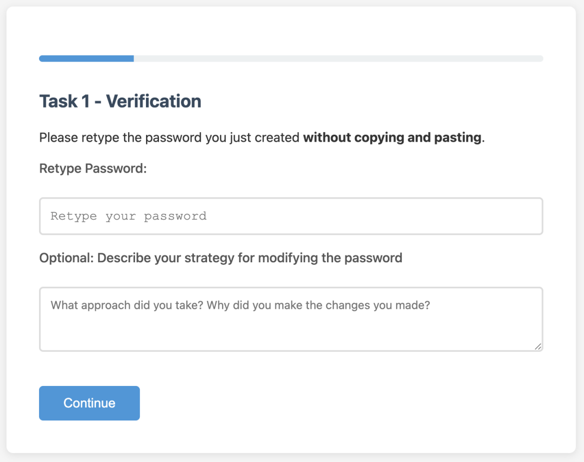
Instructions: Strengthen the following password. You can add, remove, or change characters as you see fit.

Note: You will be asked to retype your modified password on the next page.

```
Password12345
```

Modified Password:

Figure 4: Survey Page 2 (Enter Task 1 password)



Task 1 - Verification

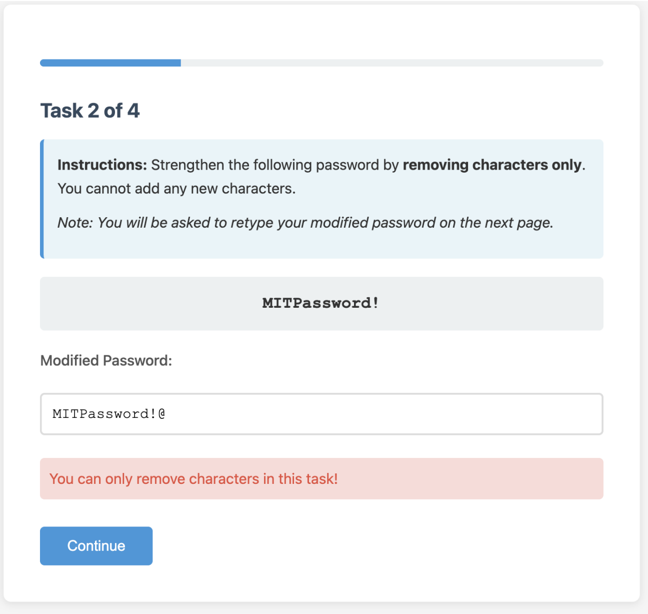
Please retype the password you just created **without copying and pasting**.

Retype Password:

Optional: Describe your strategy for modifying the password

What approach did you take? Why did you make the changes you made?
Continue

Figure 5: Survey Page 3 (Reenter Task 1 password)



Task 2 of 4

Instructions: Strengthen the following password by **removing characters only**. You cannot add any new characters.

Note: You will be asked to retype your modified password on the next page.

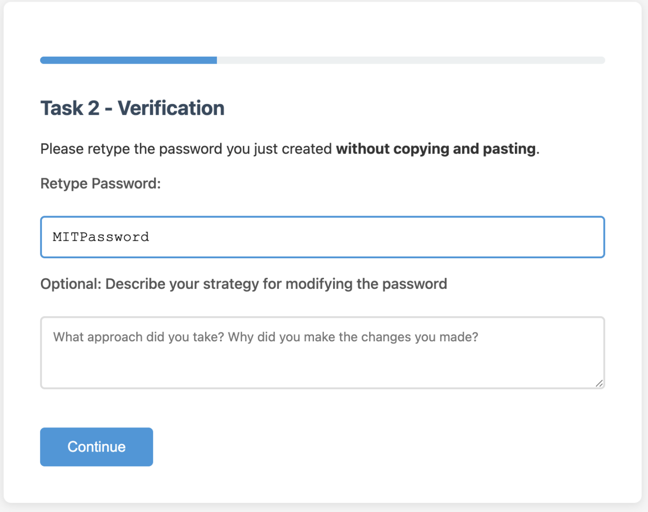
MITPassword!

Modified Password:

MITPassword!@

You can only remove characters in this task!

Figure 6: Survey Page 4 (Enter Task 2 password. Warning provided if the entered password uses any operations except for deletion.)



Task 2 - Verification

Please retype the password you just created **without copying and pasting**.

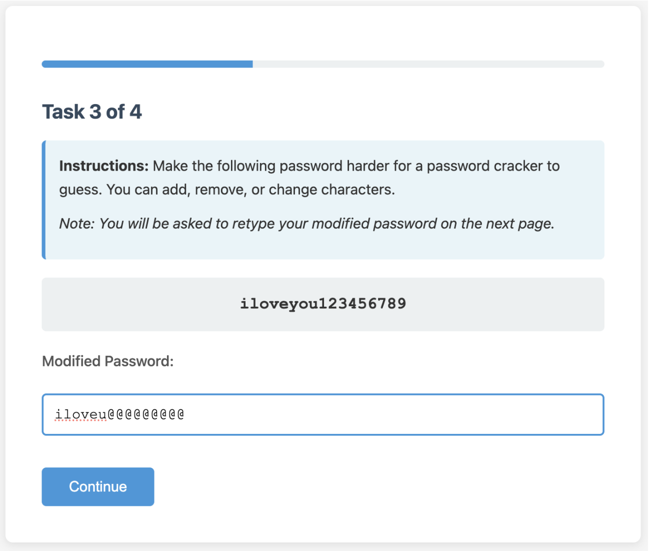
Retype Password:

MITPassword

Optional: Describe your strategy for modifying the password

What approach did you take? Why did you make the changes you made?

Figure 7: Survey Page 5 (Reenter Task 2 password)



Task 3 of 4

Instructions: Make the following password harder for a password cracker to guess. You can add, remove, or change characters.

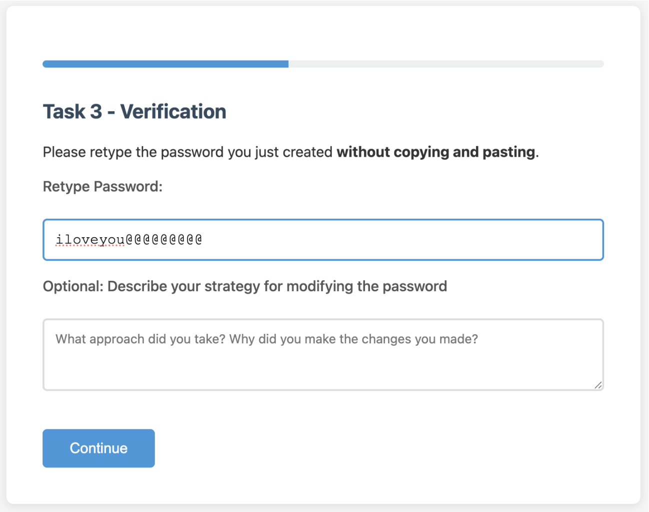
Note: You will be asked to retype your modified password on the next page.

iloveyou123456789

Modified Password:

iloveu@@@@@@@@@

Figure 8: Survey Page 6 (Enter Task 3 password)



Task 3 - Verification

Please retype the password you just created **without copying and pasting**.

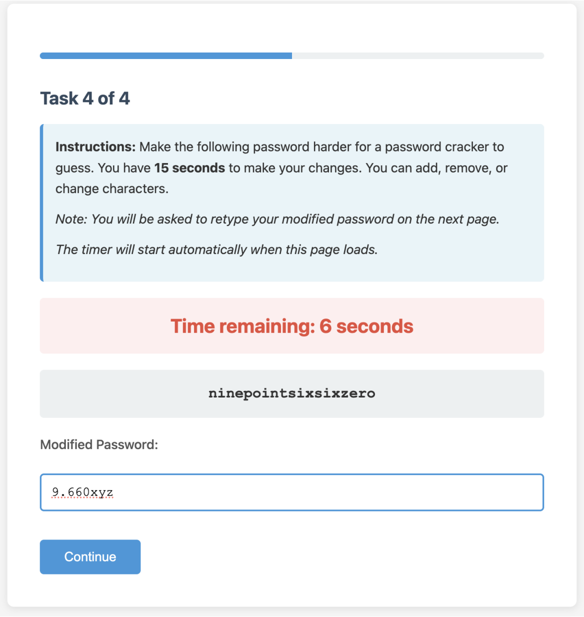
Retype Password:

iloveyou@@@@@@@@@

Optional: Describe your strategy for modifying the password

What approach did you take? Why did you make the changes you made?

Figure 9: Survey Page 7 (Reenter Task 3 password)



Task 4 of 4

Instructions: Make the following password harder for a password cracker to guess. You have **15 seconds** to make your changes. You can add, remove, or change characters.

Note: You will be asked to retype your modified password on the next page.

The timer will start automatically when this page loads.

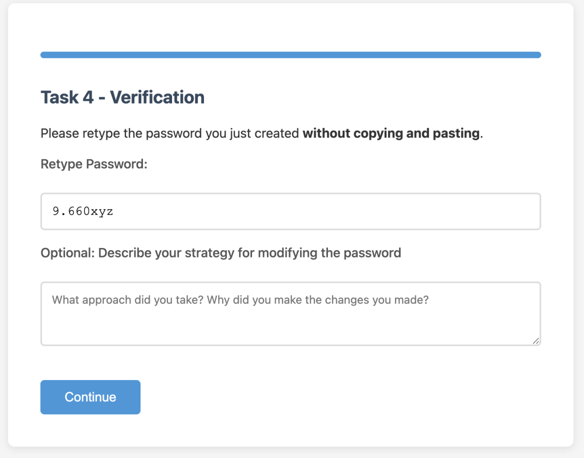
Time remaining: 6 seconds

```
ninepointsixsixzero
```

Modified Password:

```
9.660xyz
```

Figure 10: Survey Page 8 (Enter Task 4 password, with visible countdown timer)



Task 4 - Verification

Please retype the password you just created **without copying and pasting**.

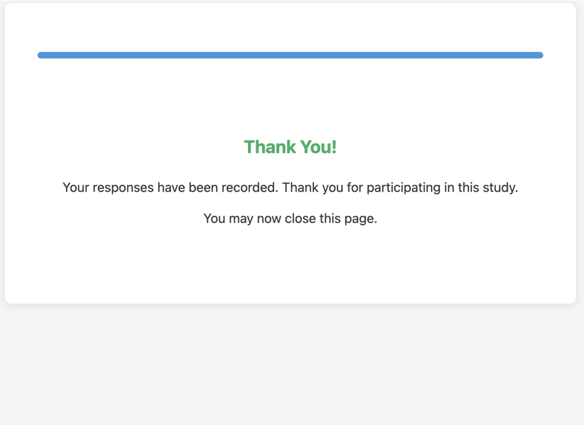
Retype Password:

```
9.660xyz
```

Optional: Describe your strategy for modifying the password

What approach did you take? Why did you make the changes you made?

Figure 11: Survey Page 9 (Reenter Task 4 password)



Thank You!

Your responses have been recorded. Thank you for participating in this study.

You may now close this page.

Figure 12: Survey Page 10 (Final page)