

USENIX

THE ADVANCED COMPUTING SYSTEMS ASSOCIATION

How Users Enter Generated Passwords on Non-Desktop Devices

John Sadik, Joshoua Rodriguez, Hector Santos Villalobos, and Scott Ruoti, *The University of Tennessee, Knoxville*

https://www.usenix.org/conference/soups2026/presentation/sadik

This paper is included in the Proceedings of the Twenty-Second Symposium on Usable Privacy and Security.

August 24–26, 2026 • Hannover, Germany

ISBN 978-1-939133-59-5

Open access to the Proceedings of the Twenty-Second Symposium on Usable Privacy and Security is sponsored by USENIX.

How Users Enter Generated Passwords on Non-Desktop Devices

John Sadik
The University of Tennessee, Knoxville

Joshoua Rodriguez
The University of Tennessee, Knoxville

Hector Santos Villalobos
The University of Tennessee, Knoxville

Scott Ruoti
The University of Tennessee, Knoxville

Abstract

Generated passwords remain a powerful tool for account security, yet they are often avoided due to fear of manual entry. Developing better generated passwords requires a fine-grained understanding of how they are entered in practice, but no such public dataset currently exists. We contribute the first such dataset, containing nearly 80,000 keystrokes from 4,916 password entries collected from 99 participants using five different input modalities. Our analysis reveals that generated password entry is driven by physical kinematic costs, such as layout switching, rather than the interface-level treatments in our study. Using these insights, we design two novel device-aware password generation algorithms as proof-of-concept models optimizing for travel distance and rhythmic entry flow. We then leverage machine learning models trained on our dataset to analytically validate these algorithms, finding that they reduce the predicted entry time by 27.55% and 31.28% respectively compared to standard randomly generated passwords while maintaining sufficient cryptographic entropy. We release our research artifacts to promote the development of new password generation algorithms, and discuss the how these insights can inform better password authentication designs.

1 Introduction

Passwords are ubiquitous despite their well-known issues [7, 10, 16, 20, 38], driven in part by their key role as a backup authentication method and their default status in many systems. Even with password-less alternatives such as passkeys and Single Sign-On (SSO), passwords remain the default for many platforms, especially Internet of Things (IoT) devices and legacy systems, which may lack support for password-less authentication.

Yet passwords carry their own costs, such as the difficulty of remembering strong passwords, leading users to create weak passwords or reuse them across accounts. Password managers aim to address these issues by replacing weak, human-chosen passwords with strong, randomly generated ones, ideally obviating direct interaction with those passwords. Unfortunately, while research has shown that users appreciate password storage and autofill [13, 39, 41], measurements have shown that users rarely adopt generated passwords [31, 37, 39].

Research has suggested this rejection is due to the difficulty of entering them on devices where autofill is unavailable [37]. Prior work has shown that users still rely on manual password entry at least semi-regularly [43]. This includes mobile devices, where password managers are considered unreliable [37, 44], as well as game consoles and smart TVs, where entry is slow and tedious [43].

One promising approach is to build password generation algorithms that account for the physical constraints of the devices on which they will be entered. However, building device-aware password generation algorithms is hindered by the lack of data on how users enter generated passwords on these devices. To bridge this gap, we collect the first fine-grained dataset of how users enter generated passwords and leverage our findings to construct and validate novel password generation algorithms that balance security and usability.

To collect our dataset, we conducted an in-person user study in which 99 participants entered 4,916 generated passwords across five input modalities: keyboard, touchscreen, air mouse, TV remote, and game controller. We release this dataset at https://zenodo.org/records/18711498. Complementing this dataset, we also release the source code for our machine learning models, allowing researchers to use

Copyright is held by the author/owner. Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee.
USENIX Symposium on Usable Privacy and Security (SOUPS) 2026.
August 23–26, 2026, Hannover, Germany.

these models to predict password entry time without new user studies.

We then analyze our dataset to address three research questions:

- **RQ1:** What are the input characteristics of entering generated passwords on various devices?
- **RQ2:** How does password composition impact the above characteristics?
- **RQ3:** To what extent do specific interface interventions (i.e., coloring the password character classes, deleting wrong entries) impact the authentication experience and input characteristics?

Key findings from our analysis include:

1. **On keyboards and touchscreens, generated password entry is fast and error-resistant.** While entry errors are relatively common on these devices, the speed of entry makes such errors a minor inconvenience. This fact, along with the community's focus on desktop devices, may explain why little effort has been applied to creating more usable password generation algorithms.
2. **On air mice, TV controllers, and gamepads, generated password entry is slow, with errors uncommon.** An exception to this is generated passphrases, which are both slow and prone to entry mistakes. This demonstrates the importance of creating device-aware generation algorithms, as schemes that work well on desktops (i.e., passphrases) can be quite problematic on non-desktop devices.
3. **Lowercase passwords are the fastest option across all devices.** This holds true even when accounting for the additional length needed to match the security of more complex compositions. This is a promising direction for improving generated password usability on non-desktop devices.
4. **The tested password entry interface treatments had a negligible impact on generated password entry.** For example, color-coding generated passwords to differentiate look-alike characters did not improve entry time or reduce mistakes. This is a surprising result, as many password managers and websites implement these features.
5. **Device-aware password generation can overcome device constraints.** We present two novel device-aware password generation algorithms as proof-of-concept demonstrations: a physics-constrained foraging algorithm that uses spatial heatmaps to ensure ergonomic coverage without local minima, and a kinetic chain generation algorithm that optimizes for rhythmic motor flow using probabilistic "soft constraints." We use machine learning models trained on our dataset to show that these algorithms reduce the expected entry time by 27.55% and 31.28% respectively compared to standard random generation baselines, confirming that device-aware generation can meaningfully improve usability without sacrificing security.

2 Related Work

We first discuss the broader authentication landscape, then organize related work by research question.

2.1 Passwords and Authentication

While typically associated with desktops and laptops, passwords are still widely used across a variety of devices, such as smart TVs and game consoles [43]. Research has explored different authentication approaches on these devices, with cross-device authentication—where a secondary device is used to authenticate on the primary device—being a prominent approach for many use cases [56]. Similarly, FIDO2-based passkeys have emerged as a promising cross-device, password-less alternative [32, 53], though they lack widespread adoption and suffer from their own usability issues. Because these mechanisms are inherently platform-dependent, users cannot always control their availability—and when they fail or are unavailable, passwords remain the de facto fallback. Empirical evidence confirms this occurs even among password manager users [43], suggesting manual password entry on non-desktop devices is a persistent interaction worth closer examination.

2.2 RQ1: Input Characteristics

Two fundamental laws of text entry frame our analysis. Fitts' Law [15] models the time required to move to a target area. On a game controller or remote, the "distance" to a key is measured by the number of button presses required to navigate the grid. Therefore, layout switching, which involves navigating to a specific button to change the character set (e.g., from letters to symbols), incurs a significant penalty. Hick's Law [21] states that decision time increases with the number and complexity of choices. For users searching for characters on unfamiliar virtual keyboards, especially special characters, this elevates cognitive load and slows entry time.

These physical costs are compounded by well-documented user behavior: research has consistently shown that users trade security for usability when selecting passwords [43], selecting weak passwords when the password policy allows it [11, 16, 35, 50]. This focus on usability also holds true for password managers [2, 13, 25, 39, 41], causing users to

eschew generated passwords [31, 37, 39]. Users attribute this behavior to concern over needing to manually enter generated passwords when autofill is unavailable, and our prior work [43] confirms that manual password-based authentication on non-desktop devices still occurs semi-regularly.

We found only a single instance of research focusing on improving the usability of generated passwords: Greene et al. [17, 19] explored grouping characters by class, finding users preferred these modified passwords over randomly generated ones. However, further progress in this area is hindered by the lack of entry data—while some data exists for user-selected passwords on keyboards and touchscreens, *there is no public dataset on the input characteristics of generated passwords on any device*, including Greene et al.'s own work [17, 18]. *Our work is the first publicly available dataset on input characteristics of generated passwords.* Given this, we summarize relevant text entry research for each device in our study below.

Keyboards Research has shown that typing decisions—such as finger choice and navigation between characters—significantly impact entry speed [12, 14, 30]. Interestingly, non-traditional typists type as automatically as traditional touch typists, but it is the degree to which they balance Fitts' law [15] and Hick's law [21] that results in their subpar typing speed and accuracy [30].

Touchscreens Compared to desktops, users create weaker passwords on mobile devices, taking longer to enter them and incurring a higher error rate when doing so [19, 28, 36, 54]. Prior work has explored improving touchscreen password entry, such as grouping character classes to avoid complex layout switches [17–19], or leveraging passphrases [22], but none have been widely adopted. Furthermore, system-generated passphrases have already been shown to perform poorly on desktops [45], and we expect that this penalty is likely even worse on touchscreens. Work has also investigated whether hiding or showing a password affects entry speed and error rate, but the effect size was quite small [40].

TV entry We could not find any research on how users enter text-based passwords with a standard remote controller or an air mouse, representing a gap our work addresses. For TV interaction more broadly, past research has shown that air mice are slow and produce complaints of fatigue [3], and that the input modality impacts usability [5, 9, 49], especially for the elderly and children [29]. This is consistent with Fitts' Law, where the linear navigation required by standard remotes maximizes the distance between characters on opposite sides of the virtual keyboard, making it more difficult to use than an air mouse or touchscreen.

Game controllers Research on game controllers has focused more on the security of password entry, such as shoulder-surfing resistance [34], rather than the input characteristics of entering passwords or how composition impacts entry. The slow entry speed on game controllers (6–15 WPM [26, 51]) has similarly motivated research into novel entry schemes [34].

2.3 RQ2: Composition and Entry Speed

Work by Greene et al. [17–19] has shown that password composition can affect entry speed on touchscreens. There is also work showing that complex passwords are viewed negatively by users [1, 46, 47], though without entry time data. *As such, there is a need for more work exploring how composition impacts the entry of passwords.*

2.4 RQ3: Interface Design and Entry Speed

Prior work has shown that password entry interface design can impact authentication usability [40]. Similarly, prior work has theorized that clearly displaying character classes when presenting a generated password can make it easier to distinguish look-alike characters and aid password entry [48]. *We formally test this hypothesis and measure how specific interface designs impact password entry.*

3 Methodology

We conducted a user study to gather fine-grained data on how users enter generated passwords using five devices, recruiting participants via flyers and social media. Conducted in-person between February 13 and April 3, 2024, the study collected 4,916 password entry events from 99 participants. Participants entered either 25 or 50 passwords in total, depending on their chosen study duration, and received $10 or $20 respectively. The user study was approved by our Institutional Review Board (IRB).

3.1 Study Setup

Our laboratory study focuses on scenarios where the users must enter generated passwords manually, mimicking situations where autofill is unavailable but the user can reference the saved password. We standardized length and entropy across participants to ensure that the differences in entry between participants are due to the input modalities and interface treatments, not differences in password complexity.

Upon arrival, the participants were asked to provide informed consent and were directed to Qualtrics to complete the study on a provided laptop. When loaded, the survey automatically assigned participants a treatment (see Section 3.2).



Figure 1: An example of a password with coloring

The survey began by asking participants basic demographic questions and whether they were colorblind. Then the study coordinator assisted participants in navigating to the entry website (see Section 3.1.1) and provided them with the appropriate input modality (see Section 3.1.2) for their tasks. After using the given device to enter passwords, participants answered questions about their experience and repeated the process for the remaining devices. Finally, participants gave overall feedback about their experience with the different input modalities. The full IRB-approved version of our study is found in Appendix A.

3.1.1 Study Website

To capture fine-grained data and control the virtual keyboard layout, we created a custom website for our study (see Appendix B). The website displays the target password in a separate box, mimicking the real-world scenario of referencing a saved password during entry. To preserve ecological validity, we based our virtual keyboard design on the most common designs observed in the wild, specifically modeling them after standard implementations found on Android tablets and smart TV applications. Notably, the tablet virtual keyboard featured number keys integrated directly onto the main letter layout. We enable natural haptic feedback where available and include device-specific shortcuts (e.g., the Xbox controller's shortcuts).

3.1.2 RQ1: Input Modalities Under Test

Our prior work [43] showed that users enter passwords across a wide range of devices, the most common being desktops and laptops, mobile phones and tablets, gaming devices, and smart TVs. Investigating these devices, we identified five input modalities for password entry:

Keyboard Used on desktops, laptops, and tablets. Keyboards are the only input mechanism in our study that does not rely on a virtual keyboard. In our study, participants used the keyboard found on the MacBook Air M2 (see Figure 14 in Appendix C).

Touchscreen Used on laptops and mobile phones. An image of a keyboard—i.e., a *virtual keyboard*—is displayed on the device, allowing users to enter the password by tapping their fingers on the virtual keyboard. In our study, participants used either an iPad Pro (2018) or a Samsung Tab S8 (see Figure 15). Both devices had the same screen size, and neither had discernible touchscreen digitizer lag, giving participants an identical experience with both devices (the virtual keyboard layout was provided by our website, not the system's virtual keyboard).

Air mouse Used on smart TVs. Using a gyroscope, the air mouse enables users to move it through the air to direct the movements of an on-screen mouse cursor. Password characters are selected by hovering the mouse cursor over the letter on a virtual keyboard and then clicking a button on the air mouse. In our study, participants used the Pepper Jobs W10 GYRO Smart Remote (see Figure 16).

TV Remote Used on smart TVs. In its simplest form, a TV remote has five buttons: four directional buttons for navigating a virtual keyboard and one button for selecting a letter on the virtual keyboard. In our study, participants used the Pepper Jobs W10 GYRO Smart Remote (see Figure 16).

Gamepad Used on gaming devices. A directional pad or analog stick on the left side of the gamepad is used to navigate a virtual keyboard, with a button on the right side used to select a letter on the virtual keyboard. Gamepads also have additional buttons that serve as shortcuts for common operations, such as deleting characters and adding spaces. In our study, participants used the Xbox Series X Wireless Controller (see Figure 17).

3.1.3 RQ2: Password Generation

To test how composition impacted entry, we generated passwords using one of five approaches: (i) lowercase letters only ("L"), (ii) lowercase and uppercase letters ("LU"), (iii) lowercase letters, uppercase letters, and digits ("LUN"), (iv) lowercase letters, uppercase letters, digits, and symbols ("LUNS"), and (v) passphrases—a collection of dictionary words separated by a dash ("W"). We chose this set to measure the incremental impact of each additional character class and the feasibility of passphrases on non-desktop devices.

Generated passwords were 10 characters long, comprised of randomly chosen characters from the character classes allowed in that composition. Passphrases were comprised of three random words joined by the '-' character. This approach mirrors what password managers in the wild do when generating passphrases, and we used a word list from the well-known password manager Bitwarden [4].

For each input modality, participants entered 1-2 passwords from each composition depending on their chosen participation duration. Device type and password composition were evaluated as within-subject variables, meaning all participants used all modalities and all compositions. To mitigate ordering and learning effects, we used two separate Latin square assignments: one for the device order and one for the composition order. We used the data collected from these standardized trials to train and validate our novel, device-aware generation algorithms, which we present later in this paper.

3.2 RQ3: Interface Treatments

To investigate whether interface treatments could improve generated password entry, we implemented between-subject treatments in our study: coloring password character class [48] (see Figure 1) and deleting passwords if they were entered incorrectly. Based on suggestions from prior work [48], we selected color-coding for its potential to reduce cognitive load when differentiating similar characters (e.g., 'O' and '0'), thereby reducing mistakes and entry time. We selected the delete condition to reflect a real-world system design choice we expected to impact entry. Specifically, the "delete" condition meant that incorrect submissions cleared the password entry entirely, requiring re-entry from scratch, whereas the "no-delete" condition allowed incorrect entries to persist for correction. Unlike password composition, the interface treatments (color and delete) were evaluated as a between-subject variables. In total, participants were randomly assigned one of four treatments without being told what they received: (i) no-color, no-delete, (ii) color, no-delete, (iii) no-color, delete, and (iv) color, delete.

3.3 Demographics

We had more males (62; 62.6%) than females (37; 37.3%). Participants were all young adults, aged 18–31. All but one had completed high school; 24 (23.8%) held a 4-year degree, with the remainder having some college education. Only two of our participants were colorblind.

3.4 Analysis and Machine Learning

3.4.1 Password metric analysis

We calculated entry metrics (e.g., entry time and error rates) from our dataset, then employed machine learning to identify the factors driving entry time. We used a randomized trial-level split rather than a participant-level split to ensure our models generalize across users rather than memorizing individual patterns.

We trained our XGBoost regressor to predict entry time based on a priori (e.g., character composition) and a posteriori (e.g., error counts) features, leveraging standard techniques like feature engineering, hyperparameter tuning using GridSearchCV, and cross-validation to optimize model performance. Using Recursive Feature Elimination with Cross-Validation (RFECV), we isolated 26 key features, achieving an R^2 score of 0.71, which indicates a strong ability to explain the variance in password entry times based on the selected features. This approach also helps us avoid overfitting and provides a higher level of interpretability regarding feature importance compared to "black box" deep learning models, allowing us to quantify the specific time cost of different error types and password characteristics. Importantly, we extracted the feature weights from this analysis to guide our design of the ergonomic cost functions for our proposed algorithms.

3.4.2 How mistakes are counted.

We split up mistakes into two types: insertion errors and deletion errors. When a user types in the wrong character or continues typing in a password after a mistake, we count that as an insertion error. When a user deletes a correct character, we count that as a deletion error. Deleting a wrong character or deleting characters on the way to fix a previous mistake is not counted as a mistake. This method enables us to consistently count mistakes while also acknowledging that even a single incorrect character can prohibit a user from logging in, as prior work has shown, accounting for up to one in ten failed login attempts [8].

3.4.3 Algorithm Validation Framework

We evaluated our novel algorithms based on entry time, as predicted by our XGBoost models, and entropy, as measured in a Monte Carlo simulation [42]. For the Monte Carlo simulation, we stepped through the generation process of our algorithms, freezing the generator state at each step and running 1,000 simulations to empirically determine the probability distribution of the next character. This yields the Min-Entropy of our state-dependent algorithms, verifying that sufficient security is maintained.

4 Results

4.1 Fine-Grained Generated Password Entry Dataset

The first contribution of this work is our dataset of generated password entry events on a variety of input modalities. This dataset contains high-quality fine-grained data on how 99 participants entered 4,916 generated passwords on a variety of input modalities, and is the basis for the analysis we present in this section.

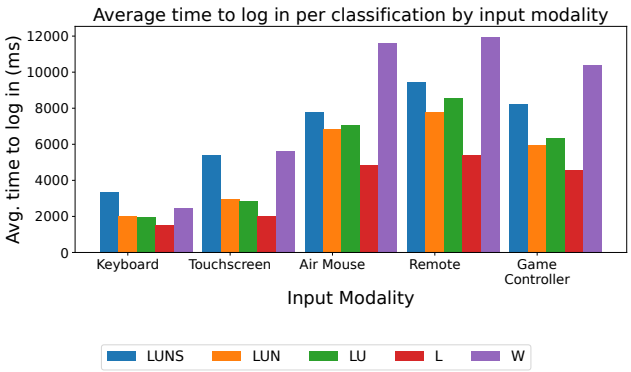


Figure 2: The average (mean) entry time for each character class, grouped by input modality

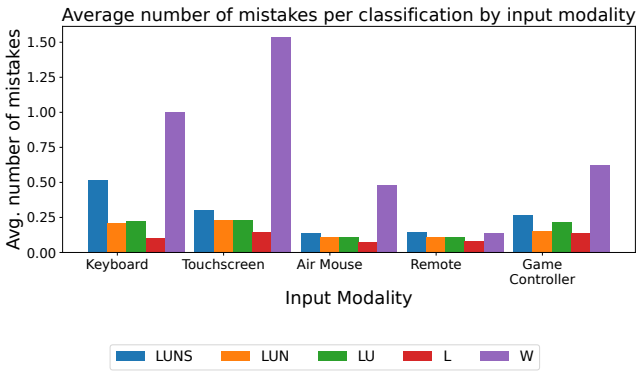


Figure 3: The average (mean) number of mistakes for each character class, grouped by input modality

We store the dataset in two CSV files: a high-level CSV containing information about each password entry trial, and an event-level CSV containing fine-grained information about each character entry event across all trials. We also provide code to help researchers load and analyze the dataset, including example analysis on the dataset that recreates our results and presents additional results beyond what is included in this paper.

4.2 The Impact of the Input Modality

Guided by prior work findings on how important speed is to users [43], we considered speed and mistakes as two key performance indicators for password entry.

4.2.1 Results at a high-level.

Figures 2 and 3 show the average (mean) time to enter a character and the average number of mistakes made for each PCP by input modality. Despite having the most number of mistakes, we see that keyboards and touchscreens are the fastest input modalities to enter a password on.

To explore these differences, we conduct a one-way repeated measures ANOVA test for time and input modality, finding that the input modality had a statistically significant effect ($p < 0.001$) on the time to enter a password. We followed this up with a paired t-test with a Bonferroni correction, finding that all input modalities had statistically significant differences when compared to other input modalities ($p < 0.001$) except for comparing the air mouse and the game controller. These input modalities were potentially the least familiar to participants, which may explain the similar performance. When considering the number of mistakes, we used a linear mixed-effects model (LMEM) with participant as a random effect to test for significance, finding that it had a statistically significant effect ($p < 0.001$) on time (See Appendix D for more details on this analysis).

4.2.2 Fine-Grained mistake analysis.

Our XGBoost model trained to predict entry time had a final R^2 of 0.71 and relied on both a priori features (e.g., password composition) and a posteriori features (i.e., mistakes). Using a similar model, but without a posteriori features (mistakes), we found an R^2 of 0.41, indicating that mistakes account for a significant portion of the variance in how users enter passwords. This suggests that a little under half of the variance in how users enter passwords is in control of the password generation algorithm and system designers, rather than being locked behind user behavior.

To further explore mistakes, Table 1 reports on the mean cost of insertion errors, deletion errors, total wasted actions (any wasted keystroke), and incorrect submissions (any time a user submits an incorrect password). We calculated these costs by using a partial dependence analysis on our trained statistical model (XGBoost), which isolates the marginal effect of a single error type while controlling for the other variables.

The results highlight that the cost for inefficient actions varies significantly across input modalities and is tied to the device's ergonomics. For example, the marginal cost of a single wasted action on a remote (1.24s) is almost four times that of a keyboard (0.33s). For deletion errors, there were not enough occurrences on the remote or air mouse to allow us to calculate a meaningful marginal cost for these input modalities.

Table 1: Average (mean) cost of mistakes per input modality

	Input Modality	Cost (seconds per mistake)
Insertion Errors	Keyboard	0.15
	Touchscreen	0.28
	Remote	0.28
	Air Mouse	0.22
	Game Controller	0.26
Deletion Errors	Keyboard	0.23
	Touchscreen	0.17
	Remote	NA
	Air Mouse	NA
	Game Controller	0.56
Total Wasted Actions	Keyboard	0.33
	Touchscreen	0.57
	Remote	1.24
	Air Mouse	0.54
	Game Controller	0.82
Incorrect Submissions	Keyboard	4.74
	Touchscreen	3.71
	Remote	6.62
	Air Mouse	5.43
	Game Controller	6.47

Key Takeaways: Our analysis reveals that input modalities fit into two main categories: fast but error-prone (keyboards and touchscreens) and slow but less error-prone (remotes, air mice, and game controllers). This divide shows that input-modality-specific optimizations may be a valuable direction for future research, as there is a clear need for improvement on these slow input modalities.

4.3 The Impact of Composition

The composition of the generated password was also a key factor in our analysis, and it represents the most promising way to address challenging password entry on the slow input modalities (remotes, air mice, and game controllers).

4.3.1 Composition at a high-level.

Returning to Figures 2 and 3, we see password entry time and mistakes were both impacted by the password composition. While we expected that less complex passwords (passwords with fewer character classes) would be faster to enter and have fewer mistakes, this is not always the case. For example, LUN passwords had a lower entry time and fewer mistakes than LU passwords, suggesting that the cost of adding the first additional character class (U) is higher than the cost of adding additional character classes (N).

Even more surprising is that passphrases (W) had the highest entry time and number of mistakes. One reason for this may be that because passphrases are longer than other passwords (due to their word-based construction), they have more opportunities for mistake propagation. The length of passphrases and the number of mistakes might also explain why entry time is higher for passphrases than for other compositions. We did consider normalizing time and mistakes for passphrases based on their length, but we decided against this as passphrases are meant to be their own type of PCP: trading length for memorability.

We also considered what would happen if we adjusted passwords of different compositions to have the same strength, measured in bits of entropy. Table 2 shows the predicted time to enter these same-strength passwords (based on the security of a 10-character LUNS password; approximately 65.5 bits of entropy). Even though they are longer, lowercase (L) passwords are still the fastest to enter across all input modalities, aligning with prior work that shows that user-created lowercase passwords can be more usable [46].

To further understand how entry time differed by character class, we again conducted a repeated-measures ANOVA test to model within-participant correlations, finding a statistically significant difference ($p < 0.001$) in time based on classification. Again, we conducted pairwise t-tests with a Bonferroni correction, finding statistically significant differences ($p < 0.001$) between all character classes for entry time except for between LU and LUN ($p = 0.296$). For mistakes, we use another LMEM to test for significance, but we do not find a statistically significant effect for the character class on mistakes.

4.3.2 Fine-Grained character analysis.

To analyze the impact of character classes on entry time, we employ an LMEM. This approach accounts for the nested structure of our data and the non-independence of observations by utilizing the participant as a random effect. Specifically, we incorporated random intercepts to capture individual baseline speed differences ($Var = 0.019$) and random slopes to capture individual differences in sensitivity to character class changes ($Var = 0.003$).

Through the LMEM, we found a strong, positive fixed effect for a character class change (Coefficient = 0.204 in log-units, $p < 0.001$) and a baseline keystroke entry time of 11.5 seconds (log-scale Intercept = 9.354). Since our dependent variable was log-transformed, this coefficient indicates that, on average, a character class change is associated with a 22.6% increase in the time required for the subsequent keystroke. This confirms that character class changes present a significant and consistent cost to entry speed, even when accounting for individual differences in typing habits.

Table 2: Predicted time, in seconds, to enter passwords of the same security strength on different input modalities, by composition

Input Modality	LUNS (10 chars)	LUN (11 chars)	LU (12 chars)	L (14 chars)	W (6 words)
Keyboard	3.35	2.41	2.38	2.29	4.91
Touchscreen	5.39	3.54	3.44	3.06	11.21
Gamepad	8.25	7.14	7.62	6.90	20.80
Air-mouse	7.80	8.22	8.47	7.29	23.19
Remote	9.44	9.37	10.28	8.11	23.88

Table 3: p-values for character class change types from LMEM analysis

Character Class Change Type	p-value	Time Penalty
L->S	<0.001	51.6%
N->S	<0.001	53.3%
S->L	<0.001	46.9%
S->N	<0.001	30.7%
S->U	<0.001	34.4%
U->S	<0.001	59.0%
L->N	0.65	1.2%
L->U	0.71	0.5%
N->L	0.24	-3.2%
N->U	0.12	4.6%
U->L	0.19	2.0%
U->N	0.22	3.6%

To further investigate this, we use another LMEM to split the character class changes by type (Table 3). This reveals a significant insight: the performance cost is almost exclusively due to switching to or from symbols, which incurred a 30% to 59% time penalty per keystroke (all $p < .001$). This may, at least in part, represent the cost of searching for symbols across different layouts, as symbols often have different placements across devices and layouts, unlike letters and numbers.

Key Takeaways: Our results show that while passphrases (W) are not very usably on non-desktop devices, passwords with just lowercase letters (L) are a strong candidate for usable generated passwords across all input modalities. In the case that L passwords are not possible, we recommend LUN passwords as a close alternative as even just removing symbols can significantly reduce entry time, and LUN passwords perform similarly to LU passwords while allowing for shorter passwords.

4.4 The Impact of the Interfaces

To answer our third research questions, we consider whether our selected interface treatments can help mitigate the challenges of entering generated passwords on different input modalities. Specifically, we were interested to see whether coloring the character classes would reduce the cognitive load of entering generated passwords on slower devices, and whether not deleting incorrect entries would make up for the physical constraints of the input modalities. However, a two-away ANOVA test did not find a statistically significant effect for the treatment on mistakes ($p = 0.072$), and while we did find one for time ($p < 0.05$), the effect size was negligible ($\eta^2 = 0.018$) (see Appendix D for graphs). This leads us to conclude that the usability of generated password entry is primarily driven by input device, so much so that the tested interface treatments were insufficient to overcome the physical constraints of the input modalities.

4.5 Perceived Usability and Rankings

Finally, we examine how users perceived the different input modalities. We asked participants to rank the input modalities they used (Figure 4) and to reflect on their experience with each input modality (Figure 5). As seen in the figures, participants were more favorable towards traditional physical keyboards and touchscreens than the other input modalities, likely due to the speed of authentication. Participants also preferred game controllers over remotes—even though both are limited to grid-based movements—and air mice—even though it might provide more freedom of movement than game controllers—likely due to the game controller's shortcuts which allow easier navigation of the grid.

When considering participants' open-ended feedback, we find 60 participants (60.1%) that mentioned the speed of authentication as a motivating reason for their rankings, and 22 participants (22.2%) that mentioned easily making a mistake or being able to easily correct a mistake as a reason for their rankings. These two comments were the most common reasons participants gave for their perceptions, which is consistent with prior work showing that participants favor speed of authentication over other considerations [43].

Related to this, 46 participants (46.5%) mentioned that they enjoyed the direct access they had to every key on keyboards and touchscreens, or that they disliked the lack of this direct access on the other input modalities.

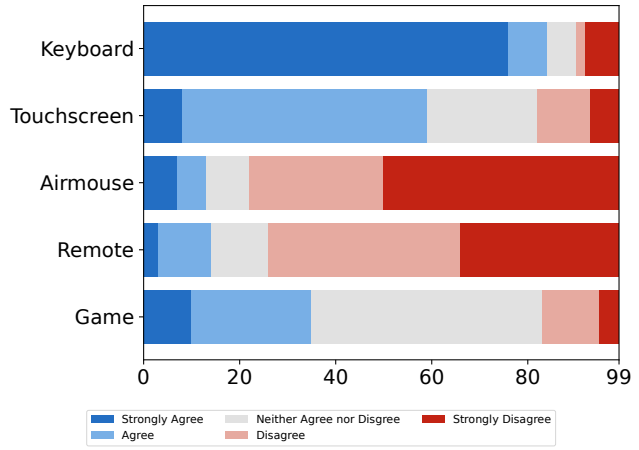


Figure 4: User rankings of the input modalities after using all input modalities from 1 (most favorite) to 5 (least favorite).

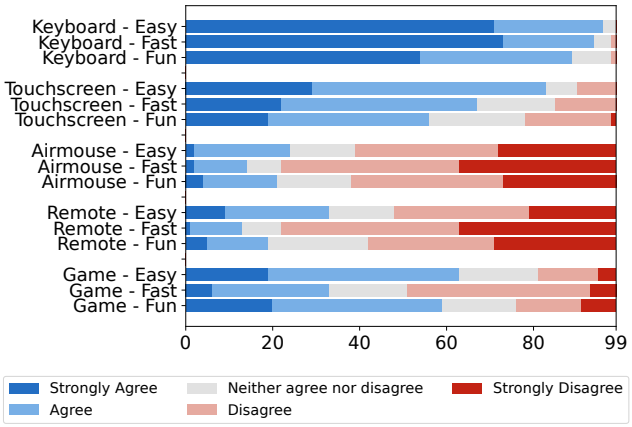


Figure 5: User sentiment towards the different input modalities based on reflection questions

> "*Clicking to go across the entire keyboard just to switch to [capital] letters and then go back up to enter one letter just to go back and switch it back to normal case. . . felt annoying and like a waste of time.*" (P98)

Overall, we find that keyboards and touchscreens are well-liked and fast, but also the most error-prone input modalities, but participants care less about mistakes on these input modalities:

> "*. . . I'm not afraid to type fast because making a mistake is not costly at all.*" (P97)

Notably, participants did not mention interface treatments (coloring or deletion) in their open-ended feedback, which reinforces our quantitative finding that the input modality outweighs the tested interface-level interventions in shaping the user experience.

Key Takeaways: In line with prior work on usability, we find that optimizing generated passwords for speed might be the most important usability factor, even more so than optimizing for mistakes. This also suggests that if we can improve speed, users may be more likely to adopt generated passwords.

5 Evaluating Password Generation Algorithms

A key benefit of our work is that it allows researchers to quickly test new password generation algorithms using machine learning methods before committing to a full user study, which can be time-consuming and expensive.

5.1 Existing Methods

To showcase this, we highlight the work of Greene et al. [18] as a case study, which suggested grouping character classes to avoid layout switching to decrease the time to enter a password. Additionally, we consider what would happen if we replaced special characters with numbers in generated passwords, as we found that switching to or from symbols incurs a large time penalty (Section 4.3.2).

In table 4, we report the predicted time to enter 10-character passwords of different types: a baseline random password with LUNS characters, the same baseline passwords but grouped according to Greene et al.'s approach, randomly generated LUN passwords, and randomly generated LUS passwords. The improvement column shows the percentage improvement in mean entry time compared to the baseline. We find that grouping the password by character class reduced the mean entry time by almost 25%, supporting Greene et al.'s findings. We also confirm that special characters have a significant cost, as LUS passwords are ~6 seconds slower than LUN passwords, making them slower than even the baseline LUNS passwords.

5.2 Device-Aware Password Generation

Leveraging our previous findings and input cost models (Section 4), we developed and tested two novel password generation algorithms focused on minimizing the time cost of moving from one character to the next. These generation algorithms consider the physical constraints of the input modalities through mapping the keyboard layout, distance between characters, and the natural cadences of human motor behavior when entering passwords. The full implementation of our algorithms is available with our research artifacts.

5.2.1 Physical-Constrained Foraging Algorithm

For the first algorithm, we modeled password generation as a navigation task within a metric space. When generation

Table 4: Comparison of entry time for 10-character passwords of different types.

Algorithm	Mean (Median) Time (s)	Improvement
Baseline (Random LUNS)	31.71 (34.90)	-
Grouped (Greene et al.)	23.82 (27.35)	24.88%
LUN	26.73 (29.45)	15.70%
LUS	32.75 (35.31)	-11.35%

starts, the generator randomly selects a number of characters per valid character class in the password, ensuring that at least one character from each class is included. Then for each generation step, we generate a candidate character for each character class left to be generated and score the candidates using a distance-based cost function, selecting the candidate with the lowest cost as the next character in the password. We then decrement the number of characters of that character class left to be generated and repeat the process until the password is fully generated.

To resolve the "local minima" problem, where a distance-minimizing generator might stagnate in a single zone of the keyboard, we apply a cumulative "heat" penalty to the recently visited keys and their surrounding neighborhood, effectively pushing the generator to explore "cooler," unvisited regions of the interface. This heat penalty is derived from the physical layout of the device and decays over time, allowing it to act as a dynamic addition to the cost function to encourage exploration while still optimizing for spatial locality (e.g., avoiding oscillating between distant keys). This approach provides us with wide key coverage while maintaining ergonomic clustering.

5.2.2 Kinetic Chain Generation

For the second algorithm, we took note of natural cadences in human motor behavior when entering passwords and modeled password generation as a rhythmic motor flow. In this algorithm, we use a Top-K Softmax Random Walk driven by an empirically derived fluency matrix that captures the natural transitions between keys scored based on a "Valley Cost Function" that penalizes transitions that deviate from the device's optimal stride length. This scoring function applies a quadratic penalty to the transitions that deviate from the device's optimal path length, effectively penalizing both low-entropy neighboring key transitions and high-effort cross-keyboard jumps.

To prevent the generator from getting trapped in physical dead ends (e.g., corners of the keyboard), which is an issue with a purely greedy algorithm, we implemented a state-aware constraint machine with short-term memory. We use a blacklist of size $N = 3$ to prevent immediate backtracking and oscillation (e.g., A→B→A), forcing the generation path to "snake" across the keyboard in a more progressive manner. As the password grows, the generator monitors the remaining password length and unsatisfied character requirements, transitioning from a free flow state (optimizing for speed), to a nudging state (applying soft cost bonuses to missing character types), and finally to a decisive mode (enforcing hard constraints when remaining slots equal missing types) to ensure that the generated password satisfies the required composition. The resulting password contains natural key sequences and transitions that users are able to enter with smooth motor frequency but that retain sufficiently high entropy.

5.2.3 Algorithm Security and Performance Evaluation

Maintaining resistance to guessing attacks is also a key part of a new password generation algorithm. We compare the security and expected entry time of our algorithms to the same baseline as before.

Entropy Using a Monte Carlo simulation, we measured the theoretical minimum entropy (H_{min}) of 100,000 generated passwords from each algorithm. For the baseline random passwords, we use the formula $H_{min} = \log_2(N^L)$, where N is the size of the character set and L is the length of the password, to calculate the theoretical minimum entropy. Because our device-aware algorithms induce non-uniform probability distributions, we evaluated their practical guessing difficulty using Bonneau's α-work-factor metrics [6], which directly quantify the $\log_2$ of the number of guesses an optimal attacker must make to compromise a given fraction (α) of the generated passwords. Using a Monte Carlo simulation with 100,000 iterations, we map the underlying probability distributions at each generative step to accurately evaluate the guessing difficulty of our algorithms.

Table 5 shows that while our algorithms inherently trade a portion of maximum theoretical entropy to achieve their significant physical usability gains, they remain highly secure in practice. For example, compromising the median generated password ($W_{0.50}$) requires over 2^{44} guesses. Furthermore, even the most vulnerable 1% of the distribution ($W_{0.01}$) requires nearly 275 billion guesses ($> 2^{38}$), which presents a significant computational barrier, and would be more than sufficient for online guessing attacks due to guess throttling. This proof-of-concept demonstrates that device-aware generation can reduce physical entry costs without compromising practical security.

Table 5: Comparison of security (α-work-factor guessing difficulty and Min-Entropy, expressed as $\log_2$ values) and entry time for 10-character passwords

Algorithm	Min-Entropy	$W_{0.01}$	$W_{0.10}$	$W_{0.50}$	Mean (Median) Time (s)	Improvement
Baseline (Random LUNS)	65.24	58.60	61.92	64.24	31.71 (34.90)	-
Physics-Constrained	39.01	40.21	43.31	47.97	22.97 (27.42)	27.56%
Kinetic Chain	43.01	38.53	43.02	52.62	21.79 (26.03)	31.28%

Predicted Entry Time We report on the average entry time for 10,000 10-character passwords on a random mix of all five devices. As shown in Table 5, both of the new algorithms significantly reduce the predicted entry time compared to the baseline, with the kinetic chain algorithm showing a greater improvement (31.28%) than the physics-constrained foraging algorithm (27.55%) over the baseline. Notably, both new algorithms perform better than Greene et al.'s grouping algorithm, which had a 24.88% improvement over the baseline, suggesting that device-aware password generation can better capture the nuances of entry on various input modalities.

6 Discussion

6.1 A Password Generation Testbed

A primary contribution of our work is the release of our fine-grained dataset and the accompanying XGBoost models. By leveraging our predictive models, research can quickly prototype and evaluate new password generation algorithms without the immediate need for resource-intensive user studies.

Future work. While our work uses the XGBoost models to evaluate the performance of whole passwords, the granularity of our dataset would allow for a more detailed analysis of password performance if a character-by-character model were trained instead. Such a model could identify micro-patterns in password entry, enabling more nuanced optimization of generation algorithm design choices.

6.2 The Hierarchy of Password Generation Improvements

Our results establish a hierarchy of the relative impact of different design choices for improving generated password entry. At the bottom of this hierarchy are the tested interface treatments, specifically coloring the password character classes or deleting wrong entries, which had a negligible impact on entry time and error rates. This null result is important, as it suggests that the difficulty of entering a generated password is not a perceptual issue, but rather a structural issue related to the physical constraints of the input modality. However, it is worth noting that there may be other interface treatments not explored in this paper that future work could investigate, which may yield a more significant impact on generated password entry.

In contrast, the password composition had a significant impact on entry time and error rates. For example, passphrases, which are often suggested as a usable alternative to passwords, were the slowest composition on all input modalities except for physical keyboards. This is compounded by the fact that we intentionally limit our passphrases to three words, sacrificing security in hopes of improving usability (passphrases had $\approx$ 39 bits of entropy, LUNS passwords had $\approx$ 65). Furthermore, prior work suggests that changing the number or length of the words in a passphrase would not have impacted these results, suggesting that system-assigned passphrases are not a good option for generated passwords, aligning with some prior work [45, 52]. Conversely, long lowercase-only passwords were the fastest passwords to enter on all devices, even when adjusting for length to increase their security, making them a promising avenue to explore if they are not hindered by password composition policies (PCPs).

Finally, the input device had the largest impact on entry time and error rates. We believe future work should focus on understanding and improving generated password entry on various input modalities instead of treating generated password entry as a one-size-fits-all problem.

A clear example of this is that LUN passwords were relatively easy to enter on our devices due to having numbers accessible on the same layout as lowercase letters. Without this, LUN passwords would likely have been slower due to requiring a layout change.

Since designers cannot control what devices users authenticate on, and the tested interface treatments alone are insufficient, the most promising path forward is better password generation algorithms.

6.3 A New Generation of Password Generators

Our work culminates in the design and evaluation of two novel device-aware password generation algorithms, analytically verifying that they can significantly reduce the expected entry time of generated passwords. By calibrating our generators with our dataset, we steered the algorithms toward more ergonomic passwords that better leverage individual device characteristics, leading to passwords that

were up to 31.28% faster to enter than the baseline random passwords.

From these results we suggest that the era of purely randomized password generation should give way to device-aware approaches designed with usability in mind. Password managers should also be upgraded to leverage different algorithms based on the devices in use, meaning every algorithm does not have to work well on every device. Importantly, we have demonstrated that these algorithms do not need to sacrifice security to achieve better usability, unlike how users approach user-created passwords [11, 16, 35, 50]. Ultimately, password managers may even abstract password composition and instead ask users to select a desired level of security, then the password manager can choose the best device-aware algorithm to generate a password that meets the desired security level while being optimized for usability.

An important consideration in this line of work is the practical usage of device-aware passwords in a multi-device ecosystem, as users often authenticate into the same account across multiple devices. We suggest password managers prioritize lowercase-heavy compositions, which our results show are the fastest across all devices, and optimize character sequencing for the most constrained device on which that password will be needed. In practice, the most constrained device is likely to be a TV or console, where the user is restricted to a directional pad on a grid layout. However, this again emphasizes the need for empirical user studies to verify the real-world benefits of device-aware password generation and to understand how optimizing for one device impacts the usability of generated passwords on other devices. Future work should determine whether and how much optimizing for the most constrained device impacts usability on others.

Future work. Our algorithms are a proof of concept, rather than a proposed standard, to show that device-aware password generation should be the focus of future work in this space. For example, even though passphrases performed poorly in our study, we suggest that a new password generation algorithm could leverage chunking strategies to create passwords that are easier to remember in the short term while still taking advantage of device-level characteristics to increase usability. While our proof-of-concept algorithms demonstrate the theoretical benefits of device-aware password generation, there is still a critical need for future work to empirically verify these benefits through formal user studies. Longitudinal studies will also be needed to evaluate the long-term effectiveness of these algorithms and to see if they can promote the adoption of generated passwords.

We also saw that participants appreciated the speed of entry on keyboards and touchscreens, even though the error rate was high, suggesting that communicating these design trade-offs to users could promote generated password adoption. Finally, for service providers, we suggest they concentrate on improving PCPs, as prior work has shown that these have an impact on usability (e.g., [36, 47]) and they sometimes disallow the most usable compositions, such as long lowercase passwords.

7 Limitations

We chose a laboratory study despite its scale limitations. For example, trying to collect this data on-device would be infeasible for some devices (e.g., on game consoles), and a remote user study would have made it difficult to ensure ecological validity (especially that participants were using the correct devices).

Our participants are skewed heavily towards younger adults (18-31), which limits the generalizability of our results to older adults. This also suggests our results are an upper bound—if younger, more familiar participants still struggled, older or less experienced users likely would too. Furthermore, our core findings relate to the physical kinematic costs of entering generated passwords, such as layout switching, which reflect fundamental structural problems with these input modalities that are not likely to diminish even with increased user familiarity.

Our study does not include virtual reality (VR) systems, as password-based authentication on VR remains nascent [24] and a variety of other authentication schemes available to them [23, 27, 33, 55]. Our methodology is also incompatible with VR, as it requires referencing a password on a separate screen, which is not the current paradigm for VR entry.

Finally, budget and time constraints limited the password compositions we tested, and we did not control for device experience, which may have introduced small learning effects.

8 Conclusion

Passwords remain a critical backbone of authentication due to their default status in many systems and their role as the universal fallback. We improve generated passwords by contributing the first fine-grained dataset of how users enter generated passwords on non-desktop devices, finding that the physical constraints of the input modalities are the primary drivers of entry time and error rates. We then leverage these insights to design and evaluate novel device-aware password generation algorithms as proof-of-concept models, finding that this approach can significantly reduce the predicted entry time of generated passwords. Consequently, we advocate for more work in this vein of research, as device-aware password generation is a promising path toward a future where strong, generated passwords are usable across all devices, even without autofill.

Acknowledgments

This material is based upon work supported by the National Science Foundation under Grant No. 2226404. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

References

[1] Anne Adams and Martina Angela Sasse. Users are not the enemy. *Communications of the ACM*, 42(12):40–46, 1999.

[2] Fahad Alodhyani, George Theodorakopoulos, and Philipp Reinecke. Password managers—it's all about trust and transparency. *Future Internet*, 12(11):189, 2020.

[3] Aurora Barrero, David Melendi, Xabiel G Pañeda, Roberto García, and Laura Pozueco. Evaluation of Text Entry Methods for Interactive Digital Television Applications with Devices Alternative to Conventional Remote Controls. *International Journal of Human–Computer Interaction*, 32(10):765–776, 2016.

[4] Bitwarden. wordlist.ts. https://github.com/bitwarden/clients/blob/main/libs/common/src/platform/misc/wordlist.ts, 2023. GitHub repository file.

[5] Jan Bobeth, Johann Schrammel, Stephanie Deutsch, Michael Klein, Mario Drobics, Christina Hochleitner, and Manfred Tscheligi. Tablet, Gestures, Remote Control? Influence of Age on Performance and User Experience with iTV Applications. In *Proceedings of the ACM International Conference on Interactive Experiences for TV and Online Video*, pages 139–146, 2014.

[6] Joseph Bonneau. The science of guessing: analyzing an anonymized corpus of 70 million passwords. In *2012 IEEE symposium on security and privacy*, pages 538–552. IEEE, 2012.

[7] Joseph Bonneau, Cormac Herley, Paul C. van Oorschot, and Frank Stajano. The quest to replace passwords: a framework for comparative evaluation of web authentication schemes. In *Proceedings of the 33rd IEEE Symposium on Security and Privacy*. IEEE, 2012.

[8] Rahul Chatterjee, Anish Athayle, Devdatta Akhawe, Ari Juels, and Thomas Ristenpart. password typos and how to correct them securely. In *2016 IEEE Symposium on Security and Privacy (SP)*, pages 799–818. IEEE, 2016.

[9] José Coelho, Carlos Duarte, Pradipta Biswas, and Patrick Langdon. Developing Accessible TV Applications. In *The proceedings of the 13th international ACM SIGACCESS conference on Computers and accessibility*, pages 131–138, 2011.

[10] Anupam Das, Joseph Bonneau, Matthew Caesar, Nikita Borisov, and XiaoFeng Wang. The Tangled Web of Password Reuse. In *NDSS*, volume 14, pages 23–26, 2014.

[11] Matteo Dell'Amico, Pietro Michiardi, and Yves Roudier. Password strength: An empirical analysis. In *2010 Proceedings IEEE INFOCOM*, pages 1–9. IEEE, 2010.

[12] Vivek Dhakal, Anna Maria Feit, Per Ola Kristensson, and Antti Oulasvirta. Observations on Typing from 136 Million Keystrokes. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, pages 1–12, 2018.

[13] Michael Fagan, Yusuf Albayram, Mohammad Maifi Hasan Khan, and Ross Buck. An investigation into users' considerations towards using password managers. *Human-centric computing and information sciences*, 7(1):1–20, 2017.

[14] Anna Maria Feit, Daryl Weir, and Antti Oulasvirta. How we type: Movement strategies and performance in everyday typing. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, pages 4262–4273, 2016.

[15] Paul M Fitts. The information capacity of the human motor system in controlling the amplitude of movement. *Journal of Experimental Psychology*, 47(6):381, 1954.

[16] Dinei Florencio and Cormac Herley. A large-scale study of web password habits. In *Proceedings of the 16th International Conference on World Wide Web*, pages 657–666. ACM Press, 2007.

[17] Kristen K Greene. Effects of Password Permutation on Subjective Usability Across Platforms. In *International Conference on Human Aspects of Information Security, Privacy, and Trust*, pages 59–70. Springer, 2015.

[18] Kristen K. Greene, Joshua M. Franklin, and John M. Kelsey. Tap on, tap off: Onscreen keyboards and mobile password entry. In *ShmooCon XI*, 2015.

[19] Kristen K Greene, Melissa A Gallagher, Brian C Stanton, and Paul Y Lee. I can't type that! p@ w0rd entry on mobile devices. In *Human Aspects of Information Security, Privacy, and Trust: Second International Conference, HAS 2014, Held as Part of HCI International 2014, Heraklion, Crete, Greece, June 22-27, 2014. Proceedings 2*, pages 160–171. Springer, 2014.

[20] Cormac Herley and Paul Van Oorschot. A Research Agenda Acknowledging the Persistence of Passwords. *IEEE Security & Privacy*, 10(1):28–36, 2012.

[21] William E Hick. On the rate of gain of information. *Quarterly Journal of Experimental Psychology*, 4(1):11–26, 1952.

[22] Markus Jakobsson and Ruj Akavipat. Rethinking Passwords to Adapt to Constrained Keyboards. *Proc. IEEE MoST*, pages 1–11, 2012.

[23] John M. Jones, Reyhan Duezguen, Peter Mayer, Melanie Volkamer, and Sanchari Das. A Literature Review on Virtual Reality Authentication. In *Human Aspects of Information Security and Assurance*, volume 613 of *IFIP Advances in Information and Communication Technology*, pages 189–198. Springer, 2021.

[24] Emiram Kablo, Yorick Last, Patricia Arias Cabarcos, and Melanie Volkamer. The (Un)suitability of Passwords and Password Managers in Virtual Reality. *arXiv preprint arXiv:2503.18550*, 2025.

[25] Sabrina Klivan, Sandra Höltervennhoff, Nicolas Huaman, Yasemin Acar, and Sascha Fahl. " would you give the same priority to the bank and a game? i do {not!}" exploring credential management strategies and obstacles during password manager setup. In *Nineteenth Symposium on Usable Privacy and Security (SOUPS 2023)*, pages 171–190, 2023.

[26] Thomas Költringer, Poika Isokoski, and Thomas Grechenig. Twostick: Writing with a game controller. In *Proceedings of Graphics Interface 2007*, pages 103–110, 2007.

[27] Pınar Kürtünlüoğlu, Beste Akdik, and Enis Karaarslan. Security of Virtual Reality Authentication Methods in Metaverse: An Overview. *arXiv preprint arXiv:2209.06447*, 2022.

[28] Seungyon Lee and Shumin Zhai. The Performance of Touch Screen Soft Buttons. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*, pages 309–318, 2009.

[29] Li Li and Ren Yi. MEMS-based digital TV interaction innovation research. In *2010 IEEE 11th International Conference on Computer-Aided Industrial Design & Conceptual Design 1*, volume 1, pages 665–667. IEEE, 2010.

[30] Gordon D Logan, Jana E Ulrich, and Dakota RB Lindsey. Different (key)strokes for Different Folks: How Standard and Nonstandard Typists Balance Fitts' Law and Hick's Law. *Journal of Experimental Psychology: Human Perception and Performance*, 42(12):2084–2102, 2016.

[31] Sanam Ghorbani Lyastani, Michael Schilling, Sascha Fahl, Michael Backes, and Sven Bugiel. Better managed than memorized? studying the impact of managers on password strength and reuse. In *Proceedings of the 28th USENIX Security Symposium*. USENIX, 2018.

[32] Sanam Ghorbani Lyastani, Michael Schilling, Michaela Neumayr, Michael Backes, and Sven Bugiel. Is fido2 the kingslayer of user authentication? a comparative usability study of fido2 passwordless authentication. In *2020 IEEE Symposium on Security and Privacy (SP)*, pages 268–285. IEEE, 2020.

[33] Florian Mathis, John H. Williamson, Kami Vaniea, and Mohamed Khamis. Fast and secure authentication in virtual reality using coordinated 3d manipulation and pointing. *ACM Trans. Comput.-Hum. Interact.*, 28(1), January 2021.

[34] Peter Mayer, Nina Gerber, Benjamin Reinheimer, Philipp Rack, Kristoffer Braun, and Melanie Volkamer. I (don't) see what you typed there! shoulder-surfing resistant password entry on gamepads. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, pages 1–12, 2019.

[35] Michelle L Mazurek, Saranga Komanduri, Timothy Vidas, Lujo Bauer, Nicolas Christin, Lorrie Faith Cranor, Patrick Gage Kelley, Richard Shay, and Blase Ur. Measuring password guessability for an entire university. In *Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Security*, pages 173–186. Association for Computing Machinery, 2013.

[36] William Melicher, Darya Kurilova, Sean M Segreti, Pranshu Kalvani, Richard Shay, Blase Ur, Lujo Bauer, Nicolas Christin, Lorrie Faith Cranor, and Michelle L Mazurek. Usability and Security of Text Passwords on Mobile Devices. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, pages 527–539. Association for Computing Machinery, 2016.

[37] Sean Oesch, Scott Ruoti, James Simmons, and Anuj Gautam. "it basically started using me:" An observational study of password manager usage. In *Proceedings of the 40th ACM CHI Conference on Human Factors in Computing Systems*. ACM, 2022.

[38] Sarah Pearman, Jeremy Thomas, Pardis Emami Naeini, Hana Habib, Lujo Bauer, Nicolas Christin, Lorrie Faith Cranor, Serge Egelman, and Alain Forget. Let's go in for a closer look: Observing passwords in their natural habitat. In *Proceedings of the 24th ACM SIGSAC*

Conference on Computer and Communications Security. ACM, 2017.

[39] Sarah Pearman, Shikun Aerin Zhang, Lujo Bauer, Nicolas Christin, and Lorrie Faith Cranor. Why people don't use password managers effectively. In *Proceedings of the 15th Symposium On Usable Privacy and Security*. USENIX, 2019.

[40] Catlin Pidel and Stephan Neuhaus. Breaking: password entry is fine. In *HCI for Cybersecurity, Privacy and Trust: First International Conference, HCI-CPT 2019, Held as Part of the 21st HCI International Conference, HCII 2019, Orlando, FL, USA, July 26–31, 2019, Proceedings 21*, pages 67–80. Springer, 2019.

[41] Hirak Ray, Flynn Wolf, Ravi Kuber, and Adam J. Aviv. Why Older Adults (Don't) Use Password Managers. In *30th USENIX Security Symposium (USENIX Security 21)*, pages 73–90. USENIX Association, 2021.

[42] Samik Raychaudhuri. Introduction to Monte Carlo simulation. In *2008 Winter Simulation Conference*, pages 91–100. IEEE, 2008.

[43] John Sadik and Scott Ruoti. A Large-Scale Survey of Password Entry Practices on Non-Desktop Devices. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, 9(3), September 2025.

[44] Sunyoung Seiler-Hwang, Patricia Arias-Cabarcos, Andrés Marín, Florina Almenares, Daniel Díaz-Sánchez, and Christian Becker. "I don't see why i would ever want to use it:" Analyzing the usability of popular smartphone password managers. In *Proceedings of the 26th ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2019.

[45] Richard Shay, Patrick Gage Kelley, Saranga Komanduri, Michelle L Mazurek, Blase Ur, Timothy Vidas, Lujo Bauer, Nicolas Christin, and Lorrie Faith Cranor. Correct horse battery staple: Exploring the usability of system-assigned passphrases. In *Proceedings of the eighth symposium on usable privacy and security*, pages 1–20, 2012.

[46] Richard Shay, Saranga Komanduri, Adam L Durity, Phillip Huh, Michelle L Mazurek, Sean M Segreti, Blase Ur, Lujo Bauer, Nicolas Christin, and Lorrie Faith Cranor. Designing Password Policies for Strength and Usability. *ACM Trans. Inf. Syst. Secur.*, 18(4):1–34, 2016.

[47] Richard Shay, Saranga Komanduri, Patrick Gage Kelley, Pedro Giovanni Leon, Michelle L Mazurek, Lujo Bauer, Nicolas Christin, and Lorrie Faith Cranor. Encountering stronger password requirements: user attitudes and behaviors. In *Proceedings of the sixth symposium on usable privacy and security*, pages 1–20, 2010.

[48] James Simmons, Oumar Diallo, Sean Oesch, and Scott Ruoti. Systematization of Password ManagerUse Cases and Design Paradigms. In *Proceedings of the 37th Annual Computer Security Applications Conference*, pages 528–540. Association for Computing Machinery, 2021.

[49] Heloisa Simon, Eros Comunello, and Aldo Von Wangenheim. Enrichment of Interactive Digital TV using Second Screen. *International Journal of Computer Applications*, 64(22):58–64, 2013.

[50] Wayne C Summers and Edward Bosworth. Password policy: the good, the bad, and the ugly. In *Proceedings of the winter international synposium on Information and communication technologies*, pages 1–6, 2004.

[51] Andrew D Wilson and Maneesh Agrawala. Text entry using a dual joystick game controller. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 475–478. Association for Computing Machinery, 2006.

[52] Xiaoyuan Wu, Collins W Munyendo, Eddie Cosic, Genevieve A Flynn, Olivia Legault, and Adam J Aviv. User Perceptions of Five-Word Passwords. In *Proceedings of the 38th Annual Computer Security Applications Conference*, pages 605–618, 2022.

[53] Leon Würsching, Florentin Putz, Steffen Haesler, and Matthias Hollick. Fido2 the rescue? platform vs. roaming authentication on smartphones. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, pages 1–16, 2023.

[54] Yulong Yang, Janne Lindqvist, and Antti Oulasvirta. Text Entry Method Affects Password Security. In *The LASER Workshop: Learning from Authoritative Security Experiment Results (LASER 2014)*, 2014.

[55] Zhen Yu, Hai-Ning Liang, Charles Fleming, and Ka Lok Man. An exploration of usable authentication mechanisms for virtual reality systems. In *2016 IEEE Asia Pacific Conference on Circuits and Systems (APCCAS)*, pages 458–460. IEEE, 2016.

[56] Xin Zhang, Xiaohan Zhang, Huijun Zhou, and Bo Zhao. Anchors of trust: A usability study on user awareness, consent, and control in cross-device authentication. In *NDSS*, 2026.

A Survey

After the consent form, the participants were shown the following pages. Pages 3 and 4 were shown to the participant five times in order to collect data on five different password entry tools. Each time, one of the remaining input modalities was randomly chosen and substituted for *DEVICE_TYPE*.

A.1 Page 1

What is your gender?

- Male
- Female
- Prefer not to answer

How old are you?
`[Text Entry]`

What is the highest level of school you have completed or the highest degree you have received?

- Less than high school
- High school
- Some college
- 2 year degree
- 4 year degree
- Professional degree
- Doctorate
- Prefer not to answer

Are you colorblind?

- No
- Yes, I have red-green color blindness
- Yes, I have blue-yellow color blindness

A.2 Page 2

You will now use five different tools, one after the other to enter a series of passwords. The study coordinator will help you through this process.

Study coordinator, please select the duration of the study. 5 is the 15-minutes study ($10). 10 is the 30-minutes study ($20).

- 5
- 10

A.3 Page 3

You will now use *DEVICE_TYPE* to enter several passwords. After you are done entering the passwords, return to this page and click "Next Page" to provide some feedback about your experience.

Please call the study coordinator over now.

Study coordinator, please setup the user's device using these values:
Link: <URL to website>
Color: <Color treatment 1 or 0>
Delete: <deletion treatment 1 or 0>
ID: <Participant unique ID>

Scan the QR code for touchscreen:
<QR code>

A.4 Page 4

Please rate how much you agree with the following statements.
(Strongly Disagree | Disagree | Neutral | Agree | Strongly Agree)

- It was easy for me to enter a password using *DEVICE_TYPE*
- Entering a password with *DEVICE_TYPE* was fast
- I enjoyed entering a password with *DEVICE_TYPE*

A.5 Page 5

Please rank each entry method based on which you liked best. **1 is your most favorite**, 5 is you least favorite.
(1 | 2 | 3 | 4 | 5)

- Keyboard
- Touchscreen
- Air Mouse
- Remote Controller
- Game Controller

Please describe why you ranked the entry methods as you did in the previous question.
`[Text Entry]`

When thinking about your **favorite** entry method, what are some of the things you **liked** the most?
`[Text Entry]`

When thinking about your **least favorite** entry method, what are some of the things you **disliked** the most?
[Text Entry]

Do you have any other comments related to your experience today that you would like to share?
[Text Entry]

B Device Interface Screenshots

NOTE: The remote and air mouse used the same interface, so the screenshots are the same.

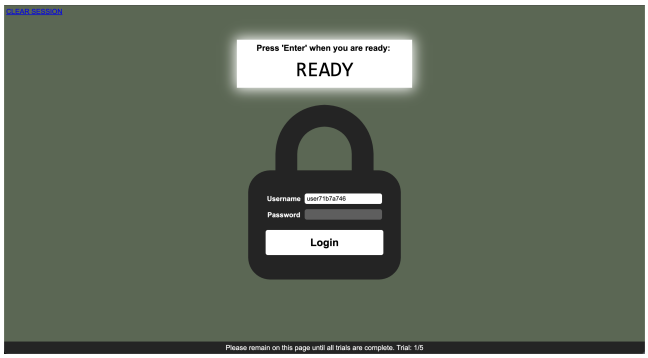


Figure 6: Ready state of a keyboard password trial



Figure 7: An example of a keyboard password without coloring



Figure 8: Ready state of a touchscreen password trial



Figure 9: An example of a touchscreen password with coloring

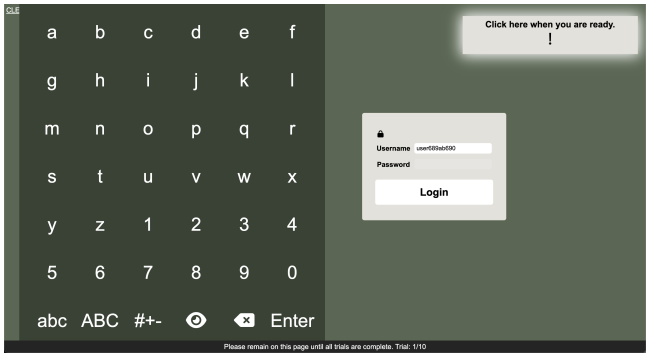


Figure 10: Ready state of a remote/air mouse password trial

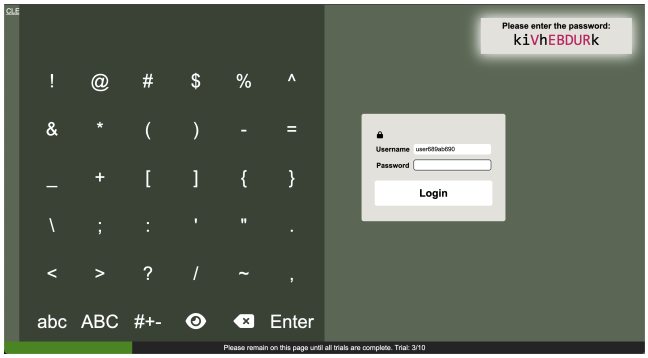


Figure 11: An example of a remote/air mouse password with coloring

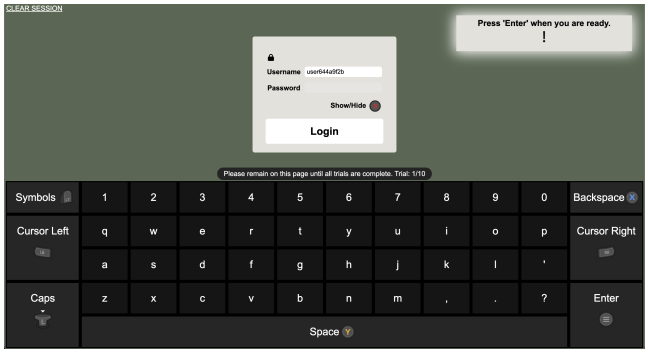


Figure 12: Ready state of a game controller password trial

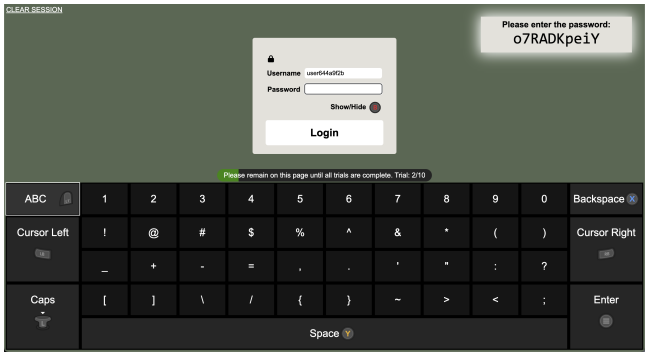


Figure 13: An example of a game controller password without coloring

C Study Input Modality Images



Figure 14: Keyboard. Image source: https://m.media-amazon.com/images/I/71NmNRqG3WL._AC_SL1500_.jpg



Figure 15: Samsung Tab and Apple iPad. Image sources: https://www.zdnet.com/article/samsung-galaxy-tab-s8-plus-review/ and https://www.macrumors.com/2018/11/07/ipad-pro-2018-first-impressions/



Figure 16: Air mouse and remote control. Image source: https://www.pepper-jobs.com/products/w10-gyro-smart-remote



Figure 17: Xbox controller. Image source: https://www.gamestop.com/consoles-hardware/products/microsoft-xbox-series-x-controller/11108369.html

D Extended Results

D.1 Interaction Effect Analysis

Here we present our analysis of the statistical interaction effects between our primary independent variables (Input Modality, Password Composition, and Interface Treatments) using Linear Mixed-Effects Models (LMEM) with the participant as a random effect to examine these interactions. We found a statistically significant interaction effect between Input Modality and Password Composition ($p < 0.001$), finding that the time penalty for LUNS passwords and passphrases (W) was highly correlated with the input modality used. This was especially true for passphrases, which were disproportionately penalized on grid-based devices but had a significantly smaller performance penalty on physical keyboards ($p < 0.001$). Conversely, we failed to find a statistically significant interaction effect between input modality and the treatment assigned to the participant ($p > 0.05$ for all Device $\times$ Color, Device $\times$ Delete, and Device $\times$ Color $\times$ Delete terms).

D.2 Treatment Graphs

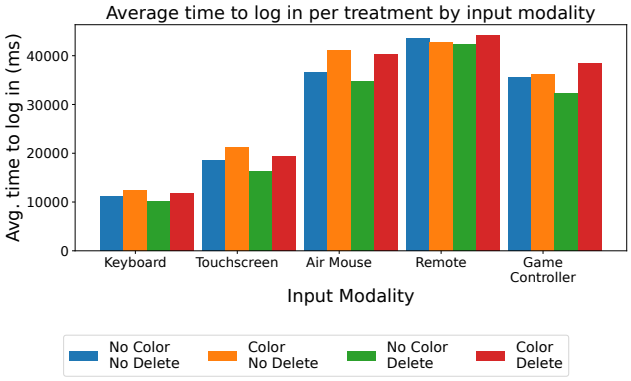


Figure 18: Time to enter a password per treatment grouped by input modality

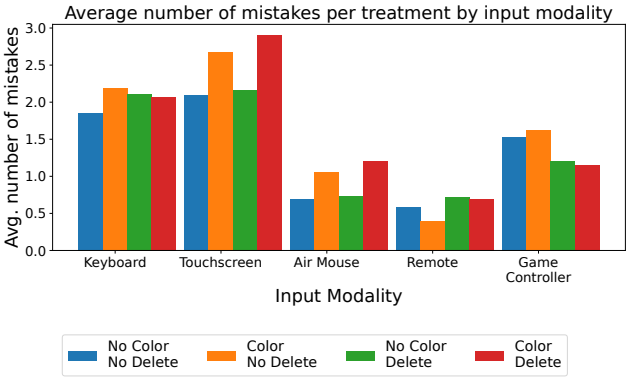


Figure 19: Average number of mistakes per treatment grouped by input modality