Password Entry Errors: Memory or Motor?

Kristen K. Greene (kristen.greene@nist.gov)
National Institute of Standards and Technology, 100 Bureau Drive, MS 8940
Gaithersburg, MD 20899-8940 USA

Franklin P. Tamborello, II (franklin.tamborello.ctr@nrl.navy.mil)
National Research Council Postdoctoral Research Associate, 4555 Overlook Ave SW
Washington, DC 20375 USA

Abstract

As we increasingly rely upon our computer information systems to store and operate on sensitive information, the methods we use to authenticate user identity also become more important. One of the most important such methods is the password. However, passwords that provide better security also tend to be more difficult to remember. They also tend to be difficult to type, and typing errors can have negative consequences such as being locked out of a critical information system. We present a computational cognitive model of password rehearsal and a typing extension to the ACT-R cognitive architecture intended to study human-computer interaction issues in the usable security domain.

Keywords: Human-Computer Interaction; Learning; Memory; Typing; Human Error

Introduction

As cyber attacks on user-chosen passwords abound, there are large-scale, long-term research efforts underway (e.g., the National Strategy for Trusted Identities in Cyberspace, 2011) to ultimately replace passwords as an authentication mechanism. In the near-term however, password research is still important, as better understanding of the cognitive and perceptual motor components of creating, rehearsing, recalling, and typing passwords is necessary to help inform password policies and password requirements. Furthermore, even as alternative authentication mechanisms become more prevalent (e.g., biometrics), there will undeniably be legacy systems reliant upon passwords for quite some time.

While the importance and impact of password research is clear, it can be difficult to obtain real-world password data due to security and privacy concerns, or in the case of leaked password datasets, due to ethical concerns. It would be prohibitively expensive and time-consuming to collect laboratory data from large numbers of participants across relevant password requirements, specifically different combinations of password rules for length and complexity. There are also issues of experimental control versus external validity; in researching password requirements, does one assign passwords or have participants generate their own?

As in other domains where access to human data can be challenging, behavioral data from existing password experiments can be supplemented with predictive models of human performance. Unfortunately, most password studies do not collect sufficiently detailed data to assess model validity and plausibility. The cybersecurity and modeling fields could both benefit from computational cognitive models across a variety of password-related tasks: initial learning and rehearsal strategies; recall and entry of well-memorized passwords; and cross-platform (i.e., desktop versus mobile) password typing. The current work focuses on support for modeling desktop password rehearsal and typing, specifically for complex, system-generated passwords found in higher-security enterprise environments.

Transcription Typing Versus Password Typing

There is certainly a large and longstanding body of expert typing and transcription typing literature (e.g., Coover, 1923, Gentner, 1981, Salthouse, 1986), including examination of a variety of factors such as age and skill (e.g., Salthouse, 1984). However, there are several important distinctions between general transcription typing and password typing.

In the higher-security enterprise environments for which the current work is intended, passwords are quite different from words—in fact, most password policies explicitly prohibit the sole use of words, as dictionary attacks on passwords are so successful, dating back to the late 1970s (Morris & Thompson, 1979). Higher-entropy passwords differ quite significantly from the words commonly found in most traditional transcription typing experiments. “Better” passwords are supposed to be as random as possible in order to make guessing them more difficult; they should not follow orthographic rules as do regular words. Therefore, when typing complex passwords, we cannot leverage many of the benefits of natural language. Beyond simple inclusion of lowercase and uppercase letters, most higher-entropy passwords also include numbers and special characters, making it difficult or impossible to leverage error correction techniques during password typing. Furthermore, password text is usually masked, whereas normal text is not. These factors may contribute to changes in strategy for carefully typing passwords in comparison to normal transcription typing.

In addition to behavioral studies of transcription typing (see Salthouse, 1986 for a good review), there have also been cognitive models of the task. By far the most comprehensive and well-known computational cognitive model of transcription typing is Bonnie John’s TYPIST (John, 1996). John’s TYPIST quantified 19 of the 29 previously reviewed Salthouse (1986) phenomena as well as two additional phenomena. While TYPIST quantified transcription typing along the time dimension with scheduling charts, it did not simulate decreased performance

variability with higher typing skill, nor brain areas' activation patterns, as have more recent queuing network models (e.g., Wu & Lui, 2004). Regardless, to the best of the authors' knowledge, there does not currently exist an ACT-R model of rehearsal and typing for complex, system-generated passwords on a standard desktop QWERTY keyboard. The current work is a necessary first step to begin addressing this gap.

Typing System-Generated Passwords

Prior Work

The current work was motivated by a desire to use cognitive modeling as an error exploration technique to supplement prior usable security research. The primary goal was to better understand the underlying cause of errors reported in a recent study of complex password entry on desktop computers (Stanton & Greene, 2014). This section describes relevant methodology and results of interest from said study.

Method In the Stanton and Greene (2014) study, participants were given ten system-generated[1] passwords in a random order. Passwords ranged in length from six to 14 characters (see Table 1).

Table 1: Stimuli (Stanton & Greene, 2014).

Password	Length
5c2'Qe	6
3.bH1o	6
m3)61fHw	8
ua7t?C2#	8
p4d46*3TxY	10
q80<U/C2mv	10
d51)u4;X3wrf	12
6n04%Ei'Hm3V	12
m#o)fp^2aRf207	14
4i_55fQ$2Mnh30	14

Each password had to contain at least one uppercase letter, one lowercase letter, one number, and one special character. Passwords could not end with an exclamation mark, nor could passwords begin with a capital letter. Note the variety of symbols in the preceding stimuli, many of which are not supported for typing in standard ACT-R.

Two groups of participants were tested in the Stanton and Greene (2104) study. One group was from the Washington, DC (WDC) metropolitan area in the United States, and the other was from the University College London (UCL) in the United Kingdom. This sampling distinction is important, as results differed somewhat by participant group. The authors proposed that this might be due to differences in age and/or typing ability between the two groups, as the WDC group was older than the younger UCL group, which was composed of mainly undergraduates.

In the Stanton and Greene (2014) study, participants received one password at a time, and for each password, had to complete a series of three tasks: practice, verification, and entry. During practice, the password was visible, and participants could practice typing the password in a large text field. Participants could practice typing the password as many or as few times as they wished. There was no feedback given during the practice task, and typed text was visible (i.e., not masked as in a regular password field).

During verification, the password was not visible, and participants had to enter the memorized password correctly in order to move onto the entry screen. Typed text was visible (i.e., not masked) during verification. If participants failed the verification task, they could continue to attempt verification, or choose to return to the preceding practice screen to practice the password again. Regardless, after participants completed the verification task, they moved onto the entry task.

During entry, participants had to enter the memorized password ten times. On the entry screen, the password was not visible, nor was typed text visible. Instead, it was masked with asterisks, as password fields tend to be in use. After participants completed the three phases—practice, verification, and entry—for all ten passwords, they received a surprise recall test. During the surprise recall test, typed text was visible (i.e., not masked).

Note that although modeling cognitive rehearsal and disambiguating memory from motor errors were the foci of the current work, it was necessary to include a description of the larger experiment here as well, since expanding the current model to account for additional phases of the experiment is potential future work. Furthermore, planning for future model expansion to address those additional experimental tasks was influential in determining implementation of the current password typing model.

Results Here we focus on errors rather than timing results from the Stanton and Greene (2014) study. Both are important to test the validity of a model, but the decision to emphasize errors rather than times for password typing parallels their importance in the real world. Accounts are often locked for too many erroneous login attempts, but it is virtually unheard of for a user to be locked out for typing too slowly. Furthermore, knowing which error categories were most prevalent was helpful for determining where to focus our modeling efforts, as well as for evaluating model plausibility.

There were several error classes reported in the aforementioned study. Table 2 presents results from Stanton and Greene (2014) in order of decreasing error category prevalence. Note that the table is ordered based on total

[1] Advanced Password Generator from BinaryMark was used. Disclaimer: Any mention of commercial products or reference to commercial organizations is for information only; it does not imply recommendation or endorsement by the National Institute of Standards and Technology nor does it imply that the products mentioned are necessarily the best available for the purpose.

error percentages; the order would differ slightly if based on either the WDC or UCL participant group.

Table 2: Categorized errors (Stanton & Greene, 2014), rounded to nearest percentage.

Error Category	WDC	UCL	Total
Incorrect capitalization	38%	51%	45%
Missing character	25%	10%	17%
Adjacent key	8%	10%	9%
Wrong character	6%	12%	9%
Transposition of characters	10%	6%	8%
Extra character	7%	7%	7%
Zero instead of an "O"	3%	3%	3%
Wrong place within password	3%	1%	2%

It should be immediately obvious from Table 2 above that incorrect capitalization was the largest class of errors for both participant groups. It is by far the most interesting class of errors to model for several reasons: 1) the sheer magnitude of the incorrect capitalization error class in comparison to other error categories, 2) the practical significance of that error class given current password requirements, and 3) the interesting theoretical question posed by the nature of that particular error.

1) At 45% of the total error corpus, incorrect capitalization errors were nearly three times as likely as the second most prevalent error class (missing character errors, 17% total), and incorrect capitalization errors were five times as likely as the third most prevalent error categories (adjacent key and wrong character errors, each 9% total).
2) The fact that the most frequently occurring error was incorrect capitalization is quite significant given that most modern password policies require at least one uppercase letter. Furthermore, the majority of special characters—also required by many password policies—require shifting. Twenty-one of the total 32 possible symbols require shifting, whereas only 11 do not.
3) A particularly interesting point about incorrect capitalization errors is that based purely on the behavioral data reported in Stanton and Greene (2014), it is unclear whether those errors were memory errors or motor execution errors. Answering this question would help inform typing theory specifically for complex passwords.

An ACT-R Model of Password Rehearsal

Before enabling ACT-R to type capital letters, a cognition-only (i.e., no use of the motor module) model of password rehearsal was constructed, to test whether it alone could account for the errors seen in the behavioral data.

Stimulus Selection

Given the artificiality of having people learn 10 randomly generated passwords in a single session, rather than attempt to model the entire stimuli set at once, a single password was selected for the initial model: *q80<U/C2mv*. This allowed the model focus to be on the cognitive phenomena of interest: rehearsal and retrieval of a single password, which is more reminiscent of a real-world scenario, where we attempt to rehearse a newly generated password to login to a single account. Why select that particular password though? Of the 10 passwords in Table 1, the "q80" password seemed the most interesting to model for several reasons. First, at 10 characters long, it was one of the two middle length passwords. (The shorter passwords are really too easy by today's more stringent password rules, as most higher-security enterprise environments require a minimum length of 10 to 12 characters.) Of the two length-10 passwords, the "q80" password had two non-alphanumeric symbols, whereas the other length-10 password had only a single non-alphanumeric symbol. This is important, as prior work on the linguistic and phonological difficulty of system-generated passwords suggested chunking passwords between non-alphanumeric symbols (Bergstrom, et al. 2014). Furthermore, when asked, people consistently verbalize that password as "Q eighty is less than U over C two M V" and that they "break the password up" at the non-alphanumeric symbols. Since there were no interview data asking people about their chunking or rehearsal strategies reported in the Stanton and Greene (2014) desktop study, it seemed reasonable to use such qualitative observations to inform the current desktop password rehearsal model. When verbally reciting a password, it certainly makes sense that people might chunk passwords (at least initially) in similar ways across platforms.

Model Implementation

For the initial model, the password was broken up into the following chunks based on splitting it at the non-alphanumeric symbols:

1) q80
2) <
3) U
4) /
5) C2mv

In the model's declarative memory, chunks were encoded with their contents, an ID, and a pointer to the next chunk in the sequence. A more complete model of the task would build up these chunks character-by-character. However, since participants in the Stanton and Greene (2014) study were allowed to practice each password as many or as few times as they wished, the initial practice strategies and number of practice repetitions that would account for building up such a representation could vary widely. Rather than implement different models to simulate a variety of practice methods, the model assumes the initial pieces of the password are starting knowledge, and employs a very simple rehearsal strategy. It cycles through chained retrieval of the various chunks in the password to mentally rehearse the stimulus for a period of time that is settable by the modeler. Since ACT-R did not natively support typing the less-than symbol, nor did it support typing errors of any kind, rather than having the model type the retrieved chunks, it simply output them to a file.

The set-similarities option in ACT-R benefits from a principled, ideally a priori, hypothesis as to the nature of the similarities between chunks. In this case we assume that non-alphanumeric symbols are more similar to, and thus more confusable with, one another than are letters to non-alphanumeric symbols, and letters are more similar to one another than are letters to numbers. However, the exact value to assign each similarity is still an open question, and there are 10 such pairwise similarities to set. This seems less than ideal for the current password, and even worse when considering longer passwords that contain a greater number of chunks.

Although the model did predict the nature of the jump-transposition errors humans made (where they transposed the two symbols that were separated by a single letter), it could not account for failure to capitalize the "U", nor could it account for failure to capitalize the "C", which were errors seen in the Stanton and Greene (2014) study. As capitalization errors were by far the most prevalent error in said study, a mechanism for typing capital letters in ACT-R was sorely needed.

Investigating the source of password entry errors is a perfect application opportunity for cognitive modeling to shed light on the root cause of an error (or errors) that was difficult to ascertain through prior behavioral data alone. By implementing support for an ACT-R model that can type capital letters, one could then test different models to see whether those incorrect capitalization errors were memory errors or motor execution errors (where a shift key press had been attempted but simply not executed properly, such as by prematurely releasing the shift key). The ability to type capital letters raises interesting theoretical questions. For each letter of the alphabet, do people have two distinct versions in their memory, one lowercase and one uppercase? Or is an uppercase letter encoded as the lowercase plus a required shift action?

Implementation Issues in ACT-R

In order to support modeling of incorrect capitalization typing errors, two limitations in ACT-R first required addressing: missing special characters, and lack of case-sensitivity in typing.

Missing Special Characters Of the special characters in Table 1, ACT-R previously included support only for the period, semicolon, slash, and quote (Bothell, 2014, see "key" on page 320 of the ACT-R Reference Manual). Therefore, in order to enable modeling typing of the remaining symbols in Table 1 (right parenthesis, question mark, number sign, asterisk, less-than sign, percent sign, caret, underscore, dollar sign), it was necessary to address the somewhat limited prior support for non-alphanumeric symbol typing. As we want to support modeling of *any* possible password, not merely those in Table 1, we added support for all remaining ASCII printable characters not previously supported by ACT-R.

Lack of Case-Sensitivity Regardless of whether calling ACT'R's "press-key" motor module request (Bothell, 2014, see page 317 of the ACT-R Reference Manual) with a capital or lowercase letter, the output will be the same in ACT-R's current instantiation. This is somewhat problematic for modeling incorrect capitalization errors, which requires that ACT-R be capable of press-and-hold capability for the left and right shift keys, combined with a simultaneous key press of a second key (i.e., chorded typing). Therefore we added to ACT-R a capability to type key chords and output case-sensitive text, as described in the following section.

Stochastic Typing Extension for ACT-R

The standard ACT-R distribution (Anderson, et al, 2004; Anderson 2007) does not commit any typing errors as a matter of motor error (Bothell, 2014). However, real humans, even very skilled typists, are imperfect, and tend to err at rates from 0.5% to 35% (Salthouse, 1986; Panko, 2008; Landauer, 1987). We wished to explain password entry errors, but because some errors are due to memory processes and some are due to motor processes, we had to extend our modeling framework of choice, ACT-R, so that it, too, would be capable of such motor errors. Furthermore, we needed to implement the low-frequency, non-alphanumeric characters that information systems often require their users to incorporate into their passwords as a matter of security policy, e.g. "*" or "?". Source code for the ACT-R stochastic typing extension may be downloaded from https://github.com/usnistgov/CogMod.

Motor Errors in Typing

Our typing extension for ACT-R redefines some of ACT-R's existing code so that any requested typing action can stochastically result in the output of a typed key other than the one intended. To do so it adapts the ellipsoid motor movement error equation of May (2012) and Gallagher and Byrne (2013), which leads to greater error along the axis of movement than off the axis, the off-axis error being scaled to .75 of the on-axis. However, because here the units are keys rather than pixels as in May's study, and ACT-R assumes most keys are the same width, the width term in May's equation is simplified to 1.

Hold-Key Because typing non-alphanumeric characters typically involves holding a shift key while striking another key, and standard ACT-R provides no way to hold any such modifier key, it was necessary to invent such a method. Our errorful typing extension provides two motor module request extensions (see "extend-manual-requests" on page 325 of the ACT-R Reference Manual, 2014) to enable the holding and releasing of modifier keys such as shift.

The new hold-key motor module request acts like press-key, translating the requested key to be held into a peck movement (Bothell, 2014, pp. 315-6) with the appropriate features. Once the hold-key motor movement is executed, ACT-R will have a state indicating that the appropriate key is being held. This state in turn causes ACT-R to now output a different character for the same press-key requests that follow for the given keys. The model can request the release-key function to release the given modifier key and end the modifier key state.

Additional Characters With a shift key held, ACT-R can now type a set of ASCII-compatible, non-alphanumeric characters such as "*" and "?." It can now also type capital letters as well as lower-case letters, a critical feature for case-sensitive passwords that standard ACT-R lacks.

Discussion and Future Directions

To address the question of setting appropriate chunk similarities in the initial password rehearsal model, a revised model is underway that has restructured the chunks in declarative memory, and does not use partial matching and set-similarities, instead relying upon spreading activation. This new model is now ready to interface with the stochastic typing extension.

Beyond using the new typing extension, one obvious expansion of the model would be to account for additional phases of the Stanton and Greene (2014) experiment, such as the initial practice and verification tasks. The model should also be expanded to test against the remaining nine passwords and additional stimuli. Modeling the experiment in its entirety would require interfacing with a real or virtual window to control presentation of the stimuli; this would allow the model to visually obtain the stimuli and build up representations of each password in the imaginal buffer letter-by-letter. As an initial model of a larger complex experiment, it seemed more prudent to focus the current work on a single interesting phenomenon, in this case, support for disambiguating memory from typing errors.

We chose to focus on support for disambiguating the most prevalent error in the Stanton and Greene (2014) study, which was incorrect capitalization. As "missing character" was the second most common class of errors in said study, the current stochastic typing extension for ACT-R should be modified to support typing omissions. Furthermore, there are further refinements we would like to make to the ACT-R typing extension to reflect other systematic effects that we did not yet incorporate, such as the likelihood of specific error classes should depend on which fingers are pressing which keys. For example, in traditional transcription typing studies, omissions are more likely with the weaker little finger. Adding support for ACT-R sensitivity to finger/key combinations would benefit future work.

In the future, it would be informative to construct models of the data from the Washington, DC and University College London groups separately to investigate age effects and/or typing skill differences suggested in the Stanton and Greene (2014) desktop password typing study. This would first necessitate updating ACT-R's virtual keyboard to support a standard UK QWERTY keyboard layout. We could then explore modeling parameters for older adults, as recent research suggests that they are task- and device-dependent, and strategy may interact with task and device (Howie, 2015). A deeper understanding of participants' rehearsal and memorization strategies would help inform and test future models.

Regardless of platform, it is important that ACT-R have the ability to commit motor errors when typing so that we can model both memory and typing components of password entry tasks. This is critical to determine which parts of the task are platform-agnostic versus platform-dependent. We should test the password rehearsal model on mobile password typing for smartphones and tablets. Clearly the stochastic typing extensions for ACT-R that we created for modeling desktop password typing would not be appropriate for modeling interactions with mobile keyboards. Instead, we could utilize recent work by Gallagher (2015) and Gallagher and Byrne (2015) on mobile password typing. No doubt device interacts with password complexity, but it would be interesting to see how the initial password learning and rehearsal is affected by device constraints. Are basic password rehearsal strategies similar across devices? A model that utilized the articulatory loop for rehearsal could be viable across multiple platforms.

We think typing differs qualitatively between platforms, especially between desktop and mobile touchscreen computers. Motor scheduling errors should occur in desktop typing when people are typing in parallel and depressing two keys simultaneously. Mobile password typing is more sequential (although it can be interleaved depending on one- versus two-fingered typing style) than is desktop typing. Therefore motor errors on mobile platforms should be more a matter of motor execution accuracy errors than scheduling errors. This would make sense due to the large size of the input device (i.e., a finger) in comparison to the small size of the onscreen keyboard buttons. In fact, research replicating the desktop Stanton and Greene (2014) study on mobile devices (Greene et al., 2014) found that the proportion of adjacent key errors was significantly greater on a smartphone than on a tablet, and the smartphone adjacent key errors were more than twice as prevalent as in the desktop study. Testing the current password rehearsal model across platforms would contribute significantly to disambiguating typing from memory errors.

Regardless of platform, comparison of current and future model predictions to human data could utilize more quantitative measures for comparing errors between passwords. For example, a measure of edit distance such as the Levenshtein distance or the Damerau-Levenshtein distance would be appropriate (Navarro, 2001). Both of these metrics measure differences between sequences based on the number of edit operations required to change the given string into the target sting. However, the former only allows insertions, deletions, or substitutions, while the latter allows those and also transpositions.

Overall, this work illustrated several challenges in modeling a dataset not originally intended for modeling. For example, we do not know if participants in the Stanton and Greene, (2014) study were touch typists, and ACT-R assumes a "moderately skilled touch typist" (Bothell, 2014, see page 317 of the ACT-R Reference Manual). While we made significant progress constructing a model and extending ACT-R's typing ability to better model previously reported behavioral data, it would be ideal to conduct an entirely new study for model validation purposes. A study specifically designed to inform and test model predictions should include more controlled practice with feedback and reinforcement; assign participants fewer passwords but force them to practice them many more times; use a within-

subjects design to test password entry across multiple devices (i.e., desktop and mobile); include a baseline typing test to assess whether participants are touch typists; and explicitly query participants regarding their chunking and rehearsal strategies.

Although there is certainly much work that remains to be done, we feel the current effort was an important first step toward testing theories of password learning and typing on what is still the most prevalent platform for text-heavy tasks: the desktop computer. We now have the capability to begin disentangling memory from motor errors. Both memory and motor are sources of error that must be addressed separately, but that interact with each other within a single integrated system, people.

Acknowledgments

This research was performed while Dr. Tamborello held a National Research Council Research Associateship award at the US Naval Research Laboratory.

A special thanks to Dr. Stefan M. Wierda for his mentorship of Dr. Greene during the 2014 ACT-R Master Class.

References

ACT-R Research Group. (2013). *ACT-R: Theory and architecture of cognition.* [Web page] Retrieved from http://act-r.psy.cmu.edu/

Anderson, J. R., Bothell, D., Byrne, M. D., Douglass, S., Lebiere, C., & Qin, Y. (2004). An integrated theory of the mind. *Psychological Review, 111(4)*, 1036-60. doi: 10.1037/0033-295X.111.4.1036

Anderson, J. R. (2007). *How can the human mind exist in the physical universe?* New York, NY: Oxford University Press. Retrieved from Google Scholar.

Bergstrom, J. R., Frisch, S. A., Hawkins, D. C., Hackenbracht, J., Greene, K. K., Theofanos, M. F., & Griepentrog, B. (2014). Development of a scale to assess the linguistic and phonological difficulty of passwords. In Cross-Cultural Design. Lecture Notes in Computer Science, Volume 8528, 131-139

Bothell, D. (2014). *ACT-R 6.0 Reference Manual.* ACT-R Research Group. Retrieved from act-r.psy.cmu.edu

Coover, J. E. (1923). A method of teaching typewriting based upon a psychological analysis of expert typing. National Education Association, 61, 561-567.

Gallagher, M. A. (2015). Modeling Password Entry on Mobile Devices: Please Check Your Password and Try Again. Doctoral Dissertation, Rice University, Houston TX.

Gallagher, M. A., & Byrne, M. D. (2015). Modeling Password Entry on a Mobile Device. To appear in *Proceedings of the International Conference on Cognitive Modeling.*

Gallagher, M. A., & Byrne, M. D. (2013). The devil is in the distribution: Refining an ACT-R model of a continuous motor task. In *Proceedings of the 12th International Conference on Cognitive Modeling*. Ottawa, Canada.

Gentner, D. (1981). Skilled finger movements in typing. Center for Information Processing, University of California, San Diego. CHIP Report 104.

Greene, K. K., Gallagher, M. A., Stanton, B. C., & Lee, P. (2014). I can't type that! P@ssw0rd entry on mobile devices. In Human Aspects of Information Security, Privacy, and Trust. Lecture Notes in Computer Science, Volume 8533, 160-171.

Howie, N. T. (2015). The generalizability of cognitive modeling parameters for older adults. Doctoral Dissertation, Rice University, Houston TX.

John, B.E. (1988). Contributions to Engineering Models of Human-Computer Interaction, Department of Psychology, Carnegie-Mellon University, Ph.D. thesis.

John, B.E. (1996). TYPIST: A theory of performance in skilled typing. Human Computer Interaction, 11, 321-355.

Landauer, T. K. (1987). Relations between cognitive psychology and computer systems design. In J. M. Carroll (Ed.), Interfacing thought: Cognitive aspects of human-computer interaction (pp. 1-25). Cambridge, MA: MIT Press.

May, K. (2012). A model of error in 2D pointing tasks. Undergraduate Honors Thesis, Rice University, Houston, TX.

Morris, R., & Thompson, K. (1979). Password Security: A Case History. *Communications of the ACM,* 22(11): 594-597.

National Strategy for Trusted Identities in Cyberspace. Enhancing Online choice, Efficiency, Security, and Privacy. (2011). Retrieved online from http://www.whitehouse.gov/sites/default/files/rss_viewer/NSTICstrategy_041511.pdf

Navarro, G. (2001). A guided tour to approximate string matching. ACM Computing Surveys 33 (1): 31–88. doi: 10.1145/375360.375365.

Panko, R. R. (n.d.). Basic error rates. [Web page] Retrieved from http://panko.shidler.hawaii.edu/HumanErr/Basic.htm

Salthouse, T. (1984). Effects of age and skill in typing. Journal of Experimental Psychology, Vol. 113, No. 3, 345-371.

Salthouse, T. (1986). Perceptual, Cognitive, and Motoric Aspects of Transcription Typing. Psychological Bulletin, Vol. 99, No. 3, 303-319.

Stanton, B. C., & Greene, K. K. (2014). Character strings, memory, and passwords: What a recall study can tell us. In Human Aspects of Information Security, Privacy, and Trust. Lecture Notes in Computer Science, Volume 8533, 195-206

Wu, C., & Liu, Y. (2004). Modeling behavioral and brain imaging phenomena in transcription typing with queuing networks and reinforcement learning algorithms. In *Proceedings of the 6th International Conference on Cognitive Modeling.* Pittsburg, PA, USA.