



Human Subtlety Proofs: Using Computer Games to Model Cognitive Processes for Cybersecurity

Ignacio X. Domínguez, Prairie Rose Goodwin, David L. Roberts & Robert St. Amant

To cite this article: Ignacio X. Domínguez, Prairie Rose Goodwin, David L. Roberts & Robert St. Amant (2016): Human Subtlety Proofs: Using Computer Games to Model Cognitive Processes for Cybersecurity, International Journal of Human-Computer Interaction, DOI: 10.1080/10447318.2016.1232229

To link to this article: <http://dx.doi.org/10.1080/10447318.2016.1232229>



Accepted author version posted online: 03 Oct 2016.
Published online: 03 Oct 2016.



Submit your article to this journal [↗](#)



Article views: 26




View related articles [↗](#)



View Crossmark data [↗](#)

Human Subtlety Proofs: Using Computer Games to Model Cognitive Processes for Cybersecurity

Ignacio X. Domínguez, Prairie Rose Goodwin, David L. Roberts, and Robert St. Amant 

Department of Computer Science, North Carolina State University, Raleigh, North Carolina, USA

ABSTRACT

This article describes an emerging direction in the intersection between human–computer interaction and cognitive science: the use of cognitive models to give insight into the challenges of cybersecurity (cyber-SA). The article gives a brief overview of work in different areas of cyber-SA where cognitive modeling research plays a role, with regard to direct interaction between end users and computer systems and with regard to the needs of security analysts working behind the scenes. The problem of distinguishing between human users and automated agents (bots) interacting with computer systems is introduced, as well as ongoing efforts toward building Human Subtlety Proofs (HSPs), persistent and unobtrusive windows into human cognition with direct application to cyber-SA. Two computer games are described, proxies to illustrate different ways in which cognitive modeling can potentially contribute to the development of HSPs and similar cyber-SA applications.

1. Introduction

Cognitive science is the study of the mind and intelligence, where intelligence manifests itself in human behavior through perception, decision making, learning, remembering, taking action, and so forth (Busemeyer & Diederich, 2010; Thagard, 2014). Within cognitive science, researchers have developed explicit models of cognition that can be expressed both qualitatively and in terms of mathematics or a computer program (Farrell & Lewandowsky, 2015), for purposes ranging from testing theories of cognition (Anderson et al., 2004) to improving the design of decision support tools (Segall, Kaber, Taekman, & Wright, 2013). This article focuses on such models in the context of human–computer interaction and cybersecurity (cyber-SA).

Cognitive issues have informed computer security for decades. Saltzer and Schroeder (1975) identified psychological acceptability as an essential aspect of human–computer interfaces to secure systems; such acceptance should make correct use of protection mechanisms “routine.” Some early evaluations of interactive security mechanisms made reference to cognitive phenomena. For example, in the 1980s, Barton and Barton (1984) drew out relationships between password selection and the need to recall passwords from long-term memory. Given the basic division between semantic and episodic memory, they observe that general factual knowledge in semantic memory may be widely shared, which means that users must choose carefully to avoid their password being easily guessed; the typically autobiographical knowledge in episodic memory may be more personal but also public

knowledge, again requiring care. (These recommendations focused on guessability by human attackers, before the common use of dictionary-based attacks.)

Comparable work on passwords has continued based on qualitative accounts of cognition—for example, using graphics for authentication (Mihajlov, Jerman-Blažič, & Shuleska, 2016) or the check-off password system (Warkentin, Davis, & Bekkering, 2004). These efforts can lead to useful guidance, offering theoretical and empirical support for our intuitions. For example, a system-generated password will typically be harder to remember than a password derived from one’s own knowledge and experience. Our intuition tells us that this is true, and a cognitive account explains why.

Cognitive models, applied to comparable problems, generally go deeper into the details of cognition to produce more precise, even quantitative predictions. These models, whether they consist of tracing information through the cognitive system, computational simulations, or closed-form equations, can give more insight than a purely qualitative model of human behavior. Further, such models are based on general principles of cognition (in contrast to data-driven statistical models like regression models or factor analysis that simply need to fit the data), which can support easier generalization to new phenomena (Busemeyer & Diederich, 2010).

In the remainder of this article, we will explore work on cognitive modeling in different areas of cyber-SA. We can break down this work along a few different dimensions. One distinction is between modeling to improve security in the direct interaction between end users and computer systems versus

modeling that improves the abilities of security analysts working behind the scenes. In the former area, we find modeling for automated agent detection, password recall, mental representation of tasks, and expected normal user behavior. In the latter area, cognitive models can be used to support information presentation to analysts seeking to detect and understand cyber-SA exploits; when used for adversary modeling, they can also identify or predict security weaknesses in a system.

Gonzalez, Ben-Asher, Oltramari, and Lebiere (2014) observe that “the development of cognitive models and computational approaches to represent and support cyber [situation awareness] and decision making of the analyst are only in their infancy.” The same is true of cognitive modeling in other areas of cyber-SA. Progress is being made, however.

This article has several goals. One is to highlight areas of cyber-SA in which cognitive modeling research plays a role. Another is to give a brief account of different cognitive modeling approaches, showing how they can be applied to problems in cyber-SA. A third goal is to explicate the process of building cognitive models in this domain—not a unique or canonical process, but one that illustrates the strengths and limitations of modeling in this domain. In particular, we focus on cognitive models that can enable Human Subtlety Proofs (HSPs)—a mechanism to differentiate human behavior from that of automated agents—which we describe below in the context of games research.

2. Related Work

Cognitive modeling techniques have been applied to a range of topics in cyber-SA. This section gives an overview of a few representative approaches on representative problems.

Recalling passwords, as mentioned above, is a topic of continuing interest. Zhang, Luo, Akkaladevi, and Ziegelmayer (2009) take on a modern problem faced by computer users: remembering not only a single password but also multiple passwords and correctly associating them with different systems. Zhang et al. (2009) rely on the stage of memory theory (SMT) (Atkinson & Shiff 1968) to understand why users find difficulty recalling passwords from long-term memory. SMT posits three stages of memory: information is sensed as a stimulus–response pair, this information advances (if it is not lost) to a short-term store, it then enters long-term storage (again, if it is not lost). Retrieval of the information is hindered by decay, interference, and loss of “trace strength.”

Participants in an experiment were asked to create four passwords for four different accounts, following conventional requirements (a minimum of eight characters, at least one number, one uppercase and one lowercase letter, and so forth), and then to recall those passwords to log in to those accounts. A control group was shown a conventional login screen. In one treatment group, the login screen also showed the first character of the user’s password for the associated account; in another treatment group, the rules for constructing passwords were shown. Zhang et al. (2009) identified and analyzed different types of errors in recalling passwords. As expected, errors were due to numbers and special characters, common across the experiment conditions, but interference errors—where having used multiple passwords for different

accounts hampers the ability to recall any specific password—were by far the most common in the control group, occurring three times as often as in either treatment groups. This work relies on a structural model of memory processing, drawing on the assumptions of the model to identify potential difficulties in carrying out a security task.

These experiments serve both a theoretical and a practical purpose: they help cyber-SA researchers better understand the challenge of multiple passwords, in cognitive terms, and they suggest potential strategies for end users in selecting passwords that may be more easily recalled.

Blythe and Camp (2012) describe more detailed cognitive models over a broader set of security behaviors. Their modeling work is based on the identification by Wash and Rader (2011) of different mental models—mental representations of a task or of beliefs and attitudes of others—relevant to security. The authors divided mental models into two categories: “virus” models focused on any generic malware, and “hacker” models that focused on the attacker. End users decide to follow security advice based on their understanding of malware as “buggy” software, software that causes “mischief,” or software to aid in cyber “crime;” they conceptualize attackers as “burglars,” “vandals,” or those interested in “big fish” (entities more important than ordinary users).

Blythe and Camp (2012) developed explicit representations of these mental models, as operators in a simulation that can be executed to determine the security implications of specific actions. For example, will a user find antivirus software of value? Not if the user holds a burglar or big-fish mental model, in which data on the computer under attack are stolen but not destroyed. These models were validated against the survey responses provided by Wash and Rader (2011), giving a good match to the human data.

Blythe (2012) also describes the more general framework in which this work is situated: a cognitive architecture in which reasoning can be modeled at different time scales, allowing for a rich account of security-relevant behavior in different contexts: immediate response to a stimulus, deliberative reasoning, and long-term strategic planning.

In a different area of cyber-SA, support for analysis, we find other approaches to cognitive modeling. Benjamin (2007) and Benjamin, Pal, Webber, Rubel, and Atigetchi (2008) describe the development of a cognitive agent for cyber-SA, monitoring users on a network for signs of suspicious activities, focusing in particular on intrusion detection and vulnerability analysis. Their agent can create virtual copies of a network, simulate the actions of an attacker, and compare with the unfolding results on the real network. Internally, the agent relies on a set of rules for the cognitive architecture Soar (Laird, Newell, & Rosenbloom, 1987). Soar encodes long-term memory as production rules and short-term memory as a semantic graph where object attributes and relationships are maintained. These production rules and encoded knowledge can be used to plan, reason, and execute actions. Several advantages are described for the use of Soar: it can learn from experience, it can reason about attacker’s plans and attempt to predict future behavior, it can engage in natural language communication with human analysts.

Gonzalez et al. (2014) describe how the processes in a cognitive model can be mapped onto the concepts of situation

awareness in cyber-SA. They identify several gaps that need to be filled in order for progress to be made in the area. The most basic is the lack of an explicit model, at the detailed level of a cognitive architecture, of cyber-SA. Other gaps include the decision gap (models of learning, experience, and decision-making), the semantic gap (language and concepts for the domain), the adversarial gap, and the network gap (between models of individuals and of larger security contexts and entities).

Dutt, Ahn, and Gonzalez (2011) and Gonzalez (2013) describe development of a model of cyber-SA based on Instance-Based Learning Theory (IBLT). IBLT represents learning and decision-making for dynamic tasks in terms of situations, decisions, and outcomes/utilities, each of which forms an instance. Instances from past decisions are accumulated over time and used for current situations; an IBLT model chooses among alternative actions based on their computed value. This computation takes into account the retrieval probability of a situation, its outcome, the degree of match to the current situation, and other factors.

IBLT is a cognitive approach in that it relies on the memory mechanisms of the cognitive architecture, ACT-R (Anderson et al., 2004), including base-level activation of instances (based on recency and frequency of use) and spreading activation (in which instances can “reinforce” each other). ACT-R is a general cognitive architecture that has been used in HCI research. The architecture simulates internal cognitive processing (shifts of attention, memory storage and retrieval, decisions) and external (visual and motor) behavior. Structurally, ACT-R contains a set of modules, each representing a different cognitive faculty, with buffers that act as interfaces for the exchange of information between the modules. A model represents specific tasks to be carried out, in the form of production rules to be executed by the architecture.

These research projects give some idea of the scope and promise of cognitive modeling for cyber-SA. We have examples of structural cognitive models and models based on unified cognitive architectures. We see cognitive models applied toward the improvement of practice for end users in security contexts as well as better results for cyber-SA analysts. In the next section, we focus on understanding end users with respect to specific problems in cyber-SA—distinguishing human behaviors from those of automated bots—using casual games as a research platform.

3. HSPs and Cyber-SA Modeling Using Games

User authentication is a critical issue for cyber-SA. Authentication, typically in the form of challenge and response, is something most of us experience every day on our own desktop and laptop computers, on social media Web sites, even when paying for groceries at the store or using a banking machine. In this section, we expand the common view of user authentication, as the process by which users identify themselves individually, to include users confirming that they are members of a group authorized to carry out tasks.

This generalization is useful in framing a relatively new challenge for cyber-SA, the use of bots. Bots are software programs that use a computer, typically a personal computer, for malicious use or use unintended by its owner. For example, so-called aiming bots

were once very popular in online multiplayer first-person shooter games; they allowed players to bypass the game mechanics for targeting opponents, giving them perfect aim every time, and enabling them to artificially improve their standing in the game. Today, bots are more commonly used to register for free email accounts and send spam or phishing messages. Bot detection techniques are designed to prevent such exploits.

Two families of techniques for bot detection are in common use. One is represented by CAPTCHA (Completely Automated Public Turing test to tell Computers and Humans Apart) technology (Von Ahn, Blum, Hopper, & Langford, 2003). The premise behind CAPTCHA technology is to require a human to interactively solve a problem that is difficult (or more desirably impossible) for a computer to solve. The now ubiquitous CAPTCHA technology on the Internet involves having users look at distorted images of words or listen to distorted audio of words and type in the letters. CAPTCHAs have also been explored on touch-enabled mobile interfaces (Leiva & Álvaro, 2015). Another approach, common in massively multiuser online games (MMOGs), involves monitoring a user’s input to identify characteristic differences between human and bot-like behavior. Gianvecchio, Wu, Xie, and Wang (2009), for example, show that differences in the distributions of keystroke durations and the efficiency of mouse movement can be used to distinguish humans from bots. In online poker, systems can use this information to identify poker bots, along with other heuristics such as playing too many games continuously for too long a period of time.

Both of these approaches require users to “prove” that they are human; one requires explicit action on the part of the user, while the other is passive. In other words, one is a human interactive proof (HIP), the other a human observational proof (HOP) (Gianvecchio et al., 2009). Some HIPs can be viewed as challenge-response techniques; they can be integrated into an interactive system to support one-time or periodic bot detection. For example, a poker site might pop up a CAPTCHA between games to ensure that a given player is human. For the human player, this is clearly an extra effort unrelated to the game itself. A different approach can be applied in some games, however. Chow, Susilo, and Zhou (2010) propose that CAPTCHAs can be integrated into a MMOG as a mini-game: for example, making progress in a fantasy adventure game might require players to decode spells, presented visually as CAPTCHAs.

HOPs, in contrast, passively examine the ways in which users complete the tasks they would normally be completing and look for patterns that are indicative of humans versus bots. An example of an observational proof is examining the spatial signature of mouse click locations as influenced by an interface layout (Jorgensen & Yu, 2011). A major advantage to HOPs is that they tend to satisfy the criteria for natural interfaces, making them less obtrusive to users and more likely to be accepted. Another advantage of HOPs over HIPs is that the latter only provides a check at the point in time where it is presented, whereas the former can constantly monitor usage as the task is being performed, making it harder to bypass.

HIPs and HOPs both have significant limitations, however. HOPs are susceptible to imitation attacks, in which bots carry

out scripted actions designed to look like human behavior. Imperfect HOP implementations may also incorrectly classify a legitimate user as a bot. HIPs, on the other hand, tend to be more secure because they require explicit action from a user to complete a dynamically generated test. Because these tests are dynamically generated, solutions to them cannot (reasonably) be predicted, scripted, or generated by computer systems; however, because humans have to expend cognitive effort in order to pass HIPs, they can be disruptive or reduce productivity, and even result in users seeking alternative systems to use.

In response to these limitations, we propose HSPs as an emerging alternative; one that blends the stronger security characteristics of HIPs with the unobtrusiveness of HOPs. HSPs also incorporate the “always-on” property of HOPs in that they monitor usage as it happens. Our approach is to examine how cognitive biases affect interaction with software in predictable and repeatable ways by looking at input device usage patterns. Our goal is to leverage those biases to make small changes to interfaces that will subtly, but not substantively, affect the interaction of either bots or humans. By making changes to interfaces strategically and looking for physical manifestations of the subtle changes in the cognitive processes that only humans would exhibit—and bots would find difficult to fake—we expect HSPs to combine the strengths of both HIPs and HOPs.

We are working with computer games in our research, which provides some advantages. First, because game mechanics often result in changes to the details of tasks, users tend to be more accepting of changes to an interface or expectations on their performance. Second, computer games provide motivational context. In order to get reasonable data, users need to have an incentive to perform the task well. The “gamification” of tasks enables us to study users under experimental conditions with relatively higher engagement when compared with more conventional information processing tasks. Third, using games allows us to precisely control the complexity of the task, in a consistent context. This is important for experimental control and is familiar to players. However, HSPs are applicable to not just games, but to any task that can be slightly altered to trigger subtle changes in cognitive processes.

We describe two games to illustrate different ways in which cognitive modeling can potentially contribute to cyber-SA, and in particular to the development of HSPs.

3.1. The Concentration Game

The Concentration Game, also known as the Memory Game, is a classic solitaire card game in which cards are laid face down on a board or table. On each turn, the player turns over a first card and then a second card so that both are face up. If the two cards match (i.e., if they show the same symbol), the cards are removed from the board. If the cards mismatch, then they are turned face down again and the next turn proceeds. The object of the game is to turn over pairs of matching cards until all of the cards have been removed from the board. For every card on the board, there exists exactly one matching card. Concentration has been used for decades in cognitive science to study memory (Eskritt, Lee, & Donald, 2001) and reasoning about probabilities (Kirkpatrick, 1954).

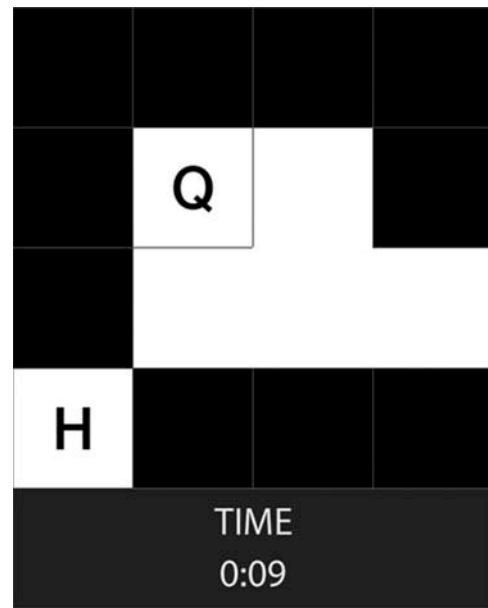


Figure 1. The Concentration interface, speed condition.

We implemented a computer-based version of the Concentration game, as shown in Figure 1. The interface consists of a 4×4 grid of tiles, each tile of 100 pixels square, to represent cards. When representing a face-down card, a tile is black. A face-up tile shows a white background with a single-centered black letter from the set A, B, C, E, H, I, P, and Q. Letters are presented in the Helvetica Neue LT Std 65 Medium typeface. We chose these letters and typeface to minimize letter confusion (Mueller & Weidemann, 2012).

In a game, a turn consists of two actions: turning over a tile by clicking it (which we refer to as a *first-tile move*), and then clicking a different tile (a *second-tile move*). After the second-tile move, the symbols of both tiles are displayed for 1 s; at that point, both tiles are turned face down (in case of a mismatch) or cleared from the board (in case of a match). The player may proceed without waiting for the system to turn tiles back over, by clicking on any face-down tile in the case of match, or by clicking on any tile at all in the case of a mismatch. In either case, the clicked tile then becomes the first-tile move for the next turn. A game is complete when all tiles have been cleared from the board.

We instrumented our implementation of the game to collect mouse pointer motion (position) and click events time stamped with millisecond precision. We associated these events with the state of the game, and of the board. Specifically, each recorded event was annotated with the tile that was clicked or hovered, the turn in which the event occurred, and whether it occurred within the first-tile or second-tile move of the turn.

Modeling the Concentration Game (1)

Our first Concentration Game experiment examined players’ abilities to trade off speed versus accuracy, a well-studied phenomenon in psychology (Wickelgren, 1977). The conditions were distinguished by instructions and payoffs. In the accuracy condition, experiment participants were shown the number of mismatches on the screen during their game play, and they were scored on their ability to minimize mismatches,

a surrogate for the number of turns in a game. In the speed condition, participants were continuously shown the elapsed time in minutes and seconds on the screen, and they were scored on minimizing the time that it takes to clear the board.

To recruit participants, we used snowball sampling, a technique in which initial participants help recruit additional acquaintances through various online social networks. After confirming a recruitment message, participants completed a consent form followed by an optional survey asking for their age, gender, computer skills, and the type of pointing device that they would use for the study. The participant then played a small practice round with an in-game tutorial using a 2×2 board to become familiar with the game rules. Finally, the participants played the game. A total of 179 out of 260 participants (69%) finished the experiment, though not all answered every survey question.

Participants in the accuracy condition averaged 15.7 turns and 40.99 s to complete the game; under the speed condition, they took 18.4 turns and 32.11 s. Full results are described elsewhere (Barik, Chakraborty, Harrison, Roberts, & St. Amant, 2013); here, our discussion is limited to the model developed for the accuracy condition, which we will call the *baseline model*. As is not uncommon in modeling, our analysis was in part exploratory: it is not always obvious which factors determine behavior, or how they should be modeled.

For example, one question for model design is whether tile locations should be treated as positions in space or simply as unique identifiers. We initially hypothesized that after a first-tile move, if the participants have already seen the match, they will either select that match or make a mistake by choosing a nearby tile. We found no clear pattern in the spatial distribution of errors of this type, however. Figure 2 shows a representative distribution over all participant trials, for a matching tile in the top left corner. Given the small probabilities, we chose to model the locations of tiles only as identifiers.

A related issue is the interaction between vision and memory when exploring the board. To narrow the space of possible mechanisms for this choice, we examined the relationship between the duration between clicks and the number of tiles seen; we found a near-zero correlation. We also saw no relationship between click duration and the number of tiles remaining on the board. This suggests that if memory is involved in the choice of new tiles, it is not a simple serial elimination of tiles that have been seen. For simplicity in modeling, we adopted a mechanism in which when a new tile is chosen, it comes from the set of those that have not been recently visually attended.

We considered a number of such issues in the analysis of the experiment data and built a model in the ACT-R architecture. Our goal in modeling with ACT-R is to use the architecture to explain how our experimental results could arise. We worked within the basic architecture, without

0.803 (477)	0.017 (10)	0.017 (10)	0.013 (8)
0.015 (9)	0.024 (14)	0.007 (4)	0.019 (11)
0.017 (10)	0.007 (4)	0.010 (6)	0.008 (5)
0.013 (8)	0.008 (5)	0.015 (9)	0.007 (4)

Figure 2. Participant probability of correctly choosing a previously visited tile, top left (with raw counts) in the accuracy condition. The grid represents the 4×4 game board.

extensions, and we altered ACT-R parameters to fit models to participant data under the different conditions.

The baseline model begins by choosing a visually unattended tile, clicking it, and reading the symbol. The chunk in the visual buffer is copied into the imaginal buffer for storage. The model then attempts to retrieve from memory a second tile that matches the first. If a matching tile is found (subject to memory limitations), the motor module is directed to click that tile. Otherwise, another unattended tile is chosen, clicked, read, and stored in memory. In terms of production firing and module behaviors, the baseline model represents the sequence of actions after the mouse click for a first-tile move as in Figure 3.

A comparison between the baseline model and participant results is shown in Table 1. These cannot be taken as predictions, because the model was fitted to the participant data; instead, the model acts as a contingent explanation of participant performance. Another limitation is that the model is of aggregate (mean) participant performance rather than individual behavior. This can be seen in Figures 4 and 5, where the variance due to different board configurations does not account for the spread in participant performance, which we attribute to individual differences and behaviors that are not captured by the model.

The baseline model is nevertheless a reasonable model of participant performance, at an abstract level. Most of its estimates of the performance measures listed above are within 15% of the observed values, with the exception of Revisit_{2-} at 39%. These estimates work at two levels: they are quantitative timing estimates and they indicate specific choices among

1. Read the text on the first tile
2. Store the text and the location of the first tile in declarative memory
3. Retrieve a matching second tile from memory
4. If the retrieval fails, find an unattended second tile elsewhere on the board
5. Visually attend the second tile
6. Move the cursor to the location of the second tile
7. Click the mouse on the second tile

Figure 3. Cognitive processing in the baseline model.

Table 1. Mean participant and model performance, for turns and time per game; time per move, between clicks; mean probability per game of visiting a previously seen tile on a first-tile move (Revisit_1), visiting a second tile when it is a match (Revisit_{2+}), and visiting the second tile when it is not a match (Revisit_{2-}); the mean probability of choosing a matching second tile when it has been seen before.

	Turns	Time (s)	Move interval (s)	Revisit_1	Revisit_{2+}	Revisit_{2-}	Match
Participants	15.7	40.99	1.048	0.336	0.883	0.226	0.740
Baseline model	15.9	36.60	1.113	0.383	0.848	0.315	0.805

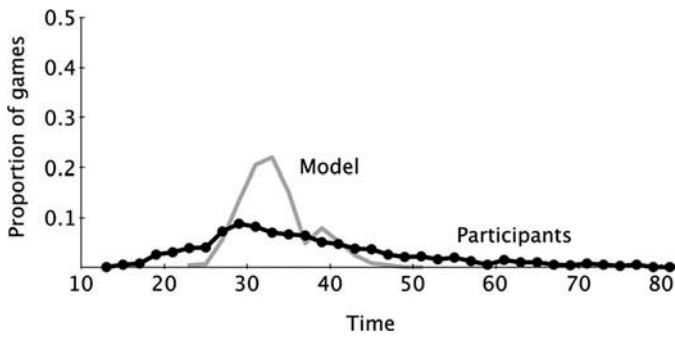


Figure 4. Time per game, accuracy condition.

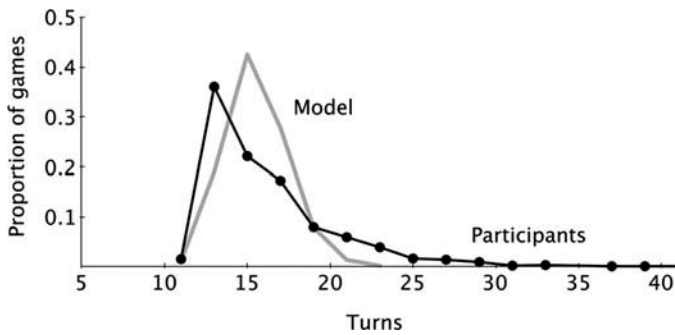


Figure 5. Turns per game, accuracy condition.

alternatives that we expect players of the Concentration game to exhibit.

Although we have not discussed the model for the speed condition in the experiment, it provides a comparable match to human performance. It differs from the baseline model in that it may skip the memory retrieval of a possible match after visiting the first tile on a turn and instead choose an unattended file. We based this on the intuition that participants may sometimes bypass the time needed for memory retrievals, at the cost of missing matches, in order to act faster. It is well understood in psychology that changes to payoffs can influence performance; more specifically, that people can control the tradeoff between speed and accuracy appropriately under different payoff conditions. Our work extended the set of domains for ACT-R modeling in which a speed–accuracy tradeoff can be observed and captured. Our results also suggested features that may be useful in identifying characteristically human behavior: the spatial distribution of tile choices early in the game (most players began in the top left corner and worked their way first to the right and then down), the lack of a spatial correlation between the location of a correct second-tile choice and an incorrect tile, the performance statistics given in Table 1. HSPs could leverage these subtle characteristics of human behavior to differentiate legitimate users from automated agents, particularly in tasks where the user’s goal may differ from that of a malicious bot (i.e., when the user strives for accuracy while the bot aims for speed, or vice versa).

Modeling the Concentration Game (2)

In a second experiment, we modified the game to support examination of a phenomenon more directly related to cyber-SA: deception. The game is as described above, but depending

on the experimental condition, some players had the option to use the space bar on their keyboards to toggle *reveal mode*. This mode would allow players to see the letter of every available tile when these were face down, essentially allowing them to cheat. To avoid confusion about the state of a tile, we used gray for the color of the letters of face down tiles when reveal mode was enabled, as shown in Figure 6.

We used snowball sampling to recruit participants for our study. The recruiting message contained a link to a website where interested people could read the consent form and sign up for a study time slot. Participants were offered a base compensation of \$5.00 for participating, a maximum of an additional \$2.00 for each game round they played, for a total compensation amount of up to a maximum of \$25.00. Their compensation for a round began at the highest value (\$2.00) and decreased by \$0.10 (until it is \$0) for every mismatch that the participant made on that round. Because our control group did not have the ability to enable reveal mode, participants on this condition were given an additional \$5.00 on their base compensation for a total of \$10.00 base compensation. The compensation received for the round is displayed at the bottom right of the game screen, as shown on Figure 6.

The full experiment and results are described elsewhere (Domínguez, Goel, Roberts, & St. Amant, 2015); here, we focus on two conditions: a reveal condition and a no reveal condition, depending on whether reveal mode was enabled or disabled. The difference was made known to participants in the language used when providing instructions in the in-game tutorial and game rounds. Participants with the option to enable reveal mode could toggle it on and off at their leisure during a round. In order to promote deceptive behavior, some participants were told that a cheating detection module was active in the game, and that if they were caught cheating, they

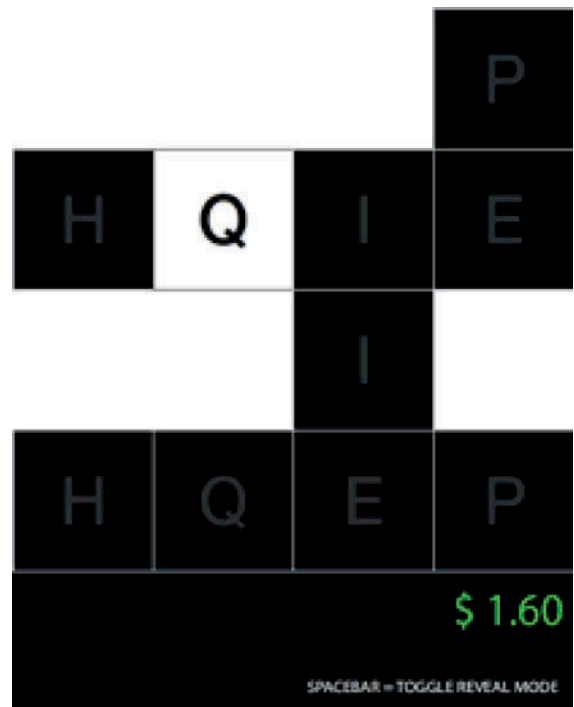


Figure 6. The concentration interface, reveal condition.

would forfeit their entire compensation. In reality, no such module existed.

In terms of mathematical utility, participants in this experiment have the same goal as those in the accuracy condition of the first concentration experiment: to minimize mismatches. The difference between the two experiments is motivational. While participants in the first experiment were trying to achieve a higher score, participants in this experiment were trying to maximize their monetary compensation. Particularly in the no reveal condition, where no deceptive behavior is expected, a monetary reward—as opposed to the intrinsic motivation of playing the game—is the only difference from the accuracy condition of the first experiment. To explore the effects of this motivational change, the baseline model from the first experiment was used to predict performance in the no reveal condition.

While it is well understood that people can control the tradeoff between speed and accuracy under different payoff conditions, deceptive behavior has not been as extensively explored. This, and the fact that participants could toggle reveal mode on and off at any given time, made fitting a model to participant data difficult. For this reason, a new *revealed model* was developed for the reveal condition that relied on visual search to find matches, as a simple bot might do, without attempting to model any specific deception strategy.

The revealed model is a modification of the baseline model that bypasses the storage of a tile's text in memory after a first-tile move, instead relying on its persistence in the visual environment. The attempt to retrieve a matching tile from memory for the second-tile move is also unneeded. All of the tiles are revealed so that a match can be determined visually. Further, because the letters chosen for the game are visually distinct, we model the process of identifying a visual match as a pop out effect, making the location of the match available in constant time. In the revealed model, the sequence of actions given in Figure 3 is streamlined to the form shown in Figure 7.

This allows the revealed model to complete a game with no incorrect choices and a completion time mean of 15.15 s, from the baseline model estimate of 36.6 s. The time interval per move, between clicks, is reduced to 0.947 s (baseline model estimate, 1.113 s).

The predictions of the baseline model have relatively large error for basic performance measures in this experiment, as shown in Table 2. The mean number of turns taken is over-predicted by about 9%, and the total time is underpredicted by about 20%. The revisit statistics show large prediction errors as well. It appears that participants in this experiment follow a superior strategy compared with those in the first experiment. Notably, the lower value for $Revisit_1$ suggests that

1. Read the text on the first tile
2. Find an unattended second matching tile elsewhere on the board
3. Attend the second tile
4. Move the cursor to the location of the second tile

Figure 7. Cognitive processing in the revealed model.

they explore more than in the first experiment; if a participant had perfect memory, it is easy to see that this would result in better performance than choosing an already seen tile as a first-tile move.

In the end, the results of cognitive modeling for the Concentration game to improve our understanding of deception are equivocal. More detailed models may help, but the difference in motivation between the two experiments—the intrinsic motivation of playing the game versus playing for a small monetary reward—appears to produce differences in performance. Further, it is unclear exactly how the participants are spending the time; it may be a rehearsal process to improve later recall, for example, but we have little evidence on which to base a model. What this means for HSPs, however, is that the effects of motivation on human behavior can be leveraged as a mechanism to differentiate humans from bots.

This is not to say that these results are not predictable per se. A data-driven statistical model, relying on measures including the time between clicks, the time between a click and a succeeding mouse movement, change in direction of mouse movement, the number of times the mouse hovers over specific regions, gives almost perfect accuracy in prediction (Domínguez et al., 2015). In practice, and for this particular task, this statistical model can be used as an HSP to classify and distinguish between human and bot activity. However, these measures are at too low a level for our models to capture, which is needed to properly understand the cognitive processes involved and produce HSPs that can better generalize to different tasks.

3.2. A Touch Target Selection Game

The final project we discuss is based on touch interfaces on tablet computers. Touch interfaces are associated with a much higher error rate in target selection than GUI interfaces used with a mouse or touchpad, and in practice, a touch sometimes fails to register. Our work in this area is qualitative rather than quantitative.

The target selection game is a simple one, comparable to those used in Fitts' Law experiments (MacKenzie, 1992). The game presents a scattering of circular red targets on the

Table 2. Mean participant and model performance across conditions, for turns and time per game; time per move, between clicks; mean probability per game of visiting a previously seen tile on a first move ($Revisit_1$), visiting the second tile when it is a match ($Revisit_{2+}$), and visiting the second tile when it is not a match ($Revisit_{2-}$); the mean probability of choosing a matching second tile when it has been seen before.

	Turns	Time (s)	Move interval (s)	$Revisit_1$	$Revisit_{2+}$	$Revisit_{2-}$	Match
No reveal condition	14.6	50.42	1.727	0.285	0.571	0.429	0.857
Baseline model	15.9	36.60	1.113	0.383	0.848	0.315	0.803
Reveal condition	8.3	30.14	1.722	0.017	0.969	0.031	0.063
Baseline model	8.0	15.15	0.947	0.000	1.000	0.000	0.000

display; to complete a round of the game, the player taps all the targets, in any order. The experimental apparatus consisted of an EyeTribe eye tracker attached to the bottom front of a Surface Pro 2 tablet. Both devices were attached to an adjustable height tripod with a universal mount; the neck of the tripod could be raised or lowered to adjust to a user's height. Software instrumentation collected touch timing and locations as well as gaze data.

Participants were told that the experiment focused on how users perceive error on a touch screen device. When the program started, a start button would appear. Once this button was pressed, participants would see five red targets to select with touch input. When a target was touched successfully, it turned grey, as seen in Figure 8. After all targets were touched, the screen cleared itself. Ten participants completed the experiment; all who wore glasses removed them before the experiment began and reported no trouble seeing the targets.

Without greater constraints on the task, users prioritized accuracy over speed. The result was that users did not make target selection errors. Instead, the interface introduced artificial errors by ignoring some events. While performing the experiment task, the system manipulated the nominal error rate, at one of six levels: 0%, 10%, 20%, 30%, 40%, and 50%. For example, in the 10% condition, one out of every ten successful touches is ignored, simulating a missed touch. This accounts for cases in which a participant touches outside the boundary of any target; such a touch is unsuccessful. Nevertheless, it does not capture cases in which a participant touches inside a target boundary and the touch is not detected or misidentified by the system. The experiment started with three rounds of 0% error so that the user could familiarize him or herself with the task. After this initial training, each error level was seen three times but in random order so that every participant completed 21 iterations of the game.

After each round (i.e., once a screen has been cleared of targets), participants were asked to guess the total error rate for that screen, choosing from 10 decile percentages. Participants selected the button corresponding to what they believed to be the closest value. The start button then reappeared for the next round.

Our goal in this experiment was to identify and represent patterns participants exhibited under different error rates. One behavior involves a gaze fixation on a target, a tap, and then a pause to verify that the tap has been recognized. If the tap is successful, the next target is handled, but if the tap fails, then the target is tapped again. Another behavior is to tap targets without waiting for verification, returning to those that

were missed. In either case, visual attention may remain on the target under consideration until a successful tap or move to the next target. A different behavior, apparently derived from gaming experience, relies on peripheral vision to locate targets, with no obvious relationship between gaze fixations and tap locations. Yet, another behavior involves a brief planning phase in which gaze moves between different targets before any one is tapped.

These behaviors can be interpreted in cognitive terms as micro-strategies: low-level processes that describe the interactive behavior between the design of the available artifacts and the cognitive, perceptual, and motor processors (Gray & Boehm-Davis, 2000). Strategies can be identified by analysis of gaze fixations, tap locations, and the duration and ordering of events.

Touch Models

GOMS (goals, operators, methods, and selection) is a modeling framework introduced by Card, Newell, and Moran's *The Psychology of Human-Computer Interaction* (Card, Moran, & Newell, 1983). It is specifically designed to compare alternative ways to complete a task. The goals are objectives to be accomplished using the available operators (i.e., actions) that a user can perform. The Methods are the goals and subgoals necessary to complete the task. When a goal can be accomplished more than one way, the path branches with every alternative method. When the model encounters a branch, selection rules based on contextual heuristics choose which path is taken.

Our models of behavior for this experiment are created in Cogulator (The MITRE Corporation, 2014), a calculator for constructing task analysis models using GOMS. Cogulator defines a basic set of 21 GOMS operators. Each operator has a name, an optional label, and a default duration. New operators can be defined, and default durations can be modified.

Exploratory data analysis led to the identification of several reoccurring microstrategies, in different categories we have defined. Selected microstrategies are visualized in Figure 9.

The first category involves searching for a target to activate. Three patterns are evident.

- *Visual search (VS)* comprises multiple sequential fixations with no touch input. This microstrategy suggests that some decision-making takes place about what action to perform next. VS occurs in 50% of all screens, and the number of fixations in the search versus its occurrence decreases in a logarithmic pattern. It is not correlated with error rate.
- *No visual search* is the absence of visual search.
- *Peripheral focus (PF)* represents a single, unmoving fixation during multiple touch events. Unlike the other microstrategies, this is likely a conscious strategy by the user to keep their eyes still and only use their peripheral vision. Seen in so-called twitch gaming, the user is trying to minimize reaction time by eliminating eye movement. It was only used by one participant three times, all in low-error situations. The PF microstrategy below is a continuation of this one.

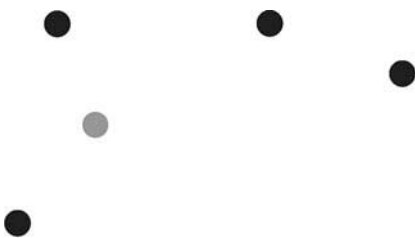


Figure 8. The target selection game interface. One target has been successfully activated.

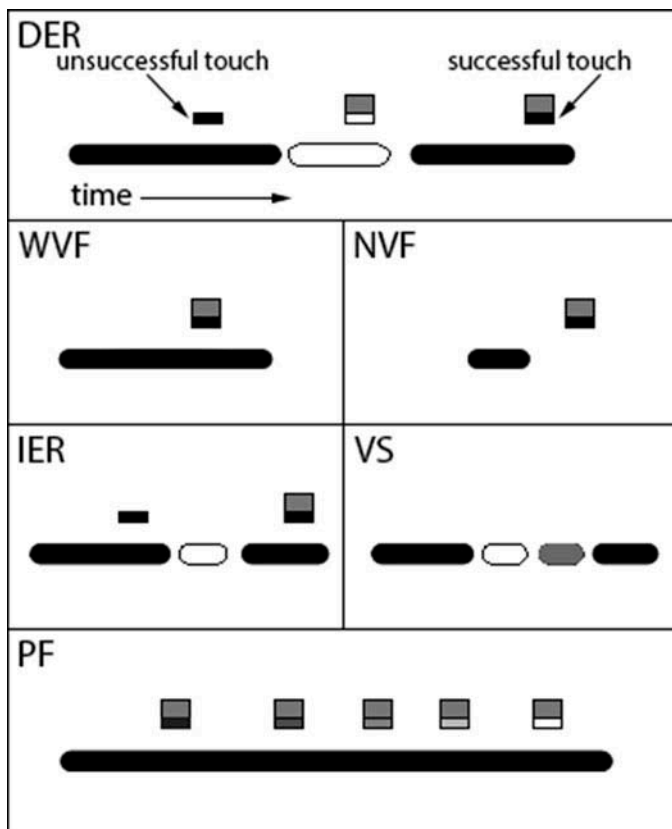


Figure 9. Visualizations of six microstrategies along a timeline. Rounded rectangles represent fixations, with shade uniquely mapping to a target. Touch events are shown as a rectangle above the fixations at the time they occurred. The gray square that is sometimes attached is indicative of a successful action. The rectangle by itself represents an error. The shade of the rectangle follows the same scheme as fixations. Therefore, if a black rectangle appears above a black rounded rectangle, it represents a touch on the target that the participant is currently fixating upon. From top to bottom, the visualization shows the following microstrategies: Delayed-Error-Recovery, With-Visual-Feedback (left), No-Visual-Feedback (right), Immediate-Error-Recovery (left), Visual-Search (right), and Peripheral Focus.

The second category deals with shifting attention away from a target.

- *No visual feedback (NVF)* comprises a fixation start, a fixation end, and then a touch. Both successful and unsuccessful touches are grouped together in this strategy, because they are the same action in a cognitive sense, differentiated only by the effect on the environment. The user anticipates the completion of a touch action without waiting for visual feedback on its success. It is most likely to be used in low-error environments, although it is only used about 20% of the time.
- *With visual feedback (WVF)* comprises a fixation start, a touch, and then a fixation end, or a fixation start, an unsuccessful touch, a repeated touch, etc. This microstrategy is by far the most common one identified in all error levels. Occurring in 96% of all rounds, it appears on average 4.6 times per round.
- *PF* represents cases where the eyes do not move, implying that shifts of attention are entirely cognitive. This microstrategy can result in either NVF or WVF.

The third category deals with choosing the next action.

- *Success with feedback* comprises a fixation start, a successful touch, and a fixation end. In this case, there is no need for error recovery, and the user will either return to the first category of microstrategies, searching for a target, or the task will end.
- *Delayed error recovery* comprises an unsuccessful touch, a fixation end, a sequence of unspecified actions, and finally a touch on a different target. This microstrategy is defined by an indication that the user has noticed the error but has chosen to move on and come back to it later. This microstrategy is seen most in high-error environments.
- *Immediate error recovery (IER)* is defined by an unsuccessful touch followed by another touch on the same target, indicating that the participant saw the error and attempt to fix it immediately. In this microstrategy, fixations can be in many places. This strategy is used twice as often as IER and is most seen in high-error environments.

Table 3 shows the counts of the microstrategies described above, over all participants, grouped into low (<20%), medium (20–40%), and high (>40%) error conditions.

Notably, participants were able to estimate the error rate with some accuracy: over all participants, the mean perceived error rate was 5% higher than the nominal error rate (standard deviation 15%) produced by system manipulation, with the median estimate being 3% higher than the nominal error rate. Per participant, estimates differed by an average of 12% (standard deviation 10%).

While these results remain to be validated in larger experiments, they suggest further avenues for exploration. Our results indicate that microstrategies vary, but whether or not users perceive the errors as coming from themselves or whether they are losing trust in the system remains an open question. If users are changing their behavior based on personal actions, it opens the possibility of interventions and observation of user behavior. If the error is perceived as coming from the system, then behavior can be matched to a taxonomy of responses based on human trust of the system reliability (Muir, 1994). Our results indicate only that touch-based systems do not impede users' sensitivity to different error rates.

Critically, we can influence the target selection error rate, either directly (by making targets larger or smaller) or indirectly (by simply ignoring taps, with some probability). The implication for HSPs is that if users are sensitive to the difference in error rates (we have evidence that this is the

Table 3. Count of microstrategy occurrences by type in low-, medium-, and high-error conditions.

	Total	Low	Medium	High
With visual feedback	930	423	250	257
No visual feedback	189	80	52	57
Immediate error recovery	307	24	90	193
Delayed error recovery	192	68	58	100
Visual search (≥ 2 targets)	192	68	66	58
Visual search (≥ 3 targets)	58	22	23	13
Peripheral focus	3	3		

case), then we may be able to manipulate the interface to see if the user reacts in a way that we expect, for example adopting a slower, more “careful” strategy for a higher error rate. Multiple target selection is a common enough task in touch-based interfaces that it could potentially act as the background for an HSP; this is part of our ongoing research.

4. Conclusion

Nicol, Sanders, Scherlis, and Williams (2012) identify “understanding and accounting for human behavior” as one of the five hard problems to address at the foundation of the science of security. Modeling behavior of both users and adversaries can help security analysts and computer systems to identify and differentiate microstrategies employed by legitimate users from those with malicious intent. Furthermore, cognition, as a predictable and measurable human characteristic, can be leveraged to recognize the differences between human behavior and that of an automated agent.

What do these microstrategies and models tell us about cyber-SA? The answers lie in the future of HSPs described in this article. Every interaction we have with a computer is a task that requires certain predictable cognitive, perceptual, and motor processes to successfully complete. Interaction designers have long leveraged this insight in the modeling and implementation of new interfaces. These approaches to interface design involve the specification of interface tasks while taking into account how those tasks will require different cognitive processes. For example, we know that under normal circumstances, the Concentration game players rely on working memory, but under our “reveal mode” condition, the game requires visual search. This subtle change in behavior, while natural for humans, would be difficult for bots to replicate. While that example may seem a bit contrived, it highlights an important point about human interface usage and how it can be modeled and leveraged to improve security:

Changing the tasks humans perform with interfaces has predictable, albeit sometimes probabilistic, effects on the cognitive, perceptual, and motor processes required to complete them that can be detected through the modeling of input device and sensor analytics.

In other words, by making small changes to the tasks people perform with interfaces, we can have detectable effects on the various perceptual, cognitive, and motor processes required to complete those tasks. By modeling how these changes in cognitive, perceptual, and motor operations occur in humans and comparing them with observed behavior, we can determine whether the observed actions are being made by a human or a bot.

While this work is in its early stages, what we have learned so far is encouraging. First and foremost, the settings in which we have conducted experiments have all revealed actionable insights that will contribute to the base of knowledge for HSPs in the coming years. Beyond that, storage, processing, and sensing technologies have matured enough to make data-driven exploration of these phenomena feasible using inexpensive, commercially available hardware. When our understanding of how cognitive, perceptual, and motor phenomena manifest

themselves in interface analytics matures beyond the early stages of the research, we described that in this article, the technology of HSPs will enable new advances in persistent security proofs for a wide range of software systems, not only in differentiating human behaviors from those of bots, but also in differentiating desired from undesired human behavior.

ORCID

Robert St. Amant  <http://orcid.org/0000-0003-1417-9278>

References

- 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), 1036–1060.
- Atkinson, R. C., & Shiff, R. M. (1968). Human memory: A proposed system and its control processes. *The Psychology of Learning and Motivation*, 2, 89–195.
- Barik, T., Chakraborty, A., Harrison, B., Roberts, D. L., & St. Amant, R. (2013). Modeling the concentration game with act-r. In *Proceedings of the 12th international conference on cognitive modeling (ICCM)*. Ottawa, Ontario, Canada.
- Barton, B. F., & Barton, M. S. (1984). User-friendly password methods for computer-mediated information systems. *Computers & Security*, 3 (3), 186–195.
- Benjamin, D. P. (2007). A cognitive approach to intrusion detection. In *IEEE symposium on Computational intelligence in security and defense applications (CISDA 2007)* (pp. 161–168). Honolulu, Hawaii.
- Benjamin, D. P., Pal, P., Webber, F., Rubel, P., & Atigetchi, M. (2008). Using a cognitive architecture to automate cyberdefense reasoning. In *Ecsis symposium on bio-inspired learning and intelligent systems for security* (pp. 58–63). Edinburgh, Scotland.
- Blythe, J. (2012). A dual-process cognitive model for testing resilient control systems. In *International symposium on resilient control systems* (pp. 8–12). Salt Lake City, UT.
- Blythe, J., & Camp, L. J. (2012). Implementing mental models. In *2012 IEEE symposium on Security and privacy workshops (spw)* (pp. 86–90). San Francisco, CA.
- Bussemeyer, J. R., & Diederich, A. (2010). *Cognitive modeling*. Thousand Oaks, CA: Sage.
- Card, S., Moran, T., & Newell, A. (1983). *The psychology of human-computer interaction*. Taylor & Francis. Retrieved from <https://books.google.com/books?id=30UsZ8hy2ZsC>
- Chow, Y. W., Susilo, W., & Zhou, H. Y. (2010). Captcha challenges for massively multi-player online games: Mini-game captchas. In *2010 international conference on Cyberworlds (cw)* (pp. 254–261). Singapore, Singapore.
- Corporation, T. M. I. T. R. E. (2014). *Cogulator*. Retrieved May 15, 2016, from <http://cogulator.io/> (Online)
- Domínguez, I. X., Goel, A., Roberts, D. L., & St. Amant, R. (2015). Detecting abnormal user behavior through pattern-mining input device analytics. In *Proceedings of the 2015 symposium and bootcamp on the science of security (HotSoS 2015)* (pp. 11:1–11:13). ACM. Urbana, IL.
- Dutt, V., Ahn, Y.-S., & Gonzalez, C. (2011). Cyber situation awareness: Modeling the security analyst in a cyber-attack scenario through instance-based learning. In Y. Li (Ed), *Data and applications security and privacy XXV* (pp. 280–292). Richmond, VA: Springer.
- Eskritt, M., Lee, K., & Donald, M. (2001). The influence of symbolic literacy on memory: Testing Plato’s hypothesis. *Canadian Journal of Experimental Psychology/Revue Canadienne de Psychologie Expérimentale*, 55 (1), 39–50.
- Farrell, S., & Lewandowsky, S. (2015). An introduction to cognitive modeling. In B. U. Forstmann and E.-J. Wagenmakers (Eds.), *An introduction to model-based cognitive neuroscience* (pp. 3–24). New York, NY: Springer.

- Gianvecchio, S., Wu, Z., Xie, M., & Wang, H. (2009). Battle of botcraft: Fighting bots in online games with human observational proofs. In *Proceedings of the 16th ACM conference on computer and communications security* (pp. 256–268). Chicago, IL.
- Gonzalez, C. (2013). From individual decisions from experience to behavioral game theory: Lessons for cybersecurity. In S. Jajodia, A. K. Ghosh, V.S. Subrahmanian, V. Swarup, C. Wang, X. S. Wang (Eds.), *Moving target defense II* (pp. 73–86). New York, NY: Springer.
- Gonzalez, C., Ben-Asher, N., Oltramari, A., & Lebiere, C. (2014). Cognition and technology. In Kott, C. Wang, R. F. Erbacher (Eds.), *Cyber defense and situational awareness* (pp. 93–117). New York, NY: Springer.
- Gray, W. D., & Boehm-Davis, D. A. (2000). Milliseconds matter: An introduction to microstrategies and to their use in describing and predicting interactive behavior. *Journal of Experimental Psychology: Applied*, 6 (4), 322.
- Jorgensen, Z., & Yu, T. (2011). On mouse dynamics as a behavioral biometric for authentication. In *Proceedings of the 6th ACM symposium on information, computer and communications security* (pp. 476–482). Hong Kong, China.
- Kirkpatrick, P. (1954). Probability theory of a simple card game. *The Mathematics Teacher*, 47, 245–248.
- Laird, J. E., Newell, A., & Rosenbloom, P. S. (1987). Soar: An architecture for general intelligence. *Artificial Intelligence*, 33 (1), 1–64.
- Leiva, L. A., & Álvaro, F. (2015). μ CAPTCHA: Human interaction proofs tailored to touch-capable devices via math handwriting. *International Journal of Human-Computer Interaction*, 31(7), 457–471.
- MacKenzie, I. S. (1992). Fitts' law as a research and design tool in human-computer interaction. *Human-Computer Interaction*, 7 (1), 91–139.
- Mihajlov, M., Jerman-Blažič, B., & Shuleska, A. C. (2016). Why that picture? discovering password properties in recognition-based graphical authentication. *International Journal of Human-Computer Interaction*. In press.
- Mueller, S. T., & Weidemann, C. T. (2012). Alphabetic letter identification: Effects of perceivability, similarity, and bias. *Acta Psychologica*, 139 (1), 19–37.
- Muir, B. M. (1994). Trust in automation: Part I. theoretical issues in the study of trust and human intervention in automated systems. *Ergonomics*, 37 (11), 1905–1922.
- Nicol, D. M., Sanders, W. H., Scherlis, W. L., & Williams, L. A. (2012, November). *Science of security hard problems: A lablet perspective*. Science of Security Virtual Organization Web. Retrieved from <http://cps-vo.org/node/6394>
- Saltzer, J. H., & Schroeder, M. D. (1975). The protection of information in computer systems. *Proceedings of the IEEE*, 63 (9), 1278–1308.
- Segall, N., Kaber, D. B., Taekman, J. M., & Wright, M. C. (2013). A cognitive modeling approach to decision support tool design for anesthesia provider crisis management. *International Journal of Human-Computer Interaction*, 29 (2), 55–66.
- Thagard, P. (2014). Cognitive science. In E. N. Zalta (Ed.), *The Stanford encyclopedia of philosophy*. Retrieved from <http://plato.stanford.edu/archives/fall2014/entries/cognitive-science/>
- Von Ahn, L., Blum, M., Hopper, N. J., & Langford, J. (2003). Captcha: Using hard ai problems for security. In E. Biham (Ed.), *Advances in cryptology - EUROCRYPT 2003* (pp. 294–311). New York, NY: Springer.
- Warkentin, M., Davis, K., & Bekkering, E. (2004). Introducing the check-off password system (COPS): An advancement in user authentication methods and information security. *Journal of Organizational and End User Computing*, 16 (3), 41–58.
- Wash, R., & Rader, E. (2011). Influencing mental models of security: A research agenda. In *Proceedings of the 2011 workshop on new security paradigms workshop* (pp. 57–66). Marin County, CA.
- Wickelgren, W. A. (1977). Speed-accuracy tradeoff and information processing dynamics. *Acta Psychologica*, 41 (1), 67–85.
- Zhang, J., Luo, X., Akkaladevi, S., & Ziegelmayr, J. (2009). Improving multiple-password recall: An empirical study. *European Journal of Information Systems*, 18 (2), 165–176.

About the Authors

Ignacio X. Domínguez is a doctoral student in Computer Science at North Carolina State University. He is interested in creating computational models that can be used to identify and predict human behavior in virtual environments. He likes to use computer games as tools to elicit different behaviors.

Prairie Rose Goodwin has a B.A. in Computer Science from Vassar College, and a MS in Computer Science from North Carolina State University where she is currently pursuing her Ph.D. Her research focuses on improving technology for regular people by creating cognitive models to improve touch screen usability.

David L. Roberts is an Associate Professor of Computer Science at North Carolina State University. He received his Ph.D. from the Georgia Institute of Technology in 2010. His research interests lie at the intersection of behavior, data, and computational modeling, particularly on the role of computation in understanding and influencing behavior.

Robert St. Amant (Ph.D., UMass, 1996) is an associate professor of computer science at NCSU. The target of his research is models of interaction, drawing on concepts in human-computer interaction, cognitive science, and artificial intelligence. He has also written a popular science book, *Computing for Ordinary Mortals*.