Playful Security: A Computer Game for Secure Wireless Device Pairing

Alexander Gallego
Polytechnic Institute of NYU
Six MetroTech Center
Brooklyn, NY 11201
agalle01@students.poly.edu

Nitesh Saxena
University of Alabama at Birmingham
1300 University Boulevard
Birmingham, Alabama 35294
saxena@cis.uab.edu

Jonathan Voris
Polytechnic Institute of NYU
Six MetroTech Center
Brooklyn, NY 11201
jvoris@isis.poly.edu

***Abstract*—The secure "pairing" of wireless devices based on out-of-band communication is an established research direction. Unfortunately, this approach is prone to human errors that lead to man-in-the-middle attacks. To address this and to better motivate users, this paper proposes the use of computer games for pairing. Games make the pairing process enjoyable and engaging, thus improving its usability and security. The technical contribution of this work is a new pairing system called "Alice Says." This is a game that achieves pairing and is based on the memory game Simon. We also discuss the design and implementation of Alice Says.**

On a broader note, this paper also points to other security problems that are currently lacking optimal solutions and suggests how games and entertainment can be applied to improve them.

Keywords: Games, Entertainment, Device Pairing, Security, Mobility, Ubiquitous Computing, Usability.

I. Introduction

Short-range wireless communication systems are growing in popularity and will continue to do so in the future. Their widespread deployment unfortunately brings security risks along with it. Wireless channels are easy to eavesdrop upon and manipulate. Protecting them is therefore a fundamental security objective. In this paper, the term "pairing" refers to the initialization of secure communication between two wireless devices in a way that is resistant to eavesdropping and man-in-the middle attacks.

A promising pairing research direction is to leverage an out-of-band (OOB) channel that is governed by human users. Unlike classical radio channels, OOB channels are "human-perceptible," i.e., the underlying transmission and reception that drives them can be perceived by human senses. In contrast to traditional wireless channels, this property means that OOB communication provides authentication and integrity.

The usability of a pairing method is critical. OOB channels typically have low bandwidth. Thus the less data that a pairing method needs to transmit over these channels, the better the technique is in terms of usability. A recent innovation to this end is the development of Short Authenticated String (SAS) based protocols [46], [47], [33], [50] that limit the length of data transmitted over OOB channels to approximately 15 bits. We refer the reader to several excellent surveys and comparative usability analyses of various OOB pairing methods [4], [28], [21]. These pairing methods will also be reviewed in the related work section on pairing techniques.

The focus of this paper is on *social pairing* scenarios, as discussed in [3], in which two different users control their respective devices while pairing them. Examples include pairing laptops, tablets, or cell phones for social or professional reasons.

A. Research Challenges

In practice, the social pairing problem is daunting and has not been addressed in a satisfactory manner despite recent research. Prior work on pairing raises several fundamental usability and security related concerns The most prominent of these are as follows:

- **Short OOB Strings:** Most existing pairing methods are based on SAS protocols that use very short strings. The level of security provided by these methods may therefore not be sufficient for certain applications. Increasing the length of SAS strings, however, may lead to poor usability because the process takes longer to complete. Methods that are automated and can transmit longer SAS strings have also been shown to have undesirable usability properties [4].
- **Human Errors:** Even while using short OOB strings, comparison based pairing methods do not offer the theoretical level of security guaranteed by their underlying protocols [4]. This is due to the potential these protocols have for human error. Mistakes made during the pairing process can impact both the usability and security of a system. There are two types of these errors: *fatal* and *safe* [15]. Fatal errors occur when a user accepts a pairing instance despite the OOB strings on the two devices not matching. This kind of error may result in an attack. Safe errors, on the other hand, occur when a user rejects a pairing instance even when the displayed OOB strings match. Such errors undermine the usability of pairing and can also indirectly impact security. This is because a failed pairing necessitates repetition, which may lead to user annoyance and eventually translate into an attack.
- **Rushing User Behavior:** The security of OOB pairing relies upon decisions made by users. As a result, a *rushing user* [37] may simply just "accept" the pairing session

without correctly taking part in the decision making process.

These challenges motivate the design of a radically different approach to pairing. The central research question this raises is: *can users be incentivized to correctly take part in the pairing process, improving its security and enjoyability as a result?*

B. Motivation: Games for Pairing

To answer this question, we propose a novel research direction involving the application of computer games to device pairing. The incentive that we aim to provide to users is fun and entertainment. Our hypothesis is that games may improve the security and usability of pairing and will therefore help address the aforementioned challenges. Our overarching idea is rooted in psychology and based on the principle of extrinsically motivated design [45]. Based on the sheer popularity of games [52], playful approaches to pairing promise to appeal to a large population of users, particularly younger individuals [40].

Next, we delve deeper as to *why games should be used to address usable security problems such as device pairing*. While performing security tasks, users may not be aware of or care about the impact their actions have on the security and privacy of their devices. Users may not do their best at the task, or may attempt to skip it entirely, due to this lack of engagement in the security process

To address this issue, we propose the reframing of security tasks not as tedious procedures that place a costly burden on users, but rather as playful processes that are enjoyable to complete. It is our aim to transform pairing from a chore that users seek to avoid or complete as quickly as possible into something that they relish. As a result, users will be more attentive to and aware of the steps they must follow while executing this security operation and will perform better at it. Furthermore, if a game involves competitiveness, this will provide another layer of motivation for users. Another important side effect of utilizing a game is that users might be willing to dedicate more time to the security process. In the context of pairing, longer OOB strings can thus be used, providing a higher level of security.

In essence, by contextualizing a security task as playful rather than a chore, the usability burden it imposes may be greatly reduced. We dub this the *Tom Sawyer Effect* after a well known event in Mark Twain's literary classic, "The Adventures of Tom Sawyer" [35]. In this novel, the boy Tom is punished by being forced to paint a fence on his day off. To escape his plight, the clever Tom treats the task as fun rather than resenting it. Upon observing his delight, his friends insist that they be given an opportunity to paint the fence so that they can enjoy it as well. Much in the same way that Tom convinces his friends to complete what would otherwise be considered an uninteresting job by treating it as a game, we seek to persuade users to be attentive during security operations by making these operations enjoyable. Much like Tom's friends, users will aim to achieve precisely the same security goals before and after the addition of playfulness, but will be more inclined to participate due to the perception of fun.

The pairing mechanism we present is suitable for a significant audience due to its playful yet intuitive nature. We base this conjecture on the sheer popularity of games. According to a 2007 study, 72% of the US population with ages between six to forty four have played video games [52]. The gender distribution of gamers is balanced, with 40% being female [16]. Of this user population, children are the most avid players of games. According to the results of a survey reported in 2008 [40], 97% of children between twelve and seventeen years of age play video games. Game oriented security solutions seem ideally suited for children due to their enthusiasm for games as well as their critical security and privacy needs.

C. Our Contributions

The technical contribution of this work is a new pairing system which we call "Alice Says." Alice Says is based on a popular memory game called Simon. It accomplishes the underlying task of the manual transfer of OOB strings between two devices. We report on the design and implementation of Alice Says. We also discuss several other security problems which are lacking optimal solutions and provide suggestions as to how entertainment can be used to address them. At a higher level, our work opens up a new area of research in usable security where security tasks are presented in playful ways by making use of computer games. Designing games that are optimal in terms of speed, error tolerance, and psychological acceptability for a given application is an ongoing research challenge.

II. Design of a Pairing Game

A. Threat Model

We use the adversarial model suggested by Vaudenay [50]. Wireless devices may establish two types of communication channels. The first is a traditional wireless connection, which is characterized by a large bandwidth capacity and bidirectionality. The second comprise the set of OOB channels, which feature modest bandwidths but are physically authenticatable. That is, OOB channels are crafted from output which can be perceived by unassisted humans, which allows them to verify transmission sources themselves. This implies that malicious entities are not capable of modifying messages sent via OOB means. OOB channels are not generally secret, however. In other words, adversaries can observe the OOB transmission. In contrast, opponents have a complete control of the conventional wireless channel. Denial-of-service attacks are beyond the scope of this model.

B. Choosing a Game

In order to leverage the Tom Sawyer effect to improve the device pairing experience, a suitable game had to be designed. We took our inspiration from Hasbro's Simon [7]. This game was selected as a basis for our pairing game for several reasons. Rather than create a new solution from scratch which users may not have found to be enjoyable, we hoped to leverage the known popularity of Simon. Further, this game is relatively uncomplicated when compared with many contemporary computer games. This was desirable both due to its ease of implementation as well as its suitability for players of different ages and levels of experience. Finally, an important factor in the selection of this game is its close relation to existing device pairing solutions. Previous work has established the use of patterns of synchronized audio and visual output as a viable method of securely associating devices [44], [54], [9]. At its core, playing Simon involves nothing more than the short term memorization of audiovisual patterns and thus minimal changes were required to adapt it for use in pairing.

C. Alice Says Game Design

Upon initially starting Alice Says, users are provided with two menu choices: a *single player training mode* and a *two player pairing mode*. A single player mode is provided to allow users an opportunity to unwittingly train themselves to improve their device pairing performance. The two player mode is what actually accomplishes device pairing.

1) Single Player Mode: The single player mode is essentially identical to the classic version of Simon, only adapted to the context of a mobile device. The various steps involved in this mode are discussed below.

- The user is shown a screen with four adjacent squares which fully occupy the screen, dividing it into quadrants. Each of these squares is a unique and distinctive color. Clockwise starting in the upper left, the colors are green, red, blue, and finally yellow.
- The only other item visible while the game is underway is a counter which tracks the number of rounds that have been played thus far. The counter is incremented with each round that is completed successfully.
- One of the four quadrant buttons is randomly selected by the device during each round. The selection is indicated to the user in two complementary ways. First, the screen section is lit by increasing its luminance. Secondly, a tone corresponding to that quadrant is played. The notes associated with the four portions of the screen are harmonically compatible irrespective of the order in which they are played. The use of such notes was critical to Simon's popularity [41].
- Each round encodes two randomly chosen bits in the following manner: "00" corresponds to green, "01" is indicative of red, "10" means blue, and "11" is aligned with yellow.
- If a user presses the correct quadrant, the device lengthens the puzzle pattern by displaying the previous pattern element followed by the color encoding another pair of randomly chosen bits. Similar to the previous pattern element, this is again conveyed to the user by brightening the relevant portion of the screen and playing a corresponding melodic tone.
- The above process continues until the user makes an error or a certain pre-determined round threshold value is reached. At this point a "Game Over" message is displayed, informing the user of the number of rounds that he or she was successfully able to play.

2) Two Player Mode: The two player mode accomplishes the underlying pairing task of transfer of OOB strings between two devices. It differs from the single player, traditional Simon approach in two main ways:

1) The game is split across two devices. One device (A) displays the pattern to the user, but does not handle input. The other device (B) does not display the pattern, but only accepts user input. Using the game, an input OOB string is transferred from A to B. Please recall that each device computes an OOB string as a result of the SAS protocol. In a transfer-based pairing method, these strings then need to be exchanged over the OOB channel and compared by the devices.
2) The game does not conclude when a mistake is made. It continues until a sufficient number of OOB bits have been relayed between the two devices. This makes the game robust to human errors.

As a result of the second difference, a mechanism is required to keep the two devices in sync during the pairing procedure. This is because the device accepting input must be aware of what rounds the user has played to be able to conclude whether or not he or she has committed a mistake. A naive way to handle this would be to simply transmit the current round over the wireless channel. This would ruin the security of the system, however, as the wireless channel is assumed to be totally insecure as per the security assumptions detailed in our threat model. Thus, an adversary could simply transmit an arbitrary round number over this channel, bypassing as much of the process as desired. Instead, we addressed the synchronization issue by integrating "previous" and "next" buttons into A's interface. The use of these buttons and other steps involved in the two player mode are provided in Algorithm 1 and intuitively described in the following list.

- In the first round, the user will be provided with a pattern, on device A, of length one. Then the pattern will be extended to length two, and so on. That is, the original pattern consisting of a single color and tone that encodes the first two bits of A's OOB string will be concatenated with a new value encoding the next two bits. This will form a new pattern that is comprised of two colors and two tones which express four bits. Next, another two bits will be appended to the pattern to form a six bit pattern that is displayed to the user in the form of three colors and three tones. This process continues in an iterative fashion.
- Upon a successful round, the player in control of the display phone A presses next to advance the state of the game. A then displays the new pattern in the next round. If an error is made during the round, on the other hand, A's user can indicate this to A by pressing the previous button.
- In the event of an error in a round, a new pattern is

Algorithm 1 Alice Says Pseudocode

```
input: OOB string a on device $A$ and b on device $B$
display string d = [a[0],a[1]]
k = 1
while k $<$ threshold {threshold is the number of bits that need to be matched, which is equivalent to the lengths of a and b} do
    displayPattern(d)
    I = getInputPattern()
    correctMatch = true
    for n $<$ length(I) {n is used as an index into the input string and device $B$'s OOB string b} do
        if I[n] != b[n+offset] {offset is the index into b, of bits that have been successfully matched so far} then
            correctMatch = false
        end if
    end for
    if correctMatch then
        d = [d[0],d[1],...,d[m-1],a[k+1],a[k+2]] {m is the length of d}
        k = k + 2
        displayCorrect() {user of $A$ presses the next button}
    else
        d = [a[k+1],a[k+2]]
        displayError() {user of $A$ presses the previous button}
    end if
end while
displaySuccess()
```

crafted starting with the pattern portion that was not copied successfully. This is done because, as an invariant, the devices know that all of the rounds up until this point have been successful.

- After an error is made, the input device B will compare its running tally of successful matching bits, of its own OOB string, to how many rounds need to be performed. If more rounds are necessary, the game continues.
- If mutual authentication is desired, the roles of the phones can be swapped following a successful game in a single direction. After this, the game play would proceed in precisely the same way with the roles of the two phones and users reversed.

3) Example Usage Scenario: The following is an example of an anticipated game play pattern. Let us assume a legitimate pairing session in which a user is consistently able to follow 5 rounds. There are 30 bits in the OOB string that need to be compared.

- The user will first be provided with a pattern of length one in round one. The user will successfully match this pattern. Then the pattern will be extended to length two in round two, and so on.
- Let us say that on the sixth round, the user makes a mistake. At this point, the user has successfully transferred the first 10 bits of the OOB string. In the next round, the game will begin a new pattern starting with the 11th and 12th bits of the OOB string, One color will be displayed. The process is repeated as before.
- Assume that the user makes another mistake at round 11. Now, the game will begin with a new pattern starting with the 21st and 22nd OOB bits. The process is repeated as before.
- After successfully completing the next 5 rounds, all 30 bits will have been conveyed over 15 rounds in total, concluding the game.

D. Security Guarantees

An important aspect of Alice Says is how it handles attacks. If a session has been attacked, the OOB strings calculated on the two devices will be different. Even if a user "correctly" transfers the displayed pattern, Alice Says will register an error. This will either occur as the first bit of a pattern or a subsequential bit of the pattern. If the attack occurs as a subsequential bit, the bits prior to the error will be registered as a match for that session and the pattern will begin anew with the attacked bit, making it the first bit of the next pattern. Thus, one way or another, the attacked bit will end up as the first bit of a pattern, and users will be unable to proceed by identifying the single color pattern that has been displayed to them.

Users therefore need a mechanism for restarting a pairing session. To achieve this, after a certain threshold of single color pattern mismatches have occurred, an error message will be displayed. At this point, users can discard the session and start over with a new one. Note that single color pattern mismatches are unlikely in unattacked sessions as most users can match at least one color. Thus, the only way for a critical error to occur in this system is for users to incorrectly match a single color. Given an OOB string of length n, there is only a $1/4 * n * 2^{n-1}$ chance of this occurring even if the user is not paying any attention. This is because there is a $1/4$ chance of a user randomly striking a particular color and a $n{*}2^{n-1}$ chance that two OOB strings, in the presence of an attack, mismatch in just one bit. Note that while it is theoretically possible for a player to complete pairing with Alice Says by making random color quadrant choices, this would take prohibitively long to achieve and can be ruled out in practice.

III. Implementation of Alice Says

To develop Alice Says, we preserved the popular aspects of Simon while updating it to a two player mobile device setting. Its user interface is dominated by four large color buttons as was the case with its ancestor. Also intended to mimic the original was the association of a unique tone with each of these keys. A critical aspect of the original game's appeal was the fact that these sounds were designed to be harmonic irrespective of the order in which they were played. This is important as the game play involves striking the inputs in a random order. We thus tried to mimic the original game's sounds by assigning an A note to the first input, an A note one octave higher to the second, a D note that is a perfect fourth above the initial A note to the third button, and finally a G note that was a perfect fourth higher than the D to the last key [41].

We utilized two Nokia N97 mobile phones to realize our Alice Says prototype. These devices support the Java Platform, Micro Edition (Java ME) environment, which we designed our code in. We crafted a user interface that was as intuitive and user friendly as possible. We utilized the lower level APIs provided by the Nokia N97 SDK. The device accepting user input kept track of whether or not an error occurred and automatically adjusted the pattern length accordingly. The N97s were well suited for exploring the usability of device pairing. They featured a resistive touch screen and stereo speakers, for example.

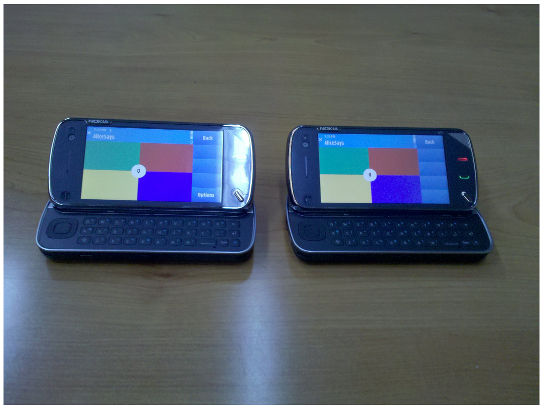

Fig. 1. Alice Says Game Setup

IV. Other Applications of Games for Security

The Tom Sawyer effect can be applied to various security tasks in order to enhance usability, especially in the context of wireless and mobile computing. Halprin and Naor have already applied this principle to the dilemma of random number generation to great effect [42]. The following section discusses several other security challenges that could potentially benefit from a game inspired approach.

A. User Authentication

Another area where it may be fruitful to apply this concept is that of authentication. For example, the usability of current mobile phone password managers, such as KeePassMobile [10], can be improved. These applications suffer from poor usability by requiring that users manually transfer passwords from a phone screen to the authentication terminal. Games similar to Alice Says can be adopted to address this. Furthermore, games may be designed to supplement the security and privacy of "something you have" authentication techniques by having users play a short movement game in order to unlock their access tokens, such as RFID tags. This idea is similar in spirit to the recently proposed Secret Handshakes scheme [1] but is aimed at providing an enhanced level of usability.

1) Graphical Passwords: Graphical passwords are authentication systems that replace standard alphanumeric secrets with a sequence of choices related to images. Although a preliminary evaluation of graphical passwords suggests that they can be more difficult to break compared to text passwords, they are still subject to efficiency, memorability, and predictability problems. To improve the usability and efficiency of existing graphical passwords, we propose that scoring functionality be added based on the speed and accuracy with which the password is recalled. For example, if a user spends less time and commits fewer errors while recognizing his or her password, or is able to reconstruct this password very closely to the one originally selected, he or she can be rewarded with a high score. A user who chooses to work with a random password rather than a weaker self-selected password can be provided with a higher score as well. This will provide incentive for users to behave more securely.

The usability of existing graphical passwords can also be enhanced by tying such a scheme to a natural physical context. This is because evidence indicates that context can improve memorability [53]. With image recognition based passwords, for example, an animation can be constructed whereby the user is placed in an art gallery displaying all the possible challenge images. Users can move around the gallery and identify the images that were originally selected as their password. This introduces a playful environment that provides an aesthetic experience every time the user attempts to login, and may help improve the system's usability and recognition rate.

Alternatively, users can be shown a shopping mall or grocery store. As they walk down its aisles, users would identify their password by selecting objects, such as clothing or grocery products, among a large quantity of other objects. This game can also be combined with a "Hidden Objects Game" in which players are required to identify items from a given list that are hidden within an image. The idea behind this technique is that it will be significantly easier for users to recognize their previously memorized objects than it will be for an adversary, who must search for unknown hidden items.

A different potential graphical password type of game can be constructed based on mazes. Unlike traditional mazes, which have one entry and exit point, there will be multiple pairs of these points. Users will travel from one opening point to another. They may also traverse back and forth between any two entry or exit points and choose circuitous routes. The selected route will become a user's password. Once the password has been selected and memorized, a user will be asked to enter the same route in order to authenticate. To aid in password recall, users may be allowed to customize the maze by adding pictures at different points in the maze, such as faces or landmarks. Research has shown that such personalization will help users be able to memorize longer and more complex paths [56].

2) Biometrics: Biometric authentication based on intrinsic user characteristics could also potentially be improved by integrating game like constructs. This is because many existing biometric designs fail to take the role of users into account.

This causes them to suffer from severe usability issues. Many individuals consider biometrics to be invasive or "scary." Because they are hard to use and understand, most users find biometrics difficult to accept psychologically [5], [14]. For instance, users often express reluctance at using public fingerprint scanning devices. Notably, they have shown concerns that fingerprint scanners present at border checkpoints could facilitate the spread of contagious diseases [36].

To overcome the limitations of current biometric systems, we propose a novel form of biometrics based on games. Whenever users wish to authenticate to a device or service, rather than requiring them to submit an awkward or invasive physiological scan, game biometrics would simply request that they play a video game for a certain interval of time. Users will be uniquely identified by extracting features from their game play habits and tendencies.

Some examples of potentially applicable game play characteristics include active and idle time within a game [26] and typing habits [12]. A variety of mouse measurements are also pertinent, such as mean click length, average click rate, as well as the distance, speed, and angle at which the mouse is moved [43], [34]. This type of data is already being collected and mined by video game companies for marketing [13], [18] and quality control purposes, which supports the plausibility of gaming biometrics as an authentication solution deployable in the near future. As a prime example, the Valve Corporation collects extensive information on users through its Steam platform and publishes real time statistics for its most popular games online [55].

Several research challenges will need to be overcome in order to achieve this goal. First and foremost, a game that is suitable for use as an authentication measure must be designed or identified. While previous work [43], [26] indicates that it may be possible to extract potentially useful features from any type of computer game, it remains an open question as to whether or not certain games perform better than others in this regard. A primary research objective is therefore to isolate and identify what characteristics of games are conducive to identifying an individual in as efficient and robust manner as possible.

Furthermore, the prior research focuses on the use of a single gameplay metric as a feature of interest. While elegant from a design perspective, the performance reported for these single attribute systems indicate that this technique is neither sufficiently efficient nor robust for use as a component of a usable authentication solution. Another core objective is therefore to find ways in which multiple game based biometric measures can be combined to identify users quickly and with enough certainty to provide ample security assurances. The possibility of enhancing identification performance via the application of multiple independent biometric characteristics is supported by previous work [57].

Finally, we point out that game based biometrics is a particularly promising and a natural solution for fall-back authentication [51]. Being infrequent, a fall-back can tolerate increased delays that game biometrics may exhibit.

B. *Human Identification*

An additional way in which games may be of use is as a replacement for CAPTCHA mechanisms. Currently, the most commonly encountered CAPTCHAs take the form of a garbled string of words or characters that a user must type. Unfortunately, existing CAPTCHA technology suffers from several flaws. The same distortions that are used to hide the underlying content of a puzzle from computers can also negatively impact human usability [24].

Moreover, researchers have had success in designing algorithms for breaking many existing CAPTCHA systems. Programs for detecting individual characters have surpassed their human counterparts in ability [25]. Algorithms have recently been designed that can achieve character segmentation with a 90% success rate [23]. Attacks have also been conceived where challenges are relayed to users on different web sites in order to solve them [8]. Real world attacks have also been launched against the CAPTCHA technology used by Internet giants including PayPal [27], Microsoft [19], and Google [48].

CAPTCHAs were never intended to be coupled with a single stagnant artificial intelligence problem. Their designers hoped that in practice their use would foster research on solving the difficult problems on which they are based [31], at which point CAPTCHA users would presumably migrate to a different form of challenge. Based on the flaws and attacks detailed above, we strongly feel that the time has arrived for a new type of CAPTCHA.

As a substitute, we propose the use of game based CAPTCHAs. In order to prove to remote servers that a human user is really behind a given request, users will be challenged by playing a game that is relatively easy for humans to complete but difficult for computers. Using computer games to construct CAPTCHAs would address all the aforementioned concerns. Their usability would certainly be higher due to their fun context and lack of direct distortion. Since mobile games are already among the most popular, CAPTCHAs can be custom tailored to portable devices where text solutions are impractical. Finally, the time sensitive nature of many games would cause their CAPTCHA derivatives to be resilient to relay attacks.

Many research challenges exist that will need to be overcome to make gaming CAPTCHAs a reality. Like other playful security solutions, a primary task is to identify a suitable game. In the context of CAPTCHAs, desirable games must meet the stringent criteria of being easier to play for humans than they are for computing devices. Since the artificial intelligence community has developed algorithms for many classes of games, this is more easily said than done.

V. Related Work

A. *Prior Pairing Methods*

In this subsection, we present prior pairing methods and their weaknesses. Stajano and Anderson [17] proposed establishing a shared secret between two devices using a link created through a physical cable. However, in many settings establishing physical contact might not be possible; the devices might not have common interfaces or it might be too cumbersome to carry the cables. Balfanz, et al. [11] extended this approach through the use of an infrared channel. Here devices exchange their public keys over a wireless channel followed by exchanging hashes of their respective public keys via infrared. The main drawback of this technique is that it is only applicable to devices that are equipped with infrared transceivers. Moreover, the infrared channel is not easily perceptible by human users.

Another approach is to perform the key exchange over a wireless channel and authenticate it by requiring that users manually compare the established secret on both devices. Since manually comparing the established secret is cumbersome for users, methods were designed to simplify it. These include Goldberg's Snowflake mechanism [20] and the Random Arts visual hash [6] by Perrig and Song. These methods require high resolution displays and are thus only applicable to a limited number of devices such as laptops.

Based on the pairing protocol of Balfanz et al. [11], McCune et al. proposed the "Seeing-is-Believing" (SiB) method [22]. SiB involves establishing two unidirectional visual channels; one device encodes data into a two-dimensional barcode and the other device reads it using a camera. Since it requires both devices to have cameras, it is only suitable for pairing devices such as camera phones. Moreover, a recent study [4] shows that users may not be comfortable handling cameras.

Uzun et al. [15] carried out a comparative usability study of pairing methods. They consider scenarios where devices have at least 4-digit displays. In what they call the "Compare-and-Confirm" approach, users read and compare SAS data. The "Select-and-Confirm" approach, on the other hand, requires users to select a string on one device that matches with a string on the other device. The third approach, "Copy-and-Confirm," is a DC method. It requires that users read data from one device and input it on another.

Recent papers have focused upon pairing devices which lack good interfaces. Access points and headsets are examples of this kind of device. These constraint oriented pairing solutions include the BEDA method [9], which requires that users transfer SAS strings from one device to another using button presses. In [44], [54], Saxena et al. presented a similar pairing method that is universally applicable. It involves users comparing very simple audiovisual patterns such as "beeping" and "blinking." The approach of [44] was extended by making use of an auxiliary device such as a smartphone [38].

B. Computer Games and Security

Our work was inspired by that of Halprin and Naor [42] who proposed the use of games to address random number generation. Computers often use inputs from users as an entropy source. Unfortunately, when asked to cooperate in this endeavor, human users tend to perform poorly by interacting with the machine in a predictable fashion. This is because humans are notoriously bad at behaving randomly. When asked to construct random sequences, people's outputs are riddled with biases. Interestingly, when placed in competitive situations humans demonstrate a heightened aptitude for behaving randomly [42]. Therefore, gains are noted when users are asked to participate in a game that forces them to behave randomly and then harvests entropy from their actions.

The pairing mechanism that we present in this paper is an example of a Game with a Purpose (GWAP) as conceptualized by von Ahn [30]. This is because it is not simply a game for its own sake, but rather a form of entertainment that simultaneously achieves a well-defined objective. The reCAPTCHA [32] project of von Ahn et al. is also related. It does not involve any entertainment but also fools users into doing work beyond what they may realize. reCAPTCHA not only serves the purpose of a reverse Turing test but also utilizes the responses it receives to aid in the digitization of words. A crucial difference between our proposals and von Ahn's is that our games are meant to accomplish human work as part of the underlying security mechanism itself rather than solving an offline problem such as labeling images [29]. Also relevant to our proposal is an independent line of research on offline security education and training through playful approaches [39], [2], [49].

VI. Conclusions

In this paper, we contributed "Alice Says," a novel system for pairing devices via a game. More broadly, we considered the problem of designing pairing methods that in some way incentivize users to put forth more effect and correctly take part in the pairing process, thus providing improved security as well as enhancing the overall user experience. We dubbed this the Tom Sawyer Effect. To this end, we proposed a general research direction of applying computer games to solve tricky issues in usable security. The incentive that we provide to users while they pair their devices is fun and entertainment. Since games are a popular form of entertainment, our hypothesis is that they may improve the security as well as usability of pairing and help solve the challenges outlined above. As part of our future work, we will conduct a formal usability study of Alice Says and contrast its usability and security with that of traditional pairing mechanisms.

At a higher level, we believe that our work opens a new area of research in usable security where security tasks are presented in a playful way by making use of computer games. Designing games that are optimal in terms of speed, error tolerance, and psychological acceptability for a given application remains an open research challenge. We hope that our work will motivate other researchers and practitioners to come up with novel games for addressing lingering problems in usable security.

References

[1] A. Czeskis and K. Koscher and J. Smith and T. Kohno. RFIDs and Secret Handshakes: Defending Against Ghost-and-Leech Attacks and Unauthorized Reads with Context-Aware Communications. In *Conference on Computer and Communications Security*, 2008.

[2] A. Forget and S. Chiasson and R. Biddle. Lessons from Brain Age on Password Memorability. In *Conference on Future Play*, 2008.

[3] A. Kumar and N. Saxena and E. Uzun. Alice Meets Bob: A Comparative Usability Study of Wireless Device Pairing Methods for a “Two-User” Setting. In *Computer Research Repository*, 2009.

[4] A. Kumar and N. Saxena and G. Tsudik and E. Uzun. Caveat Emptor: A Comparative Study of Secure Device Pairing Methods. In *Conference on Pervasive Computing and Communications*, 2009.

[5] A. Patrick. Usability and Acceptability of Biometric Security Systems. In *Financial Cryptography*, Available at: http://www.andrewpatrick.ca/biometrics/NATO-BiometricsAbstract.pdf, 2004.

[6] A. Perrig and D. Song. Hash Visualization: a New Technique to improve Real-World Security. In *Workshop on Cryptographic Techniques and E-Commerce*, 1999.

[7] BoardGameGeek. Simon. Available at http://www.boardgamegeek.com/boardgame/5749/simon, 1978.

[8] C. Doctorow. Solving and Creating CAPTCHAs with Free Porn. In *Boing Boing*, Available at: http://www.boingboing.net/2004/01/27/solving_and_creating.html, 2004.

[9] C. Soriente and G. Tsudik and E. Uzun. BEDA: Button-Enabled Device Association. In *Workshop on Security for Spontaneous Interaction*, 2007.

[10] C. Sperle. KeePassMobile. Available at http://www.keepassmobile.com, 2010.

[11] D. Balfanz and D. Smetters and P. Stewart and H. Wong. Talking To Strangers: Authentication in Ad-Hoc Wireless Networks. In *Network and Distributed System Security Symposium*, 2002.

[12] E. Lau and X. Liu and C. Xiao and X. Yu. Enhanced User Authentication Through Keystroke Biometrics. Available at: http://people.csail.mit.edu/edmond/projects/keystroke/keystroke-biometrics.pdf, 2004.

[13] E. Schonfeld. Turiya Media: Data Mining Social Games To Find The Most Valuable Players. In *Tech Crunch*, Available at: http://techcrunch.com/2010/04/06/turiya-media-games, 2010.

[14] E. Schultz and R. Proctor and M. Lien and G. Salvendy. Usability and Security: An Appraisal of Usability Issues in Information Security Methods. *Computers and Security*, 20(7), 2001.

[15] E. Uzun and K. Karvonen and N. Asokan. Usability Analysis of Secure Pairing Methods. In *Usable Security*, 2007.

[16] Entertainment Software Association. Essential Facts About the Computer and Video Game Industry. Available at http://www.theesa.com/facts/pdfs/ESA_EF_2009.pdf, 2009.

[17] F. Stajano and R. Anderson. The Resurrecting Duckling: Security Issues for Ad-hoc Wireless Networks. In *Workshop on Security Protocols*, 1999.

[18] G. Goldwasser. Collecting Data (and Strangers) Online. In *The Faster Times*, Available at: http://thefastertimes.com/videogames/2010/02/21/collecting-data-and-strangers-online, 2010.

[19] G. Keizer. Spammers’ Bot Cracks Microsoft’s CAPTCHA . In *Computer World*, Available at: http://www.computerworld.com/s/article/9061558/Spammers_bot_cracks_Microsoft_s_CAPTCHA_, 2008.

[20] I. Goldberg. Visual Key Fingerprint Code. Available at http://www.cs.berkeley.edu/iang/visprint.c, 1996.

[21] I. Ion, M. Langheinrich, P. Kumaraguru, and S. Capkun. Influence of user perception, security needs, and social factors on device pairing method choices. In *SOUPS*, 2010.

[22] J. McCune and A. Perrig and M. Reiter. Seeing-Is-Believing: Using Camera Phones for Human-Verifiable Authentication. In *Symposium on Security and Privacy*, 2005.

[23] J. Yan, A. Ahmad. A Low-cost Attack on a Microsoft CAPTCHA. In *Conference on Computer and Communications Security*, 2008.

[24] J. Yan, A. Ahmad. Usability of CAPTCHAs Or usability issues in CAPTCHA design. In *Symposium On Usable Privacy and Security*, 2008.

[25] K. Chellapilla and K. Larson and P. Simard and M. Czerwinski. Computers beat Humans at Single Character Recognition in Reading based Human Interaction Proofs (HIPs). In *Conference on Email and Anti-Spam*, 2005.

[26] K. Chen and L. Hong. User Identification based on Game-Play Activity Patterns. In *Workshop on Network and Systems Support for Games*, 2007.

[27] K. Kluever. Breaking the PayPal.com CAPTCHA. Available at: http://www.kloover.com/2008/05/12/breaking-the-paypalcom-captcha, 2008.

[28] A. Kobsa, R. Sonawalla, G. Tsudik, E. Uzun, and Y. Wang. Serial hook-ups: A comparative usability study of secure device pairing methods. In *SOUPS: Symposium on Usable Privacy and Security*, 2009.

[29] L. von Ahn. The ESP Game. Available at http://www.gwap.com/gwap/gamesPreview/espgame/.

[30] L. von Ahn. Games with a Purpose. In *Computer Magazine*, 2006.

[31] L. von Ahn and M. Blum and N. Hopper and J. Langford. CAPTCHA: Using Hard AI Problems For Security. In *Advances in Cryptology - Eurocrypt*, 2003.

[32] L. von Ahn, B. Maurer, C. McMillen, D. Abraham, M. Blum. reCAPTCHA: Human-Based Character Recognition via Web Security Measures. In *Science Magazine*, 2008.

[33] M. Cagalj and S. Capkun and J. Hubaux. Key Agreement in Peer-to-Peer Wireless Networks. In *Proceedings of the IEEE*, 2006.

[34] M. Pusara and C. Brodley. User Re-Authentication via Mouse Movements. In *Workshop on Visualization and Data Mining for Computer Security*, 2004.

[35] M. Twain. The Adventures of Tom Sawyer. 1876.

[36] B. Marshall. Does Fingerprint ID at Entry Portals Spread Swine Flu? Available at http://barryjmarshall.blogspot.com/2009/06/does-fingerprint-id-at-entry-portals.html.

[37] N. Saxena and M. Uddin. Secure Pairing of “Interface-Constrained” Devices Resistant against Rushing User Behavior. In *Applied Cryptography and Network Security*, 2009.

[38] N. Saxena and M. Uddin and J. Voris. Universal Device Pairing Using an Auxiliary Device. In *Symposium on Usable Privacy and Security*, 2008.

[39] P. Kumaraguru and S. Sheng and A. Acquisti and L. Cranor and J. Hong. Teaching Johnny Not to Fall for Phish. In *Transactions on Internet Technology*, 2010.

[40] Pew Internet and American Life Project. Teens, Video Games and Civics. Available at http://www.pewinternet.org/Reports/2008/Teens-Video-Games-and-Civics.aspx, 2008.

[41] R. Baer. The SIMON Story. Available at http://www.dieterkoenig.at/ccc/english/se_story_simon.htm, 2003.

[42] R. Halprin and M. Naor. Games for Extracting Randomness. In *Symposium on Usable Privacy and Security*, 2009.

[43] R. Kaminsky and M. Enev and E. Andersen. Identifying Game Players with Mouse Biometrics. Available at: http://abstract.cs.washington.edu/∼miro/docs/mouse_ID.pdf, 2008.

[44] R. Prasad and N. Saxena. Efficient Device Pairing using “Human-Comparable” Synchronized Audiovisual Patterns. In *Applied Cryptography and Network Security*, 2008.

[45] R. Ryan and E. Deci. Self-determination theory and the facilitation of intrinsic motivation, social development, and well-being. *American Psychologist*, 55(1):68–78, 2000.

[46] S. Laur and K. Nyberg. Efficient Mutual Data Authentication Using Manually Authenticated Strings. In *Conference on Cryptology and Network Security*, 2006.

[47] S. Pasini and S. Vaudenay. SAS-Based Authenticated Key Agreement. In *Conference on Theory and Practice of Public-Key Cryptography*, 2006.

[48] S. Prasad. Googles CAPTCHA Busted in Recent Spammer Tactics. Available at: http://securitylabs.websense.com/content/Blogs/2919.aspx, 2008.

[49] S. Srikwan and M. Jakobsson. Using Cartoons to Teach Internet Security. In *Cryptologia*, 2008.

[50] S. Vaudenay. Secure Communications over Insecure Channels Based on Short Authenticated Strings. In *Crypto*, 2005.

[51] S. E. Schechter and R. W. Reeder. 1 + 1 = you: measuring the comprehensibility of metaphors for configuring backup authentication. In *Proceedings of the 5th Symposium on Usable Privacy and Security (SOUPS)*, 2009).

[52] Shacknews. NPD: 72% of U.S. Population Played Games in 2007. Available at http://www.shacknews.com/onearticle.x/52025.

[53] M. A. Upal. Role of context in memorability of intuitive and counter-intuitive concepts. In *Proceedings of the 27th Annual Meeting of the Cognitive Science Society*, 2005.

[54] V. Roth and W. Polak and E. Rieffel and T. Turner. Simple and Effective Defense Against Evil Twin Access Points. In *Conference on Wireless Network Security*, 2008.

[55] Valve Corporation. Steam: Game and Player Statistics. Available at: http://store.steampowered.com/stats, 2010.

[56] N. G. Vinson. Design guidelines for landmarks to support navigation in virtual environments. In *CHI ’99: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 278–285, 1999.

[57] Y. Wang and T. Tan and A. Jain. Combining Face and Iris Biometrics for Identity Verification. In *Audio- and Video-Based Biometric Person Authentication Conference*, 2003.