Exploring Extrinsic Motivation for Better Security: A Usability Study of Scoring-Enhanced Device Pairing

Alexander Gallego[1], Nitesh Saxena[2], and Jonathan Voris[3]

[1] Hashable, Inc.
[2] University of Alabama at Birmingham
[3] Columbia University

Abstract. We explore the use of extrinsic motivation to improve the state of user-centered security mechanisms. Specifically, we study applications of visual scores as user incentives in the context of secure device pairing. We develop a scoring functionality that can be integrated with traditional pairing approaches which involve the manual comparison of numbers. We then report on a usability study that we performed to evaluate the effect of scoring on the performance of users in comparison operations. Our results demonstrate that individuals are likely to commit fewer errors and show more acceptance when working with the scoring based pairing approach.

1 Introduction

Short range wireless communication is becoming increasingly popular and promises to remain so. This popularity unfortunately brings about various security risks. Wireless channels are easy to eavesdrop upon and manipulate. A fundamental security objective is therefore to secure them. In this paper, the term "pairing" refers to the process of bootstrapping secure communication between two wireless devices in a way that is resistant to eavesdropping and man-in-the middle attacks. A promising research direction towards solving the pairing dilemma is to leverage an Out-Of-Band (OOB) channel that is governed by human users. Examples of OOB channels include audio, visual, and tactile channels. 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. Due to this property, OOB communication provides authentication and integrity, unlike radio communication.

The usability of an OOB-based pairing method is very important. Since OOB channels typically have low bandwidth, the shorter the data that a pairing method needs to transmit over these channels, the better its usability. A recent innovation to this end is the development of Short Authenticated String (SAS) based protocols (e.g.,[8]) that reduce the length of data transmitted over OOB channels. A variety of pairing methods based on visual, audio, tactile, and infrared OOB channels have been proposed based on these protocols [4].

Unfortunately, device pairing has not been addressed in a fully desirable manner. Prior work on pairing raises several fundamental usability and security related concerns and research challenges. One of the most prominent of these is that the amount of

security that a SAS based pairing method provides is dependent on the size of strings that it uses; a k-bit SAS limits the probability of a successful attack to 2^{-k}. Existing pairing methods use short strings in their SAS protocols. Typically, these values are only 15 bits long. SASs of this size are not large enough to provide sufficient security for certain applications. Unfortunately, increasing the length of a pairing system's SASs causes pairing to take longer to complete. This, in turn, leads to poor usability and can also have an impact on security.

Further, even while using short OOB strings, several comparison-based pairing methods do not offer the theoretical level of security guaranteed by their underlying protocols, as demonstrated in [4]. This is due to the potential for human errors in these protocols. Such errors can be of two forms: *fatal* and *safe* [1]. Fatal errors (also known as false positives or "Type I" errors) occur when a user accepts a pairing instance, although the OOB strings on the two devices did not match, which may lead to a man-in-the-middle attack. Safe errors (or false negatives or "Type II" errors), on the other hand, occur when a user rejects a pairing instance even when the OOB strings on the two devices match. Such errors undermine the usability of pairing, but can also have an indirect impact on security; a failed pairing necessitates repetition, which may lead to user annoyance and translate into attacks eventually.

Our overall solution to these challenges is to make use of a reward system as a way of measuring and improving users' performance during the pairing process. The system draws from motivation research in human psychology. Motivation can be intrinsic or extrinsic [7]. Intrinsic motivation emanates from oneself, i.e., when one is inherently interested in a task. Clearly, human users lack intrinsic motivation for security tasks. Extrinsic motivation, on the other hand, relates to how users can be externally motivated for non-intrinsically interesting tasks [6], and is directly applicable to security tasks, such as pairing. In this paper, we consider a very simple form of extrinsic motivation, visual scores, to improve the security and usability of secure device pairing.

The scoring-based pairing mechanism that we proposed in this paper is an example of a "Game with a Purpose" (GWAP) [5] A crucial difference between our proposals and existing GWAPs is that our games are meant to accomplish human work *as part of* the underlying security mechanism itself, rather than solving an "offline" problem, e.g., labeling of images.

Our contributions in this paper are as follows: (1) We develop a scoring functionality that can be integrated with traditional pairing approaches that involve the manual comparison of numbers displayed on two devices; (2) We report on a between-subjects study to evaluate the effect of scoring on the performance of numeric comparisons.

2 Threat Model for Device Pairing

We summarize the threat model for device pairing based on OOB communication [8]. In this model, wireless devices can establish two types of communication mediums. The first is a traditional wireless radio channel, which is characterized by a large bandwidth capacity. We imagine a worst case scenario in which an adversary has full control over the wireless channel. The second medium comprises 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 users to verify transmission sources themselves.

3 Design of Pairing Methods

In this section, we present the design and implementation of two pairing approaches. One of these, referred to as Plain Comparison, is a variant of the traditional approach involving numeric comparisons. The other, referred to as Scored Comparison, integrates a scoring and grading functionality with plain numeric comparisons. Traditional pairing approaches involving numeric comparisons employ the comparison of a single 5 digit number displayed on the screen of two devices. For better security, one possibility is to employ longer numbers; however, longer numbers become harder for the users to compare. In our Plain Comparison method, we display four 5 digit numbers on four seperate quadrants of the screen, implying four times the security provided by the traditional method.

In order to add a scoring functionality to the Plain Comparison method, we needed to incorporate some comparison instances that could be used to calculate the score based on the performance of the user. Note that the "pairing instances," that is, the numbers resulting from the OOB strings generated by the pairing protocol, cannot be used for this purpose because the devices themselves are not aware whether these instances are matching or non-matching. To this end, we create some dummy "scoring instances"; whether these are matching or non-matching is pre-determined and known to the devices. A score is calculated based on how accurately users compare the scoring instances. The scoring instances are required to calculate a user's score, since unlike the pairing instances, the device pair shares knowledge regarding whether or not they match.

3.1 Scored Comparison Method

The scored pairing GUI consists of colored quadrants which are used as a visual mnemonic technique to help users associate the numbers displayed within. We also used another type of mnemonic technique called chunking, where information is broken up into more manageable sizes to allow short-term memory to operate more efficiently. This pairing method consisted of 2 rounds, and therefore 8 comparison instances. 4 instances were shown per round; this was done to keep the design similar to the Plain Comparison method. 4 instances were used for pairing and 4 instances were used for scoring.

Initialization Step In order for our pairing game to calculate a score, the two devices that are being paired must have a mechanism for determining which displayed values match and which differ. In order to accomplish this, the computers can first agree upon a seed value. There are several ways in which the two devices can settle on a seed. Two possibilities are the use of a publicly known value that is controlled by a service provider or synchronized system times. We employed the latter approach in our implementation. After a seed has been agreed upon, it is used to populate two binary arrays. These arrays are used to generate the numeric values that will be displayed to users during the pairing process. Note that our prototype implementation also utilizes the same seed in order to generate pairing instances. In a real world implementation, however, pairing instances would be generated from OOB strings that are created as a result of the underlying SAS pairing protocol.

Pairing Step The pairing step is the iterative game loop of our system. To start, all four color quadrants of the pairing game's display will be empty. The basic idea of

this portion of the pairing procedure is to populate these quadrants with random five digit integer values. First, two of the quadrants are filled with pairing instances; these will be used to determine the success or failure of the pairing operation. Next, the remaining half of the display is populated with scoring instances; whether or not a user can successfully detect matches in these values will determine his or her score. In case the pairing failed, a score is still displayed but the user will be asked to play another round until pairing succeeds. Matching values are generated by hashing a portion of the pertinent bit string, while mismatches are generated pseudorandomly.

Score Calculation The method's state was recorded with each press of the screen (i.e., user's input) and at the end of every round. This was an important part of our score calculation. The score had two major components: performance points based on accuracy of comparison and free give-away points for trying. First, we compared the screen presses array with that of the non-matching instances. If the instance was a scoring instance and it was a correct identification of a non-matching number, we gave the user 5 points; otherwise, we gave the user 1 point. The last free give-away point was added to the score as part of our extrinsically motivated design, disregarding whether it was a pairing instance or a failed scoring instance. The give away points were the user's emotional reward, and to prevent the user from feeling bad about his performance, which could potentially affect future pairing attempts. At this point, we give the user another reward for his or her performance. If the user scored a perfect score or 24 points, we displayed "You Performed Excellent! - Great job". If the user scored within 3/4 but less than perfect, we displayed "You Performed Well! - Good work". Otherwise, we displayed "You Performed Fair".

3.2 Plain Comparison Method

In order to have a meaningful comparison of our system, we developed a non-scoring version of the implementation which we refer to as the "Plain Comparison" method. The non-scoring version differed from the scoring version in significant ways. First, as the name indicates there was no scoring involved. Second, this version of the implementation had exactly half of the comparisons the scoring version had. This was due to the fact that we added artificial sets of numbers to the scoring version, in order to account for the score. In other words, in the Plain Comparison method, there were 4 comparisons of 5 digit numbers, one number per quadrant. Since both the Plain and Scored Comparison techniques used the same amount of scoring instances, they each provided the same theoretical level of security.

4 Usability Evaluation

4.1 Testing Framework

To implement our pairing mechanisms as part of our study, we used two Nokia N97 phones. In order to ensure that the data we collected from our test subjects was as meaningful as possible, we gave users hands on experience with an implementation of our prototype deployed on these devices. Although we were attempting to simulate as realistic of a pairing scenario as possible, we still desired to eliminate unnecessary complexity from our testing system. To this end, we removed the traditional wireless

link between the two phones that were used throughout our study. Rather than transmitting key information over this channel and using it to derive OOB pairing values, OOB strings were generated by the N97s on the fly using a pseudorandom number generator.

Comparison pairing method were asked two additional questions: (5) I was annoyed with the fact that the score was shown at the end of the method, and (6) I would prefer to see the score after every comparison rather than at the end of the method.

To capture the efficiency and efficacy of our prototypes, user actions were logged. We captured both the time taken to complete pairing tasks as well as any errors committed along the way. We provided users with a post-conditional questionnaire in order to gauge the opinions users had about our pairing system. Rather than developing our own survey instrument from scratch, we made use of the System Usability Scale (SUS) [2]. SUS is a technique for measuring the usability of a system in an efficient and reliable manner [2]. Our survey consisted of the questions that comprise SUS along with demographic questions and the following queries: (1) The method was enjoyable, (2) The method took a long time, (3) I would like to pair with another user's devices by making use of this method, and (4) I perceive this method to be secure. Users who evaluated the Scored Comparison pairing method were asked two additional questions: (5) I was annoyed with the fact that the score was shown at the end of the method, and (6) I would prefer to see the score after every comparison rather than at the end of the method.

4.2 Participant Information

We decided to conduct a between-subjects study with two subsets of 20 volunteers.We recruited these individuals from students working and studying at our college campus. We spread awareness of the study through emails and by signing people up face to face. Movie theater coupons were provided to participants as compensation for their assistance. We aggregated the following data about our users' backgrounds: age, gender, education level, experience pairing wireless devices, and experience with video games (since scoring is a common element of most games).

Table 1 presents demographic information about our study volunteers. The two sets of participants had similar levels of experience with wireless technology. 61.9%, of scored users, stated that they had performed wireless device pairing before, while just over half, 52.4% of plain users said they had done so. All users unanimously responded that they had played video games in the past.

4.3 Experimental Design

To begin a test with a subject, the administrator navigated to the Scored or Plain Comparison entry in the devices' application menus. The two phones were then turned over to the testers for the remaining duration of the study. Before allowing users to begin their pairing trials, instructions on how to utilize the Scored or Plain Comparison pairing application were displayed to them. After making their way through the instructions, a "Play" button appeared. This initiated the numeric comparisons. Users then proceeded to use either the Scored or Plain Comparison method. The act of pairing was repeated five times in order to provide subjects with ample experience. When finished, the "Play" button that initialized the pairing procedure was replaced with an "Exit" button. After getting several opportunities to perform simulated pairing with one of our two prototypes, each volunteer was presented with a post-conditional questionnaire.

Demographic Information	Plain Comparison	Scored Comparison
AGE		
17 - 25	52.4	52.4
26 - 29	33.3	38.1
30 - 40	14.3	9.5
GENDER		
Male	42.9	61.9
Female	57.1	38.1
EDUCATION		
School graduate	14.2	19.0
Bachelor degree	28.6	38.1
Masters degree	52.4	33.3
Doctorate degree	4.8	9.5

Table 1. Demographics of participants

	Scored Comparison		Plain Comparison	
Execution Time	Average	Standard Deviation	Average	Standard Deviation
	22.0 sec	11.2 sec	16.6 sec	13.4 sec
Error Rates	Safe	Fatal	Safe	Fatal
	1.8%	2.8%	6.5%	4.8%
SUS Scores	Average	Standard Deviation	Average	Standard Deviation
	74.3	11.7	69.2	15.2

Table 2. Summary of Experimental Results (The execution time and error rates are averaged across the two test devices.)

This included the demographic queries listed above followed by a series of five point Likert items. There were 14 of these questions for the plain survey and sixteen for the scoring version. The first ten of these comprised the System Usability Scale [2] with the slight modification that our pairing implementation was referred to as a method as opposed to a system.

4.4 Experimental Results

We have summarized the main results of our study in Table 2. Each volunteer who tested the Plain or Scored Comparison prototype executed 5 pairing sessions, accounting for a total of 100 test cases. The average execution time for the Scored Comparison method was 22.0 seconds for one device and 22.1 seconds for the other with standard deviations of 11.2 and 11.3 respectively. The Plain Comparison application completed in 16.7 seconds on average and a 13.6 standard deviation for the first device and 16.4 seconds with a standard deviation of 13.3 on the second. The small variation in execution time between the two mobile devices is attributable to the brief delay in pressing the "Finish" buttons on the two devices.

For the method of Scored Comparison, users were presented with 2 matching values and 6 mismatching values per pairing session for a total of 200 matches and 600 mismatched numbers over all 100 test cases. Out of these, 3 safe errors and 11 fatal errors were committed on one device. This yields a safe error rate of $3/200 = 1.5\%$ and a

fatal error rate of $11/600 = 1.8\%$. On the other device, 4 safe errors and 23 fatal errors were committed, producing a $4/200 = 2.2\%$ safe error rate and a $23/600 = 3.8\%$ fatal error rate. The average of the error rates on the two devices is 1.8% safe and 2.8% fatal.

With respect to Plain Comparison, each pairing attempt was comprised of one matching entry and 3 mismatching entries, resulting in a total of 100 matches and 300 mismatches over the 100 test cases completed by our volunteers. On one device, 6 safe errors and 18 fatal errors were committed, while users committed 7 safe and 11 fatal errors on the other device. This results in a safe error rate of $6/100 = 6.0\%$ and a fatal error rate of $18/300 = 6.0\%$ for the first device; and a safe error rate of $7/100 = 7.0\%$ and a fatal error rate of $11/300 = 3.7\%$ for the second device. The average error rate across both devices is 6.5% and 4.8% for safe and fatal errors respectively.

On average, users assigned the Scored Comparison technique a 74.3 on the SUS scale with a standard deviation of 11.7. The Plain Comparison method was given a SUS rank of 69.2 and a standard deviation of 15.2. Test subjects who utilized the Scored Comparison method responded with a 3.5 on average when asked if they felt that the pairing technique they used was enjoyable, while those using Plain Comparison provided a slightly higher 3.7 response to this query on average. Scored users assigned an average score of 2.5 when asked if their pairing method took a long time. Volunteers who worked with the Plain Comparison method gave this question a 2.6 average. When asked if they would like to use the method, users of the Scored Comparison approach provided an average response of 3.6 and Plain Comparison volunteers gave an average rank of 3.7. When asked the question regarding their perception of the security of their given method, the average responses were 3.4 from scored users and 3.7 from users of the plain method. This was the last question posed to the Plain Comparison user group. Scored users provided 2.5 and 3.0 average responses when asked if they were annoyed about the score being withheld from them until after pairing and whether they would prefer to see the score after each comparison, respectively.

4.5 Interpretation and Analysis of Results

Looking at the average execution times, the Plain Comparison method was faster than the Scored Comparison variant by 5.2 seconds on one device and 5.7 seconds on the other (unpaired t-tests, however, did not indicate any statistically significant difference between the two methods). This is intuitive because the latter method involved more comparisons due to the presence of scoring instances. It must be noted, however, that users were required to make twice as many comparisons with the scored approach but time taken was not doubled. This suggests that providing users with a score which assesses their pairing performance could possibly encourage them to compare individual pairing values more rapidly.

The effect of scoring on our participants' pairing performance was one of the most interesting results of our study. The inclusion of a score had a dramatic impact on users' ability to successfully detect which pairing entries were matches and which were not. The average safe error rate across the two devices fell from 6.5% to 1.8% when a score was in use. Similarly, both devices' average fatal error rate dropped from 4.8% to 2.8%. The two proportions z-tests indicate that this difference is significant. In particular, the change was statistically significant at a 95% confidence level with p-value of 0.0327 for safe errors, and marginally significant at a 90% confidence level with p-value of 0.0874

for fatal errors. Since the only difference between the two techniques was the presence or absence of a score, it can be deduced that providing users with a numeric evaluation of their performance caused them to be more aware of their pairing decisions. Users made errors at a far lower rate as a result.

Average SUS scores traditionally fall between 60 and 70 [3]. Therefore both the Scored and Plain Comparison pairing methods can be considered rather positive. We ran an unpaired t-test on the SUS responses provided by the users of each method. This resulted in a low-end p-value of 0.12, but indicated a lack of statistical significance between the two sets of answers. It is also worth noting that the standard deviation of our participant's SUS responses fell by 23%, or 3.5 points, between the group that did not have scoring integrated into their pairing solution and the one that did. This positive result indicates that there was more agreement among the testers of the scored solution. On the other hand, the users of the unscored solution held more varied opinions regarding the usability of their system.

5 Conclusions

In this paper, we explored the use of visual scores as user incentivesin the context of the secure device pairing task. By means of scoring and grading users' performance, we hoped to extrinsically motivate users and thereby improve the security and usability of the pairing task. We developed a scoring functionality that can be integrated with traditional pairing approaches. We also reported on a between-subjects study which we performed to evaluate the effect of scoring on the performance of numeric comparisons. The results of our study demonstrate that users are likely to commit fewer errors when working with the pairing approach based on scored comparisons. We believe that our work opens up a new area of research in usable security where security tasks can be combined with perceptible scores and other rewards.

Acknowledgements: We thank FC'13 anonymous reviewers for their constructive feedback. This work is funded, in part, by NSF CNS-1255919 and a Google research award.

References

1. E. Uzun and K. Karvonen and N. Asokan. Usability Analysis of Secure Pairing Methods. In *Usable Security*, 2007.
2. J. Brooke. SUS - A quick and dirty usability scale. In *Usability Evaluation in Industry*, 1996.
3. J. Lewis and J. Sauro. The Factor Structure of the System Usability Scale. In *Conference on Human Centered Design*, 2009.
4. R. Kainda, I. Flechais, and A. W. Roscoe. Usability and security of out-of-band channels in secure device pairing protocols. In *SOUPS: Symposium on Usable Privacy and Security*, 2009.
5. L. von Ahn. Games with a Purpose. In *Computer Magazine*, 2006.
6. 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.
7. R. M. Ryan and E. L. Deci. Intrinsic and extrinsic motivations: Classic definitions and new directions. *Contemporary Educational Psychology*, 25(1):54–67, 2000.
8. S. Vaudenay. Secure Communications over Insecure Channels Based on Short Authenticated Strings. In *Crypto*, 2005.