Improving the Robustness of Wireless Device Pairing Using Hyphen-Delimited Numeric Comparison

Ambarish Karole and Nitesh Saxena
Polytechnic Institute of New York University
Six MetroTech Center, Brooklyn, NY 11201
akarol01@students.poly.edu, nsaxena@poly.edu

***Abstract*—The operation of achieving authenticated key agreement between two human-operated mobile devices over a short range wireless communication channel, such as Bluetooth or Wi-Fi, is known as "pairing." The devices being paired are ad hoc in nature, i.e., they can not be assumed to have a prior context (such as pre-shared secrets) or a common trusted on- or off-line authority. However, the devices can generally be connected using auxiliary physical channel(s) (such as audio or visual) that can be authenticated by the user(s) of the devices. These authenticatable channels can thus be used to form a basis for pairing.**

One of the simplest pairing methods requires user to compare short (typically 4 digit long) numbers displayed on two devices. Prior usability studies investigating the numeric comparison method indicate that although users hardly ever reject matching numbers on two devices, a *critical* task of detecting non-matching numbers (and thus potential *man-in-the-middle* attacks) can be error-prone. In this paper, we propose a very simple and an intuitive method of employing "*hyphen-delimited*" numbers in device pairing. Our usability studies and analysis of test results show that the proposed method improves the robustness as well as usability of pairing based on numeric comparison.

Keywords: Authentication, Key Agreement, Device Pairing, Usable Security

I. Introduction

Short and medium range wireless communication, based on technologies such as Bluetooth and Wi-Fi, is becoming increasingly prevalent and promises to remain so in the future. This surge in popularity brings about security risks. Wireless communication channels are easy to eavesdrop upon and manipulate. Therefore, a fundamental security objective is to secure these channels of communication. In this paper we will use the term "pairing" to refer to the operation of bootstrapping secure communication between two devices connected by a short- or medium-range wireless channel. Examples of pairing operations performed during the course of daily life include connecting two Bluetooth phones, a Wi-Fi laptop and an access point or a Bluetooth keyboard and desktop. Pairing would be easy to achieve if there existed a global infrastructure that enabled all personal wireless devices to share an on- or off-line trusted third party, certification authority, PKI, or any preconfigured secrets. Such a global infrastructure is nearly impossible to come by in practice, however, thereby making pairing an interesting and challenging real-world research problem (the pairing problem has been at the forefront of various recent standardization activities, see [13]).

Out-of-Band Channels: A well-established direction in pairing research is to use an auxiliary physically authenticatable channel, called an out-of-band (OOB) channel, which is governed by the human users who operate the devices. Examples of OOB channels include audio and visual channels. Unlike wireless channels, an adversary is assumed to be incapable of modifying messages transmitted on an OOB channel. It can eavesdrop on, delay, drop and replay them, however. A pairing scheme should therefore be secure against such an adversary. The usability of a pairing scheme based on OOB channels is clearly of the utmost importance. Since OOB channels typically have low bandwidth, the shorter the data that a pairing scheme needs to transmit over these channels, the better the scheme becomes in terms of usability.

Various pairing protocols have been proposed thus far. These protocols are generally based on bidirectional automated device-to-device (d2d) OOB channels. Such d2d channels require both devices to have transmitters and corresponding receivers. In settings where d2d channel(s) do not exist (i.e., when at least one device does not have a receiver) equivalent protocols can be based upon device-to-human (d2h) and human-to-device (h2d) channel(s) instead. Depending upon the protocol, only two d2h channels may be a sufficient replacement. This is the case when the user has to perform a very simple operation (such as "comparison") on the data received over these channels. Clearly, the usability of d2h and h2d channel establishment is even more critical than that of a d2d channel.

Short Authenticated Strings: Earlier pairing protocols required at least 80 bits of data to be transmitted over the OOB channels. The simplest protocol [1] involves the devices exchanging their public keys over the wireless channel and authenticating them by exchanging (at least 80-bit long) hashes corresponding to the public keys over the OOB channels. The more recent, so-called Short Authenticated Strings (SAS) based protocols [7][5] reduce the length of data transmitted over the OOB channels to approximately 15 bits.[1]

A number of pairing schemes that utilize various OOB channels have been proposed that are based on the protocols

[1]The concept of SAS-based authentication was first introduced by Cagalj et al. [16], followed by Vaudenay [15]. MANA protocols [2] addressed a similar problem.

listed above. The simplest of these methods, which we call "Numbers", is based on numeric comparison. Numbers requires the users to compare the SAS values displayed on each device encoded into (typically 4 digit long) numbers. Since 4 digit numbers can be easily displayed on low-resolution screens, pairing based on numeric comparison is applicable to a large number of devices. Given its utility, this method has been studied considerably in the past [14], [10], [4]. We observe that the prior usability studies investigating Numbers [14], [10], [4] indicate that although users hardly ever reject matching numbers on two devices, a *critical* task of detecting non-matching numbers (and thus potential *man-in-the-middle* attacks) can be error-prone. In other words, although Numbers exhibit no false positives (or *safe errors* [14]), it is quite likely to lead to false negatives (or *fatal errors*).

Our Contributions: Motivated by a common knowledge that humans tend to (better) remember numeric representations in smaller blocks or chunks (e.g., phone numbers), we propose a very simple and an intuitive method of employing "hyphen-delimited" numbers in device pairing. Similar to Numbers, the proposed method (we call Hyphen-Delimited Numbers) appeals to a large number of pairing scenarios where both devices are equipped with basic low-resolution displays capable of showing few numbers or characters. We validated our proposal by performing usability studies, the results of which show that Hyphen-Delimited Numbers improve the robustness as well as usability of the pairing based on numeric comparison. Our results provide sufficient statistical evidence that whenever pairing based on numeric comparison is deployed in practice, Hyphen-Delimited Numbers should be used as opposed to Numbers. Our work demonstrates that minor modifications at the "User Layer" can make remarkable impact on the usability and security of computer systems.

Organization: The rest of this paper is organized as follows. In Section II, we describe the security model and summarize relevant protocols. We present the design and implementation of our new scheme in Section III, and discuss our usability studies and test results in Section IV. In Section V, we review prior work on pairing schemes and in Section VI, we draw conclusions from the results of our usability tests.

II. Communication and Security Model

The pairing protocols used in this paper are based upon the following communication and adversarial model [15]. The devices to be paired are connected via two types of channels: (1) a short-range, high-bandwidth, bidirectional wireless channel and (2) an auxiliary, low-bandwidth, physical OOB channel(s). An adversary attacking the pairing protocol is assumed to have full control on the wireless channel; namely, he or she can eavesdrop on, delay, drop, replay and modify messages. On the OOB channel, the adversary can eavesdrop on, delay, drop, replay, and re-order messages. It can not modify them, however. In other words, the OOB channel is assumed to be authenticated.

To date, two three-round pairing protocols based on short authenticated strings (SAS) have been proposed [7][5]. In a communication setting involving two users restricted to running three protocol instances, these SAS protocols need to transmit only k $(= 15)$ bits of data over an OOB channel. As long as the cryptographic primitives used in these protocols are secure, an adversary attacking them can not win with a probability significantly higher than 2^{-k} $(= 2^{-15})$. This provides security equivalent to that provided by 4- to 5-digit PIN-based ATM authentication [15].

III. Hyphen-Delimited Numbers

The results of prior usability investigations of the Numbers pairing method [14], [10], [4] indicate that there are no safe errors or false positives, i.e., users never reject matching numbers displayed on two devices. This is a positive result implying that under normal circumstances, i.e., when no attacks occur, the pairing process will always succeed in the first attempt. On the other hand, prior results show that fatal errors or false negatives are likely to occur, i.e., users can possibly accept non-matching numbers shown on two devices. This, unfortunately, is a negative result which shows that it is possible that the users are likely to accept pairing with an attacker's device.

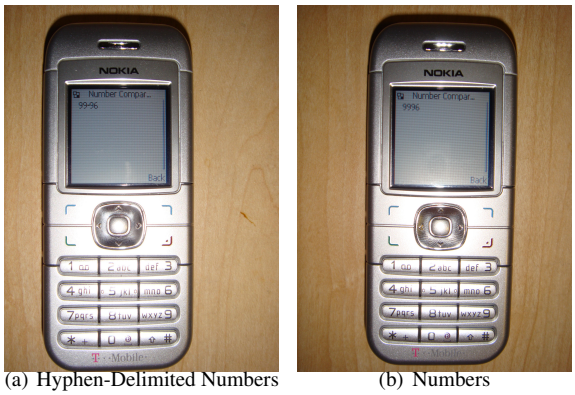
Our objective was to address the above drawback with Numbers and thus improve the robustness of numeric comparison. In other words, we wanted to develop a pairing method that would have the simplicity of Numbers and be applicable on a wide variety of devices, and at the same time, have minimal safe as well as fatal errors (ideally, none at all). To this end, we develop Hyphen-Delimited Numbers, a very simple and intuitive method of numeric comparison. The Hyphen-Delimited Numbers pairing method is motivated by a common knowledge that humans tend to (better) memorize numeric representations (e.g., phone numbers, social security numbers) in blocks of small chunks, generally separated by hyphens[2]. During the pairing process, users do not need to memorize number(s) displayed on their devices for a long period of time or recall them later. However, to compare the two numbers, a user would first need to read one of the numbers, memorize it momentarily until she compares it with the other number. Our hypothesis was that by making use of hyphen-delimited numbers, the user's cognitive load in the whole process can be reduced, thereby improving the accuracy of comparison.

Design: In our new pairing method, we simply display numbers separated by hyphens. For example, a 4-digit number 9996 is displayed as 99-96. A pairing scheme, in its entirety, consists of three phases: (1) the device discovery phase, wherein the devices exchange their identifiers over the wireless channel prior to communicating, (2) the pairing protocol execution phase, wherein the devices execute the desired pairing protocol over the wireless channel, and (3) the authentication phase, where the devices authenticate the messages exchanged during the previous phase using OOB channels. For the sake of our experimentation, we skipped the first two phases and concentrated on the third phase. We did this because our main goal was to test the feasibility of the way we intended to

[2]For example, 10-digit cell phone numbers are commonly represented as xxx-xxx-xxxx. Similarly, 9-digit social security numbers are represented as xxx-xx-xxxx.

implement the OOB channels, i.e., by using Hyphen-Delimited Numbers and its comparison with Numbers.

Implementation: For our implementation, we used two Nokia 6030b mobile phones. This phone model was selected for testing because they are affordable entry-level models that have been commercially available for a few years. As such, their features are representative of those one would expect to find on today's average personal mobile device. This phone model has a reasonable size and quality (128 x 128 pixel) display, capable of showing several digits of numeric and ASCII representations. The Nokia 6030b runs the Nokia operating system and supports version 2.0 of the Mobile Information Device Profile (MIDP) specification and version 1.1 of the Connected Limited Device Configuration (CLDC) framework, which are both part of the Java Platform, Micro Edition, or J2ME. To utilize these APIs we wrote our test programs in the Java programming language using the Java Wireless Toolkit version 2.5.2 for CLDC. Because we were only working with the authentication phase of the pairing scheme and not the device discovery or pairing protocol execution phase and did not make use of a wireless channel, no actual wireless connection between the two mobile devices was necessary for our tests. Figure 1 depicts a snapshot of our implementation of Hyphen-Delimited Numbers (showing the hyphen-delimited number 99-96) and Numbers (showing the number 9996).



(a) Hyphen-Delimited Numbers (b) Numbers

Fig. 1. Pairing using Hyphen-Delimited Numbers and Numbers: showing 99-96 and 9996

IV. Usability Testing

After implementing the two pairing methods, Numbers and Hyphen-Delimited Numbers, on a common platform, we are ready to start the usability study. The goal of our study is to evaluate the respondents ability to perform numeric comparison with respect to the two methods, in terms of the following factors:

- Efficiency: time it takes to compare the two numbers
- Robustness: how often each method leads to false positives (or rejection of a successful pairing instance) and false negatives (or acceptance of a failed pairing instance). As mentioned earlier, following the terminology introduced in [14], we will refer to the errors in the former category as safe errors and the latter as fatal errors.
- Usability: how each method fares in terms of personal user preference

Each respondent was asked to compare two five sets of numbers, each four digits long. The first was a set of five numbers for the Numbers method, i.e, with no spaces or delimiters between their digits, and the second set contained numbers for the Hyphen-Delimited Numbers method, i.e., the very same numbers but delimited with a hyphen (-) between the second and third digits.

One challenge we faced while performing our usability study was was how the respondents be kept from influencing him/herself during the second round of tests (i.e., while testing Hyphen-Delimited Numbers), because the numbers would be the same as those used in the first round (i.e., while testing Numbers), the only difference being the inserted hyphen between second and third digits of each number. This problem was remedied by asking the respondents to perform the tests corresponding to the two methods within a day's gap – on the first day of the testing, the respondent was asked to compare Numbers and on the second day the Hyphen-Delimited Numbers.

A. Study Participants

We recruited 40 participants for our study which lasted over a period of four days overall. The participants were chosen on a first-come first-serve basis from respondents to recruiting posters and emails. Prior to recruitment, each participant was briefed on the estimated amount of time required to complete the tests and on the importance of completing the two tests.

Half of the participants were mostly university students (between the ages of 18 – 25 years), both graduate and undergraduate. This resulted in a fairly young, well educated and technology-savvy group. The remaining half belonged to an age group of 30 – 45 years. The second group did not necessarily comprise of technically savvy individuals. Thus our study represents a sample for identifying methods suitable for the broad cross-section of user population.

We prepared two questionnaires: *Pre-test* – to obtain user demographics and *post-test* – for user feedback. None of the study participants reported any physical impairments that could interfere with their ability to complete given tasks. The gender split was: 80% male and 20% female. (We attribute the uneven numbers to the nature of the test location – in and around an engineering school.)

B. Test Cases

For both tested methods, we created several test-cases simulating normal scenarios (i.e., when no attacks or faults occur) as well as abnormal scenarios (i.e., when attacks or faults do occur).

The respondents were asked to compare two numbers on the two mobile handsets. The numbers were chosen by the test administrator, making sure that there was no way the respondent could have known before hand which numbers will follow.

To keep the respondent from being influenced in anyway, as mentioned earlier, the two tests of comparing Numbers (denoted Test 1) and Hyphen-Delimited Numbers (denoted Test 2), were administered on two different days.

To test for fatal errors, we asked the users to compare numbers which are similar looking or mismatched only in one of the digits. This was done deliberately to test for worst case scenarios. Note that on an attacked protocol session, all numbers are equally likely to occur as the SAS values are uniformly distributed. On the first day the respondents were asked to compare the following numbers: (1) 9996 and 9969, (2) 6653 and 6653, (3) 3323 and 3323, (4) 1245 and 1254, and (5) 3233 and 3323. Whereas on the second day, the following was asked to be compared: (1) 99-96 and 99-69, (2) 66-53 and 66-53, (3) 33-23 and 33-23, (4) 12-45 and 12-54, and (5) 32-33 and 33-23.

C. Testing Process

Our study was conducted in a variety of on-campus and off-campus venues. This was possible since the test devices were mobile, test set-up was more-or-less automated and only a minimal involvement from the test administrator was required. After giving a brief overview of our study goals (prior to the first batch of study), we asked the participants to fill out the pre-test questionnaire in order to collect demographic information. Next, the participants were given a brief introduction to the cell-phone devices used in the tests. Each participating user was then given the two devices and asked to follow on-screen instructions shown during each task to complete it. As already mentioned in Section IV-B, to reduce the learning effect on test results, the tasks were always presented to the user in random order. User interactions throughout the tests and timings were logged automatically by the testing framework. After completing the tasks in each batch of the tests, each user filled out a post-test questionnaire form, where they provided their feedback. The users were also given a few minutes of free discussion time, where they explained to the test administrator about their experience with the various methods they tested.

D. Test Results and Interpretations

We collected data in two ways: (1) by timing and logging user interaction, and (2) via questionnaires and free discussions.

For each method, completion times was automatically logged by the software. In the post-test questionnaire, we solicited user opinions about all tested methods. Participants were asked about their preferences with wireless devices and opinion about the method to pair two devices, i.e., Numbers or Hyphen-Delimited Numbers.

In this section, we present the results of our study. We also interpret the obtained results, wherever applicable. We first consider various mechanical data, i.e., time to completion and error rates. We then analyze the user preference ratings.

Time Results: Our observations of response time (averages with standard deviation over all 5 test cases per respondent) are reflected in Figure 2 for both younger and older respondents when tested with the two methods. The overall average response time for younger respondents turned out to be 2.471 seconds (sd = 1.297 seconds) for Numbers and 2.906 seconds (sd = 1.214 seconds) for Hyphen-Delimited Numbers. These timings were consistent with prior usability studies [10], [4] of Numbers, which were also performed with younger user population. For the older respondents, the overall average was 5.139 seconds (sd = 1.313) for Numbers and 5.690 seconds (sd = 1.446 seconds) for Hyphen-Delimited Numbers.

Our results prompted one particular observation that the average response time of both younger and older respondents seemingly went up when they are asked to match Hyphen-Delimited Numbers. However, only for the older respondents, the (dependent samples) t-tests indicated that there was highly significant difference ($p \ll 0.001$) in timing of the two methods (Hyphen-Delimited Numbers being more time consuming than Numbers). The t-tests on the response time of younger respondents, on the other hand, did not show any significant difference ($p = 0.209$). This is probably because our older users became somewhat more careful and slower while using Hyphen-Delimited Numbers.

As expected, the (independent samples) t-tests indicated highly significant difference ($p \ll 0.001$) in the average timings of younger and older respondents (the latter being higher) for both Numbers and Hyphen-Delimited Numbers. This implies that age has a negative effect on the efficiency of both the methods.

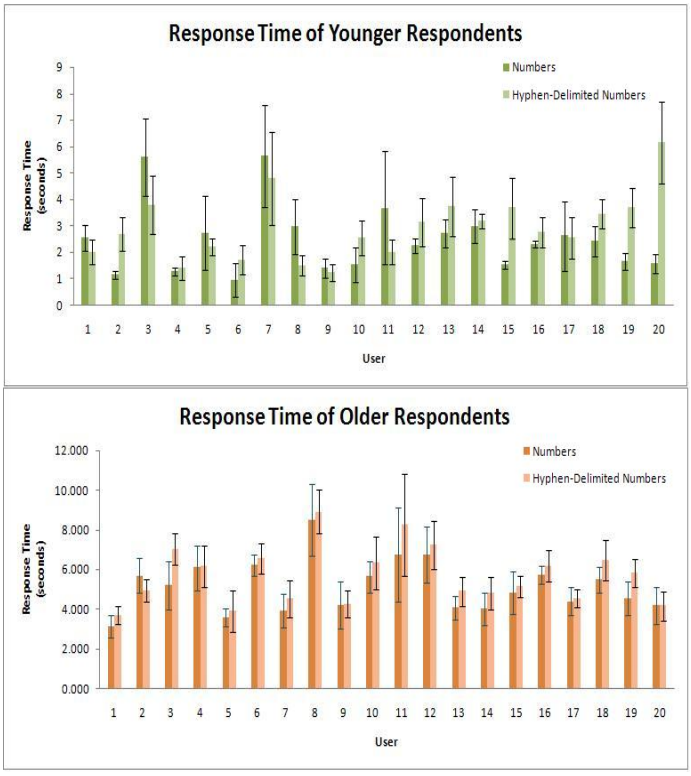


Fig. 2. Average Response Time (with standard deviation) Per User

Error Results: Throughout our tests, we did not encounter any safe error rates with both methods. For Numbers, this confirms the results shown in [10], [4]. For Hyphen-Delimited Numbers, this was a positive sign because it was aimed at improving the robustness of pairing based on numeric comparison.

Our observations of fatal error rates (per user), similar to

the timing results of Figure 2, are depicted in Figure 3, for both younger and older respondents. For easier comparison, overall error rates are depicted in Table I. Validating our motivation behind proposing Hyphen-Delimited Numbers, our results show that the percentage of fatal errors drop significantly in both younger and older respondents, when Hyphen-Delimited Numbers are used compared to Numbers. This was confirmed by the (dependent samples) two-proportions Z-tests, which resulted in highly significant difference ($p \ll 0.001$) between the fatal error rates of the two methods (Hyphen-Delimited Numbers being less error-prone) for both of our user populations.

Although the fatal error rates appear higher for our older respondents, we did not find any significant effect of age on the error rates of the two methods, as shown by the Fisher's exact test ($p = 0.478$).

Fig. 3. Average Fatal Error Rates Per User

	Numbers	Hyphen-Delimited Numbers
Young	Safe Errors = 0 % Fatal Errors = 12 %	Safe Errors = 0 % Fatal Errors = 2 %
Old	Safe Errors = 0 % Fatal Errors = 23 %	Safe Errors = 0 % Fatal Errors = 6 %

TABLE I
OVERALL ERROR RATES

User Preferences: The 3-part graph in Figure 4 shows user preferences for various questions which we asked our respondents before and after the pairing tests.

- Out of the 40 candidates who were asked to perform the usability tests, a good 75% preferred using wireless devices to wired ones. While 20% of the of our users did not prefer using wireless devices, 5% where indifferent to the type of devices they used. Our results confirms the growing ubiquity of wireless devices among everyday users.
- 65% of the respondents felt it was more convenient to match Hyphen-Delimited Numbers as opposed to plain Numbers, 25% of the population was indifferent to the way in which the numbers were displayed and 10% did not believe that Hyphen-Delimited Numbers were convenient when it came to matching two numbers. It was intuitive that a large majority of our users preferred Hyphen-Delimited Numbers. Also, as indicated by our error results, even the users who preferred Numbers or were indifferent, were subconsciously more comfortable detecting mismatching numbers when delimited by a hyphen.
- When asked if they would prefer this method of pairing two Bluetooth devices, a clear majority of 55% believed that this was in fact a better way of pairing, because it was much faster and hassle free than the present method. 30% of the sample were not sure of their opinion and a mere 15% preferred the present method of pairing.

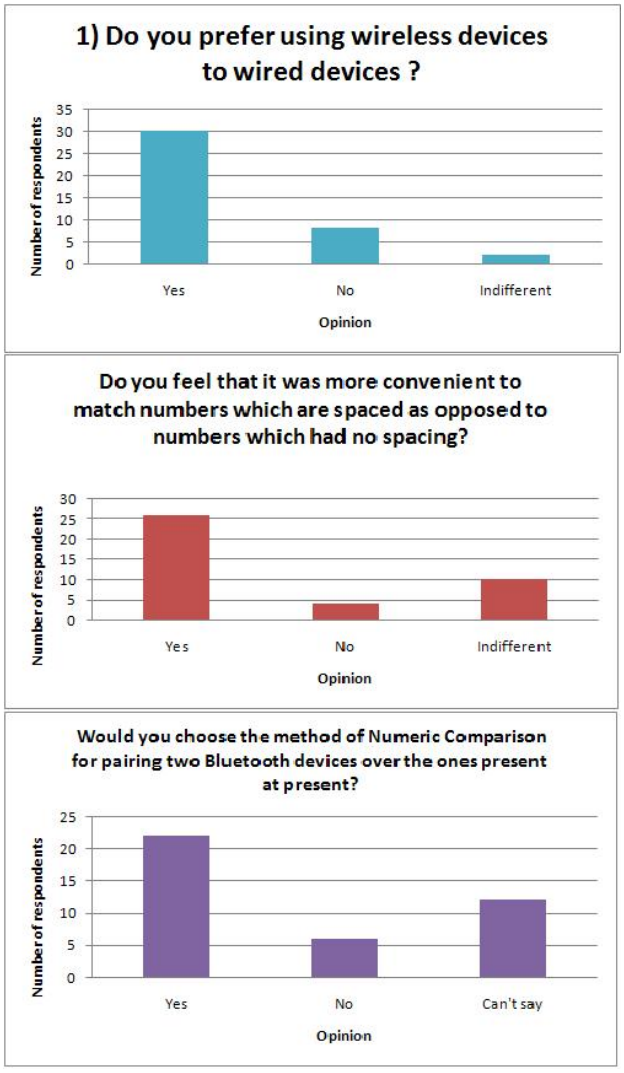


Fig. 4. User Feedback

V. RELATED WORK

In addition to the most closely related work to the theme of our paper, i.e., [14], [10], [4], there exists a significant

amount of prior work on the general topic of pairing. Due to space constraints, we only summarize it here. For a complete description, we refer the reader to [4] (related work section).

In their seminal work, Stajano, et al. [12] proposed the establishment of a shared secret between two devices using a link created through a physical contact (such as an electric cable). Balfanz, et al. [1] extended this approach through the use of infrared as a d2d channel; the devices exchange their public keys over the wireless channel, then exchange (at least 80-bit long) hashes of their respective public keys over the infrared channel.

A number of pairing methods were based on comparison of random images, e.g., the visual hash based on Random Arts by Perrig et al. [8]. McCune et al. proposed the "Seeing-is-Believing" (SiB) scheme [6]. SiB involves establishing two unidirectional visual d2d channels; one device encodes the data into a 2-D barcode and the other device reads it with a camera. As an improvement to SiB, Saxena et al. [9] proposed a new scheme based on a visual OOB channel. The scheme uses one of the SAS protocols [5] and is aimed at pairing two devices of which only one has a relevant receiver.

Goodrich, et al. [3] proposed a pairing scheme based on "Mad Lib" sentences that is also built upon the protocol of Balfanz et al. The main idea of their procedure is to establish a d2h channel by encoding the pairing data into English sentences, which users can then easily compare.

A very recent proposal, [11], focuses on pairing two devices with the help of "button presses" by the user. This scheme is based upon a protocol that first performs an unauthenticated Diffie-Hellman key agreement, then authenticates the established key using a short password.

VI. Conclusion and Future Work

In this paper, we proposed a simple, natural and an intuitive way of using hyphen-delimited numbers to improve the robustness of device pairing based on numeric comparison. Similar to Numbers, the proposed method appeals to a large number of pairing scenarios where both devices are equipped with basic low-resolution displays capable of showing few numbers/characters. We validated our proposal by performing usability studies (with respondents belonging to two age groups), the results of which show that Hyphen-Delimited Numbers improve the robustness as well as usability of the pairing process. Our results provide sufficient statistical evidence that whenever pairing based on numeric comparison is deployed in practice, Hyphen-Delimited Numbers should be used as opposed to Numbers. Our work demonstrates that minor modifications at the "User Layer" can make remarkable impact on the usability and security of computer systems.

The specific conclusions of the analysis of our test results are as follows.

- Hyphen-Delimited Numbers exhibits no safe errors, similar to Numbers.
- Compared to Numbers, the fatal errors rates in Hyphen-Delimited Numbers drop significantly among both younger and older respondents. (We did not find any statistical evidence of the effect of age on fatal error rates of two methods).
- A large majority of respondents prefer Hyphen-Delimited Numbers over Numbers.
- Among the older respondent pool, the response time was higher for Hyphen-Delimited Numbers compared to that of Numbers. (We did not find any statistical evidence for the same among our younger respondents.) Also, age has a negative effect on the speed of the two methods.

In our future work, we would perform further usability studies to test for the effect of age on error rates of the two methods. A potential advantage of Hyphen-Delimited Numbers is that it can be used to compare numbers longer than just 4 digits and thus improve the security of the pairing process without compromising the usability. To this end, we plan on evaluating Hyphen-Delimited Numbers with longer numeric representations.

References

[1] D. Balfanz, D. Smetters, P. Stewart, and H. C. Wong. Talking to strangers: Authentication in ad-hoc wireless networks. In *NDSS*, 2002.

[2] C. Gehrmann, C. J. Mitchell, and K. Nyberg. Manual authentication for wireless devices. *RSA CryptoBytes*, 7(1):29 – 37, Spring 2004.

[3] M. T. Goodrich, M. Sirivianos, J. Solis, G. Tsudik, and E. Uzun. Loud and Clear: Human-Verifiable Authentication Based on Audio. In *ICDCS*, 2006.

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

[5] S. Laur and K. Nyberg. Efficient mutual data authentication using manually authenticated strings. In *CANS*, 2006.

[6] J. M. McCune, A. Perrig, and M. K. Reiter. Seeing-is-believing: Using camera phones for human-verifiable authentication. In *IEEE S&P*, 2005.

[7] S. Pasini and S. Vaudenay. SAS-Based Authenticated Key Agreement. In *PKC*, 2006.

[8] A. Perrig and D. Song. Hash visualization: a new technique to improve real-world security. In *CrypTEC*, 1999.

[9] N. Saxena, J.-E. Ekberg, K. Kostiainen, and N. Asokan. Secure device pairing based on a visual channel. In *S&P06*.

[10] N. Saxena and J. Voris. Pairing Devices with Good Quality Output Interfaces. In *ICDCS WISP Workshop*, June 2008.

[11] C. Soriente, G. Tsudik, and E. Uzun. BEDA: Button-Enabled Device Association. In *IWSSI*, 2007.

[12] F. Stajano and R. J. Anderson. The resurrecting duckling: Security issues for ad-hoc wireless networks. In *Security Protocols Workshop*, 1999.

[13] J. Suomalainen, J. Valkonen, and N. Asokan. Security associations in personal networks: A comparative analysis. In *ESAS*, 2007.

[14] E. Uzun, K. Karvonen, and N. Asokan. Usability analysis of secure pairing methods. In *USEC*, 2007.

[15] S. Vaudenay. Secure communications over insecure channels based on short authenticated strings. In *CRYPTO05*.

[16] M. Čagalj, S. Čapkun, and J.-P. Hubaux. Key agreement in peer-to-peer wireless networks. *Proceedings of the IEEE*.