Journal of Computer Security 21(3), 347-370.

Can Jannie Verify? Usability of Display-Equipped RFID Tags for Security Purposes

Alfred Kobsa* Rishab Nithyanand† Gene Tsudik*
Ersin Uzun‡

November 26, 2012

Abstract

The recent emergence of RFID tags capable of performing public key operations enables a number of new applications in commerce (e.g., RFID-enabled credit cards) and security (e.g., ePassports and access-control badges). While the use of public key cryptography in RFID tags mitigates many difficult security issues, certain important usability-related issues remain, particularly when RFID tags are used for financial transactions or bearer identification.

In this paper, we focus exclusively on *techniques with user involvement* for secure user-to-tag authentication, transaction verification, reader expiration and revocation checking, as well as pairing of RFID tags with other personal devices. Our approach is based on two factors: (1) recent advances in hardware and manufacturing have made it possible to mass-produce inexpensive passive display-equipped RFID tags, and (2) high-end RFID tags used in financial transactions or identification are attended by a human user (typically, their owner). Our techniques rely on user involvement coupled with on-tag displays to achieve better security and privacy. Since user acceptance is a crucial factor in this context, we conducted comprehensive user studies to assess usability of all considered methods. This paper reports on our findings.

1 Introduction

Radio Frequency Identification (RFID) technology was initially envisaged as a replacement for barcodes in supply chain and inventory management. A small device with no power source of its own (called an RFID tag) could be read from some distance away by a special device (called an RFID reader), without line-of-sight alignment as is needed

*University of California, Irvine; {*kobsa,gene.tsudik*}*@uci.edu*
†Stony Brook University; *rnithyanand@cs.stonybrook.edu*
‡Palo Alto Research Center; *ersin.uzun@parc.com*

for barcodes. However, its many advantages have greatly broadened the scope of possible applications today. Current and emerging applications range from visible and personal tags (e.g., toll transponders, passports, credit cards, access badges, livestock or pet tracking devices) to stealthy tags in merchandise (e.g., clothes, pharmaceuticals and books/periodicals). The costs and capabilities of RFID tags vary widely depending on the target application. At the high end of the spectrum are the tags used in e-Passports, electronic ID (e-ID) Cards, e-Licenses, and contactless payment instruments. Such applications involve relatively sophisticated tags that only cost a few dollars (usually under $10), though they are powerful enough to perform sophisticated public key cryptographic operations, security and privacy issues remain when these tags are used as a means of payment or for owner/bearer identification. In this paper, we address four such issues:

User-to-Tag Authentication: Many applications of RFID technology in electronic payments or identification documents require user-to-tag authentication before disclosing any information. This is needed to prevent leakage of valuable or private information. Current systems require trust in readers for the purpose of authentication. For example, users must enter PINs into ATMs or Point-of-Sale (POS) terminals to authenticate themselves to the RFID tag embedded into their ATM or credit card. However, this makes users vulnerable to attacks, since secret PINs are being disclosed to third-party readers that are easy to hack and modify [12, 14].

Transaction Verification: RFID tags are commonly used as payment and transaction instruments (e.g., in credit, debit, ATM and voting cards). In such settings, a malicious reader can easily mislead the tag into signing or authorizing a transaction different from the one communicated to, or intended by, the user. This is possible because there is no direct channel from a tag to its user on regular RFID tags (i.e., no secure user interface), and the only information a user receives (e.g., a receipt, or an amount displayed on the cash register) is under the control of a potentially malicious reader. Thus, it seems impossible for a user to verify transaction details (e.g., amount or currency) in real time,

Note that the goal of transaction verification is to allow users to check transaction details, rather than to defend against man-in-the-middle (MiTM) attacks, e.g., a transaction is approved for a different merchant than intended. We assume that a merchant trusts that its readers have not been maliciously manipulated. Furthermore, the use of location-limited channels such as NFC (Near Field Communication) and frequency-restricted RFID can prevent attacks on reader-tag communication.

Reader Revocation and Expiration: Any certificate-based Public Key Infrastructure (PKI) needs an effective expiration and revocation mechanism. In RFID systems, it intuitively concerns two entities: RFID tags and RFID readers. The former only becomes relevant if each tag has a "public key identity," and we claim that revocation of RFID tags is a non-issue since, once a tag identifies itself to a reader, the reader can use any current method for revocation status verification. In contrast, expiration and

revocation of *reader* certificates constitutes a challenging problem in any public key-enabled RFID system. This is because RFID tags, being power-less passive devices, cannot maintain a clock. In other words, an RFID tag, on its own, has no means to verify whether a given certificate has expired or whether any revocation information is recent.

Secure Pairing of RFID Tags: Current high-end RFID tags cannot establish a secure ad-hoc communication channel to another device, unless the latter is part of the same RFID infrastructure (i.e., an authorized reader). Establishing such a channel seems important as it would give tag owners the ability to manage their tags. Previously proposed secure device pairing solutions require an auxiliary communication channel to authenticate devices and establish a secure communication channel [29, 28]. Until recently, however, RFID tags lacked user interfaces and thus could not be paired with other devices. Novel display-equipped RFID tags open a new chapter in RFID security and give users more control over their tags. Using an Near Field Communication (NFC) capable personal device like a smart-phone, for instance, a user can change settings on a personal RFID tag.

Focus: In this paper, we take advantage of recently developed technology that equips high-end RFID tags with a small passive display, e.g., see Figure 1 for an example tag by NXP Semiconductors. We refer to such tags as **D**isplay-**E**quipped **R**FID **T**ags or DERTs. The only publicly known application of DERTs are eID cards used in Germany since November 2010 [4]. As we show in the remainder of this paper, carefully designed user interaction with personal DERTs can yield solutions to aforementioned problems. We present several simple techniques that require little or no change to already well-established RFID infrastructures, e.g., back-end processing systems of ePassports and payment instruments. Thereafter, we conduct a thorough study to assess usability of these techniques. Since this paper is primarily focused on usability (rather than security), no security analyses or proofs are presented.

One key motivating factor for this work is the fact that DERTs are already being produced and are available on the market. Moreover, they cost only a little more than their display-less counterparts. We note that our work and usability studies with DERTs are also somewhat relevant to passive cards with displays and buttons that require physical contact with readers.

The rest of this paper is organized as follows: we summarize related work in Section 2, describe our technical approach in Section 3, present a comprehensive usability evaluation of the proposed techniques in Section 4, and conclude with a summary in Section 5.

2 Related Work

We now overview related work in several RFID-relevant categories: (1) user-to-tag authentication, (2) transaction verification, (3) reader revocation, and (4) device pairing.

Figure 1: NXP Display-Equipped RFID Tag (DERT) with Two Buttons

2.1 User-to-Tag Authentication

User authentication is a fundamental problem that has received a great deal of attention in the security community, for several decades. Solutions range from simple modifications of the standard PIN/password entry techniques [42, 19] to schemes that pose more complicated cognitive tasks to users [40, 20].

Authentication of users to passive devices (such as RFID tags) is a very recent issue. In the initial proposal by Czeckis *et al.* [18], users authenticate to an RFID tag by moving or shaking it (or the wallet containing it) in a certain pattern. However, this method assumes that RFID tags are equipped with an accelerometer, and requires users to memorize movement patterns. Also, it is prone to passive observer attacks. A similar technique called "PIN-Vibra" was suggested by Saxena *et al.* [39] for authenticating to an accelerometer-equipped RFID tag using a mobile phone. In it, a vibrating mobile phone is used to lock or unlock RFID tags. While the usability of PIN-Vibra seems promising, it has a some drawbacks: (1) high error rates – accelerometers on tags can not perfectly decode PINs encoded in phone vibrations, (2) the user's phone must be present and functional (e.g., not be out of battery) whenever the tag has to be used, and (3) accelerometer-equipped RFID tags are relatively expensive and do not lend themselves well to other applications that would help amortize their cost.

The user-to-tag authentication solution described and tested in this paper is most similar to the approach first proposed by Abadi *et al.* [8] for authentication using contact-based smartcards, where a displayed random number is modified by a user to match a PIN.

2.2 Transaction Verification

Current systems that address transaction verification and amount fraud utilize data mining (e.g., [16]), machine learning techniques (e.g., [9]), and out-of-band communication. Most banks verify transactions via alternate communication mediums such as email or telephone. A complete survey of modern fraud detection techniques for Card Present (a.k.a, off-line) and Card not Present (a.k.a, on-line) transactions is given by

Kou *et al.* [30]. In this paper, we present a simple technique that permits user-aided verification using DERTs and fully mitigates amount and currency fraud for Card Present transactions. To the best of our knowledge, this is the first work that offers a real solution and provides a comprehensive analysis of its usability.

2.3 RFID Reader Revocation Checking

Three popular methods to check the status of a public key certificate (PKC) are: Certificate Revocation Lists (CRLs) [25], Online Certificate Status Protocol (OCSP) [34] and Certificate Revocation System (CRS) [33, 32]. CRLs are signed lists of revoked certificates periodically published by certification or revocation authorities (CAs or RAs). The usage of CRLs is problematic in RFID systems since they require the tag to have a clock in order to determine whether a given CRL is sufficiently recent, and since the communication overhead can be quite high if the number of revoked entities is large. OCSP is an online revocation checking method that reduces storage requirements for all parties involved, while providing timely revocation status information. Although well suited for large connected networks, it is a poor fit for RFID systems as it requires constant connectivity between readers and OCSP responders. Furthermore, the need for a two-round challenge-response protocol with OCSP responders may make it susceptible to network congestion and slow turnaround times. CRS offers implicit, efficient and compact proofs of certificate revocation. However, it is unworkable in the RFID context as it also requires verifiers (RFID tags) to have a clock.

Despite much prior work in certificate revocation and RFID security little has been done to address reader PKC revocation and expiration problems. This is not for the lack of trying since, in fact, these issues have been noted in [24, 27, 23], Recently, a method that entails user involvement and DERTs has been proposed in [36, 37]. A preliminary usability study in [36] was followed by a comprehensive usability analysis of the proposed method with actual DERTs and realistic user tasks. Further details can be found in [37] and [36].

2.4 Device Pairing

A number of device association/pairing methods have been proposed over the past few years. They use various out-of-band (OOB) channels in the process of establishing a secure connection, and as a result, exhibit different usability characteristics. Recent work in [29, 28] and [31] surveys many pairing methods and reports on their usability. However, because of the nature of (very) basic displays that can be integrated into RFID tags, only visual text-based methods are appropriate for DERTs.

In this paper, we use the "Copy" method introduced by Uzun *et al.* [41], and evaluate its usability in the DERT setting. In the *copy* pairing technique, one device displays a randomly generated passkey, which the user types into the second device. The devices automatically run a password-based authenticated key agreement protocol (e.g., [13]), that succeeds or fails depending on the user's ability to copy the passkey correctly between the devices and the presence of an active attack on the communication channel, e.g., man-in-the-middle or denial of service attacks.

3 Proposed Techniques

3.1 General Assumptions

All methods described below share the following general assumptions:

1. Tags are owned and operated by individuals (users/owners) who understand their roles in each context (users only need to know the actions they are required to perform, but not the reasons for performing them).
2. Tags are powerful enough to perform public key operations (at least signature verification). This is true for all our target applications.
3. Tags are equipped with an one-line alpha-numeric display (OLED or ePaper) capable of showing at least 8 characters. This is made possible by current DERT technology.
4. Tags can maintain simple counters or timers *while* powered by a reader.
5. Each tag has a programmable button.[1]

3.2 User-to-Tag Authentication

The authentication method described in Figure 2 is designed for DERTs but can be used on any wireless, interface-constrained device.

We make three additional assumptions:

1. Tags are capable of generating short random numbers (i.e., 4-6 decimal digits).
2. Users have access to a possibly *untrusted* keypad (or keyboard) with cursor keys. The keypad can be part of the reader, or be connected to it.
3. Tags always clear and reset their displays after authentication. Note that this is possible even in the case of malicious readers due to the presence of residual charges in a DERT.

3.2.1 The Protocol

In order to unlock a tag for a transaction (e.g., a credit card at a store, a cash card at an ATM, or an e-passport at a hotel), the user needs to be authenticated by proving knowledge of a secret, such as a PIN. The following method, which is a variant of the method proposed in [8] for battery powered smart-cards, allows user-to-tag authentication without requiring any buttons/keys on the tag. Moreover, the PIN is protected from potentially malicious (and certainly untrusted) readers.

1. Powered by the reader, DERT generates a one-time random number of the same length as the PIN. DERT proceeds to display this random number. Note that this $nonce$ is not known by the reader that powers the DERT.

[1] We used NXP tags with two buttons in our usability tests. Note that one of the button actions can always be substituted with a timeout.

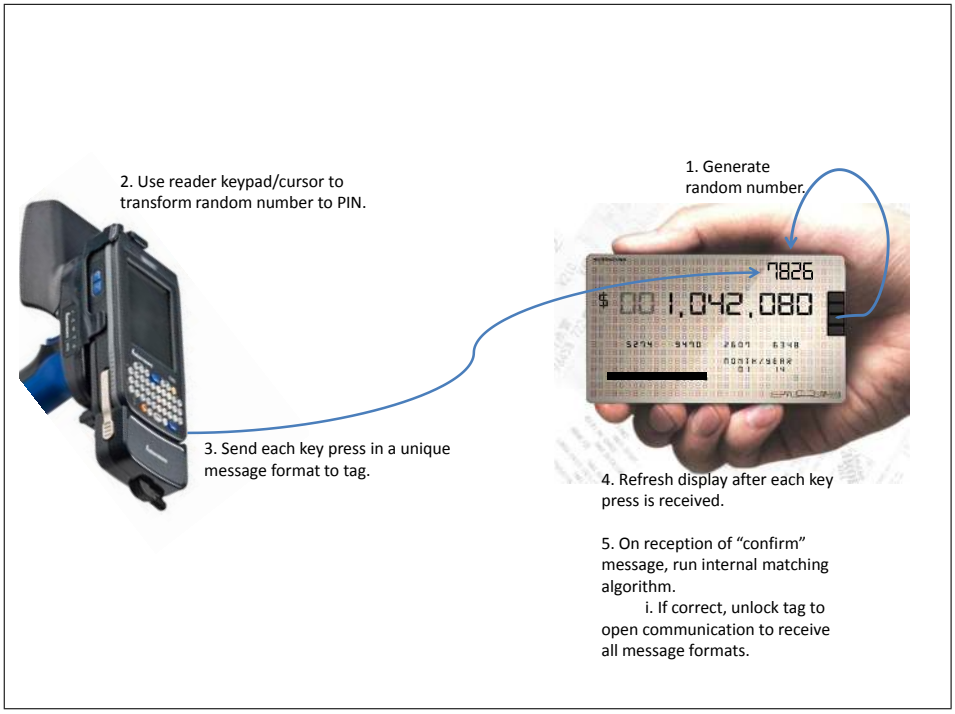


Figure 2: Secure User-to-DERT Authentication

2. User operates the cursor keys ($\uparrow, \downarrow, \leftarrow, \rightarrow$) on the reader keypad to basically *adjust* this random number on the DERT to his/her PIN. This is done digit by digit. For example, if the random number displayed by DERT is "5723" and the user's PIN is "296", the necessary sequence of key presses is: 1) 4 times $\downarrow$, $\rightarrow$, 2) 5 times $\uparrow$, $\rightarrow$, 3) 3 times $\downarrow$, $\rightarrow$, 4) 3 times $\uparrow$, followed by *Confirm*. For each user key-press, the reader sends a corresponding message to the tag detailing the key-press, thereby prompting the tag to update its display.

3. Upon receipt of the *Confirm* message, DERT unlocks itself for a transaction if the PIN was entered correctly.

Since the reader is unaware of the nonce initially generated by the DERT, it is impossible (even with knowledge of the sequence of keys pressed by the user) to reconstruct the PIN used to unlock the DERT. Note that this method's security is based on several factors. The first is our assumption about the DERT's ability to generate cryptographically secure random numbers. The second security requirement is that the user *must alternate* $\uparrow$ and $\downarrow$ movements between digits. In other words, if only the $\downarrow$ key is used for small PIN digits (i.e., < 5) instead of sometimes going past "9" to reach it, or vice versa for large digits, then such a pattern may leak information about the PIN if the protocol is executed repeatedly with the same reader. If there is a concern about such leaks, they can be easily prevented by allowing only one of the $\uparrow$ or $\downarrow$ keys to be used

when modifying the digits.

Shoulder-Surfing Resistant Variant: In a shoulder-surfing attack, an adversary somehow observes the user's actions to obtain critical information (e.g., the PIN entered into an ATM). Such attacks range from simply looking over the victim's shoulder to using a camera to observe him or her. They are simple to launch and effective in public areas where large crowds or long queues are likely to occur. By masking all digits except the one being modified, it is easy to make the above protocol shoulder-surfing resistant (It does not become *shoulder-surfing proof*, however).

We tested both flavors of this protocol and used '\' as the masking character. Although '$*$' is more commonly used for this purpose, the prototype firmware on our test tags was not yet capable of displaying it.

3.3 Transaction Verification

Our approach to transaction amount verification is designed to work with any RFID-enabled payment instrument. Its primary goal is to provide simple, secure and usable transaction verification at a Point-of-Sale (PoS). The following additional assumptions are necessary:

- The user knows the correct amount for the intended transaction (e.g., has access to a printed receipt).
- It is possible to display the amount of the intended transaction – within some degree of accuracy – without the need for decimal points and/or commas. This assumption is introduced due to current limitations of the character set supported by the DERT.
- The transaction amount (and possibly the currency code) can be displayed within the DERT display size, i.e., 10 digits.

3.3.1 The Protocol

1. DERT receives transaction details from the reader (seller/merchant).
2. DERT verifies that the details (e.g., issuing bank, account number, etc.) match their counterparts in the reader PKC. Protocol is aborted in case of a mismatch.
3. DERT extracts and displays user-verifiable data, i.e, the amount and optionally the currency code. It then enters a countdown stage that lasts for a predetermined period of time (e.g., 10 seconds).
4. User observes transaction information and, if the transaction amount and other details are deemed correct, presses the *Confirm* button on DERT before the timer runs out. At this point, DERT signs the time-stamped transaction statement and sends it to the reader. This signed statement is then sent to the payment gateway and eventually to the financial institution that issued the payment DERT.

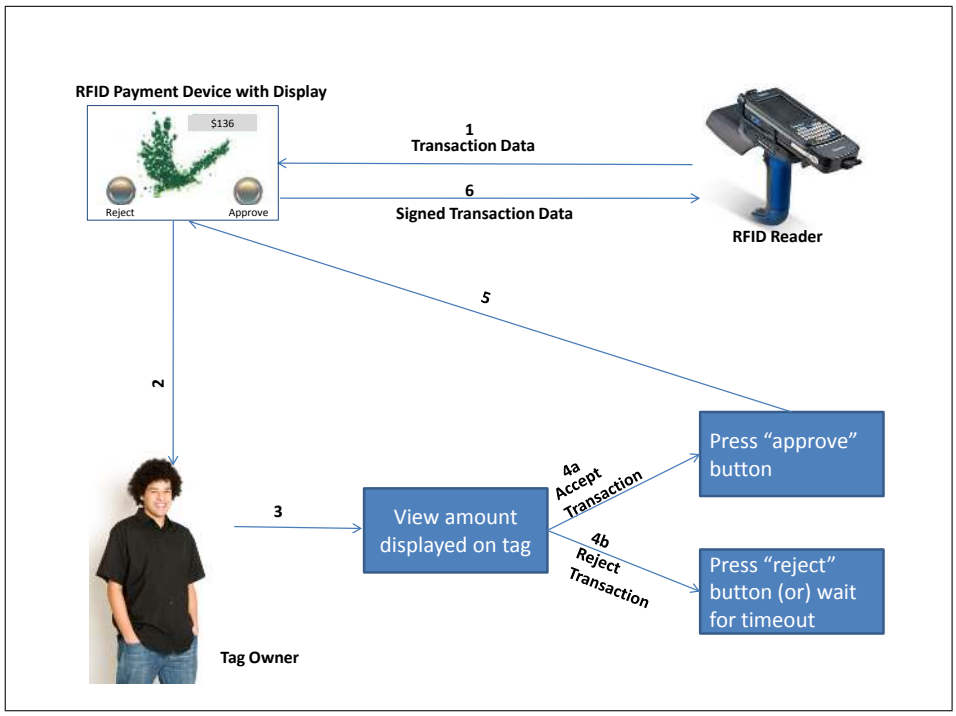


Figure 3: DERT-enabled Transaction Verification

However, if the user decides that transaction details are incorrect, the timer runs out (or the user presses the reject button, if one is available) and DERT automatically aborts the protocol.

The same protocol is also illustrated in Figure 3.

3.4 Reader Revocation Status Checking

Our approach for reader certificate expiration and revocation checking [36] is aimed at personal RFID tags – such as ePassports, e-licences or credit/debit cards – when used in places where trust is not implicit. For example, trust in readers might be implicit in international airports (immigration halls) or at official border crossings. Whereas, it is not implicit in many other locations, such as car rental agencies, hotels, flea markets or duty-free stores.

This approach entails the following additional assumptions:

- Each tag is owned and physically attended by a person who is reasonably aware of the current date.
- Tags are aware of the identity and public key of the system-wide trusted Certificate Authority (CA), e.g., the ICAO CVCA [3]. In other words, all tags and readers are subsumed by a system-wide Public Key Infrastructure (PKI),

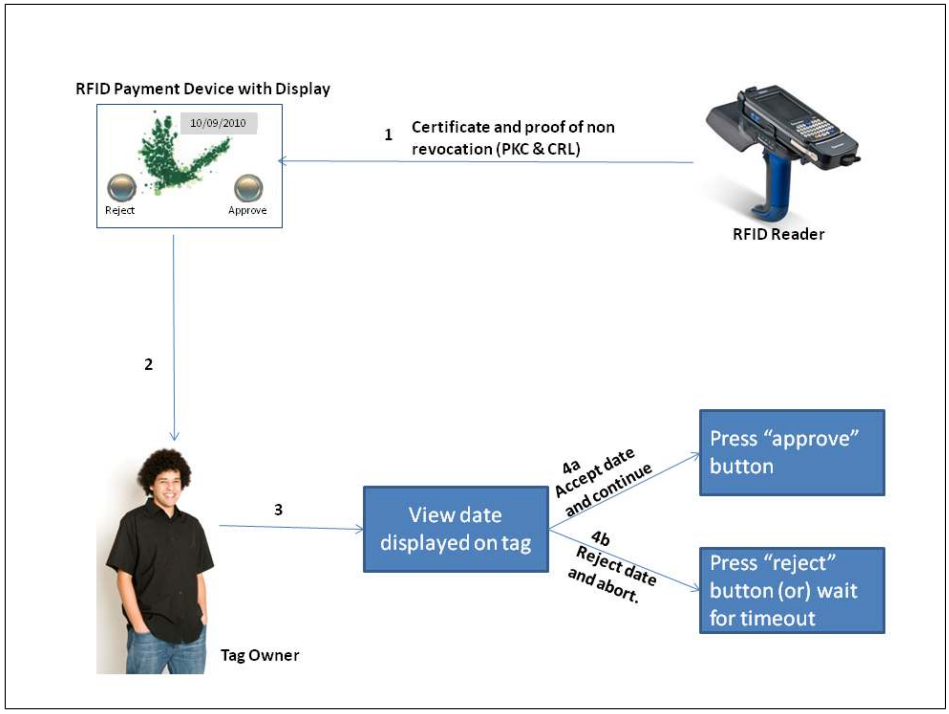


Figure 4: Reader Certificate Expiration/Revocation Checking

- The CA is assumed to be infallible: anything signed by the CA is guaranteed to be genuine and error-free.
- With fixed frequency, the CA issues an updated revocation structure, such as a CRL.
- All tags are aware of the periodicity of issuance of the revocation information and thus can determine expiration time of the revocation structure by simply consulting its issuance time-stamp.
- A tag can retain, in its local non-volatile storage, the last valid time-stamp it has encountered.

Note that our usage of the term "time-stamp" is not restricted to time, i.e., hours and minutes. It is meant to express (at appropriate granularity) issuance and expiration of both certificates (PKCs) and revocation information.

3.4.1 The Protocol

Before providing any information to the reader, a tag has to validate the reader's certificate (PKC). The verification process is as follows (also illustrated in Figure 4):

1. Freshly powered-up DERT receives the Certificate Revocation List (CRL) and the reader's Public Key Certificate (PKC).

   Let CRL_{iss}, CRL_{exp}, PKC_{iss} and PKC_{exp} denote issuance and expiration times of CRL and PKC, respectively. The last encountered valid time-stamp kept by DERT is denoted as Tag_{Curr}.

2. If either CRL_{exp} or PKC_{exp} is smaller than Tag_{curr}, or $CRL_{iss} \geq PKC_{exp}$, DERT aborts.

3. DERT checks whether CRL includes the serial number of the reader certificate. If so, it aborts.

4. DERT checks the CA signatures of PKC and CRL. If either check fails, DERT aborts.

5. If CRL_{iss} or PKC_{iss} is more recent than the currently stored date, DERT updates it to the more recent of the two.

6. DERT displays the lesser of: CRL_{exp} and PKC_{exp}. It then enters a countdown stage of fixed duration (e.g., 10 seconds).

7. The user decides whether the displayed time-stamp is in the future. If so, the user presses the DERT button before the timer runs out, and communication with the reader continues. Otherwise, the user does nothing: the timer runs out and DERT automatically aborts the protocol.

NOTE: we use the term CRL above to denote a generic revocation structure.

3.5 Secure Device Pairing

Our protocol for bootstrapping a secure communication channel between DERTs and more powerful computing devices such as laptops or cell-phones (i.e., pairing) is based on the "Copy" pairing technique introduced in [41] and described in Section 2.

3.5.1 Additional Assumptions

This DERT application requires the following additional assumption that:

> *DERT can generate short random passcodes for the purpose of device pairing and can run one of the secret based key agreement protocols mentioned in 3.5.3.*

3.5.2 The Protocol

The method operates as follows.

1. DERT generates and displays a sufficiently long (e.g., 6-9 digit) decimal passcode.

2. The software on the other device prompts the user to enter this passcode.

3. Using this (presumably the same) passcode, DERT and the second device run an authenticated key agreement protocol based on the short shared secret to establish a stronger common key and then confirm its possession by both parties.

3.5.3 Secret-Based Key Agreement Protocols

Unlike previously mentioned protocols (where standard cryptographic primitives may be *plugged in* to achieve security goals), it is not clear how device pairing can be used to bootstrap a secret channel. One possibility is to use so-called Password Authenticated Key Exchange (PAKE) protocols, that involve two or more parties sharing a low-entropy secret. As a result of running PAKE, the parties securely establish a strong (high-entropy) cryptographic key, even in the presence of an adversary in full control of the communication channel. PAKE examples include: encrypted key exchange (EKE) [11], simple password exponential key exchange [26], and password-authenticated key exchange by juggling (J-PAKE) [22].

4 Usability Analysis

Since all proposed methods require varying degrees of user involvement, it is very important to assess their usability in order to gauge eventual user acceptance in a real-world deployment scenario. To this end, we conducted a comprehensive usability study with prototype implementations. The goal of the study was to provide answers to the following questions:

1. How do subjects rate usability of proposed methods in each problem context?
2. Can subjects perform required tasks with sufficiently low error rates?
3. Are subjects willing to perform these tasks on a regular basis?

4.1 Apparatus and General Experimental Procedures

Our study was conducted using display-equipped RFID tags (DERTs) from NXP Semiconductors and an HID Omnikey 5321 desktop reader [5]. DERTs were equipped with an integrated 10-position alpha-numeric (ePaper) display unit and two buttons. All code was written in Java 2 Platform Standard Edition with the Java Smart Card I/O API [6].

All tests were conducted in a designated conference room at a university campus, over a period of 25 days. Subjects were introduced to the concept of personal RFID tags, with RFID-enabled credit cards and ePassports serving as our main motivating examples. A short presentation using the same set of slides (to ensure consistency) was made to each subject, explaining each usage scenario and subject's task in each protocol. These tasks were explained again before each protocol was tested. Subjects were informed of the importance of maintaining natural behavior during the study and were requested not to ask questions during the testing process. However, they were allowed to talk to the test administrator before and after each protocol was tested. Subjects

were then presented with DERTs used in the tests in order to familiarize them with the "hardware". After completing a background questionnaire to collect demographic data, tests were conducted for each protocol described in Section 4.3, and task performance times and error rates were measured.

After testing each protocol, every subject completed a post-test survey. It included the System Usability Scale (SUS) questionnaire [15], a widely used and highly reliable 10-item 5-point Likert scale to measure user satisfaction, and several other questions framed to gain insights into the potential acceptance of the proposed methods.

On average, it took about 30 minutes to finish the entire series of tests, whereupon, each subject was rewarded with either a movie coupon or a $10 Starbucks gift card.

4.2 Subjects

Our study involved 35 subjects recruited through email and flyers, selected on a first-come first-serve basis. The first 5 respondents were assigned to the pilot test (phase 1) subject pool. Data obtained from this pilot phase was used to make important decisions regarding the need for additional test cases in each protocol. Phase 1 was also important to verify the stability and the limits of our RFID hardware setup. Due to several changes made after the pilot tests in phase 1, data obtained in this phase was not comparable to the data gathered from the remaining 30 participants. Consequently, phase 1 data is not reflected in the results discussed in this paper.

Of the 30 subjects who took part in phase 2, 30% (9 subjects) were aged 18 to 24, 36.67% (11 subjects) 25 to 30, 16.67% (5 subjects) 30-34, and 16.67% (5 subjects) over 40. Gender distribution was nearly even with 53.33% (16 subjects) males and 46.67% (14 subjects) females. The subject pool was quite well-educated, with 86.67% (26 subjects) having a bachelors degree or higher. We attribute this to the specifics of the study venue – a university campus. While this sample is clearly not representative of, e.g., the U.S. population, college students are regarded as potentially good surrogates for future *e*arly adopters in the context of the diffusion of an innovation [21]. Only 6.67% (2 subjects) reported a disability that impaired their visual perception.

4.3 Test Procedures and Results

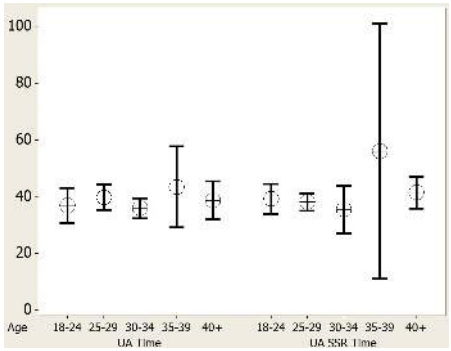
4.3.1 User Authentication Variants

In tests of user-tag authentication, each subject was presented with an Automated Teller Machine (ATM) simulator and was asked to authenticate as the tag owner. While our protocol can be used to lock and unlock tags for any purpose, the ATM environment was used to aid the understanding of potential use cases.

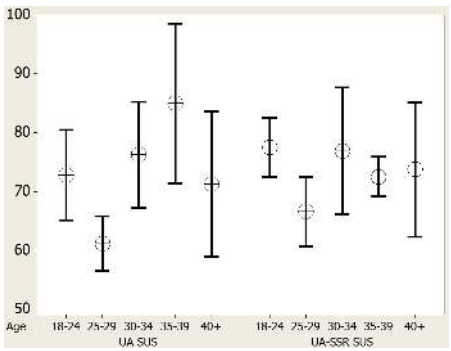
After being informed about his/her role in the protocol, each subject was presented with a Logitech N305 wireless number pad [7] that had four highlighted cursor keys to aid in digit manipulation. Next, a subject was asked to complete four test cases (two for each variant). For all test cases, the same four digit PIN was used by the same subject. Furthermore, the initial random number generated by the tag always required a minimum of 13 key presses total for successful authentication. This was done in order to compare completion times between subjects more accurately. In this section, we

present our results and attempt to provide insight into which protocol is better suited for the real world.

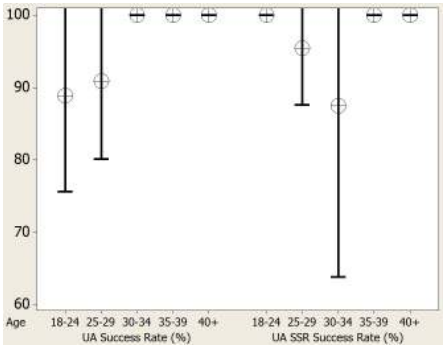
Completion Time and Error Rates. Each variant had 60 test cases, and the average time to completion for both variants was well under a minute. As shown in Fig. 5, subjects of all age groups performed reasonably well given the tasks associated with both regular authentication and shoulder-surfing resistant authentication protocols. The study yielded an average completion time of 38.47 seconds for the regular authentication protocol (UA), and 39.68 seconds for the shoulder-surfing resistant variant (UA-SSR). A paired t-test showed that this difference is not statistically significant. Looking at error rates does not give us better insight either: the study yielded low error rates of 6.67% and 3.33% for the UA and UA-SSR protocols, respectively.



(a) Age vs.Time to Completion (seconds) – 90% CI



(b) Age vs. SUS Score – 90% CI



(c) Age vs. Success Rate (%) – 90% CI

Figure 5: Interval Plots: Performance of UA and UA-SSR Variants (Crosshairs denote mean)

SUS Scores and Usability Analysis. The UA protocol was rated at 68.58 out of 100 on SUS, while the UA-SSR protocol received a higher score of 72.58. The possible

reasons for this are noted in the following discussion section.

When asked if they would like to see the protocols implemented in the real world for the purpose of user authentication, 50% (15 subjects) indicated that they would like to see an implementation of UA, while 36.67% (11 subjects) were neutral). When asked the same question about UA-SSR, 60% (18 subjects) agreed that they would like to see it implemented, while 23.33% (7 subjects) were neutral. Finally, when asked if they preferred using UA-SSR over UA, 50% (15 subjects) picked UA-SSR while 20% (6 subjects) did not have a preference. The question received a score of 2.89 on the 5-point Likert scale.

To better understand usability characteristics of UA and UA-SSR protocols, we computed cross-correlations between four variables: (1) time to completion, (2) SUS scores, (3) success rate, and (4) subject's willingness to use the protocol on a regular basis for authentication (labeled "Application Use"). Table 1 shows Pearson correlation

	Time Taken	SUS Score	Success Rate
SUS Score	-.258 (.047)	-	-
Success Rate	.188 (.152)	.044 (.589)	-
Application Use	-.168 (.207)	.685 (0)	-.072 (.507)

Table 1: Pearson Correlation Coefficient (r, p) Value Matrix for UA.

coefficients for UA. We observe only one statistically significant correlation – between the application use and SUS score variables (as one might expect). Another (lower) negative correlation was observed between the SUS score and time to completion. This leads us to conjecture that time required for authentication may have been undesirable for some subjects.

Table 2 shows Pearson correlation coefficients for UA-SSR. As for UA, there is only one statistically significant correlation – between the application use and SUS score variables. Another (less significant) correlation was observed between the application use and success rate variables. Interestingly, no negative correlation between the SUS score and time to completion variables was observed.

Discussion. An analysis of completion times and error rates does not point at a clear winner between UA and UA-SSR. However, SUS scores and subjects' opinions indicate that UA-SSR is the preferred variant. Interestingly, our analysis also reports that younger individuals, in general, rated UA-SSR protocol as more usable than UA. Post-

	Time Taken	SUS Score	Success Rate
SUS Score	.070 (.491)	-	-
Success Rate	.133 (.301)	.236 (.070)	-
Application Use	.108 (.416)	.625 (0)	.344 (.007)

Table 2: Pearson Correlation Coefficient (r, p) Value Matrix for UA-SSR.

test subject interviews lead us to conclude that UA-SSR is preferred because of the presence of the '*cursor*' that indicated which digit was currently being manipulated. (Recall that all digits that are not currently being manipulated are replaced by a '\'). Since this feature was not present in the UA protocol, subjects often lost track of the digit they were manipulating, which caused some of them to become frustrated during the authentication process.

Several subjects indicated concern with the usability of our protocols for visually challenged individuals. Current authentication and PIN-entry techniques allow individuals with visual impairments to perform their roles with reasonable ease through the use of Braille. In contrast, our protocols do not seem to be easily accessible for this subject group, and may require special hardware such as personal radio frequency headphones. This is an important concern that we hope to address in future work.

We note that, while other user-to-tag authentication techniques, such as [39], take significantly less time to complete (mean: 7.12 seconds), their error rates are prohibitively high at 78.75%.

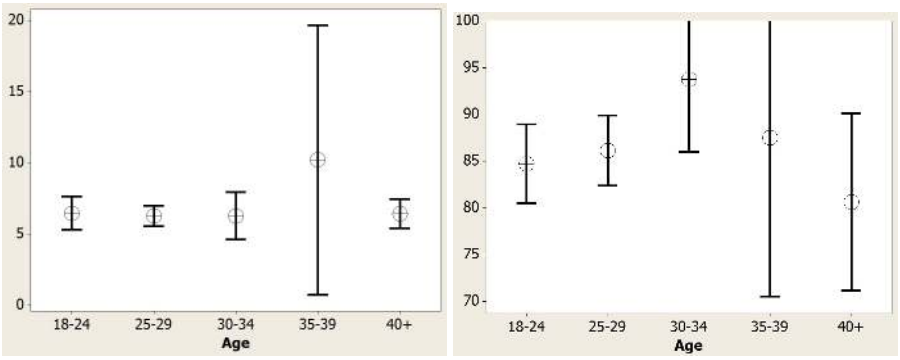
4.3.2 Transaction Verification

While the transaction verification method can be used with any RFID payment/transaction instrument, we focused on the common case of RFID-enabled credit cards in a Point-of-Sale (PoS) environment. This was done not only to help subjects understand use cases more clearly, but also because we envision this case as the primary application domain for this protocol.

Test procedure. After an explanation of their tasks and roles, each subject was presented with a vending machine simulation, with the structure and products (i.e., "look-and-feel") similar to the Best Buy airport vending machines common in US airports [2]. Each subject was then asked to make two separate sets of purchases (each set was a test case). Upon pressing the *checkout* button on the machine, a digital receipt appeared on the display monitor of the vending machine. Next, the total amount that the vending machine intended to charge was displayed by the tag. Each subject was asked to check whether the two amounts matched. If they matched, the vending machine was deemed to be "honest". Otherwise, an amount mismatch indicated a malicious vendor attempting to overcharge the user. For each subject, one of the (randomly selected) test cases involved a malicious vending machine that attempted to over-charge by \$1, \$10 or \$100 (the amount was selected at random).

Completion Time and Error Rates. For the 60 $(= 30 * 2)$ test cases, the study yielded an average completion time of 6.6 seconds, with a standard deviation of 3.0 seconds. Surprisingly, all 30 subjects completed their tasks successfully and no errors were recorded in the process. Furthermore, subjects from all age groups completed the transaction verification task in very little time, as shown in Fig. 6(a).

SUS Scores and User Opinion. Subjects rated usability at 86 out of 100 on SUS [15]. This is far above the "industry average" of 70.1 reported in [10], and indicates



(a) Age vs. Time to Completion (seconds) – 90% CI (b) Age vs. SUS Score – 90% CI

Figure 6: Interval Plots: Performance of Transaction Verification Protocol (Crosshairs denote mean)

	Time Taken	SUS Score	Success Rate
SUS Score	.036 (.689)	-	-
Success Rate	-	-	-
Application Use	-.016 (.805)	.485 (0)	-

Table 3: Pearson Correlation Coefficient (r, p) Value Matrix for Transaction Verification

excellent usability and acceptability. Also, a staggering 96.67% (29 subjects) stated that they would like to see the system implemented on their own personal tags. Only 1 subject opposed this idea. The average score on a 5-point Likert scale was 4.57, with a standard deviation of 0.64.

As before, to better understand the usability characteristics of the transaction verification protocol, we computed the cross correlations between four variables – (1) time to completion, (2) SUS scores, (3) success rate, and (4) willingness of to use the protocol on a regular basis for transaction verification (labeled "Application Use"). Table 3 shows Pearson correlation coefficients for the transaction verification protocol. Due to the absence of failures, there are no correlation coefficients with the success rate variable. Only one medium positive correlation was observed between the application use and SUS score variable. Furthermore, there no correlation was observed between the time taken and other variables; this was expected since the time to completion was very small.

Discussion. As the results indicate, our method takes 6.6 seconds to complete (on average), which is well below 21 seconds considered to be the maximum acceptable time to users [17]. However, low error rates might be a consequence of our specific implementation and test cases. It is possible that user errors arise often in real-world de-

ployments if malicious vendors manipulate placement of decimal points on the DERT display (e.g., $344.1 instead of $34.41). We were unable to test this attack in our study since NXP prototype DERTs can not display decimal points. However, we believe that a careful design would likely help keep error rates low, even in cases of malicious or erroneous placement of decimal points.

Some improvements could be made to aid both usability and security with decimal points. One way is to displaying decimal points in a visually distinct manner, e.g., using special color, contrast, font, or background. This needs to be evaluated via further user studies with more sophisticated DERTs.

Other errors might occur due to malicious merchants using wrong currency identifiers. To prevent this, DERTs should be capable of displaying currency codes. Furthermore, a currence code for a specific transaction should be part of reader-supplied transaction details and it should also be encoded in reader'ss PKC. (For example, if the merchant is in Australia, the reader's PKC should encode AUD$ and no other currency ought to be allowable).

4.3.3 Reader Revocation Status Checking

To help subjects understand the concept of personal RFID tags and the reader certificate expiration/revocation problem, the ePassport example was used throughout this test. Care was taken to prevent subjects from checking clocks, watches or cell phones for the current date, in order to put an upper-bound on the error rate. After being informed of their role in the protocol, each subject was presented with our implementation and asked to execute the protocol eight times. Finally, opinions were solicited via the post-test questionnaire.

Test procedure. Each subject was presented with eight test cases in a random order. These corresponded to DERT-displayed dates of: +/-1 day, +/-3 days, +7 days, -29 days, -364 days and -729 days from the actual test date, where "+" and "-" indicate future and past dates, respectively. All dates were presented in the MM/DD/YYYY format. Our choices of -29, -364 and -729 days were deliberate so as to make their "staleness" more obscure to the subjects. After a date was displayed on the DERT, each subject was asked to decide to: (1) accept the date by pressing the *OK* button, or (2) reject it by pressing the *CANCEL* button. A *safe default* timeout of 10 seconds was selected. If no subject input was provided within this time, the date was automatically rejected.

Completion Time and Error Rates. For the 240 (=8*30) test cases, the study yielded average completion time of 6.39 seconds, with a standard deviation of 2.39 seconds, as shown in Table 4. This illustrates that subjects made quick decisions regarding timeliness of displayed dates. Among the 240 test cases, the false negative rate (reject dates that are not stale) was quite low, at 4.44%. No one rejected a date that was seven days in future, and only 6.67% (2 subjects) of the sample rejected dates that were one and three days in the future.

The false positive rate (stale date accepted) was considerably higher, 17.3% on average. When subjects saw dates that were 1 and 3 days earlier, error rates were only

Case	Mean Time (secs)	StDev (secs)	Mean Error Rate (%)
+1 Day	6.19	1.66	6.67
+3 Days	6.45	2.80	6.67
+7 Days	7.16	2.83	0.00
-1 Day	5.48	1.86	10.00
-3 Days	7.11	2.64	16.67
-29 Days	6.82	2.26	6.67
-364 Days	6.37	2.51	30.00
-729 Days	5.51	1.87	30.00
Overall	6.39	2.39	12.50

Table 4: Completion Times and Error Rates

	Time Taken	SUS Score	Success Rate
SUS Score	-.070 (.277)	-	-
Success Rate	-.033 (.618)	.020 (.756)	-
Application Use	.067 (.291)	.535 (0)	.037 (.610)

Table 5: Pearson Correlation Coefficient (r, p) Value Matrix for Reader Revocation Checking

10% and 0%, respectively. Surprisingly though, when subjects saw dates that were 29, 364 and 729 days earlier, error rates shot up to 16.7%, 30% and 30%, respectively. We elaborate on possible reasons for this spike below.

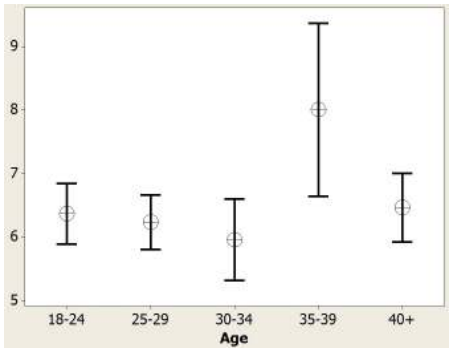
In terms of performance (i.e., time to completion, and error rates), results indicated that younger subjects (under 35) were more likely to complete the task faster and more accurately. This is illustrated in Fig. 7.

SUS Scores and User Opinion. Subjects that tested our implementation rated its usability at 76 on SUS [15]. We note that this is almost identical to the score of 77 obtained in [36], where subjects rated it based on a mock-up implementation on a Nokia N95 cell phone. The overall SUS score that we obtained is appreciably above the "industry average" of 70.1 [10], and indicates good usability and acceptability characteristics.

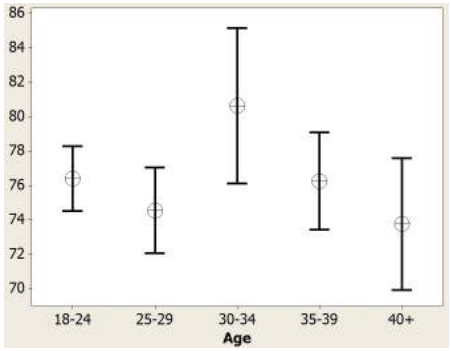
Furthermore, 70% (21 subjects) stated that they would like this system on their own personal tags, while 23.33% (7 subjects) were neutral to the idea. The average score on a 5-point Likert scale was 3.78 with a standard deviation of 0.77.

Table 5 shows the Pearson correlation coefficients for the reader revocation checking protocol. Surprisingly, the only statistically significant correlation was observed between the application use and SUS score variables.

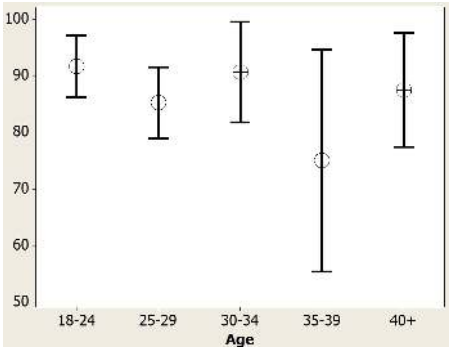
Discussion. As results show, our method very rarely yields false negatives: subjects can generally distinguish valid (future) from past dates. Whereas, with false positives,



(a) Age vs. Time to Completion (seconds) – 90% CI



(b) Age vs. SUS Score – 90% CI



(c) Age vs. Success Rate (%) – 90% CI

Figure 7: Interval Plots: Performance of Reader Revocation Checking Protocol (Crosshairs denote mean)

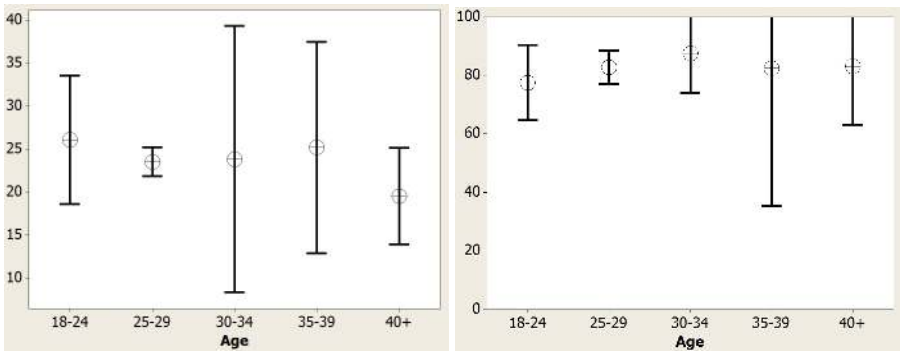
our results are mixed. Stale days are, for the most part, easily recognized as such. However, with stale years, error rates are quite high, at 30%. While we do not claim to know the exact reason(s) for this fact, some conjectures can be made. When confronted with a date, e.g., current dates on documents or expiration dates on perishable products, most people are used to first check day and month. They might not pay as much attention to more blatant errors such as "wrong year" perhaps because they consider it to be an unlikely event. However, we anticipate that year mismatches will be quite rare in practice, since (as mentioned earlier in the paper) tags can record the most recent *valid* date they encounter. Therefore, dates with stale year values will be mostly automatically detected and rejected by tags without the need for any user interaction. However, high error rates in wrong year values can still pose a threat if a tag is not used for a year or longer.

In all of our studies, dates were presented to the subjects in the American (MM/DD/YYYY) format. However, DERTs can be programmed to display dates in other formats, such as alphabetic month encodings (e.g., Apr 18, 2012) or the European (DD/MM/YYYY) format. This would require that dates be communicated and stored in some standard

universal format, such as Unix system time. We anticipate that errors can be substantially reduced if users could select preferred date display formats. Some relevant experiments and the discussion of various date formats can be found in [37].

4.3.4 Secure Device Pairing

We chose the "Copy" method described earlier for all device pairing tests. This choice was based on two factors: (1) our previous studies [41, 35] pointed at its low error rates, and (2) it is device-controlled and therefore resistant to so-called *rushed user behavior* [38].



(a) Age vs. Time to Completion (seconds) - 90% CI (b) Age vs. SUS Score – 90% CI

Figure 8: Interval Plots: Performance of Device Pairing Protocol (Crosshairs denote mean)

Test procedure. First, each subject was briefed on the purpose of pairing personal RFID tags with personal devices (in this case, a laptop). Next, the subject's role in the protocol was described. Subjects were then asked to enter a random 5-digit number generated by the tag into the laptop. Upon correct entry, they were notified of successful pairing via the tag and laptop displays, and a mock user interface depicting possible applications of the pairing was displayed on the laptop. Only a single test case was performed for each subject.

Completion Time and Error Rates. A total of 30 test cases were performed, yielding the average completion time of 23.904 seconds, with the standard deviation of 8.272 seconds. Only 3.33% of the sample (one subject) entered an incorrect number into the laptop that resulted in an error.

SUS Scores and Usability Analysis. Before rating the pairing protocol on SUS, subjects were clearly informed of the distinction between rating the pairing protocol and rating its applications. SUS was only used to understand the usability of the former, and resulted in a score of 81.83%. This indicates very good usability and acceptability.

	Time Taken	SUS Score	Application Use
SUS Score	-.148 (.385)	-	-
Application Use	-.188 (.245)	.475 (.094)	-
Pairing Use	-.407 (.024)	.323 (.081)	.618 (.071)

Table 6: Pearson Correlation Coefficient (r, p) Value Matrix for Tag-to-PC pairing

Furthermore, 86.67% (26 subjects) indicated that they found the "Copy" method easy to use and wanted to see it more often in the context of device pairing. 83.33% (25 subjects) indicated that they were likely or very likely to use the applications that were now available as a result of being able to pair their personal tags with other devices.

Discussion. High SUS scores, low error rates and positive user feedback point to good usability of the "Copy" device pairing approach and potential applications of tags paired with more sophisticated devices. An effective and usable pairing method should demonstrate high scores on all three of these. To better understand the dependencies among four selected variables, we computed their cross-correlations. Table 6 shows the Pearson correlation coefficients. Interestingly, there are three medium-to-high correlations. These are between: (1) perceived ease of use of the pairing method and time to completion (medium: -0.407), (2) likelihood of using applications of pairing and SUS score (medium: 0.475), and (3) perceived ease of use of pairing method and likelihood of using applications of pairing (high: 0.618).

5 Conclusions and Future Work

Recent advances in display technology and hardware integration have resulted in relatively inexpensive display-equipped RFID tags (DERTs). Their low cost and achievable security properties make DERTs desirable and ready for real world applications.

In this paper, we motivated the use of DERTs in several security-related contexts. In particular, we presented simple and intuitive techniques that address several security problems with personal RFID tags. These techniques take advantage of the newly available user interface (passive display) for RFID tags and presence of their (human) owners. Preliminary usability studies suggest that subjects found proposed methods quite usable. Moreover, subjects performed assigned tasks with reasonably low error rates. As more applications for DERTs are identified, we believe that they will soon enter mass production and methods proposed in this paper will become applicable to a wider range of usage scenarios.

However, further user studies are clearly needed. In particular, future work could address some limitations of the study presented in this paper by considering a more diverse subject pool, especially, in terms of age and educational background, as well as conducting studies outside the United States. Also, more experiments are needed to evaluate the effects of various protocol changes and potential improvements, including:

- Support for various date display formats

- Use of different time-out methods
- Increasing visibility of the decimal point and/or currency symbols

Finally, future studies could also benefit from looking at the effects of varying ambient elements, such as lighting conditions and introducing user distractions.

Acknowledgements

The authors are grateful to NXP Semiconductors, especially to Thomas Suwald and Arne Reuter, for providing us with display-equipped tags used in our studies. This work was supported in part by NSF grants #0831526 and #0953071.

References

[1] S. Bellovin; M. Merritt. "Encrypted Key Exchange: Password-Based Protocols Secure Against Dictionary Attacks". Proceedings of the I.E.E.E. Symposium on Research in Security and Privacy, Oakland (May 1992).

[2] Bestbuy To Put Gizmo Vending Machines In Airports. `http://www.pcworld.com/article/149684/best_buy_to_put_gizmo_vending_machines_in_airports.html`.

[3] BSI: Country Verifying Certificate Authority. `https://www.bsi.bund.de/cln_174/DE/Themen/ElektronischeAusweise/CVCAePass/CVCAePass_node.html`.

[4] BSI: The New ID-Card. `https://www.bsi.bund.de/cln_174/ContentBSI/Themen/Elekausweise/Personalausweis/ePA_Start.html`.

[5] Hid Omnikey 5321 Cl Usb Reader. `http://www.hidglobal.com/documents/OK5321_cl_ds_en.pdf`.

[6] Java Smart Card I/O. `http://java.sun.com/javase/6/docs/jre/api/security/smartcardio/spec/`.

[7] Logitech Wireless N305. `http://www.logitech.com/en-us/keyboards/keyboard/devices/6355`.

[8] M. Abadi, C. Burrows, C. Kaufman, and B. Lampson. Authentication and delegation with smart-cards. *Science of Computer Programming*, 21(2):93–113, 1993.

[9] E. Aleskerov, B. Freisleben, and B. Rao. Cardwatch: A Neural Network Based Database Mining System For Credit Card Fraud Detection. In *Computational Intelligence for Financial Engineering (CIFEr), 1997., Proceedings of the IEEE/IAFE 1997*, pages 220 –226, 1997.

[10] A. Bangor, P. Kortum, and J. Miller. An Empirical Evaluation Of The System Usability Scale. *Int. J. Hum. Comput. Interaction*, 24(6):574–594, 2008.

[11] S. M. Bellovin and M. Merritt. Encrypted Key Exchange: Password-Based Protocols Secure Against Dictionary Attacks. In *IEEE Symposium on Security and Privacy*, 1992, pages 72–85.

[12] K. Blumenthal. *Getting Going: ATM Fraud Gets Even More Brazen*, Wall Street Journal, November 2010. URL: `http://online.wsj.com/article/SB10001424052748703688704575621122308129984.html`

[13] V. Boyko, P. MacKenzie, and S. Patel. Provably Secure Password-Authenticated Key Exchange Using Diffie-Hellman. In *Advances in CryptologyEurocrypt 2000*, pages 156–171. Springer, 2000.

[14] D. Bradbury, *A Hole in the Security Wall: ATM Hacking*, Network Security, Vol. 2010, No. 6, June 2010, pp. 12-15. URL: `http://www.sciencedirect.com/science/article/pii/S1353485810700829`

[15] J. Brooke. SUS: A "Quick And Dirty" Usability Scale. In P. Jordan, B. Thomas, B. Weerdmeester, and A. McClelland, editors, *Usability Evaluation in Industry*. Taylor and Francis, London, 1996.

[16] P. Chan, W. Fan, A. Prodromidis, and S. Stolfo. Distributed Data Mining In Credit Card Fraud Detection. *IEEE Intelligent Systems*, 14(6):67–74, 1999.

[17] D. Cvrcek, J. Krhovjak, and V. Matyas. PIN (and Chip) or SignatureBeating the Cheating?. *International Workshop on Security Protocols*, LNCS 4631, Springer-Verlag, pages 69-75, 2007.

[18] A. Czeskis, K. Koscher, J. R. Smith, and T. Kohno. RFIDs And Secret Handshakes: Defending Against Ghost-And-Leech Attacks And Unauthorized Reads With Context-Aware Communications. In *CCS '08: Proceedings of the 15th ACM conference on Computer and communications security*, pages 479–490, New York, NY, USA, 2008. ACM.

[19] A. Evans, W. Kantrowitz, and E. Weiss. A User Authentication Scheme Not Requiring Secrecy In The Computer. *Commun. ACM*, 17(8):437–442, 1974.

[20] A. Forget, S. Chiasson, and R. Biddle. Shoulder-Surfing Resistance With Eye-Gaze Entry In Cued-Recall Graphical Passwords. In *CHI '10: Proceedings of the 28th international conference on Human factors in computing systems*, p. 1107–1110, ACM, New York, 2010.

[21] K. Gallagher, J. Parsons, and K.D. Foster. A Tale of Two Studies: "Replicating Advertising Effectiveness and Content Evaluation in Print and on the Web". In *Journal of Advertising Research*, 41(4):71–81, 2001.

[22] F. Hao and P. Ryan. Password Authenticated Key Exchange by Juggling. In *Security Protocols XVI*, Lecture Notes in Computer Science, pages 159–171, 2011.

[23] T. Heydt-Benjamin, D. Bailey, K. Fu, A. Juels, and T. O'Hare. Vulnerabilities In First-Generation RFID-Enabled Credit Cards. In *Financial Cryptography*, pages 2–14, 2007.

[24] J. Hoepman, E. Hubbers, B. Jacobs, M. Oostdijk, and R. Schreur. Crossing Borders: Security And Privacy Issues Of The European E-Passport. In *IWSEC*, pages 152–167, 2006.

[25] R. Housley, W. Ford, W. Polk, and D. Solo. Rfc 5280: Internet X.509 Public Key Infrastructure Certificate and CRL profile, May 2008.

[26] D. Jablon. Strong Password-Only Authenticated Key Exchange. In *Computer Communication Review*, ACM SIGCOMM, vol. 26, no. 5, pp. 5-26, October 1996.

[27] A. Juels, D. Molnar, and D. Wagner. Security And Privacy Issues In E-Passports. *Security and Privacy for Emerging Areas in Communications Networks, International Conference on*, 0:74–88, 2005.

[28] R. Kainda, I. Flechais, and A. Roscoe. Usability And Security Of Out-Of-Band Channels In Secure Device Pairing Protocols. In *SOUPS: Symposium on Usable Privacy and Security*, 2009.

[29] 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.

[30] Y. Kou, C. Lu, S. Sirwongwattana, and Y. Huang. Survey Of Fraud Detection Techniques. In *Networking, Sensing and Control, 2004 IEEE International Conference on*, volume 2, pages 749 – 754 Vol.2, 2004.

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

[32] S. Micali. Efficient Certificate Revocation. Technical Memo MIT/LCS/TM-542b, Massachusetts Institute of Technology, 1996.

[33] S. Micali. Certificate Revocation System. United States Patent 5,666,416, Sept. 1997.

[34] M. Myers, R. Ankney, A. Malpani, S. Galperin, and C. Adams. Internet Public Key Infrastructure Online Certificate Status Protocol- Ocsp. RFC 2560, `http://tools.ietf.org/html/rfc2560`, 1999.

[35] R. Nithyanand, N. Saxena, G. Tsudik, and E. Uzun. Groupthink: Usability Of Secure Group Association For Wireless Devices. In *12th ACM International Conference on Ubiquitous Computing (Ubicomp 2010)*, 2010.

[36] R. Nithyanand, G. Tsudik, and E. Uzun. Readers Behaving Badly: Reader Revocation In PKI-Based RFID Systems. In *15th European Symposium on Research in Computer Security (ESORICS 2010)*, 2010.

[37] R. Nithyanand, G. Tsudik, and E. Uzun. User Aided Reader Revocation in PKI Based RFID Systems. In *Journal of Computer Security. 2011 December; 19 (6): 1147-1172.*

[38] N. Saxena and M. Uddin. Secure Pairing Of "Interface-Constrained" Devices Resistant Against Rushing User Behavior. In *International Conference on Applied Cryptography and Network Security (ACNS 2009)*, 2009.

[39] N. Saxena, M. Uddin, and J. Voris. Treat 'em Like Other Devices: User Authentication of Multiple Personal RFID Tags. In *SOUPS '09: Proceedings of the 5th Symposium on Usable Privacy and Security*, New York, NY, USA, 2009. ACM.

[40] T. Perkovic, M. Cagalj, and N. Saxena. Shoulder-surfing Safe Login in a Partially Observable Attacker Model. In *Financial Cryptography and Data Security*, volume 6052, pages 351–358, 2010.

[41] E. Uzun, K. Karvonen, and N. Asokan. Usability Analysis of Secure Pairing Methods. In *FC'07/USEC'07: Proceedings of the 11th International Conference on Financial cryptography and 1st International conference on Usable Security*, Berlin, Heidelberg, 2007. Springer-Verlag.

[42] M. Wilkes. *Time Sharing Computer Systems*. Elsevier Science Inc., New York, 1975.