To appear in the Proceedings of the 2011 European Symposium
on Research in Computer Security, Leuven, Belgium

Usability of Display-Equipped RFID Tags for Security Purposes

Alfred Kobsa[1], Rishab Nithyanand[2], Gene Tsudik[1], and Ersin Uzun[3]

[1] University of California, Irvine, CA, USA
{kobsa,gtsudik}@uci.edu
[2] Stony Brook University, NY, USA
rnithyanand@cs.stonybrook.edu
[3] Palo Alto Research Center, CA, USA
ersin.uzun@parc.com

Abstract. The recent emergence of RFID tags capable of performing public key operations has enabled 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 for 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 association 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 usually attended by a human user (namely the 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 thoroughly evaluate the usability of all considered methods through comprehensive user studies and report 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 RFID tag) could be read from some distance away by a special device (called RFID reader), without line-of-sight alignment as is needed 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/pet tracking devices) to stealthy tags in merchandize (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<10).

Even 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: In many applications of RFID in electronic payment and in identification documents, authentication of the user to the tag before disclosing any information is necessary to prevent leaks 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 leaves users vulnerable to attacks, since secret PINs are being disclosed to third party readers that are easy to hack and modify.

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 that is 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 gets (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 (in real time) transaction details, e.g., the amount or the currency. This problem becomes especially important with current electronic credit cards.

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, namely 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 powerless 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 [21], [20]. 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 NFC-capable personal device (such as a smart-phone), for instance, a user can change settings on a personal RFID tag.

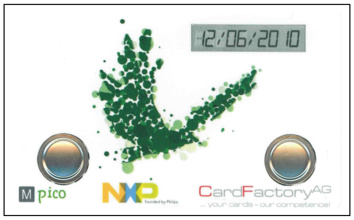


Fig. 1. NXP Display-Equipped RFID Tag (DERT) with two buttons

The gist of our approach is to take advantage of recently developed technology that allows high-end RFID tags to be equipped with a small passive display (see Figure 1 for a tag manufactured by NXP Semiconductors). We refer to such tags as **D**isplay-**E**quipped **R**FID **T**ags or DERTs. The only other publicly known application of DERTs are eID cards in Germany since November 2010 [3]. As we will show in the remainder of this paper, carefully designed user interaction with personal DERTs can yield solutions to the aforementioned problems. We present several simple techniques that require little or no change to already well-established RFID back-end infrastructures (e.g., the back-end processing systems of ePassports, payment instruments, etc.). Thereafter we conduct a thorough study to assess the usability of these techniques.

One of the key motivating factors for our work is the fact that DERTs are already being produced and are available on the market. Moreover, they cost only a few dollars (or euros) more than their display-less counterparts. We note that our work and usability studies are also to a small degree relevant to 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

2.1 Secure 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 [33, 14] to schemes that pose more complicated cognitive tasks to users [31, 15].

The authentication of users to passive devices (such as RFID tags) is a very recent issue. In the first proposed solution by Czeckis *et al.* [13], users authenticate to an accelerometer-equipped 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 it 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.* [30] 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 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 secure user-to-tag authentication solution described and tested in this paper is most similar to Abadi *et al.*'s [7] proposal for authentication on 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., [12]), machine learning techniques (e.g., [8]), 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.* in [22]. In this paper, we present a simple solution 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 Reader Revocation Checking

Three popular methods to verify the status of a public key certificate (PKC) are: Certificate Revocation Lists (CRLs) [18], Online Certificate Status Protocol (OCSP) [26] and Certificate Revocation System (CRS) [25, 24]. 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 RFID security and certificate revocation, coupled with the fact that the problem had been spotted by researchers [17, 19, 16], little has been done to address reader PKC revocation and expiration checking problems. Only very recently, Nithyanand *et al.* [28] proposed a method that entails user involvement and DERTs to determine PKC validity. We adopt and experiment with this solution. Although [28] includes a preliminary usability study using a mocked-up implementation on mobile phones, this paper presents a comprehensive analysis of the usability of the method tested using actual DERTs and realistic user tasks.

2.4 Secure 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 [21, 20] and [23] 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 adopt the "Copy" method that was introduced by Uzun *et al.* [32], 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., [10]), which 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.[4]

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).

[4] We used NXP tags with two buttons in our usability tests. One of the button actions can always be substituted with a timeout though.

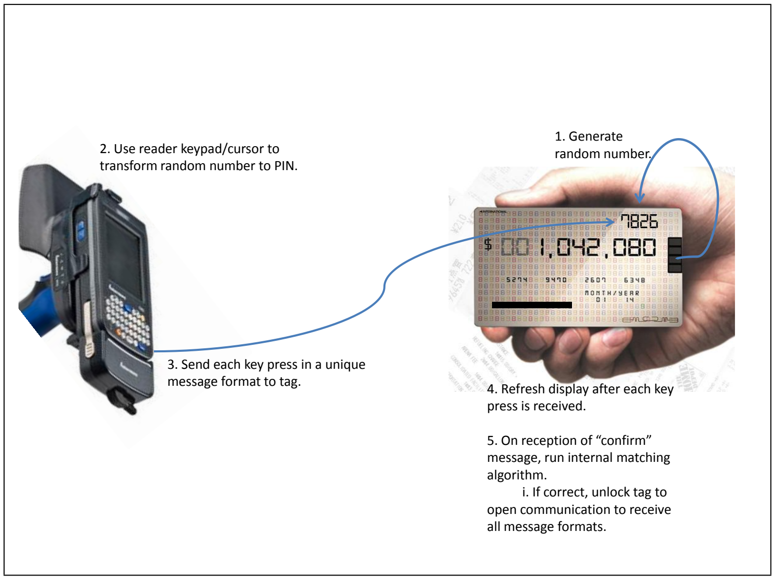

1. Generate random number.
2. Use reader keypad/cursor to transform random number to PIN.
7826
$ 00 1,042,080
3. Send each key press in a unique message format to tag.
4. Refresh display after each key press is received.
5. On reception of "confirm" message, run internal matching algorithm.
i. If correct, unlock tag to open communication to receive all message formats.

Fig. 2. Secure user-to-DERT authentication

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.

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 [7] 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.
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 '$\backslash$' 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 assumption is necessary:

- The user has access to either a printed or a digital (e.g., displayed on the cash register) receipt for the transactions to be verified.

The Protocol (also see Figure 3)

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.

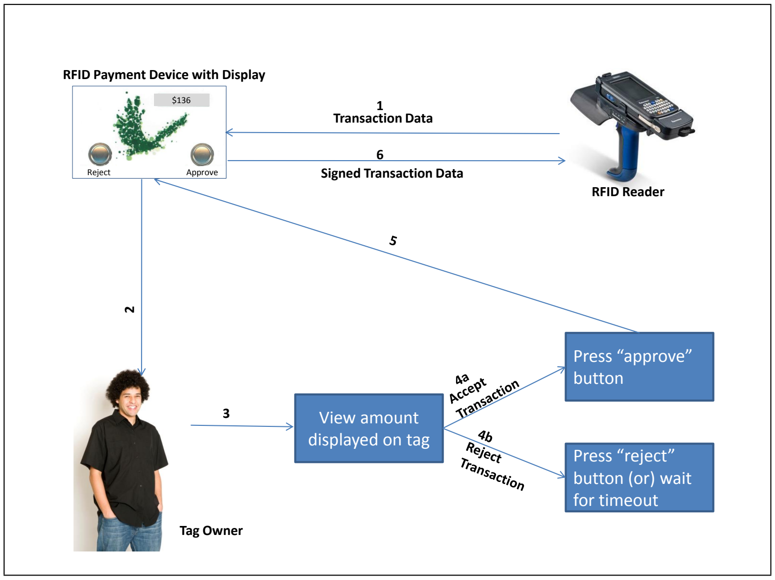


Fig. 3. DERT-enabled transaction verification

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.

3.4 Reader Revocation Status Checking

Our approach for reader certificate expiration and revocation checking [28] 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:

- Tags are aware of the identity and public key of the system-wide trusted Certificate Authority (CA). In other words, all tags and readers are subsumed by a system-wide Public Key Infrastructure (PKI). An example of such a CA is the ICAO CVCA [2].
- The CA is assumed to be infallible: anything signed by the CA is guaranteed to be genuine and error-free.
- The CA periodically (at fixed intervals) 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.

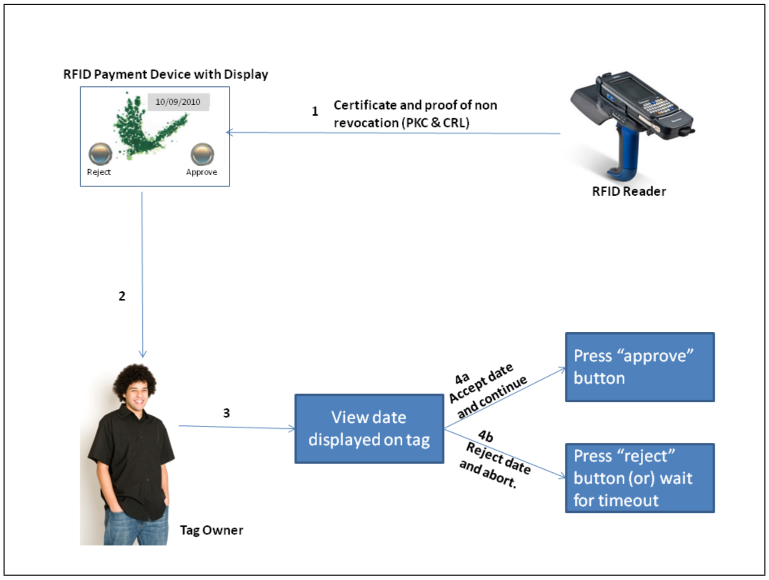

RFID Payment Device with Display
10/09/2010
Reject
Approve
1
Certificate and proof of non revocation (PKC & CRL)
RFID Reader
2
3
View date displayed on tag
4a Accept date and continue
Press "approve" button
4b Reject date and abort.
Press "reject" button (or) wait for timeout
Tag Owner

Fig. 4. Reader certificate expiration/revocation checking

– A tag can retain (in 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.

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 [32] and described in Section 2.

Additional Assumptions. This technique entails the following additional assumption:

- DERT can generate short random passcodes for the purpose of device pairing and can run secret based key agreement protocols, such as [10].

The protocol. The method operates as follows.

1. DERT generates and displays a sufficiently long decimal passcode (e.g., 6-9 digits).
2. The software interface on the other device prompts the user to enter this passcode.
3. Using the (presumably common) passcode, DERT and the second device run an authenticated key agreement protocol to establish a (stronger) common key and confirm its possession by both parties.

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 their eventual user acceptance in real-world deployment. To this end, we conducted a comprehensive usability study with prototype implementations. The goal of the study was to provide answers to the following concrete questions:

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

4.1 Apparatus, Implementation and Setup

Our study was conducted using display-equipped RFID tags (DERTs) from NXP Semiconductors and an HID Omnikey 5321 desktop reader [4]. 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 [5].

All tests were conducted in a designated conference room at a university campus. Participants 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 subjects' task as potential users in each protocol. These tasks were re-explained before each protocol was tested. Participants 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. Participants were then presented with the 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 participant completed a post-test survey. It included the System Usability Scale (SUS) questionnaire [11], a widely used and highly reliable 10-item 5-point Likert scale, and several other questions framed to gain insights into the potential acceptance of the proposed methods.

On average, each person took about 30 minutes to finish the entire series of tests. Everyone was allowed to take part in the study only once. Each participant was rewarded with either an open movie coupon or a $10 Starbucks gift card.

4.2 Subject Background

Our study was conducted over a period of 25 days, in two phases. It involved a total of 35 participants who were chosen on a first-come first-serve basis from the respondents to recruitment emails and flyers. 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, and 33.33% (10 subjects) 30 and over. Gender distribution was nearly even with 53.33% (16 subjects) males and 46.67% (14 subjects) females. The subject pool was extremely 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. 6.67% (2 subjects) reported a disability that impaired their visual perception.

4.3 Test Procedures and Results

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 [6] 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 for 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. The study yielded an average completion time of 38.469 seconds for the regular authentication protocol (UA), and 39.684 seconds for the shoulder-surfing resistant variant (UA-SSR). A paired t-test showed that this difference is not statistically significant. Unfortunately, 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.
- **SUS Scores and Usability Analysis:** The UA protocol was rated at 68.58 out of 100 on the SUS scale, 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 five point Likert scale.
- **Discussion:** An analysis of the completion times and error rates does not yield a clear winner between the UA and UA-SSR protocols. However, the SUS scores and user opinions indicate that UA-SSR is the preferred protocol for users. Post-test subject interviews lead us to believe that the UA-SSR was preferred because of the presence of the '*cursor*' that indicated which digit was currently being manipulated (recall, all digits which were not being manipulated were replaced by a '$\backslash$'). This, however, was not present in the UA protocol, and as a result, subjects often lost track of which digit they were manipulating, causing 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 user 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.

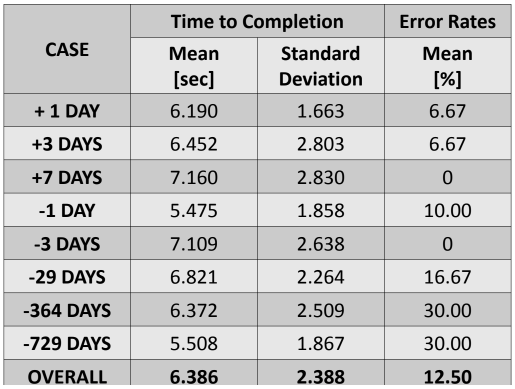
We point out that while other solutions to the user-to-tag authentication problem such as [30] take significantly less time to complete (mean: 7.122 seconds), the error rates are prohibitively high at 78.75%.

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 simulator (with structure and products similar to the Best Buy airport vending machines [1]). Then, each subject was asked to make two separate sets of purchases (each set was a test case). On pressing the *checkout* button on the machine, a digital receipt appeared on the display monitor of the vending machine. Next, the total amount the 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 “honest”. Otherwise, an amount mismatch indicated a malicious vendor attempting to overcharge the user. For each participant, 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. Furthermore, all 30 subjects completed their tasks successfully and no errors were recorded in the process.
- **SUS Scores and User Opinion:** Subjects rated usability at 86 out of 100 on the System Usability Scale (SUS) [11]. This is far above the “industry average” of 70.1 reported in [9], and indicates 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.
- **Discussion:** As the results indicate, our method is unlikely to cause errors. However, we note that this is possibly a consequence of our specific implementation. We anticipate that user errors are likely to arise quite often in real-world deployments if malicious vendors manipulate the placement of decimal points on the DERT (e.g., displaying $344.1 instead of $34.41). We were unable to test this attack in our study since the specific NXP prototype tags that we used are incapable of displaying decimal points. This fact in return prompts us to recommend an implementation such as ours when applicable, since it does not display the fractional part of a number (i.e., cents), thereby making it resistant to such attacks. Such an implementation would not be suitable though if micro-payments (less than a dollar) or attacks at the level of decimal fractions are expected.

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 upper-bound 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 ("+" and "-" indicate future and past dates, respectively). The choices of -29 days, -364 days 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.



CASE	Time to Completion		Error Rates
	Mean [sec]	Standard Deviation	Mean [%]
+ 1 DAY	6.190	1.663	6.67
+3 DAYS	6.452	2.803	6.67
+7 DAYS	7.160	2.830	0
-1 DAY	5.475	1.858	10.00
-3 DAYS	7.109	2.638	0
-29 DAYS	6.821	2.264	16.67
-364 DAYS	6.372	2.509	30.00
-729 DAYS	5.508	1.867	30.00
OVERALL	**6.386**	**2.388**	**12.50**

Fig. 5. Completion times and error rates for various test cases

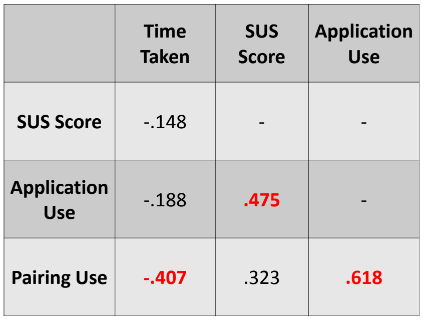
– **Completion Time and Error Rates:** For the 240 (=8*30) test cases, the study yielded an average completion time of 6.386 seconds with a standard deviation of 2.388 seconds (see Figure 4.3). This shows that subjects made quick decisions regarding the timeliness of displayed dates. Among the 240 test 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, namely 17.33% on average. When subjects were shown dates that were 1 and 3 days earlier, the error rates were only 10% and 0%, respectively. Surprisingly though, when subjects were shown dates that were 29, 364 and 729 days earlier, the error rates shot up to 16.67%, 30% and 30%. We will elaborate on possible reasons for this spike in the discussion below.

- **SUS Scores and User Opinion:** Subjects that tested our implementation rated its usability at 76 on the System Usability Scale (SUS) [11]. We note that this is almost identical to the score of 77 obtained in [28], where subjects rated it based on a mock-up implementation on a Nokia N95 cell phone. The overall SUS score obtained is appreciably above the "industry average" of 70.1 [9], 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.
- **Discussion:** As the results show, our method very rarely yields false negatives: users are capable of not mistaking valid (future) dates for past dates. Regarding false positives, however, the 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 may not tend to pay as much attention to more blatant errors such as wrong year, perhaps because they consider it to be an unlikely event. We anticipate though that year mismatches will be quite rare in practice, since (as we 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 user error rates in wrong year values can still pose a threat if a tag is not used for a year or longer.

Secure Device Pairing. We chose the "Copy" method described earlier for the device pairing tests. There were two primary reasons for this choice: our previous studies [32, 27] had indicated low error rates, and the method is device-controlled and therefore resistant to rushed user behavior [29].

- **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 number 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 user.
- **Completion Time and Error Rates:** A total of 30 test cases were performed, yielding an average completion time of 23.904 seconds with a standard deviation of 8.272 seconds. Only 3.33% of the sample (1 subject) entered an incorrect number into the laptop that resulted in an error.
- **SUS Scores and Usability Analysis:** Before rating the pairing protocol on the System Usability Scale, subjects were clearly informed of the distinction between rating the pairing protocol and rating its applications. The SUS scale 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.

Furthermore, 86.67% (26 subjects) indicated that they found the “Copy” method easy to use and that they wanted to use it more often for 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 the ability to pair their personal tags with other devices.



	Time Taken	SUS Score	Application Use
SUS Score	-.148	-	-
Application Use	-.188	**.475**	-
Pairing Use	**-.407**	.323	**.618**

Fig. 6. Pearson correlation coefficient matrix for tag-to-PC pairing

- **Discussion:** High SUS scores, low error rates and positive user feedback point to the 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 measures. To better understand the correlations among four selected measures, we computed their cross correlations. Fig. 6 shows the Pearson correlation coefficients. Interestingly, there exist three medium to high correlations. These are between perceived ease of use of the pairing method and time to completion (medium: -.407), likelihood of using applications of pairing and SUS score (medium: .475), and perceived ease of use of pairing method and likelihood of using applications of pairing (high: .618).

5 Conclusions

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

In this paper, we made the case for using DERTs in several security-related contexts. In particular, we presented simple, intuitive solutions to several security problems with personal RFID tags. Our methods take advantage of the newly available user interface (display) for RFID tags and the presence of human owners. Preliminary usability studies suggest that target users find all our methods usable, and they are capable of performing their roles with reasonably low error rates. As more applications for DERTs are found, we believe that they will soon be in mass production and methods proposed in this paper will become applicable to a wide range of personal RFID tags.

Acknowledgements

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

References

1. Bestbuy To Put Gizmo Vending Machines In Airports. http://www.pcworld.com/article/149684/best_buy_to_put_gizmo_vending_machines_in_airports.html.
2. BSI: Country Verifying Certificate Authority. https://www.bsi.bund.de/cln_174/DE/Themen/ElektronischeAusweise/CVCAePass/CVCAePass_node.html.
3. BSI: The New ID-Card. https://www.bsi.bund.de/cln_174/ContentBSI/Themen/Elekausweise/Personalausweis/ePA_Start.html.
4. Hid Omnikey 5321 Cl Usb Reader. http://www.hidglobal.com/documents/OK5321_cl_ds_en.pdf.
5. Java Smart Card I/O. http://java.sun.com/javase/6/docs/jre/api/security/smartcardio/spec/.
6. Logitech Wireless N305. http://www.logitech.com/en-us/keyboards/keyboard/devices/6355.
7. M. Abadi, C. Burrows, C. Kaufman, and B. Lampson. Authentication and delegation with smart-cards. *Science of Computer Programming*, 21(2):93–113, 1993.
8. 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, 23-25 1997.
9. 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.
10. 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.
11. J. Brooke. SUS: A "Quick And Dirty" Usability Scale. In P. W. Jordan, B. Thomas, B. A. Weerdmeester, and A. L. McClelland, editors, *Usability Evaluation in Industry*. Taylor and Francis, London, 1996.
12. P. K. Chan, W. Fan, A. L. Prodromidis, and S. J. Stolfo. Distributed Data Mining In Credit Card Fraud Detection. *IEEE Intelligent Systems*, 14(6):67–74, 1999.
13. 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.
14. A. Evans, Jr., W. Kantrowitz, and E. Weiss. A User Authentication Scheme Not Requiring Secrecy In The Computer. *Commun. ACM*, 17(8):437–442, 1974.
15. 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.
16. T. S. Heydt-Benjamin, D. V. Bailey, K. Fu, A. Juels, and T. O'Hare. Vulnerabilities In First-Generation RFID-Enabled Credit Cards. In *Financial Cryptography*, pages 2–14, 2007.

17. J.-H. Hoepman, E. Hubbers, B. Jacobs, M. Oostdijk, and R. W. Schreur. Crossing Borders: Security And Privacy Issues Of The European E-Passport. In *IWSEC*, pages 152–167, 2006.
18. R. Housley, W. Ford, W. Polk, and D. Solo. Rfc 5280: Internet X.509 Public Key Infrastructure Certificate and CRL profile, May 2008.
19. 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.
20. R. Kainda, I. Flechais, and A. W. Roscoe. Usability And Security Of Out-Of-Band Channels In Secure Device Pairing Protocols. In *SOUPS: Symposium on Usable Privacy and Security*, 2009.
21. 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.
22. Y. Kou, C.-T. Lu, S. Sirwongwattana, and Y.-P. Huang. Survey Of Fraud Detection Techniques. In *Networking, Sensing and Control, 2004 IEEE International Conference on*, volume 2, pages 749 – 754 Vol.2, 2004.
23. 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.
24. S. Micali. Efficient Certificate Revocation. Technical Memo MIT/LCS/TM-542b, Massachusetts Institute of Technology, 1996.
25. S. Micali. Certificate Revocation System. United States Patent 5,666,416, Sept. 1997.
26. 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.
27. 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.
28. 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.
29. N. Saxena and M. B. Uddin. Secure Pairing Of “Interface-Constrained” Devices Resistant Against Rushing User Behavior. In *International Conference on Applied Cryptography and Network Security (ACNS 2009)*, 2009.
30. N. Saxena, M. B. 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*, pages 1–1, New York, NY, USA, 2009. ACM.
31. P. Toni, C. Mario, and S. Nitesh. Shoulder-surfing Safe Login in a Partially Observable Attacker Model. In *Financial Cryptography and Data Security*, volume 6052, pages 351–358, 2010.
32. 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*, pages 307–324, Berlin, Heidelberg, 2007. Springer-Verlag.
33. M. V. Wilkes. *Time Sharing Computer Systems*. Elsevier Science Inc., New York, 1975.