User-aided Reader Revocation in PKI-Based RFID Systems [*]

Rishab Nithyanand [†] Gene Tsudik [‡] Ersin Uzun [§]

Abstract

Recent emergence of RFID tags capable of performing public key operations motivates new RFID applications, including electronic travel documents, identification cards and payment instruments. In this context, public key certificates form the cornerstone of the overall system security. In this paper, we argue that one of the prominent challenges is how to handle revocation and expiration checking of RFID reader certificates. This is an important issue considering that these high-end RFID tags are geared for applications such as e-documents and contactless payment instruments. Furthermore, the problem is unique to public key-based RFID systems, since a passive RFID tag has no clock and thus cannot use (time-based) off-line methods.

In this paper, we address the problem of reader certificate expiration and revocation in PKI-Based RFID systems. We begin by observing an important distinguishing feature of *personal* RFID tags used in authentication, access control or payment applications – the involvement of a human user. We take advantage of the user's awareness and presence to construct a simple, efficient, secure and (most importantly) feasible solution. We evaluate the usability and practical security of our solution via user studies and discuss its feasibility.

1 Introduction

Radio Frequency Identification (RFID) is a wireless technology mainly used for identification of various types of objects, e.g, merchandise. An RFID tag is a passive device, i.e., it has no power source of its own. Information stored on an RFID tag can be read by special devices called RFID readers, from some distance away and without requiring line-of-sight alignment. Although RFID technology was initially envisaged as a replacement for barcodes in supply chain and inventory management, its many advantages have greatly broadened the scope

[*] The preliminary version of this paper appeared in the proceedings of ESORICS 2010.

[†] Stony Brook University, email: {rnithyanand@cs.stonybrook.edu}

[‡] University of California - Irvine, email: {gts@ics.uci.edu}

[§] Palo Alto Research Center, email: {ersin.uzun@parc.com}

of possible applications. Current and emerging applications range from visible and personal (e.g., toll transponders, passports, credit cards, access badges, livestock/pet tracking devices) to stealthy tags in merchandise (e.g., clothes, pharmaceuticals and library books). The cost and capabilities of an RFID tag 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 each costing a few (usually < 10) dollars. These tags are powerful enough to perform public key cryptographic operations.

In the "real world", one of the main security issues in using public key cryptography is certificate revocation. Any certificate-based public key infrastructure (PKI) needs an effective revocation mechanism. Revocation can be handled implicitly, via certificate expiration, or explicitly, via revocation status checking. Most PKI-s use a combination of implicit and explicit methods. The latter can be done off-line, using Certificate Revocation Lists (CRLs) [11] and similar structures, or on-line, using protocols such as Online Certificate Status Protocol (OCSP) [25]. However, as discussed later in this section, these approaches are untenable in public key-enabled RFID systems.

Intuitively, certificate revocation in RFID systems should concern two entities: RFID tags and RFID readers. The former only becomes relevant if each tag has a "public key identity", i.e., if each tag has its own public/private keypair and (optionally) a public key certificate (PKC) binding its identifier to a public key. We claim that revocation of RFID tags is a non-issue, since, once a tag identifies itself to a reader, the latter (as the entity performing a revocation check) can use any current revocation method –except perhaps OCSP which requires full-time connectivity. This is because an RFID reader is a full-blown computing device with ample power, memory, various communication interfaces and its own clock.

In contrast, revocation of readers is a problem in any public key-enabled RFID system. A tag may or may not have public key identity but a reader must have one; otherwise, the use of public key cryptography becomes non-sensical. Therefore, before a tag discloses any information to a reader, it must make sure that the reader's public key certificate (PKC) is not expired or revoked.

1.1 Why Bother?

One common and central purpose of all RFID tags and systems is to enable tag identification (at various levels of granularity) by readers. With that in mind, many protocols have been proposed to protect the identification process (i.e., the tag-reader dialog) from a range of attacks. In systems where tags can not perform cryptographic operations or where they are limited to symmetric cryptography, reader revocation is not an issue, since it is essentially impossible. Whereas, in the context of public key-enabled tags, reader revocation is both imperative and possible, as we show later in this paper. It is imperative, because not doing it prompts some serious threats. For example, consider the following events: a reader is *lost*, *stolen*, *compromised* (perhaps without its owner's knowledge), or *decommissioned*.

In all of these cases, if it cannot be revoked effectively, a reader that has fallen into the wrong hands can be used to identify and track tags. In case of personal tags – e.g., ePassports, credit-cards or eIDs – other threats are possible, such as identity theft or credit card fraud. For example, in the case of tags used in payment instruments, an adversary may obtain financially valuable information (such as credit-card number, expiration date, owner name, etc.), which can later be sold for money [39], used for creating forged non-RFID credit cards (using the data obtained from the illegal reader to reproduce the magnetic strip of a non-RFID card) or to commit other types of credit fraud.

Thus far, it might seem that our motivation is based solely on the need to detect *explicitly revoked* reader certificates [1]. However, what if a reader certificate naturally expires? This indicates *implicit revocation* and a well-behaved reader would not be operated further until a new certificate is obtained. However, if a reader (or rather its owner) is not well-behaved, it might continue operation with an expired certificate. Without checking certificate expiration, an unsuspecting tag could be tricked into identifying itself and possibly divulging other sensitive information.

In the remainder of this paper, we make no distinction between explicit revocation (i.e., revocation before expiration) and implicit revocation (i.e., certificate expiration) checking. The reason is that both tasks are essential for security and both require current time.

1.2 Why Is Reader Revocation Hard?

When presented with a PKC of a reader, a tag needs to check three things: (1) *signature* of the issuing certification authority (CA), (2) *expiration* and (3) *revocation status*.

The first is easy for any public key-enabled (pk-enabled) tag and has been already incorporated into some reader authentication schemes [5], [13]. However, (2) and (3) are problematic. Note that even a high-end tag is a passive device lacking a clock. Thus, a tag, by itself, has no means of deciding whether a presented certificate is expired.

Checking revocation status is even more challenging. First, similar to expiration, off-line revocation checking (e.g., CRL-based) requires current time because the tag needs to check the timeliness of the presented proof of non-revocation. Also, communicating a proof of non-revocation entails extra bandwidth from the reader to the tag. For CRLs, the bandwidth is $O(n)$ and, for more efficient CRTs, the bandwidth is $O(log\ n)$ – a non-negligible number for large values of n, where n is the number of revoked readers[2]. Whereas, online revocation checking protocols (e.g., OCSP) offer constant-size proofs of non-revocation. However, such protocols would entail the tag contacting (via the reader) a trusted OCSP responder. The proof of non-revocation would be

[1] "Explicitly" means before the expiration of the PKC.

[2] The problem of the high communication cost of CRL-s in current solutions has been noted by Blundo, et al. [3].

constant-size, but for many RFID systems the real-time connectivity to an authorized OCSP responder would be problematic. Since an OCSP responder is accessed over some network, the readers must always have a high-speed and low-delay connection to a network infrastructure and OCSP responders are required to be always available.

There have been revocation handling proposals that attempted to compensate for lack of a clock on an RFID tag. For example, [35] suggested using a simple monotonically increasing time-stamp which is updated after every successful tag-reader interaction to the reader's PKC issuance date. This method is adopted by the German Federal Office for Information Security (BSI) for certificate validation in e-Passports [5]. Whenever a tag is presented with a signed certificate or a CRL, it compares the date of expiry with the stored time-stamp and accepts it only if the certificate's expiration date exceeds the time-stamp. However, as pointed out in section 3, this approach does not solve the problem as it leaves a large window of vulnerability between time-stamp updates. This is especially problematic in case of infrequently used tags, such as e-Passports.

1.3 Roadmap

We focus on a class of pk-enabled RFID systems where tags are both personal and attended. This includes e-Passports, e-Licenses and contactless credit cards. *Personal* means that a tag belongs to a human user and *attended* means that a tag is supposed to be activated only with that user's (owner's) consent. Our approach is based on several observations:

- User/owner presence and (implicit) consent are already required for the tag to be activated.
- Low-cost and low-power flexible display technology is a reality, e.g., e-paper and OLED. In fact, passive RFID tags with small (6-10 digit) displays have been demonstrated and are currently feasible.
- Since certificate revocation and expiration granularity is usually relatively coarse-grained (i.e., days or weeks, but not seconds or minutes), users can distinguish between timely and stale date/time values.

The rest is straight-forward: a display-equipped tag receives, from a reader, a PKC along with a signed and time-stamped proof of non-revocation. After verifying the respective signatures on the reader's PKC and the non-revocation proof, the tag displays the lesser of: (1) PKC expiration time and (2) non-revocation proof expiration time. The user, who is assumed to be reasonably aware of current time, validates the timeliness of the displayed time. If it is deemed to be stale, the user aborts the interaction with the reader. Otherwise, user allows the interaction to proceed.

1.4 Organization

We summarize related work in Section 2 and overview some trivial solutions in Section 3. We describe our approach in Section 4, followed by results of the usability study in Section 5. We conclude the paper by discussing application feasibility of our method to ePassports in Section 6 and finally, a summary in Section 7.

2 Related Work

There are many ways of handling certificate revocation. Of these, Certificate Revocation Lists (CRLs) are the most commonly used mechanism. Notably, CRLs are used by the X.509 Public Key Infrastructure for the Internet [11]. Some techniques improve the efficiency of revocation checking. Certificate Revocation Trees (CRTs) [18] use Merkle's Hash Trees [22] to communicate a relatively short non-revocation proofs (of size log n). Skip-lists [8] and 2-3 Trees [26] improve on the CRT update procedure through the use of dynamic data structures, offering asymptotically shorter proofs. Online Certificate Status Protocol (OCSP) [25] is an on-line method that reduces storage requirements and provides timely revocation status information. Certificate Revocation System (CRS) [23] offers fully implicit certificate revocation by placing the bulk of revocation burden on the prover (certificate owner) and yields compact proofs of certificate validity.

In spite of substantial prior work in both certificate revocation and RFID security, very little has been done with respect to reader revocation and expiration checking. However, the problem has been recognized in previous literature [24, 10, 14, 9, 6, 29].

As mentioned earlier, our solution to the problem of reader revocation checking takes advantage of the presence and awareness of a human user. Another security challenge that has been solved by such an approach is – secure device pairing [33, 34, 30]. Usability studies conducted in the context of device pairing indicate that humans are reliable and capable of performing their roles in these security protocols within reasonable error rates.

In this paper, we propose using a small display on tags to solve the problem of revocation checking. To the best of our knowledge, the idea of outfitting pk-enabled RFID tags with display units for enhanced security was introduced by Ullman [36] to establish secure and authenticated wireless channels using short passwords.

Our approach does not have a large window of vulnerability (beyond that already inherent to any off-line revocation method). Furthermore, it is very efficient in terms of reader-tag bandwidth and tag storage.

3 Trivial Solutions

We now consider some trivial reader revocation techniques and discuss their shortcomings.

3.1 Date Register & Time Stamps

Every PKC has a validity period defined by its effective date (D_{eff}) and expiration date (D_{exp}). During certificate verification, a tag can use the date stored in its register (D_{curr}) to determine whether a certificate has expired. Verification steps are as follows:

1. Tag verifies the CA signature of the reader's certificate.
2. Tag checks that D_{exp} is greater than D_{curr}.
3. If (1) and (2) succeed, the tag accepts the certificate. If D_{eff} is greater than D_{curr}, the tag updates D_{curr} to D_{eff}.

With this approach, the estimate of the current date – D_{curr} – stored by the tag is not guaranteed to be accurate and thus can not always protect it from readers with expired or revoked certificates. This is especially the case for a tag that has not been used for some time. The value of D_{curr} might reflect a date far in the past, exposing the tag to attacks from readers revoked at any point after D_{curr}. Even for tags that are used relatively more frequent, a recently revoked reader would always pose a danger.

3.2 On-line Revocation Checking

Online revocation-checking approaches, such as OCSP [25], alleviate client storage requirements by introducing trusted third parties (responders) that provide on-demand and up-to-date certificate status information. To validate a certificate, a client sends an OCSP status request to the appropriate responder and receives a signed status. In its basic form, OCSP requires a clock on the client, as it uses time-stamps to assure freshness. However, an optional OCSP extension supports the use of nonces as an alternative.

Although suitable for a large and well-connected infrastructure, such as a private network or the Internet, OCSP is problematic in RFID systems. Its use would require a tag to generate random challenges and conduct a 2-round (on-line) challenge-response protocol with an OCSP responder. This would necessitate constant infrastructure connectivity for all readers and availability of OCSP responders. Moreover, the turnaround time for tag-reader interaction would become dependent on external factors, such as congestion of the communication infrastructure (e.g., the Internet) and current load on OCSP responders. Either factor might occasionally cause significant delays and the responders present a single point of failure, if not carefully implemented.

3.3 Internal Clocks

An internal clock would allow tags to accurately determine whether a certificate is expired and whether a non-revocation proof is current. However, a typical RFID tag is a purely passive device powered by radio waves emitted from a nearby reader. Since a real-time clock needs uninterrupted power, it cannot be sustained by passive tags. One might consider equipping RFID tags with batteries, however, this raises a slew of new problems, such as battery cost, clock synchronization and battery replacement.

4 Proposed Technique

We re-emphasize that our approach is aimed only at pk-based RFID systems. It has one simple goal: secure and reliable revocation checking on RFID tags. In the rest of this section, we discuss our assumptions and details of the proposed solution.

4.1 Assumptions

Our design entails the following assumptions[3]:

1. Each tag is owned and physically attended by a person who understands tag operation and who is reasonably aware of the current date.

2. Each tag is equipped with a one-line alpha-numeric (OLED or ePaper) display capable of showing a 6-8 digit date. (Figure 1 shows an example of such a tag manufactured by NXP Semiconductors)

3. Each tag has a mechanism that allows it to become temporarily inaccessible to the reader (i.e., to be "turned off").

4. Each tag is aware of the name and the public key of a system-wide trusted certification authority (CA).

5. The CA issues an updated revocation structure (e.g., a CRL) periodically. It includes serial numbers of all revoked reader certificates.

6. Each tag knows the periodicity of revocation issuance (i.e., it can calculate the expiration date of revocation status information by knowing its issuance date.)

7. While powered up by a reader, a tag is capable of maintaining a countdown timer.

8. A tag can retain (in its non-volatile storage) the last valid date it encountered.

[3] Although we use ”date” as the revocation/expiration granularity, proposed technique is equally applicable to both coarser- and finer-granular measures, e.g., month or hour.

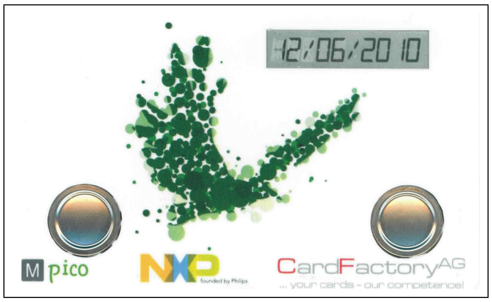


Figure 1: A Display and Button Equipped RFID Tag from NXP Semiconductors

9. [**OPTIONAL**] A tag has some mechanism *(i.e., a button)* that permits user input.

4.2 Basic Idea

Before providing any information to the reader, a tag has to validate the reader PKC. Recall our assumption that the user is physically near (e.g., holds) his tag during the entire process. Verification is done as follows:

1. The freshly powered-up tag receives the CRL and the reader PKC. Let CRL_{iss}, CRL_{exp}, PKC_{iss} and PKC_{exp} denote issuance and expiration times for purported CRL and PKC, respectively. Let the last valid date stored in the tag be Tag_{Curr}.

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

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

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

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

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

7. The user views the date on the display.
   [**OPTION A:**]

   (a) If the displayed date is not in the past, the user does nothing and interaction between the tag and the reader resumes after the countdown stage.

(b) Otherwise, the user terminates the protocol by initiating an escape action while the tag is still in countdown stage.

[**OPTION B:**] (If Assumption 9 holds)

(a) If the displayed date is in the future, the user indicates acceptance to the tag (e.g., by pressing a button on the tag) before the timer runs out, and communication with the reader continues normally.

(b) Otherwise, the timer runs out and the tag automatically aborts the protocol.

4.3 Escape Actions

As evident from the above, an escape action is required whenever the user decides that the displayed date is stale. These escape actions prevent malicious readers from gaining access to sensitive information stored on a tag. Although the choice of an escape action is likely to be application-dependent, we sketch out several simple and viable examples.

4.3.1 Using a Button

Recent developments in low-power hardware integration on contactless cards have led to deployment of buttons on RFID tags [19, 37]. On such tags, the user can be asked to press a button (within a fixed interval) as a signal of acceptance[4]. If the button is not pressed within that interval, the protocol is automatically terminated by the tag. Thus, the escape action in this case involves no explicit action by the user. We recommend this variant over alternatives discussed below, since it complies with the *safe defaults* design principle, i.e., without explicit approval by the user, the tag automatically aborts its interaction with the reader.

4.3.2 Faraday Cages

A Faraday Cage is a jacket made of highly conductive material that blocks external electric fields from reaching the device it encloses. Since tags are powered by the electric field emitted from a reader, it is theoretically [5] possible to isolate them from all reader access by simply enclosing them in a Faraday cage. For tags that have an enclosing Faraday cage – such as e-Passports that have one inside their cover pages – the natural escape action is simply closing the passport.

[4] For tags that have no buttons but built-in accelerometers, gestures (see [7] for more details) can also be used to signal user acceptance.

[5] A rump session talk at PETS'09 shed some doubts on today's Faraday Cage-enclosed e-passports.

4.3.3 Disconnecting Antennas

An RFID tag communicates and receives power through a coil antenna attached to its chip. Disconnecting the antenna from the chip immediately halts communication and shuts down the tag. A simple physical switch (even a mechanical one, e.g, a slide-switch operated by a finger) placed between a tag and its antenna can be used as an escape action. Similar mechanical actions aimed to halt communication between a tag and a reader are described in [16]. One drawback of such techniques is that physical damage to the tag is possible if the switch is handled roughly.

4.4 Efficient Revocation Checking

Although we hinted at using CRLs earlier in the paper, our approach would work with CRTs or any other off-line revocation scheme. However, both CRLs and CRTs become inefficient as the number of revoked readers increases. CRLs are linear and CRTs – logarithmic, in the number of revoked certificates. Our goal is to minimize bandwidth consumed by revocation information by making it constant, i.e, $O(1)$. To achieve this, we take advantage of a previously proposed modified CRL technique originally intended to provide privacy-preserving revocation checking [27].

In traditional CRLs, the only signature is computed over the hash of the entire list of revoked PKCs. Consequently, the entire list must be communicated to the verifier. To make CRLs bandwidth-optimal, [27] requires the CA or a Revocation Authority to sign each (sorted) entry in a CRL individually and bind it with the previous entry. In more detail, the modified CRL technique works as follows: assume that the CRL is sorted in ascending order by the revoked certificate serial numbers. For a CRL with n entries, the CA generates a signature for the i-th entry ($1 < i \leq n$) as follows:

$$Sign(i) = \{h(CRL_{iss}||SN_i||SN_{i-1})\}_{SK_{RA}}$$

where, CRL_{iss} is the issuance date of this current CRL, SN_i is the i-th certificate serial number on the ordered CRL, SN_{i-1} is the immediately preceding revoked serial number, SK_{RA} is the secret key of the CA and h is a suitable cryptographic hash function. To mark the beginning and the end of a CRL, the CA uses two fixed sentinel values: $+\infty$ and $-\infty$.

When authenticating to a tag, a non-revoked reader provides its own PKC as well as the following constant-size non-revocation proof:

$$SN_j,\ SN_{j-1},\ CRL_{iss},\ Sign(j)$$

where reader certificate serial number SN_{rdr} is such that $SN_{j-1} < SN_{rdr} < SN_j$. The reader PKC, along with the above information, allows the tag to easily check that: (1) the range between adjacent revoked certificate serial numbers contains the serial number of the reader PKC, and (2) the signature $Sign(j)$ is valid. If both are true, the tag continues with the protocol by displaying the lesser of the CRL_{exp} and PKC_{exp}, as in step 6 of Section 4.2.

As discussed below, this scheme requires lower storage and communication overhead compared to traditional CRLs. The overhead comparison is summarized in Table 1.

Table 1: Modified CRLs vs. Traditional CRLs

	Modified CRLs	Traditional CRLs
Storage and Communication	Readers store and communicate 1 signature, 2 certificate serial numbers, and 1 date. Communication overhead is $O(1)$.	Readers store and communicate an entire list of certificate serial numbers, 1 signature, and 1 date. Communication overhead is $O(n)$.
Computation	CA has to separately sign each CRL entry. Tags require 1 signature verification.	CA has to sign the entire CRL only once. Tags require 1 signature verification.

4.4.1 Storage Overhead

With traditional CRLs, readers must store entire lists of revoked certificate numbers. This can cause significant storage overhead. In the above method, storage overhead for both readers and tags is negligible since only one signature, two certificate serial numbers and the issuance date are needed for effective revocation checking.

4.4.2 Computational Overhead

The modified CRL method calls for the CA to separately sign each CRL entry, whereas, only one signature is needed for a traditional CRL. Although this translates into significantly higher computational overhead for the CA, we note that CAs are powerful entities running on resource-rich systems and CRLs are not usually re-issued very frequently, i.e., weekly or daily, but not every minute or even every hour. Computation overhead for tags is minimal in the modified CRL scheme. Verifying traditional CRLs requires hashing $O(n)$ serial numbers, in contrast to hashing a constant-length tuple in modified CRLs. On the other hand, both methods require one signature verification which usually overshadows the cost of hashing.

4.4.3 Communication Overhead

CRLs impose linear communication overhead, whereas, the modified CRL method is bandwidth-optimal, requiring only the transmission of two serial numbers, issuance date and a signature.

4.5 Security Considerations

Assuming that all cryptographic primitives used in the system are secure and the user executes necessary escape actions in case of expired (or revoked) reader certificates, the security of the proposed reader revocation checking mechanism is evident.[6]

We acknowledge that user's awareness of time and ability to abort the protocol (when needed) are crucial for the overall security. To this end, we conducted some usability studies, including both surveys and experiments with a mock implementation. As discussed in section 5, our studies showed that people are reasonably aware of date and also able to execute the protocol with low error rates. At the same time, awareness of date/time among general population is known to be quite universal [38]. Thus, we can assume that people, especially those who might be exposed to this technology, are reasonably aware of current date and time (especially in the case of e-Passport reader revocation checking, since people are more likely to remember the date with much better accuracy while traveling). Although human errors on the order of hours are to be expected, this is not a problem for most RFID systems since revocation update periods are usually measured at least in days, or more commonly in weeks or months.

Another critical assumption about, and requirement for, the user is the undivided attention during the reader authentication process and the ability to react whenever a stale expiration or revocation date is observed. However, we believe that users can be educated – e.g., via manuals and warning labels – about the meaning of their participation in the protocol and operation procedure of their tags. Taking the *safe default* approach in reader authentication, where explicit user approval is required before disclosing any sensitive information, would also help eliminate security critical user errors. Utilizing a button for date acceptance on the tag is a good example of *safe default* design approach.

4.6 Cost Assessment

Recent technological advances have enabled mass production of small inexpensive displays (e.g., ePaper) that can be easily powered by high-end RFID tags aided by nearby readers. The current (total) cost of an ePaper display-equipped and public key-enabled RFID tag is about 17 Euros in quantities of $100,000$ and the cost goes down appreciably in larger quantities [37]. Although this might seem high, we anticipate that the cost of cutting-edge passive display technologies (i.e., ePaper and OLED) will sharply decrease in the near future. Moreover, once a display is available, it can be used for other purposes, thus amortizing the expense. We briefly describe some potential alternative uses for display-equipped RFID tags:

4.6.1 Transaction Verification

RFID tags are commonly used as payment and transaction instruments (e.g., credit, ATM and voting cards). In such settings, a direct auxiliary channel be-

[6] Man-in-the-middle (or evil-twin) attacks are not a real threat here as both devices share a common CA in PKI-based RFID systems and an interaction with a properly authorized (i.e., certified) device is not a threat as long its privileges are not expired or revoked.

tween the tag and the user is necessary to verify the details of a transaction. This problem becomes especially apparent with payment applications. A malicious reader can easily fool the tag into signing or authorizing a transaction for an amount different from that communicated to the user. A display on a contactless payment card would solve this problem by showing the transaction amount requested by the reader on its display and waiting for explicit user approval before authorizing it.

4.6.2 Device Pairing

A display may be used for secure pairing of tags with other devices that do not share a CA with the tag. Visual channel-based secure device pairing methods that are proposed for personal gadgets can be used with display-equipped RFID tags (See [20] and [17] for a survey of such methods). The ability to establish a secure ad-hoc connection with arbitrary devices is a new concept for RFID tags that might open doors for new applications, e.g., the use of NFC-capable personal devices (e.g., cell-phones) to change and control settings on personal RFID tags.

4.6.3 User/Owner Authentication

In some scenarios, it might be necessary for a user to authenticate to a tag (e.g., credit card or passport). Currently this can be done only via trusted third party devices such as readers, mobile phones [31], personal computers and wearable beepers [15]. However, in the future, with a display-equipped RFID tag, the need for additional trusted devices might be obviated.

5 Usability

Since the proposed technique requires active user involvement, its usability is one of the key factors influencing its potential acceptance. Also, due to the nature of the protocol, certain type of user errors (i.e., accepting an incorrect or stale date) can result in a loss of security. Thus, we conducted three separate usability studies: online surveys and two sets of hands-on usability experiments. The goal of these studies was to answer the following questions:

1. Do everyday users worry about the reader revocation problem?

2. How do these users rate the usability of our solution?

3. Are users reasonably aware of the current date? What are the expected error rates?

5.1 Usability Experiments

In order to assess the usability of our method in the context of real users, two sets of experiments were conducted.

The first study involved a total of 30 participants at the University of California - Irvine, where the second study involved a total of 21 participants from Stony Brook University. The 21 subjects were mutually exclusive from the subjects of the first experiment and the set of tests varied, as described in the following sections.

In order to prevent subjects from being explicitly aware of the date during the tests, care was taken to avoid setting up prior test appointments for both experiments. Instead, subjects were recruited by the test coordinator at various campus venues, e.g., cafés, dorms, classrooms, offices, labs and other similar settings.

5.1.1 Apparatus and Implementation

Our study was conducted using a display-equipped RFID tags (DERT-s) from NXP Semiconductors and an HID Omnikey 5321 desktop reader. DERT-s 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. The time period for the automatic reject was set to 10 seconds.

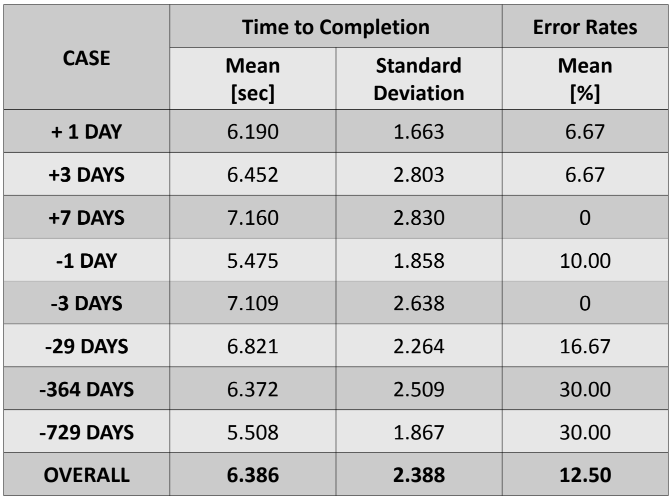
5.1.2 Subjects

For the first 30 subjects who took part in the first experiment, age was well distributed in three groups: 30% – 18 to 24, 36.67% – 25 to 30, and 33.33% – 30 and over. Gender distribution was nearly even with 53.33% and 46.67% male and female subjects, respectively. The subject pool was well-educated, with 86.67% having a bachelors degree or higher (we attribute this to the specifics of the study venue).

For the 21 subjects who took part in the second experiment, age was distributed into three groups: 28.57% – 18 to 24, 38.10% – 25 to 29, 33.33% – 30 and over. Gender distribution was split at 61.90% and 38.10% for male and female subjects, respectively. The subject pool was again well educated, with 76.19% having a bachelors degree or higher.

5.1.3 Procedure

To help subjects in understanding the concept of personal RFID tags, the ePassport example was used throughout the test and the questionnaire phases. First, subjects were asked not to consult any source of current date/time before and during the tests. Then, they were given a brief overview of our method and the importance of maintaining natural behavior during the experiments. Next, each subject was presented with a mock-up implementation and was asked to execute the protocol multiple times. Finally, subject opinions were solicited via the post-test questionnaire.



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**

Figure 2: Completion times and error rates for various test cases in Experiment 1

Experiment # 1:

- **Test procedure:** Each subject was presented with eight test cases in a random order, in the MM/DD/YYYY format. These corresponded to 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 deceiving to the subjects. After a date was displayed on the tag, 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, as also shown in Fig. 2, the study yielded average completion time of 6.386 seconds, with the standard deviation of 2.388 seconds. This shows that subjects made quick decisions regarding the timeliness of displayed dates. Among 30 subjects, 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 two subjects (6.67% of the sample) rejected dates that were one and three days in the future. The false positive rate (accept stale dates), on the other hand, was higher at 17.33%. When subjects were shown dates that were, respectively: 1 and 3 days earlier, the corresponding error rates were 10% and 0%. However, surprisingly, when subjects were shown dates that were, respectively: 29, 364, and 729 days earlier, the corresponding error rates shot up to 16.67%, 30%, and

30%. We will elaborate on possible reasons for this spike in error rates 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) [4], 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 above the "industry average" of 70.1 reported in [2], and indicates good usability and acceptability characteristics.

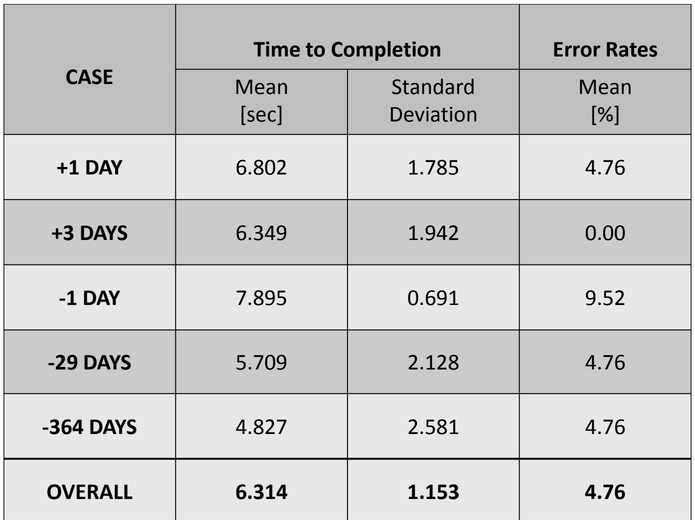
  Furthermore, 70% of the subjects stated that they would like this system implemented on their own personal tags, while 23.33% were neutral to the idea; the average score on a 5-point Likert scale was 3.78 with the standard deviation of 0.77.

- **Discussion:** As results show, our method very rarely yields false negatives: users are capable of not mistaking valid (future) dates for being in the past. As far as false positives are concerned, however, 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%. This deserves a closer look. While we do not claim to know the exact reason(s), some conjectures can be made. When confronted with a date, most people are conditioned to first check day and month, e.g., current dates on documents and expiration dates on perishable products. At the same time, users do not tend to pay as much attention to more gross or blatant errors (such as wrong year) perhaps because they consider it to be an unlikely event. This inadvertently conditions the subjects to pay more attention to the month/day fields of the dates. However, we anticipate that year mismatches will be quite rare in practice, since (as we mentioned earlier in the paper) tags 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. To analyze this further, we conducted a second series of tests with other date formats and even a slightly modified protocol.

Experiment # 2:

- **Test procedure:** Each subject was presented with nine cases as part of two different tests (with 5 and 4 cases, respectively).

  First we presented each subject with 5 test cases, in the YYYY/MM/DD format. This was done to understand if this format resulted in less errors stemming from accepting stale year values. The displayed dates were: +/- 1 day, +3 days, -29 days, and -364 days from the actual test dates. After the date was displayed on the tag, the subjects were (as before), asked to either accept or reject the date by pressing the appropriate button on the



CASE	Time to Completion		Error Rates
	Mean [sec]	Standard Deviation	Mean [%]
+1 DAY	6.802	1.785	4.76
+3 DAYS	6.349	1.942	0.00
-1 DAY	7.895	0.691	9.52
-29 DAYS	5.709	2.128	4.76
-364 DAYS	4.827	2.581	4.76
OVERALL	**6.314**	**1.153**	**4.76**

Figure 3: Completion times and error rates for YYYY/MM/DD test cases in Experiment 2

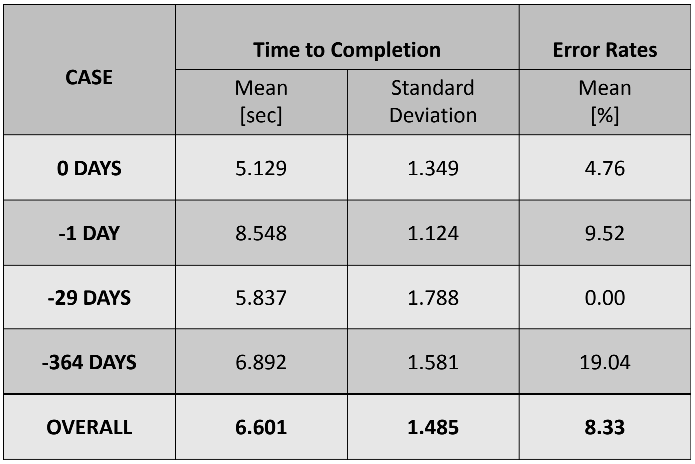
tag. The *safe default* timeout of 10 seconds was maintained, and the date was automatically rejected if no user input was provided within this time.

For the second test, each subject was presented with four test cases which tested the following slightly modified protocol for reader revocation. Here, we showed *today's date* to the users instead of an expiration date. This approach was simulating the following modified protocol:

1. The reader sends the tag its claimed value for "today's date" (D_{curr}) in addition to its PKC and the most recent CRL.
2. The tag checks that $D_{curr} < PKC_{exp}$ and $D_{curr} < CRL_{exp}$. If either check fails, the tag aborts.
3. The tag displays D_{curr} to the user.
4. The user is now required to verify that the displayed date is indeed "today's date" by pressing the button within 10 seconds, otherwise tag terminates the protocol.

The four dates shown (in random order) on the tag were: the actual test date, -1 day, -31 days, and -365 days from the test date.

- **Completion Time and Error Rates:** For the first 105 (= 5*21) test cases where the effect of the $YYYY/MM/DD$ format was studied, as shown in Figure 3, the study yielded an average completion time of 6.314 seconds, with a standard deviation of 1.153 seconds, indicating to us that subjects were capable of making quick decisions regarding the timeliness of the displayed dates and further, that our safe default of 10 seconds was sufficient.



CASE	Time to Completion		Error Rates
	Mean [sec]	Standard Deviation	Mean [%]
0 DAYS	5.129	1.349	4.76
-1 DAY	8.548	1.124	9.52
-29 DAYS	5.837	1.788	0.00
-364 DAYS	6.892	1.581	19.04
OVERALL	**6.601**	**1.485**	**8.33**

Figure 4: Completion times and error rates for verifying *today's* date in Experiment 2

Among the 21 subjects, the error rates were significantly lower than in our first series of experiments. While the overall error rate dropped from 12.50% to 4.76%, the most significant reduction in errors was from the case of displaying a date 364 days earlier (from 30% to 4.76%). No participant made an error when presented a date 3 days in advance, and other error rates for the cases of +1 day, -1 day, and -29 days were 4.76%, 9.52%, and 4.76%, respectively. Overall, the false positive rate (users accepting stale dates) was low at 6.34%. The results of the test were encouraging and lead us to hypothesize that the change in date format from $MM/DD/YYYY$ to $YYYY/MM/DD$ is helpful in reducing error rates significantly.

For the second part of the experiment which studied to effectiveness of asking users if the displayed date was *today's date*, the 84 (= 4*21) test cases yielded an average completion time of 6.601 seconds, with a standard deviation of 1.486 seconds, reaffirming our above conclusions regarding the ability of users to make quick decisions.

Among the 21 subjects, the overall error rate was 8.33% – still a significant reduction from the first series of experiments. There were four test cases, each of which yielded the following error rates: -1 year – 19.04%, -1 month – no errors, -1 day – 9.52%, and rejecting the correct date – 4.76%. The overall results are illustrated in Figure 4.

- **SUS Scores and User Opinion:** Subjects that tested our implementations rated their usability at 71.5% and 76% on the System Usability Scale (SUS) for the YYYY/MM/DD and *today's date* tests, respectively. These are comparable to the score of 76% obtained in the first series of experiments conducted earlier. The scores indicate good usability and acceptability characteristics.

About 66% of our subjects stated they would like the YYYY/MM/DD presentation to be implemented on their own personal tags compared to about 71% that were happy to see the second, *today's date*, implemented. The average score on a 5 point Likert scale was 3.81 and 3.95, for the two tests respectively.

- **Discussion:** Results indicate that the modifications made to our implementations of the reader revocation procedure do result in lower error rates and comparable usability scores. In general, the methods do not result in high number of false positives or negatives. With the change in the date format, stale years are drastically reduced from 30% to 4.76%. This leads us to believe that the conjecture made in the earlier experiment regarding the users lack of attention to the year when confronted with a date in its typical format, is likely to be true.

  The second test, which required users to verify *today's* date, resulted in slightly lower error rates (compared to Experiment #1) and higher usability characteristics. When questioned about their preferences, users in general indicated that they were more comfortable verifying if a date was *today*, rather than verifying if a date had already passed.

5.2 On-line Survey

We created an online survey [1] that was used to anonymously sample 98 individuals. The purpose was to collect information regarding perceived usability and general acceptance of our solution, rather than its actual usability. Participants were given an explanation of the reader revocation problem. Then, they were presented with the detailed description of our approach that included all user interaction. Next, participants rated the proposed technique via the 10-question System Usability Scale (SUS) [4]. They also answered 4 additional questions, as discussed later in this section.

Survey Results: The proposed technique yielded a score of 68/100 on the system usability scale (SUS). 66% of the participants stated that they would like to see it implemented on their E-passports, while 26% were neutral (the average score on 5-point Likert scale was 3.67 with the standard deviation of 0.87). 84% of the participants were worried about identity theft and 88% stated that they are concerned about revealing personal information to unauthorized parties in general.

In the online survey, we did not ask the subjects for their estimate of the current date or whether a displayed is stale, as this data would have been severely biased owing to the availability of the current date on their computer screen. Instead, participants were asked about their general awareness of the current date. 40% indicated that they are usually aware of the exact date, 35% were confident to know it with at most one-day error margin and 22% claimed to be within the +/- 3-day range. The remaining 3% indicated that 7 or more days error would be possible on their estimate of the current date.

5.3 Discussion

Based on our usability results, we now attempt to answer the questions raised at the beginning of this section:

Are people concerned with the problem we aim to solve? Among the 149 total participants (98+30+21, in both experiments and the survey) 84% are worried about revealing information to unauthorized parties. Slightly over 70% said that they wanted to see one of the proposed techniques implemented on their personal tags.

How do people rate the usability of our approach? Given the detailed description of the method and required interaction, 98 participants rated its usability at 68% on SUS scale. The usability rating was even higher (76%) for the 30 subjects who actually experimented with the mock-up implementation in Experiment #1 and was 71.5% and 76% for the two tests in Experiment #2, respectively. Both scores are above industry averages [21] and indicate good usability and acceptability characteristics.

Are users aware of current date? As results show, our methods very rarely yield false negatives: users are capable of not mistaking valid (future) dates for being in the past. As far as false positives, however, results are mixed. Stale days and months are, for the most part, easily recognized as such. However, as indicated by results from Experiment #1, it is clear that with the stale year, the error rate is quite high at 30% when the date is displayed in standard formats (i.e., MM/DD/YYYY or DD/MM/YYYY). However, our modification of switching the format to YYYY/MM/DD in Experiment #2 helped significantly to lower these errors. Another approach that yielded quite lower error rates is showing *today's date* to the users instead of an expiration date. Altogether, it is safe to say that users are reasonably aware of the current date but the formulation of the question and the format that the date is displayed may significantly affect error rates.

6 Application Feasibility

In this section, we present a sample case to show how our solution can be integrated into the current standards and procedures in place for typical pk-enabled personal RFID tags. In our sample case, we explain how our solution can be incorporated into e-Passport PKI-s.

6.1 The ePassport Public Key Infrastructure

The key elements in the Public Key Infrastructure (PKI) for e-Passports are the Country Verifying Certificate Authority (CVCA), the Document Verifiers (DV), and Inspection Systems (*a.k.a* readers). The primary role of the CVCA of a state is to issue certificates to Document Verifiers (national and international) and determine their access rights to e-Passports issued by the state.

The Document Verifier is a body that operates between readers and the CVCA. It is authorized by the CVCA to issue certificates to readers in its domain. The certificates issued by the Document Verifier to the readers contain information such as their access rights and validity period. The access rights and validity period of readers are restricted by the values issued to their Document Verifier by the CVCA. In order to access data on an individuals e-Passport, the reader must have the Document Verifier certificate issued to it by the individuals' home state. To achieve this, the Document Verifier distributes all Document Verifier certificates (it received from other CVCA's) to every reader it is responsible for. For easy distribution, the ICAO (International Civil Aviation Organization) provides a Public Key Directory (PKD) which contains the public keys of all participating Document Verifiers [13].

6.2 Modified Operation Procedure

In order to integrate our solution, currently deployed personal tags themselves and their operation procedures have to be slightly modified. The change needed on these tags is the integration of low-power displays attached displays (the feasibility of this requirement is analyzed in Appendix A).

The modified operation procedure that integrates our solution is summarized with respect to the known operation procedure of ePassports in Figure 5. Note that only the validation phase is modified to integrate our solution. This is an important observation, since it demonstrates that integrating our solution requires only an addition at the top of any protocol stack, removing the need for any changes to the control flow and the deployed PKI.

7 Conclusions

In this paper, we presented a simple and effective method for user-aided reader revocation on pk-enabled RFID tags. Our approach requires each tag to be equipped with a small display and be attended by a human user during certificate validation. We also conducted a series of comprehensive and in-depth user studies to understand the usability and security of our method in real life and how they can be improved with various changes in the protocol and the date representation.

Recent advances in display technology, such as ePaper and OLED, have already yielded inexpensive display-equipped RFID tags. The low cost of these displays combined with the better security properties and potential new application domains make displays on RFID tags a near reality. Moreover, our usability studies suggest that users find this solution usable and they are capable of performing their roles within reasonable error rates. We believe that display-equipped RFID tags will soon be in mass production and the method

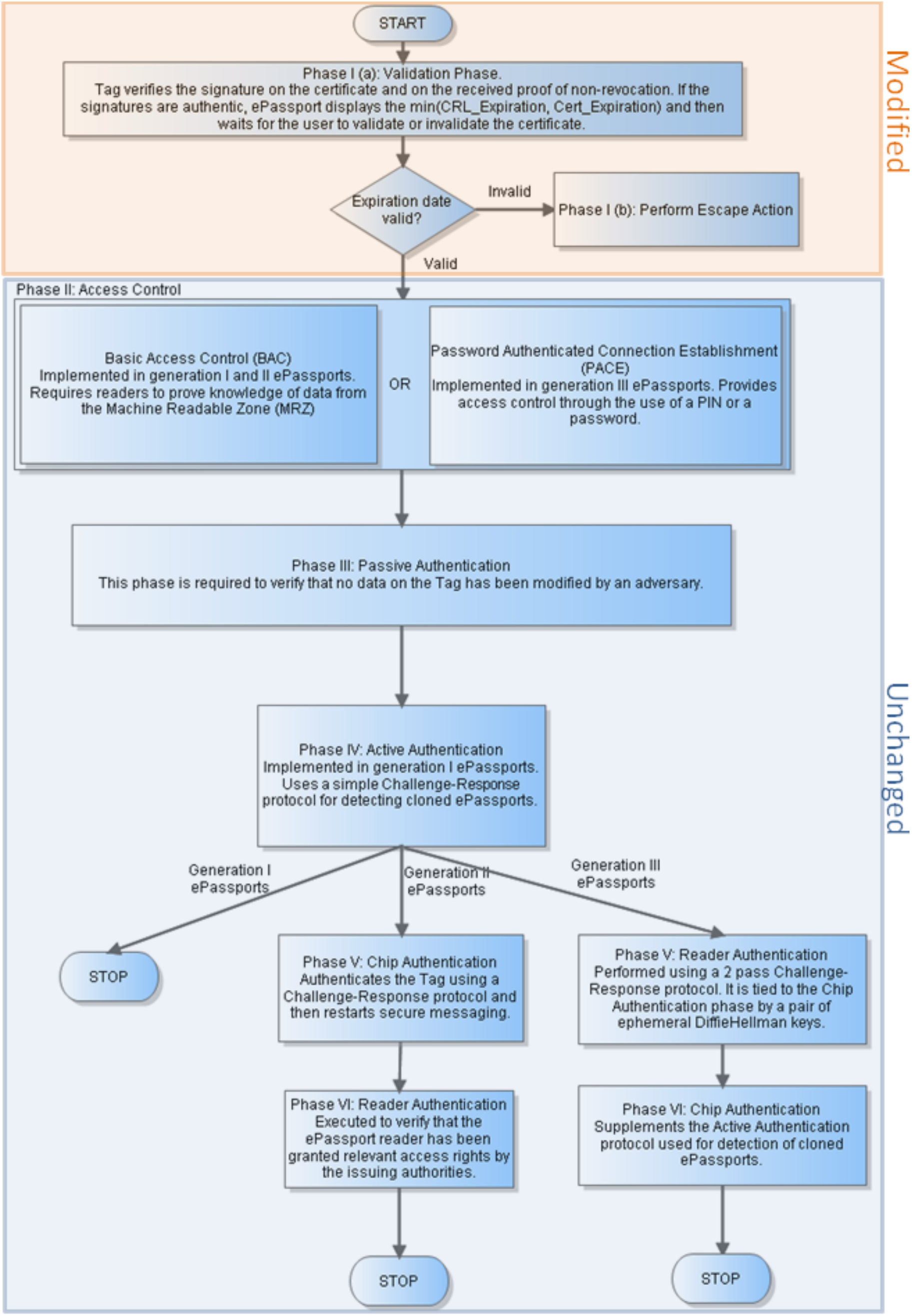

START
Phase I (a): Validation Phase.
Tag verifies the signature on the certificate and on the received proof of non-revocation. If the signatures are authentic, ePassport displays the min(CRL_Expiration, Cert_Expiration) and then waits for the user to validate or invalidate the certificate.
Expiration date valid?
Invalid
Phase I (b): Perform Escape Action
Valid
Modified
Phase II: Access Control
Basic Access Control (BAC)
Implemented in generation I and II ePassports. Requires readers to prove knowledge of data from the Machine Readable Zone (MRZ)
OR
Password Authenticated Connection Establishment (PACE)
Implemented in generation III ePassports. Provides access control through the use of a PIN or a password.
Phase III: Passive Authentication
This phase is required to verify that no data on the Tag has been modified by an adversary.
Phase IV: Active Authentication
Implemented in generation I ePassports. Uses a simple Challenge-Response protocol for detecting cloned ePassports.
Generation I ePassports
Generation II ePassports
Generation III ePassports
STOP
Phase V: Chip Authentication
Authenticates the Tag using a Challenge-Response protocol and then restarts secure messaging.
Phase V: Reader Authentication
Performed using a 2 pass Challenge-Response protocol. It is tied to the Chip Authentication phase by a pair of ephemeral DiffieHellman keys.
Phase VI: Reader Authentication
Executed to verify that the ePassport reader has been granted relevant access rights by the issuing authorities.
Phase VI: Chip Authentication
Supplements the Active Authentication protocol used for detection of cloned ePassports.
STOP
STOP
Unchanged

Figure 5: Modified e-Passport Operation procedure

proposed in this paper will be applicable to a wide variety of public key-enabled tags.

Acknowledgments: The authors are grateful to Bruno Crispo, Markus Ullman and the anonymous reviewers of ESORICS 2010 for their valuable comments on the preliminary version of this paper. This work is supported in part by NSF Cybertrust grant #0831526.

References

[1] Display enabled identification and payment instruments. `http://sprout.ics.uci.edu/projects/usec/survey.html`, November 2009.

[2] 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.

[3] C. Blundo, G. Persiano, A.-R. Sadeghi, and I. Visconti. Resettable and Non-Transferable Chip Authentication for ePassports. In *Conference on RFID Security*, 2008.

[4] 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.

[5] Bundesamt fur Sicherheit in der Informationstechnik. *Advanced Security Mechanisms for Machine Readable Travel Documents : Version 2.0*, 2008.

[6] J. H. Cheon, J. Hong, and G. Tsudik. Reducing RFID Reader Load with the Meet-in-the-Middle Strategy. Cryptology ePrint Archive, Report 2009/092, 2009.

[7] 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 *Computer and communications security – CCS*, 2008.

[8] M. Goodrich and R. Tamassia. Efficient authenticated dictionaries with skip lists and commutative hashing, May 7 2003. US Patent App. 10/416,015.

[9] T. Heydt-Benjamin, D. Bailey, K. Fu, A. Juels, and T. Ohare. Vulnerabilities in first-generation RFID-enabled credit cards. *Financial Cryptography and Data Security*, 2007.

[10] J.-H. Hoepman, E. Hubbers, B. Jacobs, M. Oostdijk, and R. Wichers Schreur. Crossing Borders: Security and Privacy Issues of the European e-Passport. In *International Workshop on Security – IWSEC*, 2006.

[11] R. Housley, W. Ford, W. Polk, and D. Solo. RFC 2459: Internet X.509 public key infrastructure certificate and CRL profile, Jan. 1999.

[12] Infineon Technologies AG, AIM CC. *Preliminary Short Product Information: Chip Card and Security IC's*, 2006.

[13] International Civil Aviation Organization. *Machine Readable Travel Documents: Specifications for Electronically Enabled Passports with Biometric Identification Capability*, 2006.

[14] A. Juels, D. Molnar, and D. Wagner. Security and privacy issues in e-passports. In *Security and Privacy for Emerging Areas in Communications Networks – SECURECOMM*, 2005.

[15] B. Kaliski. Future directions in user authentication. In *IT-DEFENSE*, 2005.

[16] G. Karjoth and P. A. Moskowitz. Disabling rfid tags with visible confirmation: clipped tags are silenced. In *Workshop on Privacy in the electronic society – WPES*, 2005.

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

[18] P. C. Kocher. On certificate revocation and validation. *Lecture Notes in Computer Science*, 1465, 1998.

[19] D. Kugler and M. Ullman. Contactless security tokens - enhanced security by using new hardware features in cryptographic based security mechanisms. In *Dagstuhl Seminar Proceedings of Foundations for Forgery - Resilient Cryptographic Hardware*, July 2009.

[20] A. Kumar, N. Saxena, G. Tsudik, and E. Uzun. Caveat eptor: A comparative study of secure device pairing methods. In *IEEE Pervasive Computing and Communications – PerCom*, 2009.

[21] J. Lewis and J. Sauro. The factor structure of the system usability scale. In *Proceedings of the Human Computer Interaction International Conference (HCII 2009), San Diego CA, USA*, 2009.

[22] R. C. Merkle. Secrecy, authentication, and public key systems. Technical report, Stanford University, 1979.

[23] S. Micali. Certificate revocation system. United States Patent, Sept. 1997. US Patent 5,666,416.

[24] J. Monnerat, S. Vaudenay, and M. Vuagnoux. About Machine-Readable Travel Documents. In *Conference on RFID Security*, 2007.

[25] 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.

[26] M. Naor and K. Nissim. Certificate revocation and certificate update. Technical report, 1999.

[27] M. Narasimha, J. Solis, and G. Tsudik. Privacy preserving revocation checking. *International Journal of Information Security*, 8(1):61 – 75, February 2009.

[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] Y. Oren and M. Feldhofer. A Low-Resource Public-Key Identification Scheme for RFID Tags and Sensor Nodes. In *ACM Conference on Wireless Network Security – WiSec*, 2009.

[30] N. Saxena, J. Ekberg, K. Kostiainen, and N. Asokan. Secure device pairing based on a visual channel. In *IEEE Symposium on Security and Privacy*, 2006.

[31] N. Saxena, M. B. Uddin, and J. Voris. Treat 'em like other devices: user authentication of multiple personal rfid tags. In *SOUPS*, 2009.

[32] P. Scholz, C. Reihold, W. John, and U. Hilleringmann. Analysis of energy transmission for inductive coupled rfid tags. *International Conference on RFID*, 2007.

[33] C. Soriente, G. Tsudik, and E. Uzun. BEDA: button-enabled device pairing. In *International Workshop on Security for Spontaneous Interaction, UbiComp 2007 Workshop Proceedings*, 2007.

[34] C. Soriente, G. Tsudik, and E. Uzun. Hapadep: Human asisted pure audio device pairing. *Information Security Conference – ISC*, 2008.

[35] G. Tsudik. Ya-trap: Yet another trivial rfid authentication protocol. *Pervasive Computing and Communications Workshop*, 2006.

[36] M. Ullman. Flexible visual display units as security enforcing component for contactless smart card systems. In *International EURASIP Workshop on RFID Technology*, 2007.

[37] M. Ullman. personal communication, Sept 2009.

[38] G. Whitrow. *Time in history: the evolution of our general awareness of time and temporal perspective.* Oxford University Press, 1988.

[39] T. Zeller. Black market in stolen credit card data thrives on internet. *The New York Times*, June 2005.

A Power Feasibility Analysis

The aim of this section is to show that it is completely feasible to integrate low power display technologies on passive RFID tags without any change on reader specifications. We analyze the maximum power requirements of the proposed system and its effect on the (theoretical) maximum working distance with current readers. In the rest of this section, we use ePassports as an example due to their clear tag and reader specifications.

We propose the use of display technologies such as ePaper, OLED, and other such low-power bistable displays. These displays require power of the order of 100mW (for a 2" display unit) during display updates and 0mW of power during standby.

A.1 Power Analysis

ePassport tags such as those supplied by Infineon Technologies, require up to 55mW of power to operate [12] while the display unit requires a maximum power of 100mW to operate. We analyze the power requirements of the proposed system from three aspects:

1. The ePassport tag is operating at maximum power and the display unit is static or non-existent.
2. The ePassport tag is on standby and the display unit is being updated (*i.e.*, refreshed).
3. The ePassport tag is operating at maximum power and the display unit is being updated (*i.e.*, refreshed).

In the first case, the power required by the ePassport circuit to operate will be ∼ 55mW (the power required by the display unit at this time is zero). In the second case, the power required by the ePassport circuit to operate will be ∼100mW (the power required by the tag during standby is negligible). In the final case, the power required by the ePassport circuit to operate will be ∼155mW (the sum of the maximum power required by the tag and display). The ePassport tag and reader when placed parallel to each other can be represented as a circuit, with circuit parameters set in the manner described by Scholz *et al.* [32].

First, we establish a relationship between the mutual inductance (M) and the distance (x) between the antenna of the tag and the reader.

$$M = \frac{\mu \pi N_1 N_2 (r_1 r_2)^2}{2\sqrt{(r_1^2 + x^2)^3}} \tag{1}$$

Where μ is the Permeability $[H/m]$; N_1 and N_2 are the number of turns in the antennas of the tag and reader; r_1 and r_2 are the radii $[mm]$ of each of these turns. Substituting default values [32] we get the relation

$$M = \frac{1.57 \times 10^{-12}}{x^3} \tag{2}$$

Now we establish a relationship between the power required by the tag (P_{Tag}) and distance (x). This is done through the series of equations below.

$$P_{Tag} = I_1^2 R_T \tag{3}$$

Where I_1 is the current running in the reader circuit $[mA]$ and R_T represents the tag impedance which is given by (4).

$$R_T = \frac{M^2 R_L}{L_2^2} \tag{4}$$

Where L_2 is assigned a value of 168nH [32] and R_L is the load resistance given by (5).

$$R_L = \frac{V_T^2}{P_{Tag}} \tag{5}$$

V_T is the voltage required in the tag circuit (5.5 Volts). The value of R_L is 195.1 Ω in the case that the ePassport tag and display unit operate at maximum power together (case 3). R_L is 302.5 Ω in the case that the ePassport tag is on standby when the display unit is refreshed (case 1). Finally, by combining equations 2 through 5, we can get a relationship between x and P_{Tag}.

$$x^6 = \frac{(1.57 \times 10^{-12})^2 \times (I_1)^2 \times (R_L)}{P_{Tag} \times (L_2)^2} \tag{6}$$

Making the necessary substitutions, we get the following values for x, where x represents the maximum possible operating distance:

- An ePassport tag without a display unit or with display on stand-by (*i.e.*, not refreshing):

$$P_{Tag} = 55\ mW,\ R_L = 550\ \Omega \Longrightarrow x = .097\ m \tag{7}$$

- An ePassport display unit while refreshing output when the tag is in standby mode:

$$P_{Tag} = 100\ mW,\ R_L = 302.5\ \Omega \Longrightarrow x = .080\ m \tag{8}$$

- An ePassport tag and the display unit operating at maximum power:

$$P_{Tag} = 155\ mW,\ R_L = 195.1\ \Omega \Longrightarrow x = .069\ m \tag{9}$$

From the above results it is clear that even with the current reader and antenna specification, adding a display reduces the maximum operating distance between the tag and reader only by 2.8 cm. Therefore, adding a display unit to the current ePassport circuit is feasible and doesn't require any changes over the power specifications in the original proposal [5]. If longer operating distances (over 6.9 cm) are needed, it can be achieved with small modifications on the RFID antenna design or by increasing power of a reader.