Privacy and Security in Library RFID Issues, Practices, and Architectures

David Molnar[*] David Wagner [†]

ABSTRACT

We expose privacy issues related to Radio Frequency Identification (RFID) in libraries, describe current deployments, and suggest novel architectures for library RFID. Libraries are a fast growing application of RFID; the technology promises to relieve repetitive strain injury, speed patron self-checkout, and make possible comprehensive inventory. Unlike supply-chain RFID, library RFID requires item-level tagging, thereby raising immediate patron privacy issues. Current conventional wisdom suggests that privacy risks are negligible unless an adversary has access to library databases. We show this is not the case. In addition, we identify *private authentication* as a key technical issue: how can a reader and tag that share a secret efficiently authenticate each other without revealing their identities to an adversary? Previous solutions to this problem require reader work linear in the number of tags. We give a general scheme for building private authentication with work logarithmic in the number of tags, given a scheme with linear work as a sub-protocol. This scheme may be of independent interest beyond RFID applications. We also give a simple scheme that provides security against a passive eavesdropper using XOR alone, without pseudo-random functions or other heavy crypto operations.

1. INTRODUCTION

Many libraries are starting to tag every item in their collections with radio frequency identification (RFID) tags, raising patron privacy concerns. An RFID tag is a small, low-cost device that can hold a limited amount of data and report that data when queried over radio by a reader. Several libraries, such as the Santa Clara City Library in California, the University of Nevada, Las Vegas library, and the Eugene, Oregon public library have already tagged every book, tape, CD, or other item in their collections. In an item-level tagging regime, the ability to track tags raises the possibility of surveillance of library patrons and their reading habits. We investigate privacy risks in libraries' use of RFID technology and methods for minimizing such risks.

The major driving force behind commercial deployment of RFID technology is presently logistics and supply chain applications. The U.S. Department of Defense uses RFID to manage shipments to armed forces worldwide. Meanwhile, several major retail chains, including WalMart, Target, and Albertsons, have mandated that all suppliers introduce RFID. Aside from supply chain applications, RFID technology is also found in proximity cards, car security devices, pet tracking, and other specialized applications.

Most supply chain applications focus on tagging cases or pallets holding merchandise. A key question has been the feasibility, security, and privacy of *item-level tagging*, in which each individual item is given its own RFID tag. Many have raised concerns over the privacy implications of item-level tagging. Still, item-level RFID tagging is often considered to be 5 or more years in the future for retail RFID applications, due to the cost of tags, reader infrastructure, and uncertainty about near term applications. In contrast, library RFID applications require item-level tagging, because RFIDs are used to manage each item in a library collection. Thus, library RFID applications may be the first major deployment of item-level tagging. This provides an interesting opportunity to study the privacy implications of item-level RFID tagging in a concrete, real-world setting.

Our contributions are twofold. First, we survey libraries' usage of RFID technology and analyze the privacy risks of current deployments (Sections 2 and 3). In the process, we have discovered several serious vulnerabilities that can compromise patrons' privacy. For example, the lack of appropriate access control allows tracking of people and books; the collision-avoidance protocol used in today's tags do not conceal tag identity; and poor key management practices threaten tag security. This analysis shows that today's practices and standards fail to protect patron privacy, and vulnerabil-

[*]dmolnar@eecs.berkeley.edu. Supported by Intel OCR Fellowship.

[†]daw@eecs.berkeley.edu. Supported by DARPA NEST contract F33615-01-C-1895.

CCS'04, October 25-29, 2004, Washington, DC, USA.

ities are present at all layers of the system. We further analyze these vulnerabilities in context of two real RFID deployments.

Second, we propose new architectures for using RFID technology securely in libraries without compromising privacy (Section 4). We identify *private authentication* as one of the key technical challenges in this area. We want tags to reveal their identity to authorized RFID readers (e.g., those owned by the library), so that the library can track books as they are checked in and out. However, for privacy, the tag must not disclose its identity until the reader has been authenticated; thus, the reader must authenticate itself to the tag before doing anything else. Also, prudent key management requires that each tag hold a different symmetric key. The paradox is that a legitimate reader cannot authenticate itself until it knows which key to use, which requires knowing the tag's identity, but for privacy reasons the tag dares not reveal its identity to an unknown reader before that reader has been authenticated. Nonetheless, despite the seeming impossibility of solving this problem, we show that it is possible to reconcile these two demands. In particular, we show efficient protocols for privacy-friendly symmetric-key authentication, which we expect will be well-suited to library RFID applications and of interest beyond RFID.

Finally, we wrap up with a discussion of related work (Section 5) and conclude (Section 6).

2. RFID BACKGROUND

A Radio Frequency Identification (RFID) tag is an electronic device that holds data. Typically these tags are attached to an item and contain a serial number or other data associated with that item. We will focus on *passive RFID* technology, in which the tag carries no power source, but is instead powered by a radio signal from a separate RFID reader. For a detailed introduction to RFID technology, see Finkenzeller [11].

RFID tags operate under severe restrictions compared to most personal computers, or even most embedded systems. First, an RFID tag is powered *only when within range of a reader.* This means that the tag has only an extremely limited amount of time to carry out computation. Pre-computation of results is impossible during times when the tag is out of range.

Second, an RFID has *extremely few gates*, and many of these are taken up by logic required for basic operation. Weis et al. estimate as few as 500-5000 gates total in a typical RFID design, leaving little for "extras" such as security [30]. In particular, symmetric encryption schemes such as AES, hash functions such as SHA1, or pseudo-random functions are not possible on today's RFID tags. While some low-end smart cards and tags have incorporated constructions based on stream cipher designs, no standardized low-gate primitive exists. Simple password comparisons and XOR operations are all that can be expected on most current-generation RFID tags. In addition, an RFID has almost no physical security.

Moore's Law tells us that the number of transistors per unit silicon doubles every 18 months [24]. These extra transistors might be used to enable cryptographic primitives on tags of equal cost as today's tags. It is more likely, however, that economic pressures will lead manufacturers to focus on ever-cheaper tags with a feature set similar to current-generation RFIDs. Because tags are manufactured on a massive scale,[1] even a half cent difference in unit cost makes an impact.

RFID tags used in libraries operate on the 13.56 MHz band and are manufactured by several companies, including Checkpoint Systems, Texas Instruments, and TAGSYS. Checkpoint and TAGSYS make proprietary tags, while the TI Tag-It! platform follows the ISO 15693 standard. ISO tags and TAGSYS tags are then resold by a variety of integrators, including 3M, TechLogic, and VTLS. Checkpoint tags, on the other hand, are installed only by the library services division of Checkpoint. In Figure 1 we give a table showing the most popular types of library RFID tags. We also give example libraries where these tags are deployed, and a partial list of library RFID vendors using each type of tag.

Recently, a new standard for RFID, ISO 18000, reached final stages of approval. ISO 18000-3 defines the physical interface and commands for 13.56 MHz tags. The 18000-3 standard is further divided into two "MODEs." MODE 1 is intended to be backwards compatible with the command set defined in ISO 15693, but standardizes various elements of the RF interface. MODE 2, on the other hand, is intended to be a next-generation RFID standard capable of supporting high-speed data transfer and communications with large numbers of tags at once. In addition, MODE 2 tags are explicitly required to support a random number generator and a small amount of semi-nonvolatile RAM. While MODE 2 tags are beginning to be manufactured, no library RFID vendor currently offers them for deployment.

The EPCGlobal consortium also publishes a series of specifications for RFID tags. These tags are aimed at supply chain markets and do not have a presence in the library setting. We note that most previous works on RFID privacy have focused on 915 MHz EPC Class 0 and Class 1 tags, and we will discuss these tags when appropriate for comparison. We will also consider the EPC Class 1 13.56 MHz tag specification. These tags also incorporate a special "kill" command that renders the tag permanently inoperative; while the kill command is protected by password, reads and writes are not.

The 13.56 MHz tags used by libraries have several material differences from the 915 MHz tags considered for supply chain applications. First, the bandwidth available to 13.56MHz tags is strictly limited by regulations in the US, the EU, and Japan. Second, the read range of 13.56 MHz tags is much less than that of 915MHz tags. As a result, RF air interface protocols, such as collision avoidance, differ between 915 MHz and 13.56 MHz tags. We will focus in more detail on collision-avoidance protocols in Section 3.2.

[1]At this writing, the RFID manufacturer Alien Technologies had announced plans to open a new plant in Fargo, ND capable of providing one billion tags per year, mostly aimed at the supply chain market.

Tag Type	Example Library	Example Vendors
Checkpoint WORM	Santa Clara City	Checkpoint
Checkpoint writeable	None	Checkpoint
TAGSYS C220-FOLIO	U. Delaware	VTLS, TechLogic
ISO 15693/18000-3 MODE 1	National U. Singapore	3M, Bibliotheca, Libramation
ISO 18000-3 MODE 2	Not yet available	Coming soon
EPC Class 1 13.56MHz	Not for library	WalMart
EPC Class 0 915MHz	Not for library	WalMart
EPC Class 1 915MHz	Not for library	WalMart

Figure 1: Summary of current RFID types.

RFID tags communicate with the reader by passively modulating a radio signal broadcasted by the reader. Because a reader is little more than a radio transceiver, this means that attackers will be able to obtain illegitimate readers that can be used to query RFID tags from some distance. Library RFID vendors claim that their readers can interact with tags from a distance of 2 feet (for large sensors at library exits), and hand-held readers might work up to 8 inches away from the tag [28, 3].[2] These distances are limited primarily by regulation on reader power and antenna size; thus, we should be prepared for illegal readers that might have a read range several times larger.

Even a few feet of read range is sufficient for scanning people passing through doorways and other close spaces. In fact, the sensors used to detect theft of library books look remarkably like, and have similar read range to, the RF-based anti-theft sensors already used in thousands of shops (see Figure 2). Later we will give more specific scenarios in which reading in these close spaces raises privacy risks. For a detailed discussion of the physics of RFID reading, see Reynolds [27].

Because the communication between reader and tag is wireless, there is a possibility for third parties to eavesdrop on these signals. One unusual aspect of RFID communication is an asymmetry in signal strength: because tags respond by passively modulating a carrier wave broadcast by the reader, it will be much easier for attackers to eavesdrop on signals from reader to tag than on data from tag to reader [30]. We make use of this property later, in our proposals for improved reader-tag authentication.

Because many RFIDs may be in range of a reader at the same time, collision-avoidance protocols must be used. The details of these protocols are often kept secret in proprietary tags. The ISO 18000 standard, however, specifies a collision-avoidance protocols for each of its two modes, as does the EPCGlobal suite of tag protocols [18, 6, 5, 7]. These protocols require a separate identifier, which we will call a *collision-avoidance ID* that may be independent of the data stored on the tag. In Section 3.2 we show that the collision-avoidance ID can often be used to track tags.

3. LIBRARY RFID ISSUES

[2]Compare to the 915 MHz tags used in supply chain and retail applications, which in contrast can be read from a distance of eight meters or more.

Figure 2: On the left, a Checkpoint library RFID tag. On the right, an exit gate.

3.1 Current Library RFID Architectures

Once a library selects an RFID system, it is unlikely that anything short of catastrophe could motivate a library to spend the money and labor required to physically upgrade the tags. Currently, tags cost in the neighborhood of US$0.75 (exact prices are confidential and may vary widely) [3], while readers and other equipment may cost multiple thousands of dollars.

Libraries make use of a *bibliographic database* to track circulation information about items in a collection. Each book, upon being acquired by the library, is assigned a unique number, usually called a *bar code*. There is no fixed relation between author, title, and bar code. In today's library RFID deployments, tags are programmed with at least the bar code. In addition, some vendors suggest placing extra information on the tag, such as shelf location, last checked out date, author, and title [22].

Check-out occurs at either a circulation desk or a special "self-check" machine that allows patrons to check out their own books. In both cases, the RFID tag is read and the association between ID number and book looked up in the bibliographic database, and the status of the book is changed to "checked out" in the bibliographic database. Later, when the book is checked in, the tag is read again and the bibliographic database updated.

The RFID tag also acts as a security device. Special RFID *exit sensors* are placed at the exit of a library, just as most libraries today have exit sensors for magnetic strip anti-theft devices. When a patron exits, the sensors scan for books that have not been checked out.

Depending on the vendor, the security check is achieved in at least one of two ways. One method, used by 3M, VTLS, and Libramation among others, stores the status of the book on the tag; a specific bit, often called a "security bit," reveals whether the book is checked in or checked out. It is important to note that the security bit does not necessarily affect whether the tag can be read. The security bit must be correctly set at every check-in and check-out, or else false alarms may be triggered. A second method, used by Checkpoint, does not store the circulation status on the tag. Instead, the readers query the bibliographic database for the circulation status of the book as it passes through the exit sensors; this introduces issues of latency due to query time.

Privacy concerns in today's deployments have focused on the bibliographic database and short range of RFID readers. Without the bibliographic database, an adversary cannot directly map a bar code number to the title and author of a book, and so cannot immediately learn the reading habits of people scanned. Some library RFID proponents have argued that an adversary without the database and with only short-range readers poses little to no risk. In the next section, we show this is not the case.

3.2 Attacks on Current Architectures

In what follows, unless otherwise specified, we assume the adversary does not have access to the bibliographic database. We do assume that the adversary has access to an RFID reader, however, and where indicated has the power to perform passive eavesdropping or even active attacks. Our attacks are summarized in Figure 3.

3.2.1 Static Tag Data and No Access Control

Referring to Figure 3, we see that none of today's library RFID tags employ read passwords or other read access control.[3] Because the identifier on the RFID tag never changes throughout its lifetime, the ability to read the tag at will creates several privacy risks.

First, the adversary may determine which library owns the book and infer the origin of the person carrying the book. In particular, bar codes for libraries with the Innovative bibliographic database have well-known, geographically unique prefixes. Vendors may also place library IDs on tags to prevent tags from one library from triggering readers at another. Learning origin data can be a privacy problem. For example, police at a roadblock may scan for patrons from specific city libraries in predominantly minority areas and search them more carefully; this would raise issues of racial profiling.

Second, any static identifier can be used both to *track* and *hotlist* books. In book tracking, the adversary tracks a book by correlating multiple observations of the book's RFID tag. The adversary may not necessarily know the title and author of the book unless the bibliographic database is available, but the static identifier can still be used to track the book's movements. Combined with video surveillance or other mechanisms, this may allow an adversary to link different people reading the same book. In this way, an adversary can begin profiling individuals' associations and make inferences about a particular individual's views, e.g. "this person checked out the same books as a known terrorist" or "mainly younger people have been seen with this book, so this person is young-thinking."

In hotlisting, the adversary has a "hotlist" of books in advance that it wishes to recognize. To determine the bar codes associated with these books, the adversary might visit the library to read tags present on these books. Later, when the adversary reads an RFID tag, it can determine whether that tag corresponds to a book on the hotlist. With current architectures, hotlisting is possible: each book has a single static identifier, and this identifier never changes over the book's lifetime.

Hotlisting is problematic because it allows an adversary to gather information about an individual's reading habits without a court order. For example, readers could be set up at security checkpoints in an airport, and individuals with hotlisted books set aside for special screening. For another example, readers could be set up at the entrance to stores and used to tailor patron experience or target marketing; these readers would look almost identical to the anti-theft gates used today.

Hotlisting is not a theoretical attack. We recall FBI warnings regarding almanacs as an indicator of terrorist activity [8]. We have also heard anecdotal reports from librarians that they refuse requests by law enforcement to track specific titles, and there are troubling historical precedents surrounding law enforcement and libraries. In the 1970s, the FBI Library Awareness Program routinely monitored the reading habits of "suspicious persons"; this was stopped only after public outcry and the passage of library privacy laws in many jurisdictions. Under the USA PATRIOT act, however, patron records may be accessed by order of the Foreign Intelligence Surveillance Court, or via a National Security Letter, as well as by a regular court order[9].

We have experimentally verified that tags can be read without access control at two library deployments of RFID. One library is the Cesar Chavez branch of the Oakland Public Library, which uses ISO 15693 tags; the other is the University of Nevada, Las Vegas library, which uses Texas Instruments Tag-It! tags. We used a TAGSYS Medio S002 short-range reader for our experiments. We saw both deployments use static identifiers that enable tracking and hot-listing.

3.2.2 Collision-Avoidance IDs

Even if RFID tags were upgraded to control access to bar codes using read passwords or some other form of access control, many tags can still be identified uniquely by their radio behavior. In particular, many tags use a globally unique and static *collision ID* as part of their collision-avoidance protocol. This typically will allow unauthorized readers to determine the tag's identity merely through its collision-avoidance behavior. We give some concrete examples of this issue.

- In ISO 18000-3 MODE 1 tags, the current draft of the standard specifies that each tag will have a globally unique, 64-bit "MFR Tag ID." Further, tags are mandated to support an "Inventory" command that returns the MFR Tag ID as part of the

[3] Proprietary tag formats may raise the cost of building unauthorized readers, but such minor barriers will inevitably be defeated. As always, security through obscurity is not a good defense.

response; no access control is in place for this command. Thus, an attacker with a reader could learn the tag's identity simply by asking for it.

  This ID is also used for the collision-avoidance protocol of MODE 1, which introduces a second way that the tag's identity can leak. The MODE 1 collision-avoidance protocol operates in two modes: slotted or non-slotted. In non-slotted mode, the reader broadcasts a message with a variable-length *mask*. All tags with least significant bits matching the mask respond, while others remain silent. To learn a tag's ID, an adversary need only make two mask queries per bit and see to which one the tag responds. By extending the mask by one bit each time, the adversary can learn a tag's collision ID in 64 queries. Because in the MODE 1 collision-avoidance protocol this ID is the same as the MFR Tag ID, this allows unique identification of the tag. In the slotted verion of the MODE 1 protocol, time is divided into 16 slots based on the most significant bits of the ID, and the process is similar.

  EPC Class 1 13.56 MHz tags use their EPC identifier directly in a similar collision-avoidance protocol [7].

- ISO 18000-3 MODE 2 also specifies a 64-bit manufacturer ID. The ID is not used directly for collision avoidance. The collision avoidance protocol requires the generation of random numbers, however, and the standard specifies the use of "at least a 32-bit feedback shift register or equivalent." While it is not explicitly specified, we expect that each tag will have a globally unique seed in practice. In particular, we note that 32 bits of the 64 bit manufacturer ID are defined to be a globally unique "specific identifier"; it would be natural to use this specific identifier to seed a PRNG.

  If a 32-bit LFSR is used, then tags can be uniquely identified. Specifically, if as few as 64 outputs of the LFSR are observed in the collision-avoidance protocol, the entire state of the LFSR can be reconstructed using the Berlekamp-Massey algorithm and run backwards to obtain the unique seed. In general, if a weak PRNG is used with the ISO 18000-3 MODE 2 protocol, tags can be identified.
- In EPC 915 MHz tags, there are three different modes for "singulation" or collision avoidance, one of which uses the globally unique Electronic Product Code (EPC) ID. The choice of modes is controlled by the reader. An adversarial reader can simply ask the tag to use its EPC ID; because there is no authentication of this command, the tag will obey.

As a consequence, any library system using one of these tags will be vulnerable to tracking and hotlisting of books and patrons. The collision-avoidance behavior is hard-coded at such a low layer of the tag that, no matter what higher layers do, privacy will be unachievable. This is unfortunate, because it means that much of today's RFID hardware is simply incompatible with privacy for library patrons. It is also dangerous, as vendors and libraries may implement privacy-enhancing methods that focus on tag data and then be unaware that tags are not in fact private.

3.2.3 Write Locks, Race Conditions, and Security Bit Denial of Service

In deployments with rewritable tags, some method must be used to prevent adversaries from writing to the tag. Otherwise, an adversary can commit acts of vandalism such as erasing tag data, switching two books' RFID data, or changing the security status of tags with "security bits." Unfortunately, vandalism is a real threat to libraries, especially from people who feel certain books should not be available; it would be naive to expect such people to ignore RFID-based vandalism for long.

Unfortunately, several current specifications have write protection architectures that are problematic in the library application. The EPC 13.56 MHz tag specification, as well as ISO 18000-3 MODE 1, include a "write" and a "lock" command, but no "unlock" command. In addition, write commands are not protected by password; this is consistent with a supply chain application that writes a unique serial number to a tag, then never needs to re-write the number. While the lock command is only an optional part of the ISO 18000-3 MODE 1 standard, it is supported by many tags, including the Phillips ICode tags purchased by the National University of Singapore to supplement its 3M library system [10]. In ISO 18000-3 MODE 2, locking is also irrevocable, but protected by a 48-bit password.

Once locked, a page of memory cannot be unlocked by any reader. A page containing a security bit needs to be unlocked when a book is checked in or out, or else the status of the bit can not be changed. An adversary can change the security bit to "not checked out" and then lock that page of memory. The resulting tag is then unusable, as the memory cannot be unlocked; physical replacement of the tag is required before the book can be checked out. We refer to irrevocable locking of the security bit as a *security bit denial of service*.

In addition to the issues with implementing security bits, there is a privacy concern as well. If there exists unlocked memory on the tag, an adversary can write its own globally unique identifier and track tags based on this ID; the RF-DUMP software by Grunwalds makes this a one-click operation [15]. This attack could bypass other mechanisms intended to prevent tracking or hotlisting of tags, such as rewriting tag IDs as we discuss in Section 4.1.1. Therefore, care should be taken to always lock all unused memory on writeable library RFID tags.

In our experiments with ISO 15693 tags in a real library deployment, we experimentally verified that none of the tag data blocks were locked. We also verified that tag blocks could be locked irrevocably on these tags, enabling security bit denial of service.

TAGSYS C220 tags avoid security bit denial of service by having a special area of memory dedicated to the security bit built into the tag, separate from regular data storage. Checkpoint tags, in contrast, do not implement security bits, but rely on a database of checked-out books.

An alternative RFID architecture might implement separate "unlock," "write," and "lock" commands, either on a per tag or per data page basis. Such an architecture is suggested by Weis et al. in the context of "hash locks" [30]. Weis et al. note that session hijacking is possible in such an architecture. In such a system, it is also possible for an active adversary to bypass the write lock mechanism by racing a legitimate reader. After waiting for the legitimate reader to unlock the tag, the adversary can then send write commands which will be accepted by the tag.

In practice, tags may be left unlocked by accident if a tag is prematurely removed from a reader's field of control before the tag can be re-locked. We have anecdotal evidence that this occurs in self-check stations when patrons place a large stack of books on the machine, but remove them before all can be locked. In this case, the tag is vulnerable to malicious writes of all unlocked data.

In addition, several tag types support command sequences that force a tag to restart collision avoidance protocols. If a unlock-write-lock architecture is overlaid on these tags, special care must be taken that tags transition to the "locked" state on receipt of any such commands.

3.2.4 Tag Password Management

The ISO 18000 standard and EPC specifications only allow for static passwords sent in the clear from reader to tag. As noted, current deployments do not seem to use read passwords, but write passwords are employed. There are two natural approaches to password management: (1) use a single password per site; or, (2) endow each tag with its own unique password.

If a single password is used for all tags, then a compromise of any tag compromises the entire system. In deployments that use writable security bits, the write password is used on every self-checkout; in systems with read passwords, exit sensors must use the read password every time a book leaves the library. In either case, passwords are available to a passive eavesdropper. Consequently, eavesdropping on a single communication reveals the password used by every tag in the system, a serious security failure. Once learned by a single adversary, a password can be posted on the Internet. Then, anyone with a reader can mount the attacks we have discussed.

If different passwords per tag are used, then some mechanism is required to allow the reader to determine which password should be used for which tag. Unfortunately, most obvious mechanisms for doing so, such as having a tag send an index into a table of shared secrets to the reader, provide tags with static, globally unique IDs. These globally unique IDs allow tracking and hotlisting of tags, which would defeat the entire purpose of read access control. Thus, privacy appears incompatible with prudent password management. We will return to this question in Section 4.2.

4. TOWARDS PRIVATE LIBRARY RFID ARCHITECTURES

Unfortunately, as we have shown, many types of current tags can be uniquely identified by their collision-avoidance behavior. This identification is independent of any read access control on the tag data. Consequently, it appears to be impossible to build privacy-preserving architectures for library RFID on today's tags.

4.1 Tags With Private Collision Avoidance

If we have a tag with private collision avoidance, then we have a hope for achieving a private library RFID architecture.

4.1.1 Random Transaction IDs on Rewritable Tags

Our first proposal is similar to the Anonymous ID scheme proposed by Ohkubo et al. [21]; we adapt it to the library setting. On each check-out, the reader picks a new random number r, reads the tag data D, and stores the pair (r, D) in a backend database. The RFID reader then erases D from the tag and writes r. On check-in, the library reader reads r, looks up the corresponding D, and writes D back to the tag. While tracking a book is still possible with this scheme, hotlisting is not. This scheme also offers a measure of forward privacy if the database securely deletes r after the book is checked in. Special care must be taken that the book's identifier has been written correctly, as RFIDs have difficulty writing at a distance. For example, the process may involve reading back and validating the new ID at check-in and check-out.

4.1.2 Improved Passwords Via Persistent State

One of the problems with simple passwords is that a passive eavesdropper can overhear the password. In the library RFID application, this is especially serious, as the exit sensors must read every book leaving the library. It has been observed by several authors that the channel from tag to reader is much harder to eavesdrop than the channel from reader to tag [30, 12]. With that in mind, we propose a simple protocol for enhancing passwords in RFID tags; the same protocol was independently discovered and proposed as part of the EPC-global Gen II standards process. The main idea is for the tag to send a random nonce to the reader; an adversary who misses the nonce cannot recover the password from reader to tag communication alone.

Let s be the shared secret password, and cmd the command to execute. Schematically, our protocol is:

$$\begin{array}{lcl}
\text{Reader} \quad s \in \{0,1\}^n & & \text{Tag} \\
 & \xrightarrow{\texttt{HELLO}} & \\
 & \xleftarrow{\quad r \quad} & r \in_R \{0,1\}^n \\
 & \xrightarrow{\text{cmd},\ \ p=r\oplus s} & \text{check that } p \oplus r = s
\end{array}$$

The tag then returns the result of the check to the reader by either responding to the command or raising an error. This protocol is only intended to provide security against passive eavesdropping on the reader-to-tag link; in particular, it does not provide security against man-in-the-middle attacks or attacks that modify transmitted messages. If the adversary does not see

Tag Type	Read PW	Write PW	DoS	Priv. C.A.	Priv. Auth.
Checkpoint WORM	No	n/a	n/a	Unknown	No
Checkpoint writeable	No	Yes	n/a	Unknown	No
TAGSYS C220 FOLIO	No	Yes (32 bits)	Unknown	Unknown	No
ISO 15693/18000-3 MODE 1	No	No (Lock)	Yes	No	No
ISO 18000-3 MODE 2	Yes (48 bits)	Yes (48 bits)	Yes*	No*	No

Figure 3: Summary of attacks. The fourth column indicates whether the tag type is vulnerable to security bit denial of service; the fifth and sixth columns show whether the tag supports private collision-avoidance and private authentication protocols. Note that all but the ISO 18000-3 MODE 2 tag lack access control and hence are vulnerable to straightforward hotlisting and tracking attacks. ISO 18000-3 MODE 2 tags leak their identity through the collision-avoidance protocol (unless a crypto-strength PRNG is used), and are vulnerable to security bit DoS attacks if the password is known.

the nonce value r, then, assuming the tag picked the nonce uniformly at random, the secret s is information-theoretically secure. Further, we note that an adversary cannot replay protocol messages, as the nonce required by the tag changes each time. Moreover, the adversary cannot even determine whether authentication succeeded from the protocol run. Finally, because the nonce r is independent of tag data or serial number, it cannot be used to distinguish different tags. The major drawback of the protocol is that it requires a good source of randomness, either physical or pseudorandom, for the RFID tag; finding such a source given the limited capabilities of a tag is an open problem.

4.2 Private Authentication

4.2.1 Motivation and Previous Work

As noted earlier (see § 3.2), good security practice dictates that each tag have a distinct secret key, raising the issue of how a reader knows which secret to use when presented with a new tag. Trying each secret in turn will take too much communication to be feasible. At the same time, most straightforward ways for accomplishing this goal provide unique identifiers for the tag, which defeats the purpose of read access control in the library RFID setting. This is the symmetric-key *private authentication* problem: how can two parties that share a secret authenticate each other without revealing their identities to an adversary?

We refer to a private RFID authentication scheme by a triple of probabilistic polynomial time algorithms (G, R, T) (for Generator, Reader, and Tag). Let k be a security parameter. The key generator $G(1^k)$ is a randomized algorithm that outputs a reader secret key RK and a tag secret key TK. Then the algorithms $R(RK)$ and $T(TK)$ interact to perform authentication. We will say a scheme is private if an adversary is unable to distinguish two different tags with different secret keys, and secure if an adversary cannot fool a tag or reader into accepting when it does not in fact know the secret key.

A key performance metric is how the amount of work performed by the reader scales with the number of tags in the system. This is especially important in the library setting, where there may be hundreds of thousands of items in a collection. There have been several proposals for private authentication of RFID tags, but all require work linear in the number of tags, which will not scale.

Weis et al. suggest a randomized hash lock protocol for private authentication [30]. At setup time, each tag is given a unique secret s and identification ID, and the reader has a database D storing the list of pairs (s, ID). In their protocol, the tag sends a message consisting of $(r, f_s(r) \oplus ID)$ to the reader, where s is a shared secret, f is a PRF, r is picked uniformly, and ID is the tag's unique identification. The reader then finds a pair $(s, ID) \in D$ that is consistent with the tag's message, and the reader authenticates itself by sending back ID.

This scheme is neither private nor secure against passive eavesdroppers. In addition, there is a further protocol attack: an adversary can query a tag and learn a valid pair $(r, f_s(r) \oplus ID)$, which then allows the attacker to later impersonate that tag to a legitimate reader. The legitimate reader's response will identify the tag. This is a serious security flaw; it would allow hotlisting, tracking, and other privacy abuses. In addition, the reader's computational workload is linear in the number of possible tags, when we use a separate key for each tag.

4.2.2 A Basic PRF Private Authentication Scheme

We propose a scheme for mutual authentication of tag and reader with privacy for the tag. Our scheme, shown in Figure 4, uses a shared secret s and a PRF to protect the messages communicated between tag and reader. The result is a private authentication scheme with reader workload linear in the number of tags. We refer to this basic PRF scheme as $(G_{\texttt{basic}}, R_{\texttt{basic}}, T_{\texttt{basic}})$.

4.2.3 Tree-Based Private Authentication

Next we discuss how to provide scalable private authentication. We build a new tree-based protocol with reader work $O(\log n)$, $O(\log n)$ rounds of interaction, and $O(\log n)$ tag storage, where n denotes the number of tags. Our scheme, $(G_{\texttt{tree}}, R_{\texttt{tree}}, T_{\texttt{tree}})$, assumes the existence of a subprotocol (G_1, R_1, T_1) that provides private authentication with constant rounds, constant tag storage, and reader work linear in the number of tags.

We consider the n tags as leaves in a balanced binary tree, then associate each edge in the tree with a secret. Each secret is generated uniformly and independently. The reader is assumed to know all secrets. Each tag stores the $\lceil \lg n \rceil$ secrets corresponding to the path from the root to the tag. The reader, when it wishes to authenticate itself to a tag, starts at the root and uses R_1 to check whether the tag uses the "left" secret or the "right" secret. If the reader and the tag successfully authenticate using one of these two secrets, the reader and tag continue to the next level of the tree. If the reader fails to convince the tag on any level, the tag rejects the reader. If the reader passes all secrets in the

Reader	$s \in \{0,1\}^n$	Tag
$r_1 \in_R \{0,1\}^n$	$\xrightarrow{\text{HELLO},\ r_1}$	
find $(s, ID) \in D$ s.t. $ID = \sigma \oplus f_s(0, r_1, r_2)$	$\xleftarrow{r_2,\ \sigma = ID \oplus f_s(0,r_1,r_2)}$	$r_2 \in_R \{0,1\}^n$
	$\xrightarrow{\tau = ID \oplus f_s(1,r_1,r_2)}$	check that $ID = \tau \oplus f_s(1, r_1, r_2)$

Figure 4: Our basic PRF-based private authentication protocol.

path, the tag accepts the reader.

This tree-based scheme requires $\lceil \lg n \rceil$ invocations of R_1 and T_1 with 2 secrets. Therefore the total scheme requires $O(\log n)$ rounds of communication, $O(\log n)$ work for the reader, and $O(\log n)$ storage at the tag. The tree-based scheme is shown in Figure 5.

For simplicity of exposition, we described the scheme in terms of a binary tree, but nothing restricts the tree-based scheme to binary trees. Larger branching factors reduce the number of rounds of interaction and improve resistance against compromise tags at the cost of somewhat increased reader work.

One way to instantiate the tree-based scheme is by using our basic PRF scheme as the subprotocol. We stress PRFs are not required; our scheme can be used with any subprotocol for private authentication. For example, we could use the XOR-based scheme, at the cost of only achieving security against passive adversaries.

The main issue with our scheme is the number of rounds of communication. Ramzan and Gentry have pointed out that some underlying protocols may allow performing all levels of the tree in parallel [14]. For instance, this optimization can be applied when using our basic PRF scheme, yielding a protocol with $O(1)$ rounds of interaction and $O(\log n)$ messages.

4.2.4 A Two-Phase Tree Scheme

As just described, the tree scheme uses a single fixed security parameter k for all instances of R_1 and T_1, which therefore require communication cost at least k for each of the $\lceil \log n \rceil$ rounds, or $O(k \log n)$ communication. We now describe how we can create a tree scheme with communication $O(k + \log n)$ by splitting into two phases.

In the first phase, we run the tree scheme using R_1 and T_1 generated with a constant security parameter (that may depend on the level of the tree) to identify the tag. If the path from root to tag is long compared to the security parameters of the edges, the probability that either an adversary identifies the tag or that a legitimate reader mis-identifies the tag will be low; we can tailor this probability by trading off the branching factor and the phase-1 security parameter. In the second phase, once the tag is identified, the reader and tag can execute R_1 and T_1 using k as the security parameter.

For a concrete example, consider the basic PRF scheme, $n = 2^{20}$ tags, and a two-level tree with branching factor $2^{10} = 1024$. We give a tag three 64-bit secret keys: two for phase 1 and the final key for phase 2. In both levels, we truncate the PRF output to 10 bits. We then expect to need only one iteration of the first and one of the second level, for a total expected $2 \cdot 2^{10} = 2^{11}$ PRF evaluations for the reader and 4 PRF evaluations for the

Algorithm 4.1: $G_{\text{TREE}}(1^k, N)$

Fix $\ell \leftarrow \log N$
for $i = 1$ **to** ℓ
  for $j = 0$ **to** 1
    $s_{i,j} \leftarrow G_1(1^k)$
for $h = 1$ **to** N
  Parse h in binary as $(b_1, \ldots, b_\ell)$
  $TK_h \leftarrow (s_{1,b_1}, \ldots s_{\ell,b_\ell})$
$RK \leftarrow (s_{1,0}, s_{1,1}, \ldots, s_{\ell,1})$
Output RK**,** $TK_1, \ldots, TK_N$**.**

Algorithm 4.2: $(R_{\text{TREE}}, T_{\text{TREE}})$ (RK, TK)

Fix $\ell \leftarrow \log N$
Parse RK as $(u_{1,0}, u_{1,1}, \ldots, u_{\ell,1})$
Parse TK as $(v_1, \ldots, v_\ell)$
for $i = 1$ **to** ℓ
  SUCCEED $\leftarrow$ **false**
  for $j = 0$ **to** 1
    if running $(R_1(u_{i,j}), T_1(v_i))$ returns true
      then SUCCEED $\leftarrow$ **true**
  if $\neg$SUCCEED
    then fail and output 0
accept and output 1

Figure 5: Unoptimized tree-based private authentication protocol.

tag in phase 1, plus 2 each for phase 2. Communication cost is then $10 + 10 + 64 = 84$ bits of PRF output, plus the same amount for the random nonces, for a total of 168 bits of communication. To fool a tag into accepting, the adversary must pass both phase 1 and phase 2. Ramzan notes that any authentication scheme with n possible tags requires $\Omega(\log n)$ communication cost, because writing a tag identifier requires $\Omega(\log n)$ bits, so we see our two-phase tree scheme is asymptotically optimal [26].

5. RELATED WORK

In the retail RFID space, the EPCGlobal suite of RFID specifications mandates that tags support an irrevocable "kill" command. In the library setting, however, tags must be re-used to check in loaned items. Irrevocably killing a tag is not an option.

Juels, Rivest, and Szydlo propose a device called a "blocker tag" [20]. The blocker tag exploits the tree-walking collision-avoidance protocol of 915 MHz EPC tags to "block" readers attempting to read tags of a consumer. Because of bandwidth constraints, the 13.56 MHz tags used in library settings do not use tree-walking, so their scheme is not applicable.

Weis et al. focus on a broad range of security and privacy issues in the RFID space [30]. Their protocol focuses on security against passive eavesdroppers who are assumed to hear the reader to tag channel but not tag to reader communication. Their proposal, however, is modelled on the 915 MHz EPC tree-walking protocol; a new protocol must be designed for 13.56 MHz tags. Weis et al. also introduce randomized hash locks. Unfortunately, as discussed in Section 4.2, the scheme requires reader computation linear in the number of secrets.

Abadi and Fournet describe the problem of private authentication [2]. We differ in that we work in the symmetric-key model, since public-key cryptography is out of reach for RFID tags. In addition, their protocols also have linear work in the number of entities, while we achieve logarithmic work. We note that the anonymous mode of IKE also achieves private authentication with public-key cryptography[16].

Ohkubo, Suzuki, and Kinoshita proposed a method of changing RFID identities on each read based on hash chains [25]. Their method also requires a hash function on the RFID tag, but does not require a random number generator. Ohkubo et al. suggest an "anonymous ID" scheme, in which tags contain only a random number that is periodically rewritten [21]. Their scheme appears similar to the scheme suggested in Section 4.1.1.

Juels suggests the use of one-time authenticators or "pseudonyms" for RFID tags [19]. He also specifically suggests a variant scheme for library applications that gives tags a single authenticator for each checkout and prevents hotlisting but not tracking; in this respect, the proposal is similar to the "anonymous ID" scheme.

Inoue and Yasuura suggested having two data banks on an RFID [17]. The authors recognize that switching between the two data banks must be secured, but leave the exact security mechanism as future work; therefore the scheme cannot be used as is.

Several activist groups have raised the issue of patron privacy for library RFID. The Electronic Frontier Foundation wrote a letter to the San Francisco Public Library raising several important policy questions surrounding library RFID [29]. A general "RFID Bill of Rights" was proposed by Garfinkel [13]; it proposes a right to notice that RFIDs are in use and a right to RFID alternatives.

Some vendors also have literature addressing the issue of library RFID and patron privacy. The 3M "eTattler" newsletter claims that the proprietary nature of 3M RFID tags and the low read range make privacy less of a concern [1]. The VTLS white paper on patron privacy cites low read range and also mentions that "encryption" can be used to protect tag data [4]. While library RFID read ranges may be low, they are still enough to provide for reading in doorways or other close spaces from vendor standard readers; adversaries willing to break the law and build more powerful readers may achieve greater range. Past experience also teaches us that it is dangerous to rely solely on security through obscurity and proprietary protocols.

Finally, the Berkeley Public Library has put together a series of "best practices" for library RFID [23]. These practices include limiting the data on the tag to a bar code only and prohibiting patrons from searching the bibliographic database by bar code. We have shown that privacy risks still exist even when data is limited to a bar code and the adversary does not have access to the bibliographic database, although in light of our results, the Berkeley practices seem to be the best possible with today's tags.

6. CONCLUSIONS

Current library RFID tags do not prevent unauthorized reading of tag data. Therefore, information such as title, author, shelf location, patron information, or last checkin/checkout time should in no circumstance be stored on library RFID tags.

At the same time, both *tracking* and *hotlisting* are possible whenever a static identifier is used. Therefore, if a static identifier is in place on the RFID tag, it is imperative to prevent unauthorized tag reads. We stress that static identifiers may include collision IDs that are not protected by access control mechanisms intended to protect tag data. To avoid tracking tags by collision ID, some mechanism for private collision avoidance must be used, as described in Section 3.2.

Would these library RFID security and privacy problems go away if tags advanced to the point where hash functions and symmetric encryption on tags became feasible? Our results on identification via collision avoidance, private authentication, and write locks show the answer is no. Careful design of the entire system is required to support privacy-enabled RFID applications.

What is more, libraries want RFID now. Over 130 libraries in North America alone have installed RFID technology, and more are considering it. The American Library Association will soon propose best practices for the library use of RFID; once these are finished, we can expect the adoption rate among libraries to rise. Waiting for next generation tags that support cryptography may not be acceptable, especially at increased cost. Tag vendors, in addition, may be unwilling to introduce special modifications for what is a comparatively small market.

We have given specific proposals for improving privacy in RFID tags. Unfortunately, such changes will require time, effort, and money, and no current library RFID system supports them. There will be a substantial cost for privacy and security in the library RFID setting.

Is the cost of privacy and security "worth it?" Put another way, should a library refuse to buy RFID until systems are available that resist these attacks? We cannot dictate answers to this question. What we have done, instead, is provide the means for libraries and their communities to make an informed decision, and the technical options to improve future library RFID systems.

7. ACKNOWLEDGEMENTS

We thank the following individuals for their feedback and advice: Alicia Abramson, Rebekah E. Anderson,

Oleg Boyarsky, Justin Chen, Karen Duffy, Elena Engel, Gerard Garzon, Craig Gentry, Nathaniel Good, Jackie Griffin, Steve Halliday, John Han, Craig K. Harmon, Susan Hildreth, Kris Hildrum, Eric Ipsen, Jayanth Kumar Kannan, Doug Karp, Elizabeth Miles, Dan Moniz, Deirdre Mulligan, Julie Odofin, Matt Piotrowski, Laura Quilter, Zulfikar Ramzan, Pam Samuelson, Karen Saunders, Rupert Scammell, David Schultz, Paul Simon, Al Skinner, Laura Smart, Ross Stapleton-Gray, Lee Tien, Jennifer Urban, Peter Warfield, and Hoeteck Wee.

8. REFERENCES

[1] 3M. eTattler newsletter, January 2004. http://cms.3m.com/cms/US/en/0-257/kkruuFX/viewimage.jhtml.

[2] Martin Abadi and Cedric Fournet. Hiding names: Private authentication in the applied pi calculus. In *Software Security – Theories and Systems. Mext-NSF-JSPS International Symposium (ISSS'02)*, pages 317–338. Springer-Verlag, 2003.

[3] Richard Boss. Library RFID technology. *Library Technology Reports*, Nov/Dec 2003.

[4] Vinod Chachra and Daniel McPherson. Personal privacy and use of RFID technology in libraries, October 2003. http://www.vtls.com/documents/privacy.pdf.

[5] EPCGlobal Consortium. EPC 868-915Mhz tag Class 1 candidate recommendation, 2004. http://www.epcglobalinc.org/standards_technology/Secure/v1.0/UHF-class1.pdf.

[6] EPCGlobal Consortium. EPC 900Mhz tag Class 0 standard, 2004. http://www.epcglobalinc.org/standards_technology/Secure/v1.0/UHF-class0.pdf.

[7] EPCGlobal Consortium. EPC ISM Band 13.56MHz Class 1 candidate recommendation, 2004. http://www.epcglobalinc.org/standards_technology/Secure/v1.0/HF-Class1.pdf.

[8] FBI Counterterrorism Division. FBI intelligence memo no. 102, December 2002. http://cryptome.quintessenz.org/mirror/fbi-almanacs.htm.

[9] Charles Doyle. Libraries and the USA PATRIOT act, 2003.

[10] Phillips Electronics. ICode SLI data sheet, 2004. http://www.semiconductors.philips.com/acrobat/other/identification/sl2ics20-fact-sheet.pdf.

[11] Klaus Finkenzeller. *RFID Handbook*. John Wiley and Sons, 2003.

[12] Kenneth Fishkin and Sumit Roy. Enhancing RFID privacy through antenna energy analysis. In *MIT RFID Privacy Workshop*, 2003. http://www.rfidprivacy.org/papers/fishkin.pdf.

[13] Simson Garfinkel. Adapting fair information practices to low cost RFID systems. In *Privacy in Ubiquitous Computing Workshop*, 2002. http://www.simson.net/clips/academic/2000_Ubicomp_RFID.pdf.

[14] Craig Gentry and Zulfikar Ramzan. Personal communication, 2004.

[15] Lukas Grunwalds. Rf-dump, 2004.

[16] D. Harkins and D. Carrel. Internet key exchange rfc 2409, 1998. http://www.faqs.org/rfcs/rfc2409.html.

[17] Sozo Inoue and Hiroto Yasuura. RFID privacy using user-controllable uniqueness. In *RFID Privacy Workshop, MIT*, 2003. http://www.rfidprivacy.org/papers/sozo_inoue.pdf.

[18] ISO/IEC JTC 1/SC 31/WG 4. Information technology AIDC techniques - RFID for item management - Air interface, - Part 3: - Parameters for air interface communications at 13.56 MHz, April 2004. Version N681R.

[19] Ari Juels. Minimalist cryptography for RFID tags, 2003. http://www.rsasecurity.com/rsalabs/staff/bios/ajuels/publications/minimalist/index.html.

[20] Ari Juels, Ronald L. Rivest, and Michael Szydlo. The blocker tag: selective blocking of RFID tags for consumer privacy. In *Proceedings of the 10th ACM conference on Computer and communication security*, pages 103–111. ACM Press, 2003.

[21] Shingo Kinosita, Fumitaka Hoshino, Tomoyuki Komuro, Akiko Fujimura, and Miyako Ohkubo. Non-identifiable anonymous-ID scheme for RFID privacy protection, 2003. In Japanese. See English description as part of http://www.autoidlabs.com/whitepapers/KEI-AUTOID-WH004.pdf.

[22] Libramation. Overview of library RFID products, 2004. http://www.libramation.com/prod_radio.html.

[23] Berkeley Public Library. Best practices for library RFID, 2004. http://berkeleypubliclibrary.org/BESTPRAC.pdf.

[24] Gordon E. Moore. Cramming more components onto integrated circuits. *Electronics*, 38(8), April 1965.

[25] Miyako Ohkubo, Koutarou Suzuki, and Shingo Kinoshita. Cryptographic approach to a privacy friendly tag. In *RFID Privacy Workshop, MIT*, 2003.

[26] Zulfikar Ramzan. Personal communication, 2004.

[27] Mark Reynolds. Physics of RFID. In *MIT RFID Privacy Workshop*, 2003. http://www.rfidprivacy.org/papers/physicsofrfid.ppt.

[28] Checkpoint Systems. ILS exit sensor data sheet, 2004. http://www.checkpointsystems.com/docs/ILSSENSOR.pdf.

[29] Lee Tien. Privacy risks of radio frequency identification "tagging" of library books, October 2003. http://www.eff.org/Privacy/Surveillance/RFID/20031002_sfpl_comments.php.

[30] Stephen A. Weis, Sanjay E. Sarma, Ronald L. Rivest, and Daniel W. Engels. Security and Privacy Aspects of Low-Cost Radio Frequency Identification Systems. In *Security in Pervasive Computing*, volume 2802 of *Lecture Notes in Computer Science*, pages 201–212, 2004.