The Blocker Tag:
Selective Blocking of RFID Tags for Consumer Privacy

Ari Juels[1] and Ronald L. Rivest[2] and Michael Szydlo[1]

[1] RSA Laboratories
Bedford, MA 01730, USA
e-mail: {ajuels,mszydlo}@rsasecurity.com
[2] Laboratory for Computer Science, MIT
Cambridge, MA 02139, USA
e-mail: rivest@theory.lcs.mit.edu

Abstract. We propose the use of "selective blocking" by "blocker tags" as a way of protecting consumers from unwanted scanning of RFID tags attached to items they may be carrying or wearing.

While an ordinary RFID tag is a simple, cheap (e.g. five-cent) passive device intended as an "electronic bar-code" for use in supply-chain management, a blocker tag is a cheap passive RFID device that can simulate many ordinary RFID tags simultaneously. When carried by a consumer, a blocker tag thus "blocks" RFID readers. It can do so universally by simulating all possible RFID tags. Or a blocker tag can block *selectively* by simulating only selected subsets of ID codes, such as those by a particular manufacturer, or those in a designated "privacy zone."

We believe that this approach, when used with appropriate care, provides a very attractive alternative for addressing privacy concerns raised by the potential (and likely) widespread use of RFID tags in consumer products.

We also discuss possible abuses arising from blocker tags, and means for detecting and dealing with them.

Key words: barcodes, privacy, RFID tags, tree walking

1 Introduction

An RFID (Radio-Frequency IDentification) tag consists of a small integrated circuit attached to a small antennae, capable of transmitting a unique serial number a distance of several meters to a reading device in response to a query. Most RFID tags are passive: they are batteryless and obtain the power necessary to operate from the query signal itself.

RFID tags are already quite common; examples include proximity cards used as replacements for metal door keys, theft-detection tags attached to consumer goods such as clothing, and the small dashboard devices for automating toll payments.

The cost of simple RFID tags is likely to fall to roughly $0.05/unit in the next several years [19], while tags as small as 0.4mm × 0.4mm, and thin enough to be embedded in paper are already commercially available [21]. Such improvements in cost and size will ensure a rapid proliferation of RFID tags into many new areas of use.

Indeed, the European Central Bank (ECB) has already indicated its plans to embed RFID tags in high-value Euro notes by 2005 [5, 24], while Gillette has recently ordered half a billion tags for use in retail environments [6].

The MIT AutoID Center (see www.autoidcenter.org) leads an industry consortium whose goals include designing exceptionally cheap RFID tags for use in supply-chain management, designing standardized protocols for querying and managing RFID tags, and exploring methods for dealing with the privacy concerns raised by the prospective pervasive use of RFID tags. (While the research we present here was not sponsored by the AutoID consortium, we believe it is very much consonant with the goals of the consortium.)

The AutoID Center envisions that RFID tags will play a major role as a means for implementing "electronic product codes" (EPC) in a standardized manner, for supply-chain and inventory management [3].

RFID tags will replace and improve upon the traditional ubiquitous printed barcode in consumer products. This change promises more flexible and intelligent handling of consumer goods and devices.

In addition to the mundane (but very important) implications for supply-chain management and automated checkout, RFID tags offer interesting new possibilities, such as microwave ovens that can read the tags on packages and cook food without explicit instructions, refrigerators that can recognize expired foodstuffs, and closets that can tally their contents.

1.1 The Threat to Privacy

The impending ubiquity of RFID tags, however, also poses a potentially widespread threat to consumer privacy [11].

The simplest RFID tag will broadcast its ID serial number – that is, its electronic product code (EPC) – to any nearby reader.

The ID number, as envisioned by the AutoID Center, is unique to a given tag. It contains not only the traditional information contained in a printed barcode (indicating manufacturer and product type), but also a unique serial number for that tag. Each consumer product or item of clothing will be uniquely identified. The ID number will be 64–128 bits in length.

This presents a clear potential for privacy violations. What woman wants her dress size to be publicly readable by any nearby scanner? Who wants the medications and other contents of a purse to be scannable? Who wants the amount of money in a wallet to be easily determinable by a scanner? Who wants his or her location to be tracked and recorded based on the unique ID number in shoes or other clothing?

The privacy issues raised by RFID tags have seen recent attention in the popular press, whose negative coverage forced the clothing retailer Benetton to withdraw plans for embedding RFID tags in its items of apparel [2, 20].

Researchers have recognized the RFID privacy problem for some time [9, 16], and are continuing to devise better approaches. No single approach is likely to be completely satisfactory, however; a combination of methods may prove to be best. This paper introduces a new, powerful tool into the arsenal of privacy-protecting technologies related to RFID tags.

In the next few subsections we first discuss some previously suggested approaches for protecting consumer privacy threatened by RFID tags. We then move on to our "blocker-tag" approach.

1.2 The “Kill Tag” approach

The most straightforward approach for the protection of consumer privacy is to “kill” RFID tags before they are placed in the hands of consumers. A killed tag is truly dead, and can never be re-activated.

The standard mode of operation proposed by the AutoID Center is indeed for tags to be killed upon purchase of the tagged product. With their proposed tag design, a tag can be killed by sending it a special “kill” command (including a short 8-bit “password”) [16, 17].

For example, a supermarket might use RFID tags to facilitate inventory management and monitoring of shelf stocks. To protect consumer privacy, checkout clerks would “kill” the tags of purchased goods; no purchased goods would contain active RFID tags.

Why the “Kill” approach is inadequate. There are many environments, however, in which simple measures like “kill” commands are unworkable or undesirable for privacy enforcement.

For example, consumers may wish RFID tags to remain operative while in their possession. The examples of home use set forth in [16, 18] – e.g., microwave oven that reads cooking instructions from food packages – rely on actively operational tags.

Similarly, new and clever consumer-specific applications for RFID-tags are already beginning to emerge. For example, a Prada store in New York City tracks the RFID tags of items held by customers in order to display related accessories on nearby screens [14].

Other examples of RFID-tag applications for ordinary consumers include effortless physical access control[1], theft-protection of belongings, and wireless cash cards.

A low-cost powerful technology like RFID tags will inevitably be used in numerous applications, many of which we don’t even imagine today. Many of these applications will require that tags still be active while in the consumer’s possession, and thus cannot be killed upon purchase. Here are a few suggestive such applications:

- Stores may wish products to have tags scannable if the products are returned as defective.
- Products may need to be scanned so they may be categorized for recycling purposes.
- Stores may issue receipts with embedded RFID tags, so they can confirm purchase details when a product is returned.
- Individuals may wish to have RFID tags embedded in their business cards, to facilitate scanning by recipients. Here the tag ID may be used to create a URL referring to the actual card data.
- A store may wish to embed RFID tags in store-issued coupons, for ease of scanning at the checkout counter.
- A user may wish to scan his possessions when a recall for a specific set of products is issued.
- Collectibles such as baseball cards or CDs may have RFID tags, to enable owners to manage their inventory better.

[1] Smartcards with RFID-enabled chips are in fact in use for this purpose today, but generally only function in very close proximity to readers.

- A merchant may wish to scan consumers for marketing purposes. (For example, what stores has this shopper previously visited today? What items has he bought previously from a competitor?)
- A refrigerator or pantry shelf may be able to tell when some food or drug product has passed its expiration ("use by") date.
- The US Postal Service may include RFID tags in postage.
- An airline ticket may contain an embedded RFID tag to allow simpler tracking of passengers within an airport.
- Businesses may include RFID tags on the invoices, coupons, and return envelopes they mail to consumers, for ease of sorting upon return.

Such "function creep" promises to result in many more uses unimagined or unimaginable today in which active tags will be valuable to consumers or businesses.

As an additional example, it may be possible for consumers to buy little "stickies" with embedded RFID tags, to attach to objects of their choice for idiosyncratic purposes. (Maybe for a "treasure hunt", for assisting a blind person, or for baggage labeling.)

Individuals may also be secretly given tags so they can be tracked or identified by an overzealous merchant, by a private detective, by a spouse, parent, or other relative.

Thus, while the "kill-tag on purchase" approach may handle many or even most instances of potential concern for privacy, it is unlikely to be a fully satisfactory solution.

It thus seems imperative to explore alternative approaches.

1.3 The Faraday Cage approach

An RFID tag may be shielded from scrutiny using what is known as a Faraday Cage—a container made of metal mesh or foil that is impenetrable by radio signals (of certain frequencies). Indeed, petty thieves are already known to use foil-lined bags in retail shops to circumvent shoplifting-detection mechanisms.

If high-value currency notes do indeed come supplied with active RFID tags, then it is likely that foil-lined wallets will become big sellers! At least one company already offers a Faraday-cage-based product for privacy purposes [12].

RFID tags will inevitably see use, however, in a vast range of objects that cannot be placed conveniently in containers, such as clothing [2, 20], wrist-watches, and even human beings [7, 11].

Faraday cages thus represent at best a very partial solution to consumer privacy.

1.4 The Active Jamming Approach

Active jamming of RF signals is another, related physical means of shielding tags from view. The consumer could carry a device that actively broadcasts radio signals so as to block and/or disrupt the operation of any nearby RFID readers.

This approach may be illegal – at least if the broadcast power is too high – and is a crude, sledgehammer approach. It could cause severe disruption of all nearby RFID systems, even those in legitimate applications where privacy is not a concern.

The approach we propose in this paper is akin to "jamming," but is much more subtle in its operation, interacting cleverly with the RFID "singulation" protocol to disrupt only certain operations.

1.5 The “Smart” RFID Tag Approach

Another general approach is to make the RFID tags a bit “smarter,” so that they interact in a way that protects privacy better, while providing the desired active functionality. This would typically involve the use of cryptographic methods.

These approaches are exceptionally challenging to design, given the severe cost constraints on the basic RFID tag. (With a budget of five cents, there is very little to spend on additional logic gates!)

Three instances of the “smart RFID-tag” approach that have been proposed are the hash-lock method, the re-encryption method (in several forms), and silent tree-walking.

The “Hash-Lock” Approach. In this approach, due to Weis et al. [22, 23], a tag may be “locked” so that it refuses to reveal its ID until it is “unlocked.”

In the simplest scenario, when the tag is locked it is given a value (or meta-ID) y, and it is only unlocked by presentation of a key or PIN value x such that $y = h(x)$ for a standard one-way hash function h.

In the supermarket example, tags may be locked at check-out time. A consumer could provide a meta-ID y for the tags (perhaps on a loyalty card), and then transmit the unlocking PIN x via some special device (perhaps requiring physical contact) to unlock tags on returning home.

To make this approach workable, it may be necessary for a reader to query a tag to find its meta-ID, so that the reader knows which PIN to use to unlock it. But this may allow tracking of tags via their meta-IDs, defeating their whole purpose. Weis et al. show how to use randomization in the hash function computation to solve this problem.

While this is an effective approach, it seems likely that consumers will find it inconvenient to manage the lock/unlock patterns and associated PINs of more than a small collection of tags. In addition, it is possible that consumers may not even be aware of which objects in their possession carry RFID tags.

The re-encryption approach. Juels and Pappu [9] address the privacy implications of RFID-tags embedded in banknotes, with a scheme where banknote tag serial numbers are encrypted with a law-enforcement public key. The resulting ciphertexts undergo periodic re-encryption to reduce the linkability of different appearances of a given tag.

Because of the severely restricted computing resources of RFID tags, they propose that re-encryption be performed by external computing agents, e.g., publicly provided privacy-enhancing stations in stores. The correct behavior of such re-encryption agents may be verified when banknotes are handled in stores and banks.

The main drawback to this approach is its resource-intensive nature. While RFID tags in their scheme do not perform cryptographic operation and would not be unrealistically costly, the required infrastructure of re-encryption agents and optical verifiers would probably be burdensome.

Golle et al. [8] describe a similar scheme that is more suitable for privacy-protection of RFID tags embedded in consumer goods. They use multiple public keys, thanks to a technique they call “universal re-encryption.” This is an extension of the El Gamal cryptosystem in which it is possible to re-encrypt a ciphertext without knowing the associated public key.

The Golle et al. scheme suffers from the same drawback as that of Juels and Pappu, namely the requirement for an infrastructure of re-encryption devices.

Silent Tree-Walking. Weis et al. [22] correctly note that the threat posed by passive eavesdroppers is more their ability to hear the signals broadcast by the tag reader, which may be picked up many hundreds of meters away, than their ability to hear the signals of an RFID tag, which can only be picked up nearby.

This is unfortunate, since the IDs read by the standard tree-walking singulation protocol can be inferred by hearing merely the signals broadcast by the reader.

Weis et al. show how to encrypt the reader's transmissions, so that a passive eavesdropper cannot infer the IDs being read. Apart from the fact that this does not defend against active attacks, the authors note that their proposal relies on the somewhat unrealistic assumption of a common, secret string shared among tags; this assumption can be removed, however, if the tags can generate their own random pseudo-ID's before singulation.

We note that our selective blocking approach is compatible with this method of protecting reader transmissions from eavesdroppers.

We note that the "silent tree-walking" and "hash-lock" approaches for constructing "smart" RFID tags (and indeed almost any conceivable approach based on smart RFID tags) involve cryptographic operations on tags. Such approaches are thus unlikely to be economically practical for the near future—the RFID chips will be smart but too expensive!

1.6 The Regulation Approach

Garfinkel [7] proposes a different approach based on an "RFID Bill of Rights," which consists of five articles proposed as a voluntary framework for commercial deployment of RFID tags. Included are: (1) the right of the consumer to know what items possess RFID tags, (2) the right to have tags removed or deactivated upon purchase of these items, (3) the right of the consumer to access of the data associated with an RFID tag, (4) the right to access of services without mandatory use of RFID tags, and finally (5) the right to know to when, where, and why the data in RFID tags is accessed.

1.7 Organization

Section 2 describes how tree-walking singulation algorithms work. Section 3.1 then describes how blocker tags work for systems that use tree-walking. We focus on use of the blocker tag as a privacy protection device in section 4. We consider blocker tags as a denial-of-service threat in section 5, and also explore detection measures. We conclude in section 6 with summary recommendations and a discussion of future research topics. In appendix A, we discuss an adaptation of our blocker scheme for use with RFID tags that implement the ALOHA singulation protocol instead of tree-walking.

2 Singulation and Tree-Walking Protocols

As noted earlier, our approach is based on selectively blocking the singulation protocol used by the RFID readers. In this section, we present and discuss the tree-walking singulation protocol, so that we may then in the following section describe how our blocker tags work.

An RFID reader is really only able to communicate with a single RFID tag at a time. If more than one tag responds to a query by the reader (as will often happen naturally, for example, in a supermarket automated checkout), the reader detects a "collision." In this case, it doesn't read accurately any of the information transmitted by the tags. The reader and RFID tags then need to engage in some sort of protocol so that the reader can communicate with the conflicting tags one at a time. Such a protocol is called a "singulation protocol"—it enables the reader to talk to each tag singly.

While there are a number of available singulation protocols, our focus in this paper is on RFID-tag systems that employ a singulation technique known as *tree-walking*, as this singulation technique is (a) likely to be the most common one deployed in practice and (b) supportive of the more flexible modes of blocking proposed here.

RFID-tag systems typically operate at a frequency of either 13.56 Mhz or 915 Mhz. Those operating at 915 Mhz usually rely on tree-walking as a singulation technique [17], and are expected to be the most common type used in the United States [13]. These are the main focus of our work here. Tags operating at 13.56 Mhz usually use what is known as the ALOHA protocol [17] for singulation. We discuss ALOHA tags in appendix A. (Other frequencies, such as 125 kHz and 2.45 GHz, are also used, and employ similar singulation protocols.)

2.1 The Tree-Walking Singulation Algorithm

The tree-walking singulation algorithm enables an RFID-tag reader to identify the serial numbers of nearby tags individually by means of a bit-by-bit query process resembling a depth-first search of a binary tree.

Suppose the tags in a given system bear unique identifiers of some fixed bit-length k (such as $k = 64$, 96, or 128).

Let $\|$ denote the concatenation operator for bit strings.

The set of all possible k-bit identifiers can be viewed as the leaves of a standard binary tree of depth k. The root of this tree has depth 0 and is labeled with the empty string. A node of depth d is labeled with a binary string x of length d; if $d < k$, then the node has a two children at depth $d + 1$: a "left child" with label $x0$, and a "right child" with label $x1$. (Here $x0$ means $x \| 0$ and $x1$ means $x \| 1$.)

We regard the branches of a given node in this tree as bearing labels '0' and '1', respectively associated with the left and right branches. Thus a node at depth d in this tree may be uniquely identified by a binary prefix $B = b_1 b_2 \dots b_d$, representing the sequence of branch labels of branches traversed in a path from the root to the node. It follows that each of the 2^k leaves in the tree has a unique associated k-bit string. We view each such leaf as representing a distinct possible tag serial number.

The tree-walking algorithm is a recursive depth-first search performed by a reader in the following way.

Let the *subtree* of a node denote all its descendents in the tree.

The reader initiates the tree-walking algorithm at the root of the tree.

Starting at a given node $B = b_1 b_2 \dots b_d$, the reader queries all tags bearing serial numbers in the leaves of the corresponding subtree, i.e., all tags whose serial numbers bear the prefix B; all other tags are instructed to remain silent.

The queried tags reply to the reader with the $d + 1$-st bit in their serial numbers; i.e., each tag broadcasts a '0' if it lies in the left subtree of the node B, and a '1' if it lies in the right subtree. Consequently, if there are tags in both the left and right subtrees of B, then the tags together simultaneously broadcast both a '0' and a '1', creating a collision in the broadcast bit.

In this case, when a collision is detected, the reader recurses (sequentially in turn) beginning at its child nodes $B \parallel 0$ and $B \parallel 1$.

If, on the other hand, the tags all reply with only a single bit b, i.e., they all lie in the same subtree, then the reader recurses on the node $B \parallel b$, and ignores the other (empty) subtree.

When the algorithm reaches a leaf (at depth k), it outputs the associated k-bit sequence, which is the serial number of the tag just read. The full output of the algorithm is a list of the ID numbers of all tags within range.

The running time of this algorithm is bounded by the product of k and the number of tags being read. In practice, a shopping cart full of goods should be scannable in a few seconds.

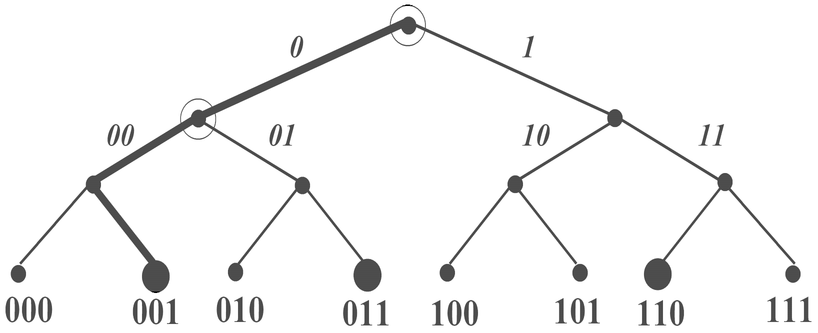

Fig. 1. Tree-walking example. Each tag has a three-bit serial number, corresponding to a leaf in this depth-three binary tree. The tree-walking singulation protocol corresponds to a depth-first search of this tree, restricted to the leaves/ID's in use and their ancestors.

Tree-walking example: We provide a toy example in Figure 1 of how the tree-walking algorithm works. This tree, which is of depth 3, has $2^3 = 8$ tag serial numbers represented at its leaves. The prefixes associated with subtrees are denoted in italics.

In this example, we consider three tags as being present, the '001','011', and '110' tag. These are indicated by large black circles at their respective leaves.

The tree-walking algorithm here first singulates the '001' tag. It does this by following the path denoted by the darkened edges. At two nodes, namely the root of the tree and the root for all tags with a '0' prefix, there are collisions in the bits broadcast by tags, because there are tags present in both the left and right subtrees. We denote these collision-points with hollow circles.

Singulation of the '011' and '110' tags would follow by recursion on the collision points.

2.2 Zones

The tree-walking method has the following nice property, which is exploited in our blocker-tag approach: all tags whose IDs share a common prefix lie in a common subtree.

Thus, for example, since all products produced by a particular manufacturer share a common prefix, all IDs on tags for products of that manufacturer lie in a common subtree. These IDs are all scanned sequentially by the tree-walking algorithm.

More generally, different ID prefixes may correspond to different *zones* (or subtrees) of the space of possible IDs. For example, all IDs beginning with a '1' may be in a "privacy zone," or all IDs beginning with '010' may be in a "recycling zone." The careful allocation of ID prefixes allows the establish of multiple "zones" of IDs that may be useful in conjunction with our blocker RFID approach to consumer privacy. We expand on this idea below.

3 Blocker Tags

We now describe our simple blocker-tag scheme for privacy-protection. We show how blocker tags selectively exploit (i.e., interfere with) the tree-walking singulation protocol. The blocker tag does not engage in an active form of jamming. Rather, by participating in the tag-reading process in a non-compliant (or more accurately, a super-compliant) way, it performs what may be thought of as a kind of passive jamming.

As briefly explained above, a blocker tag simulates the full spectrum of possible serial numbers for tags, thereby obscuring the serial numbers of other tags. The blocker tag effectively overwhelms this process by forcing it to sweep the full space of all possible tag identifiers, which is extremely large.

When carried by a consumer, a blocker tag induces a physical region of privacy protection in which a reader is incapable of singulating tags.

In this paper, we explore two guises of the blocker tag: as a privacy-protection tool, and as a malicious tool.

First, the chip can serve as a privacy-protection tool. As we show, a blocker tag may be naturally designed to prevent singulation across certain restricted ranges of serial numbers. Thus, it is possible to designate a particular zone, i.e., range of serial numbers – say, all those with a leading '1' bit – as subject to the privacy-protection of the blocker tag. As we show, this selective-blocking feature may be used to protect items in the hands of consumers, while at the same time permitting unimpeded reading of tags in commercial environments.

Second, we examine the blocker tag in its malicious guise, namely as a tool for mounting denial-of-service attacks. Such a blocker tag might shield either the full spectrum of serial numbers from reading, or might target a particular range – for example, the set of serial numbers assigned to a particular manufacturer. A blocker tag of this form might be used to disrupt business operations or to help perpetrate petty theft by shielding merchandise from inventory-control mechanisms. We are unaware of any protocol-level technique for circumventing the effects of a malicious blocker tag, but explore simple ways of detecting the presence of such a device.

3.1 How a blocker tag works

The operation of a basic blocker tag is quite simple: It simulates the full set of 2^k possible RFID-tag serial numbers. We may call such a tag a "full blocker" or a "universal blocker."

Thanks to the structure of the tree-walking algorithm, such blocking may be accomplished quite easily. Whenever the reader queries tags in the subtree of a given node B for their next bit value, the blocker tag simultaneously broadcasts both a '0' bit and a '1' bit. (The blocker tag may require two antennae to do this.) This forced collision drives the reader to recurse on all nodes, causing the reader to explore the entire tree.

If the reader had enough time, memory, and processing power to complete the tree-walking algorithm in these circumstances, it would output the entire set of all 2^k possible tag serial numbers.

This set is very large, however – of size at least 2^{64} in even the most basic system – and the reading process is designed to execute very rapidly. In practice, therefore, the reader may be expected to stall after reaching only a few hundred leaves in the tree.

The net effect is that the full blocker tag "blocks" the reading of all tags.

The idea of a special device simulating a number of tags first appears, to the best of our knowledge, in the master's thesis of Steve Weis [23]. Weis imagines the use of such a device in an attack against inventory-control systems – in particular, as a way of spoofing such systems into believing that stolen items are still present in a retail environment. Our application – namely privacy protection – is different, and the implementation details are also different.

The blocker tag can be refined so as to simulate and therefore effectively block just a subset of tags; we call such a blocker a "partial blocker" tag or a "selective blocker" tag.

For example, a selective blocker might reply to the reader only during execution of the tree-walking in the left subtree of the root. This selective-blocking feature would have the effect of obstructing only the reading of tags that bear a '0' prefix in their serial numbers; tags that begin with a '1' bit could be read without interference. In this manner, the selective blocker tag can target a particular zone for protection.

Indeed, a selective blocker tag may be easily and inexpensively created so as to block reading of all tags with an arbitrary prefix or small set of prefixes – a useful feature, as we explain below. (More generally, a selective blocker tag may be designed to simulate – and thus block the reading of – serial numbers satisfying any of a number of simple conditions, such as those matching a given regular expression.)

3.2 Reader-friendly blocking protocol

If we continue along the above line of exploration, where the blocker tags are blocking certain zones (subtrees) and leaving others alone, we see that a problem arises.

For example, if IDs beginning with '0' are blocked, then the reader may never get around to reading IDs beginning with '1'.

Some method is needed for the reader to know not to attempt to read within certain subtrees. That is, the reader needs to know when a subtree is being blocked, so that it can proceed to other parts of the tree without stalling on the blocked subtree.

There are many ways one could imagine revising the tree-walking singulation protocol to make it work efficiently even in the presence of blocker tags, by having the tree-walk ignore subtrees that are being blocked.

For example, when at a given node, the basic tree-walking protocol asks all leaves in the node's subtree to broadcast their "next bit" (the label on the next branch from the node towards the leaf in question). We could augment the protocol by allowing the reader first to pose the special query: "Is the subtree rooted at this node being blocked?" If it is not being blocked, then the reader would proceed to ask the standard next-bit question.

We might also call this "polite blocking," since the blocker tag is being polite by "declaring" which subtrees it is blocking.

Another form of polite blocking would be for a blocker tag to "announce" to readers the policy it is implementing. To do this, we might use of a small, designated range of "virtual" tag serial numbers $t, t+1, \ldots, t+k$, each corresponding to one of a range of standard, pre-specified policies labeled $0, 1, \ldots, k$. In order to indicate that it is implementing privacy policy i, a blocker tag can simulate the presence of a tag with serial number $t+i$. (Such unary representation of policy numbers is important so as to allow a reader that encounters multiple blocker tags to decipher the full policy set.) We discuss the idea of privacy policies in greater detail below.

This approach of "policy announcement" is only viable for signaling one of a small set of pre-established privacy policies. It would work especially well with a small number of privacy-designated zones. In general, policy announcement is less flexible than the approach of permitting any node to declare that its subtree is protected. On the other hand, it may be important not to allow blocker tags to implement an indiscriminately rich set of privacy policies, as a policy can then become a unique identifier – or at least distinct enough to undermine the policy of its bearer.

3.3 Cost considerations

Our blocker-tag approach is particularly attractive because of its very low cost of implementation.

First, *the ordinary consumer-product RFID tags may not need to be modified at all.* (Or, if the privacy zone recommendations below are followed, they only need to be modified slightly to allow flipping of a few initial bits of their IDs.) The RFID tags don't need any expensive cryptography. In terms of overall systems cost, this is the most important consideration, since there are many more consumer-product RFID tags than other types.

Second, the blocker tags themselves can be very cheap; they would consist essentially of just one or two standard RFID tags, with very slight circuit modifications made. If a standard RFID tag can be made for five cents, a blocker tag can probably be manufactured for at most ten cents.

Third, the background implementation is small – a password needs to be managed for each standard RFID tag, to authorize it to change privacy zones (see the description later). This is identical then to what is already proposed for the "kill" command.

Thus, the blocker tag approach is probably as cheap as the "kill" command approach, but is, as we'll see, much more flexible and useful for protecting privacy.

4 The Blocker Tag as Privacy-Protection Tool

To ensure its attractiveness as a widespread tool for protection of consumer privacy, the blocker tag must create little or no disruption of normal RFID-based commercial processes like inventory control. In this respect, a universal blocker tag would be counterproductive: it would provide privacy protection, but at the cost of indiscriminately disrupting all RFID-tag reading in its vicinity.

For the purpose of practical privacy enhancement, we must instead require the use of selective blocker tags. This involves the special designation of one or more zones for privacy protection. Thus a "privacy zone" consists of a restricted range of tag serial numbers targeted for protection (i.e., simulation) by a selective blocker tag. A selective blocker tag disrupts reader execution of the tree-walking algorithm whenever it enters a region identified by the blocker tag as privacy zone; when reading takes place outside of this zone, however, the blocker tag remains inactive.

With the use of privacy zones and some dynamic alteration of tag serial numbers, it is possible to implement a natural range of privacy policies that may simultaneously satisfy the needs of consumers and businesses. We envisage systems in which serial numbers are transferred inside or outside privacy zones depending upon the situations in which they are used.

It is simplest to engineer a selective blocker tag in which the corresponding privacy zone consists of the subtree of a single node. (Recall that this corresponds to the set of serial numbers with some common binary prefix.) Such a zone, for instance, might consist simply of the right half of the serial-number tree, namely all serial numbers whose leading bit consists of a '1'. We provide an example of how a privacy zone of this kind might be used in a retail setting of the future.

Example 1. Privateway Supermarkets makes use of blocker tags whose privacy zone consists of all serial numbers with a leading '1' bit. Packages in Privateway Supermarkets each bear an RFID tag with a unique serial number used for purposes of inventory control. As initially programmed, and while an item is inside the supermarket or its warehouses, the serial number in its RFID tag carries a leading '0' bit. At this point, blocker tags don't disrupt the reading of tags.

When the RFID-tag reader at a cash register scans an item for purchase by a customer, it also transmits a tag-specific key to the RFID tag on the item.[2] This causes the leading bit in the serial number of the tag to flip to a '1'. Privateway Supermarkets provides its customers with free blocker tags. These are available embedded in shopping bags at registers or as stickers to be placed on items.

When Alice returns home from her shopping trip to Privateway Supermarkets, she unmasks items in the privacy zone by detaching them from shopping bags or removing their privacy-enhancing stickers.[3] When she places items in her "smart" refrigerator, an attached RFID reader tallies the contents. (By keeping track of this inventory, Alice's home computer can print out a list of items for purchase on Alice's next trip to the supermarket.)

A simple scheme like this could be naturally incorporated into the EPC-code system of the AutoID center [3]. An EPC code comprises 96 bits, sequentially partitioned as follows:

[2] This key should be secret so as to prevent an attacker from transferring serial numbers arbitrarily into the privacy zone.

[3] To ensure that stickers no longer perform blocking when removed, they might be constructed to deactivate completely upon removal by means, e.g., of detachment of their antennae; bags might similarly be equipped with deactivation mechanisms. Personal blocking devices, of course, may be equipped with on/off or policy-setting switches.

(1) An 8-bit header; (2) A 28-bit “EPC-manager” code, designating the organization that owns the tag; (3) A 24-bit “object-manager” code, designating the class of object as determined by the EPC manager; and (4) A 36-bit serial number that uniquely identifies the object.

Thus we could implement our illustrated privacy scheme by having one of the bits of the object manager code designated a standard “privacy bit”. All blocker tags could then be assigned a unique collective EPC-manager code.

This scheme would be reader-friendly. To determine whether a blocker tag is present, a reader would initially check whether the EPC-manager code for blocker tags is present by following the corresponding path down the identifier tree. (Note that a blocker tag would simulate *all* EPC-manager codes, but a particular one would serve as an agreed-upon indicator of blocking.) The privacy bit in the object-manager code for a tag could be flipped on or off according to the policies of the tag EPC manager.

4.1 Multiple privacy zones

In many cases, it would be useful to have multiple, independent privacy zones. By associating different privacy-enhancing practices with different zones, it would be possible to maintain a collection of overlapping privacy policies. Different types of blocker tags or devices might then be used to implement a variety of privacy policies. We illustrate the idea here with several examples.

Example 2. Suppose that the first two bits of tag serial numbers specify a desired privacy zone ranging from zero to three. Alice might carry a “zone-one” blocker tag in her wristwatch. So as to protect her clothing and personal appliances from scrutiny, all of these items would then be marked with the “zone-one” prefix.

On the other hand, Alice might like to be able to handle groceries without blocking their tags. In this case, on checkout, her grocery items could be marked with the “zone-two” prefix, while privacy stickers for these items carry “zone-two” blocker tags. Thus, when the stickers are removed, Alice can handle them without her wristwatch interfering with the reading process. Alice might choose, on the other hand, for her automobile to implement the strongest level of protection, blocking RFID-tag reading in all four zones.

Example 3. As explained above, the European Central Bank has indicated its intention to embed RFID tags in banknotes. Serial numbers for these tags might be relegated to a special privacy zone for currency.

To protect the privacy of consumers, then, wallets could be equipped with imbedded blocker tags or with credit-card-like devices bearing blocker tags.

The presence of a “currency-zone” blocker tag would be easily detectable, as explained in the next section. Thus, in sensitive locations like airports, law-enforcement officials could take the approach of temporarily sequestering wallets in Faraday cages during security checks. They could then detect the presence of suspicious “currency-zone” blocker tags. In the absence of such tags, or following their identification and removal, it would be possible to monitor large and suspicious currency flows. (The desirable policies are obviously a subject for debate; we merely note here that the availability of blocker tags allows one to consider a realistic range of policies that was heretofore unattainable.)

Law-enforcement officials would also be able to scan banknotes quickly and without impediment when they pass through financial institutions.

Example 4. As illustrated above, tags in consumer items might be constructed so that their serial numbers and other highly individual data can be transferred to a privacy zone. At the same time, to facilitate recycling, tags on plastic items might carry and readily broadcast their polymer-type number (a value that ranges between 1 and 7). This could be accomplished, for instance, by having a special class of rudimentary RFID tags used uniquely for recycling.

A privacy risk in this approach is the effect of "clustering". In particular, the polymer numbers for a multiplicity of objects would together constitute a unique identifier. Most common consumer items made of recyclable plastic, however – e.g., soda bottles – do not remain with a consumer in large quantities for very long.

Another possible use of multiple zones is protection against spillover effects from blocker tags. For example, if Alice is carrying a blocker tag and standing in physical proximity to Bob, then her blocker tag may extend its disruptive effects to the reading of tags carried by Bob. While Bob may be carrying tags whose serial numbers lie in a privacy zone, he may wish to have full control over the circumstances in which they are shielded.

Given a reasonably large collection of privacy zones – say, one hundred – every person might make use of a blocker tag protecting a fixed, random zone, and have his or her items marked accordingly. This would reduce the likelihood of spillover.

It is important to note that there is a tradeoff between individual privacy and the number of possible privacy zones and/or policies. At an extreme, if each blocker tag were to implement a unique policy, then the policy itself would be unique identifier! Thus, the set of possible privacy zones (and likewise the richness of privacy policies) should not be too large in a given RFID-tag system. Otherwise, there is a risk of undermining the very property of privacy at whose enforcement our proposed system of blocker tags aims to begin with.

Blocker tags may, in our view, be available from many sources, Merchants may include them for free with purchased goods, or consumers may be able to buy them at the checkout counter. Consumer rights organizations may supply them for nominal cost. As noted earlier, there is no reason why blocker tags should not be cheaply and widely available.

5 Malicious Blocker Tags

In this section we explore how blocker tags may be used maliciously, and discuss defenses against such behavior.

A blocker tag may be misused to circumvent the intended RFID reader protocol by simulating multiple identifiers. While the legitimate privacy application of the blocker tag also simulates multiple identifiers, the malicious blocker tag does not respect the boundaries of an allowed privacy zone of ID's. A universal blocker tag would hence be malicious by nature.

RFID readers can be designed to cope with the intended blocker behavior within the privacy zone, but their basic functionality requires them to be able to read tags outside of this zone. Thus the malicious blocker tag effectively mounts a denial-of-service attack against the RFID reader protocol. Such attacks might be designed simply to disrupt service, or may be a component of a scam used by petty thieves. We focus here on denial-of-service, rather than the full range of RFID interference strategies.

The malicious blocker tag functions similarly to the privacy tag described above, simulating actual tags. Regardless of detection, the attack will be successful, provided that actual tags in the vicinity may not be distinguished by the reader.

A selective malicious blocker tag might be more sophisticated. It could attempt to simulate a particular distribution of tags in order to avoid detection. Regardless of this distribution, the number of spoofed tags must be large enough to delay significantly the singulation protocol.

Detection of denial-of-service blocker attacks is therefore relatively straightforward. An attack can be assumed to be in progress if the number of perceived RFID tags exceeds some reasonable threshold (for example, 1,000 tags at a checkout line). Such threshold detection is simple and robust, as it does not rely on the exact behavior of the malicious blocker tag. In other words, this approach would work for either universal or selective blocker tags of a malicious kind.

More sophisticated detection mechanisms might rely on the use of prescribed tag ID ranges. For example, the reader could be connected to a database listing every valid tag in the range of identifiers associated with a particular manufacturer (corresponding, e.g., to the "EPC manager" in an EPC). A tag whose identifier lies within the range but isn't on the list could be identified as fraudulent. If tag identifiers are at least partially random, it will be hard for an attacker to guess a valid product identifier. This defense is also not foolproof; for example, it does not protect against spoofing valid tag identifiers that have been recorded previously by the attacker. In practice, this approach would also rely on access to manufacturer databases, which may be impractical in retail settings.

It is conceivable that expensive, special-purpose readers could filter out blocker tags. For example, if a few readers working together could estimate the location of the tags, they could ignore a multitude of fake identifiers originating from a single location. Of course, existing readers are not capable of this hypothetical technique.

We emphasize that with the implementation of our privacy-zoning ideas, selective blocker tags as manufactured for consumer use will not permit abuses like those described here. Attackers can and may come to deploy malicious blocker tags (or related devices) whether or not benign blocker tags become widespread. Thus, concerns about malicious use are not a good reason for avoiding the adoption of benign blocker tags. While it is quite possible that there may be some malicious use of blocker tags, say at a checkout counter, this should be treated as a misdemeanor equivalent to, say, pouring a bottle of syrup on the floor and counter—easily detected and rather straightforward to handle.

6 Conclusions

We have proposed the use of blocker tags as a method for protecting consumer privacy threatened by the pervasive use of RFID tags on consumer products. The use of "selective blocking" by blocker tags enables consumers to "hide" certain RFID tags from scanning when they want to, and "reveal" those same tags for scanning when they want to. By allowing ID prefixes to be rewritten, tags can be moved in or out of “privacy zones” protected by various blocker tags.

We believe that blocker tags are a potent and very useful tool for protecting consumer privacy, and recommend the standardization of their behavior and utilization, along the lines proposed here.

Acknowledgments

We would like to thank Simson Garfinkel, Ravi Pappu, Christopher Rivest, Sanjay Sarma, and Steve Weis for many useful discussions about RFID privacy.

References

1. N. Abramson. The throughput of packet broadcasting channels. *IEEE Trans. On Comm.*, COM-25:117–128, Jan. 1977.
2. Benetton undecided on use of ‘smart tags’. *Associated Press*, 8 April 2003.
3. D.L. Brock. The electronic product code (EPC): A naming scheme for objects. Technical Report MIT-AUTOID-WH-002, MIT Auto ID Center, 2001. Available from http://www.autoidcenter.org.
4. AutoID Center. 13.56 MHz ISM band class 1 radio frequency identification tag interference specification: Candidate recommendation, version 1.0.0. Technical Report MIT-AUTOID-WH-002, MIT Auto ID Center, 2003. Available from http://www.autoidcenter.org.
5. Security technology: Where’s the smart money? *The Economist*, pages 69–70. 9 February 2002.
6. D.M. Ewatt and M. Hayes. Gillette razors get new edge: RFID tags. *Information Week*, 13 January 2003. Available at http://www.informationweek.com/story/IWK20030110S0028.
7. S. Garfinkel. An RFID Bill of Rights. *Technology Review*, page 35, October 2002.
8. P. Golle, M. Jakobsson, A. Juels, and P. Syverson. Universal re-encryption for mixnets, 2002. In submission.
9. A. Juels and R. Pappu. Squealing Euros: Privacy protection in RFID-enabled banknotes. In R. Wright, editor, *Financial Cryptography ’03*. Springer-Verlag, 2003. To appear.
10. L. Kleinrock and S.S. Lam. Packet switching in a multi-access broadcast channel: performance evaluation. *IEEE Trans. On Comm.*, COM-23:410–423, Apr. 1975.
11. D. McCullagh. RFID tags: Big Brother in small packages. *CNet*, 13 January 2003. Available at http://news.com.com/2010-1069-980325.html.
12. mCloak: Personal / corporate management of wireless devices and technology, 2003. Product description at www.mobilecloak.com.
13. R. Pappu, 2003. Personal communication.
14. Prada’s smart tags too clever? *Wired News*, 27 October 2002.
15. L.G. Roberts. ALOHA packet system with and without slots and capture. *Comput. Commun. Rev.*, 5:28–42, Apr. 1975.
16. S. E. Sarma, S. A. Weis, and D.W. Engels. Radio-frequency identification systems. In Burton S. Kaliski Jr., Çetin Kaya Koç, and Christof Paar, editors, *CHES ’02*, pages 454–469. Springer-Verlag, 2002. LNCS no. 2523.
17. S. E. Sarma, S. A. Weis, and D.W. Engels. RFID systems, security and privacy implications. Technical Report MIT-AUTOID-WH-014, AutoID Center, MIT, 2002.
18. S. E. Sarma, S. A. Weis, and D.W. Engels. Radio-frequency-identification security risks and challenges. *CryptoBytes*, 6(1), 2003.
19. S.E. Sarma. Towards the five-cent tag. Technical Report MIT-AUTOID-WH-006, MIT Auto ID Center, 2001. Available from http://www.autoidcenter.org.
20. R. Shim. Benetton to track clothing with ID chips. *CNET*, 11 March 2003. URL: http://news.com.com/2100-1019-992131.html.
21. K. Takaragi, M. Usami, R. Imura, R. Itsuki, and T. Satoh. An ultra small individual recognition security chip. *IEEE Micro*, 21(6):43–49, 2001.

22. S. A. Weis, S. Sarma, R. Rivest, and D. Engels. Security and privacy aspects of low-cost radio frequency identification systems. In *First International Conference on Security in Pervasive Computing*, 2003. To appear.
23. S.A. Weis. Radio-frequency identification security and privacy. Master's thesis, M.I.T. June 2003 (expected).
24. J. Yoshida. Euro bank notes to embed RFID chips by 2005. *EE Times*, 19 December 2001. Available at http://www.eetimes.com/story/OEG20011219S0016.

A Blockers for the ALOHA Singulation Protocol

A.1 The ALOHA protocol

RFID tags that operate in low frequency ranges, e.g., the 13.56 Mhz range, generally employ a singulation protocol known as ALOHA [1, 10, 15]. Use of the ALOHA protocol in this case aims at reducing reader-to-tag communications in order to meet restrictive electromagnetic compatability regulations. In this appendix, we show how our blocker design can be adapted for systems employing ALOHA singulation. Our focus in particular is on the ALOHA specification published by the AutoID Center for tags in the 13.56 Mhz class [4]. This standard employs a classic protocol variant known as "slotted" ALOHA in which a given tag broadcasts its ID during a designated, independent time interval known as a "slot."

To sketch the ALOHA protocol, let T_i denote the ID of some tag i involved in the reading protocol. The function f here is a general, preprogrammed function for scheduling tag responses. (In the AutoID Center standard, this function f is left unspecified, and presumably may be selected by individual tag manufacturers.) "Slotted" ALOHA involves essentially the following few steps:

1. The reader broadcasts S, the number of designated slots, and a random value R.
2. Tag i computes a slot value $s_i = f(T_i, R, S) \in [0, 1, \ldots, S-1]$.
3. During slot s_i, tag i transmits T_i to the reader.

In the advent of a collision in given slot s_i, i.e., a simultaneous reply from multiple tags, a reader is in general unable to receive any transmission. In other words, tag transmissions are lost. The ALOHA protocol aims to avoid such collisions through randomized scheduling of replies and selection of an appropriately large slot allocation S. There are a number of techniques for addressing the problem of collisions. For example, if many collisions occur, the protocol may be re-run with a larger value S.

An additional feature of the slotted ALOHA protocol specified by the AutoID Center is referred to as a *selection mask* [4]. This is a prefix broadcast by the reader to specify a subset of tags that should respond to its query. When a k-bit selection mask σ is specified, a tag only transmits to the reader if the σ is an exact prefix of T_i, i.e., matches the first k bits. This selection mask is optional; we denote by ϕ a null selection mask, i.e., the absence of any selection-mask specification. As a minor technical matter, when a selection mask σ is specified, a tag transmits only substring of its ID T_i that follows σ.

A.2 ALOHA blocker design

Blocker tags for the ALOHA protocol may operate according to essentially the same principle as those for tree-walking. In particular, an ALOHA blocker simulates transmission collisions during selected time slots. We now describe two different techniques for producing such blocking behavior in a selective manner.

The presence of selection masks in the AutoID Center protocol suggests a relatively straightforward selective-blocking strategy. A privacy zone may be specified in terms of a set of arbitrary-length prefixes $\Sigma = \{\sigma_1, \sigma_2, \ldots, \sigma_m\}$. If the reader specifies a selection mask σ such that σ_i is a prefix of σ or vice versa, then the blocker tag simulates collisions for all slots. Otherwise the blocker remains silent. (Note that if Σ is not empty and $\sigma = \phi$, then the blocker will block all slots.)

Example 5. Suppose that $\Sigma = \{$''0', '11'$\}$ for a given blocker tag. This tag block the reading of all tags whose ID T_i has a leading '0' bit or the leading pair of bits '11'. Thus, if the reader specifies any of the following selection masks, for example, then the blocker will be activated: '0', '1', '01', '110', ϕ. In contrast, if the reader specifies any of the following example selection masks, then the blocker will remain silent: '11','110', '11000'.

A drawback of this approach is that in order to read all tags lying outside the privacy zone specified by Σ, the reader may have to make multiple queries. In the above example, for instance, the reader would have to make queries under selection masks '110' and '111' in order to read all tags outside the privacy zone. Our view is that this should not be problematic provided that the privacy-zone specification Σ is suitably concise.

A second approach to blocking is possible through simulation of collisions only during selected time slots. This approach relies critically on the form of the function f. In order to protect tags in a privacy zone P, i.e., every tag with an ID $T \in P$, a blocker tag may simulate collisions in every time-slot s such that $s = f(R, T, S)$ for some $T \in P$. In general, this approach may result in the blocking of tags outside the privacy zone P. Given suitable selection of f and P, however, blocking behavior may proceed exactly as desired. We illustrate this with another example.

Example 6. Suppose that $S = 2^e$ for some value e, and that f simply computes a bitwise XOR of e-bit random value R and the e-bit prefix of tag ID T_i. In this case, it would be easy to create a privacy zone consisting of all tag IDs with a leading '1' bit, i.e., to permit reading only of tags whose ID carries a leading '0' bit. Let r represent the leading bit of R. The blocker tag would simply simulate a collision in any slot s whose leading bit is equal to r XOR 1.

A drawback of this second approach is its dependence on the function f implemented in a given tag. Without a widely implemented choice of f, blockers would not be able to achieve a consistent privacy policy.

A.3 Reader-friendly blocking

It remains valuable in ALOHA-based systems for blocker tags to block in a "polite" way, namely to specify their policies to readers. Clearly, the technique described in section 3.2

for tree-walking, in which a subtree is “marked” as subject to blocking, will not work in the ALOHA case. A number of other strategies are possible, however, of which we sketch one here.

By analogy with the “virtual” tag idea described in section 3.2, we may specify a special prefix σ^* for blocker tags in the ALOHA protocol. The ID T_i of a blocker tag i then assumes the form $T_i = \sigma^* \parallel \rho_i \parallel P_i$. Here $\parallel$ denotes string concatenation. The symbol ρ_i denotes a random value (of some appropriate length) specific to blocker tag i. The function of ρ_i is to prevent collisions between blocker tags, i.e., to randomize the computation of the slot s, as we will show. P_i denotes a bitstring specifying the privacy policy of the blocker tag i.

In order to learn the full set of privacy policies enforced by blocker tags within its vicinity, a reader issues an initial query under selection mask σ^*. Blocker tags respond then in a manner similar to that of ordinary tags. In particular, each blocker transmits its policy P_i in time slot $f(R, \rho, S)$. In constrast to an ordinary tag, a blocker does not transmit any other portion of T_i. The value ρ, in particular, should not be transmitted, as it would serve as a unique identifier. The reader thus receives the full set of policies of responding blockers.

A blocker policy P may assume any of a number of forms. It might, for instance, be an encoded list of nodes whose corresponding subtrees lie in the privacy zone of the blocker tag, i.e., a set Σ of blocked prefixes. Alternatively, it simply consist of a standardized privacy-zone identifier.