Security and Privacy Aspects of Low-Cost Radio Frequency Identification Systems

Stephen A. Weis[1], Sanjay E. Sarma[2], Ronald L. Rivest[1] and Daniel W. Engels[2]

[1] Laboratory for Computer Science
[2] Auto-ID Center
Massachusetts Institute of Technology
Cambridge, MA 02139, USA
{sweis,sesarma,rivest,dwe}@mit.edu

Abstract. Like many technologies, low-cost Radio Frequency Identification (RFID) systems will become pervasive in our daily lives when affixed to everyday consumer items as "smart labels". While yielding great productivity gains, RFID systems may create new threats to the security and privacy of individuals or organizations. This paper presents a brief description of RFID systems and their operation. We describe privacy and security risks and how they apply to the unique setting of low-cost RFID devices. We propose several security mechanisms and suggest areas for future research.

1 Introduction

Radio Frequency Identification (RFID) systems are a common and useful tool in manufacturing, supply chain management, and inventory control. Industries as varied as microchip fabrication, automobile manufacturing, and even cattle herding have deployed RFID systems for automatic object identification. For over twenty years, consumer items have been identified with optical barcodes. One familiar optical barcode is the Universal Product Code (UPC), designed in 1973 [30] and found on many consumer products. More recently, RFID has made inroads into the consumer object identification market. Silicon manufacturing advancements are making low-cost RFID, or "smart label", systems an economical replacement for optical barcode.

RFID systems consist of radio frequency (RF) tags, or transponders, and RF tag readers, or transceivers. Tag readers *interrogate* tags for their contents by broadcasting an RF signal. Tags respond by transmitting back resident data, typically including a unique serial number. RFID tags have several major advantages over optical barcode systems. Tag data may be read automatically: without line of sight, through non-conducting materials such as paper or cardboard, at a rate of several hundred tags per second, and from a range of several meters. Since tags typically are a silicon-based microchip, functionality beyond simple identification may be incorporated into the design. This functionality might range from integrated sensors, to read/write storage, to supporting encryption and access control. Three example tags are shown in Figure 1.

The potential benefits of a pervasive low-cost RFID system are enormous. Worldwide, over 5 billion barcodes are scanned daily [8]. However, barcodes are typically

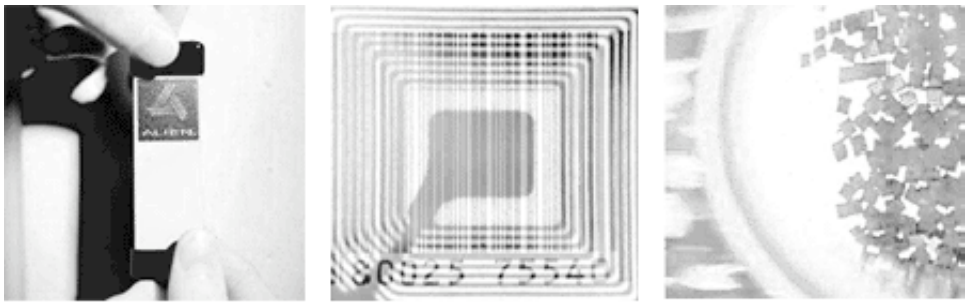

Fig. 1. A passive RFID tag, an RFID tag with a printed barcode, and dust-sized RFID microchips.

scanned only once during checkout. By integrating a unified identification system on all levels of the supply chain, all parties involved in the lifespan of a product could benefit. This includes not only manufacturers and retailers, but also consumers, regulatory bodies such as the United States Food and Drug Administration (FDA), and even the waste disposal industry. The potential cost savings will likely make RFID tags one of the most widely deployed microchips in history, illustrated by the recent purchase of 500 million low-cost RFID tags by a major consumer product manufacturer [23].

Unfortunately, the universal deployment of RFID devices in consumer items may expose new security and privacy risks not present in closed manufacturing environments. Corporate espionage is one risk. Retail inventory labeled with unprotected tags could be monitored and tracked by a business' competitors. Personal privacy may also be compromised by extracting data from unprotected tags. Most consumers would prefer to keep the RFID tagged contents of their pockets or shopping bags private. Another risk is the violation of "location privacy": the tracking of an individual by the tags they carry.

Most manufacturing processes currently deploying RFID systems are for higher value items, allowing tag costs to be in the US$0.50-US$1.00 range. Tags priced in this range could support basic cryptographic primitives or tamper-resistant packaging, similar to many smart card designs. Unfortunately, to achieve significant consumer market penetration RF tags will need to be priced in the US$0.05-US$0.10 range and will need to be easily incorporated into most paper packaging. At this price range, providing strong cryptographic primitives is currently not a realistic option. Any viable tag and reader designs must take into account security and privacy risks, while not exceeding this low-cost range. This places the burden of supporting security on the readers, whose costs are less restrictive.

General low-cost RFID research is part of ongoing work at the MIT Auto-ID Center [21]. An overview of RFID systems and their security implications is available in [27]. Issues explored in the context of smart cards are most closely related to the resource scarce environment of RFID devices. Relevant security issues are addressed in a broad range of smart card and tamper resistant hardware literature. Cost and security tradeoffs of smart cards are analyzed in [1]. RFID tags may operate in insecure environments

or subject to intense physical attacks. An analysis of smart card operation in hostile environments is presented in [9]. An comprehensive overview of many physical attacks and countermeasures appears in [31]. Specific lower cost physical attacks are detailed in [2] and are part of ongoing research at the University of Cambridge's TAMPER Lab [29].

Many results pertaining to implementations of cryptographic primitives are relevant to RFID devices. Cautionary information regarding the implementation of AES in smart cards is presented in [7]. Being passively powered and relying on a wireless interface may make RFID devices especially susceptible to fault induction, timing attacks or power analysis attacks, highlighted in [4, 16, 15] and [14]. Location privacy risks present in Bluetooth technology and relevant to RFID systems are addressed in [12].

In this paper, Section 2 gives a brief introduction to RFID system components, describes the interface between tags and readers, and presents estimates of the capacities of current low-cost tags. Section 3 details various privacy and security risks of a low-cost RFID system deployed in everyday consumer items. Section 4 states assumptions about the security properties of a low-cost RFID system. Under these assumptions, Section 5 offers several proposals for addressing the concerns of Section 3, specifically a hash-based access control scheme in Section 5.1, a randomized access control scheme in Section 5.2, and an improved anti-collision algorithm in Section 5.3. Finally, Section 6 discusses open questions and areas for future research.

2 RFID System Primer

RFID systems are composed of three key elements:

- the RFID tag, or *transponder*, carries object identifying data.
- the RFID reader, or *transceiver*, reads and writes tag data.
- the back-end database associates records with tag data collected by readers.

Every object to be identified in an RFID systems is physically labeled with a tag. Tags typically are composed of a microchip for storage and performing logical operations, and a coupling element, such as an antenna coil, used for wireless communications. Memory on tags may be read-only, write-once read-many, or fully rewritable.

Tag readers *interrogate* tags for their contents through an RF interface. As well as an RF interface to the tags, readers may contain internal storage, processing power, or an interface to back-end databases to provide additional functionality.

Tags may either be *actively* or *passively* powered. Active tags contain an on-board power source, such as a battery, while passive tags must be inductively powered via an RF signal from the reader. The distance a reader may interrogate tags from is limited by the tag's power. Consequently, active tags may be read from a greater distance than passive tags. Active tags may also record sensor readings or perform calculations in the absence of a reader. Passive tags can only operate in the presence of a reader and are inactive otherwise.

Readers may use tag contents as a look-up key into a database storing product information, tracking logs, or key management data. Independent databases may be built by anyone with access to tag contents, allowing unrelated parties to build their own

applications on any level of the supply chain. The back-end database may also perform functions on behalf of either the readers or tags.

Readers must be able to address a particular tag, or *singulate* it, from among a population of many tags. During singulation, multiple tags responses may interfere with each other, necessitating an anti-collision algorithm. Anti-collision algorithms may either be probabilistic or deterministic. A familiar probabilistic algorithm is the Aloha scheme [3, 20] used in Ethernet local area networks. In the tag-reader context, tags avoid collisions with other tags by responding to reader queries at random intervals. In the event of a collision, the culprit tags wait for another, usually longer, random interval before trying again. Higher densities of tags will result in a higher collision rate and degraded performance.

A simple deterministic algorithm is the binary tree-walking scheme. In this scheme, a reader queries all nearby tags for the next bit of their ID number. If the reader detects a collision then at least two tags among the population have different bit values in that position of the ID. The reader will send a response bit indicating which tags should continue with the protocol and which should cease responding. Each choice of bit represents choosing a branch in a binary tree. The leaves of the tree correspond to tag ID numbers. Assuming the tags have unique IDs, after walking to a leaf in the tree, a reader have singulated a tag. Benefits of binary tree-walking include simple tag implementation and efficiently broadcasting only the bits of an ID to singulate any tag.

A ubiquitous low-cost RFID system would most likely require the use of passive tags. Tight cost requirements make these tags extremely resource-scarce environments. Power consumption, processing time, storage, and gate count are all highly limited. A practical US$0.05 design, such as those proposed by the MIT Auto-ID Center [21, 26], may be limited to hundreds of bits of storage, roughly 500-5,000 gates and a range of a few meters.

The resources available in a low-cost RFID tag are far less than what is necessary for public key cryptography, even a resource-efficient scheme such as NTRU [11, 22]. Hardware implementations of symmetric encryption algorithms like AES typically have on the order of 20,000-30,000 gates [6], beyond what is available for an *entire* low-cost RFID design. Standard cryptographic hash functions such as SHA-1 [6] are also likely to be too costly for several years. Even the aptly named Tiny Encryption Algorithm [32, 33] is too costly for today's low-cost RFID tags, although may be feasible in the near future.

3 Security and Privacy Risks

RFID tags may pose security and privacy risks to both organizations and individuals. Unprotected tags may have vulnerabilities to eavesdropping, traffic analysis, spoofing or denial of service. Unauthorized readers may compromise privacy by accessing tags without adequate access control. Even if tag contents are protected, individuals may be tracked through predictable tag responses; essentially a traffic analysis attack violating "location privacy". Spoofing of tags may aid thieves or spies. Saboteurs could threaten the security of systems dependent on RFID technology through denial of service.

Any parties with their own readers may interrogate tags lacking read access control, although only within a relatively short tag read range of a few meters. While anyone could also scan nearby optical barcodes, they cannot do so wirelessly at a rate of hundreds of reads per second. The very properties making RFID technology attractive in terms of efficiency make it vulnerable to eavesdropping.

Aggregate logistics and inventory data hold significant financial value for commercial organizations and their competitors. A store's inventory labeled with unprotected tags may be monitored by competitors conducting surreptitious scans. Sales data may be gleaned by correlating changes over time. Individuals carrying items with unsecured tags are vulnerable to privacy violations. A nearby eavesdropper could scan the contents of your pockets or bag; valuable data to nosy neighbors, market researchers or thieves in search of ripe victims.

Another important privacy concern is the tracking of individuals by RFID tags. A tag reader at a fixed location could track RFID-labeled clothes or banknotes carried by people passing by. Correlating data from multiple tag reader locations could track movement, social interactions, and financial transactions. Concerns over location privacy were recently raised when a major tire manufacturer began embedding RFID tags into all their products [24]. Even if the tags only contain product codes rather than unique serial numbers, individuals could still be tracked by the "constellation" of products they carry. Someone's unique taste in brands could betray their identity.

In addition to threats of passive eavesdropping and tracking, an infrastructure dependent on RFID tags may be susceptible to denial of service attacks or tag spoofing. By spoofing valid tags, a thief could fool automated checkout or security systems into thinking a product was still on a shelf. Alternatively, a thief could rewrite or replace tags on expensive items with spoofed data from cheaper items. Saboteurs could disrupt supply chains by disabling or corrupting a large batch of tags.

4 RFID Security Settings and Assumptions

To address the security risks of low-cost RFID tags, we will first state a set of assumptions about the operation of the system. Assuming a minimalist approach, tags will be passive and provide only simple read-only identification functionality. We will arbitrarily assume our tags contain a few hundred bits of storage and have an operating range of a few meters.

In 2003, cost requirements dictate that low-cost tags may have 200-2000 gates available for security. This is far below what is feasible for standard public-key or symmetric encryption, including efficient algorithms such as NTRU or TEA [11, 32]. Furthermore, performance requirements dictate that at least 100-200 tags must be able to be read each second, which limits the clock cycles available for security protocols. Power consumption may also be a limiting factor, although highly dependent on the particular implementation.

We assume tag memory is insecure and susceptible to physical attacks [29, 31] revealing their entire contents. This includes a myriad of attacks such as shaped charges, laser etching, ion-probes, TEMPEST attacks, clock glitching and many others. Fortunately, these attacks require physical tag access and are not easily carried out in public

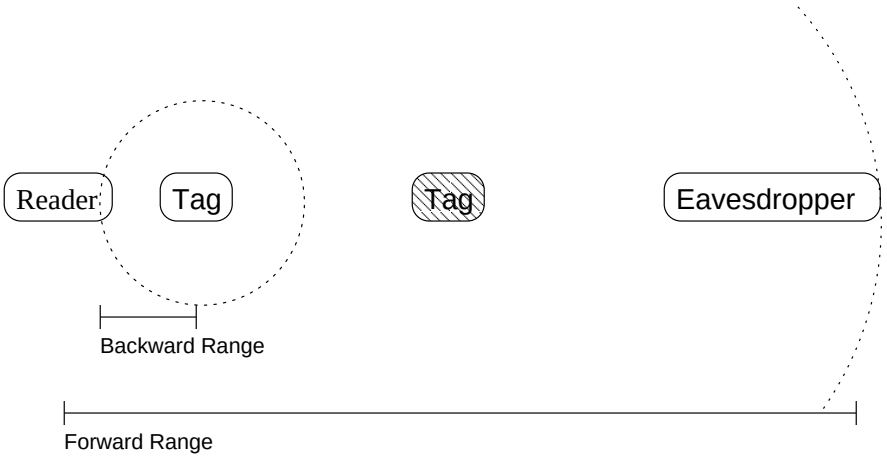

Fig. 2. Forward vs. Backward Channels: The reader will detect the nearby tag, but cannot detect the shaded tag. A distant eavesdropper may monitor the forward channel, but not the tag responses.

or on a wide scale without detection. Privacy concerns are rather moot if someone can remove a tag or steal the item it is attached to without detection. The key point is that tags cannot be trusted to store long-term secrets, such as shared keys, when left in isolation.

Tags may also be equipped with a physical contact channel, as found on smart cards, for critical functions or for "imprinting" tags with secret keys [28]. Additionally, we may assume the tag packaging contains some optical information such as a barcode or human-readable digits. This information may corroborate tag data, as in the design presented in [13].

Tag readers are assumed to have a secure connection to a back-end database. Although readers may only read tags from within the short (e.g. 3 meter) tag operating range, the reader-to-tag, or *forward* channel is assumed to be broadcast with a signal strong enough to monitor from long-range, perhaps 100 meters. The tag-to-reader, or *backward* channel is relatively much weaker, and may only be monitored by eavesdroppers within the tag's shorter operating range. Generally, it will be assumed that eavesdroppers may only monitor the forward channel without detection. This relationship is illustrated in Figure 2

Tags will be assumed to have a mechanism to reveal their presence called a *ping*. Anyone may send a ping, which tags respond to with a non-identifying signal. Tags are also equipped with a *kill* command rendering them permanently inoperable. The kill command may be assumed to be a slow operation which physically disables the tag; perhaps by disconnecting the antenna or short circuiting a fuse.

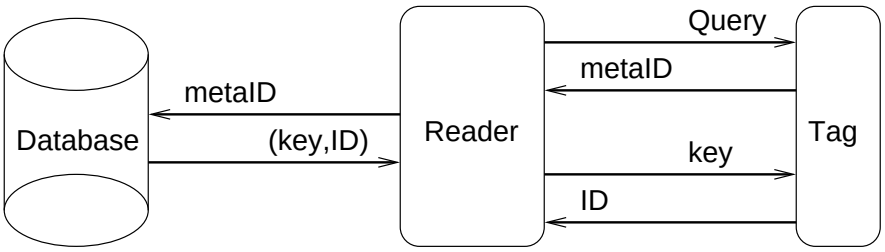

Fig. 3. Hash-Locking: A reader unlocks a hash-locked tag.

5 Security Proposals

5.1 Hash-Based Access Control

Accepting the resource limitations of low-cost tags, we offer a simple security scheme based on one-way hash functions [19]. In practice, a hardware-optimized cryptographic hash would suffice. Each hash-enabled tag in this design will have a portion of memory reserved for a temporary *metaID* and will operate in either a locked or unlocked state.

To lock a tag, a tag owner stores the hash of a random key as the tag's metaID, i.e. $metaID \leftarrow hash(key)$. This may occur either over the RF channel or a physical contact channel for added security. After locking a tag, the owner stores both the key and metaID in a back-end database. Upon receipt of a metaID value, the tag enters its locked state. While locked, a tag responds to all queries with only its metaID and offers no other functionality.

To unlock a tag, the owner queries the metaID from the tag, looks up the appropriate key in the back-end database and finally transmits the key to the tag. The tag hashes the key and compares it to the stored metaID. If the values match, it unlocks itself and offers its full functionality to any nearby readers. This protocol is illustrated in Figure 3. To prevent hijacking of unlocked tags, they should only be unlocked briefly to perform a function before being locked again.

Based on the difficulty of inverting a one-way hash function, this scheme prevents unauthorized readers from reading tag contents. Spoofing attempts may be detected under this scheme, although not prevented. An adversary may query a tag for its metaID, then later spoof that tag to a legitimate reader in a replay attack. A legitimate reader will reveal the key to the spoofed tag. However, the reader may check the contents of the tag (often collectively referred to as a tag's ID) against the back-end database to verify that it is associated with the proper metaID. Detecting an inconsistency at least alerts a reader that a spoofing attack may have occurred.

The hash-lock scheme only requires implementing a hash function on the tag and managing keys on the back-end. This is a relatively low-cost requirement and may be economical in the near future. This scheme may be extended to provide access control for multiple users or to other tag functionality, such as write access. Tags may still function as object identifiers while in the locked state by using the metaID for database lookups. This allows users, such as third-party subcontractors, to build their own databases and to take advantage of tag functionality without necessarily owning

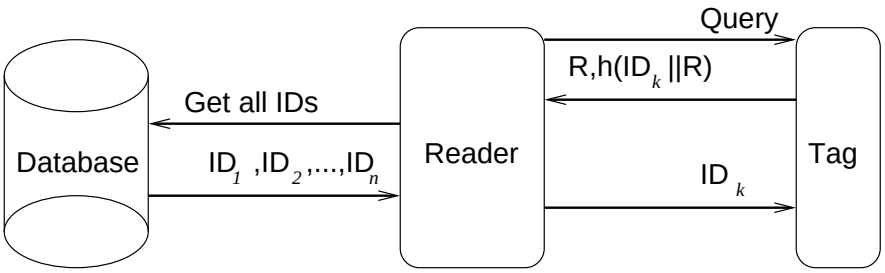

Fig. 4. Randomized Hash-Locking: A reader unlocks a tag whose ID is k in the randomized hash-lock scheme.

the tags. Unfortunately, since the metaID acts as an identifier, tracking of individuals is possible under this scheme.

5.2 Randomized Access Control

Preventing the tracking of individuals motivates an additional mode of operation. While in this mode, a tag must not respond predictably to queries by unauthorized users, but must still be identifiable by legitimate readers. We present a practical heuristic based on one-way hash functions, best suited for consumers with a small number of tags. We also offer a theoretically stronger variant based on pseudo-random functions (PRFs).

As in Section 5.1, tags are equipped with a one-way hash function, but now also have a random number generator. Tags respond to reader queries by generating a random value, r, then hashing its ID concatenated with r, and sending both values to the reader. That is, tags respond to queries with the pair $(r, h(ID||r))$, where r is chosen uniformly at random. This protocol is illustrated in Figure 4. A legitimate reader identifies one of its tags by performing a brute-force search of its known IDs, hashing each of them concatenated with r until it finds a match. Although impractical for retailers, this mode is feasible for owners of a relatively small number of tags.

This scheme may suffice in practice, but is not theoretically robust. The formal definition of a one-way function only establishes the difficulty of inverting the function output [19, 10]. There is no provision of secrecy, technically allowing bits of the input to be revealed. We must use a stronger primitive to ensure ID bits are not leaked.

To address this issue, suppose each tag shares a unique secret key k with the reader and supports a pseudo-random function ensemble, $\mathcal{F} = \{f_n\}_{n\in\mathbb{N}}$. When queried, tags will generate a random value, r, and reply with $(r, ID \oplus f_k(r))$. The reader will once again perform a brute-force search, using all its known ID/key pairs to search for a match.

A minor fix allows readers to only store tag keys on the back-end, without needing to also store the tag IDs. Tags may pad their ID its hash, and reply with $(r, (ID||h(ID)) \oplus f_k(r))$. Readers may identify tags by computing $f_k(r)$ for all their known keys, XORing it with the second part of the tag's response, and searching for a value ending in the form $(x||h(x))$. To anyone without the key value, the tag's output is random and meaningless.

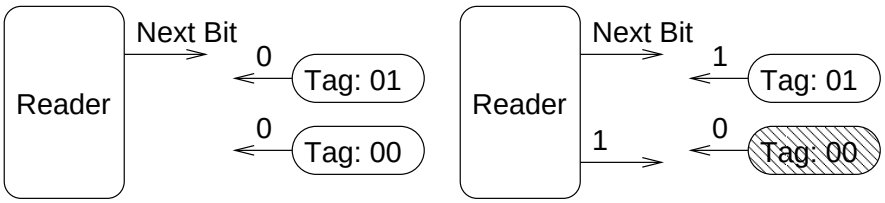

Fig. 5. Silent Tree Walking: The left-hand figure illustrates reading the first bit, which does not collide. The right-hand figure illustrates a collision. To singulate tag 01, the reader responds with "Last Bit" $\oplus$ "Tag 01" $= 0 \oplus 1 = 1$. Tag 01 proceeds, while the shaded tag 00 ceases the protocol.

It is unknown whether PRF ensembles may be implemented with significantly fewer resources than symmetric encryption. There may be no be practical difference in the context of low-cost RFID tags. Many symmetric encryption algorithms employ PRFs as a core building block in a Luby-Rackoff style design [18]. The minimal hardware complexity of a PRF ensemble remains an open problem [17].

5.3 Silent Tree Walking and Backward Channel Key Negotiation

One security concern is the strong signal of the reader-to-tag forward channel. Eavesdroppers may monitor this channel from hundreds of meters and possibly deriving tag contents. Of particular concern is the binary tree walking anti-collision algorithm, because the reader broadcasts each bit of the singulated tag's ID.

We present a variant of binary tree walking which does not broadcast insecure tag IDs on the forward channel and does not adversely affect performance. Assume a population of tags share some common ID prefix, such as a product code or manufacturer ID. To singulate tags, the reader requests all tags to broadcast their next bit. If there is no collision, then all tags share the same value in that bit.

A long-range eavesdropper can only monitor the forward channel and will not hear the tag response. Thus, the reader and the tags effectively share a secret, namely the bit value. If no collisions occur, the reader may simply ask for the next bit, since all tags share the same value for the previous bit. When a collision does occur, the reader needs to specify which portion of the tag population should proceed.

Since we assumed the tags shared some common prefix, the reader may obtain this prefix on the backward channel. The shared secret prefix may be used to conceal the value of the unique portion of the IDs. Suppose we have two tags with ID values $b_1 b_2$ and $b_1 \overline{b_2}$. The reader will receive b_1 from both tags without a collision, then will detect a collision on the next bit. Since b_1 is secret from long-range eavesdroppers, the reader may send either $b_1 \oplus b_2$ or $b_1 \oplus \overline{b_2}$ to singulate the desired tag without revealing either bit. Figure 5 illustrates a reader performing silent tree walking on two bits.

Eavesdroppers within the range of the backward channel will obviously obtain the entire ID. However, this silent tree walking scheme does effectively protect against long-range eavesdropping of the forward channel with little added complexity. Performance is identical to regular tree walking, since a tag will be singulated when it has broadcast its entire ID on the backward channel.

Readers may take advantage of the asymmetry of the forward and backward channels to transmit other sensitive values. Suppose a reader needs to transmit the value v to a singulated tag. That tag can generate a random value r as a one-time-pad and transmit it in the clear on the backward channel. The reader may now send $v \oplus r$ over the forward channel. If eavesdroppers are outside the backward channel, they will only hear $v \oplus r$, and v will be information theoretically secure.

Another deterrent to forward channel eavesdropping is to broadcast "chaff" commands from the reader, intended to confuse or dilute information collected by eavesdroppers. By negotiating a shared secret, these commands could be filtered, or "winnowed", by tags using a simple MAC. This procedure is detailed in [25].

5.4 Other Precautions

Several other measures may be taken to strengthen RFID systems. First, RFID-enabled environments should be equipped with devices to detected unauthorized read attempts or transmissions on tag frequencies. Due to the strong signal strength in the forward channel, detecting read attempts is fairly simple. Deploying read detectors helps identify unauthorized read requests or attempts to jam tag operating frequencies.

Another measure to detect denial of service is to design tags which "scream" when killed, perhaps by transmitting a signal over a reserved frequency. RFID enhanced "smart shelves" may be designed to detect the removal of items, unauthorized read attempts or the killing of tags.

To enable end users to access the functionality of tags affixed to items they have purchased, a master key could be printed within a product's packaging, possibly as a barcode or decimal number. A similar mechanism is proposed for banknotes in [13]. After purchasing an item, a consumer could use the master key to toggle a tag from the hash-lock mode of Section 5.1 to the randomized mode of Section 5.2. The master key may also function as a key recovery mechanism, allowing users to unlock tags they have lost the keys to. Since the master key must be read optically from the interior of a package, adversaries cannot obtain it without obtaining the package itself. For further security, all functions using the master key could be required to use a physical contact channel, rather than RF.

Two final precautions take advantage of the physical properties of passively powered tags. First, readers should reject tag replies with anomalous response times or signal power levels. This is intended as a countermeasure to spoofing attempts by active devices with greater operating ranges than passive tags. Readers may also employ frequency hopping to avoid session hijacking. Passive tags may be designed such that their operating frequency is completely dictated by the reader. This makes implementing random frequency hopping trivial, since tags and readers do not need to synchronize random hops. Readers can just change frequencies, and the tags will follow.

6 Future Research

An area of research which will greatly benefit RFID security and privacy is the development of hardware efficient cryptographic hash functions, symmetric encryption,

message authentication codes and random number generators. General advances in circuit fabrication and RFID manufacturing will lower costs and allow more resources to be allocated for security features. Continued research into efficient symmetric encryption algorithms, such as TEA [32, 33], may yield algorithms appropriate for low-cost RFID devices. One open question from Section 5.2 is whether pseudo-random function ensembles can be implemented with significantly less complexity than symmetric encryption. Designing efficient implementations of perfect one-way functions [5] may be a relevant avenue of research as well.

New RFID protocols resistant to eavesdropping, fault induction and power analysis need to be developed. The silent tree walking algorithm presented in Section 5.3 offers protection against long range eavesdropping, but is still vulnerable to nearby eavesdroppers and fault induction. It also requires that a population of tags share a common prefix unknown to eavesdroppers, which is not always a valid assumption. In general, readers and tags must be designed to gracefully recover from interruption or fault induction without compromising security.

With new technology advances allowing more features to be incorporated into tags, the line between RFID devices, smart cards, and general purpose computers will blur. Research benefiting RFID security today will aid in development of secure ubiquitous computing systems in the future. Recognizing inherent privacy or security threats of RFID systems will also help guide policy decisions regarding the obligations of RFID manufacturers and the privacy rights of end users.

7 Acknowledgments

Thanks to Peter Cole and Tom Scharfeld for RFID design contributions. Thanks to Levente Jakab and Simson Garfinkel for input and editorial review.

References

[1] Martin Abadi, Michael Burrows, C. Kaufman, and Butler W. Lampson. Authentication and Delegation with Smart-cards. In *Theoretical Aspects of Computer Software*, pages 326–345, 1991.

[2] Ross Anderson and Markus Kuhn. Low Cost Attacks on Tamper Resistant Devices. In *IWSP: International Workshop on Security Protocols, LNCS*, 1997.

[3] Benny Bing. *Broadband Wireless Access*. Kluwer Academic Publishers, 2002.

[4] Dan Boneh, Richard A. DeMillo, and Richard J. Lipton. On the Importance of Checking Cryptographic Protocols for Faults. In *EUROCRYPT'97*, volume 1233, pages 37–51. Lecture Notes in Computer Science, Advances in Cryptology, 1997.

[5] Ran Canetti, Daniele Micciancio, and Omer Reingold. Perfectly One-Way Probabilistic Hash Functions. In *30th Annual ACM Symposium on Theory of Computing*, pages 131–140, 1998.

[6] CAST Inc. AES and SHA-1 Cryptoprocessor Cores. http://www.cast-inc.com.

[7] Suresh Chari, Charanjit Jutla, Josyula R. Rao, and Pankaj Rohatgi. A Cautionary Note Regarding Evaluation of AES Candidates on Smart-Cards. In *Second Advanced Encryption Standard (AES) Candidate Conference*, Rome, Italy, 1999.

[8] EAN International and the Uniform Code Council. http://www.ean-int.org.

[9] Howard Gobioff, Sean Smith, J. Doug Tygar, and Bennet Yee. Smart Cards in Hostile Environments. In *2nd USENIX Workshop on Elec. Commerce*, 1996.
[10] Oded Goldreich. *Foundations of Cryptography*. Cambridge University Press, 2001.
[11] Jeffrey Hoffstein, Jill Pipher, and Joseph H. Silverman. NTRU: A Ring-Based Public Key Cryptosystem. *Lecture Notes in Computer Science*, 1423:267–, 1998.
[12] Markus Jakobsson and Susanne Wetzel. Security Weaknesses in Bluetooth. *Lecture Notes in Computer Science*, 2020:176+, 2001.
[13] Ari Juels and Ravikanth Pappu. Squealing Euros: Privacy Protection in RFID-Enabled Banknotes. In *Financial Cryptography*, 2002. Submitted for publication.
[14] Burton S. Kaliski Jr and Matt J. B. Robshaw. Comments on Some New Attacks on Cryptographic Devices. RSA Laboratories' Bulletin No. 5, July 1997. http://www.rsasecurity.com/rsalabs/bulletins/.
[15] Paul Kocher, Joshua Jaffe, and Benjamin Jun. Differential Power Analysis. *Lecture Notes in Computer Science*, 1666:388–397, 1999.
[16] Paul C. Kocher. Cryptanalysis of Diffie-Hellman, RSA, DSS, and other Systems Using Timing Attacks. Technical report, Cryptography Research, Inc., 1995.
[17] Matthias Krause and Stefan Lucks. On the Minimal Hardware Complexity of Pseudorandom Function Generators. In *Theoretical Aspects of Computer Science*, volume 2010, pages 419–435. Lecture Notes in Computer Science, 2001.
[18] Michael Luby and Charles Rackoff. How to Construct Pseudorandom Permutations from Pseudorandom Functions. *SIAM Journal on Computing*, 17(2):373–386, April 1988.
[19] Alfred J. Menezes, Paul C. van Oorshot, and Scott A. Vanstone. *Handbook of Applied Cryptography*, chapter 1.9. CRC Press, 1996.
[20] Robert M. Metcalfe and David R. Boggs. Ethernet: Distributed Packet Switching for Local Computer Networks. *Communications of the ACM*, 19(5):395–404, July 1976.
[21] MIT. Auto-ID Center. http://www.autoidcenter.org.
[22] NTRU. GenuID. http://www.ntru.com/products/genuid.htm.
[23] RFID Journal. Gillette to Purchase 500 Million EPC Tags. http://www.rfidjournal.com, November 2002.
[24] RFID Journal. Michelin Embeds RFID Tags in Tires. http://www.rfidjournal.com, January 2003.
[25] Ronald L. Rivest. Chaffing and Winnowing: Confidentiality without Encryption. *CryptoBytes (RSA Laboratories)*, 4(1):12–17, Summer 1998.
[26] Sanjay E. Sarma. Towards the Five-Cent Tag. Technical Report MIT-AUTOID-WH-006, MIT Auto-ID Center, 2001.
[27] Sanjay E. Sarma, Stephen A. Weis, and Daniel W. Engels. RFID Systems and Security and Privacy Implications. In *Workshop on Cryptographic Hardware and Embedded Systems*, pages 454–470. Lecture Notes in Computer Science, 2002.
[28] Frank Stajano and Ross Anderson. The Resurrecting Duckling: Security Issues for Ad-hoc Wireless Networks. In *7th International Workshop on Security Protocols*, volume 1796, pages 172–194. Lecture Notes in Computer Science, 1999.
[29] TAMPER Lab. University of Cambridge Tamper and Monitoring Protection Engineering Research Lab. http://www.cl.cam.ac.uk/Research/Security/tamper.
[30] Uniform Code Council. Homepage. http://www.uc-council.org.
[31] Steve H. Weigart. Physical Security Devices for Computer Subsystems: A Survey of Attacks and Defences. In *Workshop on Cryptographic Hardware and Embedded Systems*, volume 1965, pages 302–317. Lecture Notes in Computer Science, 2000.
[32] David J. Wheeler and Robert M. Needham. TEA, a Tiny Encryption Algorithm. Technical report, Computer Laboratory, University of Cambridge, 1995.
[33] David J. Wheeler and Robert M. Needham. TEA Extensions. Technical report, Computer Laboratory, University of Cambridge, 1997.