

Strengthening EPC Tags Against Cloning

Ari Juels

RSA Laboratories
Bedford, MA 01730, USA
e-mail: ajuels@rsasecurity.com

12 October 2004

Abstract. One incarnation of RFID (Radio-Frequency IDentification) is a device known as an EPC (“Electronic Product Code”) tag. EPC tags are emerging as a successor technology to the printed barcode. They emit static codes which, like barcodes, serve to identify and track shipping containers and individual objects. Some industry segments are also coming to view RFID as an anti-counterfeiting tool. RFID tags enable the compilation of detailed object histories and pedigrees that can help combat counterfeiting of goods. While a potent tool for object identification, though, EPC tags are themselves weak authenticators. In particular, they are vulnerable to elementary cloning attacks.

In this paper, we present a simple technique that strengthens the resistance of EPC tags to cloning attacks. Our technique requires no modification to standard EPC tags. Rather, we show how PINs present in EPC tags to support access-control functions can be leveraged to provide a kind of crude challenge-response authentication. Our techniques can even strengthen tags against cloning attacks in environments where reading devices are untrusted.

Key words: authentication, challenge-response, counterfeiting, EPCglobal, RFID

1 Introduction

We propose simple techniques to help defend RFID tags against basic cloning attacks, i.e., against the duplication of legitimate tags.

The United States Department of Defense and several dominant retail corporations such as Wal-mart have mandated the use of RFID tags by their top suppliers beginning in 2005 [27]. In these deployments, a form of RFID tag known as the *EPC* (“Electronic Product Code”) tag will almost certainly predominate. EPC tags are an evolving standard under development by an organization called EPCglobal [3]. EPC tags are widely viewed as a form of successor to the printed barcode. Indeed, EPCglobal is a joint venture between the EAN and UCC, the organizations that oversee barcode standards respectively in the U.S. and Europe. An EPC is the form of identifier that an individual RFID tag emits as prescribed by the EPCglobal standard. An EPC includes not just the information contained in a conventional printed barcode, namely the manufacturer and type of a particular product, but also a unique identifier or serial number.

The attractiveness of EPC tags (and RFID tags more generally) over barcodes is twofold. First, EPC tags can transmit information over short distances to reading devices automatically via radio frequency. Unlike a barcode scanner, an RFID

reader does not require line-of-sight or physical contact to scan an EPC tag; this feature reduces the cumbersome need for manual intervention in the scanning process. A second benefit of EPC tags is their unique identifiers. A barcode typically specifies the type of product it is printed on, e.g., a bar of Valrhona chocolate. An EPC tag assigns a unique serial number to an individual item, i.e., it would indicate not just that an object is bar of Valrhona chocolate, but also *which* bar it is among the millions that have been manufactured. The unique identifier associated with an object can serve as a pointer to a database entry containing a detailed history of the object. Thanks to the features of automated scanning and unique identification, RFID systems promise fine-grained tracking of inventory on an unprecedented scale.

In initial deployments, EPC tags will serve primarily to identify pallets or crates of items within the industrial segments of supply chains, e.g., in warehouse-to-warehouse shipping. Although some tagging of individual retail items is already taking place in, e.g., garments at Marks and Spencer [2], this practice is likely to see restriction to high-value items for some time to come.

While RFID is a decades-old concept, it is becoming viable now as a ubiquitous technology thanks to dropping cost. Optimistic estimates suggest that an individual EPC tag may cost as little as five cents in the next several years [20]. The flip-side of this low cost is low functionality. Most importantly for our purposes here, a basic EPC tag is incapable of performing cryptographic operations like encryption or authentication.

EPC tags of the next generation, so-called Class 1 Generation 2 tags, are likely to see the most widespread industrial use for some time to come. These tags are therefore the main focus of our paper. When we speak of EPC tags, it is the Class 1 Generation 2 variety we mean to speak of. At the date of writing of this paper, the EPCglobal standard for such tags is still in progress. Although we shall in general write of EPC tag features in the present tense, some of our description is speculative or based on extrapolation from earlier, published EPCglobal standards, such as the current Class 1 tag specification [1].

1.1 EPC-tag security

EPC tags have a small set of sensitive functions, namely:

1. “Write”: Tags contain a small amount of writable memory (on the order of tens of bits).
2. “Sleep”: Tags are able to enter a special state in which they do not emit their identifiers; a complementary “wake” command triggers emergence from this state.
3. “Kill”: Tags contain a self-destruct mechanism, i.e., one that renders them permanently non-functioning.

EPC tags enforce a form of PIN-controlled access to these functions. In order to execute a sensitive function in an EPC tag, a reader must provide a correct PIN

that is specific to the tag. The EPCglobal standard is likely to specify PINs for sensitive EPC tag operations of roughly 32 bits in length.

Apart from these access-control PINs, EPC tags have essentially no data-security features. In particular, they emit their EPCs promiscuously, i.e., to *any* querying reader. EPC tags have no capability for authenticating readers scanning their EPCs. By means of their PINs, tags can only authenticate readers attempting to execute sensitive operations like writing.

Similarly, there is no prescribed mechanism for readers to authenticate the validity of the tags they scan. The result is that EPC tags are vulnerable to elementary cloning. An attacker can learn a tag's full EPC simply by scanning it or by gaining access to the appropriate tag database. Alternatively, if the unique identifiers in some manufacturer's EPC tags are not random, e.g., if they are sequential, then an attacker that sees an EPC on one item can guess or fabricate another valid EPC. In brief, "identity theft" of EPC tags is a simple matter. We refer to such attacks as *skimming*.

This situation raises the question: Once an attacker has possession of a valid EPC, how easy is it to create a counterfeit tag bearing that EPC? It will be difficult to offer a precisely calibrated answer to this question until RFID technology and product offerings reach a more advanced stage of maturity. It is quite conceivable, for example, that manufacturers will offer EPC tags that are fully field programmable so as to offer tight manufacturer control over tag configuration. Field programmability would be a ready-made solution for tag counterfeiting. Even in the absence of such a tool, it seems likely that EPC tags, being simple devices, will be easily forgeable. Certainly, simulation of an EPC tag in a larger device, e.g., an RF-enabled PDA, will be a simple exercise. In some cases such simulation may be sufficient to fulfill the aim of an attacker: A counterfeiter that wishes to forge an EPC tag on a crate or pallet, for example, can probably use a fairly large device to do so without detection.

While EPC tags are therefore likely to be poor authenticators, vulnerable to relatively straightforward counterfeiting, some industries are in fact contemplating their use precisely to combat counterfeiting of consumer goods and other items. Media reports have suggested such a plan by the European Central Bank to combat counterfeiting of Euro banknotes [4, 12, 14, 24]. More recently, the U.S. FDA (Food and Drug Administration) has issued a report that endorses RFID as a tool to combat the counterfeiting of pharmaceuticals [7].

To be fair, even with their weak resistance to forgery, EPC tags can play a role in combatting counterfeiting. The FDA report emphasizes that by aiding in the compilation and analysis of item pedigrees, EPC tags can help furnish a clearer picture of the structure of a supply chain and of potential sources of counterfeit goods. Nonetheless, it is easy to envision scenarios in which the vulnerability of EPC tags to skimming can create or enhance opportunities for counterfeiting. Here are a couple of hypothetical examples:

Example 1. EXCON Corp., a shipping company, wishes to steal prescription medications that it has been entrusted with transporting. These medications are trans-

ported in tamperproof cases with attached RFID tags. Rather than attempting to defeat the tamperproofing of the cases, EXCON creates bogus medications and cases, and forges the associated EPC tags. It swaps in the bogus cases while it has custody of the real ones.¹

Example 2. Some manufacturers are beginning to furnish RFID readers for inclusion in mobile phones [16]. One can imagine in the future that a forger of luxury handbags may wish to create RFID tags for their wares that appear to be legitimate when scanned by consumers. A forger can easily copy an RFID tag by skimming it. Indeed, so as to avoid using a single identifier that might be targeted or traced by authorities, the forger can skim the EPCs from tags on the legitimate handbags of passersby and use them to forge new EPC tags.

Organization

In section 2, we briefly review the literature related to RFID authentication. We propose our EPC-tag authentication scheme in section 3. In section 4, we discuss strong cloning attacks, and describe some possible countermeasures, both within and beyond the scope of the EPCglobal standard. In section 5, we show how our ideas may be extended to strengthen RFID architectures against potentially untrusted reading devices. We conclude in section 6.

2 Previous Work

Some commercially available RFID tags can perform cryptographic challenge-response protocols. SpeedpassTM is an example of one in common use, with over five million users [22]. Tags that perform cryptographic challenge-response protocols are resistant to skimming. They cost significantly more than EPC tags, though, and are therefore viable only for niche applications like consumer payments.

Weis et al. have proposed privacy-protecting authentication protocols for tags; their proposals require cryptographic hash functions, however, are thus unsuitable for EPC tags [19, 25]. Juels [10] has proposed a security model specific to RFID environments that would permit a form of dynamic challenge-response protocol without the use of cryptography. Apart from the fact that this proposal would require a new RFID-tag design, it also would require greater tag resources than the current generation of EPC tags. Another proposal of Juels called “yoking” allows a pair of tags with minimal resources to construct a one-time proof that they have been read simultaneously [11]. The techniques underlying “yoking” could be used to enable tags to authenticate themselves to readers, but aim to secure only one-time use, rather than repeated use.

¹ This *modus operandi* is not an uncommon one. It is in fact one way in which corrupt officials have purportedly altered vote tallies in elections. Rather than tampering with ballot-boxes, they have created fake ballot-boxes and ballots offsite, applied countefeited seals, and substituted these for legitimate ballot boxes in transit from polling stations. See, e.g., [21].

There is a considerable body of research on the design of lightweight public-key encryption and digital-signing algorithms – largely intended for use in smart cards and similarly small computational devices. These algorithms include identification or digital-signature schemes such as the classic Guillou-Quisquater algorithm [8] and also newer algorithms like the NTRU cryptosystem [9]. Even the most lightweight of these many schemes, e.g., [23], is likely to be well beyond the capabilities of small RFID tags for quite some time to come. A related area is security for sensor networks. While lightweight, these devices are still more capable than RFID tags, as they typically include their own power sources. Although recent work has led to more compact implementations of symmetric-key primitives like AES for RFID tags citeFeldhoferEtal:2004, these are still well beyond the reach of basic EPC tags.

One key idea in our proposals in this paper is the presentation of spurious PIN to RFID tags as a means of testing their authenticity. This is similar in flavor to the notion of “winnowing” introduced by Rivest [18]. Rivest’s idea is to leverage an authentication protocol to achieve data privacy by inserting false packets into a data stream: Only by picking out the correct ones can a receiver extract the transmitted message. Similarly, we show here how to leverage reader-to-tag authentication so as to achieve tag-to-reader authentication.

To date, the majority of the scientific literature, e.g., [6, 12, 13, 25, 26] and media coverage, e.g., [5, 15, 27] on RFID security has focused on privacy-related aspects, rather than authentication.

3 Our Proposal

We propose a simple scheme to help defend standard EPC tags against skimming attacks. Our key observation involves the PINs that EPC tags use for sensitive function calls like writing. These PINs serve to authenticate a reader to a tag. As we show, when used the right way, the direction of use of these PINs can in fact be reversed; they can serve instead to *authenticate the tag to a trusted reader*.

There is no explicit functional support for tag-to-reader authentication. We show how to achieve it, though, using any repeatable EPC-tag command for which a PIN is used as an access-control mechanism. We refer to any such command generically as `Unlock`, the notion being that the command involves release of some sensitive tag capability. Thus `Unlock` might be a PIN-controlled read, write, or “sleep” command. The “kill” command or a one-time write command would not be workable for our purposes, as these are not repeatable operations.²

² While workable for our purposes, writing is a relatively unreliable tag operation that requires high power availability for the tag and carries the risk of leaving the tag in a corrupt state in case of protocol failure. The “sleep” command can introduce key-management problems around the PIN using to “wake” tags. Thus the ideal `Unlock` operation for our protocols in this paper would be either a PIN-controlled read operation or else an atomic operation that authenticates the reader to the tag in preparation for reading or writing. Class 1 Generation 2 tags may include this last operation.

In a system with N tags, let the integer i (with $1 \leq i \leq N$) denote the unique index of an EPC tag. Let us denote the EPC identifier, i.e., the unique RFID readable string for tag i , by T_i . Let P_i be a k -bit PIN for tag i ; we assume that the PIN for a tag is generated uniformly at random upon tag initialization. Let $r \leftarrow \text{Unlock}(P)$ denote the execution of `Unlock` using PIN P . We assume that an EPC tag replies with $A = \text{"ack"}$ if the `Unlock` command is successful; otherwise it returns $A = \text{"nil"}$ value – either an explicit indication of failure, or else no response at all.

We present an elementary protocol `BasicTagAuth` in Figure 1. In this and following figures, “ $\mathcal{A} \rightarrow \mathcal{B}$ ” indicates a data flow from entity \mathcal{A} to entity \mathcal{B} , while “ $\mathcal{A} :$ ” indicates an operation performed locally by \mathcal{A} . In the protocol `BasicTagAuth`, a trusted RFID device \mathcal{R} attempts to authenticate a (presumed) tag \mathcal{T} .

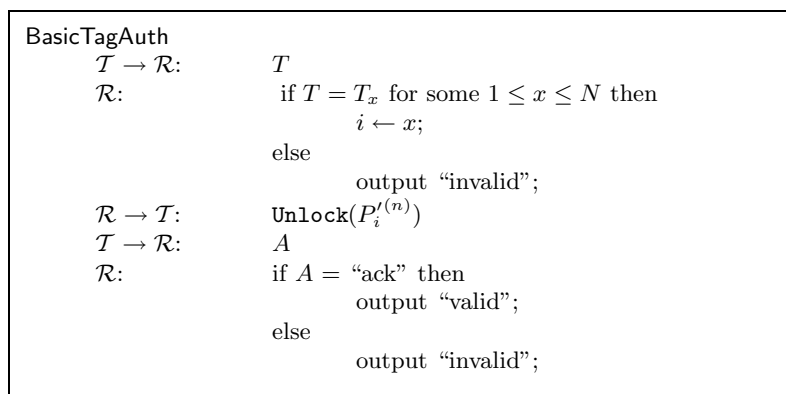


Fig. 1. The `BasicTagAuth` protocol

A tag that does not carry a valid identifier T_x (for some x) will not achieve successful authentication in this protocol. Thus an adversary cannot successfully clone a tag without knowledge of a valid T_x obtained, for example, via skimming.

On the other hand, consider a clone \tilde{i} that is EPC-compliant but created via a skimming attack, and therefore without knowledge of P_i . Such an EPC-compliant clone \tilde{i} might be easily created, for instance, through configuration of a field-programmable EPC tag. For \tilde{i} to cause a “valid” output, its creator would need to guess P_i correctly. The probability of successful cloning here would therefore be 2^{-k} .

Remarks: RFID protocols are subject to frequent failures due to power loss in the tag. Thus `BasicTagAuth` has the potential to produce a false negative, i.e., to lead to the conclusion that a tag is a clone when it is in fact valid. There are a couple of ways of mitigating this problem. One is to test the response rate of the tag to other commands in order to ascertain whether it is responding reliably. Indeed, in many environments, e.g, warehouses, continuous reading is likely to be common,

so that some gauge of a tag’s reliability may already exist. Another possibility is simply to repeat the authentication protocol. Of course, in the rare instance where a tag behaves persistently like a clone, it may be subjected to scanning in a well controlled environment.

When performing active skimming against a tag i , an adversary can test multiple possible values of P_i . Some EPC tags currently defend against PIN-guessing by temporarily disabling a tag when multiple incorrect PINs are presented [17]. (PINs in Class 1 Generation 1 EPC tags are only 8 bits in length.) Once PIN lengths grow to 32 bits or more, as anticipated in the Class 1 Generation 2 standard, harvesting of tag PINs via active skimming will be largely impractical.

3.1 Non-compliant clones

The `BasicTagAuth` protocol we have just proposed has a basic vulnerability: If the cloned tag \tilde{i} is *not* EPC-compliant, then it can spoof the reader. It suffices for \tilde{i} simply to accept any PIN P , in which case the protocol will always output “valid.”

To detect non-compliant clones of this kind, we propose the introduction of *spurious PINs* into our protocol. In this approach, the reading device tests the response of a tag to some PINs that are not valid. If the tag accepts the `Unlock` operation in response to any of these PINs, then the reader can identify it as counterfeit. We include these ideas in a protocol that we call `TagAuth`.

We describe the protocol `TagAuth` in Figure 2. Here the value q is a security parameter that specifies the number of spurious PINs to be generated. The function `GeneratePINSet` generates a set of $q - 1$ spurious PINs selected uniformly at random. Among these is randomly admixed the one correct PIN P_i in a random position j' , which is also output by the function `GeneratePINSet`. We detail the exact operation of `GeneratePINSet` below.

The best strategy for a cloned tag to defeat the protocol `TagAuth` is to guess the correct PIN-trial j' uniformly at random. The probability of successful attack, i.e., authentication in this case is clearly just $1/q$ (assuming truly random selection of spurious PINs). We sketch an attack model for this protocol in appendix A.

`TagAuth` is naturally time-consuming for large values of q . To prevent more than casual introduction of counterfeit tags into an RFID system, however, it would suffice to detect such tags with significant but not overwhelming probability. For this purpose, even $q = 2$, i.e., a single spurious PIN, would generally be adequate.

Creating the PIN set: There are two ways that the function `GeneratePINSet` can generate spurious PINs. The first method is random selection. In particular, the set $\{P_i^{(j)}\}$ may be selected uniformly at random without duplication from $\{0, 1\}^k$. The true PIN P_i should then replace a random element $P_i^{(j')}$ for $j' \in_U \{1, 2, \dots, n\}$.

The PIN set $\{P_i^{(j)}\}$ must remain static over all invocations of `TagAuth`. This is an important feature: If the set of spurious PINs were to change from session to session, then an adversary could determine P_i by computing the intersection between

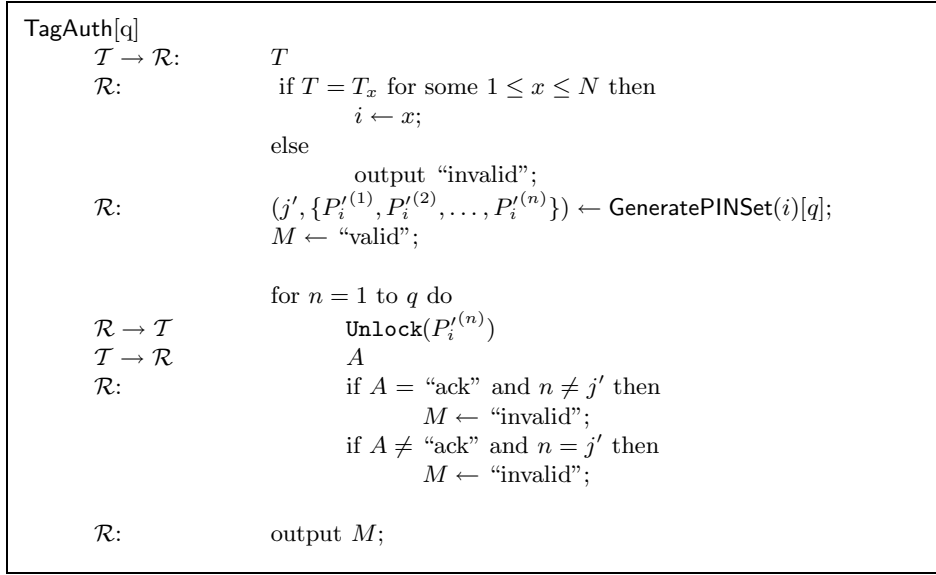


Fig. 2. The TagAuth protocol

or among PIN sets. Thus, if already invoked for tag i , the function `GeneratePINSet` should simply output the existing set $\{P_i^{(n)}\}_{n=1}^q$.

As a second approach to spurious-PIN generation, it is possible to avoid the need for storing the set $\{P_i^{(n)}\}_{n=1}^q$ by generating it pseudorandomly. To use informal notation here, let f denote a one-way hash function, and x denote a master secret-key held by the reader \mathcal{R} . For a positive integer z and non-empty set $Q = \{q_0, q_1, \dots, q_{|Q|-1}\}$, let $Q_{[z]}$ denote the element $q_{z \bmod |Q|}$. `GeneratePINSet` may be constructed as follows:

```

GeneratePINSet( $i$ )[ $q$ ]
  for  $n = 1$  to  $q$  do
     $Q \leftarrow \{0, 1\}^k - \{P_i \cup P_i^{(1)} \cup \dots \cup P_i^{(n-1)}\}$ ;
     $P_i^{(n)} \leftarrow Q_{[f(x, i)]}$ ;
   $j' \in_U \{1, 2, \dots, q\}$ ;
   $P_i^{(j')} \leftarrow P_i$ ;
  output( $j', \{P_i^{(n)}\}_{n=1}^q$ );

```

4 Stronger Attacks and Defenses

Skimming is perhaps the easiest and most practical cloning attack and therefore the most important to defend against. Stronger attacks are possible, however, that would defeat the protocol `TagAuth`. We now enumerate some of these.

1. **Database breaches:** An adversary capable of breaching the database containing the PINs of tags will of course be able to clone a tag perfectly. We note that compromise of a valid reader can potentially have the effect of giving an adversary access to this database.
2. **Reverse engineering:** EPC tags are simple devices that provide no real tamper resistance. A moderately sophisticated adversary will therefore be able to reverse-engineer a captured tag and extract its PIN. Such an adversary can of course clone the tag perfectly.
3. **Eavesdropping:** An adversary capable of full eavesdropping on the communications between the reader and tag can harvest the correct PIN P_i for a tag i . There are some important technical nuances to consider, though. The signal strength of the reader-to-tag channel is considerably stronger than that of the tag-to-reader channel. The reader is an active device, while RFID tags are passive devices that receive their transmission power from the reader. An adversary is therefore much more likely to be able to eavesdrop on the reader-to-tag channel. Such eavesdropping may take place at a distance of hundreds of meters, while eavesdropping on tag emissions is feasible at the very most from at most some tens of meters away (using off-the-shelf readers, at least).

An eavesdropping attack on the reader-to-tag channel alone in principle reveals no information when the same set of spurious PINs is employed in every reader-to-tag session as we propose. If, however, tag behavior differs between the cases where correct and incorrect PINs are presented, then some reader compensation may be necessary to conceal leakage of PIN information to an eavesdropper. For example, it is conceivable that tags may require resingulation when incorrect PINs are presented (on the assumption that signal corruption is at fault). In this case, to avoid betraying a correct PIN to an eavesdropper, a reader would have to resingulate tags even after presentation of a correct PIN.

It should also be noted that in some RFID-system proposals, the tag generates random one-time pads used to conceal communications on the reader-to-tag channel. In this case, it may be feasible for an adversary to gain useful information only by eavesdropping on the tag-to-reader channel.

An adversary that is capable of full eavesdropping, i.e., monitoring of communications in both directions, will, of course, learn P_i immediately.

Our proposed authentication protocol can be strengthened to defend to some extent against all three of the attacks described above.

EPC tags will very likely include mechanisms to enable industry users to program the write PINs of EPC tags in the field. In this case, we propose periodic updates to the PIN of a valid tag to help reduce the severity of the attacks described above. An adversary capable of full eavesdropping on only a periodic basis may not be able to learn the most up-to-date PIN employed by a given system. Likewise, an adversary that reverse-engineers and clones a tag multiple times will not be able to seed a system with multiple clones that remain up-to-date: Only one clone at a time will remain valid if readers update tag PINs.

There is a limitation on the frequency with which PIN changes can be viably performed. This is due to the factors mentioned above in section ??, namely the unreliability of the write operation in RFID systems and the limited lifetime of EEPROM. Nonetheless, updates need be performed with only modest frequency to offer considerably strengthened security.

4.1 Privileged reading

EPC tags do not support privileged reading, i.e., commands for the operation of reading are not PIN-controlled. Privileged reading would involve only a small change in tag resources, though, and would permit strengthening of our proposed authentication protocol. In particular, suppose that a tag i has memory F for which both reading and writing are privileged operations (enabled with either the same or separate PINs). There are then two ways to strengthen our proposed protocol:

- *Stored-value testing*: Suppose that F in tag i is programmed with a random, secret value S . Then a reader can test the validity of tag i by checking the result of a privileged read operation. In particular, we might employ a modified protocol $\text{TagAuth}^*[q]$ in which the reader tests the tag response to P_i by performing a read operation on F . In this case, the protocol would accept only if the tag accepted PIN P_i exclusively *and* transmitted the stored value S at that time.

Suppose that S is l bits in length. Consider, then, an attack via skimming. For a cloned device to spoof $\text{TagAuth}^*[q]$, it would have to guess both j' and S successfully. The probability of doing so successfully is of course equal to $2^{-l}/q$.

- *Partial read-write*: Suppose that F in fact comprises a collection of memory cells $\{F_1, F_2, \dots, F_m\}$. In this case, $\text{TagAuth}^*[q]$ might examine only a portion of F , e.g., a single cell F_b ($1 \leq b \leq m$) in a given session. Similarly, if the PIN P_i is stored in multiple, independent cells, an RFID system that updates this PIN periodically might perform only a partial update at any given time. The advantage of performing partial operations of this kind is that it would defend against periodic adversarial eavesdropping on the tag-to-reader channel (as discussed above). Without sufficiently frequent eavesdropping, an adversary would be unable to compromise a tag's sensitive values in their entirety. This idea is essentially just a rather stripped-down version of the “minimalist cryptography” techniques in [?].

One form of attack that our techniques do not defend against is denial-of-service (DoS). An adversary can create a device that creates the appearance of one or many cloned RFID tags. In contrast to Internet-based DoS attacks, a DoS attack against the perimeter of RFID system would necessarily be physically circumscribed, and therefore generally easier to contain and investigate.

5 Defending Against Cloning in the Face of Reader Compromise

For the sake of simplicity and focus on the issue of tag authentication, our working assumption thusfar has been that the trusted entity to which a tag authenticates is

identical with the reader scanning the tag. In other words, we have assumed that the reader \mathcal{R} is entrusted *a priori* with the corresponding PIN P_i for any given tag i .

An interesting architectural possibility arises if we instead assume that the trusted entity \mathcal{V} with knowledge of P_i is *not* identical with the reading device \mathcal{R} participating in the protocol. \mathcal{V} might instead be a secure, centralized server that interacts with readers. We may then view the authenticating entity in our protocol as a combination of \mathcal{R} and a presumed EPC tag \mathcal{T} : The reader \mathcal{R} tries to prove to \mathcal{V} that it is scanning a particular tag i . This view yields a new protocol variant with entities \mathcal{V} , \mathcal{R} , and presumed tag \mathcal{T} . Our modified protocol is given in Figure 3.

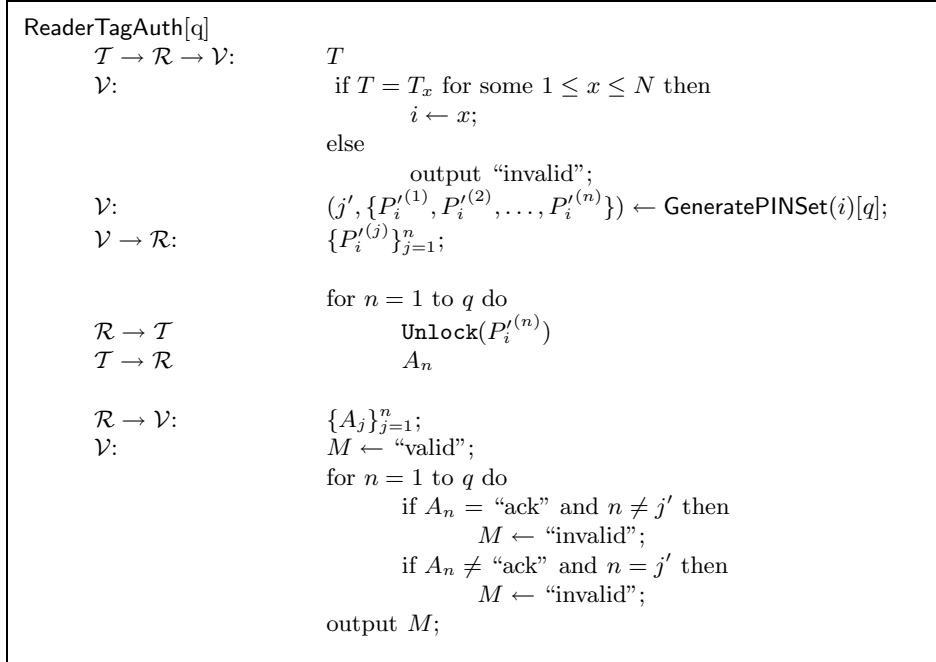


Fig. 3. The ReaderTagAuth protocol

In this new protocol, the reader may be viewed simply as an untrusted communications medium by which the tag communicates with \mathcal{V} . *Without access to tag i* , the reader \mathcal{R} itself does not learn which of the presented PINs is the correct one. Hence the security properties of this scheme with respect to an attacker that has compromised \mathcal{R} and knows T_i alone are similar to those for **TagAuth**[q] with respect to an attacker that only knows T_i . In brief, with knowledge of T_i alone, the best an attacker can do in creating a clone is to guess the correct PIN uniformly at random from a set of q PINs. Thus, the attacker can only forge a tag capable of defeating the authentication protocol with probability $1/q$. On the other hand, once it scans tag i , of course, the reader \mathcal{R} (and attacker that has compromised the reader) does learn P_i .

The protocol variant `ReaderTagAuth` is particularly interesting because readers represent a salient point of compromise in RFID systems. In a naïve deployment, a reader might be capable of accessing a PIN P_i (from a database, for instance) for any tag identifier T_i . In such a system, compromise of a single reader would result in *en bloc* compromise of tag PINs. An attacker with access to the compromised reader would be able to learn the PIN P_i associated with any tag identifier T_i and then clone the tag perfectly.

This situation is particularly problematic because it will be increasingly the case in RFID deployments that reading devices are ubiquitous peripherals. They will be placed in warehouses, storage rooms, trucks, and retail environments. Use of our proposed protocol `ReaderTagAuth[q]` can help address the problems associated with reader compromise.

In fact, to mitigate the effect of network failures, it seems likely that in many RFID architectures, readers or associated devices may store large numbers of tag PINs locally. The protocol `ReaderTagAuth[q]` can offer stronger security even in this environment. Rather than storing PINs locally, readers can instead store the PIN sets generated by `GeneratePINSet`. Compromise of the reader would no longer then lead to direct compromise of true tag PINs and the ability to clone skimmed tags. A drawback to this approach is that to execute sensitive tag operations, a reader would have to try multiple PINs, i.e., cycle through the stored PIN set for a tag. With $q = 2$, i.e., a single spurious PIN per tag, however, we believe an RFID system could offer reasonably strong defense against general tag cloning, with minimal impact on performance.

6 Conclusion

We have proposed a simple, practical authentication technique for combatting skimming attacks against EPC tags. Our scheme involves a kind of role-reversal for the PINs in EPC tags. While these PINs are meant by design to serve for reader-to-tag authentication, we show how they may in fact provide tag-to-reader authentication and thereby help prevent skimming attacks. As we anticipate that many industry uses of EPC tags will come to rely either implicitly or explicitly on their resistance to counterfeiting, we believe that our proposal will prove valuable in real-world systems.

The secure assignment and distribution of PINs in RFID systems is a major, open research and deployment question that we do not address in this paper. Without careful deployment, compromise of a single reader in an RFID system may lead to compromise of a large quantity of sensitive tag data. We have also shown, however, how our ideas may offer some help in this area by mitigating the risk of tag cloning in systems where RFID reading devices are subject to compromise.

Acknowledgments:

Thanks to Dan Bailey, Burt Kaliski, and Steve Weis for their comments on and refinements to this paper and its ideas.

References

1. Auto-ID Center. 860 MHz - 930 MHz class 1 radio frequency identification tag radio frequency and logical communication interference specification: Candidate recommendation, version 1.0.1. Technical Report MIT-AUTOID-TR-007, EPCglobal Inc., 14 November 2002. Available at http://www.epcglobalinc.org/standards_technology/Secure/v1.0/UHF-class1.pdf.
2. J. Collins. Marks & spencer expands RFID retail trial. *RFID Journal*, 10 February 2004. Available at <http://www.rfidjournal.com/article/articleview/791/1/1/>.
3. EPCglobal Web site. www.epcglobalinc.org, 2004.
4. Security technology: Where's the smart money? *The Economist*, pages 69–70, 9 February 2002.
5. RFID: eWeek.com special report, 2004. Available at <http://www.eweek.com/category2/0,1738,1568291,00.asp>.
6. K. P. Fishkin, S. Roy, and B. Jiang. Some methods for privacy in RFID communication. In *1st European Workshop on Security in Ad-Hoc and Sensor Networks (ESAS 2004)*, 2004. To appear.
7. United States Food and Drug Administration. Combatting counterfeit drugs: A report of the Food and Drug Administration, 18 February 2004. Available at http://www.fda.gov/oc/initiatives/counterfeit/report02_04.html.
8. L. Guillou and J. Quisquater. A practical zero-knowledge protocol fitted to security microprocessor minimizing both transmission and memory. In C. G. Günther, editor, *EUROCRYPT '88*, pages 123–128. Springer-Verlag, 1988. LNCS no. 330.
9. J. Hoffstein, J. Pipher, and J.H. Silverman. NTRU: A ring based public key cryptosystem. In *ANTS III*, pages 267–288. Springer-Verlag, 1998. LNCS no. 1423.
10. A. Juels. Minimalist cryptography for low-cost RFID tags. In C. Blundo, editor, *Security in Communication Networks (SCN 04)*. Springer-Verlag, 2004. To appear.
11. A. Juels. ‘Yoking-proofs’ for RFID tags. In *PerCom Workshops 2004*, pages 138–143. IEEE Computer Society, 2004.
12. A. Juels and R. Pappu. Squealing Euros: Privacy protection in RFID-enabled banknotes. In R. Wright, editor, *Financial Cryptography '03*, pages 103–121. Springer-Verlag, 2003. LNCS no. 2742.
13. A. Juels, R.L. Rivest, and M. Szydlo. The blocker tag: Selective blocking of RFID tags for consumer privacy. In V. Atluri, editor, *8th ACM Conference on Computer and Communications Security*, pages 103–111. ACM Press, 2003.
14. J. Mara. Euro scheme makes money talk. *Wired News*, 9 July 2003. Available at <http://www.wired.com/news/print/0,1294,59565,00.html>.
15. D. McCullagh. RFID tags: Big Brother in small packages. *CNet*, 13 January 2003. Available at <http://news.com.com/2010-1069-980325.html>.
16. Nokia unveils RFID phone reader. *RFID Journal*, 17 March 2004. Available at <http://www.rfidjournal.com/article/view/834>.
17. RFID, privacy, and corporate data. *RFID Journal*, 2 June 2003. Feature article. Available at www.rfidjournal.com on subscription basis.
18. R. L. Rivest. Chaffing and winnowing: Confidentiality without encryption. *CryptoBytes*, 4(1):12 – 17, Summer 1998.
19. S. E. Sarma, S. A. Weis, and D.W. Engels. Radio-frequency-identification security risks and challenges. rsa laboratories. *CryptoBytes*, 6(1), 2003.
20. S.E. Sarma. Towards the five-cent tag. Technical Report MIT-AUTOID-WH-006, MIT Auto ID Center, 2001. Available from <http://www.epcglobalinc.org>.

21. M.I. Shamos. Paper v. electronic voting records - an assessment, 2004. Paper written to accompany panel presentation at Computers, Freedom, and Privacy Conference '04. Available at <http://euro.ecom.cmu.edu/people/faculty/mshamos/paper.htm>.
22. Stop & Shop supermarket company to test ExxonMobil Speed-pass. *Texas Instruments RFID eNews*, 10, July 2002. Available at <http://www.ti.com/tiris/docs/news/eNews/eNewsVol10.pdf>.
23. J. Stern and J. Stern. Cryptanalysis of the OTM signature scheme from FC'02. In R. Wright, editor, *Financial Cryptography '03*. Springer-Verlag, 2003. To appear.
24. C.P. Wallace. The color of money. *Time Europe*, 158(11). 10 September 2001.
25. 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.
26. S.A. Weis. Radio-frequency identification security and privacy. Master's thesis, M.I.T. June 2003.
27. Wal-Mart, DoD Forcing RFID. *Wired News*, 3 November 2003. Available at <http://www.wired.com/news/business/0,1367,61059,00.html>.

A Attack Model

As we explain in the body of the paper, our protocol aim is to defend against the cloning of tags via skimming. Here we specify our attack model for the trusted reader case of section 3 in a little more detail. (The model for the case of untrusted readers is quite similar.) We summarize this attack model as an informally defined experiment **TagClone** involving an adversarial algorithm that we denote by \mathcal{A} interacting with a tag authentication protocol **TA**. We assume that **TA** outputs either “valid” or “invalid” at the end of an authentication session:

TagClone(N, q, k)

1. PINs $\{P_i\}_{i=1}^N$ are selected uniformly at random from $\{0, 1\}^k$.
2. \mathcal{A} selects tag identifiers (T_1, \dots, T_N) . (This step models the ability of the adversary to skim tags and learn their identifiers. In fact, the ability of the adversary to select identifiers is unrealistic, but results in a stronger attack model.)
3. \mathcal{A} may simulate any tag in an invocation of **TA**. The number of invocations made by \mathcal{A} is arbitrary, and invocations may be concurrent. All protocol sessions are terminated at the end of this step.
4. \mathcal{A} invokes **TA**. The output of this last invocation is $M \in \{\text{“valid”}, \text{“invalid”}\}$. (If the protocol is terminated prematurely, then $M = \text{“invalid”}$.)
5. The output of the experiment is ‘0’ if $M = \text{“invalid”}$ and ‘1’ otherwise.

We regard the success of the adversary as the probability of the adversary causing the experiment to output a ‘1’.

As noted in the body of the paper, for our authentication protocol **TagAuth**, the probability of success of \mathcal{A} is clearly just $1/q$ if spurious PINs are selected at random in **GeneratePINSet**. It is negligibly larger if spurious PINs are instead generated using an appropriate one-way function f .