Strengthening EPC Tags Against Cloning

Ari Juels

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

16 March 2005

Abstract. The EPC (Electronic Product Code) tag is a form of RFID (Radio-Frequency IDentification) device that is emerging as a successor to the printed barcode. Like barcodes, EPC tags emit static codes that serve to identify and track shipping containers and individual objects. EPC tags, though, have a powerful benefit: they communicate in an automated, wireless manner.

Some commercial segments, like the pharmaceutical industry, are coming to view EPC tags as an anti-counterfeiting tool. EPC tags are a potent mechanism for object identification, and can facilitate the compilation of detailed object histories and pedigrees. They are poor authenticators, though. EPC tags are vulnerable to elementary cloning and counterfeiting attacks.

In this paper, we present techniques that strengthen the resistance of EPC tags to elementary cloning attacks. Our proposals are compliant with EPCglobal Class-1 Generation-2 UHF tags, which are likely to predominate in supply chains. We show how to leverage PIN-based access-control and privacy enhancement mechanisms in EPC tags to achieve what may be viewed as crude challenge-response authentication. Our techniques can even strengthen EPC tags against cloning in environments with untrusted reading devices.

Key words: authentication, cloning, counterfeiting, EPC, PIN compromise, RFID

1 Introduction

In this paper, we propose techniques to help protect against RFID-tag cloning. Our focus is a type of inexpensive RFID (Radio-Frequency Identification) device known as an *EPC* (Electronic Product Code) tag. EPC tags will soon see very broad use in supply chains around the world for the purposes of identifying and tracking goods.

As supply chains expand and automation becomes commonplace, we believe that users will come to rely implicitly on RFID tags to authenticate goods. Protecting EPC tags against cloning, however, is challenging, as they possess no explicit authentication functionality. EPC tags do possess features geared toward privacy protection and access control. We show how to leverage these in unintended ways to construct rudimentary authentication protocols. Viewed another way, we demonstrate the repurposing of reader-to-tag authentication protocols to construct tag-to-reader authentication protocols.

Our proposed schemes are compliant with the EPC^{TM} Class-1 Generation-2 UHF-RFID standard [4], which the major RFID standards body known as EPCglobal has recently ratified [24]. This standard is likely to become internationally dominant. For brevity, we refer to it henceforth as the EPCglobal standard.

As we show, our techniques for tag authentication can also support a security goal that is only tangentially related, namely the prevention of *en bloc* theft of tag PINs by compromised reading devices. Toward this end, we propose a scheme that we call *fulfillment-conditional PIN distribution (FCPD).*

1.1 EPC in a nutshell

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 [34]. In these deployments, EPC tags will almost certainly predominate. EPC tags are an evolving standard under development by an organization called EPCglobal [3]. The RFID community views the EPC tag as a successor to the printed barcode. Indeed, EPCglobal is a joint venture between the UCC and EAN, 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. See [4] for particulars.

The attractiveness of EPC tags (and RFID tags more generally) over barcodes is twofold. First, EPC tags can transmit information over short distances to RFID readers 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 in large quantities, individual EPC tags may cost as little as five cents in the next several years [26]. EPC tags do not carry any on-board source of power, a feature that helps in cost reduction. They are *passive*, which is to say that they receive their power during interrogation by a reader.

The flip-side of low cost in EPC tags is low functionality. A basic EPC tag is incapable of performing cryptographic operations such as encryption or authentication, unlike more expensive RFID or RF devices.

Throughout this paper, we make a distinction between EPC tags and EPCs. An EPC tag is a physical RFID device, while an EPC is the digital information belonging to and generally contained in a particular tag. The starting point of our investigation of EPC-tag cloning is an elementary observation: *An EPC is a just a piece of data, and thus separable from an EPC tag.*

This version of the paper supersedes an earlier one dated 12 October 2004, which included ultimately incorrect predictions regarding features of the EPCglobal standard.

1.2 The problem of EPC-tag cloning

EPC tags possess no explicit anti-cloning features. That is, EPCglobal standards prescribe no mechanism for EPC readers to authenticate the validity of the tags they scan. An EPC tag emits its EPC promiscuously, i.e., to *any* querying reader. Readers accept the validity of the EPCs they scan at face value.

The result is that EPC tags are vulnerable to elementary cloning attacks. An attacker can learn a tag's essential data, its EPC, simply by scanning it or by gaining access to an appropriate tag database. If the unique identifiers in a manufacturer's EPCs 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 straightforward matter because EPCs are data objects that are easily separable from EPC tags.

We use the term *skimming* to denote the process of scanning an EPC for the purpose of cloning an EPC tag. A key question is this: Once an attacker has skimmed a valid EPC, how easy is it to create a counterfeit tag bearing that EPC? It is difficult to offer a precisely calibrated answer to this question until RFID technology and product offerings reach a more mature state. For example, manufacturers will probably come to offer EPC tags that are fully field programmable so as to enable tight manufacturer control over tag configuration. Field programmability would be a ready-made tool for tag counterfeiting.

Even in the absence of field-programmability, EPC tags, being simple devices, will be easily forgeable at the protocol level. Simulation of an EPC tag in a larger device, e.g., an RF-enabled PDA, is already a simple exercise. Researchers have recently demonstrated this by cloning proximity cards [17, 33], RFID devices used to control physical door access, and similar in functionality to EPC tags. EPC-tag simulation may be sufficient to fulfill the aims of many attackers: 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, and vulnerable to counterfeiting, some industries are 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 [5, 14, 18, 30].[1] 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 [9].

To be fair, even with weak resistance to cloning, EPC tags can play a role in combatting counterfeiting. The FDA report emphasizes that by aiding the compilation and analysis of item pedigrees, EPC tags can help furnish a clearer picture of supply chains and of potential sources of counterfeit goods. This benefit does not require protocol-layer authentication of tags. Nonetheless, it is easy to envision scenarios in which the vulnerability of EPC tags to cloning can facilitate counterfeiting. Here are a couple of hypothetical examples:

Example 1. EXCON Corp., a shipping company, is plotting to steal prescription medications that it has been entrusted with transporting. These medications are transported in tamper-

[1] This plan is seeming increasingly unlikely, particularly as original media reports specified a 2005 target date.

proof cases with attached RFID tags. Rather than going to the trouble of bypassing the tamper-proofing of the cases, EXCON creates bogus medications and cases, and clones the associated EPC tags. It swaps in the bogus cases while it has custody of the real ones.[2]

Example 2. Consumers will very likely make direct use of RFID tags at some point. Indeed, manufacturers are already incorporating both RFID readers and tags into mobile phones [21]. For example, the manufacturer of luxury handbags might encourage customers to register their purchases by scanning attached EPC tags. If these tags do not possess resistance to cloning, then a seller of counterfeit handbags can attach EPC tags carrying duplicated, valid EPCs. Indeed, this forger could skim EPCs from tags on legitimate handbags in shops or from passersby, and use these EPCs in bogus tags.

Organization

We present background material in section 2, both summarizing EPC-tag features relevant to our work and reviewing related literature. In section 3, we propose authentication techniques for *basic* EPC tags, i.e., those containing only the mandatory features of the EPCglobal standard. In section 4, we describe authentication techniques for *enhanced* EPC tags, i.e., those containing an optional access-control feature from the EPCglobal standard. We consider the problem of authentication via untrusted readers in section 5, and describe our fulfillment-conditional PIN distribution (FCPD) scheme. In section 6, we briefly characterize the strong cloning attacks that are capable of undermining our proposed techniques. We conclude in section 7 with avenues for further research.

2 Background

2.1 EPC tag capabilities

While EPC tags carry no explicit mechanisms for authentication, as we have explained, they do possess some basic data-security features. We briefly describe them here. We distinguish between two types of tags. A *basic* EPC tag is one that carries only the mandatory features of the EPCglobal standard. An *enhanced* EPC tag additionally includes an access-control function that is optional in the EPCglobal standard.

Basic EPC tags: Basic EPC tags have only one security feature that we exploit here, namely the privacy-enhancing kill command. When an EPC tag receives this command, it “self-destructs,” which is to say that it renders itself completely and permanently inoperable. To protect against accidental or malicious killing of tags, the kill command only takes effect when accompanied by a valid PIN. In the EPCglobal standard, the kill PIN is 32 bits in length.

Killing may be viewed as an access-control operation that succeeds only once. The EPCglobal standard, though, has a feature that can serve in principle to permit multiple presentations of a valid kill PIN. Recall that an EPC tag is passive, meaning that it receives

[2] 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 counterfeited seals, and substituted these for legitimate ballot boxes in transit from polling stations. See, e.g., [27].

its power from a reader. When it receives a kill command and valid PIN, *but has insufficient power to disable itself*, an EPC tag remains operational, and emits an error code. (When it receives a kill command with an invalid PIN, the tag effectively ignores the command.)

In consequence, given the ability to cause an EPC tag to register insufficient power for the kill operation, one can cause a tag effectively to emit a "yes" or "no" indicating the validity of a kill PIN. There are two ways that kill-PIN verification might be consistently achievable. The first involves modification of tags, the second, modification of readers:

1. *Hobbling the* kill *command:* The EPCglobal standard does not specify a criterion, e.g., a minimum power level, for a tag to accept a kill command. Thus, a manufacturer could create an EPC-compliant tag that always registers insufficient power. In effect, such tags would possess the mandatory kill function in a degenerate form. As the kill function aims at consumer privacy protection, and EPC tags will not reach the hands of consumers in many applications for quite some time, this modification may prove acceptable in some sectors.
2. *Power calibration:* In principle, precise positioning of an EPC tag near a reader and precise calibration of the reader power level could cause a tag to register insufficient power for the kill command. This would depend on the design of the tag and many other factors, and therefore stands only as a hypothetical approach.

In section 3, we propose authentication techniques that assume the ability to validate kill PINs repeatedly in basic EPC tags.

In section 5.1, we describe a situation in which the kill command is useful as a one-time authentication operation, i.e., without the need for non-standard implementation. Our aim there is to use tag authentication as a subsidiary tool to achieve the goal of preventing *en bloc* theft of tag PINs by compromised readers.

Enhanced EPC tags: Enhanced EPC tags respond to a command called access, whose implementation is optional in the EPCglobal standard. When accompanied by a valid 32-bit access PIN, the access command causes a tag to transition into what is called a "secured" state. Tags may be configured such that certain commands only function when a tag is "secured." In particular, read access to the memory banks for the access and kill PINs may be made dependent on an EPC tag being "secured." (The standard supports no other PINs.)

In consequence, although the EPC of a tag may be readily skimmed, a properly configured EPC tag does not promiscuously emit its PINs. Thus the PINs are resistant to skimming. We show how to exploit this feature to achieve a kind of crude challenge-response protocol.

Another useful feature of the EPCglobal standard is the word-level granularity of read and write operations. In particular, it is possible to read or write the upper or lower half of a PIN exclusively.

2.2 Related work

Mandel, Roach, and Winstein have demonstrated the ability to scan certain proximity cards, i.e., the RFID tags used for physical access to buildings, from a range of several feet, despite the fact that these cards have an advertised read range of several inches [17]. They have further showed how to produce low-cost clones, as has Westhues [33] in independent work. Proximity cards of the type that these researchers examined are much like EPC tags: They have no cryptography, and thus no logical-layer resistance to cloning.

Some commercially available RFID tags *can* perform cryptographic challenge-response protocols. Such tags offer resistance to attacks involving skimming and cloning. They cost significantly more than EPC tags, though, and are therefore viable only for niche applications like consumer payments.

Even the use of cryptography, moreover, has not guaranteed resistance to cloning in commercial RFID devices. The Digital Signature Transponder (DST) manufactured by Texas Instruments is an example. DSTs serve as a theft-deterrent in tens of millions of automobiles, supporting an RFID-based authentication protocol between readers in these automobiles and the physical ignition keys of their owners. DST are also present in Speedpass^{TM} payment transponders, which have over six million users [28]. Researchers at Johns Hopkins and RSA Laboratories have recently described a practical cloning attack against Texas Instruments DST devices [1], achievable in consequence of the use of mere 40-bit keys.

Weis et al. have proposed privacy-protecting authentication protocols for tags; their proposals require cryptographic hash functions, however, are are thus unsuitable for Class-1 EPC tags [25, 31].

Juels [12] has proposed what he calls "minimalist" cryptography, namely 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. Nonetheless, our EnhancedTagAuth protocol may be viewed as a very stripped-down adaptation of some of the ideas there.

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 [13]. 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.

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 [10] and also newer algorithms like the NTRU cryptosystem [11]. Even the most lightweight of these many schemes, e.g., [29], 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 [7], these are still well beyond the reach of Class-1 EPC tags today, and unsupported in the EPCglobal standard.

One key idea in this paper for basic EPC tags is the presentation of spurious PINs as a means of testing their authenticity. This is similar in flavor to the notion of "winnowing" introduced by Rivest [23]. 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.

To date, the majority of the scientific literature, e.g., [8, 14, 15, 31, 32] and media coverage, e.g., [6, 19, 34] on RFID security has focused on privacy-related aspects, rather than authentication.

3 Authenticating Basic EPC Tags

We now describe our schemes to help defend basic EPC tags against skimming attacks. Recall that we must assume, as explained above, that the kill command may serve repeatedly to check the correctness of a kill PIN presented by a reader. We shall exploit the PIN-based reader-to-tag authentication feature in the kill operation, turning it on its head to construct tag-to-reader authentication protocols. For clarity of notation, let us denote by PIN-test(K) an EPC-tag (meta-)command that causes a tag to output a bit-response b. The value of A is a '0' if K is the correct kill PIN for the tag and '1' otherwise.

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 K_i denote the currently valid kill PIN for tag T_i. We assume that K_i is generated uniformly at random, and held as a shared secret between the tag and a trusted reader.

We begin by presenting an elementary protocol SimpleTagAuth 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 SimpleTagAuth, presented in Figure 1, a trusted RFID device $\mathcal{R}$ attempts to authenticate a tag $\mathcal{T}$.

```
SimpleTagAuth
    $\mathcal{T}$:                          $T \leftarrow T_i$
    $\mathcal{T} \rightarrow \mathcal{R}$:  $T$
    $\mathcal{R}$:                          if $T = T_x$ for some $1 \leq x \leq N$ then $i \leftarrow x$
                                                else output "unknown tag" and halt
    $\mathcal{R} \rightarrow \mathcal{T}$:  PIN-test($K_i$)
    $\mathcal{T} \rightarrow \mathcal{R}$:  $b$
    $\mathcal{R}$:                          if $b$ = '1' then output "valid"
                                                 else output "invalid"
```

Fig. 1. The SimpleTagAuth protocol

A tag that does not carry a valid identifier T_x for some x (or at least one known to the reader) 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 ĩ that is EPC-compliant but created via a simple skimming attack. Such an EPC-compliant clone ĩ might be easily created, for instance, through configuration of a field-programmable EPC tag. Obviously skimming reveals the EPC of a tag, but not the secret K_i. For ĩ to cause a "valid" output, therefore, its creator would need to guess K_i correctly. For an l-bit PIN, the probability of successful cloning is therefore just 2^{-l}. As a kill PIN in Class-1 Generation-2 tags is 32-bits long, the probability of successful cloning of a single, given EPC tag is therefore less than one-in-a-billion.

When performing active attacks against a tag i, an adversary can of course actively test multiple possible values of K_i. With 32-bit PINs, though, this form of active attack is largely impractical.[3]

3.1 Non-compliant clones

The SimpleTagAuth protocol we have just proposed has a basic vulnerability: If the cloned tag $\tilde{\imath}$ is *not* EPC-compliant, then it can spoof the reader. It suffices for $\tilde{\imath}$ simply to accept *any* PIN, 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 randomly presented PINs that are not valid. If the PIN-test operation yields a '1' 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 BasicTagAuth.

We describe the protocol BasicTagAuth 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 uniformly at random (without duplication). Among these is is randomly inserted the one correct kill PIN K_i in a random position j, which is also output by the function GeneratePINSet. We detail the exact operation of GeneratePINSet at the end of this section.

```
BasicTagAuth[q]
    $\mathcal{T}$:                      $T \leftarrow T_i$
    $\mathcal{T} \to \mathcal{R}$:      $T$
    $\mathcal{R}$:                      if $T = T_x$ for some $1 \le x \le N$ then $i \leftarrow x$
                                            else output "unknown tag" and halt
    $\mathcal{R}$:                      $(j, \{P_i^{(1)}, P_i^{(2)}, \ldots, P_i^{(q)}\}) \leftarrow$ GeneratePINSet($i$)[$q$];
                                        $M \leftarrow$ "valid"

                                        for $n = 1$ to $q$ do
    $\mathcal{R} \to \mathcal{T}$:          PIN-test($P_i^{(n)}$)
    $\mathcal{T} \to \mathcal{R}$:          $b$
    $\mathcal{R}$:                          if $b =$ '1' and $n \neq j$ then $M \leftarrow$ "invalid";
                                            if $b =$ '0' and $n = j$ then $M \leftarrow$ "invalid"

    $\mathcal{R}$:                      output $M$;
```

Fig. 2. The BasicTagAuth protocol

For an attacker that performs skimming attacks only, the best strategy to defeat the protocol BasicTagAuth is to create a clone device that attempts to guess the correct PIN-

[3] Some EPC tags currently defend against PIN-guessing by temporarily disabling a tag when multiple incorrect PINs are presented [22]. These tags have short PINs, e.g., 8 bits in length. It is unclear whether manufacturers of tags with 32-bit PINs will adopt this approach – or whether it is even necessary.

trial j uniformly at random (or contains a pre-programmed guess).[4] The probability of successful attack in this case, i.e., of the cloned tag appearing to be valid, is clearly just $1/q$.

BasicTagAuth 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. Moreover, it is possible to implement this protocol – or any of our other proposed protocols – on a periodic or probabilistic basis, i.e., to test the authenticity of just a fraction of tags.

Stronger attacks: An adversary that performs eavesdropping on the authentication protocol itself, of course, can defeat it completely, as can an adversary that performs the following three-step attack: (1) The adversary skims a tag or otherwise obtains T_i; (2) The adversary interacts with a valid reader and obtains the PIN set $\{P_i{}^{(j)}\}_{j=1}^{q}$; (3) The adversary actively tests values in the PIN set $\{P_i{}^{(j)}\}_{j=1}^{q}$ on tag i.

Generating 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 $\{P_i{}^{(n)}\}_{n=1}^{q}$ may be selected uniformly at random without duplication from $\{0,1\}^k$. The true PIN K_i should then replace a random element $P_i{}^{(j)}$ for $j \in_U \{1, 2, \ldots, q\}$.

The PIN set $\{P_i{}^{(n)}\}$ must remain static over all invocations of BasicTagAuth. 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 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 $S = \{q_0, q_1, \ldots, q_{|S|-1}\}$, let $S_{[z]}$ denote the element $q_{z \bmod |S|}$. GeneratePINSet may be constructed as follows:

```
GeneratePINSet$(i)[q]$
    $Q \leftarrow K_i$;
    for $n = 1$ to $q$ do
        $P_i{}^{(n)} \leftarrow \{\{0,1\}^k - Q\}_{[f(x,i,n)]}$;
        $Q \leftarrow Q \bigcup \{P_i{}^{(n)}\}$;
    $j \leftarrow \{1, 2, \ldots, q\}_{[f(x,i,q+1)]}$;
    $P_i{}^{(j)} \leftarrow K_i$;
    output$(j, \{P_i{}^{(n)}\}_{n=1}^{q})$;
```

Of course, there are many alternative approaches to generating PIN sets, e.g., selecting $P_i^{(n)}$ uniformly from $\{0,1\}^k$ and rejecting if it is in Q – or even retaining duplicates at the cost of a small degradation in security.

[4] Note that the main protocol loop is not halted on determination of tag validity. While continuation is not strictly necessary, our aim is to emphasize the value of concealing timing information that can shed light on the validity of a given tag or a given PIN.

4 Authenticating Enhanced EPC Tags

As we have explained, enhanced EPC tags permit configuration such that a reader must transmit the access PIN to a tag in order to read its resident kill PIN. This opens up the possibility of using the kill PIN for an unintended purpose, as a secret permitting tag authentication. Our proposal here is a conceptually simple one. We show how to authenticate tags using a fix-value mutual-authentication protocol, in which the access PIN serves to authenticate the reader, and the kill PIN in turn serves to authenticate the tag. Let A_i denote the access PIN for tag i. Stripping away the command-layer syntax, we propose the protocol EnhancedTagAuth, presented in Figure 3.[5]

```
EnhancedTagAuth
    $\mathcal{T}$:                          $T \leftarrow T_i$
    $\mathcal{T} \rightarrow \mathcal{R}$:  $T$
    $\mathcal{R}$:                          if $T = T_x$ for some $1 \leq x \leq N$ then $i \leftarrow x, A \leftarrow A_i$
                                                else output "unknown tag" and halt
    $\mathcal{R} \rightarrow \mathcal{T}$:  $A$
    $\mathcal{T}$:                          if $A = A_i$ then $K \leftarrow K_i$
                                                else $K \leftarrow \phi$
    $\mathcal{T} \rightarrow \mathcal{R}$:  $K$
    $\mathcal{R}$:                          if $K = K_i$ then output "valid"
                                                else output "invalid"
```

Fig. 3. The EnhancedTagAuth protocol

An adversary that has skimmed tag i and attempts to simulate it in the EnhancedTagAuth protocol can create a counterfeit device that implicitly accepts the access PIN and then guesses the kill PIN. This will succeed with probability 2^{-l}, where l is the bit-length of the kill PIN – just as for SimpleTagAuth. Thus, an enhanced EPC tag will resist a single invocation of this attack with probability more than one-in-a-billion. This is significantly better than achievable through BasicTagAuth with any practical value of q. On the other hand, the EnhancedTagAuth protocol is still vulnerable to eavesdropping and to the three-step active attack outlined above for BasicTagAuth.

As we noted above, EPC tags support partial reads and writes. PINs are two words long; read and write operations may address just a single word. Thus, it is possible to combat (passive) eavesdropping attacks to a very limited extent by treating the 32-bit kill PIN as two 16-bit secrets $K_{i,1}$ and $K_{i,2}$. For example, readers within one security zone might verify the correctness of $K_{i,1}$, while those in a second security zone check $K_{i,2}$. (In the language of our protocol, $K \leftarrow K_{i,z}$ in security zone $z \in 1, 2$.) Passive eavesdropping within one perimeter, then, would not permit tag cloning in the other.

Finally, we observe that the BasicTagAuth protocol can be used for enhanced EPC tags in the case where the reader has access to only an access PIN or kill PIN for a given tag – or where the tag itself has only one programmed PIN.

[5] A tag effectively ignores an incorrect access PIN. At the security-protocol layer, this is effectively like returning a null response. We reflect this in the assignment $K \leftarrow \phi$.

5 Untrusted Readers

Our working assumption thusfar has been that readers are trustworthy verifiers. We have assumed that the reader $\mathcal{R}$ may be entrusted *a priori* with the PINs for a given tag. This assumption may not always be architecturally desirable, though.

We may wish instead to consider 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 an allegedly valid EPC tag: 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 $\mathcal{T}$.

We show how to modify the protocol BasicTagAuth to achieve this scenario with untrusted readers. Our modified protocol, which we call BasicTagAuth$^+$, is given in Figure 4.

```
BasicTagAuth$^+$[q]
    $\mathcal{T}$:                                  $T \leftarrow T_i$
    $\mathcal{T} \to \mathcal{R} \to \mathcal{V}$:  $T$
    $\mathcal{V}$:                                  if $T = T_x$ for some $1 \leq x \leq N$ then $i \leftarrow x$
                                                        else output "unknown tag" and halt
    $\mathcal{V}$:                                  $(j, \{P_i{}^{(1)}, P_i{}^{(2)}, \ldots, P_i{}^{(q)}\}) \leftarrow \mathsf{GeneratePINSet}(i)[q]$;
    $\mathcal{V} \to \mathcal{R}$:                  $\{P_i{}^{(n)}\}_{n=1}^{q}$;

                                                    for $n = 1$ to $q$ do
    $\mathcal{R} \to \mathcal{T}$ :                     PIN-test$(P_i{}^{(n)})$
    $\mathcal{T} \to \mathcal{R}$ :                     $R^{(n)}$

    $\mathcal{R} \to \mathcal{V}$:                  $\{R^{(n)}\}_{n=1}^{q}$;
    $\mathcal{V}$:                                  $M \leftarrow$ "valid";
                                                    for $n = 1$ to $q$ do
                                                        if $R^{(n)} =$ '1' and $n \neq j$ then $M \leftarrow$ "invalid";
                                                        if $R^{(n)} =$ '0' and $n = j$ then $M \leftarrow$ "invalid"
                                                    output $M$
```

Fig. 4. The BasicTagAuth$^+$ 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 BasicTagAuth 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 successfully clone a tag 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 K_i.

The protocol variant BasicTagAuth$^+$ 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 K_i (from a database, for instance) for any tag identifier T_i. In such a system, compromise of a single reader would result in massive compromise of tag PINs. An attacker with access to the compromised reader would be able to learn the PIN K_i associated with any tag identifier T_i and then clone the tag perfectly.

This situation is particularly problematic because RFID readers will inevitably become ubiquitous peripherals. They will populate warehouses, storage rooms, trucks, and retail environments. In many RFID architectures readers may be given unfettered access to backend systems in order to query for PINs. Use of our proposed protocol $\mathsf{BasicTagAuth}^+$ can help address the problems associated with reader compromise, by limiting access on the part of misbehaving readers.

In some architectures where network failures are a concern, readers or associated devices might store large numbers of tag PINs locally. The protocol $\mathsf{BasicTagAuth}^+$ can offer stronger security even in this environment. Rather than storing PINs locally, readers can instead store the kill 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 the $\mathsf{BasicTagAuth}^+$ 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.

In systems where tag IDs are sparse and hard to predict, we note that a reader might effectively demonstrate radio contact with a tag merely by submitting its identifier T_i.

Achieving authentication of enhanced EPC tags with untrusted readers is a simpler exercise than for basic EPC tags. For enhanced EPC tags, it is not necessary to store kill PINs on readers, but instead sufficient to have a reader transmit a kill PIN to $\mathcal{V}$ for verification. Of course, if kill PINs are also being used for killing, then readers may need more general access to these PINs. In that case, it is possible to take more or less the same approach as with $\mathsf{BasicTagAuth}^+$: Readers do not store correct kill PINs alone, but rather the sets generated by GeneratePINSet.

Remarks: (1) Spurious PINs themselves might be used to trace the origin of counterfeiting attempts. For example, suppose that a counterfeit tag $\tilde{\imath}$ is encountered in the field with an invalid PIN $\tilde{K}_i$ that corresponds to one of the spurious PINs for tag i. In this case, we might flag $\tilde{\imath}$ might as likely to have been fabricated using information from a compromised reader. It is even possible to customize spurious PINs not just for particular tags, but for particular readers or sets of readers. A counterfeit tag emitting a spurious PIN would then provide information on which reader or set of readers leaked the data used in its fabrication. This latter approach, however, would need careful deployment, as intersection among spurious PIN sets leaks information about which PIN is the valid one. Thus, some degree of overlap among sets would be desirable.
(2) While spurious PINs help prevent cloning in our protocols, they do not defend against certain attacks made directly on legitimate tags. For example, if an attacker wishes to kill a tag, and has a small set of candidate kill PINs for it, she can simply try all of the PINs exhaustively. Executing $\mathsf{BasicTagAuth}^+$ on untrusted readers can indeed exacerbate such problems. Of course, this is not an issue when the kill operation is hobbled as we propose above.

5.1 Preventing *en bloc* PIN theft: *fulfillment-conditional* PIN distribution (FCPD)

We have considered techniques by which untrusted readers may be used to authenticate tags. The techniques we have introduced here, though, may subserve a different and somewhat unrelated security goal, namely the prevention of *en bloc* theft of PINs by compromised reading devices. The problem of secure distribution of PINs is one that has seen rather limited treatment in the literature; a rare example is [20]. We propose an approach that we call *fulfillment-conditional* PIN distribution, and abbreviate FCPD.

We observe that our $\mathsf{BasicTagAuth}^{+}$ protocol does not merely verify the authenticity of a tag $\mathcal{T}$; it verifies that a reader $\mathcal{R}$ actually has physical access to $\mathcal{T}$. The idea behind fulfillment-conditional PIN distribution is thus as follows. We ensure that a reader may only download PINs for a particular set of tags if it is entitled to do so by merit of its physically accessing the tags. Viewed another way, we furnish PINs to a reader only if it can prove that it is using them successfully.

Let us assume that a reader $\mathcal{R}$ (being a computationally high-powered device) is capable of strong authentication to a central authority $\mathcal{V}$. The idea is for $\mathcal{V}$ to honor PIN requests by $\mathcal{R}$. $\mathcal{V}$ executes $\mathsf{BasicTagAuth}^{+}$ with $\mathcal{R}$ periodically. If $\mathcal{R}$ provides valid answers, then $\mathcal{V}$ continues to satisfy PIN requests. If not, then $\mathcal{V}$ concludes that $\mathcal{R}$ is not successfully making use of the PINs it receives. This implies that either (1) $\mathcal{R}$ is scanning forged tags or (2) $\mathcal{R}$ is not scanning the tags for which it is requesting PINs.

An important observation is that FCPD works even when $\mathsf{PIN\text{-}test}$ is just a one-time operation, e.g., a conventionally executed kill operation. Since the aim is to ensure proper behavior by $\mathcal{R}$, and not the authenticity of tags themselves, multiple successful tests of PIN validity for a given tag – e.g., multiple tag kills – are unnecessary. Also, FCPD, like our authentication protocols, may be effective even when executed for just a fraction of tags, e.g., probabilistically.

There is a kindred but more naïve approach to preventing *en bloc* theft of PINs by compromised reading devices. $\mathcal{V}$ may simply *meter* the rate at which $\mathcal{R}$ receives PINs. If this rate exceeds a certain threshold, $\mathcal{V}$ may either refuse to transmit further PINs for some period of time or may flag $\mathcal{R}$ as compromised. For example, if $\mathcal{R}$ is a reader at a retail point of sale, then $\mathcal{V}$ might limit the number of PIN requests by $\mathcal{R}$ to, say, 10,000 per hour.

FCPD, though, has a couple of advantages over metering:

1. **Detection efficiency:** FCPD can efficiently detect reader compromise after just a small number of requests. When based, for example, on $\mathsf{BasicTagAuth}^{+}$ with $q = 2$, FCPD can detect rogue downloading of mere tens of PINs with overwhelming probability. Metering does not permit this level of sensitivity. In fact, an attacker can easily evade detection in the metering approach by downloading PINs at a fractionally lower rate than the detection threshold. FCPD, on the other hand, detects attacks probabilistically, and thus renders evasion of detection more difficult.
2. **Calibration:** Metering requires careful calibration of the detection threshold. If set too low, false positives will result; set too high, and false negatives will result. FCPD requires less delicate calibration. False positives, for example, should not occur in a system where tags function properly, while the false negative rate may be set quite low with little sacrifice of efficiency.

Of course, an attacker that uses a compromised reader to access tags directly, e.g., to kill them, can defeat the FCPD approach. FCPD, however, at least constrains an attacker to using PINs immediately in order to exploit them. It may therefore be desirable to strengthen FCPD with complementary countermeasures, including metering and detection of kill-command or access-command emissions in inappropriate physical locations, e.g., near store shelves.

6 Stronger Attacks

Skimming is perhaps the easiest and most practical cloning attack and therefore the most important to defend against. Stronger attacks, however, would defeat our proposed protocols, as in the following examples.

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. In a naïvely architected system, compromise of a valid reader could 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 can therefore reverse-engineer a captured tag and extract its PIN. Such an adversary can of course clone the tag perfectly.
3. **Active attacks:** As we have noted, an adversary can extract the PINs from a target EPC tag i and clone it on performing the following three steps: (1) Obtain T_i; (2) Interact with a valid reader that executes one of our authentication protocols; and (3) Interact with the tag i. Without more functionality in EPC tags, we believe it is not possible to defend effectively against such attacks. Thankfully, such attacks require more sophistication than skimming alone.

   Man-in-the-middle attacks (in which the attacker creates a real-time "wormhole" between the target EPC tag and reader) are a general security problem for RFID systems [16].

   A very important point to note here is that ***not all readers in an RFID system need be entrusted with tag-authentication capabilities***. The authentication protocols we propose may be executed on just a small set of highly secured readers. By restricting counterfeit-tag detection to within a narrow perimeter, we can help mitigate system vulnerability to active attacks, and in particular the three-step attack described above.
4. **Eavesdropping:** As we have noted already, an adversary capable of full eavesdropping on the communications between the reader and tag can easily harvest the correct PINs for a tag. There are some important technical qualifications 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 can therefore more easily 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).

   Recognizing this asymmetry in signal strength, the EPCglobal standard prescribes protocols in which tags transmit random pads (bit-strings) to readers. Readers use these

pads effectively to encrypt sensitive data, namely PINs, when communicating with tags. Assuming good random-number generation on the tags, this approach renders eavesdropping feasible only on the tag-to-reader channel.
Thus, to mount a successful cloning attack against our protocols, an adversary would need to eavesdrop on the weaker tag-to-reader channel.

To a limited extent, periodic re-writing of EPC tag PINs can help defend against these attacks. An adversary capable of 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 a tag will be unable to seed a system with clones that remain up-to-date. There is a limitation on the regularity with which PIN changes can be viably performed, though. In general, writing to RFID tags is a more difficult operation than reading; it is less reliable, and requires greater reader proximity. Additionally, writeable memory, i.e., EEPROM, has a limited lifetime. Nonetheless, updates performed with only modest frequency could offer considerably strengthened security. This idea is broadly explored in [12].

7 Conclusion

We have proposed simple, practical authentication techniques that combat skimming attacks against EPC tags that are compliant with the EPC Class-1 Generation-2 UHF-RFID standard of EPCglobal. Our schemes involve 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 proposals will prove valuable in real-world systems.

The authentication protocols we propose do not defend against attacks that are substantially more sophisticated than skimming. We feel, however, that our techniques can provide valuable enhancement to real-world security. They are probably the best one can do within the constraints imposed by the EPCglobal Class 1 Generation 2 standard which, again, contains no explicit anti-cloning features at all. Moreover, as we have explained, by creating a more highly secure perimeter for the set of readers executing our authentication protocols, it is possible to limit vulnerability to active attacks.

We have also showed how our techniques can subserve a different goal, namely the secure distribution of PINs in RFID systems. We have proposed an approach called fulfillment-conditional PIN distribution that can help address the problem of *en bloc* theft of PINs by compromised readers.

Acknowledgments:

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

References

1. S. Bono, M. Green, A. Stubblefield, A. Juels, A. Rubin, and M. Szydlo. Security analysis of a cryptographically enabled RFID device, 2005. Pre-print. Available at www.rfidanalysis.org.

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, 2005.
4. EPC^{TM} Radio-Frequency Identity Protocols Class-1 generation-2 UHF RFID Protocol for Communicaitons at 860 MHz - 960 Mhz, Version 1.0.8, 2005. Available at http://www.autoid.org/2005docs/SG3_200501_435_UHFGen2_v1.0.8.pdf.
5. Security technology: Where's the smart money? *The Economist*, pages 69–70, 9 February 2002.
6. RFID: eWeek.com special report, 2004. Available at http://www.eweek.com/category2/0,1738,1568291,00.asp.
7. M. Feldhofer, S. Dominikus, and J. Wolkerstorfer. Strong authentication for RFID systems using the AES algorithm. In M. Joye and J.-J. Quisquater, editors, *Cryptographic Hardware and Embedded Systems (CHES)*, pages 357–370. Springer-Verlag, 2004. LNCS no. 3156.
8. 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.
9. 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.
10. 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.
11. 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.
12. A. Juels. Minimalist cryptography for low-cost RFID tags. In C. Blundo and S. Cimato, editors, *Security in Communication Networks (SCN 04)*, pages 149–164. Springer-Verlag, 2004. LNCS no. 3352.
13. A. Juels. 'Yoking-proofs' for RFID tags. In *PerCom Workshops 2004*, pages 138–143. IEEE Computer Society, 2004.
14. 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.
15. 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.
16. K. Kfir and A. Wool. Picking virtual pockets using relay attacks on contactless smartcard systems, 2005. Pre-print. Available at http://eprint.iacr.org/2005/052.
17. J. Mandel, A. Roach, and K. Winstein. MIT Proximity Card Vulnerabilities. Technical report, Massachusetts Institute of Technology, March 2004. Slide presentation. Available at http://web.mit.edu/keithw/Public/MIT-Card-Vulnerabilities-March31.pdf.
18. 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.
19. D. McCullagh. RFID tags: Big Brother in small packages. *CNet*, 13 January 2003. Available at http://news.com.com/2010-1069-980325.html.
20. David Molnar and David Wagner. Privacy and Security in Library RFID : Issues, Practices, and Architectures. In B. Pfitzmann and P. McDaniel, editors, *Computer and Communications Security*, pages 210 – 219. ACM, 2004.
21. Nokia unveils RFID phone reader. *RFID Journal*, 17 March 2004. Available at http://www.rfidjournal.com/article/view/834.
22. RFID, privacy, and corporate data. *RFID Journal*, 2 June 2003. Feature article. Available at www.rfidjournal.com on subscription basis.
23. R. L. Rivest. Chaffing and winnowing: Confidentiality without encryption. *CryptoBytes*, 4(1):12 – 17, Summer 1998.
24. M. Roberti. EPCglobal ratifies gen 2 standard. *RFID Journal*, 16 December 2004. Available at http://www.rfidjournal.com/article/articleview/1293/1/1/.
25. S. E. Sarma, S. A. Weis, and D.W. Engels. Radio-frequency-identification security risks and challenges. RSA Laboratories. *CryptoBytes*, 6(1), 2003.
26. 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.
27. 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.

28. Stop & Shop supermarket company to test ExxonMobil Speedpass. *Texas Instruments RFID eNews*, 10, July 2002. Available at http://www.ti.com/tiris/docs/news/eNews/eNewsVol10.pdf.
29. J. Stern and J. Stern. Cryptanalysis of the OTM signature scheme from FC'02. In R. Wright, editor, *Financial Cryptography '03*, pages 138–148. Springer-Verlag, 2003. LNCS no. 2742.
30. C.P. Wallace. The color of money. *Time Europe*, 158(11). 10 September 2001.
31. 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.
32. S.A. Weis. Radio-frequency identification security and privacy. Master's thesis, M.I.T., June 2003.
33. J. Westhues. Proximity cards, October 2003. Web site. Available at http://cq.cx/prox.pl.
34. Wal-Mart, DoD Forcing RFID. *Wired News*, 3 November 2003. Available at http://www.wired.com/news/business/0,1367,61059,00.html.