A Robust and Secure RFID-Based Pedigree System (Short Paper)

Chiu C. Tan and Qun Li

Department of Computer Science
College of William and Mary

Abstract. There has been considerable interest recently on developing a system to track items like pharmaceutical drugs or food products. Such a system can help prevent counterfeits, aid product recall, and improve general logistics. In this paper, we present such system based on radio frequency identity (RFID) technology. Our solution provides the means of storing the entire movement of the item from original manufacturer to final consumer on the RFID tag itself, and also makes it more difficult to introduce large numbers of counterfeits. The solution also allows the end user to easily verify the authenticity of the item.

1 Introduction

A tracking system, or electronic pedigree system, is an architecture for creating digital documentation for movement of goods. With this documentation, the entire route from beginning to end can be recreated. For instance, consider the case of some cargo shipped from a supplier to a customer. The electronic pedigree tracks the journey from the supplier's warehouse until it reaches the customer. It includes information like which intermittent stops were made and possibly more detailed information like which trucks were used. This form of documentation is useful for routine inventory control and tracking, as well as rare time-sensitive operations like product recalls. An electronic pedigree that tracks goods on an individual packaging basis can be used to defend against the counterfeits. Instead of relying on random checks at large warehouses, a per item electronic pedigree allows the end user who just purchased a product to verify the authenticity using electronic pedigree, thus improving the detection of counterfeits.

Recent developments in radio frequency identity (RFID) technology have made it possible to implement an electronic pedigree on a per item basis. RFID technology is made up of small powerless tags and their corresponding readers. These tags can be attached to different products like shipping crates or bottles of medication, and can contain information like the unique identity number of the product, origin, transit locations, storage instructions. RFID readers obtain the stored information by querying the tag from a distance without line of sight. One possible method of integrating RFID technology is described in [12] as "track and trace". It uses a central database to keep track of the unique ID number embedded in each tag. When products with attached RFID tags are received,

P. Ning, S. Qing, and N. Li (Eds.): ICICS 2006, LNCS 4307, pp. 21–29, 2006.

an RFID reader reads in the ID from each tag. The ID can be verified against the central database containing information like the location of particular ID. An ID that shows up in the wrong location, or does not exist in the database could indicate potential problems. However, the paper also pointed out that this method is not robust enough against inevitable human errors, or instances where access to the database is limited.

Furthermore, the use of RFID tags introduces new security problems. Since an RFID tag can be read from a distance without line of sight, an adversary can steal large numbers of RFID data and then place the real data onto counterfeit RFID tags. This way, both the real and fake RFID tags contain legitimate information. Trials on RFID-enhanced passports reported RFID readers being able to access RFID data from 30 feet away [13]. For an RFID based pedigree system to function, it has to be robust enough to function without constant access to a central database. It also has to defend against counterfeits which can be introduced anywhere inside the supply chain.

In this paper, we present an RFID-based electronic pedigree system that does not depend on constant access to a database to function. Our system adds pedigree data onto the RFID tag itself in a secure manner. Since the RFID tag is always attached to the object, receiving the object means receiving the tag as well. With more information stored on the tag, pedigree information can be accessed more conveniently. Our solution also provides the end user, or consumer, with a means of easily verifying the authenticity of a tag while preserving his privacy.. The rest of the paper is as follows. The next section discusses some related work on RFID. Section 3 formalizes our problem and section 4 present our basic pedigree scheme. Section 5 improves on the basic scheme and section 6 concludes.

2 Related Work

There has been relatively little research that explicitely addresses using RFID for tracking purposes. Gonzalez et. al [3] addresses the problem of managing large quantities of RFID data generated when RFID tags are widely used for tracking. The work focuses on techniques for aggregating and indexing RFID data and query processing. Staake et. al [12] discusses how RFID used in tracking inventory can also be used for anti-counterfeiting purposes.. It described the track and trace method whereby each object is tagged with its own RFID tag embedded with some unique data. A main database is used to keep track of the tag data. As each object moves through the supply chain, information like object location can be matched against the tag unique data and database. This makes introducing counterfeits more difficult. However, this method requires all entities to update the database promptly, making it less robust to inevitable errors.

Texas Instruments (TI) [8] presented the *authenticated RFID* model which combines public key and RFID, and is targeted at pharmaceutical products. Under this model, the unique id of each RFID is first hashed, and then digitally signed with private key. This signature is stored onto the RFID tag itself together with the tag unique id before leaving the drug manufacturer. Later, authorized RFID readers like the pharmacist receiving the RFID tag authenticate the tag by reading in the digital signature and unique id. The pharmacist decrypts the signature with the public key, and compares the value against the hashed result of the unique id. If they match, then the tag is considered genuine. This model also allows additional information like timestamps to be signed and placed onto the RFID tag for additional security. However, as Juels [6] pointed out, this model has a vulnerability. An adversary will be unable to forge the signature, but is perfectly able to copy it. This means that an adversary could simply copy the genuine RFID tag data, and then place them onto the counterfeit drugs. Our solution also uses public key cryptography, but specifically addresses the problem of copying.

The copying of data from real RFID tags is know as skimming the tag, and placing real tag data onto fake RFID tags is known as cloning the tag. Juels [5] discusses the risks of RFID tag cloning, and provided solutions for a reader to authenticate a tag. The basic solution assumes that each RFID tag has a secret that is not reveled when queried. An authenticated reader will know this secret, and challenges the tag with it. The RFID tag is designed to return a 1 bit if the challenge secret matches its own secret, otherwise returns 0. So the RFID reader issues a series of challenges, some using the the tag secret, others not. A real tag will be able to return the correct answer each time. A counterfeit tag which was cloned from the real tag will not know this secret. However, this particular solution may require several interactions between reader and tag before the reader is satisfied that the tag is genuine, make it less efficient.

Another cloning resistant scheme by Dimitriou [2] uses a different approach. His approach uses a secure external server for authentication. The RFID tag returns a reply that can only be decrypted by the external server. The server releases the tag data to the reader only after authenticating him. This means that an adversary will not be able to obtain the RFID tag data without going through the secure server, thus preventing skimming. However, this scheme like track and trace, requires persistent access to a database.

There are other security protocols that can prevent cloning, and we refer interested readers to the excellent website maintained by Avoine [1], and recent survey papers [6,11]. In general, they all rely on only having authenticated RFID readers having access to RFID tag data. However, this concept can create potential privacy problems when applied to the electronic pedigree system. The problem lies in authenticating the RFID readers. Consider the example of a drug company shipping drugs to the clinic. After a patient purchases the medication, he would like to read the RFID tag data to make sure it is genuine. If only authenticated readers can read the RFID tag, then the patient will have to authenticate himself to the drug company, thus violating his privacy. Allowing *any* RFID reader to reader the tag protects the patient's privacy, but also allow malicious agents to clone the RFID tags. To prevent large scale RFID tag data to be stolen without use authorized RFID readers, we borrow a similar idea from [7] that uses both a optical and radio channel. Their paper focuses on banknotes embedded with an RFID tag. The RFID data is changed periodically so that it does not always return the same value, thus serving as a pseudo identifier for the banknote. The serial number is the optical channel that controls the changing of RFID data so that the data cannot be changed by malicious agents remotely.

3 Problem Formulation and Assumptions

We can abstract the problem of moving products from manufacturer to consumer as

$$D_0 \rightarrow (D_1 \cdots D_n) \rightarrow C$$

where D_0 is the original manufacturer, and C is the final consumer. D_0 is assumed to be always trusted, and C is assumed to always verify his purchase. $D_1 \cdots D_n$ are the different intermediaries that the product goes through before reaching the consumer. These intermediaries are entities that come into contact with the product, for example resellers, warehouse operators or delivery trucks. Each individual product has a unique RFID tag, T, with identity, id. Subscripts are used to distinguish one tag from another. Since every product has an RFID tag, referring to a particular tag, T_i, refers to both the RFID tag and the product. We consider an adversary denoted as α that can attack anywhere between $(D_1 \cdots D_n)$. The goal of α is to create large numbers of counterfeit RFID tags that are indistinguishable from real RFID tags.

We assume that different intermediaries like D_i and D_j can verify each other's identity and create a secure channel to exchange information. We also assume that consumers will have easy access to RFID readers and barcode readers. This is a realistic assumption since these readers are beginning to be integrated with cell phones [9],[10]. The RFID tags used in this paper are assumed to have a memory divided into multiple cells. This division of RFID memory into different cells was also adopted in [7] in which the RFID attached to a banknote has two memory cells. Finally the memory cells in the basic pedigree scheme are write once only, while the cells in the improved scheme can be written multiple times. Both types of RFID tags are currently available [4].

4 Basic Pedigree Scheme

In the basic scheme, the tags attached to each product have multiple memory cells, in which each cell can only be written once. We assume that the tag has n memory cells, and there are less than n intermediaries. Furthermore, each product also contains a 2D barcode which stores more data than a conventional 1D barcode. This 2D barcode is place in such a manner that is difficult to read without damaging the packaging. In a packet of medication, for example, the RFID tag can be attached to the outside packaging while the barcode is placed inside the packaging. The only way to read the 2D barcode is to open the packaging.

Consider the case when D_0 is manufacturing a product with a particular tag T_i. D_0 first generates an id_i and stores the pairing of id_i and T_i. It then creates a 2D barcode embedded with id_i and attaches the barcode to the product. Finally, D_0 stores the hashed result of id_i, $h(id_i)$ into the first cell of T_i. Figure 1 illustrates T_i and barcode after preprocessing. When D_0 prepares to hand T_i off to D_1, both parties first authenticate each other. Then, D_1 sends a random number n_{D_1} to D_0. D_0 signs the concatenation of this random number and D_1's identity using his private key, $(n_{D_1}||D_1)_{D_0}$, and stores the result into the next empty memory cell of T_i. When D_1 receives T_i, he reads in the last written memory cell in T_i and applies D_0 public key to the result. If D_1 gets back n_{D1}, he is convinced that T_i comes from D_0. This entire transaction can occur in real time just as D_0 hands off T_i to D_1. Figure 2 illustrates T_i when D_1 receives it. Figure 2 illustrates T_i when D_1 receives it. The same authentication process is performed by the remaining intermediaries when they receive T_i. Thus when D_1 hands T_i off to D_2, D_1 will add $(n_{D_2}||D_2)_{D_1}$ to T_i, and so on. D_2 can also verify that D_1 is supposed to possess T_i by checking the earlier memory cells in T_i. D_2 first asks D_1 who it receive T_i from. Then, D_2 can use D_0's public key to open the package $(n_{D_1}||D_1)_{D_0}$ found in the earlier memory cell and check if the D_1 identity is indeed stored the earlier memory cell. More generally, an intermediary D_i can *backtrack* back to D_0 by reading the data off the RFID tag and asking earlier intermediaries and thus recreating the entire movement of a particular product from the data stored in the RFID tag. This approach is feasible when the intermediaries are related and their public keys easily available., for example when T_i is passed from one FedEx truck to another, or when intermediaries are compelled to cooperate by the relevant authorities.

Barcode	Memory Cell 1	Memory Cell 2	$\cdots$	Memory Cell n
id_i	$h(id_i)$		$\cdots$	

Fig. 1. T_i after preprocessing

Barcode	Memory Cell 1	Memory Cell 2	$\cdots$	Memory Cell n
id_i	$h(id_i)$	$(n_{D_1}, D_1)_{D_0}$	$\cdots$	

Fig. 2. T_i after D_0 passes off to D_1

When the consumer receives T_i, he opens the package to reveal the 2D barcode. He then checks if the hashed result of the 2D barcode is equivalent to the data stored in the first memory cell of T_i. If they match, he then checks $h(id_i)$ against a public website managed by D_0. Since D_0 stores the pairing of id_i and T_i during preprocessing, D_0 will be able to identify a valid $h(id_i)$. If either test fails, the consumer rejects the package and contacts the relevant authorities.

4.1 Evaluating the Basic Scheme

A robust pedigree system needs to store and recover information from the RFID tag without using a persistent central server. From the scheme above, we see that storing data onto the RFID tag does not require a central server. Here, we show how to obtain information from the RFID tag data. A secure pedigree system has to prevent large number of counterfeit RFID tags from being accepted by intermediaries.

A key function of a pedigree record is to retrieve information about a particular product like which warehouse it was stored in or which truck transported it. The difference of a pedigree system using RFID is that it allows the creation of a pedigree record on a *per item* basis. Thus, an effective pedigree system will be able to easily retrieve this information. Every intermediary D_i, that comes into contact with T_i stores the identity of the next intermediary D_j it passes T_i to by storing $(n_{D_j}||D_j)_{D_i}$. Thus, when there is a need to identify all the products that came into contact with D_j due to a contamination or product recall, the relevant authorities can release the identities and the public keys of the intermediaries around like D_i, D_j, D_k. Concerned consumers can scan the RFID tag of their own products and apply the different public keys to verify if they have a product that passed through D_j. Intermediaries can also verify their inventories since RFID tags can be read quickly without line of sight. Note that the electronic pedigree based on RFID tags does not supplant existing inventory management, but complements it. Thus we can assume that relevant authorities can identify the potential intermediaries and disseminate their public key information. The entire route taken by a particular product can also be recreated by backtracking back to D_0.

For an adversary α to create a large number of counterfeits to flood the system, α will also need to convince the intermediaries that it is a legitimate recipient of the product. Consider the case where D_j is supposed to pass T_i to D_k. α can scan T_i from D_j, attach it to its counterfeits, and try to pass it off to D_k. Assuming that D_j got T_i from D_i, the contents of T_i scanned by α will be

$$\{h(id_i)\big|(n_{D_1}, D_1)_{D_0}\big|\cdots\big|(n_{D_j}, D_j)_{D_i}\}$$

After α passes of T_i to D_k, T_i will become

$$\{h(id_i)\big|(n_{D_1}, D_1)_{D_0}\big|\cdots\big|(n_{D_j}, D_j)_{D_i}\big|(n_{D_k}, D_k)_{\alpha}\}$$

When D_k asks α to verify that it is a legitimate recipient of T_i, α will have to provide the identity of the intermediary he received the product from. However, the previous memory cell contains $(n_{D_j}, D_j)_{D_i}$, and not $(n_\alpha, \alpha)_{D_i}$ which D_k is expecting. Thus, α will not be able to convince D_k is a legitimate recipient of T_i. Since D_k can continue to ask each previous intermediary up till the original D_0 which is always trusted, multiple adversaries colluding can still be identified.

However, the above scheme does not protect against a legitimate intermediary who is also an α. Consider the case where D_k receives a legitimate tag T_i from D_j. D_k is also malicious, so he reads the data from the T_i, and place the data onto another RFID tag attached to a counterfeit product. Let us term this counterfeit product's RFID tag as $\hat{T}_i$. Now, the backtracking approached used above does not work, since T_i and $\hat{T}_I$ both contain the same data. To detect this form of counterfeit, we rely of the consumer verifying the RFID tag. When the consumer wishes to verify his purchase, he will first read the id_i stored in the 2D barcode and compare the hashed result of the 2D barcode against the first memory cell of T_i which is $h(id_i)$. Since a one-way hash is used, α will not be able to derive id_i from $h(id_i)$. Thus, the counterfeit product will not have a 2D barcode whose hashed value matches the value on the RFID tag. An alternative is for α to create a fake id_i termed $\hat{id}_i$, and create a fake tag $\hat{T}_i$ that has $h(\hat{id}_i)$. However, when the consumer checks the hashed value against the public website maintained by D_0, he will discover $h(\hat{id}_i)$ is invalid. Finally, α can obtain a legitimate 2D barcode by physically opening one product, and then replicate the same T_i and 2D barcode on multiple counterfeits. While this form of attack is able to fool a consumer, the scope of such an attack is rather limited. Since barcode contains a unique identifier, all the counterfeit RFID tags by α will have the same $h(id_i)$ stored in the first memory cell, making it easy for intermediaries to detect.

5 Improved Pedigree Scheme

One drawback of the basic scheme is it is unsuitable when there are too many intermediaries. The number of memory cells needed will be too expensive to attach to individual products. The improved scheme limits the number of memory cells needed by compressing the data. The improved scheme retains the use of the 2D barcode, but uses a re-writable RFID tag. This means that the data on a particular cell on the RFID tag can be overwritten.

The improved scheme requires three memory cells on the RFID tag. The first cell is used to store the hashed result of the barcode. The remaining two cells are used to store signatures from the different intermediaries. The improved scheme retains the same preprocessing step as the basic scheme. For sake of brevity, we denote $h(id_i)$ as r_1 and $(n_{D_1}, D_1)_{D_0}$ as d_1. Both r_1 and d_1 are stored in the first memory cell of T_i. This cell cannot be over written. Figure 3 shows T_i when D_1 receives it. When D_1 hands off T_i over to D_2, it will generate $d_2 = (n_{D_2}, D_2)_{D_1}$ and $r_2 = h(r_1||d_1)$ and store it into the next empty memory cell. The $||$ denotes concatenation. D_2 handing off to D_3 will have $d_3 = (n_{D_3}, D_3)_{D_3}$ and $r_3 = h(r_2||d_2)$. Figure 4 shows T_i when D_3 receives the it from D_2. When D_3 prepares to pass T_i to D_4, there are no more empty cells left in T_i. D_3 then replaces the contents of memory cell 2 with information regarding $d_4 = (n_{D_4}, D_4)_{D_3}$ and

Barcode	Memory Cell 1	Memory Cell 2	Memory Cell 3
id_i	$d_1 = (n_{D_1}, D_1)_{D_0}$		
	$r_1 = h(id_i)$		

Fig. 3. T_i when passed to D_1

Barcode	Memory Cell 1	Memory Cell 2	Memory Cell 3
id_i	$d_1 = (n_{D_1}, D_1)_{D_0}$	$d_2 = (n_{D_2}, D_2)_{D_1}$	$d_3 = (n_{D_3}, D_3)_{D_2}$
	$r_1 = h(id_i)$	$r_2 = h(r_1 \| d_1)$	$r_3 = h(r_2 \| d_2)$

Fig. 4. D_3 getting T_i from D_2

$r_4 = h(r_3||d_3)$. Figure 5 illustrates T_i when D_4 receives it. D_4 can verify that d_4 is correct by applying D_3's public key and checking the random number n_{D_4}. D_4 uses r_3 and d_3, both found in memory cell 3, to verify that D_3 computed the correct r_4 value. Using r_4, we can derive the structure shown in Figure 6, where the information captured in the basic scheme can be derived.

As in the basic scheme, T_i can be backtracked to D_0 by having the intermediary ask each previous intermediary whom they received T_i from. This information is then checked against the data found in the tag. The consumer can verify the product using the 2D barcode and r_i found in memory cell 1 as in the basic scheme. However, unlike the basic scheme, this solution does not permit the consumer or an intermediary from checking whether T_i had passed through any particular intermediary simply by releasing the identity and public keys. This information can only be found via backtracking.

Barcode	Memory Cell 1	Memory Cell 2	Memory Cell 3
id_i	$d_1 = (n_{D_1}, D_1)_{D_0}$	$d_4 = (n_{D_4}, D_4)_{D_3}$	$d_3 = (n_{D_3}, D_3)_{D_2}$
	$r_1 = h(id_i)$	$r_4 = h(r_3 \| d_3)$	$r_3 = h(r_2 \| d_2)$

Fig. 5. D_4 getting T_i from D_3

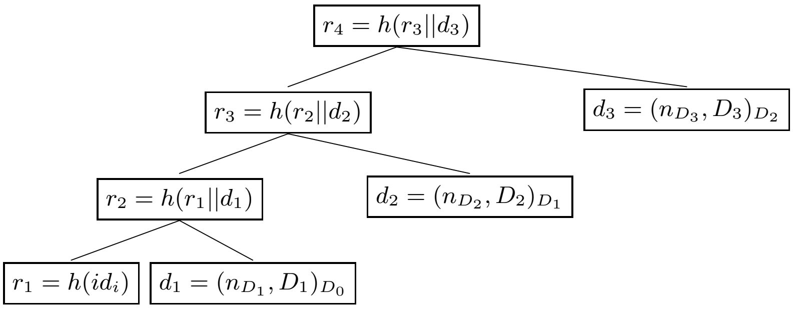
$r_4 = h(r_3||d_3)$
$r_3 = h(r_2||d_2)$
$d_3 = (n_{D_3}, D_3)_{D_2}$
$r_2 = h(r_1||d_1)$
$d_2 = (n_{D_2}, D_2)_{D_1}$
$r_1 = h(id_i)$
$d_1 = (n_{D_1}, D_1)_{D_0}$

Fig. 6. Building pedigree from r_3

6 Conclusion

In this paper, we examine how RFID tags can be used to establish an electronic pedigree. We present two schemes that allow pedigree information to be stored directly onto the RFID tag itself. The end user can verity the authenticity of his purchase. Finally, both schemes make large scale counterfeits difficult to accomplish.

Acknowledgment. The authors would like to thank all the reviewers for their helpful comments. This project was partially supported by US National Science Foundation award CCF-0514985.

References

1. G. Avoine. http://lasecwww.epfl.ch/∼gavoine/rfid/.
2. T. Dimitriou. A lightweight RFID protocol to protect against traceability and cloning attacks. In *Conference on Security and Privacy for Emerging Areas in Communication Networks – SecureComm*, Athens, Greece, September 2005. IEEE.
3. H. Gonzalez, J. Han, X. Li, and D. Klabjan. Warehousing and analyzing massive rfid data sets. In *ICDE '06: Proceedings of the 22nd International Conference on Data Engineering (ICDE'06)*, page 83, Washington, DC, USA, 2006. IEEE Computer Society.
4. T. Hassan and S. Chatterjee. A taxonomy for rfid. *hicss*, 8:184b, 2006.
5. A. Juels. Strengthening EPC tags against cloning. In *WiSe '05: Proceedings of the 4th ACM workshop on Wireless security*, pages 67–76, New York, NY, USA, 2005. ACM Press.
6. A. Juels. RFID security and privacy: A research survey. *IEEE Journal on Selected Areas in Computing*, 24(2):381–394, February 2006.
7. A. Juels and R. Pappu. Squealing euros: Privacy protection in RFID-enabled banknotes. In R. N. Wright, editor, *Financial Cryptography – FC'03*, volume 2742 of *Lecture Notes in Computer Science*, pages 103–121, Le Gosier, Guadeloupe, French West Indies, January 2003. IFCA, Springer-Verlag.
8. Texas Instruments Incorporated. Securing the pharmaceutical supply chain with rfid and public-key infrastructure (PKI) technologies. In *http://www.ti.com/rfid/docs/customer/eped-form.shtml*.
9. MobileMagazine. http://www.mobilemag.com/content/100/104/c2607/.
10. Nextel.$http://www.nextel.com/en/solutions/special_devices/roadrunner.shtml$.
11. M. Rieback, B. Crispo, and A. Tanenbaum. The evolution of RFID security. *IEEE Pervasive Computing*, 5(1):62–69, January–March 2006.
12. T. Staake, F. Thiesse, and E. Fleisch. Extending the EPC network: the potential of rfid in anti-counterfeiting. In *SAC '05: Proceedings of the 2005 ACM symposium on Applied computing*, pages 1607–1612, New York, NY, USA, 2005. ACM Press.
13. J. Yoshida. Tests reveal e-passport security flaw. *EE Times*, 2004.