Tamper Detection in RFID Tags using Fragile Watermarking

Vidyasagar Potdar, Elizabeth Chang
Digital Ecosystems and Business Intelligence Institute, Curtin Business School, Curtin University of Technology
Perth, Western Australia
Vidyasagar.Potdar@cbs.curtin.edu.au, Elizabeth.Chang@cbs.curtin.edu.au
www.debii.curtin.edu.au www.rfidtamperdetection.com

***Abstract*- Security and privacy are one of the two primary concerns with RFID (Radio Frequency Identification) adoption. While the mainstream RFID research is focused on solving the privacy issues, this paper focuses on security issues in general and data tampering in particular. We specifically consider the issue of detecting data tampering on the RFID tags for applications such as data integrity management. To address this issue, we present a novel fragile watermarking scheme, which embeds a fragile watermark (or pattern) in the serial number partition of the RFID tag. This pattern is verified to identify whether or not the data on the RFID tags has been tampered with. The novelty of this watermarking scheme lies in the fact that we have applied watermarking technology to RFID tags; in comparison, most of the existing watermarking schemes are limited to images, or audio or video applications. We term this scheme *TamDetect* because it is a tamper detection solution. *TamDetect* is designed such that it can be easily plugged into existing RFID middleware applications. This proposal is one of the first works that integrates watermarking and RFID technologies together. This paper provides a detailed theoretical foundation for the *TamDetect* solution.**

I. INTRODUCTION

A RFID tag is an electronic device that holds identification data. Typically, the RFID tag is attached to items and contains a serial number, which is used to uniquely identify them. RFID technology uses radio waves to automatically identify items which have RFID tags attached to it.

This new generation technology was initially developed with the aim to manage and track items, but is used in many other applications these days e.g. supply chain automation, asset tracking, medical applications, people tracking, manufacturing, retail and inventory tracking, livestock tracking and tracking exact timing in sports events. As pointed out by RFIDExchange "*RFID applications are limited only by imagination*" [19]. It can be used any where and every where if possible.

RFID technology is composed of three main components; *firstly,* a RFID tag, which contains the identification number, *secondly,* a RFID Reader, which activates the tag to broadcast its identification number and *finally*, a RFID Middleware, which integrates the information from the reader to the backend database systems [16, 17]. This is shown in Fig. 1.

However widespread adoption of RFID technology has been hindered because of several inherent issues that arise from its usage. The main issues are privacy, security, and cost, however deployment, scalability and resilience cannot be excluded from this list. *Privacy* is the main issue as far as the adoption is considered, whereas *security* is the main issue as far as implementation is considered.

Since RFID tags can be used to track items and people it raises many privacy issues. If RFID is deployed in full scale it would raise several privacy concerns because RFID tags can be used to track consumer behavior, which can further be used to analyze consumer habits. It can be even be used for steganographic surveillance i.e. deploying secret RFID tags for tracking. With the size of RFID tags reducing day by day it has now become possible to hide them within products without the owners consent. E.g. Henning et al. (2005) pointed out that RFID tags are already been hidden in packaging [16]. A scenario of hidden RFID testing was discovered in a Wal-Mart store in Broken Arrow where secret RFID readers were kept to track customer action [19]. Using RFID technology could even trigger anti-social activities. Criminals with RFID readers could look for people carrying valuable items and can launch selective attacks [16]. However most of these issues can be tackled by privacy enforcement laws, which can be incorporated in the nation's legal framework.

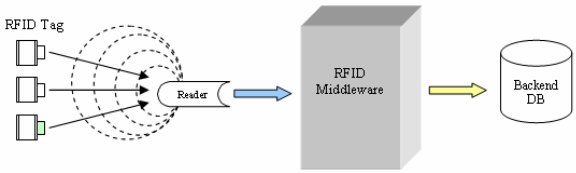

Fig. 1 RFID Architecture

Now considering the security aspect of RFID tags, it is worth noting that most of the Class 0 and Class 1 RFID tags are not capable of any secure communication. This is attributed to the fact that this class of RFID tags does not have enough computation power and storage capacity to perform any encrypted communication as a result all the data is transmitted in open which leaves doors open for eavesdroppers and attackers. For example Weis et al. (2004) estimate that as few as 500-5000 gates are employed in a typical RFID design, which are normally used for basic logic operation; hence there is no room for extras such as security [23]. In particular, symmetric encryption schemes such as Advanced Encryption Standard (AES) or hash functions such as Secure Hash Algorithm (SHA) or pseudo-random functions are not possible

on today's low end RFID tags. In addition to this the wireless nature of this communication architecture complicates this issue even further. Because the communication between reader and tag is wireless, it increases the possibility of eavesdropping by third parties. Considering this insecure communication, data tampering with the RFID tags cannot be ruled out and this is one of the most important security issues, which needs to be tackled if large scale RFID deployment is to be achieved in a cost effective manner.

It is clearly evident from the discussion that in today's environment deployment of RFID systems is prone to security issues and *data tampering* is one key issue, which needs to be addressed immediately for secure and reliable deployment of this technology. Lukas Grunwald (2005) showed how vulnerable RFID tags can be, when he used a small program called *RFDump* to show, how the tags could be read, altered or even deleted using an inexpensive tag reader which can be plugged into a notebook [6, 8]. This small software showed how anyone could tamper the RFID data easily. In this paper we address the problem of data tampering in RFID tags, and present a solution based on the concepts of information hiding and fragile watermarking.

The paper is organized as follows. In Section 2, we survey the existing literature on RFID security. In Section 3, we formalize the problem description. In Section 4, we propose the *tamper detection* solution, termed as *TamDetect*. In Section 5, we provide a discussion and conclude the paper in Section 6.

II. Literature Review

While researchers are just starting to address security questions, privacy advocates and legislators have for some time been attempting to address the privacy issues. A lot of work has been done to address the privacy issues in RFID deployment, however literature addressing the security issues is quite limited. The main aim of this section is to discuss the security issues in RFID systems and survey the relevant literature that is proposed to address the same.

Wong and Raphael (2006) classify attacks on RFID systems into two categories – passive attacks and active attacks. *Passive attackers* are those who eavesdrop on the communications channel, but do not affect or interfere with the communication in any way [24]. Passive attacks compromise the *confidentiality* and *anonymity* in communication. Consider the warehouse management scenario, if a malicious reader can eavesdrop (spy) the communication between the tags and the readers, *confidentiality* and *anonymity* in such communication is lost because the entity involved in the communication is unaware when it is being attacked.

Active attackers are those who directly interfere with the communication of messages, either by interrupting, fabricating or modifying communicated messages [24]. Active attacks compromise the *availability*, *authenticity* and *integrity* in communication. *Interruptions* refer to denial of service attacks

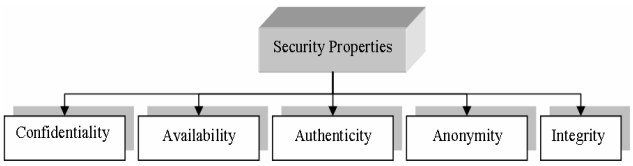

Fig. 2 Major Security Issues with RFID Adoption

(*availability*) on RFID tags. Engberg, Harning, and Jensen (2004), argued that if such an attack is launched, the RFID reader cannot query the tags which could have an adverse effect on a warehouse management system (or related applications) as it may stop responding and real-time status of the warehouse cannot be made available [4].

Fabrication refers to attacks on the *authenticity* of the information on the RFID tag, such as tag forgery. RFID tags might be forged in order to get access to restricted locations within an organization.

Modifications refer to attacks on the *integrity* of the information on the RFID tag (or system) such as data tampering. Data on the RFID tag could also be tampered with by malicious readers. Consider the warehouse scenario once again: if the data on the tag is tampered with, it could result in shipping wrong items from the warehouse. For instance, if the malicious reader changes the information on RFID tag from Orange to Apple, then a palette containing Apples might be shipped when the intention was to ship Oranges. Data tampering (or integrity) can raise issues like QoS (Quality of Service) and Trust in logistics and supply chain and hence needs to be addressed thoroughly.

We now discuss the current literature which addresses some of the security issues highlighted in the Fig. 2. Most of the proposed solutions discussed here address the first four security properties i.e. confidentiality, availability, authenticity and anonymity. We begin the discussion with solutions to manage anonymity.

A. Anonymity

RFID technology shows the characteristics that can invade personal privacy; hence anonymity is highly desired if this technology would be deployed in mass scale. A lot of work has already been conducted in this area and several proposals are put forward to address the issue of privacy [9, 11, 12, 13, 14, 17, 20, 21, 22, 23, 24]. In this section, we discuss several approaches that can be used to provide consumer privacy.

One of the simplest approaches to address the issue of privacy is to kill the tag once it leaves the supply chain and enters the consumer market. This approach is used by EPC standard which make the tags permanently inoperative. It is envisioned that the point-of-sale (POS) operator would have RFID reader that can send the command to kill the tag once it is sold to the consumer. However, to address the issue of malicious tag writes, the kill command is protected by a secret PIN which in this case is assumed to be with the POS RFID reader. Another approach is to add a RFID tag on the price tag. Hence, when the price tag is removed, the RFID is removed as

well and can guarantee privacy. However, as pointed out by Juels (2005), removing or killing the tags can restrict the post purchase benefits of RFID tags like receiptless item returns [11]. As a result, it would be useful if the tags could be temporarily deactivated. This could be achieved by access control mechanisms similar like using a PIN. Several other approaches to anonymity and privacy are outlined in Table 1.

TABLE 1
ANONYMITY (PRIVACY)

Proposal	Approach
Inoue & Yasuura 2003	Using two tags – one for unique identification and other for product details. Does not address clandestine inventorying or tracking.
Juels and Pappu 2003	Re-encrypting the tag content using El-Gamal cryptosystem. The solution is presented in the context of securing RFID enabled banknotes.
Juels, Rivest & Szydlo 2003	Blocker Tags: A tag that specifies whether it can be read or not. A privacy bit (0 or 1) is assigned on the tag which determines whether the tag can be publicly scanned (bit 0) or can be used privately (bit 1).
Ateniese, Camenisch & de Medeiros 2005	This solution is based on using bilinear pairing in elliptic curve cryptography. Authenticity of the tag identifier is maintained by digitally signing the ciphertext with a trusted CA. This approach cannot address the issue of ciphertext swapping, i.e. when eavesdropper changes the content of two RFID tags simultaneously by swapping their content.
Rakesh Kumar	A Faraday cage is an enclosure designed to exclude electromagnetic fields. As a result, certain radio frequencies cannot penetrate through it. It can address privacy concerns, e.g. if high values currency notes start embedding a RFID tag, then using foil lined wallets can guarantee privacy

B. *Confidentiality*

Several approaches to access control are proposed in the literature. We will discuss a few of the approaches in greater detail in this section.

Juels, Rivest and Szydlo (2003), discuss a hash based Access Control Protocol [13]. Here the tag is first in a *locked state*. When the tag moves to the *unlocked state* the reader can access the tags details. In order to change the state the tag first transmits Meta ID' which is the hash value of a key. An authorized reader looks up the corresponding key in a backend system and sends it to the tag. The tag verifies the key by hashing it, returns the clear text ID, and remains only for a short time in an 'unlocked' state which provides time for reader authentication and offers a modest level of access security.

C. *Authenticity*

The literature on the authentication of the RFID tag is also very mature as of today. Several proposals are presented in the domain of tag authentication, reader authentication, and anti-counterfeit tag. Some of these approaches are outlined in Table 2.

This concludes the survey of the most relevant literature on RFID security. We observed that most of the solutions addressed the issue of authentication, confidentiality and anonymity. Existing solutions do not address the issue of data integrity of the RFID tag in detail.

TABLE 2
AUTHENTICITY

Proposal	**Approach**
Juels 2005	PIN: Authenticate the tag to the reader
Juels 2004	Yoking Proofs – provides cryptographic proofs that two tags were scanned simultaneously and in physical proximity. Can be used in a pharmacy to prove to a government agency that the pharmacy scanned a RFID tagged medicine bottle and delivered the exact medicine as prescribed on the RFID tagged prescription
Engberg et al. (2004)	Zero-knowledge based protocols for communication between reader and tag so that they can authenticate each other without revealing any secrets that may allow them to be tracked.
Molnar & Wagner, 2004	Mutual authentication schemes using challenge-response based on the use of pseudo-random function in the computation of responses to challenges.
Feldhofer et al., 2004	Proposes the Simple Authentication and Security Layer (SASL) protocol with AES encryption and analyses the hardware requirements
Dimitriou, 2005	Provides forward secrecy by using nonces (random numbers that are never reused) by both the reader and tag in their challenges to each other.

III. PROBLEM DESCRIPTION

RFID tag carries data which represent unique item identifiers as well as product details to which it is attached. This data is very significant and if this is tampered with, it can have severe consequences. For example, if data representing the "*nature of good*" is changed, it can have severe implications. For instance, instead of *Lethal Weapons* the RFID could be tampered to represent that the consignment carries *Oranges*. Such data tampering needs to be detected as it can be a threat to national security.

Data tampering of this nature can raise issues in collaborative environments where this data mismatch can result in repudiation issues. For example, in distributed logistics networks and extended enterprises, collaborating peers could accuse each other for being vulnerable to security attacks, which may reduce their trustworthiness. This shows the need to address the issue of data tampering.

Normally, message authentication codes (MAC) are used for integrity check; however, in the RFID tag this is not possible (except the ISO 14443 standard) because of limited resources. Hence, *checksums* are often implemented to check the integrity of the information on these tags.

In the previous section, we conducted an in-depth literature survey of RFID security solutions and we identified that no-one has yet presented a solution to address the issues that we have highlighted here. This gives us the rationale to present our solution for tamper detection for data integrity management.

IV. PROPOSED TAMPER DETECTION SCHEME - TAMDETECT

In this section, we give a general overview of TamDetect solution. Based on the limitations of the security solutions outlined in Section 2 and 3, we then elicit the main

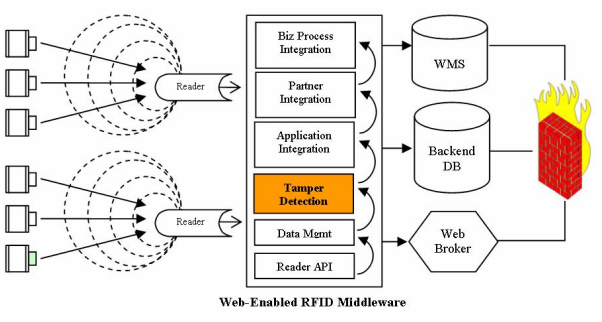

Fig. 3 RFID Middleware Architecture with Tamper Detection Facility

requirements for TamDetect, followed by the design rationale where we discuss the basic design decisions for TamDetect.

A. General Overview of the TamDetect Solution

TamDetect offers a fragile watermarking solution to address the issue of *data tampering*. The proposed solution can ascertain that data tampering has occurred and it can also identify the portion of the RFID tag that has been tampered with. To achieve this, TamDetect embeds a *fragile watermark*[1] in the serial number partition of the RFID tag. RFID tag data structure is composed of Header, EPC Manger (EM), Object Class (OC) and Serial Number (SN)[2]. This embedded watermark or secret pattern is used by the TamDetect component to identify data tampering. This is done in the following manner: suppose the secret pattern is generated using the data stored in EM and OC and then embedded in the SN partition. Now if the data on EM or OC is tampered with, it would introduce an inconsistency between the embedded pattern and the pattern that would be generated by the data from EM and/or OC. This inconsistency is detected by TamDetect and is used to identify data tampering. The detailed algorithm is explained later.

The functionality for embedding of the secret pattern is assumed to be present in the RFID reader which initially writes the tags, whereas the detection algorithm is assumed to be available as a component which can be plugged in the RFID middleware applications. The TamDetect component which would be a part of the RFID middleware is shown in Fig. 3. This component takes input data from the data management layer, and then detects whether any data tampering has occurred. If data tampering has happened, then appropriate measures can be taken to prevent such data from entering the application integration levels in the middleware architecture. The most likely source of data tampering would be a malicious RFID reader who can access and modify the contents of the RFID tag over the entire wireless communication network.

[1] A fragile watermark is a pattern which is hidden in a digital object, which is easily lost, if the object is tampered.

[2] The EM is used to identify manufacturer uniquely, whereas the OC is used to identify the product, manufactured by the manufacturer, finally the SN as the name suggests identified each unique item belonging to one product.

In order to address the issues of data tampering, the following requirements are laid for the proposed TamDetect solution.

1. *Size of Fragile Watermark*: The watermark hidden in the RFID tag should not occupy a lot of space because the amount of data that can be stored on the tag is very limited.
2. *Watermark Generation*: The inputs for generating the watermark should be available on the tag itself.
3. *Embedding Locations*: The fragile watermark should be embedded in the serial number partition.
4. *Tamper Detection*: The algorithm should be able to detect that data tampering has occurred on the RFID tag.
5. *Localization of Tampering*: The algorithms should be able to identify the portion on the RFID tag that has been tampered with, so that it can be rectified.
6. *Plug-n-Play Architecture*: The proposed solution should be designed such that it can be easily plugged into the current RFID middleware applications.

The theoretical foundation for TamDetect is proposed to satisfy the requirements outlined above. The following design decisions are proposed in this solution.

1. The size of the watermark is limited to *eight bits* which represents 25% of the available redundant space on the RFID tag. (Req. 1)
2. The watermark is a *hash* generated from the data stored in the EM and OC partitions. (Req. 2)
3. The watermark is embedded in the *serial number partition* because it offers enough space (36 bits), which can be used for embedding the watermark. (Req. 3)
4. A one-way hash function is used to generate an 8 bit string which is embedded in the serial number partition of the RFID tag. This hash value is checked to detect tampering. (Req. 4)
5. The inputs to the hash function come from the OC and EM. Hence, if the hash is not matched, we can identify the section of the RFID tag that has been tampered with. (Req. 5)
6. The algorithm is designed as a component; hence, it can be easily plugged into any existing middleware application. (Req. 6)

We now discuss the theoretical foundation for TamDetect.

B. Theoretical Foundation for TamDetect

The proposed framework is shown in Fig. 4, 5 and 6. It can be decomposed in four different stages:

1. Watermark Generation
2. Selecting the Embedding Location
3. Watermark Embedding
4. Tamper Detection

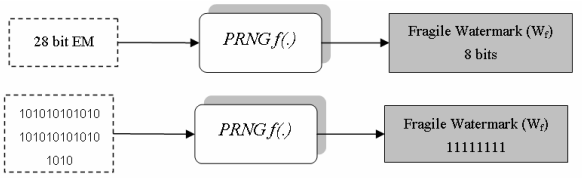

Fig. 4 Process of Generating Fragile Watermark

Step 1: Watermark Generation

Inputs	*EPC Manager (EM), or Object Class (OC), Hash Function f(.)*
Outputs	*Fragile Watermark (W_f)*

Generate Watermark

The watermark is generated by using a pseudo random number generator (PRNG). The PRGN acts as a hash function in this case. The input for the hash function is the data that represents the EM or OC. EM or OC acts as a seed that generates a unique random number of a desired length. This unique number (in binary format) is used as a fragile watermark. This fragile watermark is then embedded in the serial number partition of the RFID tag. The process of generating the fragile watermark is shown in Fig. 4. The 28-bit binary data from EM acts as a seed for the PRNG that generates an 8-bit random number which is used as a fragile watermark.

Once the watermark has been generated, we have to identify the location for embedding. We now discuss how we select the appropriate location for embedding the watermark.

Step 2: Selection the Embedding Location

Previously we mentioned that the fragile watermark (W_f) is embedded in the serial number partition of the RFID tag. In this section, we give the reason for this selection.

The basic principle of watermarking (or information hiding) is that we need some redundant space within the host signal which can be modified to embed the watermark. In this case, the RFID tag is the host signal and we want to identify the redundant space. In order to do this, we investigated the RFID data structure.

On the basis of that investigation, we determined that the serial number partition within the RFID tags can offer a reasonable amount of redundant space for embedding the fragile watermark. This selection is attributed to the following facts:

The *Header*, is fully used for identifying the EAN.UCC key and the partitioning scheme. Hence, there is no redundant space, so there is no possibility for embedding the fragile watermark.

The *EPC Manager*, is used to identify the manufacturer uniquely. Hence, this partition also does not offer any redundant space for embedding because it might be decided by the industry standard and the manufacturer has least control over this.

The *Object Class*, is used to identify the product manufactured by the manufacturer. It may follow some product convention taxonomy where the *first* two digits might represent the classification of that product, the next two may be the age of product and so on. Hence, modifying any of this data might interfere with the existing industry standard. As a result, this partition also does not offer enough room for embedding the watermark.

The *Serial Number*, which is the last partition, is used to uniquely identify an item which belongs to a particular Object Class. It is orthogonal to the *first* three partitions and can be decided by the manufacturer at will, without violating any existing industry standards. Consequently, it offers enough redundant space to embed the watermark. Meanwhile, the length of this partition is 36 bits (in EPC96) which offers enough room to accommodate the fragile watermark. Thus, this becomes the most appropriate candidate for embedding the watermark, and hence, we decided to choose this partition to embed the watermark. We now discuss the embedding and extraction algorithm in detail.

Step 3: Watermark Embedding Algorithm

Inputs	*Serial Number (SN)* *Fragile Watermark (W_f)* *Embedding Location (L)*
Outputs	*Watermarked RFID Tag (W)*

Step 1: Access the Watermark

In the *first* step, the RFID reader accesses the watermark. The watermark W_f may be generated by the RFID reader itself or by the RFID middleware. We assume that the reader has the functionality to generate the watermark

Step 2: Select the embedding location within the serial number partition

The SN partition has 36 bits; we select the *first n+1 ($0<n<36$)* bits to embed the watermark, where *n* is the size of the fragile watermark. We express this location in the SN as *L*. The extra bit is used as a parity bit to check whether the watermark bits have changed after embedding.

Step 3: Append Parity Bit

In this step we append an even parity bit to the fragile watermark i.e. if the watermark is 8 bits long (11111111) then we append a 0 to make it 111111110.

Step 4: Embedding the Watermark

The 9 bits watermark is now embedded in the SN partition of the RFID tag. The process of embedding the watermark is shown in Fig. 5.

Fig. 5a shows the generic model of how the embedding process occurs, whereas Fig. 5b shows an example of how an 8-bit watermark (11111111) appended with a even parity bit (0) is embedded in the SN partition of the RFID tag. The fragile watermark is added to the *first* eight bits of the SN. It is shown as bold and underlined. The parity bit is shown next to it. So the final SN looks like **11111110**. The advantage of

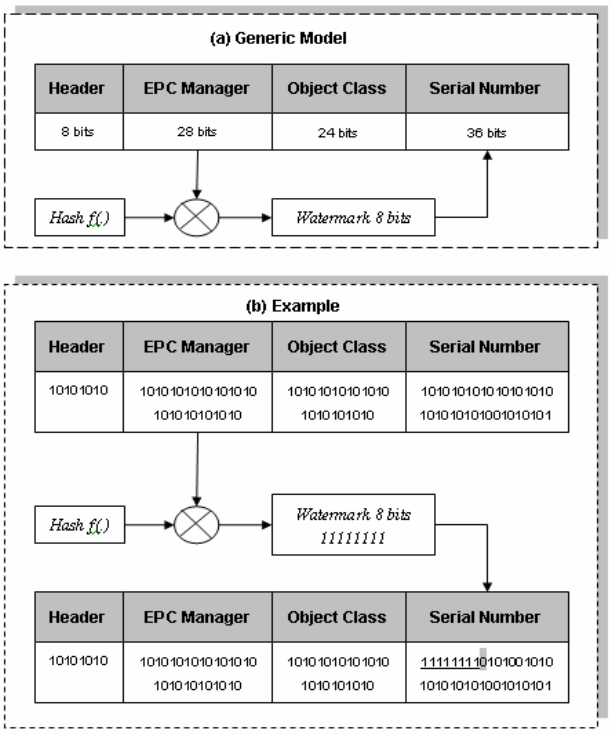

Fig. 5 Process of Watermark Embedding

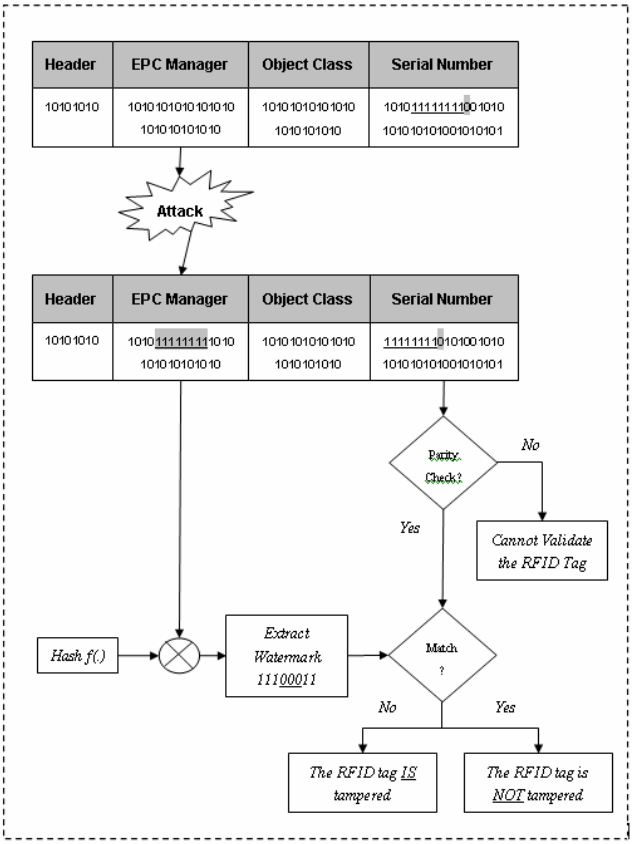

Fig. 6 Process of Tamper Detection

having the watermark in the beginning is that it would be common to all the tags and would facilitate easy numbering. We now discuss the tamper detection algorithm.

Step 4: Watermark Extraction & Tamper Detection Algorithm

Inputs	*Serial Number (SN), Embedding Location (L), Hash Function f(.)*
Outputs	*Verification if the tag is tampered or not*
Step 1: Generate Watermark Generate the watermark ($W_{f\sim}$) once again using the PRNG *f(.)* and seed *EM*. Since *EM* is used as a seed in *f(.)* the resulting watermark W_f would be the same as it was when it was embedded, unless EM has been tampered with.	
Step 2: Extract the watermark from the serial number partition Extract the *first* 9 bits (W_{fe}) from the SN partition. Check for even parity. If even parity exists, then proceed to the next step. If not, then exit.	
Step 3: Tamper Detection In this step, compare the extracted watermark (W_{fe}) with the one generated in Step 1 ($W_{f\sim}$). Do the watermarks match? IF yes → NO Data Tampering observed IF no → Data on the RFID tag is tampered.	

The process of tamper detection is shown in Fig. 6. We show that the watermarked RFID tag has been attacked and the content of the EM has changed. When the tamper detection is performed, the data stored on EM is used to generate the watermark using the hash function *f(.)*. Since the EM has changed, the watermark that is generated by *f(.)* would be different from the one that is embedded in the SN partition, and hence, tampering can be easily identified.

If, however, the SN is tampered with, then it would be difficult to identify. In this case, we assume that the attacker has no incentive to change the SN. However, the mere fact that there is an inconsistency between the EM and SN indicates data tampering.

V. Discussion and Validation

In this paper, we proposed a tamper detection solution to identify data tampering in RFID tags. The detection is achieved by embedding an 8-bit fragile watermark in the SN partition of the RFID tag. We showed how we can embed a fragile watermark that represents the EM, so if EM is tampered with, we can easily identify it. If we want to detect tampering of OC, the same approach can be further extended as well. However, the drawback would be reduced unique serial numbers. If we just use nine bits for embedding (i.e. EM only) we only consume 25% space and can uniquely identify 134,217,728 items. If we use 18 bits to detect EM and OC, we use 50% space. Even with the remaining 18 bits we can still uniquely identify 262,144 items. This number is still acceptable if we consider palette level tracking. But at item level tagging, it might be unfeasible, in which case we can reduce the size of the watermark to 5 or 6 bits. However, if the size is too small it would reduce the security of the system. Hence, there is a

tradeoff. But this is considered to be acceptable in a closed RFID system where the RFID tag data is shared amongst trading partners.

The tamper detection technique that we presented is useful is identifying whether data tampering has happened and where the data has been tampered with. We would like to emphasize that this is not a tamper proof solution: we cannot protect the RFID tags from being tampered with. However, if the RFID tag is tampered with, we can prove that tampering has happened and at the same time we can show the data that has been tampered with.

We assume that in a normal scenario, the most likely location where tampering would happen is the EM or the OC. This is because we assume that the main motivation behind tampering would to disguise one product as another (for e.g. cheaper shipping cost or other economic benefits) and this can be done only if the details with EM or OC are modified, but not the Serial Number. As a result, we embed the fragile watermark in the serial number, and hence, we can identify whether or not EM or OC have been tampered with.

As long as the serial number has not been tampered with, the proposed technique can exactly indicate whether EM or OC has been modified. To achieve robustness against serial number tampering, the fragile watermark has to be embedded in some other read-write segment of the tag's memory. The only issue with serial number tampering is that we cannot localize tampering, but we can still detect it.

To improve the security of the system, we can provide an additional number of parity checks or we can perform error correction coding. However, the problem with this approach is limited space. Hence, we decided to have just one parity bit.

We also assumed that the attacker has access in order to change the RFID tag; that is, either the tag is not write-protected, or the access control is switched so it can be bypassed. We can also assume that an insider wants to change the RFID tag so that s/he can steal products from the premises. In this case, the access control can be intentionally deactivated for a while and the attack can be launched.

VI. Conclusion And Future Work

In this paper, we proposed a tamper detection solution to address the issue of data tampering in the RFID tags. We found the majority of recent research work in RFID security has been done in the areas of anonymity, confidentiality and authenticity. Data integrity has not been tackled in detail. Hence, we propose a tamper detection solution by introducing a flexible layer into existing RFID middleware architecture. We also provided a detailed description of the tamper detection algorithm, which can detect and identify whether and what data (i.e. EM or OC) on the RFID tag has been tampered with.

In the future we would like to enhance the embedding algorithm by using Code Division Multiple Access (CDMA) scheme. This would add more randomness to the embedded fragile watermark and strengthen the solution against active attacks.

References

[1] Giuseppe Ateniese, Jan Camenisch, and Breno de Medeiros. Untraceable RFID tags via insubvertible encryption. November 2005.

[2] Caspian: "Scandal: Wal-Mart, P&G involved in secret RFID testing,"Nov 10, 2003

[3] Tassos Dimitriou. A Lightweight RFID Protocol to protect against Traceability and Cloning attacks, in Proceedings of the First International Conference on Security and Privacy for Emerging Areas in Communications Networks (SECURECOMM'05).

[4] Stephan Engberg, Morten Harning, and Christian Damsgaard Jensen. Zero knowledge device authentication: Privacy & security enhanced RFID preserving business value and consumer convenience. October 2004.

[5] Martin Feldhofer, Sandra Dominikus, and Johannes Wolkerstorfer. Strong authentication for RFID systems using the AES algorithm. 3156:357–370, August 2004.

[6] Lukas Grunwald, "RFDump Can Hack RFID Tags" , Available online: http://www.rfidgazette.org/2004/07/lukas_grunwalds.html Accessed on Sunday, 29 October 2006

[7] Dirk Henrici and Paul M¨uller. Hash-based enhancement of location privacy for radio-frequency identification devices using varying identifiers. pages 149–153, March 2004.

[8] G. V. Hulme, T. Claburn, "RFID's Security Challenge- Security and its high cost appears to be the next hurdle in the widespread adoption of RFID", in InformationWeek, Nov. 15, 2004 URL: http://www.informationweek.com/story/showArticle.jhtml?articleID=52601030

[9] Sozo Inoue and Hiroto Yasuura. RFID privacy using user-controllable uniqueness. November 2003.

[10] Ari Juels. "yoking-proofs" for RFID tags. pages 138–143, March 2004.

[11] Ari Juels. RFID security and privacy: A research survey. Manuscript, September 2005.

[12] Ari Juels and Ravikanth Pappu. Squealing euros: Privacy protection in RFID-enabled banknotes. 2742:103–121, January 2003.

[13] Ari Juels, Ronald L. Rivest, and Michael Szydlo. The blocker tag: Selective blocking of RFID tags for consumer privacy. 8th ACM Conference on Computer and Communications Security, pages 103–111, 2003.

[14] Heiko Knospe and Hartmut Pohl. RFID security. Information Security Technical Report, 9(4):39–50, November–December 2004.

[15] Rakesh Kumar. Interaction of RFID technology and public policy, Wipro White Paper, November 2003.

[16] Hennig, J. E., Ladkin, P. B., Siker, B., "Privacy Enhancing Technology Concepts for RFID Technology Scrutinized" 2005

[17] David Molnar and David Wagner. Privacy and security in library RFID: Issues, practices, and architectures. pages 210–219, October 2004.

[18] D. Molnar and D. Wagner, "Privacy and security in library RFID: Issues, practices, and architectures" In Conference on Computer and Communications Security – CCS, ACM Press, 2004 pp. 210-219

[19] RFIDExchange http://www.rfidexchange.com/applications.aspx

[20] Junichiro Saito, Jae-Cheol Ryou, and Kouichi Sakurai. Enhancing privacy of universal re-encryption scheme for RFID tags. 3207:879– 890, August 2004.

[21] William Stallings. Cryptography and Network Security. Prentice-Hall, Inc., 1999.

[22] Stephen Weis. Security and privacy in radio-frequency identification devices. Master thesis, Massachusetts Institute of Technology (MIT), Massachusetts, USA, May 2003.

[23] S. A. Weis, S. E. Sarma, R. L. Rivest, D. W. Engels, "Security and Privacy Aspects of Low-cost Radio Frequency Identification Systems", in D. Hutter et al. Edn. Security in Pervasive Computing 2003, LNCS 2802, pp. 201-212, 2004

[24] Dennis M.-L. Wong and Raphael C.-W. Phan. RFID systems: Applications versus security & privacy implications, to be published by IDEA group, 2006.