International Journal of Engineering & Technology
CENTRE OF PROFESSIONAL RESEARCH PUBLICATIONS
International Journal of Engineering and Technology Volume 1 No. 1, October, 2011

TAMPER DETECTION FOR LOW COST RFID TAGS: USING WATERMARKING WITH CHAOTIC MAPPING

Kevin Curran, Tom Lunney, Ali N M Noman
Faculty of Computing & Engineering, University of Ulster, Londonderry, UK.
kj.curran@ulster.ac.uk

ABSTRACT

Low cost RFID tags pose unique security and privacy challenges. There is an unresolved issue however of data tampering. Here we propose a watermarking based tamper detection solution for low cost RFID tags where the Object Class (OC) and the Serial number (SN) field of an RFID tag are used as the cover medium. Additionally, the use of a chaotic map (i.e. Skew Tent Map) in the watermark embedding algorithm makes this solution more secure. Unlike existing watermark based solutions, our proposed solution offers increased security and can detect tampering anywhere in the RFID tag (e.g. both in the EPC Manager and the Object Class field of an RFID tag), not just a portion such as the Object Class or the EPC manager. This proposed solution conforms to the EPC-Class1 Generation2 specification.

Key words: watermarking; RFID security; tamper detection; EPC Class1 Gen 2 tag; information hiding.

1. INTRODUCTION

There has been a notable increase in the deployment of RFID (Radio Frequency Identification) tags in supply chain management and access control applications [1, 2, 3]. The standardization bodies such as the EPCglobal and the GS1 (Global Standardization Body) are working together to ensure the proliferation of RFID applications by proposing and managing a global standard for RFID tags. The latest standard of the EPCglobal, EPC Class1 Gen2 (also known as EPC-C1G2 or Gen 2 tag) includes a complete specifications for the 96 bit EPC (Electronic Product Code) tag [5]. An EPC Gen2 tag has four fields: the 8 bit Header, H (a fixed identifier for the EPC tag); the 28 bit EPC Manager, EM (which represents a company or an organization); the 24 bit Object Class, OC (which represents unique type of products within a company); the 36 bit Serial Number, SN (which represents a unique item within a particular product/ OC). Fig 1 shows the structure of a Gen2 tag.

Header (H) 8 bit	EPC Manager(EM) 28 bit	Object Class (OC) 24 bit	Serial Number(SN) 36 bit
• Not used is watermark generation and embedding process	• 6 bit Watermark (W_em) is created from the EM	• 6 bit Watermark (W_oc) is created from the OC • Watermark (W_em) is embedded here	• Watermark (W_oc) is embedded here

Fig. 1 Structure of a Gen 2 tag- a 96 bit EPC. It also shows which fields are manipulated in the proposed tamper detection solution.

Within the security umbrella of RFID, there are several open issues and data tampering is one such which needs addressing. If the data stored on the tag is tampered, then the tag becomes useless. Unfortunately it is relatively easy to tamper with the data stored on the RFID tag. Grunwald showed how vulnerable RFIDs 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 [6]. A robust and EPC-C1G2 standard compatible tamper detection solution is required for low cost RFID tags to ensure the integrity of RFID applications. However traditional security solutions based on public key primitives (or even symmetric key primitives) are not feasible for resource limited RFID tags: those solutions ask for either additional memory or additional computational capabilities from the low cost RFID tags. For instance, current RFID tags costing below $0.5 offer up to 2000 logic gates for security purposes where ECC, AES, and SHA-1 require 15000, 3400, 4300 logic gates respectively [2, 7]. Furthermore most of the existing tamper detection solutions [8, 10, 11] for the RFID tags do not conform to the EPC-C1G2 specification. For instance, Yamamoto, et al., proposes a very promising tamper detection solution [10] based on a technique known as digitally signed journal [9] but it requires modification (in memory) in the existing EPC-C1G2 tags. In addition, Hitachi Secure Tag [11] offers a tamper detection solution but the drawback of this solution is that here a reader or writer must implement a modified version of the air protocol. Watermarking is the art of embedding data directly within content which is imperceptible to humans but readable by computers. We believe a watermarking based technique has the potential to effectively address the data tampering issue of existing low cost RFID tags as it can be deployed without asking for any modification in the existing tags. In [12, 13, 14], watermarking based tamper detection solutions for RFID tags are proposed where the Serial Number field of a RFID tag is used as the cover medium to detect any modification occurring either in the EPC Manager (EM) or in the Object Class (OC), but not for both at the same time. In this paper, we propose a unique watermark based tamper detection

solution for the EPC-C1G2 tags using the Object Class (OC) and the Serial Number (SN) field of an RFID tag as the cover medium. Our intention is to propose a tamper detection solution which conforms to the EPC-C1G2 specification. it should not require any additional capabilities (e.g. battery, or computational) from the existing RFID Tags and it should be easily deployed in any RFID application. It should also offer increased security so that it will be very difficult, if not impossible for an attacker (in this case an illegitimate reader) to tamper a tag without being detected. This paper is organized as follows. We discuss our proposed solution in detail in section 2. In section 3, we perform an extensive analysis of our proposed solution which includes implementation issues, security of the proposed solution and issues regarding compatibility towards the EPC-C1G2 standard. We draw our conclusion in section 4.

2. PROPOSED SOLUTION

Among the four fields of an EPC tag, the H and the EM fields are assigned by the EPCGlobal and hence those fields cannot be used as the cover medium for embedding the watermark. However an EPC manager has had the flexibility to assign both the OC and the SN fields [5] and that is why, we use the OC and the SN field of an RFID tag as the cover medium for embedding the watermark. Only 6 bit each from the OC and the SN fields are used for keeping the watermark and practically the remaining spaces of the OC (18 bits) and the SN (30 bits) field are suitable for almost all RFID applications as they can uniquely address up to 2^{18} types of product having up to 2^{30} unique instances of each product. Figure 1 show which fields are used for watermark generation and watermark embedding.

Assumption: Solution is based on following assumptions:

- Only the EM and the OC field of an RFID tag would be tampered (considering the financial benefit and attackers' motivation). Any tampering attempt in the H field will make the entire tag invalid as it represents the uniqueness of the EPC tag. Additionally tampering the SN field does not offer any kind of benefit to an attacker.
- Only legitimate readers and the tag itself, have all the secret keys required to execute the watermark generation, embedding and extraction algorithms.
- Watermark embedding has been done in system initialization phase by a legitimate reader/ Middleware RFID applications. Additionally a legitimate RFID Reader also executes the watermark extraction and tamper detection procedure.

Notation: We use the following notations throughout:

H: 8 bit Header field of EPC

EM: 28 bit EPC Manager field of EPC
OC: 24 bit Object Class field of EPC.
SN: 36 bit Serial Number field of EPC
OC_18, SN_30: Effective portion of the OC and the SN field that carries actual data (after keeping 6 bits for watermarking)
W_em.W_em_temp: 6 bit watermark for the EM field.
W_oc, W_oc_temp: 6 bit watermark for the OC field
EM_32,OC_32, : 32 bit EM, OC (after padding). Used them in watermark generation process.
Pad_F: A function which makes an input (which is less than 32 bit) up to 32 bits long using padding. We use this function to generate EM_32 and OC_32.
H_F: A function (Pseudo Random Number Generator) which acts like a hash function which takes 32 bit as input and produces an 8 bit output.

A. Watermark Generation

Figure 2 shows the procedure for generating watermark for both the EPC Manager and OC field of an RFID tag.

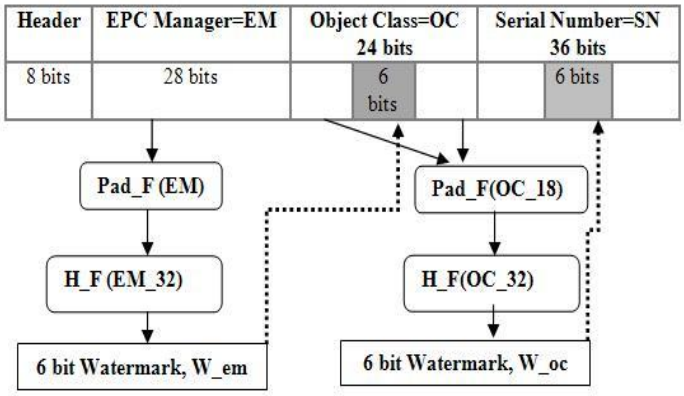

Fig. 2: Watermark generation and embedding procedure. Note that watermark location has to be found out prior to embed the watermark.

Watermark Generation Algorithm: includes the following 3 steps:

Step 1: *Identify the 28 bit long EM and the 18 bit long effective OC field (OC_18) as input for watermark generation*. A legitimate reader can directly use the output of the watermark embedding algorithm (executed during system initialization phase) or it can execute by itself up to step 3 of the watermark embedding algorithm (describe in the following subsection) for the effective OC field (OC_18).

Step 2: *Make the 28 bit effective EM and the 18 bit OC (OC_18) into 32 bits using a padding function, Pad_F.*

***Pad_F function*:** [no secret key required]
Begin

a. *Calculate the number of bits for padding (j).*
   Example: In the case of EM [0 to 27], it is 4(j=32-28=4).
b. *Use j number of MSBs (Most Significant Bits) as the pad. For example, Pad [0 to 3]:=EM [24 to 27];*
c. *Append the Pad with the Least Significant Bit (LSB) portion of the input.*
   Example: EM_32 [0 to 3]:= Pad [0 to 3];
   EM_32 [4 to 31]:= EM [0 to 27];
d. *Thus a 32 bit output is produced.*
   Example: EM_32 [0 to 31] and
   (Similarly) OC [0 to 31].

End

Step 3: *Generate 6 bit watermarks W_em and W_oc from the Em_32 and OC_32 respectively using a hash like PRNG (Pseudo Random Number Generator) function called H_F.*

***H_F Function*:** [secret key required for execution]
Begin

a. *Execute the LAMED-EPC (described below, which accepts a 32 bit input and generate a 16 bit output) function to generate a 16 bit output, Out [0 to 15] from a 32 bit input (in this case, EM_32 and OC_32).*
b. *Do an X-OR operation between Out [8 to 15] and Out [0 to 7] to produce 8 bit output, Out [0 to 7] and then take the first 6 Most significant Bits of it (i.e. Out[2 to 7]) as the final output.*
c. *Thus 6 bit long W_em and W_oc are produced.*

End

LAMED-EPC Function*:** This existing EPC-CIG2 compatible PRNG function is based on a genetic algorithm and made for low cost RFID tags. The seed of this PRNG function consists of an initialization vector, *iv* (which can be public) and a key, *s* (which is only known by legitimate readers and the tag). Whereas the detail description of LAMED-EPC can be found in [15], ***a short summary of the LAMED-EPC function is given below:

Begin

a. *A 32bit output is generated from the following equations:*

$$a_o^{n+1} = \begin{cases} a_1^n + iv \;\; if\ n\ is\ odd \\ a_1^n \oplus iv \;\; if\ n\ is\ odd \end{cases} \quad (1)$$

$$a_1^{n+1} = \begin{cases} O(a_0^n, a_1^n) \oplus s \;\; if\ n\ is\ odd \\ O(a_0^n, a_1^n) + s \;\; if\ n\ is\ even \end{cases} \quad (2)$$

Where n is the number of inner state variables(in this case it is 32), a_0 and a_1 are 32 bit long randomly generated unsigned integers, and O is the Mersenne Twister Generator.

b. *The 32 bit output is then divided in two halves: MSB $_{16\ to\ 31}$ and LSB $_{0\ to\ 15}$.*
c. *A 16 bit output is generated by x-oring MSB $_{16\ to\ 31}$ with LSB $_{0\ to\ 15}$.*

End

B. Watermark Embedding

Identifying the location of the cover medium make it easy for an attacker to break any watermarking based solution. On the other hand, a chaotic map, having the good properties of pseudo-randomness and sensitivity to initial conditions (i.e. a minor change in the initial condition makes a major change in the output, also known as butterfly effect), is suitable for randomly choosing the location of a watermark from the cover medium. In this proposed solution, we use a very simple one dimensional chaotic map, called the Skew Tent map to randomly choose 6 bit positions from the OC (24 bit) field and the SN (36 bit) field of an RFID tag in order to embed the watermark for the EM and the OC field respectively. The skew tent map stated below, generates a sequence which is expansionary everywhere in the interval [0, 1] and distributed uniformly in it. Fig. 3 depicts the sensitivity of this equation on initial conditions (shown in left) as well on the system parameters (shown in right).

$$x_{i+1} = F(\alpha, x_i) = \begin{cases} \frac{x_i}{\alpha}, & x_i = [\![0, \alpha) \\ \frac{1-x_i}{1-\alpha}, & x_i \geq (\alpha, 1] \end{cases} \quad (3)$$

Where α and x_i are system parameter and initial condition respectively.

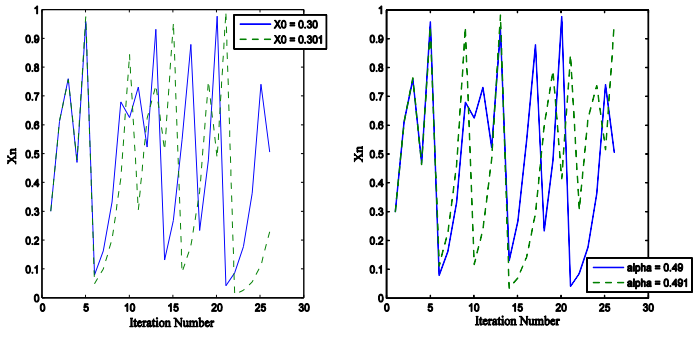

Fig. 3: The sensitivity of the Skew Tent map on the initial condition (left figure) as well on the system parameter (the right one).

Watermark Embedding Algorithm: [secret key required]
It includes the following 4 steps:

Step1: *Calculate the initial condition, x_i by converting the n bit secret key (for password management simplicity, the same key, s can be used) into decimal fractions as follows:*

$$x_i = s_{n-1} * 2^{-1} + s_{n-2} * 2^{-2} + s_{n-3} * 2^{-3} + \ldots + s_0 * 2^{-n} \quad (4)$$

Step 2: *Generate the random vector, $Rand_{vec}$ by feeding x_i (got from eqn. 4) into eqn. (3) with a fixed parameter,α (may be public or private / no problem if it becomes public).*It includes a sequence which is uniformly distributed on [0, n-1] where n is equal to 24 (for the OC) and 36 (for the SN).

$$Rand_{vec} = (n-1) * x_{i+1} \quad (5)$$

Step 3: *Apply the floor function on each element (which are real numbers) of the $Rand_{vec}$ to convert them into the integer format and then select the first 6 distinct elements (i.e. integer values) as the watermark location.* It provides the 6 bit long watermark location in the OC field and the SN field. The remaining bit positions of the OC (i.e. 18 bits long OC_18) and SN (i.e. 30 bits long SN_30) field are used for carrying the actual field data.

Step 4: *Embed the watermarks, W_em and the W_oc into the recently identified 6 bit positions of the OC field and the SN field respectively.*

Note that for reducing the computational complexity/ overhead, the watermark embedding algorithm can be executed only once (to find the watermark location) during the system initialization phase and then the output can be shared among all legitimate readers (or RFID Middleware applications).

C. Watermark Extraction & Tamper Detection

A legitimate RFID reader (or an RFID Middleware application), with knowledge of secret keys, can execute the watermark extraction & tamper detection algorithm.

Watermark Extraction & Tamper Detection Algorithm: [secret key required]

Begin

a. *Identify the watermark locations and then extract the watermarks for the EM (i.e. W_em) and the OC (i.e. W_oc) field of an RFID tag following the watermark embedding algorithm.*
b. *Follow the watermark generation algorithm to regenerate the watermark for the EM and the OC field of an RFID tag and then keep them into W_em_temp and W_oc_temp.*
c. *Use the following tamper detection rules to make a decision about the occurrence of data tampering in the RFID tag.*
    1. *If (W_em != W_em_temp)*
       Tampering occurred in the EM field
    2. *If (W_oc != W_oc_temp)*
       Tampering occurred in the OC field
    3. *If (W_em ==W_em_temp && W_oc==W_oc_temp)*
       No tampering occurred;

End

3. DISCUSSION & ANALYSIS

Apart from the LAMED-EPC function, all other operations (e.g. x-or, padding) used in the watermark generation algorithm, are very cheap and suitable for low cost RFID tags. However LAMED_EPC function meets the EPC-C1G2 specification [15]. It has been formally shown in [15] that LAMED-EPC requires slightly less than 1.6 K gates which is less than half of the memory threshold value (4K) available for security purposes [16]. Furthermore the skew tent map function used in the watermark embedding algorithm can be computed only once during system initialization phase and then shared among all legitimate RFID readers to reduce the computational complexity/overhead. Most importantly in this proposed solution, a legitimate reader takes all the burden from the low cost RFID tag by executing watermark generation and embedding algorithm along with performing the watermark extraction &tamper detection algorithm . Since this solution does not ask for any additional resources (in terms of computation or memory) from the resource limited RFID tags, it is certainly compatible to the EPC-C1G2 specification. Note that this solution is also easily *deployable to any RFID application* as it does not ask for changes in the existing RFID tags and the RFID Readers.

The most practical security challenge for a tamper detection solution of low cost RFID tags is to guard against an illegitimate reader from modifying a portion of the RFID tag without being detected which essentially includes creation of a watermark from the modified tag**.** In this proposed solution, only the legitimate readers with the knowledge of the secret keys can perform the watermark generation and embedding algorithm along with watermark extraction and tamper detection algorithm. Moreover most of the time all algorithms used in this tamper detection solutions can be kept secret among the legitimate readers. However if all algorithms used in this tamper detection solution are considered to be public, a brute force attack has to be launched by an attacker to tamper the tag as it does not have secret keys. The use of chaotic map in the watermark embedding algorithm causes a sharp increase in the search space of a brute force attack [from 2^6 to P (24, 6) * 2^6 in case of the OC field and from 2^6 to P (36, 6) * 2^6 in case of the SN field). To be

more precise, in reality an illegitimate reader has to consider the entire OC and SN field as the search space (i.e. and P (36, 6) * 2^6 respectively) to correctly find out the 6 bit long embedded watermarks from there. It really makes very difficult, if not impossible for an illegitimate reader to generate a new watermark reflecting the changes made in the RFID tag after tampering. Unlike this proposed solution, all the existing watermarking based tamper detection solutions [12, 13, 14] do not use any secret keys and are mainly based on a hash function. Moreover the watermark (8 bit watermark+1 parity bit) is directly embedded into the first Most Significant Bit positions of the SN field. It implies that this existing proposals are not secure enough since a brute force attack requires only 256 (i.e. 2^8) trials whereas if the algorithms are public, an illegitimate reader can easily tamper the tag (no secret key required).

Furthermore all existing watermark based tamper detection solutions rely on a generic hash function (security strength and compatibility towards the EPC-C1G2 specification are not formally proven) where as all the algorithms used in this tamper detection solution are based on verified mathematical properties. For instance, the main pillar of our proposed solution, the LAMED_EPC function satisfies the security requirement of the EPC-C1G2 specification which is: "*a RN16 could not be predicted with probability greater than 0.025% if the outcomes of prior draws from RNG, performed under identical conditions, are known*". A thorough standard security analysis which includes *a serial correlation test, differential analysis and an exhaustive bit and byte prediction tests* of LAMED-EPC are provided in [15]. So it is quite clear that the proposed tamper detection solution has the potential to provide better security than the existing watermarking based solutions. Additionally it could be concluded that the proposed solution meets all criteria that have been earlier as the target for this proposed solution.

4. CONCLUSION

In this paper, we proposed a unique watermarking based tamper detection solution for low cost RFID tags. With regard to the overall tamper detection of the RFID tag, this proposed solution is also capable of locating the portion of the RFID tag where data tampering has occurred. The unique nature of our proposed watermark generation algorithm, the use of chaotic map in the watermark embedding algorithm and the usage of both the OC and the SN field of the RFID tag as the cover medium, make this proposed solution a novel contribution in the field of watermarking based tamper detection solution for RFID tags. We also conducted a thorough analysis of this proposed solution to show that it conforms to the EPC-C1G2 specification and is also suitable for existing low cost RFID tags, and additionally it provides better security than other existing watermarking based tamper detection solutions for RFID tags.

REFERENCES

[1] Juels,A.: RFID Security and Privacy: A Research Survey, An invited paper, IEEE Journal on Selected Areas in Communications, vol. 24 no. 2, pp. 381-394, February 2006.

[2] Spiekermann,S., Evdokimov, S.:Privacy Enhancing Technologies for RFID - A Critical Investigation of State of the Art Research,IEEE Privacy and Security 2009, 2009.

[3] Deursen,T., Radomirovic,T.: Security of an RFID Protocol for Supply Chains, ICEBE '08: Proceedings of the 2008 IEEE International Conference on e-Business Engineering, pp. 568-573, 2008.

[4] RFIDExchange, Accessed on March 11, 2010. http://www.rfidexchange.com/applications.aspx

[5] EPCglobal Inc., "EPC Radio-Frequency Identity Protocols Class-1 Generation-2 UHF RFID Protocol for Communications at 860 MHz -960 MHz Version 1.4", 2008.

[6] Claburn, T., Hulme, G., V.: RFID's Security Challenge- Security and its high cost appears to be the next hurdle in the widespread adoption of RFID. In InformationWeek, Accessed on March11,2010. http://www.informationweek.com/story/showArticle.jhtml?articleID=52601030

[7] Lehtonen, M., et al.: From Identification to Authentication - A Review of RFID Product Authentication Techniques.in Workshop on RFID Security 2006 (RFIDSec 06). 2006. Graz: Springer Verlag.

[8] ISO/IEC, ISO/IEC 18000-6 Part 6:Parameters for air interface communications at 860 MHz to 960 MHz AMENDMENT 1: Extension with Type C and update of Types A and B, 2006.

[9] S. Suzuki and M. Harrison, "Data Synchronization Specification", Auto-ID Labs AEROID-CAM-007, 2006.

[10] Yamamoto, A.; Suzuki, S.; Hada, H.; Mitsugi, J.; Teraoka, F. \& Nakamura, O.A Tamper Detection Method for RFID Tag Data IEEE International Conference on RFID, 2008, 51-57

[11] Hitachi Ltd., Report of Rationalization of energy use in fiscal 2006-Project of RFID tag system Development Survey, 2007.

[12] M. Madan and P. Vidyasagar and C. Elizabeth, "Recovering and Restoring Tampered RFID Data using Steganographic Principles",ICIT 2006. IEEE International Conference on, 2006, 2853-2859.

[13] V. Potdar and E. Chang, "Tamper detection in RFID tags using fragile watermarking," 10th IEEE International Conference on Industrial Technology (ICIT2006), Mumbai, INDIA, Dec. 15–17, 2006.

[14] Potdar V, Wu C, Chang E (2005) Tamper detection for ubiquitous RFID-enabled supply chain. In: Computational intelligence and security—CIS 2005. LNCS,pp. 273–278. Springer, Berlin

[15] Peris-Lopez, Pedro and Hernandez-Castro, Julio Cesar and Estevez-Tapiador, Juan M. and Ribagorda, Arturo:LAMED - A PRNG for EPC Class-1 Generation-2 RFID specification,Comput. Stand. Interfaces. vol. 31,no 1,pp. 88-97,Elsevier Science Publishers B. V.(2009)

[16] D. Ranasinghe, D. Engels, P. Cole, Low-cost RFID systems: Confronting security and privacy, Auto-ID Labs Research Workshop, 2004.

[17] Rieback, M.R., et al. A Platform for RFID Security and Privacy Administration. in 20th Large Installation System Administration Conference (LISA'06). 2006. Washington DC., USA: USENIX The Advanced Computing Systems Association