A Hierarchical Anti-counterfeit Mechanism: Securing the Supply Chain using RFIDs

Zeeshan Bilal and Keith Martin

Information Security Group, Royal Holloway, University of London,
Egham, Surrey, TW20 0EX, UK
Zeeshan.Bilal.2010@live.rhul.ac.uk
Keith.Martin@rhul.ac.uk

Abstract. Counterfeiting is a very serious threat to supply chain management systems. RFID systems are widely used to automate and speed up the process of remotely identifying products, however these systems are vulnerable to counterfeiting. In this paper, we propose a hierarchical anti-counterfeiting mechanism which uses a layered approach to identify dishonest middle parties involved in both counterfeiting and processing stolen/missing items. Our layered approach, which is designed for EPC Class-1 Gen-2 standard compliant tags, offers scalability and is suitable for different sizes of groups of tagged items.

Key words: RFID security, EPC Class-1 Gen-2 standard, Counterfeiting, Supply chain management systems

1 Introduction

Radio Frequency Identification (RFID) systems are extensively used in many applications. In this paper, we discuss their deployment in supply chain management, where an RFID system is capable of identifying products throughout the supply chain process [1]. Such systems have three main components: 1) a *server* (usually centralized), 2) *readers* (from tens to hundreds, depending on the application), and 3) *tags* (potentially millions). A tagged object starts its journey in a large group from manufacturer to customer [2]. During this journey, the object may be read by readers located from the manufacturing company through to retail stores.

RFID technology has replaced barcode mainly because items can be individually identified without line-of-sight requirements [3]. Although RFID systems face similar challenges to those faced by barcode technologies, such as cloning and impersonation, RFID systems have the advantage that they are capable of providing identification as well as authentication. However, counterfeiting, caused by cloning and impersonation attacks, has been a problem for some RFID systems [3]. The counterfeiting of products is one of the most serious threats to modern commerce

according to estimates by the Counterfeiting Intelligence Bureau (CIB) of the International Chamber of Commerce (ICC), which claims that counterfeit goods account for up to 7% of world trade [4]. To address counterfeiting, RFID researchers have designed many schemes which trade-off between cost, security, and performance, however existing approaches all have significant drawbacks which we outline in Section 2.

Since a tag will respond to every query sent by any compatible reader, if no authentication mechanism is employed, an adversary can query a genuine tag and learn the sensitive information associated with the tag's identifier which can then be used to make counterfeit tags. When using authentication, a tag will respond to every query sent by a compatible reader that has been authenticated as legitimate. However, the adversary can still eavesdrop the tag's identifier and then copy this information to a counterfeit tag. So there is a need for *secure identification with authentication* in which case a tag will securely provide its information in response to every query sent by a compatible reader that has been authenticated as legitimate. Although an adversary cannot learn the sensitive information, if this information is static then it can be copied or replayed by counterfeit tags to impersonate genuine tags. Finally the adversary can collude with legitimate but dishonest middle parties to gain benefits.

Considering these threats and capabilities of the adversary, we now propose an anti-counterfeiting mechanism for EPC Class-1 Gen 2 standard [5] compliant tags (*EPC tags*). Our mechanism uses three layers of verification. It is based on the use of shared secrets to generate dynamic verification codes which change in each transaction and can be used to verify groups of tags, as well as individual tags. Our scheme not only provides protection against counterfeiting but also identifies dishonest middle parties. Additionally, it can detect any missing or stolen items and is sufficiently scalable to be applicable to the complete lifecycle of a tagged object within a supply chain management system.

2 Existing Work

There are several existing approaches to managing RFID counterfeiting (see [6,7]). We briefly review some schemes and identify their drawbacks.

2.1 Unique Serial Numbers

Several proposals [8, 9] use unique serial numbers to identify products. These numbers are compared against a database to check legitimacy and

highlight any missing items. However, this technique is detection only and does not prevent counterfeiting since the serial number is transmitted in the clear and any adversary can eavesdrop the serial number in order to clone or impersonate it. If a genuine tag is removed and a counterfeit tag is impersonated as genuine, this scheme cannot detect it.

A number of proposed schemes [8, 10, 11] include a *track and trace* method where a counterfeit or missing item can be tracked down and traced back anywhere in the process. This is done using the complete trail of the exchanges of cloned tag updated by each shipping and receiving record. However, this approach is time consuming and creates bottlenecks if multiple clones are detected at the same time as each cloned tag is individually checked using its complete shipping record from the database. Another drawback of this approach is that a genuine but dishonest retailer can copy a genuine tag and attach this copy to a counterfeit product. They can then sell the counterfeit to a customer, who verifies it to be legitimate using track and trace process, not updated by the retailer or the middle parties [10]. Since this process needs an update by each middle party, therefore it is vulnerable to both intentional and unintentional errors [6].

2.2 Cryptographic Anti-counterfeit Mechanisms

Cryptographic mechanisms can be used to tackle counterfeits. The basic idea is to base authentication on a secret value possessed by each tag, which is then disclosed to the verifier as a proof of authenticity in a challenge-response protocol [12]. This approach may be based on symmetric cryptography or asymmetric cryptography.

If symmetric cryptography is used [13–16], the secret is already known to the verifier who matches it with the secret value received from the tag. To avoid a single point of failure, each tag is given a unique secret key, hence there will potentially be millions of such keys. One approach to establishing all these keys is to distribute all keys to each reader in the form of a local database. However if a reader is compromised then this approach results in the breaking of the whole system. A preferred approach is to store all the keys in an online database which each reader can access. However, this results in extensive communication and computational overheads [17], even higher than the track and trace approach. In addition, the reader needs to be trusted by the supplier since the reader stores or accesses the secret values of the tags in this system.

In contrast, asymmetric cryptography can be used [17–19] to distribute keys. However, this still requires each tag to have a unique secret key and involves considerable computational overheads. Although

researchers have proposed some lightweight public key cryptographic systems, it is still unclear whether such schemes can be deployed in the resource-constrained low-cost RFID systems in supply chain management.

2.3 Unclonable RFID tags

Physical Unclonable Functions (PUFs) are tamper-proof, unclonable items of hardware which produce a unique signature, given an input. In [20] an offline authentication scheme based on physically printed challenge-response pairs from a certain PUF was proposed for tag authentication. However, the printout has to be physically read and cannot easily be automated. Further, it is relatively easy to program a cloned tag to give responses to particular challenges instead of using a PUF. These issues were addressed in another PUF-based scheme [21], but tracking of a tag is possible in this scheme as the PUF identifier is unique and does not change. Moreover, it is infeasible to maintain a large number of challenge-response pairs for one tag, potentially resulting in the few challenge-response entries being eavesdropped and the cloning of the tag.

2.4 Built-in Passwords

Juels has suggested a solution based on the tag's built-in passwords to counter the threat of cloning [22]. The idea is to use the existing *Kill* and *Access* password PINs to perform mutual authentication in order to avoid cloning. The reader sends a set of apparently random values except that one is the correct password PIN. The tag in response has to send the position of the correct PIN to get its legitimacy verified. However, legitimate but dishonest readers can store the complete set of PINs with a tag's responses to clone the tag. Even if the reader is honest, the challenge set of PINs and responses can be eavesdropped. Juels also noted that this scheme is not secure against a simple three-step attack [22] based on skimming a tag identifier, interacting with the reader to obtain the challenge set of PINs, and then using these to obtain the correct PIN.

3 A Hierarchical Anti-counterfeiting Mechanism

We now propose a new approach to prevent counterfeiting in supply chain processes where tags travel in groups. Our mechanism is based on a hierarchical model which involves three layers of verification. The three layers can be considered to range from low to high complexity with respect to trade-offs between cost, performance and level of security. If an

upper layer verification fails, verification drops down to the next layer. We design new first and second layers, and then use the track and trace approach [8] for the third layer.

3.1 Goals

Our mechanism is designed to achieve the following goals:

1. **Anti-cloning.** Protection against copying the data from a genuine tag attached to a legitimate product and cloning it onto another tag attached to a counterfeit (see Section 2.1).
2. **Anti-spoofing.** Protection against replay (impersonation) attacks (see Sections 2.3 and 2.4).
3. **Anti-theft.** Detection of stolen or missing items.
4. **Scalability.** Ability to operate efficiently when tags are in large groups as well as when a tag is attached to a single item.
5. **Compatibility.** Compatible with EPC Class-1,Gen-2 standard tags.
6. **Efficient Key Management.** Supportable using an efficient key management scheme (see Section 2.2).
7. **Good Throughput.** Avoidance of bottlenecks which degrade the overall supply chain system throughput (see Section 2.2).

3.2 The Layered Approach

The three hierarchical layers used for the legitimacy verification of a product (see Figure 1) are:

1. **Group Verification (GV) Layer.** For most of their journey in the supply chain products travel in groups (based on their type, specification, manufacturer and lot number, etc.) Our first layer verifies a complete group. In this layer the reader does not need continuous access to a central repository for verifying each tag because the complete group is read first and then verified as a whole.
2. **Product Verification (PV) Layer.** If GV layer verification fails, product verification is initiated using an individual tag's verification code. This lowers the performance and throughput since the reader has to access the database multiple times. Since the server verifies the legitimacy of a single product, the additional computational overheads are acceptable since the server is anticipated to be powerful. The PV layer identifies individual products that are either counterfeit or missing, their values and complete specifications.

3. **Track and Trace (TT) Layer.** After the PV layer has identified counterfeit or missing products, track and trace is initiated using the complete shipping/receiving record of the product. This gathers important information that includes the location of the anomaly and the type of anomaly (dishonest reader, counterfeit tag, or missing tag).

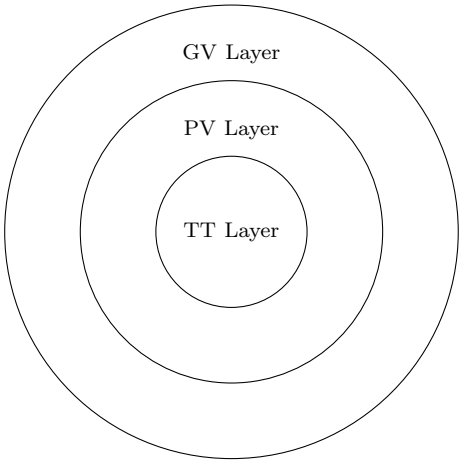

Fig. 1. Hierarchical Verification Model.

Each layer of hierarchical verification detects anomalies in the supply chain in the following order:

1. **GV Anomaly Detection.** GV mainly fails if the reader is not legitimate, a counterfeit is detected, or a tag is completely missing. When GV fails, this will generate an alarm in the server. The server will record the location and details of the reader where the alarm is raised. The server then switches to the PV layer.
2. **PV Anomaly Detection.** PV identifies the exact cause of GV failure. It highlights the exact tagged product which is either counterfeit or missing. The server makes a corresponding entry.
3. **TT Anomaly Detection.** TT is carried out as a last step which recovers the complete shipping/receiving record of the tag that was identified in PV anomaly detection. This further shows whether more clones exist in the supply chain, or whether the original product is completely missing. The server records the details of anomalies.

3.3 Hierarchical Anti-counterfeiting Mechanism

We now explain the detailed operation of our hierarchical anti-counterfeiting mechanism. The notation used is summarized in Table 1.

Table 1. Notation

Notation	Description
S	The server holding the database with shared secrets.
G_j	The group consisting of many tags with identifier j.
HVT_i	The tag employing HV mechanism with identifier i.
ID_i	Tag's (with index i) secret, static and unique identity such as EPC.
R_h	The reader with identifier h scanning groups of tags.
K_{tj}	The group secret key embedded in every tag belonging to group G_j.
KM_r	The master key given to every reader in supply chain system.
KDF	A key derivation function agreed between the server and all readers.
KS_{hj}	The session key derived from master key KM_r by the reader R_h using KDF, currently scanning group G_j.
$rand_j$	Random number generated by server to be sent as challenge for a particular group G_j or tag HVT_j verification.
TVC_i	A tag verification code used to verify the legitimacy of a tag HVT_i.
RVC_h	A reader verification code used to verify the legitimacy of a reader R_h.
GVC_j	A group verification code used to verify the legitimacy of a group G_j of tags.
$EGVC_j$	An encrypted version of group verification code GVC_j.
$T_{timeout}$	The maximum time after a server S sends $rand_j$ to the reader and acquires GVC_j or TVC_j.
L	Length of the secret keys and static identity.
$F : K \times X \to Y$	A lightweight secure PRF such as Hummingbird-2 [23] designed for EPC Class-1 Gen-2 compliant tags.
$E : K \times X \to X$	A secure PRP such as AES defined over (K, X).
$X \oplus Y$	Exclusive-OR of two values X and Y.

Key Distribution Phase. In this phase, the supplier who is responsible for shipping the tagged items in groups (or stand alone as explained in Section 3.1) to different geographic locations holds a database with shared secrets. This database is securely connected to a supplier's or Trusted Third Party's (TTP) server S. The supplier distributes the keys as shown in Figure 2. There are n tags grouped in m groups depending on their

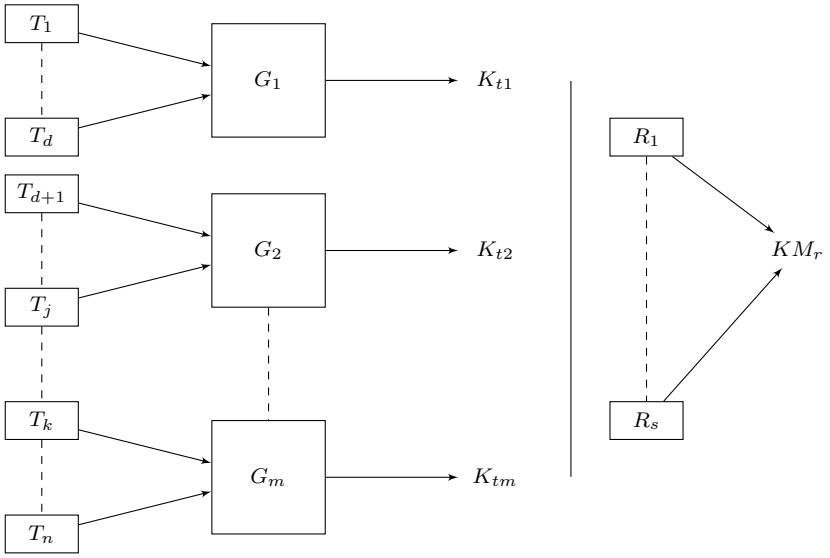

Fig. 2. Key Distribution Phase.

type, specification, application, date of manufacture, lot number, date of expiry and geographic location, etc. Since, $n \gg m$, it is easy to distribute a total of m keys to n tags (the same key for each tag belonging to one group). The number of readers that scan these groups is denoted by s. The supplier distributes one master key KM_r to each reader.

Group Verification Phase. After the key distribution phase is complete, and the supplier makes corresponding entries in the database, the groups of tagged items are shipped to their respective locations. When a group reaches a particular reader in the supply chain process, the GV phase is initiated. The protocol is shown in Figure 3 and is as follows:

1. The reader R_h initiates an EPC Class-1 Gen-2 UHF protocol.
2. The tag HVT_i (whose slot-counter is zero, see [5]) responds showing that it is an HV tag belonging to group G_j.

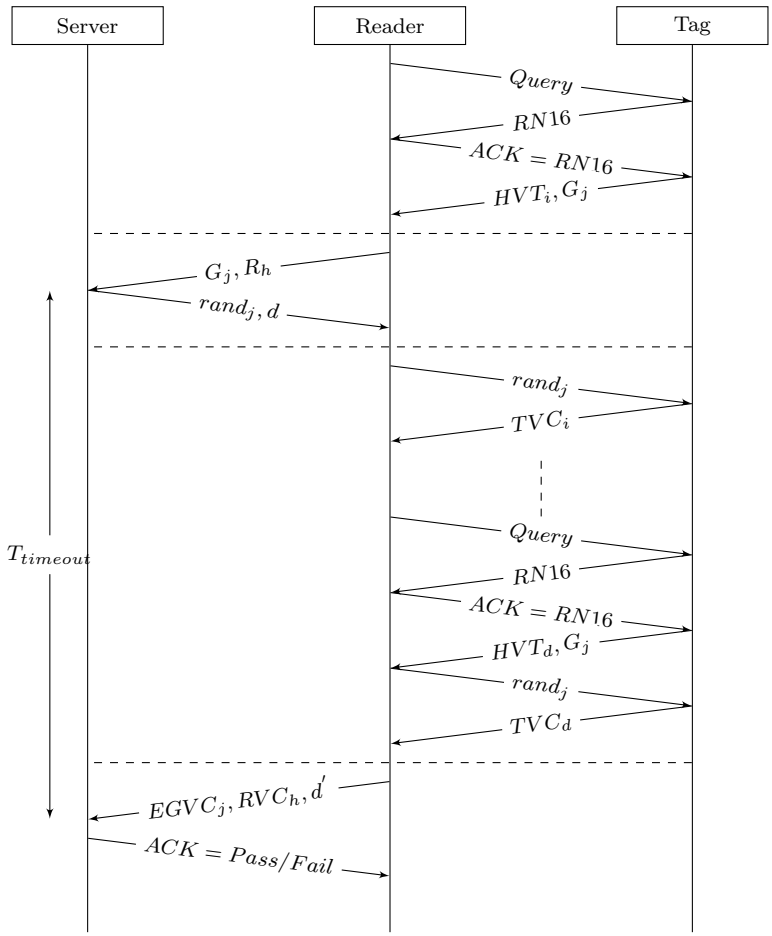

Fig. 3. Group Verification Protocol.

3. The reader sends this group identifier G_j and its own identifier R_h to server S.
4. The server S generates a random nonce $rand_j$ and sends it to the reader R_h along with the total number of tags d in group G_j.
5. The reader R_h then forwards $rand_j$ to each tag.
6. Each tag computes its verification code and sends it to reader R_h. Tag HVT_i belonging to group G_j computes its TVC_i as follows:

$$TVC_i = ID_i \oplus F(K_{tj}, rand_j). \tag{1}$$

7. The reader R_h computes a GVC by XOR-ing with the previous TVC every time it receives a new TVC, until all d tags have responded:

$$GVC_j = TVC_i \oplus ... \oplus TVC_d. \tag{2}$$

8. The reader R_h computes a session key as:

$$KS_{hj} = KDF(KM_r, R_h, G_j). \tag{3}$$

9. The reader R_h encrypts $rand_j$ to compute RVC_h and GVC_j using KS_{hj}, and sends it as $EGVC_j$ along with the total number of tags $d^{'}$ that it read within time $T_{timeout}$ to the server S. RVC_h and $EGVC_j$ are computed as follows:

$$RVC_h = E(KS_{hj}, rand_j). \tag{4}$$

$$EGVC_j = E(KS_{hj}, GVC_j). \tag{5}$$

10. The server first checks the legitimacy of reader R_h by decrypting RVC_h. The server S next checks that reader R_h has read all the tags from the value of $d^{'}$ (to determine any missing/dummy tags). The server S finally decrypts the $EGVC_j$ sent by reader R_h to check whether GVC_j is correct:

if $D(KS_{hj}, RVC_h) == rand_j$ **then**
  R_h is legitimate;
  Check;
  if $d^{'} == d$ **then**
    All tags have been read;
    Check;
    if $D(KS_{hj}, EGVC_j) == GVC_j$ **then**
      G_j is successfully authenticated;
      Send $ACK = Pass$ to R_h;
    end if
  else
    GV has failed;
    Send $ACK = Fail$ to R_h;
  end if

else
R_h is not legitimate;
Abandon the protocol;
end if

If the final $ACK = Pass$, this shows that group G_j has passed the GV phase successfully. A corresponding entry is made in the database for the group G_j scanned by reader R_h, which also helps in future transactions with this particular reader in terms of trust level. The construction of the GV layer is as shown in the example given in Figure 4, where the group G_1 consisting of four tags is being scanned by the reader R_5.

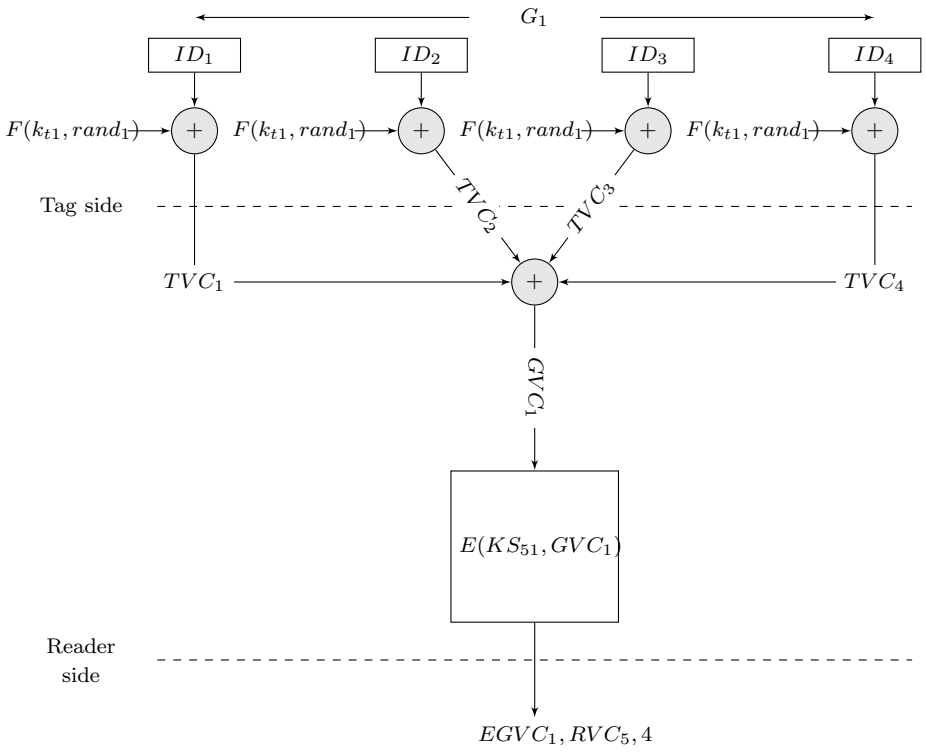

Fig. 4. Construction of the GV Layer.

Product Verification Phase. When $ACK = Fail$ is sent to reader R_h, this shows that the GV layer has not verified the authenticity of the group. In this case the PV phase is initiated as shown in Figure 5.

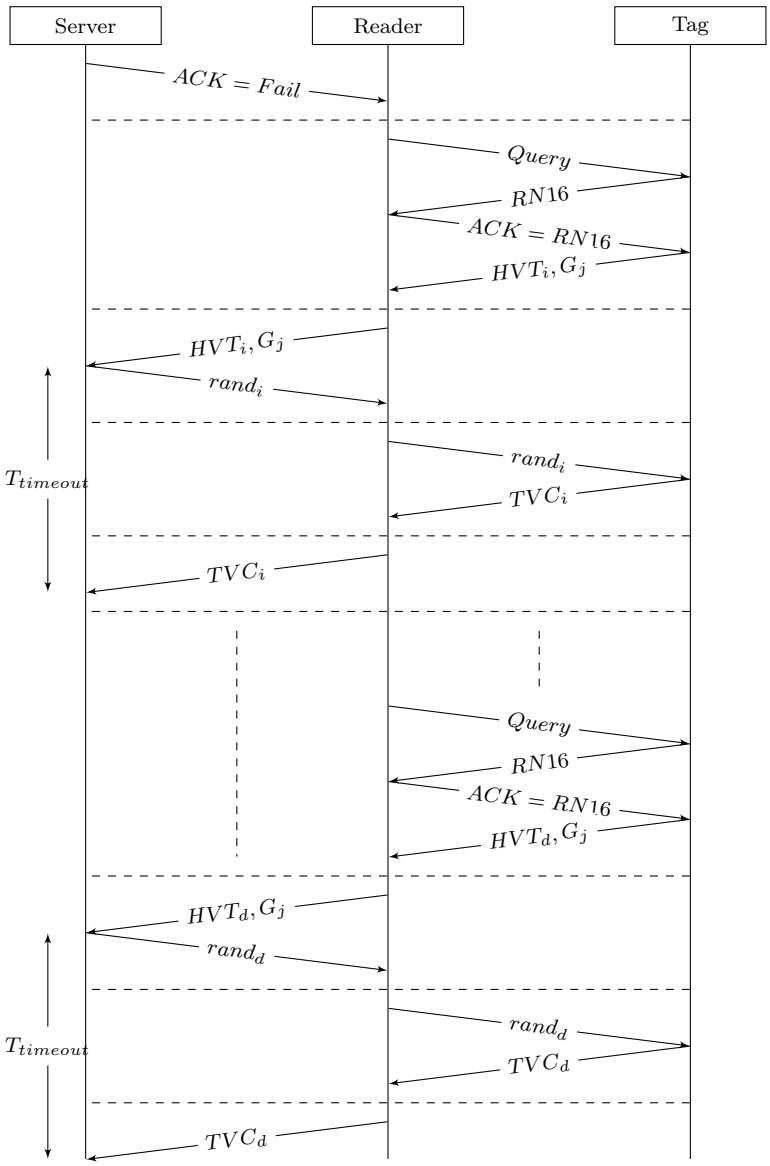

Fig. 5. Product Verification Protocol.

1. The reader R_h sends the tag identifiers $HVTs$ to the server.
2. The server S generates a random challenge $rand$ for each tag.
3. The reader R_h forwards this challenge to the corresponding tag, receives the TVC and forwards it back to the server S.

4. The server S verifies the legitimacy of an individual tag as follows:
   if $TVC_i == ID_i \oplus F(K_{tj}, rand_i)$ **then**
   Tag with identifier HVT_i is a genuine tag;
   else
   Tag with identifier HVT_i is a counterfeit tag;
   end if
5. At the end of this protocol, the server S is able to identify the counterfeit tags as well as missing/dummy tags.

Track and Trace Phase. In the EPC Global Network Class-1,Gen-2 standard, the unique and secret identifier ID (the EPC) is used to track and trace the tag's movement throughout the supply chain. We give an example in Figure 6 to explain the TT phase. Suppose that a particular

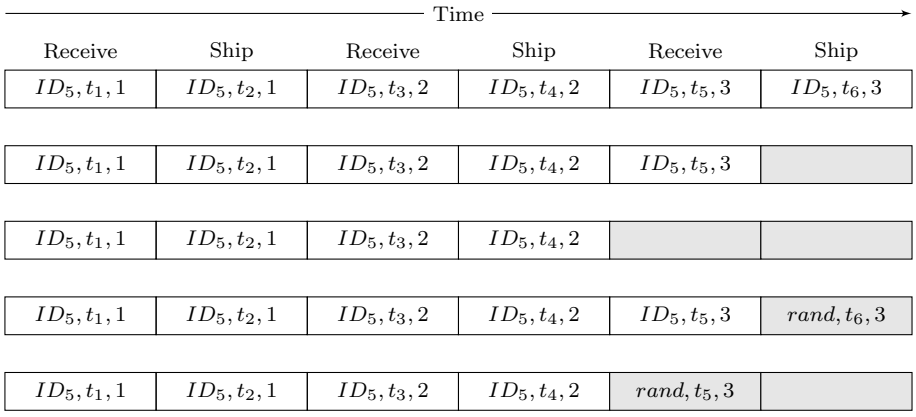

Fig. 6. Track and Trace Example.

item travels in a group through three different companies (Company 1, 2 and 3) before reaching its retailer. The server S maintains its receiving and shipping record at each company. The entry $ID_5, t_1, 1$ shows that the item with tag identifier ID_5 was received at time t_1 by Company 1. A track and trace operation results in one of the following:

- **Case 1: No anomaly.** The first record in Figure 6 shows an ideal case where a particular item ID_5 is successfully shipped to the retailer.
- **Case 2: Missing item within company.** The second record is that Company 3 received the item at t_5 but never shipped it to the retailer.

– **Case 3: Missing item en-route.** The third record shows the item was shipped by Company 2 but was never received by Company 3.
– **Case 4: Counterfeit item within company.** The fourth record shows that Company 3 received the original authentic item at t_5 but shipped a suspected counterfeit to the retailer.
– **Case 5: Counterfeit item en-route.** The last record shows that Company 2 shipped the original authentic item at t_4 but the item received by Company 3 is a suspected counterfeit.

4 Analysis

We now carry out an analysis of our proposed anti-counterfeiting mechanism as to whether it achieves the desired goals of Section 3.1.

4.1 Anti-cloning

As discussed in Section 2.1, if a tag transmits its static secret identity ID such as EPC in the clear then it can be copied easily. This unique identity of the tag is linked with its associated information, which potentially includes value, composition and other useful supplier-related data. In the proposed mechanism the tag hides this secret static identity in its verification code. The tag thus transmits its verification code which appears to be random data. Thus an adversary cannot make a copy of the secret static identity from a genuine tag. However the adversary can copy the public identifier HVT of a tag to a counterfeit tag, but this counterfeit tag will not be able to reproduce the correct verification code and thus will fail the legitimacy verification.

4.2 Anti-spoofing

As discussed in Sections 2.3 and 2.4, static secret identities can be replayed by a counterfeit tag to spoof as genuine. In the proposed mechanism the secret information, transmitted for verification, changes in every transaction because of the use of a fresh random nonce generated by the server. An adversary can thus not replay this secret information during a later transaction because of the use of a new nonce.

4.3 Anti-theft

The proposed mechanism employs a layered approach in order to detect stolen/missing products. As described in Section 3.2, this approach can

be used to identify the exact cause and location of any anomaly since the final Track and Trace layer provides the complete shipping and receiving record of the identified stolen/missing item. After tracing the root cause of the anomaly, suitable processes can be undertaken to hold the responsible parties accountable. Appropriate countermeasures can then be applied in order to prevent this anomaly from occurring again.

We note that a smart adversary can prevent this detection by relaying the genuine verification code from a stolen/missing item which is not physically present in the vicinity of the scan range. To counter such an adversary, we have employed a time-out clock within our scheme. The server pre-computes this time, depending on the number of tags involved in the scheme. The server then expects the reader to answer back within that specified time. If the reader delays its response, this is an indication of a potential relay attack. The server can thus ask the respective reader for a physical check for the completeness of this group and makes a corresponding entry for this anomaly in its database.

4.4 Scalability

In supply chain environments, tags travel most part of their journey in groups. These groups can be large, medium or even consisting of a single item depending on their size, value, and application. Sometime these groups change in their sizes en-route from manufacturer to end-users. Our proposed group verification layer can verify any group irrespective of its size and, during the product verification layer, the legitimacy of a single tagged item is checked. Therefore, our scheme scales well through from large groups, to medium and small groups, and even to stand-alone items.

4.5 Compatibility

The proposed mechanism uses the standard UHF Air Interface Protocol as specified in [5]. If a tag is not employing hierarchical verification, it can be read as per the existing standard.

4.6 Efficient Key Management

The proposed mechanism avoids some of the scalability problems of using symmetric cryptography by providing all tags belonging to a specific group with a unique key. Since the number of groups is much smaller than the number of tags, it is comparatively easy to manage the keys in

the database. Additionally, all readers involved in the supply chain management system are only given one master key. By reducing the overall number of keys in the system, the key management is considerably more efficient than schemes with a unique key for each tag as mentioned in [17].

4.7 Good Throughput

The layered approach is partly designed to reduce the likelihood of potential bottlenecks arising from readers having to stay online during authentication and verification, and regularly interact with the database. By first deploying relatively lightweight group-level checks we avoid bottlenecks in the top layer of the hierarchical verification process. Overall performance decreases in the lower layers, but these are only activated if anomalies are detected during the group-level checking. In this way a reasonable throughput can be expected of the system.

5 Conclusion

We have proposed a hierarchical anti-counterfeiting mechanism using three layers of verification to determine the legitimacy of a tagged item. The mechanism is designed for EPC Class-1 Gen-2 tags used in supply chain management where counterfeiting present a very serious threat. This threat is countered using dynamic verification codes generated using symmetric cryptography. Our model detects the stolen/missing items, provides efficient key management, avoids bottlenecks and is scalable to the complete lifecycle of tags in the supply chain. This approach also potentially lends itself to deployment in schemes based on other standards.

References

1. Y. Lee, F. Cheng, and Y. T. Leung, "Exploring the impact of RFID on supply chain dynamics," in *Simulation Conference*, vol. 2. IEEE, December 2004, pp. 1145 – 1152.
2. A. Juels, R. Pappu, and B. Parno, "Unidirectional Key Distribution Across Time and Space with Applications to RFID Security," in *Proceedings of the 17th USENIX Security Symposium*, July-August 2008, pp. 75–90.
3. A. Juels, "RFID security and privacy: a research survey," in *Journal on Selected Areas in Communications*, vol. 24, no. 2. IEEE, 2006, pp. 381–394.
4. P. Avery, F. Cerri, L. H. Fayle, K. Olsen, D. Scorpeci, and P. Stryzowski, "The Economic Impact of Counterfeiting and Piracy." Organisation for Economic Cooperation and Development (OECD), 2008.
5. G. E. Standards, "EPC Radio-Frequency Identity Protocols Class-1 Generation-2 UHF RFID Protocol for Communications at 860 MHz-960 MHz," in *Specification for RFID Air Interface*, 2008.

6. M. Lehtonen, T. Staake, F. Michahelles, and E. Fleisch, "From Identification to Authentication - A Review of RFID Product Authentication Techniques," in *Networked RFID Systems and Lightweight Cryptography*. Springer, 2008, pp. 169–187.
7. M.-J. O. Saarinen and D. Engels, "A Do-It-All-Cipher for RFID: Design Requirements (Extended Abstract)," in *Cryptology ePrint Archive, Report 2012/317*. IACR, 2012.
8. R. Koh, E. W. Schuster, I. Chackrabarti, and A. Bellman, "Securing the Pharmaceutical Supply Chain," in *White Paper, Auto-ID Labs*. Massachusetts Institute of Technology, 2003.
9. K. Takaragi, M. Usami, R. Imura, R. Itsuki, and T. Satoh, "An Ultra Small Individual Recognition Security Chip," in *Micro*, vol. 21, no. 6. IEEE, 2001, pp. 43–49.
10. T. Staake, F. Thiesse, and E. Fleisch, "Extending the EPC network: the potential of RFID in anti-counterfeiting," in *Proceedings of the 2005 ACM symposium on Applied computing*, ser. SAC '05, vol. 6. ACM, 2005, pp. 1607–1612.
11. J. Pearson, "Securing the Pharmaceutical Supply Chain with RFID and Public-key Infrastructure (PKI) Technologies." in *Texas Instruments White Paper*, June 2005.
12. I. Vajda and L. Buttyán, "Lightweight Authentication Protocols for Low-Cost RFID Tags," in *Second Workshop on Security in Ubiquitous Computing*, ser. Ubicomp 2003, 2003.
13. M. Feldhofer, S. Dominikus, and J. Wolkerstorfer, "Strong Authentication for RFID Systems Using the AES Algorithm," in *6th International Workshop on Cryptographic Hardware and Embedded Systems*, 2004, pp. 357–370.
14. M. Feldhofer, M. Aigner, and S. Dominikus, "An Application of RFID Tags using Secure Symmetric Authentication," in *International Workshop on Security, Privacy and Trust in Pervasive and Ubiquitous Computing*, 2005, pp. 43–49.
15. S. Dominikus, E. Oswald, and M. Feldhofer, "Symmetric Authentication for RFID Systems in Practice," in *The Ecrypt Workshop on RFID and Lightweight Crypto*, July 2005.
16. A. Poschmann, G. Leander, K. Schramm, and C. Paar, "A Family of Light-Weight Block Ciphers Based on DES Suited for RFID Applications," in *Workshop on RFID Security*, ser. Lecture Notes in Computer Science, July 2006.
17. A. Arbit, Y. Oren, and A. Wool, "Toward Practical Public Key Anti-Counterfeiting for Low-Cost EPC Tags," in *International IEEE Conference on RFID*.
18. L. Batina, J. Guajardo, T. Kerins, N. Mentens, P. Tuyls, and I. Verbauwhede, "An Elliptic Curve Processor Suitable For RFID-Tags," IACR, 2006.
19. S. Martínez, M. Valls, C. Roig, J. M. Miret, and F. Giné, "A Secure Elliptic Curve-Based RFID Protocol," in *Journal of Computer Science and Technology*, vol. 24, no. 2, 2009, pp. 309–318.
20. P. Tuyls and L. Batina, "RFID-Tags for Anti-Counterfeiting," in *The Cryptographers' Track at the RSA Conference*, 2006.
21. Y.-S. Lee, T.-Y. Kim, and H. J. Lee, "Mutual Authentication Protocol for Enhanced RFID Security and Anti-counterfeiting," in *26th International Conference on Advanced Information Networking and Applications Workshops*, 2012, pp. 558–563.
22. A. Juels, "Strengthening EPC Tags Against Cloning," RSA Laboratories, March 2005.
23. D. W. Engels, M.-J. O. Saarinen, P. Schweitzer, and E. M. Smith, "The Hummingbird-2 Lightweight Authenticated Encryption Algorithm," in *7th International Workshop on RFID Security and Privacy*, 2011, pp. 19–31.