

Industrial Privacy in RFID-based Batch Recalls

Leonardo Weiss Ferreira Chaves
SAP Research, Karlsruhe, Germany
leonardo.weiss.f.chaves@sap.com

Florian Kerschbaum
SAP Research, Karlsruhe, Germany
florian.kerschbaum@sap.com

Abstract

Batch recalls are an important topic for manufacturers and producers. Especially in the food and in the pharmaceutical industry, producers are obliged to implement recalls in order to comply with legislation. In extreme cases, non-compliance can cause loss of life, e.g. when perished food or medicine reaches the consumer. Current batch recall practice is expensive and difficult, since many supply chain partners need to combine the data from their ERP systems. Radio Frequency Identification (RFID) can be used to efficiently implement batch recalls, e.g. by storing batch numbers from the parts/ingredients used in all manufacturing steps. But this raises concerns on industrial privacy, since competitors could use this information to gain insight into the whole supply chain. We overcome this problem by storing tracing information on RFID tags and encrypting the information, such that it is only available in case of a recall. We encrypt the information using identity-based encryption and furthermore allow universal re-encryption along the supply chain to prevent information leakages from the ciphertexts.

1. Introduction

Batch recalls require tracing of goods or parts across the partially or entirely reconstructed supply chain. This then allows the producer to recall certain products or batches of products in case of quality problems or defects that would jeopardize product reliability, especially in the automotive or aerospace industries. In other industries batch recalls are enforced by legislation, like in the food and in the pharmaceutical industry. In extreme cases non-compliance can cause loss of life, e.g. when perished food or medicine reaches the consumer.

A batch recall is a difficult task, as supply chains are long and involve several producers and several

different materials. Further, materials from different batches may contribute to a single batch of a finished product. That means recalling cascades to lots of suppliers and batches, making it difficult to exactly determine which products have to be recalled. Therefore, the conditions for recalls become very broad, recalling many more products than needed to minimize the chance of exposing end-users to such defective products. This makes recalls very expensive.

Radio Frequency Identification (RFID) [10] attached to products (item-level tagging) allows tracing information to be stored and updated on the product itself. Thus, the conditions for a recall can be met on an item level, significantly reducing the costs of recalling too many items. The use of RFID technology would further ease data exchange along the supply chain, as batch numbers would always be available on the RFID tag. That means producers will not depend on the willingness of supply chain partners to share data.

When using RFID, producers have concerns on their industrial privacy. If the batch numbers were stored on the RFID tag, competitors could use this information to gain insight into the amount of products produced. Further, if batch numbers from the ingredients/parts of a product are stored, competitors could gain further insight into the whole supply chain. Therefore, RFID technology is seldom used for batch recalls.

Encrypting the batch numbers using traditional symmetric or asymmetric cryptography is possible, but introduces a huge key management problem, since the producer would need to generate and store a cryptographic key for each batch. Furthermore, the keys would either need to be large enough to avoid collisions due to birthday attacks or each company would need to be assigned a unique key space which makes this solution even more impractical.

We can summarize the requirements presented above as follows:

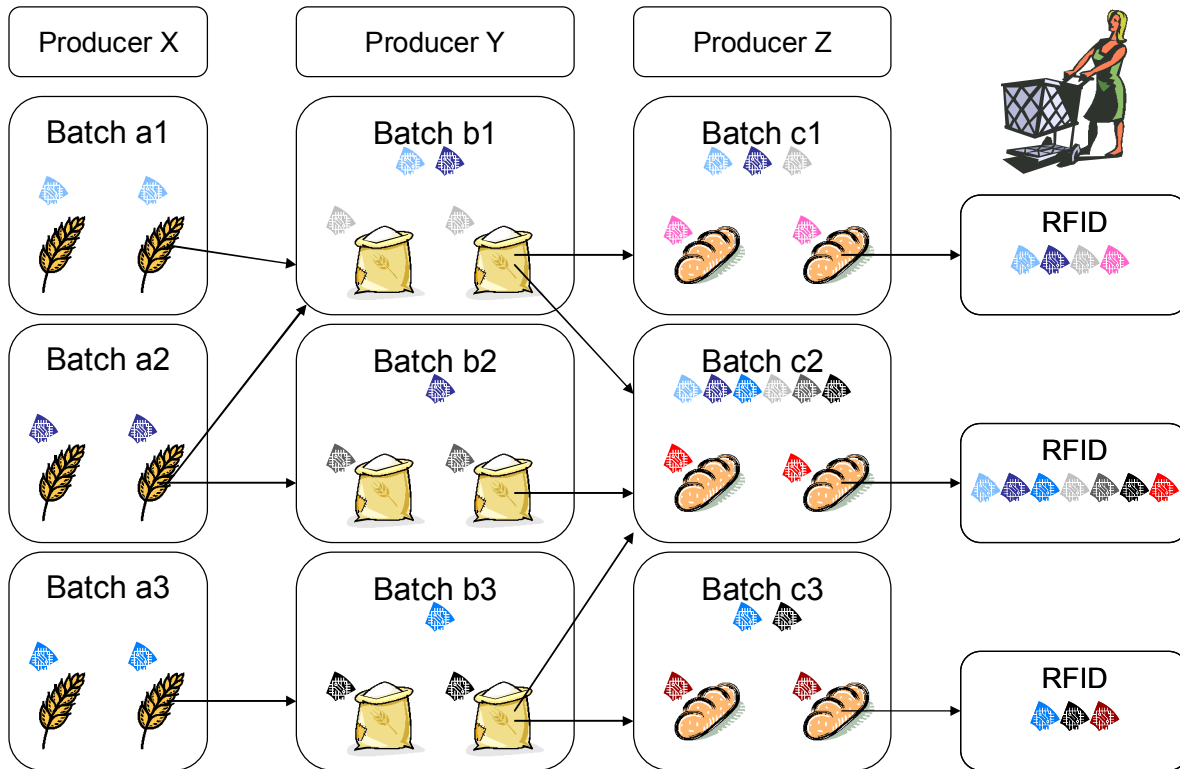


Figure 1. Example Supply Chain

1. Industrial Privacy: All non-recalled batches should not reveal any information about the supply chain.
2. Autonomy: Any company X should be able to recall all products that contain one of its batches without the help of companies downstream in the supply chain.
3. Simple Key Management: A producer is not supposed to maintain a key for each batch he produces.

In this paper we present a solution that fulfills these requirements. Our contributions to each requirement are

1. We present the first universally re-encryptable identity-based encryption scheme (for the Boneh-Franklin [2] and Boneh-Boyen-Goh [3] identity-based encryption schemes) which fulfills requirement (1), because we can re-randomize each ciphertext.
2. Each ciphertext is stored on the RFID tag, such that all of them are independently accessible to any consumer. Each company can therefore independently issue recalls which fulfills requirement (2).

3. We use the batch number as a key in an identity-based encryption scheme. Therefore, no one has to maintain any key information (except maybe the trusted third party TTP) and requirement (3) is completely fulfilled.

In the next section we outline an example scenario from the food industry that motivated our work. In Sections 2 and 3 we present the building blocks of our solution and our solution itself. Then we compare our solution to related work and close with a conclusion.

1.1. Example Scenario

The following scenario should give the reader an impression on the complexity of batch recalls. We use an example from the food industry to motivate our work. In other industries the complexity is similar.

In Europe, batch recalls are enforced through EU regulation 178/2002, and in the U.S. they are enforced by the Food and Drug Administration (FDA). In the U.S. for instance, foodborne pathogens are estimated to cause 76 million illnesses and 5,000 deaths each year [13] and societal costs are estimated between \$2.9 and \$6.7 billion per year [4].

Let us consider the example shown in Figure 1. Producer X sells grain. The grain is shipped in different batches (a1, a2, a3). Producer Y buys grain from Producer X and makes flour out of it. He also has different batches (b1, b2, b3). Note that different batches from Producer X may contribute to a single batch of flour. Producer Z uses the flour from Producer Y to bake bread.

In the food industry, samples of goods from every batch have to be analyzed in a laboratory in each part of the supply chain to ensure that goods do not contain pathogens. It is common practice to ship the goods before having the laboratory results, since results usually arrive before the goods are used in further production steps or are consumed. If there is a problem, the products are recalled. Sometimes laboratory results arrive after goods have been used in further production steps, making batch recalls more complicated.

In Section 3.4 we will describe how recalls can be implemented using RFID technology together with our solution.

2. Building Blocks

In this section we present the building blocks needed to implement our solution.

2.1. Identity-based Encryption

Identity-based encryption [16] is an alternative to public-key encryption. In identity-based encryption a trusted third party sets up some basic (public) parameters. Then the public key can be any string, e.g. a batch number. To obtain the private key, one presents the public key to the trusted third party and proves that he has the right to obtain the private key. Then the trusted third party issues the private key. The encryption is randomized, such that a ciphertext does not reveal any information.

2.1.1. Boneh-Franklin Encryption

The first practical such encryption system was the Boneh-Franklin (BF) [2] encryption system. It is based on pairings in elliptic curves. We denote points on an elliptic curve with upper case letters: $P, Q \dots$ and numbers in Z_p with lower case letters: $r, s \dots$. A pairing, such as the Weil pairing, has special properties. A good introduction to the topic is provided in [14]. Let $e(P, Q)$ denote the cryptographic pairing used in identity-based encryption. Then

$$\begin{aligned} e(rP, Q) &= e(P, rQ) = e(P, Q)^r \\ e(P, Q) &\neq 1 \text{ (w.h.p.)} \end{aligned}$$

Let P and $T = tP$ be the public parameters of the BF crypto system (and t be the private information of the trusted third party). Note that in an elliptic curve group it is hard to invert multiplication, i.e. it is hard to compute t from T and P . This corresponds to the discrete logarithm problem in prime groups, where it is hard to invert exponentiation.

For encryption with the identity ID one chooses a random number r . Let H_I be a cryptographic hash function that maps identities to points on the elliptic curve. Then one computes $e(H_I(ID), T)^r$. Let H_C be a cryptographic hash function that maps pairs to bit strings of a fixed length. The ciphertext for message m is then:

$$rP, H_C(e(H_I(ID), T)^r) \oplus m$$

For decryption one obtains $tH_I(ID)$ from the trusted third party. One computes

$$e(tH_I(ID), rP) = e(H_I(ID), tP)^r = e(H_I(ID), T)^r$$

and uses its hash to decrypt the ciphertext.

2.1.2. Boneh-Boyen-Goh Encryption

The Boneh-Boyen-Goh (BBG) encryption system [3] is a recent development that also supports hierarchical identity-based encryption. We omit this extension for simplicity, as well as its non-malleability feature.

The trusted third party chooses a random number a as its private information. It publishes $(G, G_1 = aG, G_2, G_3, H)$ as public parameters.

In order to encrypt one chooses a random number s and sets the ciphertext to

$$e(G_1, G_2)^s m, sG, s(id H + G_3)$$

For decryption one contacts the TTP and obtains

$$aG_2 + r(id H + G_3), rG$$

where r is a random number chosen by the TTP. One then computes

$$\begin{aligned} &(e(G_1, G_2)^s m) e(rG, s(id H + G_3)) / \\ &e(sG, aG_2 + r(id H + G_3)) \\ &= m \end{aligned}$$

2.2. RFID

Radio Frequency Identification (RFID) [10] is a cheap method to incorporate and store information on an item. RFID tags which are attached to items have limited storage and processing capabilities, but are cheap to produce and can be read and written to remotely. Current RFID tags can store up to 64Kbyte of data [11], and as the technology evolves, the storage capacity increases. Thus, we do not expect the storage capacity to be a limiting factor for batch recalls.

3. Our Solution

3.1. Solution Details for BF Encryption

Note that most of the ciphertext of BF encryption $R=rP, e(H_I(ID), T)^r$ can still be re-randomized by computing $r'R = r'P, (e(H_I(ID), T)^r)^{r'} = e(H_I(ID), T)^{r'r}$. We therefore transmit the message m (which is not necessarily sensitive) in plaintext alongside the partial ciphertext. As an additional benefit we aggregate the recall information (messages m) upstream along the supply chain. The details are as follows:

1. A producer X has a batch a which he intends to ship to Y and Y' . He sends the shipment along with $[R=rP, e(H_I(X|a), T)^r, inf_X]$ to Y (and Y'). inf_X can be any information he wants to reveal in case of a recall. In case there is no such information it can be a random number.
2. Consumer Y is an intermediary. He produces several batches b from a and ships them to consumer Z . He re-randomizes the information of X and sends along the re-randomized ciphertext

$$\begin{aligned} [R=r'P, e(H_I(X|a), T)^{r'}, inf_X] \\ [R=rP, e(H_I(Y|b), T)^r, inf_Y] \end{aligned}$$

3. Consumer Z produces the final good c . He re-randomizes the received information and places an RFID tag on each item with the following information:

$$\begin{aligned} [R=r''P, (inf_X|inf_Y|inf_Z|H(inf_X|inf_Y|inf_Z)) \oplus \\ H_C(e(H_I(X|a), T)^{r''})] \\ [R=r'P, (inf_Y|inf_Z|H(inf_Y|inf_Z)) \oplus \\ H_C(e(H_I(Y|b), T)^{r'})] \\ [R=rP, (inf_Z|H(inf_Z)) \oplus H_C(e(H_I(Z|c), T)^r)]. \end{aligned}$$

In case of a recall, e.g. by supplier/consumer Y , he issues the batch b to be recalled along with a proof of identity to the trusted third party. The trusted third party publishes the private key for $Y|b$, i.e. $tH_I(Y|b)$, to all resellers (or even end-users) which can then scan their entire inventory for recalled goods. Recall that only the trusted third party knows t . No involvement of intermediary supply chain partners, such as Z , is necessary in the tracing. Y himself can issue the recall to the resellers/end-users directly increasing the reaction time to market.

3.2. Solution Details for BBG Encryption

Our solution for BBG encryption allows hiding the plaintext, such that it may contain sensitive information that should only be revealed in case of a recall, but then ciphertexts cannot be aggregated upstream in the supply chain.

Let $e(G_1, G_2)^s m, sG, s(id H + G_3) = (a_1, B_1, C_1)$ be the ciphertext. Then we store the following additional part of the ciphertext on the RFID tag:

$$e(G_1, G_2)^r, rG, r(id H + G_3) = (a_2, B_2, C_2)$$

This corresponds to an encryption of I under the same identity. In order to re-randomize, one chooses two random numbers v, w and computes

$$a_1 a_2^v, B_1 + vB_2, C_1 + vC_2 \quad a_2^w, wB_2, wC_2$$

The result is a ciphertext that is completely indistinguishable ciphertext where the first part is randomized by $s + rv$ and the second part by rw . Each company X places a ciphertext for its batch with public key X ["batch number" on the RFID tag or transmits it along the supply chain until it can be stored on the final tag. Such transmission can also occur via RFID tags.

3.3. Comparison of BF and BBG Encryption

Both solutions protect the sensitive information along the supply chain and the users of the system can make a choice which encryption system they prefer. In case of BF encryption the text for the recall needs to be carried along the supply chain until the last member, while in the case of BBG encryption the plain text for the recall can be protected at the encrypting party's site. BF encryption on the other hand produces shorter cipher texts that are easier to store on an RFID tag. A BF ciphertext has the length

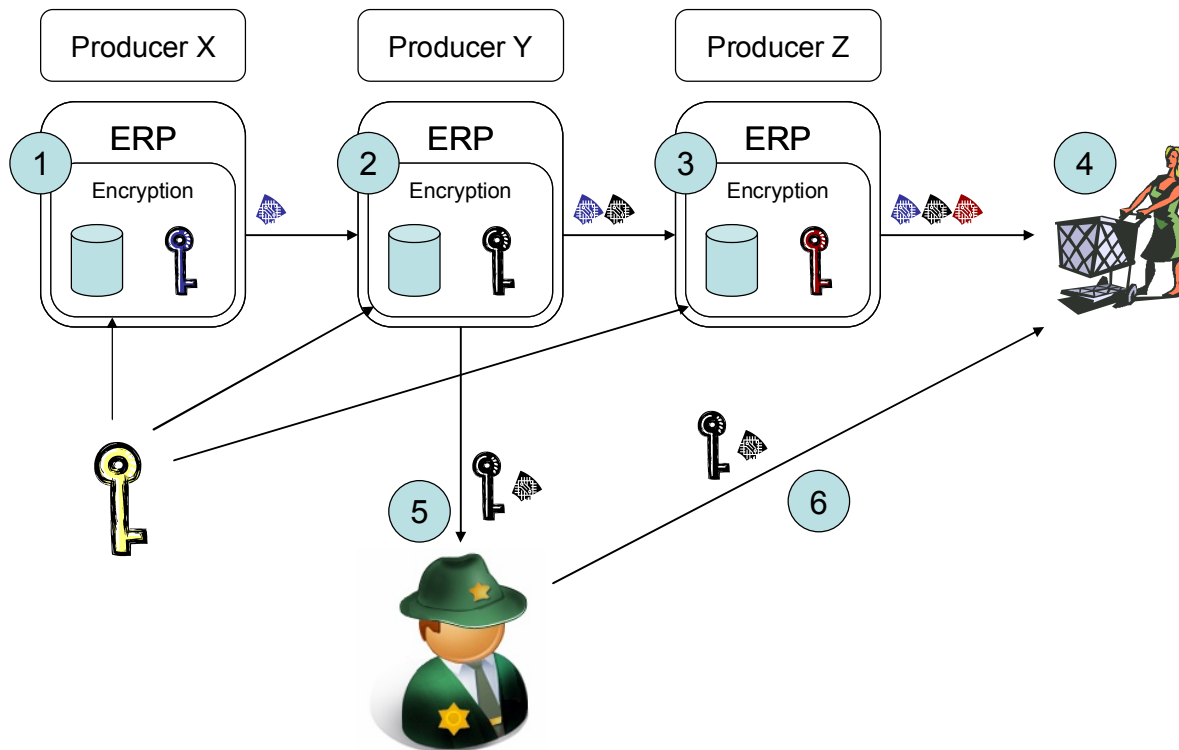


Figure 2. Process for using identity based encryption in the supply chain

of the plaintext plus one element of the elliptic curve group (i.e. key length). A BBG ciphertext has the length of three elliptic curve group elements where one needs to be the size of the message, i.e. three times the message size. Both systems rely on the same security assumptions and are provably secure. Key sizes in elliptic curve systems can usually be small on the order of hundred bits.

In identity-based encryption, the trusted third party can decrypt every message (as long as it can guess the key). The trusted third party should therefore be an independent organization, not involved in any supply chain being protected. We suggest governmental organizations or trade associations to act as a trusted third party.

3.4. Integration into Business Applications

We will outline one possible implementation based on the scenario described in Section 1.1. To allow batch recalls, each producer will attach RFID tags to their goods. The RFID tags contain tracing information. The tracing information is encrypted using identity based encryption, e.g. BF or BBG encryption. In each part of the supply chain, the producer will include encrypted information about the

batch he produced, and about the products and batches he used in his product. Figure 2 shows this process in detail.

Each producer uses an Enterprise Resource Planning (ERP) system. The ERP system has a component for identity based encryption. This component has a database that stores information about the produced batches and private cryptographic information used for encrypting information for each batch (key) which is in our case equivalent to the batch number. There is also public cryptographic information which is known by all partners of the supply chain. Further, there is a trusted third party (sheriff).

Producer X produces a good. In Step 1, information about the batch of the good is encrypted, written to an RFID tag, and sent to Producer Y. Producer X stores the information about the batch and the private cryptographic information in the database of the Encryption Component of his ERP system. Note that each ERP system only contains the data of batches used in their own production process. It is not possible to infer all the batches used in the downstream or upstream supply chain, as data is either encrypted or not available at all.

Clearly there is no additional risk with storing the encrypted information on the RFID tag. Although RFID tags can be compromised and are therefore not suitable for storing cryptographic material, such as keys, we only store the ciphertext on the RFID tag and the encryption scheme would need to be broken and not the RFID tag.

In Step 2, Producer Y produces a good using different batches from Producer X. Each batch he produces will contain encrypted information about the batch of the good. It will be written to an RFID tag and sent together with the goods to Producer Z. The problem is that the encrypted information may leak information as well, e.g. two batches from Y carrying the same encrypted information from X, i.e. they have been made from the same batch. This problem aggravates as the goods move along the supply chain and one can identify and trace goods to their originating batches.

Although identity-based encryption already offers ciphertext indistinguishability, i.e. the same plaintext maps to many (indistinguishable) ciphertexts, the choice of randomness is usually made during encryption. The parties downstream in the supply chain receive the ciphertext and without knowing the encryption key (batch number) they cannot re-randomize the ciphertext, but the batch number may be sensitive information. For public-key encryption, more precisely the El-Gamal encryption system, re-randomization without the public key has been achieved in [5]. We also add this feature to identity-based encryption as described in Sections 3.1 and 3.2. Now a downstream producer can re-randomize the ciphertext, such that

- he does not need the batch number to re-randomize and
- the resulting ciphertexts are indistinguishable, i.e. no one can tell without the decryption key that they are from the same plaintext.

Therefore the encrypted information will also be re-randomized. Producer Y modifies the encrypted information from Producer X without knowing his batch information, such that each good of X will carry encrypted information from X that looks different as long as the decryption key remains unknown. Figure 2 only shows a simple example in which each batch only contains products from one other batch. Fig. 1 contains more complex examples in which materials from different batches may contribute to a single batch.

In Step 3, Producer Z will produce a good and fit it with an RFID tag containing encrypted information likewise to Step 2. In Step 4, the finished product will be shipped to a retailer that will offer the product to consumers.

Recall that sometimes laboratory results arrive after goods have been used in further production steps? This is the case in Step 5. Producer Y produced one batch of rotten goods. The goods were already processed by Producer Z and shipped to a retailer. To allow recalling of the affected products, Producer Y will reveal the private cryptographic information about the rotten batch to a trusted third party. In Step 6, the trusted third party will publish this information to all retailers. The retailers can use this information to find the affected products and remove them from their shelves. This information might also be published to consumers. This allows consumers to check if products they bought might need to be recalled, e.g. by using a fridge equipped with an RFID reader (smart fridge).

4. Related Work

A ciphertext that reveals no information about its plaintext is usually achieved by probabilistic encryption. A problem that remains, is when one ciphertext is supposed to be used twice, i.e. one incoming batch maps to two outgoing batches. In such a case the ciphertext needs to be re-randomized. This can be done with knowledge of the (public) key, e.g. via a homomorphic operation, or without knowledge of the key which is called universal re-encryption [5]. An extension to prevent replacement of the ciphertext is presented in [1]. All currently known universally re-encryptable encryption schemes are public-key.

There are a number of identity-based encryption [16] schemes including BF [2] and BBG [3].

Privacy in RFID has mostly been investigated from a user's point of view. Unauthorized reading of RFID tags allows tracking and spying on the users who carry them. Recent results to prevent such intrusion are presented in [7]. The confidentiality of data on the tags is less frequently considered and if it is, then in the context of personal privacy. In [12] this is defined as data privacy. The privacy of RFID tags has been formally defined as indistinguishability of tags without key information [9]. Industrial privacy, as the confidentiality of business secrets in inter-organizational business applications, has been rarely considered before in RFID scenarios and has not yet been considered for recalls.

Recalls are currently managed via supply-chain wide (or partial) central or local combination of all information from the individual ERP systems. These solutions do not fulfill requirements (2) and (3), since they require a connectivity of ERP systems (in particular for a central solution) and require collaboration in case of a recall.

The idea of using RFID for batch recalls has been explored before, but the traditional use of RFID in batch recalls is to track items belonging to certain batches [6, 15] and then to precisely recall only these. This limits the extent of a recall to only damaged products which provides an advantage over non-item based recalls. On the contrary our approach is quite revolutionary. It does away with the need to store batch numbers in ERP systems and track them along the supply chain and stores the entire information on the RFID tag. Obviously this information needs to be protected and we described the necessary cryptographic tools.

An orthogonal way to protect information on RFID tags is presented in [8]. This method is not applicable for batch recalls, since the information is not available on individual items on the shelf.

5. Conclusion

We have presented a solution that fulfills the requirements for batch recalls using RFID tags with universally re-encryptable identity-based encryption which are not fulfilled by current methods for batch recalls and require a novel combination of cryptographic techniques. Our solution implements full industrial privacy overcoming the main obstacle to RFID adoption in batch recalls.

6. Acknowledgements

The developments presented in this paper were partly funded by the European Commission through the project SecureSCM, and by the German government (Bundesministerium für Bildung und Forschung) through the project LoCostix.

7. References

[1] G. Ateniese, J. Camenisch and B. de Medeiros. Untraceable RFID Tags via Insubvertible Encryption. *Proceedings of the 12th ACM Conference on Computer and Communications Security*, 2005.

[2] D. Boneh, and M. Franklin. Identity-Based Encryption from the Weil Pairing. *Proceedings of CRYPTO*, 2001.

[3] D. Boneh, X. Boyen, and E. Goh. Hierarchical Identity Based Encryption with Constant Size Ciphertext. *Proceedings of EUROCRYPT*, 2005.

[4] J. Buzby, T. Roberts, C.-T. Jordan Lin and J.M. MacDonald. Bacterial foodborne disease: Medical costs & productivity losses. *USDA-ERS Agricultural Economic Report 741*, 1996.

[5] P. Golle, M. Jakobsson, A. Juels, and P. Syverson. Universal Re-encryption for Mixnets. *Proceedings of RSA Conference Cryptographers' Track*, 2004.

[6] Y. Hu, S. Sundara, T. Chorma, and J. Srinivasan. Supporting RFID-based item tracking applications in Oracle DBMS using a bitmap datatype. *Proceedings of the 31st International Conference on Very Large Data Bases*, 2005.

[7] A. Juels. RFID security and privacy: a research survey. *IEEE Journal on Selected Areas in Communications* 24(2), 2006.

[8] A. Juels, B. Parno, and R. Pappu. Unidirectional Key Distribution Across Time and Space with Applications to RFID Security. *Proceedings of USENIX Security*, 2008.

[9] A. Juels, and S. Weis. Defining Strong Privacy for RFID. *Proceedings of the 5th IEEE International Conference on Pervasive Computing and Communications Workshops*, 2007.

[10] K. Finkensteller. RFID Handbook: Fundamentals and Applications in Contactless Smart Cards and Identification. John Wiley & Sons, Inc., 2003.

[11] Fujitsu Develops World's First 64KByte High-Capacity FRAM RFID Tag for Aviation Applications. *Press release*, 2008. Available at <http://www.fujitsu.com/global/news/pr/archives/month/2008/20080109-01.html>

[12] M. Langheinrich. RFID and Privacy. M. Petkovic, and W. Jonker (Eds.): *Security, Privacy, and Trust in Modern Data Management*. Springer, 2007.

[13] P. Mead, L. Slutsker, V. Dietz, L. McCaig, J. Bresee, C. Shapiro, P. Griffin and R. Tauxe. Food-related illness and death in the Unites States. *Emerging Infectious Diseases*, 1999.

[14] A. Menezes. An introduction to pairing-based cryptography. *Lecture Notes*, 2005. Available at <http://www.math.uwaterloo.ca/~ajmenez/publications/pairings.pdf>

[15] K. Michael, and L. McCathie. The Pros and Cons of RFID in Supply Chain Management. *Proceedings of International Conference on Mobile Business*, 2005.

[16] A. Shamir. Identity-Based Cryptosystems and Signature Schemes. *Proceedings of CRYPTO*, 1984.