ANTI-COUNTERFEITING OF FASHION BRANDS USING RFID TECHNOLOGY

Patrick C.L. Hui, Kirk H.M. Wong, and Allan C.K. Chan

ABSTRACT

Anti-counterfeiting comes to the attention of fashion brand owners concerned as this counterfeiting problem arises significantly over recent years. RFID technology provides a new alternative to serve this problem. This paper presents a new approach to design a lightweight cryptography and a simple authentication method for RFID passive tags. Tags are embedded inside apparel products, such as garments or high-value apparel accessories. When a counterfeiting situation occurs, tag authentication on that apparel product can be performed whether the apparel product is genuine or faked.

Keywords: Anti-counterfeit, Apparel Product, Authentication, RFID, and Security.

1. INTRODUCTION

Counterfeit problem has been occurred frequently particular in garments or in apparel field for a broad category. Initially Radio Frequency Identification (RFID) can be explored and used as a tool to help solve this problem. RFID technology has been used over half a century, when it was primarily used by the military. As technologies continue to advance forward in antenna technology, microchip fabrication, and radio spread spectrum, RFID is rapidly pushed to the existing markets with diversified applications, such as automatic fund transfers between different parties, animal identification and tracking, and automated manufacturing and logistics control. Regarding RFID security, few issues are related to the data protection of the tags, message interception over the air channel, the eavesdropping within the interrogation zone of the RFID reader, specified in (Weis 2004) and (Sarma et al. 2002).

In general, many sub-tasks can be defined that contribute to provide this data protection or data security on the RFID at the tag level. Data encryption is one of the approaches to be stored on the tags. In doing so, the data can be protected in cipher text format instead of in clear text format. The data retrieved by any unauthorized readers will show no interests to attackers, unless they are able to decrypt all the information they received. Authentication between the RFID reader and the tags is another approach. It means normal information retrievals from the tags to the reader can be allowed to proceed, provided that authentication has been done before. It means that both the reader and the tags identify they are the right parties to exchange information.

We target to embed low-cost RFID tags into high market-valued brand name products in the meantime. These products may be garments, bags, shoes or accessories. If tags can be used to identify genuine or fake apparel products, data protection on tags is required. Therefore, tags do not merely act as a price tag during payment and checkout at retail stores, but also as a way for the authorities (i.e. the Customs and Excise Department) to identify any fake products. In addition, data protection on low-cost tags also serves to prohibit the activities of tags' eavesdropping and the cloning of tags. These activities are probably carried out by attackers who would like to produce forged products and embed cloned tags into the products.

In this paper, we will briefly describe related works on various RFID security or authentication schemes in section 2. Our proposed security scheme and authentication method is presented in section 3. Section 4 presents how proposed method used in apparel products. Finally, the conclusion and our future work are addressed in the section 5.

2. RELATED WORK

For low cost's tags, their resources and capabilities are limited. In contrast, the tag cost will be increased if tags have the ability to perform computation, encryption, or even to have a microprocessor on them. Some work below may involve light or heavy computations on tag side; therefore it may not be feasible to have data security and authentication when applying it on passive low-cost RFID Class 1 tags.

With *cryptography approach* in (Sarma-CHES2002), it is a simple and well defined algorithm to be implemented. Each tag should carry a particular vendor type of reader's public key and its unique private key. During reading, readers and tags may mutually authenticate each other with these embedded keys. Authentication between readers and tags can be achieved by a challenge-response technique. Eavesdropping can be prevented unless attackers find out the actual private key of each tag, but this is unlikely in a short period of time. Unfortunately, a low cost passive tag does not have the capability to perform these kinds of function above.

With *hash function/lock approach* in (Sarma-CHES2002), a reader defines a "Lock" value by computing *lock=hash(key),* where the key is a random key. This lock value is sent to a tag and the tag will store this value into its reserved memory location (i.e. as a Meta-ID value), and automatically the tag enters the locked state. To unlock the tag, the reader needs to send the original key value to the tag, and the tag will perform a hash function on that key to obtain the Meta-ID* value. The tag then has to compare the Meta-ID* with its current Meta-ID value. If both match, the tag unlocks itself. Once the tag in unlocked state, it can respond its identification number (i.e. the EPC code) to readers' queries in the forthcoming cycles. This approach is simple and straight forward to achieve data protection, i.e. EPC code in the tag is being protected. Only an authorized reader is able to unlock the tag and read it, then lock the tag after reading the code.

With *randomized hash lock approach* in (Weis et al. 2004), this is an extension of hash lock based on pseudo-random functions (PRFs). An additional pseudo-random number generator is required to embed into tags for this approach. Presently, tags respond to reader queries by a pair of values *(r, hash(ID_k || r))* where r is the random number generated by a tag, ID_k is the ID of the k^{th} tag among a number of tags in ID_1, ID_2, … ID_k, …, ID_n. For reader queries, the tag returns 2 values. One is the random number. The other is computing a hash value based on the concatenation (i.e. ||) on its own ID (i.e. ID_k) and r. Once the reader gets two values, it retrieves the current N number of ID (i.e. ID_1, ID_2… ID_n) from the backend database. The reader will perform the above hash function on each ID (from 1 to n) with r until it finds a match. When the reader finds a match, it means the reader is able to identify that tag k is on its tag ID list (i.e. tag authentication). The reader will then send the ID_k value to the tag for unlocking it. Once the tag is in an unlocked state, the reader can get its EPC code in the next reading cycle. This scheme might not be practical or feasible as if it requires a longer process for the reader to do computations and find a match in order to unlock the tag. It is time consuming and might

be suitable when there are a relatively small number of tags. However, this scheme may have a scalability problem when the number of tags increases enormously. Another problem is that the cost per tag will be higher due to the presence of a random number generator on a tag. Thus, it is not a suitable solution for low-cost tags that will apply on apparel products.

Juels has proposed an idea of so called one-time "yoking-proofs" protocol using minimalist MACs. This protocol allows a pair of tags to construct a one-time proof that they have been read simultaneously (Juels 2004). As a result, this enables these two tags to authenticate themselves in pairs to readers. A yoking-proof can be used in application where it can provide evidence that one product was dispensed with another tagged product leaflet. For example, it can be used in pharmaceutical distribution application. One RFID tag might be embedded in the container for the medication, while another is embedded in an accompanying leaflet, and the leaflet is supposed to describe its side-effects of this medication. RFID authentication is used here on product packaging to deal with counterfeit drugs in (FDA 2004). Juels has also proposed another heuristic approach for achieving of the tag authentication purpose in (Juels 2005). Tag authentication is mainly to identify if this EPC tag is a cloned tag or not. However, EPC tags are themselves weak authenticators subject to cloning attacks. Attackers are easy to clone a tag when they possess a RFID reader and read the EPC value of this tag.

In (Duc et al. 2006), the authors try to propose a solution in handling security and privacy issue. They focus on the EPCglobal Class-1 Gen-2 RFID passive tags, which supports simple cryptographic primitives, like Pseudo-random Number Generator (PRNG) and Cyclic Redundancy Code (CRC). Like our approach, the authors first proposed an encryption/decryption scheme. They use the CRC as a hash function to encrypt the EPC code; and do an XOR operation with a shared random session key. The session key is generated by the PRNG on a passive tag. Initially a shared same *seed* is used at both tag side and reader's backend server side; so that the PRNG is able to generate a random number based on the input of this seed. This random number is used as the *session key* (K_i) for one operation only. For next operation, the previous session key will be used as an input to PRNG for producing the next new session key. The process in creating a new session key should be synchronized between the backend server side and the tag side, otherwise, the tag authentication by server side will have problem if in case the session key is out of synchronization. Generally speaking, the proposed solution is a good approach to enhance the security and privacy issue. However, this scheme still has its weakness. For each tag authentication by the backend server side, the operation complexity for each authentication requires O(N)*O(CRC); where N is the number of data records, i.e. data tuple (EPC, K_i), in the backend database. The authors only handle RFID "read tag" operation for the proposed solution. There is no handling on every "write tag" operation.

3. PROPOSED SCHEME FOR LOW-COST CRYPTOGRAPHY AND AUTHENTICATION ON PASSIVE RFID TAGS

The contents of RFID tags mainly contain a standard EPC code, and we name it as EPC tags as well. For tag security and preventing the tags from cloning, we firstly propose a heuristic Jigsaw encoding scheme. This simple scheme helps encrypt an EPC code (EPCglobal) into a pseudo-EPC code. The pseudo-EPC code may look like a random code that an attacker may not be able to reverse it into a valid EPC code.

Jigsaw encoding scheme serves a purpose of hiding a valid EPC code. Product manufacturers may have deployed pseudo-EPC codes on tags before distributing them. Attackers, even with a RFID reader, can only make cloned tags with these pseudo-EPCs. The encryption to protect EPC data on tags is completed. In next step, we then propose a simple tag authentication scheme using a one-way hash lock function specified in section II. With tag authentication, we can verify if this tag is cloned or not.

Jigsaw encoding scheme will not change the standard of RFID readers and tags. An authorized reader is regarded as a connection to a backend system or an authorized computer system or point-of-sale (POS) terminal. These authorized devices are able to convert pseudo-EPC codes into a valid EPC code and points to a right entry in a database in the backend for retrieving relevant product information. The valid EPC codes are even not necessary to be displayed on POS terminals. The reader that an attacker possesses is also a valid reader, but it is not an authorized reader. Thus, it can only read the pseudo-EPC codes on tags, and not be able to decode them within a short time.

3.1 Encryption / Decryption Scheme with Jigsaw Encoding

Our key idea in our proposed scheme is using a jigsaw concept to encrypt the EPC code. EPC code is a string of hexadecimal digits. For example, the present EPC code for a Class 1 Generation 1 tag is in 64-bit length. We might consider an EPC code is scattered in an M x N matrix that it looks like a jigsaw. Here if we consider the 64-bit EPC code, then the matrix can be in 8 x 8 (depending on your implementation). For 96-bit EPC code, it might be a 12 x 8 matrix for simply understanding. For decoding the pseudo-EPC code, we retrieve it in a deterministic way to trace back the original EPC code, since the product manufacturer has known the scattering way onto the matrix.

There are many ways to scatter the original EPC bit-strings onto a matrix, and shifting bits in either left or right direction is only one of the examples we show in our implementation. Without the knowledge of the scattering way, it is not feasible for attackers to resolve it in a short time at least, since this involves a permutation of 64 ways. Other than this, the product manufacturer can alter the scattering way after a certain period. Fig. 1 below shows the process of generating encrypted data and the way of reverting it to plaintext data.

The process of generating encrypted data is described as follows. A 64-bit EPC code is viewed as 8 x 8-bit hexadecimal codes. It is the same treatment for splitting a 64-bit private key code. Each 8-bit of EPC sub-code does XOR operation with a same length of the private key. Combining the 8 x 8-bit sub-codes after XOR, this will produce a 64-bit intermediate code. Then, by shifting n-bit of positions in either right or left direction, a 64-bit pseudo-EPC code is obtained. The encryption process for a 96-bit EPC code is similar as the 64-bit EPC code. Furthermore, the decryption process can be easily done by reversing the encryption process.

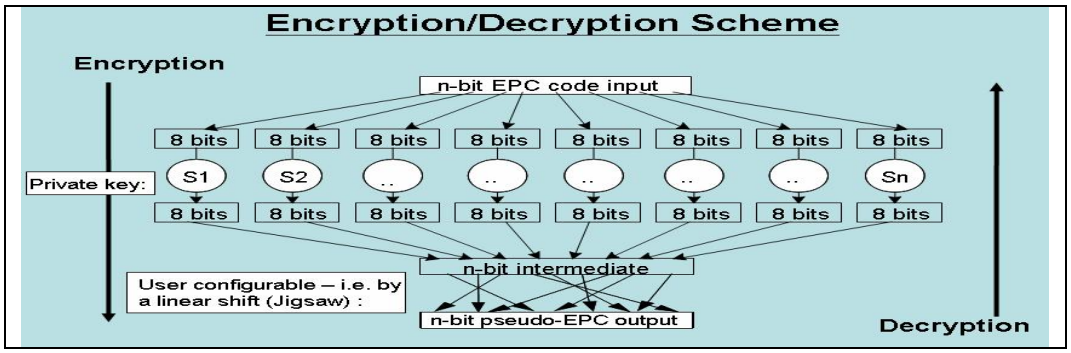

Fig 1. Proposed Encryption/Decryption Scheme

3.2 Hash Lock Function for Tag Authentication (Locking/Unlocking)

We still propose the use of a one-way hash lock function to generate a lock value, i.e. lock=hash(key). The 'key' value will be discussed later in the implementation stage on how to derive it from an EPC code, while the 'lock' value in fact becomes the PIN number on a tag. As PINs in Class 1 Generation 1 EPC tags are only 8 bits in length, an attacker can resolve it in a probability of $1/2^8$, i.e. 1/256 whatever the lock value we have. However, some EPC tags currently defend against PIN-guessing by temporarily disabling a tag when multiple incorrect PINs are attempted (RFID June 2003). For general implementation, the tag will be disabled in 1 hour if a wrong guess of PIN is attempted in (Alien). The tag will not respond reader's commands when it is in the disabled state. For Class 1 Generation 2 tags, PIN lengths are proposed in 32 bits length and thus the harvesting of tag PINs will be highly impossible.

There is no explicit functional support for tag authentication in current standard. Therefore, we test the lock value by providing a valid PIN using a valid EPC-tag command, and we refer to such command as "Unlock" to check if the testing is successful or not.

In a system with N tags, let the integer i (with $1 <= i <= N$) denote the unique index of an EPC tag. Let us denote the EPC identifier, i.e., the unique RFID readable hexadecimal string for tag i, by T_i. Let P_i be a k-bit PIN for tag i; and we calculate this PIN as a lock value by doing a hash function on T_i. Let r <--- Unlock(P) denote the execution of Unlock using PIN P. We assume that an EPC tag replies with A = "ack" if the Unlock command is successful; otherwise it returns an explicit indication of failure.

A tag authentication protocol is illustrated in Fig. 2. "Tag → Reader" or its reverse indicates a data flow from entity A to entity B, while "Reader" indicates an operation performed locally by an authorized POS terminal which is connected to a reader. A more clearly overall picture of the workflow process above can be viewed in Fig. 3.

In this work, our new technique is targeted to apply on a passive and low-cost tag. But techniques given in reference (Sarma-CHES2002) and (Weis et al. 2004) are not applicable on a passive and low-cost tag, this is the main difference. Techniques in these two references require an intelligent tag which is able to memorize its state, calculate a hash value, or generate a random number, etc. However, these functions will not be

present in a passive and low-cost tag. The passive tag can be regarded as a dummy storage device, where it only stores data and maintains a reserved area for user to place some secret information and retrieve it later. With this limitation on the passive tag, our new technique is designed to adapt it and all the decisional and computational work should come from the reader side.

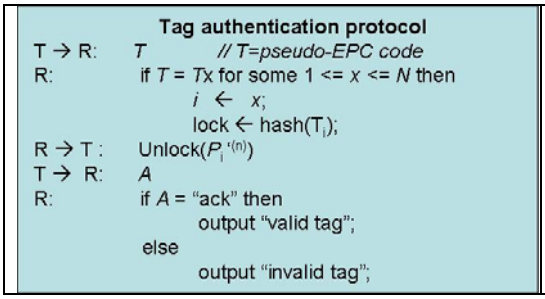

Fig 2. Tag authentication protocol

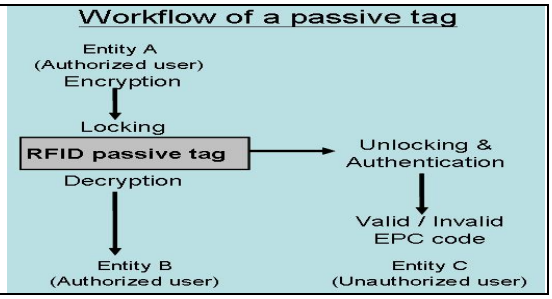

Fig 3. The workflow of a tag

4. IMPLEMENTATION OF OUR PROPOSED SCHEME

Firstly, a pseudo-EPC code given a valid EPC-tag is prepared. A valid EPC tag is assumed to have a unique EPC code already written on the tag. Since an EPC code is a unique ID number, we employ the use of public key and generate a random private key. The public key string and the private key string are non- disclosure and stored in a database for later retrieval by application program. Thus the $Key_{(i)}$ is used to hide the actual EPC code. Even an unauthorized reader can only read this value, it is no way to decode its public key if an attacker does not know the random private $key_{(i)}$. For line 4 & 5 illustrated in Fig. 4, it is only one of the possible ways for scattering the code along the matrix; here it just shifts the code to left side by n bits. Once all the required pseudo-EPC codes have been prepared on tags, they can be distributed out to apparel retail store where we can attach or embed those tags on apparel products. The following algorithm, Fig. 5, shows an authorized reader to read a tag T_i and decode the pseudo-EPC code into a real EPC code.

Let $K_{pu(i)}$ ← EPC code be a Public Key string (e.g. 96-bit) ; //for Tag (T_i)

1 $K_{pr(i)}$ ← set random Private Key (e.g. 96-bit);
2 $key_{(i)}$ ← $K_{pu(i)}$ XOR $K_{pr(i)}$; //use $key_{(i)}$ to hide info of $K_{pu(i)}$
3 lock ← hash($key_{(i)}$); //hash lock value for authentication
4 shiftLeftPos ← n;
5 $key_{(i)}$* ← shiftLeft($key_{(i)}$,n) //pseudo-EPC code generated by
//shifting bits to left in n for example.
6 write **$key_{(i)}$*** into tag's memory (i.e. pseudo-EPC code); //output
7 write **$lock_{(i)}$** value into reserved tag memory; //output

Fig 4. Prepare a pseudo-EPC code for a tag

1 Read tag data: key* //input is a pseudo-EPC
2 $key_{(i)}$ ← shiftRight($key_{(i)}$*,n); //convert back to an intermediate $key_{(i)}$
3 $K_{pu(i)}$ ← $key_{(i)}$ XOR $K_{pr(i)}$ //retrieve private key from backend database, calculate K_{pu}
4 display $K_{pu(i)}$ as the EPC-code read! //display EPC-code for demonstration purpose.

Fig 5. Perform a "read" operation on a tag.

In case, we need to verify any counterfeit tags. The algorithm, shown in Fig. 6, is carried out for recalculating the hash value, i.e. the lock value. Then it was tested by performing

the “Unlock” operation mentioned in the tag authentication protocol above.

1	Read tag data: key* //input, i.e. a pseudo-EPC
2	$key_{(i)} \leftarrow shiftRight(key_{(i)}*,n)$; //convert back to $key_{(i)}$
3	$lock_{(i)}* \leftarrow hash(key_{(i)})$; //re-compute $lock_{(i)}*$
4	Perform Unlock(P_i'); //testing the hash lock; return valid or invalid authentication

Fig 6. Perform a tag verification operation.

An example of a pseudo-EPC code and a lock value is generated and shown below:

Original EPC code (in 64-bit):	A5A5 8005 4824 6078
Intermediate code (after XOR):	5A5A 8005 5935 9F87
Pseudo-EPC (after shifting):	96A0 8196 8DA7 A196
Lock value (in 1-byte):	B

4.1 A Test Scenario for Tag Authentication

Figure 7 shows a test scenario of tag authentication for an apparel product.

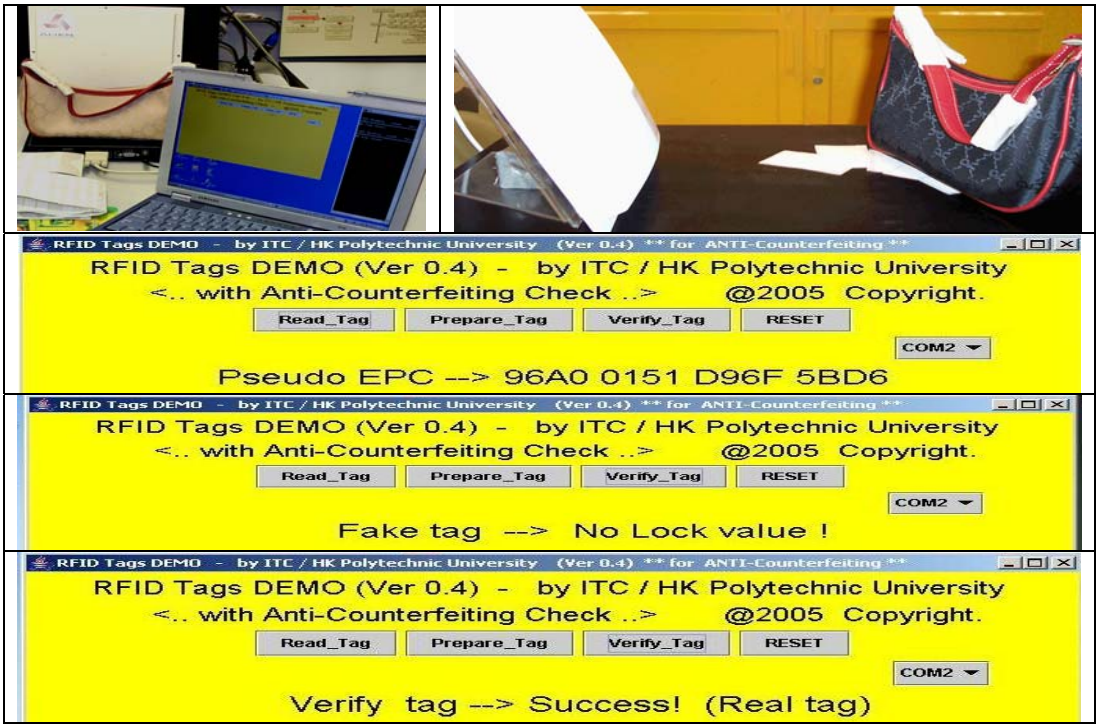

Fig 7. Top left: Equipment setting (Notebook PC with anti-counterfeit checking application, RFID antenna, RFID reader, and a tag embedded inside a bag).
Top right: Distance setting between a RFID antenna and a bag (about 30cm).
Second row: Anti-counterfeit checking application reads the bag and displays the encrypted EPC code (pseudo-EPC) of the embedded tag.
Third row: If the tag is not genuine, the checking application will show that this is a fake tag with possible reason.
Bottom row: If the tag verification is done properly, the tag inside the bag is genuine, and the apparel product is said to be genuine.

4.2 **Security Discussion**

Our proposed technique tries to protect the original EPC code by hiding it and writing a pseudo-EPC code on a passive RFID tag. Original EPC code could be recovered by attackers if and only if they are able to discover the value of private key and the number of bit position to shift. But this is unlikely or could be in much greater effort to achieve this. As a second protection to the EPC code, we can use random distribution of the intermediate code spreading over the matrix, instead of using a fixed bit-shifting technique. In doing so, we do need to find a way to remember or know the way on how to retrieve the particular way of random distribution.

There are two cases occurred when dealing with cloned tags with exact content. As we know, an attacker is easy to copy the pseudo-EPC code from an original tag to a new tag, and make this new tag as a cloned tag. First, a cloned tag can be easily identified if it is detected that there is no lock value, i.e. the cloned tag itself is not locked. Second, even the cloned tag is locked; our proposed authentication scheme is performed to check against its locked value. If we cannot unlock the tag, then this is not the original tag we have before. It is because the attacker does not know how to compute the lock value by a hash function, and also the private key string is supposed non-disclosure to an attacker. Therefore, this technique works in combating the counterfeiting problem, especially identifying counterfeiting products with cloned tags.

5. CONCLUSION AND FUTURE RESEARCH

In the meantime, there is no real commercial implementation on tag security and authentication for RFID tags of Class 1 Generation 1, which are the most common tags used in the market so far. Jigsaw algorithm is designed to address these requirements, such that many potential customers, who really concern tag security, can make use of this design and deploy it quickly on their existing applications.

Furthermore, we have carried out the implementation process from a cryptographic perspective that meets both the data protection and authentication requirements. With these techniques, those attacks against the cloning of EPC tags can be identified; and verifications of counterfeit products can be performed in terms of tag authentication. The outcome is positive and easy to implement. In our future research, a more robust security and authentication technique can be explored on Class 1 Generation 2 tags. The 96-bit of Gen-2 tag will be used in future, as it becomes the most updated and standard one, which will also replace the Gen-1 tag as well.

6. REFERENCES

[Weis 2004] Stephen Weis, *"RFID Privacy Workshop"*, IEEE Security and Privacy, March/April 2004.

[Sarma et al. 2002] Sanjay Sarma, Stephen Weis, and Daniel Engels, *"RFID Systems, Security & Privacy Implications"*, White Paper, MIT Auto-ID Center, November 2002.

[Alien] Alien Technology web site, http://www.alientechnology.com; and Reader Interface Guide.

[EPCglobal] EPCglobal web site, http://www.epcglobalinc.org/

[Sarma-CHES2002] Sanjay E. Sarma, Stephen A. Weis, and Daniel W. Engels; "*RFID Systems and Security and Privacy Implications*", In Workshop on Cryptographic Hardware and Embedded Systems (CHES) 2002, LNCS no. 2523, p.454-469, 2003.

[Weis et al. 2004] Stephen A. Weis, Sanjay E. Sarma, Ronald L. Rivest., and Daniel W. Engels; "*Security and Privacy Aspects of Low-Cost Radio Frequency Identification Systems*", Security in Pervasive Computing 2003, LNCS no. 2802, p.201-212, 2004.

[Juels 2004] A. Juels, "*Yoking-proofs for RFID tags,*" In PerCom Workshops, p.138-143, IEEE Computer Society, 2004.

[FDA 2004] United States Food and Drug Administration, "*Combating counterfeit drugs*" February 2004, http://www.fda.gov/oc/initiatives/counterfeit/report02_04.html

[Juels 2005] A. Juels, "*Strengthening EPC Tags Against Cloning*", Proceedings of the 4th ACM Workshop on Wireless Security (WiSE'05), p.67-75, Sept. 2005

[RFID June 2003] "*RFID, privacy, and corporate data*". RFID Journal, 2 June 2003, Feature article, http://www.rfidjournal.com on subscription basis

[Duc et al. 2006] D.N. Duc, J. Park, H. Lee, K. Kim, "Enhancing Security of EPCglobal Gen-2 RFID Tag against Traceability and Cloning," The *Symposium on Cryptography, and Information Security* (SCIS 2006), Japan, Jan 2006.

ACKNOWLEDGEMENTS

The authors wish to thanks the Institute of Textiles and Clothing, The Hong Kong Polytechnic University for providing the fund to support this study and Supply Chain & Logistics Technology Limited for providing technical support of RFID reader.

RESPONDENCE ADDRESS

Dr. Patrick C.L. Hui
Institute of Textiles and Clothing,
The Hong Kong Polytechnic University,
Hung Hom, Kowloon, Hong Kong
tchuip@inet.polyu.edu.hk