**SCIS 2007** The 2007 Symposium on Cryptography and Information Security
Sasebo, Japan, Jan. 23-26, 2007
The Institute of Electronics, Information and Communication Engineers

# A Lightweight Protocol Enabling Ownership Transfer and Granular Data Access of RFID Tags

Youngjoon Seo * Tomoyuki Asano † Hyunrok Lee * Kwangjo Kim *

**Abstract**— RFID(Radio Frequency Identification) is now being used in everything for economic feasibility and convenience. In contrast, RFID tags may infringe on user's privacy. A number of previous schemes exploiting hash function, symmetric cryptographic primitive like AES (Advanced Encryption Standard), asymmetric cryptographic primitive like ECC (Elliptic Curve Cryptosystem) are suitable for high-end RFID. In this paper, we propose a lightweight protocol for low-cost tags to make RFID tags widespread, which requires only one cryptographic primitive, a pseudorandom number generator. Under the strong assumption that all the channels are insecure, our protocol using a proxy for individual and the universal re-encryption has several advantages: (1) ownership transfer, (2) computational time in the back-end server to find the identifier of a tag, (3) untraceability against the compromising tags, and (5) data access authorization level-based service by the back-end server.

**Keywords:** RFID, Proxy, Scalability, Ownership Transfer, Authentication, Protocol, Untraceability

## 1 Introduction

RFID is recently becoming popular, and plays definitely an important role in moving on ubiquitous society due to deploying its convenience and economical efficiency; furthermore, RFID nowadays comes into the spotlight as a technology to substitute the bar code system since RFID can solve several problems in the bar code system: (1) to require line of sight for scanning, (2) no read/write capability including limited capacity for encoding information, (3) opportunities for human error, and more problems in [12, 13].

On the other hand, RFID is jeopardized from various attacks and problems as obstacles of widespread RFID deployment; attacks are spoofing, swapping, and DoS(Denial of Service) attack; problems are privacy, tracing, tag cloning, and computational overhead in back-end server due to a large number of tags. Table 1 shows that various countermeasures to protect against these attacks and to solve these problems have been proposed, which divided into different categories. Deactivation by permanent and temporary tags is analogous to power-off of personal computers due to the fear of being cracked. In other words, these can not be an eventual solution. On-tag cryptographic primitives and on-tag access control require high-end RFID tags. Low-cost is the most important factor to proliferate RFID technology into the billion of items. In this paper, we propose an off-tag access control mechanism [1] to proliferate low-cost tags based on universal re-encryption[2].

| Countermeasure | Example |
|---|---|
| Permanent tag deactivation | kill command[7], tag destruction |
| Temporary tag deactivation | Faraday cages, sleep/wake command |
| On-tag cryptographic primitive | stream ciphers, asymmetric or symmetric cryptographic algorithm[9] |
| On-tag access control | hash-based[15], pseudonym-based[1, 14], tree-based schemes[6, 16] |
| Off-tag access control | blocker[2, 4], noisy tag[5], proxy-based schemes[3, 11] |

Table 1: Countermeasures for preventing attacks in RFID systems

### 1.1 Notations

We summarize the notations for entities and operations in Table 2 throughout the paper.

## 2 How the proxy works

A proxy, $P$ is used for *personal usage* like RFID Guardian(GUARDIAN)[11]. $P$ is a reader which can be integrated into cellular phones, PDAs (personal Digital Assistants) or tiny portable device manages owner's tags; $P$ also enforces privacy policy desired by its owner using an access control list. In our proposed protocol, $P$ should exist around his own tags; so, the operat-

* Information and Communications University, International Research center for Information Security (IRIS), Information and Communications University (ICU), 103-6 Munji-Dong, Yusong-Gu, Daejeon, 305-714, Korea.
† Sony Corporation, 6-7-35 Kitashinagawa, Shinagawa-ku, Tokyo 141-0001, Japan.

[1] On-tag access control mechanism is located on the RFID tags themselves; in contrast, off-tag access control mechanism is taken care of by an external device in stead of the RFID tags[10].

[2] Please, See [14, 8, 18] to understand characteristic of universal re-encryption we do not handle it due to page limitation.

Table 2: Notations

| | |
|---|---|
| $R$ | RFID tag reader, or transceiver. |
| $T$ | RFID tag, or transponder. |
| $P$ | Proxy. |
| $S$ | Back-end Server. |
| $C$ | Ciphertext. |
| $C'$ | Re-encrypted ciphertext. |
| $ID$ | Identifier. |
| $M_1 \| \| M_2$ | Concatenation of messages $M_1$ and $M_2$. |
| $SK$ | Private key. |
| $PK$ | Public key corresponding to the $SK$. |
| $SK_M$ | Private key of M. |
| $PK_M$ | Public key of M corresponding to the $SK_M$. |
| $Cert_M$ | Certificate of M. |
| $Sig_M$ | Signature of M. |

ing range of $P$ works around 1 or 2 meters which is approximately from head to toe of the individual.

Juel(REP)[3]'s proxy and Rieback's Guardian meet four security properties; REP has tag acquisition, tag relabelling, tag simulation and tag release; GUARDIAN has auditing, key management, access control and authentication. $P$ has six functional security properties which are described in Figure 1; these properties in our protocol are a little different with REP and GUARDIAN. The description of each component is as follows:

- *Tag acquisition* : $P$ gets a new $SK$ corresponding to the $PK$ and $T$'s $ID$ from $S$; $P$ also gets $PIN$ from the previous tag owner's $P$. $P$ generates $C$, and then writes $C$ and $PIN'$ into acquired $T$'s memory when $P$ acquires $T$.
- *Information management* : $P$ manages $T$'s $ID$, $SK$, $PIN$ and a server location for each $T$. $P$ inserts the record in a database when acquire $T$; $P$ deletes the record about $T$ when release $T$.
- *Relabeling* : $P$ relabels $T$ contents whenever the other devices try to write data into $T$ managed by $P$, which means that $P$ writes $C'$ into $T$.
- *Authentication* : $P$ checks whether the queried $R$ are an authorized party or an unauthorized party.
- *Access control* : If an authorized party sent query, then $P$ checks a data access authorization level and passes the proper message for level. $P$ which has an access control component can considers three cases: which $R$, which $T$, which circumstances like GUARDIAN05 (See more details in [3]).
- *Tag release* : An owner of $T$ releases $T$ when the owner of $T$ does not want to keep his $T$ any more; that is, ownership transfer happens.

## 3 Our Proposed Protocol

We propose an off-tag access control mechanism using an external device. Off-tag access control provides a chance to be widespread with low-cost tags since the external device takes care of almost high-cost computations instead of $T$.

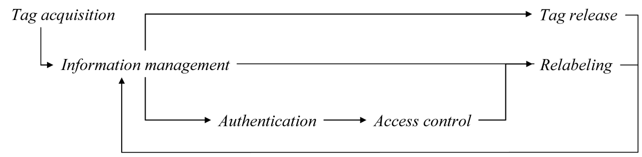


Figure 1: $P$'s process. An arrow in this figure represents a state transition.

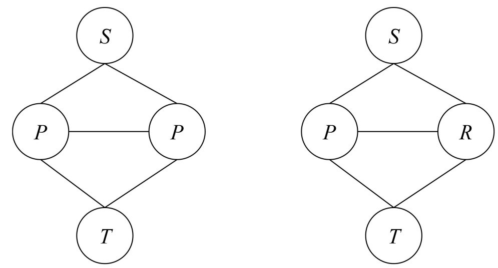


Figure 2: This figure shows all possible channels in our protocol. The solid line represents an insecure channel.

$T$ checks the first attack and second attack by itself in Saito *et al.*'s work (SAITO)[8] which is one of the on-tag access control scheme. Exponential computation is needed to check the second attack; however, it is big overhead on $T$. SAITO's protocol checks only the contents written in $T$ not to authenticate $R$; that is, anybody can get $T$'s information from $S$ upon receiving $C$ from $T$ while we authenticate $R$ exploiting the external device on behalf of $T$.

### 3.1 Initialization and Assumption

We assume that 1) PKI(Public Key Infrastructure) is established, 2) one proxy manages only one tag, 4) proxy is within backward channel which is $T$'s operating range, and 5) all channels are insecure. The possible channels are depicted in Figure 2.

$P$ has four database fields: Private key, Tag identifier[3], PIN, Server Location for each tag; Server Location field for each tag can contribute to reducing the back-end server's work. In our protocol, the back-end server has to find a server location if $SL$ is a NULL value where $SL$ denotes a server location for $T$. $P$ has also an access control list. An Example of an access control list is described in Table 3.

$T$ has a pseudorandom number generator and memory storages to store $PIN$ and $C$; $C$ is based on El-Gamal encryption algorithm. Any other cryptographic primitives like hash or symmetric or asymmetric algorithm do not need.

The owner of $T$ is defined by that a person who carries and owns a proxy and all tags which is managed by the proxy.

$S$ has six database fields: Private key, Public key, EPC,

[3] $ID$, pseudo-EPC, tag identifier, and $m$ are the same meaning in our protocol.

| Action | Source |
|---|---|
| Pass level **A** | List of readers which have authorization level **A** for some tags |
| Pass level **B** | List of readers which have authorization level **B** for some tags |
| … | … |
| No answer | The others |

Table 3: Access control list. **A** and **B** are used to represent $R$'s data access authorization level. $S$ can transfers fine granular information of $T$ based on granular data access authorization level; the degree of level depends on the system designer.

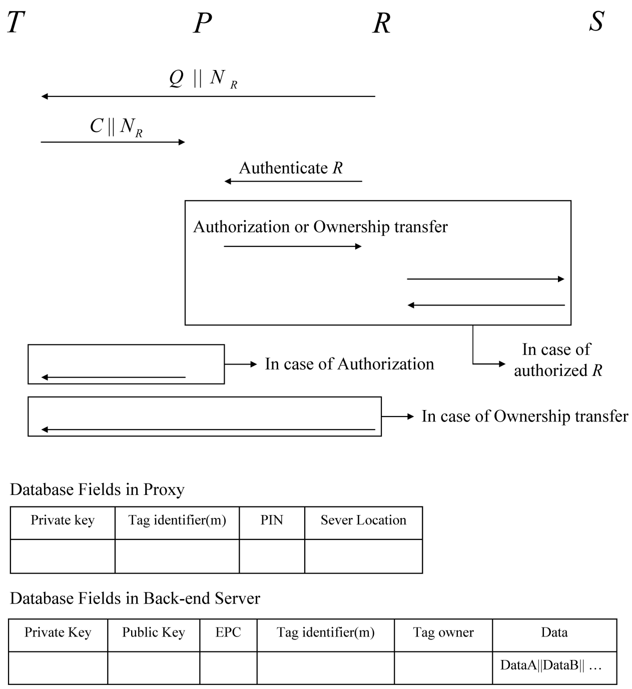


Figure 3: Our Proposed Protocol

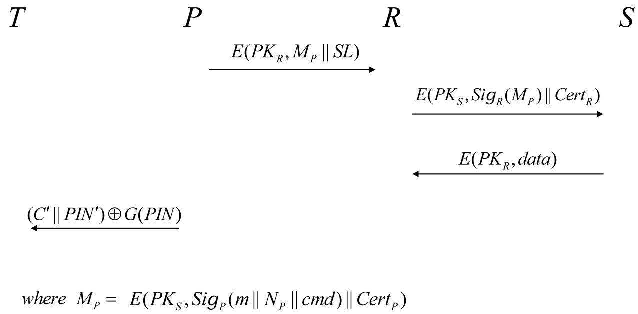


Figure 4: Our Protocol for Authorization

Tag identifier, Tag owner and Data; $SK$ and $PK$ can be generated and managed by $S$ or the other trusted entities since $R$ does not send messages included $SK$ or $PK$, Tag owner field is used for ownership transfer, Data field supports fine granular data access authorization level.

### 3.2 Protocol Description

Our protocol is shown in Figures 3, 4 and 5; Figure 3 shows our protocol, Figure 4 shows our protocol for authorization, Figure 5 shows our protocol for ownership transfer.

Our overall protocol works as follows:

**Step 1** $R$ sends $Q$ query and random nonce $N_R$ generated by $R$ to $T$.

**Step 2** $T$ sends $C$ and $N_R$ to $P$. $P$ decrypts $C$ with private key $SK\ x$.

**Step 3** The way to communicate between $R$ and $P$ is described in detail in [11]. In our protocol, $R$ sends its information like $Sig_R(N_R)||Cert_R$ to $P$ using a variety of out-of-band or in-band means (See more details in [3]). $P$ checks whether $R$ is authorized or not using an access control list, and checks data access authorization level in case of authorized $R$. As another case, ownership transfer happens in Step 4; ownership transfer is unusual case, so it require human interaction to do ownership transfer.

**Step 4** Protocol descriptions for authorization and ownership transfer is handled with in each protocol. For an unauthorized $R$, $P$ sends random value to $R$, which can not give a chance for the adversary to distinguish the tag from the other tags.

**Step 5** In case of authorization protocol, $P$ relabels $T$'s contents while $R$ relabels $T$'s contents in case of ownership transfer protocol; the detail description is described in each protocol.

Nonce($N_R$ and $N_P$) in our protocol is to ensure that old communications cannot be reused in replay attacks. Nonce can be time-variant or generated with enough random bits which ensure a probabilistically insignificant chance of repeating a previously generated value.

Our protocol in case of authorization works as follows:

**Step 1** $P$ sends $E(PK_R, M_P||SL)$ to $R$ where $M_P$ denotes $E(PK_S, Sig_P(m||N_P||cmd)||Cert_P)$; $SL$ denotes a server location for $T$, $N_P$ denotes a random nonce generated by $P$, $cmd$ represents an authorization level, and $m$ denotes a pseudo-EPC($T$'s $ID$) in our protocol. We recommend to use the pseudo-EPC rather than EPC ([17] states the reason for that)

**Step 2** $R$ decrypts $E(PK_R, M_P||SL)$ with $R$'s private key $SK_R$. $R$ gets a server location, and sends $E(PK_S, Sig_R(M_P)||Cert_R)$ to $S$ which is same with the server location.

**Step 3** $S$ decrypts $E(PK_S, Sig_R(M_P)||Cert_R)$, $Cert_R$, $M_P$, $Cert_P$ with $S$'s private key. $S$ finds out the identities of $P$ and $R$, $T$'s $ID$, and an authorization level. $S$ checks where $P$ is the owner of $T$ or not. If $P$ is the owner of $T$, then $S$ checks

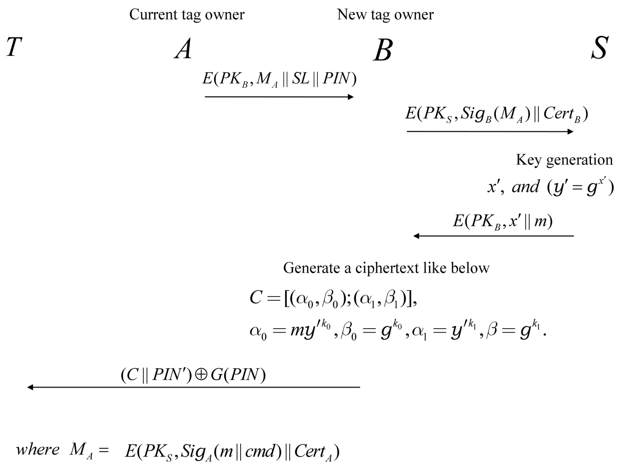


Figure 5: Our Protocol for Ownership Transfer

the authorization level of $R$ for $T$. For example, In case that an authorization level is **A**, $S$ sends $E(PK_R, DataA)$ to $R$; In case that an authorization level is **B**, $S$ sends $E(PK_R, DataA||DataB)$. The degree of an authorization level depends on the system designer. If $P$ is not the owner of $T$, $S$ sends a random value to $R$ to provide indistinguishability.

**Step 4** $P$ computes $G(PIN)$ and generates $PIN'$ where $G$ is a pseudorandom number generator and $PIN$ is used for a `seed`; $G$ is used for matching the bit size of $G(PIN)$ and $(C'||PIN')$. $P$ selects a random encryption factor $r' = (k'_0, k'_1) \in Z_q^2$, re-encrypts $C$ to $C' = [(\alpha'_0,\beta'_0); (\alpha'_1,\beta'_1)] = [(\alpha_0\alpha_1^{k'_0},\beta_0\beta_1^{k'_0}); (\alpha_1^{k'_1},\beta_1^{k'_1})]$, and sends $(C'||PIN')\oplus G(PIN)$ to $T$; lastly, updates $PIN$ with $PIN'$.

$T$ computes $G(PIN)$ with $PIN$ which is in $T$'s memory, performs $\oplus$ operation ($G(PIN)$ generated by $T$ with $(C'||PIN') \oplus G(PIN)$ received from $P$), and can get $C'$ and $PIN'$; lastly, $T$ updates $PIN$ with $PIN'$ and $C$ with $C'$.

Our protocol in case of ownership transfer works as follows:

**Step 1** $A$ sends $E(PK_B, M_A||SL||PIN)$ to $B$ where $M_A$ denotes $E(PK_S, Sig_A(m||cmd)||Cert_A)$, $A$ denotes the current tag owner, $B$ denotes the new tag owner, and $cmd$ represents `ownership transfer command`.

**Step 2** $B$ decrypts $E(PK_B, M_A||SL||PIN)$ with $B$'s private key. $B$ gets a server location and $PIN$, and sends $E(PK_S, Sig_B(M_A)||Cert_B)$ to $S$.

**Step 3** $S$ decrypts $E(PK_S, Sig_B(M_A)||Cert_B)$, $Cert_B$, $M_A$, $Cert_A$ with $S$'s private key. $S$ finds out the identities of and $A$ and $B$, $T$'s $ID$, and `ownership transfer command`. $S$ checks where $A$ is the owner of $T$ or not. If $P$ is the owner of $T$, then $S$ generates $SK$ and $PK$ corresponding to $SK$. $S$ updates previous key pairs with new key pairs for the tag and the previous tag owner with the new tag owner in the database. And then, $S$ sends $E(PK_B, x||m)$ to $B$. If $A$ is not the owner of $T$, $S$ sends a random value to $B$. Lastly, $B$ generate a new ciphertext.

**Step 4** $B$ computes $G(PIN)$ and generates $PIN'$, selects a random encryption factor $r = (k_0, k_1) \in Z_q^2$, generates $C = [(\alpha_0,\beta_0); (\alpha_1,\beta_1)] = [(my'^{k_0},g^{k_0}); (y'^{k_1},g^{k_1})]$, and sends $(C||PIN')\oplus G(PIN)$ to $T$; lastly, $B$ updates $PIN$ with $PIN'$.

$T$ computes $G(PIN)$ with $PIN$ which is in $T$'s memory, performs a $\oplus$ operation ($G(PIN)$ generated by $T$ with $(C'||PIN')\oplus G(PIN)$ received from $P$), and can get $C'$ and $PIN'$; lastly, $T$ updates $PIN$ with $PIN'$ and $C$ with $C'$.

After the ownership transfer protocol, $B$ should perform operation over the secure channel so that $PIN'$ is not eavesdropped by $A$ when writing a new ciphertext. Nevertheless, it can be easily performed with secure way since $P$ can control its operation range. For example, $P$ writes $PIN'$ and $C$ with less than one centimeter operating range by physical contact.

## 4 Security and Performance Analysis

In this section, we check whether our protocol guarantees security requirements as followings: ownership transfer, scalability, privacy, protection against several threats which are tracing spoofing, swapping, cloning, DoS, two attacks and the garbage value which is mentioned in [8, 18].

- ***Protection against tracing.*** $T$ sends different message at any time $R$ sends a query. $C$ and $C'$ is indistinguishable (See [14]), and $P$'s `write command` is secure provided that the adversary doesn't know $PIN$. Even if the adversary gets $PIN$ under tampering $T$, the adversary have to be within 1-2m to trace $T$ at all time while the other almost all the previous protocols in the literature easily can be traced under tampering $T$. In addition, `write command` by physical contact guarantees updating $PIN$ securely.

- ***Protection against cloning and spoofing.***

  Cloning $T$ and spoofing $R$ are meaningless since $P$ maintains a private key and an access control list for each tag.

  Spoofing $T$ is also meaningless. For example, $T$ doesn't have a way to check whether `write command` some devices sent is authorized or not; since the adversary doesn't have any gains, the adversary does not try to spoof $T$. The adversary's `write command` make $T$ replace $PIN$ with $PIN_A$ where $PIN_A$ is the generated by the adversary; but, $P$ also checks $PIN_A$ and can writes re-encrypted ciphertext generated by $P$ with the $PIN_A$.

- ***Privacy.*** We provide privacy since $C$ emitted is provably secure since it is based on UR[14]. As another way to provide privacy, pseudo-EPC as $T$'s $ID$ should be used(See the more details in [17]); $S$ has EPC and Tag identifier field to use pseudo-EPC. We support data access authorization level-based service, which enhances privacy for individual.

- ***Protection against DoS.*** DoS attack can cause battery consumption of $P$, which is one bing problem when using the battery-powered device to protect owned $T$.

- ***Ownership transfer.*** We described the protocol for ownership transfer. Ownership transfer is one of the advanced security requirements; but, Monar *et al.*[6] supports sophisticated ownership transfer to the best of our knowledge.

- ***Protection against swapping.*** Swapping attack is one of the vulnerabilities on UR. In our protocol, we protect from swapping attack through $PIN$.

- ***Protection against two attacks and the garbage value in UR.*** $P$ writes new $C$ into $T$ whenever the other devices try to write $C$, which means that $T$ has always $C$ generated by $P$ in $T$'s memory unless $P$'s battery is totally consumed. Sleep / wake command can defend against two attacks and the garbage value even in case that $P$'s battery is totally consumed.

- ***Scalability.*** Since $P$ sends $m$ with encrypted form to authorized $R$ which forwards message received to $S$, the complexity of tag identification on $S$ is $O(1)$. In other words, $S$ does not need computations related to non-relevant $T$, which means our protocol is *completely scalable.*

- ***Cost.*** $T$ requires only one lightweight cryptographic primitive, a pseudorandom number generator, and re-writable memory to store $C$ and $PIN$. Consequently, our protocol can be implemented with reasonable low-cost.

## 5 Comparison with Related Work

Selective RFID jamming[10] makes a signal jam up the airwaves under lots of an unauthorized $R$'s queries while an external device just re-encrypts a new valid $C$ in our protocol. In addition, the use of jamming signal is legally questionable.

REP and GUARDIAN send $T$'s secret value with unecrypted form, which is insecure since REP and GUARDIAN give the adversary a chance to eavesdrop secret values while our $P$ does not reveal $T$'s secret information.

SAITO has several weaknesses: 1) big overhead on $T$, 2) tracking with only eavesdropping within forward channel, 3) no $R$ authentication mechanism, 4) allowing swapping attack which is venerability on UR. Unlike SAITO, we resolve all the problems of SAITO using $P$ and $PIN$.

Tree-based protocol(MSW)[6] proposed by Molnar *et al.* supports ownership transfer. In contrast, in MSW, $T$ needs lots of computation time, communications cost, and memory storages since the number of tags in RFID systems are expected with uncountable number of tags; in addition, some form of tracking is possible under compromising a tag (Dimitriou[16] explains how this tracking is available.).

## 6 Concluding Remarks

The proxy is a compact powerful device, used for personal usage, and around individual person in RFID-tagged environments; moreover, the proxy provides individually a chance to enforce security policy. Our protocol is different approach with previous proxies [3, 11]. For example, our proxy has six functional components: *Tag acquisition, Information management, Relabeling, Access control, Authentication, Tag release.* As another example, our proxy supports granular data access and maintains Server Location field which makes readers connect directly the appropriate back-end server. In the other previous protocols, the back-end server has to do some of extra works to find the proper server which has the server location for tags. In other words, we alleviate the work in back-end servers.

In this paper, we propose a lightweight authentication protocol, which can contribute designing low-cost RFID tags since RFID tags needs only one cryptographic primitive, a pseudorandom number generator; a pseudorandom number generator is used only one time per session.

Our protocol has several security properties as followings: (1) ownership transfer, (2) granular data access (3) scalability, (4) untraceability, (5) privacy, protection against several attacks which are (6) spoofing, (7) cloning, and (8) swapping; (9) we introduce a untraceable way even under compromising a tag; (10) we suggest the more fast way to find a server location. Consequently, we make sure that our contribution can contribute to make RFID deployment widespread.

## References


[1] Ari Juels, "Minimalist cryptography for low-cost RFID tags", *The Fourth International Conference on Security in Communication Networks – SCN 2004, LNCS 3352*, Sep. 2004, Amalfi, Italia.

[2] Ari Juels and John Brainard, "Soft Blocking: Flexible Blocker Tags on the Cheap", *Workshop on Privacy in the Electronic Society - WPES*, Oct. 2004, Washington, DC, USA.

[3] Ari Juels, Paul Syverson and Dan Bailey, "High-power Proxies for Enhancing RFID Privacy and Utility", *Workshop on Privacy Enhancing Technologies – PET 2005*, May-Jun. 2005, Dubrovnik, Croatia.

[4] Ari Juels, Ronald Rivest and Michael Szydlo, "The Blocker Tag: Selective Blocking of RFID

Tags for Consumer Privacy", *Conference on Computer and Communications Security - ACM CCS*, Oct. 2003, Washington, DC, USA.

[5] Claude Castelluccia and Gildas Avoine, "Noisy Tags: A Pretty Good Key Exchange Protocol for RFID Tags", *International Conference on Smart Card Research and Advanced Applications - Cardis, LNCS 3928*, Apr. 2006, Tarragona, Spain.

[6] David Molnar, Andrea Soppera and David Wagner, "A Scalable, Delegatable Pseudonym Protocol Enabling Ownership Transfer of RFID Tags", *Selected Areas in Cryptography – SAC 2005, LNCS 3897*, Aug. 2005, Kingston, Canada.

[7] EPCglobal Inc. Class 1 generation 2 UHF air interface protocol standard version 1.0.9. Referenced 2005 at http://www.epcglobalinc.com/standards technology/EPCglobalClass-1Generation- 2UHFRFIDProtocolV109.pdf.

[8] Junichiro Saito, Jae-Cheol Ryou and Kouichi Sakurai, "Enhancing Privacy of Universal Re-encryption Scheme for RFID Tags", *Embedded and Ubiquitous Computing – EUC 2004, LNCS 3207*, Aug. 2004, Aizu-Wakamatsu City, Japan.

[9] Martin Feldhofer, Sandra Dominikus, and Johannes Wolkerstorfer. "Strong authentication for RFID systems using the AES algorithm". *Workshop on Cryptographic Hardware and Embedded Systems – CHES' 04, LNCS 3156*, Jul. 2004, Boston, Massachusetts, USA.

[10] Melanie R. Rieback, Bruno Crispo and Andrew S. Tanenbaum, "Keep on Blockin' in the Free World: Personal Access Control for Low-Cost RFID Tags", *International Workshop on Security Protocols – IWSP'05*, Apr. 2006, Cambridge, England.

[11] Melanie R. Rieback, Bruno Crispo, and Andrew S. Tanenbaum, "RFID Guardian: A Battery-Powered Mobile Device for RFID Privacy Management", *Australasian Conference on Information Security and Privacy – ACISP'05, LNCS 3574*, Jul. 2005, Brisbane, Australia.

[12] "Navigating the New Era of RFID" Article in EPCglobal Canada Inc.

[13] Nigel Wood, "Global Supply Chain GTIN & RFID Standards II", *EPC Global Standards Development*, EPCglobal Canada, Oct. 14, 2004.

[14] Philippe Golle, Markus Jakobsson, Ari Juels and Paul Syverson. "Universal Re-encryption for Mixnets", *The Cryptographers' Track at the RSA Conference – CT-RSA, LNCS 2964*, Feb. 2004, San Francisco, California, USA.

[15] Stephen Weis, Sanjay Sarma, Ronald Rivest and Daniel Engels, "Security and Privacy Aspects of Low-Cost Radio Frequency Identification Systems", *Conference on Security in Pervasive Computing – SPC 2003, LNCS 2802*, Mar. 2003, Boppard, Germany.

[16] Tassos Dimitriou, "A Secure and Efficient RFID Protocol that could make Big Brother (partially) Obsolete", *International Conference on Pervasive Computing and Communications – PerCom 2006*, Mar. 2006, Pisa, Italy.

[17] Youngjoon Seo, Hyunrok Lee and Kwangjo Kim, "A Scalable and Untraceable Authentication Protocol for RFID", *The Second International Workshop on Security Ubiquitous Computing Systems – SECUBIQ 2006, LNCS 4097*, Aug. 2006, Seoul, Korea.

[18] Youngjoon Seo, "A Study on Scalable and Untraceable Authentication Protocol of RFID Tags" Master thesis, I.C.U., Jan. 2007.