A Scalable and Secure RFID Ownership Transfer Protocol

Mohammad Saiful Islam Mamun and Atsuko Miyaji
Japan Advanced Institute of Science and Technology (JAIST)
Ishikawa, Japan.
{mamun, miyaji}@jaist.ac.jp

***Abstract*—Ownership transfer in an RFID inventory system experiences many security and privacy oriented problems. We consider scenarios related to ownership transfer of RFID tags in a large inventory system. In this paper, we propose a new mutual authentication protocol from Ring LPN problem that leverages the reader authentication phase to incorporate Semi-Trusted Parties (STP) seamlessly in RFID ownership transfer protocol. Employing STPs could ease the ownership transfer process for the consumers in the remote location. More precisely, we introduce a new variant of Learning Parity from Noise (LPN) based mutual authentication scheme for efficient ownership transfer protocol where ownership of multiple tags can be transferred from one owner to another by taking advantages of an efficient homomorphic aggregated signature (HomSig) and pseudo-inverse matrix properties. To the best of our knowledge, this is the first RFID ownership transfer protocol from LPN problem that is secure, private and scalable under standard model.**

***Keywords*: Ownership transfer, Ring LPN, Pseudo-inverse matrix, Homomorphic aggregated signature.**

I. Introduction

Modern inventory systems often rely upon RFID tags to allow automatic identification of tagged objects where readers can read even thousands of unique RFID tags in a single snatch. A secure RFID inventory system often needs to transfer ownership of RFID tags. Once tagged objects passes through distributed supply chain from a manufacturer to a consumer, *ownership* of the objects could be transferred among consumers several times.

Protocols without trusted party, usually termed as *ownership sharing* protocol in [2], [8], [20], [22] allow sharing a secret key among owners (previous and new). It threatens owner's privacy, since former owners as well as current owners can track the same tag legitimately. On the other hand, protocols with trusted party in [11]–[13] create a bottleneck in the inventory system and overloads the trusted server. For instance, each owner needs to contact directly to a manufacturer (resp. the trusted server). The situation would become worse in case of simultaneous ownership transfer of multiple tags.

Therefore, our solution utilizes Semi-Trusted Parties (STPs) where consumers (resp. owner) do not need to contact the manufacturer (trusted main server) each time it needs to transfer ownership. In real life, we may think that main server is located at the manufacturer's *Head office* and all other STPs are located at remote sites such as *Regional offices*. STPs on behalf of a trusted server can anonymously monitor and verify ownership transfer process without revealing any secrets. Later STPs could forward the ownership data to other STPs, or to the main server. By the term *semi-trusted*, we mean that an STP is an online designated server that follows prescribed protocol correctly and communicates to the readers so that the communicating readers yields on a mutually satisfactory agreement.

HB-family protocols based on LPN assumption are booming as one of the attractive candidates for secure low cost EPC tags [3]–[6], [9], [14], [15]. due to its security against quantum adversaries, efficient computational time and memory requirement etc. In this paper, we have designed a novel ownership transfer protocol by modifying a recently proposed RFID authentication protocol based on Ring-LPN problem [16] where the secret key and other parameters are taken over the field $\mathbb{F}_2$. It allows us to seamlessly use the same parameters (as used in authentication protocol) in the aggregated signature scheme.

Consider an *inventory management* of a large *supply chain system* where *vendors* contribute goods or services to the next link in the chain. Usually each vendor (owner) holds several RFID tags. In order to manage ownership transfer among the vendors in a large supply chain, we employ, after effective customization, the HomSig scheme described in [17] where signatures generated by the vendors[1] can be aggregated by an STP. The scheme is secure under standard model. Similarly, several aggregated signatures of a vendor (owner) could be combined by any legitimate intermediate STP and later signatures will be verified by the trusted main server.

Main contribution. In this paper, we first modify the scheme in [16] in order to achieve a MIM-attack free mutual authentication protocol[2]. The protocol employs a STP to avoid the communication overload on the trusted main server. It supports ownership of multiple tags (to update ownership record of tags in the trusted server) of an owner to be transferred simultaneously. Unlike other authentication protocols for ownership transfer system, communications between the server and readers are assumed to be insecure and over inauthentic channel. Therefore, readers and servers are not identical. For construction, we adopt several new ideas such as:

[1] each signature is generated from several tags of a vendor or resp. owners.
[2] Authentication protocol in [16] is susceptible to MIM-attack [19] and do not support mutual authentication.

- Only main server (not STPs) is assumed to be secure and keys are shared among the entities accordingly.
- Applying exact (not decisional) version of the LPN problem in order to repel completeness error.
- Using Field-LPN problem described in [16] to cope with the parameters needed in homomorphic signature scheme.
- Employing a lightweight searchable encryption and signature scheme based on the properties of pseudo-inverse matrix between the readers (resp. owners) so that an STP can verify anonymously which owner is transferring ownership to whom.
- A lightweight homomorphic aggregated signature (HomSig) to forward ownership data to the main server.

II. Preliminaries

We omit some notations, security definitions and proof that we borrow from [9] due to space constraints. In this section, we only introduce the new notations and definitions used in this paper in Table I.

TABLE I
NOTATIONS USED IN THE PAPER

F_q	an extended field on $\mathbb{F}_2$
$\mathcal{R}_{cur}, \mathcal{R}_{new}$	current and New reader corresponding to a tag $\mathcal{T}$
$\mathbf{uid}_n$	secret ID of the New owner $\mathcal{U}_n$
s, s'	λ bit random binary vector generated by the reader
c_i	shared secret between the reader and the tag for any time instance i
T	a unique identifier of a tag $\mathcal{T}$ s.t., $\mathrm{T} \in F$
$\hat{T}$	ownership index stored in the main server (first index I_i of a tag $\mathcal{T}$ after a successful ownership transfer
X^+	pseudo-inverse of a matrix $X \in \mathbb{Z}_2^{n\times m}$ i.e., $X^+ \in \mathbb{Z}_2^{m\times n}$
S_c, S_n	$m \times n$ bit matrices, *secret* keys for $\mathcal{R}_{cur}$ and $\mathcal{R}_{new}$ respectively
P_c, P_n	$n \times n$ bit matrices, shared key between $\mathcal{R}_{cur}, \mathcal{R}_{new}$ and the intermediate server $\mathcal{S}_i$. Let $S_j \leftarrow X^+$, then $P_j := XX^+$
F^*	multiplicatively invertible elements of a field F
$\mathbf{wt}(\cdot)$	Hamming weight of any vector
$\lfloor x \rceil$	the nearest integer to x
w	parameter of the Bernoullli error distribution $\mathbf{Ber}_w$ s.t., $w = \lfloor \tau n \rceil$ where $\tau \in]0, 1/4[$
$\pi_j : \{0,1\}^\lambda \to F$	a mapping to F if $\forall s, s' \in \{0,1\}^\lambda, \pi(s) - \pi(s') \in F/F^*$ iff $c = c'$
Q, V	$n \times n$ bit randomly generated non-singular matrices by the reader

A. Assumption and System architecture

An inventory system described in this paper consists of a single legitimate trusted server called *main server*, a set of intermediate servers called STPs, a set of readers and their corresponding owners, and a set of tags (EPC class). Note that STPs are assumed to be *semi-trusted*[3] and are constituted by the Main Server. Each owner has a unique ID. A reader could be shared among owners, or owned by a RFID tag owner. Readers would be connected to the back-end intermediate STPs during ownership transfer. We introduce the inclusion of STPs for 2 reasons:

- In order to ease physical communication between the owners and a remote trusted server.
- To act as a witness between the current and new owner on behalf of the trusted server.

[3]A form of *honest-but-curious* attacker model. However, multiple STPs are not allowed to collude.

Main server stores all the data related to the tags in the database. Each tag has a unique identifier T used as a permanent key, an index $\hat{T}$ and a session key c. We assume index-owner tuple $[\hat{T}, \mathcal{U}_{cur}]$ in the server database is unique for efficient searching. Since an RFID tag is not tamper-resistant, its session key is refreshed after each i^{th} session completes successfully. For updating key, each tag authenticates its legitimate reader.

We assume a hierarchical architecture where tags are placed in the lowest level in the hierarchy and trusted main server is set at the highest level. Readers and STPs are located somewhere in between. Only the main server is assumed to be trusted while other STPs are considered to be *semi-trusted*. Imagine a situation in an inventory management system where manufacturer preserves the main server and delegates its task to STPs placed in different locations for consumer's convenience.

In case of updating the ownership data on the trusted main server, the current reader should not be considered as *honest* (too strong assumption). Because the malicious current owner could claim that he/she is still the current owner without performing the last step (Step-3) of the protocol. In this protocol, we consider the new reader (resp. owner) to be honest and hence is responsible to transfer ownership records to the STP. Meanwhile, new reader has to update the keys of the tags individually to finalize ownership transfer.

B. Useful Definitions

Definition 1. Field-LPN$_w^F$ problem in [16] states that it is hard to distinguish uniformly random samples in $F \times F$ from those sampled from $\Lambda_w^{F,c}$ for a uniformly chosen c and Hamming weight w. The (decisional) Field-LPN$_w^F$ problem is (t, Q, ϵ)-hard if for every distinguisher $\mathcal{D}$ running in time t making Q queries such that

$$\Pr\ [\mathcal{D}^{\Lambda_w^{F,c}} : c \xleftarrow{\$} F = 1] - \Pr[\mathcal{D}^{U(F\times F)} = 1]] \leq \epsilon$$

Definition 2. Let $\mathbb{G}, \mathbb{G}_T$ be bilinear groups of the order p. For any randomly chosen element $x \in \mathbb{Z}_p$ and a random generator $g \in \mathbb{G}$, the 1-generator q-strong Diffie-Hellman Problem is, given $(g, g^x, g^{x^2}, \cdots, g^{x^q}) \in \mathbb{G}^{q+1}$, to compute a pair $(g^{1/(x+c)}, c) \in \mathbb{G} \times \mathbb{Z}_p$.

Definition 3. In the Man-In-the-Middle (MIM) attack, adversary $\mathcal{A}$ is allowed to eavesdrop both the connections tag-reader and reader-server, making the tag and the reader believe that they are talking directly to the reader and the server respectively over a secure connection, when in fact, the entire communication is controlled by $\mathcal{A}$. Then, $\mathcal{A}$ interacts with the server to authenticate. The goal of the attacker $\mathcal{A}$ is to authenticate successfully in Q rounds. $\mathcal{A}$ is successful if and only if it receives accept response from all Q rounds.

Definition 4. A *homomprphic aggregated signature* is defined by the following algorithms:

- $\mathsf{Kgen}(1^\lambda, m)$ On input security parameter λ and $m \geq 1$, it outputs (pk, sk) where pk is the public verification key

and sk is the secret signing key. Here m is the dimension of the vector space.
- $\mathsf{Sign}(sk, \mathbf{uid}_c, \mathbf{uid}_n, \mathrm{T}, \hat{T})$ On input secret key sk, current and new owner IDs $\mathbf{uid}_c, \mathbf{uid}_n$, a set of tag ID and index $\{\mathrm{T}, \hat{T}\}$, it outputs a signature Σ.
- $\mathsf{CombSign}(pk, \mathbf{uid}_c, \mathbf{uid}_n, \hat{T}_i, \Sigma_i)$ Given the public key pk, Owner IDs, a set of tag index $\hat{T}^{(i)}$ and their signature Σ_i, it outputs a new aggregated signature Σ.
- $\mathsf{VerSign}(pk, \mathbf{uid}_c, \mathbf{uid}_n, \mathrm{T}^{(i)}, \Sigma)$ Based on the public key pk, a set of tag ID and index $\{T_i, \hat{T}_i\}$ and a signature Σ, it can verify the signature and outputs 0 (reject) or 1 (accept).

Definition 5. A searchable encryption can be defined by the following algorithm:
- $\mathsf{Kgen}(m, n)$ On input the size of the matrix, it outputs a pair of keys (P_c, S_c) where P_c is public key and S_c is secret key.
- $\mathsf{Enc}(P_c, Q)$ Given a challenge matrix $Q^{n\times n}$ and public key P_c, it generates ciphertext $E := \mathsf{Enc}(P_c, Q)$.
- $\mathsf{TDoor}(S_c, Q)$ This algorithm takes secret key S_c, challenge matrix Q and outputs a trapdoor $T := \{C, D\}$ correspond to $\{S_c, Q\}$.
- $\mathsf{Test}(P_c, E, T)$ On input P_c, trapdoor T and ciphertext E, it proves whether T and E are generated from the same Q and outputs 0 (reject) or 1 (accept).

Definition 6. *Subset-sum problem* (SSP) is to take decision whether summation of subset of a given set of integers $L := \{a_1, \cdots a_n\}$ s.t., $a_i \in \mathbb{Z}_p, 1 \leq i \leq n$ is $t \in \mathbb{Z}_p$. Let $t = x_1a_1 + \cdots + x_na_n$ for a binary vector $X = \langle x_1, \cdots, x_n\rangle$ s.t., $x_i \in \{0, 1\}$. Then given L and t, it is hard (NP-complete) to find out X.

Definition 7. In a stateful signature scheme, the signer updates some state after every signature is produced. A stateful signature scheme consists of three efficient algorithms:
- $\mathsf{KGen}(1^k)$: On input 1^k, compute $(pk,\ sk) \leftarrow \mathsf{KGen}(1^k)$. Let $[X, X^+] := \mathsf{PseudoInvGen}(S)$, where S is a random parameter and X^+ be the initial state of a stateful signature. Then $sk := X^+$ and $pk := X^+X$.
- $\mathsf{Sign}(m, sk)$: To sign a message m using the current state, it outputs a signature σ_m and updates the current state by $\sigma_m := mX^+$.
- $\mathsf{Vrfy}(\sigma_m, pk, m)$: Verify algorithm outputs 1, if and only if $\mathsf{Vrfy}_{pk}(m, \sigma_m) = 1$ such that $\sigma_m \stackrel{?}{=} \sigma_m \cdot\ pk$.

III. Construction

We exploit Field version of the Ring-LPN problem described in [16]. We set aside significantly less computations to the tag than any other entities (e.g., readers, STPs). We divide the ownership transfer protocol in 3 phases: Step-1 describes the communication between a tag and its current and new readers. It includes a mutual authentication protocol between a reader and a tag. Step-2 delineates the protocol transactions between the current and new readers through a designated STP server. Finally, Step-3 outlines the homomorphic signature scheme applied to the readers, STPs, and the main server.

Tag registration: When a tag is registered in the inventory system main server retains the tag associated data (a unique identifier T, an owner identifier $\mathbf{uid}$ and an initial index $\hat{T}$) in the database. Similarly, the main server will set the necessary data into the tag's non-volatile memory such as public parameters for LPN problem $(F, n, \pi_1, \pi_2, \tau)$, a permanent key T, an initial session key c_0 and an initial index $\mathrm{I}_0 \leftarrow \hat{T}^0$. In addition, each tag might be registered to several readers.

Reader registration: All the readers associated to certain tags should maintain the same public and secret data as stored in the tag memory. In addition, in order to delegate ownership a reader should convey current user $\mathcal{U}_c$'s secret ID ($\mathbf{uid}_c$), the key pair (pk, sk) for homomorphic signature scheme. Moreover, to communicate with the STP and other reader $\mathcal{R}_{new}$ and $\mathcal{R}_{cur}$ retains their own key pair (P_j, S_j). Note that, P_j of both $\mathcal{R}_{new}$ and $\mathcal{R}_{cur}$ is also shared with STP for verification test. Furthermore, any two readers in ownership transfer requires a shared secret key ρ in order to transfer tag related data from $\mathcal{R}_{cur}$ to $\mathcal{R}_{new}$.

Encouraged by the proposal described in [16], we define 2 suitable mappings π_1, π_2 such that $\pi_{(i)} : \{0, 1\}^\lambda \rightarrow F$. Let $s \in \{0, 1\}^\lambda$ for the security parameter $\lambda = 80$ be defined as: $(s_1, \cdots, s_{10})$ or $(s_1, \cdots, s_{16})$ where s_i is a number between (1 to 256) or (1 to 32) respectively. Defining the coefficient of the polynomial $v = \pi_i(s) \in F$ as zero except all positions of i such as $i = 10 \cdot (j-1) + s_j,\ j = 1, \cdots 10$ (for π_1) and $i = 16 \cdot (j-1) + s_j,\ j = 1, \cdots 16$ (for π_2). Therefore, both $\pi_1(s)$ and $\pi_2(s)$ are sparse and injective since they will have exactly 10 and 16 non-zero coefficients respectively.

Step-1.1: Although we follow the Field version of the Ring-LPN problem [16], we restrict the Field-version of the Ring-LPN problem (to finite field of characteristic 2) according to the following. Let an irreducible polynomial $f(X)$ be taken over the field $\mathbb{F}_2$ where the degree of f is n, we consider an extended field[4] on $\mathbb{F}_2$ defined as: $F = \mathbb{F}_2[X]/(f) = \mathbb{F}_{2^n} = \mathbb{F}_q$. Therefore, any element $a \in \mathbb{F}_2[X]/(f)$ has a multiplicative inverse in F^*[5].

For *tag authentication*, a shared secret key pair (T, c_i) and an index $(\mathrm{I}_i \leftarrow \hat{T})$ have been derived either from initial tag registration process or from the previous $(i-1)$ successful sessions.

Step-1.1: Tag-Current reader communication (Authentication)

[4]E.g., $f(X) = X^{532} + X + 1$ of degree $n = 532$.
[5]F^* is the set of elements in F that have multiplicative inverse.

$\mathcal{R}_{cur}$	**Tag** $\mathcal{T}$
$\{\mathrm{T}, \mathrm{I}_i, c_i\} \in F$	$\{\mathrm{T}, \mathrm{I}_i, c_i\} \in F$
$s \xleftarrow{\$} \{0,1\}^\lambda$	
$\xrightarrow{s}$	
	$r \xleftarrow{\$} F^*$, $e \xleftarrow{\$} \mathbf{Ber}_w^F$
	$z := r \cdot (c_i \cdot \pi_1(s) + \mathrm{T}) + e$
	$\mathrm{I}_{i+1} = z$
$\xleftarrow{r, z, \mathrm{I}}$	
Lookup T by using I:	
Direct match: If $(\mathrm{I} \neq \mathrm{I}_i)$	
Brute-force search:	
$\exists$ $(\mathrm{T}, c_{i-1}$ or $c_i)$ that satisfies:	
IF $(r \notin F^*)$ return;	
$\hat{e} := z - r \cdot (c_i \cdot \pi_1(s) + \mathrm{T})$	
IF $\mathbf{wt}(\hat{e}) \neq w$ return;	
IF (AUTH)	
$e' \xleftarrow{\$} \mathbf{Ber}_w^F$	
$z' := r \cdot (c_i \cdot \pi_2(s) + \mathrm{T}) + \ e'$	
$c_{i+1} = c_i + \hat{e}$	
$\mathrm{I}_{i+1} = z$	
$\xrightarrow{z'}$	
IF (OT)	AUTH:
$\hat{T}_n := \mathrm{I_i}$	$\hat{e} := z' - r \cdot (c_i \cdot \pi_2(s) + \mathrm{T})$
Go to Step-2	IF $\mathbf{wt}(\hat{e}) \neq w$ return;
	$c_{i+1} = c_i + e$

- Reader: Generate a random binary λ-bit challenge string s, and send it to the tag.
- Tag: Generate a random noise vector e from $\mathbf{Ber}_w^F$ and a random element r from F^*. Next a multiple bit Field-LPN problem z is computed from $r \cdot (c_i \cdot \pi_1(s) + \mathrm{T}) + e$. Finally, the tag forwards (I_i, r, z) to the reader.
- Reader: It first searches local database to match a tuple $\{\mathrm{I}_i,\ \mathrm{T}, c_i\}$. If failed, then apply brute-force search with $\{\mathrm{T}, c_i$ or $c_{i-1}\}$. Note that, the reader would store the secret key c_{i-1} from the previous session in order to resist *De-synchronization* attack. Then it checks whether r is chosen from F^* and then calculates $\hat{e}$ and check whether $\mathbf{wt}(\hat{e})$ is exactly $\lfloor k\tau \rceil$. If the check passes, it accepts the tag and go to the reader authentication phase.

Reader authentication can be decomposed in two phases: (1) Authentication (AUTH), (2) Ownership transfer (OT)

- Reader: During AUTH phase, e' is generated from $\mathbf{Ber}_w^F$ and z' is calculated accordingly. Note that unlike [16], we use the same r as it received from the tag to resist synchronization attack and use a different mapping π_2 with the same challenge s in a view to reduce communication overhead. Update session secret key $c_{i+1} \leftarrow c_i + \hat{e}$ and index $\mathrm{I}_{i+1} \leftarrow z$.
- Tag: Verify the Field-LPN problem by checking the $\mathbf{wt}(\hat{e})$ whether it is exactly w or not. If the check passes, accept the reader and update the session key $c_{i+1} \leftarrow c_i + e$ and index $\mathrm{I}_{i+1} \leftarrow z$.

Step-1.2: New reader-tag communication (Ownership Transfer)

$\mathcal{R}_{new}$	**Tag** $\mathcal{T}$
$\{\mathrm{T}, \mathrm{I}_i, c_i\} \in F$	$\{\mathrm{T}, \mathrm{I}_i, c_i\} \in F$
$\{\mathrm{T}, \hat{T}_n, c\} \leftarrow \mathcal{D}_\rho(\Gamma_j)$	
$\mathrm{I}_i \leftarrow \hat{T}_n$	
$s' \xleftarrow{\$} \{0,1\}^\lambda$	
$e' \xleftarrow{\$} \mathbf{Ber}_w^F$	
$z' := r \cdot (c_i \cdot \pi_2(s') + \mathrm{T}) + \ e'$	
$c_{i+1} = c_i + e'$	
$\xrightarrow{s', z'}$	
	OT:
	$\hat{e} := z' - r \cdot (c_i \cdot \pi_2(s') + \mathrm{T})$
	IF $\mathbf{wt}(\hat{e}) \neq w$ return;
	$c_{i+1} = c_i + \hat{e}$

Step-1.2 During OT phase, $\mathcal{R}_{cur}$ records the ownership index (Step 1.1): $\hat{T}_n \leftarrow \mathrm{I_i}$. Later $\hat{T}_n$ would be forwarded to the $\mathcal{R}_{new}$ and consequently to the main server. If the verification in Step-2 is passed successfully, $\mathcal{R}_{new}$ commences reader authentication.

- Reader: It first decrypts Γ_j to retrieve tag data $\{\mathrm{T}, \hat{T}_n, c\}$. $\mathcal{R}_{new}$ generates s', e' and hence calculates $z' \leftarrow r \cdot (c_i \cdot \pi_2(s) + \mathrm{T}) + \ e'$. Next it updates the session key $c_{i+1} \leftarrow c_i + e'$ and forwards (s', z') to the tag. Note that we assume $\mathcal{R}_{cur}$ is honest enough not to intercept the protocol transcript (s', z'), or $\mathcal{R}_{new}$ would forward (s', z') through some secret channel. Once this protocol transaction is executed successfully, both the parties update c_i in order to achieve *forward-secure* privacy.
- Tag: Since the protocol transcripts (s', z') in OT phase is different from that of AUTH phase (z'), a tag adopts OT phase for new reader authentication. Note that unlike AUTH phase, it calculates $\hat{e}$ from $\pi_2(s')$ and index I_i is not updated in OT phase. However, if the $\mathbf{wt}(\cdot)$ check passes, it updates the session key $c_{i+1} \leftarrow c_i + \hat{e}$ from $\hat{e}$ (not from e in AUTH phase).

Step-2: Let an owner $\mathcal{U}_c$ using reader $\mathcal{R}_{cur}$ intend to transfer ownership of m tags $\mathrm{T}^{\{1,\cdots,m\}}$ with previous ownership index $\hat{T}^{\{1,\cdots,m\}}$ and new ownership index $\hat{T}_n^{\{1,\cdots,m\}}$ to a new owner $\mathcal{U}_n$ using reader $\mathcal{R}_{new}$ in the presence of an STP server $\mathcal{S}_i$. All the operating parties such as $\mathcal{R}_{cur}$, $\mathcal{R}_{new}$ and $\mathcal{S}_i$ share common secrets (P_n, P_c). In addition, $\mathcal{R}_{cur}$ and $\mathcal{R}_{new}$ have their own secrets resp. S_c and S_n. However, they also share a common secret key ρ for transferring tag related data after a successful verification by STP.

We use pseudo inverse matrix properties for key generation. Let $S_c \leftarrow X^+ \in \mathbb{Z}_2^{m \times n}$ be a pseudo-inverse of a matrix $X \in \mathbb{Z}_2^{n \times m}$ and $P_c \leftarrow XX^+ \in \mathbb{Z}_2^{n \times n}$. In the same way, we define $S_n \leftarrow Y^+ \in \mathbb{Z}_2^{m \times n}$ and $P_n \leftarrow YY^+ \in \mathbb{Z}_2^{n \times n}$.

- $\mathcal{R}_{cur}$ randomly generates 2 non-singular $n \times n$ matrices Q, V and send challenge Q to $\mathcal{R}_{new}$ as a challenge matrix.
- $\mathcal{R}_{new}$ will calculate ciphertext $E = P_c Q \in \mathbb{Z}_2^{n \times n}$ and by selecting the first column vector $q \in \mathbb{Z}_2^{n \times 1}$ of Q, it generates a signature $\alpha = q \cdot S_n \in \mathbb{Z}_2^{m \times 1}$. It then forwards E, α to the $\mathcal{S}_i$ for justification.
- Meanwhile, $\mathcal{R}_{cur}$ generates trapdoor $(C, D) := (VX^+, VX^+Q)$ on (S_c, Q) and sends (C, D) to $\mathcal{S}_i$ to justify.
- The STP server $\mathcal{S}_i$ checks $CE \stackrel{?}{=} D$. Note that $CE =$

$VX^+XX^+Q = VX^+Q = D$ i.e., $X^+XX^+ = X^+$ from pseudo-inverse matrix properties. However, if the verification passes, $\mathcal{S}_i$ will forward signature α to $\mathcal{R}_{cur}$ for notification.

- $\mathcal{R}_{cur}$ ensures that $\mathcal{S}_i$ has justified the agreement by checking $\alpha P_n = qY^+YY^+ = qY^+ = \alpha$. Then it generates signature $\beta \leftarrow \alpha \cdot S_c = \alpha X^+$ by taking α as a challenge, encrypt tag data by the secret key ρ to output Γ. Then $\mathcal{R}_{cur}$ sends (Γ, β) to S_i.
- S_i checks whether Γ has been arrived from $\mathcal{R}_{cur}$ by checking $\beta P_c = \alpha X^+XX^+ = \alpha X^+ = \beta$ and forwards Γ to $\mathcal{R}_{new}$.
- $\mathcal{R}_{new}$ decrypts Γ to retrieve tag data and enters Step-3 to update ownership information at the main server. Meanwhile, it runs Step-1.2 to complete tag/reader authentication.
- Now S_i needs to update the shared key among $\mathcal{R}_{new}$ and $\mathcal{R}_{cur}$. It generates two random $n \times n$ matrix M, N and follow the session key update procedure described in [15].
- Finally, $\mathcal{R}_{new}$ and $\mathcal{R}_{cur}$ updates their key pair as $\{S_{(c/n)+1}, P_{(c/n)+1}\}$.

Step-3: $\mathcal{R}_{new}$ is responsible for updating ownership data on the trusted main server through a legitimate STP. We customize a HomSig scheme in [17] so that it fits our ownership transfer protocol. The application specifies global parameters $m \in \mathbb{N}$ such that $m \geq 1$. Each owner in the system is registered with the Main Server with a shared secret $\mathbf{uid}_i$. Let new owner $\mathcal{U}_{new}$ want to confirm about transferring ownership of m tags ($\mathrm{T}^{\{1,\cdots,m\}}$, $\hat{T}^{\{1,\cdots,m\}}$) from $\mathcal{U}_{cur}$ to the Main Server. $\mathcal{U}_{new}$ has its *secret* identifier $\mathbf{uid}_n \leftarrow \mathbb{Z}_p^*$. Each signer ($\mathcal{R}_{new}$) has its own key pair ($pk$, sk) for the HomSig scheme. Note that all the operations in the scheme are defined over F.

Key generation: System generates a pair of keys for each reader $(pk, sk) \xleftarrow{\$} \mathsf{KGen}(1^\lambda, m)$.

$\mathsf{KGen}(1^\lambda, m)$: Let $\mathbb{G}, \mathbb{G}_T$ be bilinear groups of prime order p such that $p < q$ and $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$ is a bilinear map with $g \in \mathbb{G}$ as a generator. Let $k \xleftarrow{\$} \mathbb{Z}_p$ and $h, g_1, \cdots, g_m \xleftarrow{\$} \mathbb{G}$. Set $K := g^k$ and output public key $pk := (p, g, K, h, g_1, \cdots, g_m)$ and the secret key $sk := k$.

Ownership Transfer: On input a set of m tag identifiers $\mathrm{T}^{\{1,\cdots,m\}} \in F$, $\mathcal{R}_{new}$ generates signature $\sigma_i \xleftarrow{\$} \mathsf{Sign}(\cdot)$ on behalf of $\mathcal{U}_{new}$.

$\mathsf{Sign}(sk, \mathbf{uid}_n, \mathrm{T}^{(i)}, \hat{T}^{(i)}, \hat{T}_n^{(i)})$: $\mathcal{R}_{new}$ picks a random $t \xleftarrow{\$} \mathbb{Z}_p$ and compute the following:

$$\sigma := (h^t \prod_{i=1}^{m} {g_i}^{T_i})^{\frac{1}{k+\mathbf{uid}_n}}$$
$$\Sigma := (\sigma, t)$$

Finally, it sends the tuple $(\mathcal{U}_{cur}, \mathcal{U}_{new}, \mathbf{uid}_n, \hat{T}^{(i)}, \hat{T}_n^{(i)}, \Sigma)$ to the intermediate Server $\mathcal{S}_i$.

Intermediate Server: When an intermediate server $\mathcal{S}_i$ receives $\varkappa$ tuples from different readers/intermediate servers, it generates the combined signature $\Sigma \leftarrow \mathsf{CombSign}(\cdot)$.

$\mathsf{CombSign}(\mathcal{U}_{cur}, \mathcal{U}_{new}, \mathbf{uid}_n, pk, \hat{T}^{(i)}, \hat{T}_n^{(i)}, \Sigma_i)$: First it runs **VerSign** $(\cdot)$ algorithm to verify Σ_i. Then for each $i \in \{1, \cdots, \varkappa\}$, it randomly generates a coefficient $\eta_i \xleftarrow{\$} F$ and computes:

$$\sigma = \textstyle\prod_{i=1}^{\varkappa} (\sigma_i)^{\eta_i}, \quad t = \sum_{i=1}^{\varkappa} \eta_i \cdot t_i \bmod p$$
$$\Sigma := (\sigma, t)$$

Finally, it forwards $(\mathcal{U}_{cur}, \mathcal{U}_{new}, \mathbf{uid}_n, \hat{T}^{(i)}, \hat{T}_n^{(i)}, \Sigma)$ either to another Intermediate Server, or to the Main Server.

Main Server: When the server obtains m linearly independent vectors of tag indexes $\hat{T}^1, \cdots, \hat{T}^m$ with the respective combined signature Σ, it first searches the existing databases in order to obtain the tag identifiers $\mathrm{T}^1, \cdots, \mathrm{T}^m$ with the received index-owner tuple $[\hat{T}^(i), \mathcal{U}_{cur}]$. Then, it checks the validity of Σ by $\mathsf{VerSign}(\cdot)$ algorithm.

$\mathsf{VerSign}(\mathcal{U}_{cur}, \mathcal{U}_{new}, pk, \mathbf{uid}_n, T^{(i)}, \hat{T}_n^{(i)}, \Sigma)$: Let $\Sigma = (\sigma, t) \in \mathbb{G} \times \mathbb{Z}_p$. It returns 1 if Σ is a valid signature on $\mathrm{T}^{(i)}$ with respect to the $\mathcal{U}_{new}$'s *secret* $\mathbf{uid}_n$, otherwise returns 0 by the following:

$$e(\sigma, K \cdot g^{\mathbf{uid}_n}) \stackrel{?}{=} e(h^t \textstyle\prod_{i=1}^{m} g_i^{T_i}, g)$$

After successful verification, Server updates its database with new indexes $\hat{T}_n^{(i)}$ to the $\mathrm{T}^{(i)}$ for the new owner $\mathcal{U}_{new}$.

IV. Security Analysis

Theorem 1. *If mapping function π_i is suitable for field F and the* Field-LPN$_w^F$ *problem is (t, Q, ϵ)-hard then the authentication protocol is (t, Q, ε)-secure against* ***active adversaries****, where*

$$t' = t - Q.exp(F) \qquad \epsilon' = \epsilon + Q.2^{-\lambda} + s(\tau, \tfrac{1}{2})^{-n}$$

and $exp(F)$ is the time to perform $O(1)$ exponentiations in F.

Proof: We refer to the Ring-LPN based authentication paper in [16] for detail proof.

Proposition 1. *The hardness of decisional* exact-LPN *is polynomially related to that of search* LPN and *the protocol has no completeness error $\epsilon_c(w, n) \approx 0$.*

Proof: Interested readers are referred to [10], for further clarification and proof of the Proposition.

Theorem 2. *If Sum of Subset (SSP) problem is NP-complete then Binary Matrix Factorization (BMF) problem is also NP-complete.*

Proof: Proof is postponed to the full version of the paper.

Theorem 3. *If BMF is* hard*, then construction of the lightweight stateful signature in Step-2 is existentially unforgeable under a one-time chosen message attack (OT-CMA).*

Step-2: Reader-STP Communication

$\mathcal{R}_{\mathbf{cur}}(\mathcal{U}_c)$	IntServer $(\mathcal{S}_i)$	$\mathcal{R}_{\mathbf{new}}(\mathcal{U}_n)$
(P_c, P_n, S_c, ρ)	$\{P_n, P_c,\} \in \mathbb{Z}_2^{n\times n}$ $\{S_c, S_n\} \in \mathbb{Z}_2^{m\times n}$	(P_n, P_c, S_n, ρ)

Transfer Ownership:
$\mathrm{T}^{\{1,\cdots,m\}}$ with $\{\hat{T}^{\{1,\cdots,m\}}, \hat{T}_n^{\{1,\cdots,m\}}\}$
Non-singular $Q, V \xleftarrow{\$} \mathbb{Z}_2^{n\times n}$

$\xrightarrow{\quad Q \quad}$

$C := VS_c = VX^+$ $\qquad$ $E := P_cQ = XX^+Q \in \mathbb{Z}_2^{n\times n}$
$D := CQ = VX^+Q$ $\qquad$ $q \leftarrow Q^{1\times n}$
$\alpha := q \cdot S_n = qY^+ \in \mathbb{Z}_2^{m\times 1}$

$\xrightarrow{\quad C, D \quad}$ $\qquad$ $\xleftarrow{\quad E, \alpha \quad}$

IF $(CE \neq D)$ return;

$\xleftarrow{\quad \alpha \quad}$

IF$(\alpha \cdot P_n \neq \alpha)$ return;
$\beta := \alpha S_c = \alpha X^+$
$\Gamma \leftarrow \mathcal{E}_\rho(\mathrm{T}^{\{1,\cdots,m\}}, \hat{T}_n^{\{1,\cdots,m\}}, c^{\{1,\cdots,m\}})$

$\xrightarrow{\quad \Gamma, \beta \quad}$

IF$(\beta \cdot P_c \neq \beta)$
return;

$\xrightarrow{\quad \Gamma \quad}$

Run Step-3 and Step-1.2

Generate non-singular $M, N \in_R \mathbb{Z}_2^{n\times n}$
$S_{c+1} = M \cdot S_c \in \mathbb{Z}_2^{n\times m}$
$S_{n+1} = N \cdot S_n \in \mathbb{Z}_2^{n\times m}$
where *rank*$(S_{c+1}) = (S_{n+1}) = m$

${S_{c+1}}^+ := ({S_{c+1}}^T S_{c+1})^{-1} {S_{c+1}}^T \in \mathbb{Z}_2^{m\times n}$
${S_{n+1}}^+ := ({S_{n+1}}^T S_{n+1})^{-1} {S_{n+1}}^T \in \mathbb{Z}_2^{m\times n}$

$P_{c+1} := [S_{c+1}] \cdot [S_{c+1}]^+ \in \mathbb{Z}_2^{n\times n}$
$P_{n+1} := [S_{n+1}] \cdot [S_{n+1}]^+ \in \mathbb{Z}_2^{n\times n}$

${P_c}' := P_c \cdot M \in \mathbb{Z}_2^{n\times n}$
$P_n' := P_n \cdot M \in \mathbb{Z}_2^{n\times n}$

$\xleftarrow{\quad {P_c}' \quad}$ $\qquad$ $\xrightarrow{\quad P_n' \quad}$

$S_{c+1} = ({P_c}' S_c) \in \mathbb{Z}_2^{m\times n}$ $\qquad$ $S_{n+1} = ({P_n}' S_n) \in \mathbb{Z}_2^{m\times n}$
If *rank* $(S_{c+1}) \neq n$ $\qquad$ If *rank* $(S_{n+1}) \neq n$
return; $\qquad$ return;
$P_{c+1} = [S_{c+1}] \cdot [S_{c+1}]^+$ $\qquad$ $P_{n+1} = [S_{n+1}] \cdot [S_{n+1}]^+$

Proof: Proof is deferred to the full version.

Proposition 2. *Authentication protocol between the reader and tag is free from Man-In-the-Middle (MIM) attack.*

Proof: Authentication protocol from Field-LPN is not proved secure against MIM attack. By using a universal hash function described in [18], it can be converted to a MIM-secure scheme. However, our protocol is not vulnerable to MIM attack for the following reason. In order to recover the entire secret (T,c), an adversary $\mathcal{A}$ needs to repeat (MIM attack described in [16]) $\mathcal{O}(n)$ times[6] successful attacks and then to apply Gaussian elimination method. Since our protocol enjoys the advantage of session key c_i, it updates one of the secret keys c_i in each transaction (during authentication or ownership transfer). Consequently, it resists the adversary $\mathcal{A}$ to obtain $2n$ linearly-independent equations from the same secret key pair (T,c) and hence MIM attack. A very recent proposal in [19] claims an attack against Ring-LPN in [16]. Authors first describe a matrix varient of the Ring-LPN protocol and state their changes to reduce communication and computation complex-

[6]To obtain $2n$ linearly-independent equations

ity. They propose to query the Ring-LPN oracle repeatedly Q times with the same secret c in order to obtain a sequence of $(z_1, z_1c+e_1), (z_2, z_2c+e_2), \cdots, (z_Q, z_Qc+e_Q)$. Although the attack is not practical as they claimed but nevertheless can not break our protocol's security since we would update the secret key c in each session. However, space constraints inhibit us to present a full-blown proof here. It will appear in the full version of the paper.

We consider a potential MIM-attack scenario regarding ownership transfer protocol, while the old owner $\mathcal{R}_{cur}$, listening to the insecure channel between the new owner and the tag on Step-1.2 to compute the new session key c_{i+1}. Therefore, we assume $\mathcal{R}_{cur}$ to be *honest* minimum for a single protocol transaction just after the ownership transfer occurs. After that, both the new reader and tag would update their secret to retain forward privacy.

A. Privacy

One of the major privacy issues in Ownership transfer protocol is to satisfy previous owner and new owner privacy settings in terms of ownership transfer. In step 1.2. of the protocol the new reader is given the tuple $\{T, I_i, c_i\}$ which includes the tag's secret. But $\mathcal{R}_{new}$ immediately authenticates the tag and updates the session secret c_i. Subsequently, *Theorem 6.* ensures that the authentication protocol is forward (resp. backward) privacy secure due to updating session key. That is why the new reader is unable to interpret the tag's previous communication and current reader cannot trace the tag after the ownership transfer even if the secret is transferred from the current reader to the new reader.

In order to define privacy, we analyzed our protocol according to the privacy framework based on *zero-knowledge* (ZK) formulation [7] where it is assumed that no secret will be revealed from the protocol transactions. This model rely on the unpredictability of the entity's (e.g., the tag) output in the protocol execution $\pi \leftarrow 2\lambda + 1 \; s.t. \; \lambda \geq 1$ (our case: $\pi = 3 \; s.t. \; \lambda = 1$).

Let $\hat{\mathcal{A}}$ be a PPT CMIM (Concurrent Man in the Middle) adversary equivalent to $\mathcal{A}$ (respectively, simulator Sim) that takes on input the system public parameters $\mathbf{Pub}_T$, the reader $\mathcal{R}$ and the set of tags $\hat{\mathcal{T}}$; and interacts with $\hat{\mathcal{T}}, \mathcal{R}$ via the oracles mentioned above. Let $\hat{\mathcal{A}}$ be composed of a pair of adversaries $(\hat{\mathcal{A}}_1, \hat{\mathcal{A}}_2)$ and their corresponding simulators (Sim_1, Sim_2) for $\mathbf{Exp}^{ZK}_{\mathcal{A}}(\hat{T})$ experiments with the above oracles.

Experiment $\mathbf{Exp}^{ZK}(\hat{\mathcal{T}})$

- Initialize RFID system, the reader $\mathcal{R}$, the tag set $\hat{\mathcal{T}}$ (s.t., $|\hat{\mathcal{T}}| = l$) by **SetupTag**$(\cdot)$
- let $\mathcal{O} \leftarrow \text{Launch}, \text{Dtag}, \text{STag}, \text{SReader}, \text{Ukey}, \text{Corrupt}$
- Real: $(\mathcal{T}, st) \leftarrow \hat{\mathcal{A}}_1^{\mathbf{DTag}}(\mathcal{R}, \hat{\mathcal{T}}, \mathbf{Pub}_T)$
  Simulation: $(\mathcal{T}, st) \leftarrow Sim_1^{\mathbf{DTag}}(\mathcal{R}, \hat{\mathcal{T}}, \mathbf{Pub}_T)$
  where $\mathcal{T} = \{T_{i_1}, T_{i_2}, \cdots, T_{i_\delta}\} \in \mathcal{T}$ s.t., $0 \leq \delta \leq l$
- $c \in_R C \leftarrow \{1, 2, \cdots, l-\delta)\}$ and $C = \hat{\mathcal{T}} - \mathcal{T}$
  Real: $T_c = T_{i_c}$
  Simulation: c is unknown to Sim_2
- Real: *view* $\leftarrow \hat{\mathcal{A}}_2^{\mathcal{O}}(\mathcal{R}, \hat{\mathcal{T}}, T_c, st)$
  Simulation: *sview* $\leftarrow Sim_2^{\mathcal{O}}(\mathcal{R}, \hat{\mathcal{T}}, st)$
- Real: output $(c,$ *view*$_{\hat{A}})$
  Simulation: output $(c,$ *sview*$_{Sim})$

We assume that $\hat{\mathcal{A}}$ queries the challenger with $\mathbf{Exp}^{ZK}(\hat{T})$ in the *read world* and *simulation* mode. Note that if $\delta = 0$, no challenge tag is selected and the number of clean tags $|C| = l - \delta$. ZK-privacy implies that adversary $\hat{\mathcal{A}}$ cannot distinguish any challenge tag T_c from any set C of tags. That's why, $\hat{\mathcal{A}}_1$ is used to output an arbitrary set C and to limit $\hat{\mathcal{A}}_2$ to blind access to a challenge tag from C. Therefore, the advantage of the adversary with security parameter κ to win the privacy game can be defined as

$$\mathbf{Adv}^{ZK}_{\mathcal{A},Sim,\mathcal{D}}(\kappa, \hat{T}) = |\Pr[\mathbf{Exp}^{ZK}_{\hat{\mathcal{A}}}(c, l, view(.) = 1)] - \Pr[\mathbf{Exp}^{ZK}_{Sim}(c, l, sview(.) = 1)]| \leq \epsilon$$

Definition 8. RFID authentication protocol described in Step-1 satisfies the ZK-privacy in [7] security model if for any adversary $\hat{\mathcal{A}}$, there exist a simulator Sim such that for any distinguisher $\mathcal{D}$, $\mathbf{Adv}^{ZK}_{\mathcal{A},Sim,\mathcal{D}}(\kappa, \hat{T})$ is negligible.

Theorem 4. *From the Field*-LPN *problem, the protocol described in Step-1 satisfies ZK-privacy.*
Proof: Deferred to the full version.

Theorem 5. *An RFID authentication protocol described in* Step. 1. *is forward (resp., backward)-ZK private.*
Proof: Deferred to the full version.

V. Performance evaluation

We concentrate on the computationally weakest of the entities, the tag. Ring-LPN has an outstanding lower communication overhead targeting lightweight ultra constrained tags equipped with tiny CPUs e.g., EPC class tags (the price range of a few cents) [16]. Subsequently, we slightly modify the field version of the protocol to a mutually authentication protocol with less computation and communication complexity and to make the protocol MIM-free and to resist a very recent attack proposed in [19] against Ring-LPN.

Computation Requirement: Following exact-LPN version in [10] yields the completeness error $\varepsilon_c = 0$ (whereas $\varepsilon_c \approx 2^{-55}$ in [16]). Field-LPN as we followed can do sparse multiplication for π_i that takes $21k$ clock cycles while other multiplication requires $150k$. Time to build e from Ber^F_w need $3k$ clock cycle [16]. If we ignore EX-OR operation cost, we need approximately $345k$ clock cycle for mutual authentication and require 20 ms to respond at 2 MHz clock rate. This response time is sufficient in many application scenarios since a delay of 1 sec is often considered acceptable [16].

For anonymous verification by an STP requires 1 $(n \times n)$ matrix multiplication while the $\mathcal{U}_{cur}$ requires (2 matrix + 2 vector) multiplication and $\mathcal{U}_{cur}$ needs only (1 matrix + 2 vector) multiplication.

HomSig is comprised of only 1 group element in $\mathbb{G}$ and 1 element in $\mathbb{Z}_p$. In order to provide a typical security level of 2^{80}, we can set p a 170 bit prime number and then the element in $\mathbb{G}_1$ is 171 bits long. Then the aggregated signature size from the reader to other readers/server would be 42 bytes in total. Signing costs include a multi-exponentiation in $\mathbb{G}$

TABLE II
TAG RESOURCES AND SECURITY COMPARISON WITH HB FAMILY

Scheme	P_1	P_2	P_3	P_4	Others
Afifi *et al.'07* [8]	k_1, k_2	5 Encryption 3 PRNG	5	No	
Kuseng *et al.'10* [22]	I_n, I, s, c	2 PUF 1 LFSR 4 PRNG	2	No	$\star$
Cai *et al.'11* [20]	k, s	2 Hash 1 MAC	2	No	$\dagger$
Yang *et al.'11* [11]	k_1, k_2, k_3	3 Encryption	3	Yes	$\diamond$
Song *et al.'11* [2]	I, k, c	4 Hash 2 Encryption	4	No	$\star$
Kapoor *et al.'12* [13]	s, k_1, k_2	2 keyed Hash 2 PRNG	4	No	$\diamond$
Doss *et al.'13* [21]	I, s, r, n	3 mod-squaring 1 CRC 3 PRNG	7	Yes	$\star$
Our scheme	I, k, s	2 Field-LPN	3	Yes	$\dagger \diamond \ddagger \star$

P_1: Tag secret type, P_2: Cryptographic techniques used on tag, P_3: Number of Protocol transaction related to the tag, P_4: Mutual Authentication, *Includes EPC class compliance, $^\diamond$TTP supported, †Aggregated Signature, ‡Semi-trusted Server. PRNG:= Pseudo Random Number Generator

and verification requires to compute only two pairings, one exponentiation in $\mathbb{G}$.

Communication complexity: During reader-tag communication, the protocol requires 4 elements from field F and 1 λ-bit string for authentication while 2 λ-bit string for ownership transfer. However, reader-reader communication involves total $4n^2 + 2n$-bit for communication.

Storage Requirement: All the parties in the protocol need to store the public parameters. However, a tag needs to store 3 secrets from F A reader requires to store the same for authentication. However, for ownership transfer it needs to store 3 keys for pseudo inverse matrix operation $(2n{\cdot}n{+}1m{\cdot}n)$ bits, user identifiers it works for (1 element from F for each user), tag ownership index for a set of m tags (m elements from F) and 1 shared secret key for suitable encryption. Nevertheless, storage requirement for the tag can be expressed by $\mathcal{O}(1)$ while that is $\mathcal{O}(m)$ for the readers/server such that m is the number of tags in an RFID system.

VI. CONCLUSION

This paper presents a novel scalable RFID ownership transfer protocol leveraging the reader authentication phase based on a lightweight Field-LPN problem that can meet the hardware constraints of the EPC Class tags. Moreover, using an efficient homomorphic aggregated signature facilitates transferring ownership of a set of tags together without direct-attachment to a trusted main server that makes the protocol to be compliant with an inventory system context. Furthermore, our protocol enables ownership transfer with readers verification that preclude operating partners in an inventory management system from injecting fake products.

REFERENCES

[1] Ari Juels and Stephen A. Weis. Authenticating pervasive devices with human protocols. CRYPTO 2005, volume 3621 of LNCS, pages 293-308. Springer, 2005.

[2] B. Song and C. J. Mitchell. Scalable RFID security protocols supporting tag ownership transfer. Comput. Commun. , vol. 34, no. 4, pp. 556566, 2011.

[3] H. Gilbert, M. Robshaw, and H. Sibert, An active attack against HB+ - a provably secure lightweight authentication protocol, IEEE Letters, vol 41(21), 2005.

[4] Julien Bringer, H. Chabanne, and Emmanuelle Dottax, HB++: a lightweight authentication protocol secure against some attacks, In SecPerU, pages 28-33, 2006.

[5] Cao, X , ONeill, M. (2011). F-HB: An Efficient Forward Private Protocol. (Lightsec2011), March 14-15, 2011, Istanbul, Turkey.

[6] MSI Mamun, A. Miyaji. A privacy-preserving efficient RFID authentication protocol from SLPN assumption. International Journal of Computational Science and Engineering (IJCSE), Inderscience Publishers, Vol. 9, 2014.

[7] R. H. Deng, Y. Li, M. Yung, and Y. Zhao. A new framework for RFID Privacy. in Proceedings of the ESORICS 10, vol. 6345 of LNCS, pp. 118, Springer.

[8] S. Fouladgar and H. Afifi. An efficient delegation and transfer of ownership protocol for RFID tags. In EURASIP Workshop on RFID Technology, Austria, 2007.

[9] MSI Mamun, A. Miyaji, M. Rahman. A Secure and Private RFID Authentication Protocol under SLPN Problem. NSS 2012, LNCS 7645, pp. 476-489, 2012.

[10] A. Jain, S. Krenn, K. Pietrzak. Commitments and Efficient Zero-Knowledge Proofs from Learning Parity with Noise. ASIACRYPT 2012, LNCS,Vol 7658.

[11] M. H. Yang. Across-authority lightweight ownership transfer protocol. Electronic Commerce Res. Applicat., vol. 10, no. 4, pp. 375383, 2011.

[12] K. Osaka, T. Takagi, K. Yamazaki, and O. Takahashi. An efficient and secure RFID security method with ownership transfer. in Proc. ICCIS, 2006.

[13] G. Kapoor and S. Piramuthu, Single RFID tag ownership transfer protocols. IEEE Trans. Systems, Man, Cybern. C, Appl. Rev., vol. 42, no. 2, pp. 164173, Mar. 2012.

[14] Katz, Jonathan, Ji Sun Shin, and Adam Smith. Parallel and concurrent security of the HB and HB+ protocols. Journal of cryptology 23, no. 3 (2010): 402-421.

[15] MSI Mamun, A. Miyaji. A fully-secure RFID authentication protocol from exact LPN assumption, IEEE TrustCom'13, page 102-109, DOI: 10.1109/TrustCom.2013.17.

[16] H. Stefan, E. Kiltz, V. Lyubashevsky, C. Paar, and K. Pietrzak. Lapin: an efficient authentication protocol based on Ring-LPN. FSE, pp. 346-365. Springer 2012.

[17] D. M. Freeman. Improved security for linearly homomorphic signatures: A generic framework. PKC 2012.

[18] D. Yevgeniy, E. Kiltz, K. Pietrzak, and D. Wichs. Message authentication, revisited. In Advances in EUROCRYPT 2012, pp. 355-374. Springer, 2012.

[19] Bernstein, Daniel J., and Tanja Lange. Never trust a bunny. In Radio Frequency Identification. Security and Privacy Issues, pp. 137-148. Springer, 2013.

[20] Cai, Shaoying, et al. Protecting and restraining the third party in RFID-enabled 3PL supply chains. Information Systems Security. LNCS, 246-260, 2011.

[21] Doss, Robin, Wanlei Zhou, and Shui Yu. Secure RFID Tag Ownership Transfer Scheme based on Quadratic Residues. 1-1, 2013.

[22] L. Kuseng, Z. Yu, Y. Wei, and Y. Guan. Lighweight mutual authen tication and ownership transfer for RFID systems. in Proc. IEEE Infocom, pp. 15, 2010.

[23] Murty, Katta G., and Santosh N. Kabadi. Some NP-complete problems in quadratic and nonlinear programming. Mathematical programming 39.2: 117-129, 1987.