A fully-secure RFID authentication protocol from exact LPN assumption

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

***Abstract*—In the recent years, several light-weight cryptographic solutions have been proposed for RFID system. HB-family is one of promising protocol series, based on the hardness of the Learning Parity with Noise (LPN) problem. Most protocols in HB-family are not suited for mobile/wireless reader applications due to secure channel assumptions. In this paper, we present a fully secure collaborative mutual authentication protocol for an RFID system where both channels tag-reader and reader-server are considered to be insecure. More precisely, we introduce a new variant of an HB-like protocol where the complete RFID system is authenticated under LPN-based commitment scheme by taking advantages of properties of perfect computational hiding commitment scheme, *pseudo-inverse* matrix, and randomized Hill cipher. In addition, through detailed security and privacy analysis, we show that our scheme achieves required security and privacy properties, under not the random oracle model, but the standard model.**

***Keywords*: Mutual authentication, exact LPN problem, pseudo-inverse matrix, Hill cipher.**

I. Introduction

Mutual authentication protocol adds an additional protection for an RFID system in the protocol construction to safeguard the query is, in fact, coming from a legitimate entity, and therefore, ensures that the tag information is available to only valid reader and server. Most authentication protocols proposed so far either presume reader and server as an identical entity, or assume the communication channel between a server and a reader is secure [8]–[12], [14], [19]–[23]. Unfortunately, this is not always the case. For instance, some emerging applications like detecting fraudulent production [5], or an RFID inventory management systems where products are sold to the customers, and later ownership of the product need to be transferred to new customers. In the aforementioned cases, readers (e.g., owners) are different entity to the trusted server.

On the other hand, the limited processing and storage capability of traditional RFID tags limit the effective use of cryptographic techniques such as RSA, ECC [2] [4]. HB-family protocols based on LPN assumption require a few thousand gates for implementation making them an attractive option for securing low cost EPC tags [25]. LPN assumption used in HB-like protocols, inquires to distinguish *noisy* linear equation from uniformly random. Since its first introduction in 2001, numerous applications i.e., lightweight crypto system, symmetric encryption etc. have introduced LPN problem as the assumption underlying provably secure cryptosystems [6]. Its popularity is due to robust security against quantum algorithms. Unlike most number theoretic problems used in applied cryptography, LPN based constructions are inclined to be extremely efficient in view of computation time and memory requirement which lead LPN based cryptosystem to be a good candidate for resource-constraint devices like RFID tags, smart phone device etc. There has been a lot of research on HB protocol that outputs a number of protocols s.t., HB^{+}, HB^{++}, $HB^{\#}$, HB-MP, HB-MP^{+}, HB^{*}, F-HB etc. [8]–[12], [14], [21]. Unfortunately, most of them later shown to be insecure, or susceptible to particular attacks [13], [14]. In addition, no scheme consider each entity in the RFID system individually against security and privacy threats.

We followed a very simple, efficient and perfectly binding string commitment scheme with an *exact* version of the LPN-problem, whose security is based on the hardness of the LPN problem [28]. Unlike other HB-protocols, our protocol follows the exact LPN based commitment scheme for authentication, the secret keys are *binary matrix* and pair of secret keys shared between entities are different. In order to update session key and to verify protocol transcripts, we introduce pseudo-inverse matrix properties and randomized Hill cipher techniques. This makes the proposed protocol more robust against quantum adversaries while being efficient like the previous HB-protocol family.

Our contribution. In this paper, we propose a new variant of RFID authentication system from exact LPN problem, that can provably withstand all known attacks. In addition, unlike other traditional authentication protocols for RFID system, all communications between a server and a reader are assumed to be insecure and over inauthentic channel. Therefore, reader and server are not identical but two individual entities. More precisely, we use an identical scheme to authenticate all the entities (Tag, Reader, Server) together in an RFID system. The main objective of our scheme is to improve the security scope of a recently proposed variant of HB-protocol in [18] and [25] by adding some non-linear components without increasing its complexity significantly. Unlike authentication scheme described in [18] [25], we adopt several new ideas for construction such as:

- Only a server is considered to be *fully trusted* and *keys* are shared among the entities accordingly.
- Use the commitment scheme from exact LPN, in compare

to the decisional-LPN problem in [18] and subspace-LPN problem in [25] in order to remove completeness/correctness error.
- More properties of pseudo-inverse matrix, such as signature-like light authentication in the reader-tag transaction.
- A variant of Hill cipher in the reader-server communication.

To the best of our knowledge, we propose the first HB-like authentication protocol for RFID system that is *fully secure*[1], private and scalable. Moreover, the protocol supports forward privacy under zero-knowledge (ZK) indistinguishable notion and also provides all security proof under standard model. Consequently, the protocol could be realized through several RFID security applications in the real life environment like *authorization recovery, ownership transfer, controlled delegation* etc. [26] [27].

Organization. The rest of our paper is organized as follows. Section II introduces useful notations, assumptions and definitions. Our protocol is described in Section III. In Section IV-V, security and privacy are discussed with proof. Section VI covers the performance analysis and comparison results with others. Finally, Section VII concludes the paper.

II. Preliminaries

In this section, we first briefly introduce the notations used in the paper in Table I. Then we discuss some inevitable assumptions followed by useful definitions for primitives and security notions.

A. Assumption

RFID system in this paper consists of a single legitimate server, a set of readers and a set of tags (EPC global Class 1 generation 2). Readers are connected to the back-end server that stores all the data related to the tags and their corresponding readers in the database. Each tag has its unique identification T_{id}, a permanent key $\mathbf{S}'$ and a session key $\mathbf{S}''$. However, T_{id} is used as the shared secret among all the 3 parties while $\mathbf{S}\leftarrow\mathbf{S}' \,\|\mathbf{S}''$ is shared only between the tag and the server.

We refer to the computational hiding property of the commitment scheme described in [28] that is polynomially equivalent to the security of the well known LPN problem. Note that the hardness of *exact* LPN lies under the hardness of traditional LPN problem. This assumption inquires to distinguish *noisy* linear equations from uniformly random.

Our protocol borrows some basic ideas from Hill cipher in [29] that is computationally hard under matrix multiplication with random permutations. We use pseudo-inverse matrix in order to transfer session key from the server to the tag and to offer The most widely known and popular pseudo-inverse is the **Moore-Penrose** pseudo-inverse, which was described by E. H. Moore [15].

[1]Where both the channels: tag/reader and reader/server are assumed to be insecure.

TABLE I
Notations used in the paper

$\mathbb{Z}_p$	set of integers modulo an integer $p \geq 1$
$l \in \mathbb{N}$	length of the Tag's ID
$v \in \mathbb{N}$	length of the commitment message s.t., $v \leq l$
$k \in \mathbb{N}$	length of the secret key s.t., $k \leq (l+v)$
$\mathbf{T_{id}}$	l bit unique ID of a tag
$\mathbf{I_i}$	$k$bit index of the tag during time period i
$\mathbf{P_i}$	$k \times k$ bit matrices as the session key for the reader during time period i
$\mathbf{S'_i}$	$k \times l$ bit matrices as the *secret* commitment key between the server and the tag during time period i
$\mathbf{S''_i}$	$k \times v$ bit matrices as the *session* key between the server and the tag during time period i
s	v bit random binary vector generated by the reader
σ_i	a lightweight signature on a message s
s'	v bit random binary vector generated by the tag
$\mathbf{w}(\cdot)$	Hamming weight of any vector
τ	Parameter of the Bernoulli error distribution $\mathbf{Ber}_\tau$ where $\tau \in]0, 1/4[$
τ'	Authentication verifier acceptance threshold (Tag/Reader) where $\tau' = 1/4 + \tau/2$
e	k bit vector from Bernoullli distribution $\mathbf{Ber}^k_{k\tau}$ with parameter $k\tau$ s.t., $Pr[e=1] = k\tau$
$\mathbf{Q}$	$k \times k$ bit randomly generated non-singular binary matrices by the server
$[\mathbf{S}]^\mathbf{T}$	transpose of matrix $\mathbf{S}$ i.e., $T : \mathbb{Z}_2^{k\times v} \rightarrow \mathbb{Z}_2^{v\times k}$
$\mathbf{A^+}, \mathbf{A^{-1}}$	pseudo-inverse and inverse of a matrix $\mathbf{A}$ respectively i.e., $\mathbf{A}^{(+/-)} : \mathbb{Z}_2^{k\times v} \rightarrow \mathbb{Z}_2^{v\times k}$
$\oplus, \|$	bitwise XOR operation, concatenation of two vectors
$\bigvee$	logical OR operation
$\lfloor x \rceil$	the nearest integer to x
$]a, b[$	$x \in \mathbb{R}$ s.t., $a < x < b$

Since an RFID tag is not tamper-resistant, its session key $\mathbf{S}''_i$ is refreshed after each i^{th} session completes successfully. To update the key, each tag authenticates not only its licit reader but also the legitimate server. In addition, we assume the tag identifier T_{id} be unique and secure within an RFID system. However, an adversary cannot corrupt the reader and the tag until it compromises their secrets P and (T_{id},$\mathbf{S}$) respectively at a time.

Nevertheless, if all the secret keys are exposed at a time, the adversary can trace the tag for a period i until the next authentication cycle starts. To avoid exhaustive database search at the server hash-index I_i is used. Database at the server associates the tag index with other tag related data e.g., T_{id},$\mathbf{S}_i$,P_i etc.

B. Security definitions

Some of the notations and security definitions we use from [25]. However, we omit the description of some definitions due to space constraint. Interested readers are referred to [25] for a through discussion.

Definition 1. Let for a noise-parameter τ, a k-bit *Bernoulli distribution* Ber_τ^k output 1 with probability τ and 0 with probability $(1-\tau)$. First, we define the *decisional* version of LPN. For $k, l \in \mathbb{Z}_2$, let $\mathbf{Ber}_\tau$ be an error distribution over $\mathbb{Z}_2^k$. The *decisional* LPN problem is (t, Q, ϵ)-hard if for all (Q, t) adversary $\mathcal{A}$ can distinguish uniform binary vector (r) from noisy inner products of vector $(A.x \oplus e)$ such that:

$$\Pr\ [\mathcal{A}(A, A.x \oplus e) = 1] - \Pr[\mathcal{A}(A, r) = 1]] \leq \epsilon$$
$$\Pr\ [\mathcal{A}(A, A.x \oplus e) = x] \leq \epsilon$$

where $A \in_R \mathbb{Z}_2^{k \times l}, e \in_R \mathbf{Ber}_\tau^k, r \in_R \mathbb{Z}_2^k$; and $x \in_R \mathbb{Z}_2^l$ is the secret. Let $y = A.x \oplus e$. However, the *computational* LPN problem is to compute x and low weight e from a given (A, y) pair.

Note that, in the standard definition of the LPN problem, the error distribution is the Bernoulli distribution with some parameter $0 < \tau < 1/2$ where e is sampled uniformly from $\mathbf{Ber}_\tau^k$ such that, $\Pr[e[i] = 1] = \tau$ and $\mathbf{w}(\cdot) \leq k\tau$. However, in case of *exact*-**LPN** in [28], the problem is defined exactly like LPN except that the hamming weight $\mathbf{w}$ of the error vector is defined exactly $\lfloor k\tau \rceil$ such that $\mathbf{w}(\cdot) \leq \lfloor k\tau \rceil$. That means, e is chosen independently and identically from $\mathbf{Ber}_{\lfloor k\tau \rceil}^k$.

Definition 2. In *Hill Cipher* described in [29], a plaintext vector $X \in \mathbb{Z}^k$ is encrypted to get ciphertext Y as:

$$Y = XK \pmod m \in \mathbb{Z}_m^k$$

where the key $K \in \mathbb{Z}_m^{k \times k}$ is an invertible matrix, $\mathbb{Z}_m$ is a ring of integers modulo m and $gcd(\text{Det } K, m) = 1$. The encryption procedure proceeds by encoding the resulted ciphertext row vector into alphabets of the main plaintext. The value of m in the Hill cipher was 26 but its value can be optionally selected. It requires the key matrix K be shared between the participants. However decryption is straight forward:

$$X = YK^{-1} (\text{mod } m).$$

Definition 3. *Commitment scheme* is a two-phase protocol between a *sender* and a *receiver* where the sender holds a message m and, in the first phase, it picks a random key ck and then encodes m using ck and sends the encoding message c (a commitment to m) to the receiver. In the second phase, the sender sends the key ck to the receiver and it can open the commitment and find out the content of the message m.

More formally, A triple of algorithms **(KGen, Com, Ver)** is called a commitment scheme if it satisfies the following:

- On input 1^l, the key generation algorithm **KGen** output a commitment key ck.
- The commitment algorithm **Com** takes as input a message m from a message space $\mathcal{M}$ and a commitment key ck, and output a commitment-opening pair (c, d).
- The verification algorithm **Ver** takes a key ck, a message m, a commitment c and an opening d and output 1 or 0.

Our construction is based on the LPN-based commitment scheme [28] but customized to work with the authentication protocol.

III. Construction

We adopt three different cryptographic tools: LPN based commitment scheme, pseudo inverse matrix properties and a secure variant of Hill cipher in order to achieve 3-round mutual authentication protocol described in Fig. 1. We use the term *fully-secure*, because the protocol attains mutual authentication not only in Tag-Reader pair, but also in Reader-Server. The protocol is partitioned/organized into a hierarchy of computation units. Therefore, it sets aside significantly less computations to the tag. On the other hand, the most expensive computations of the protocol are handled by the server. We use only random vector generation, bitwise XOR and matrix multiplication as tag operation. The protocol uses (τ, k, l, v, τ') as public parameters, while (τ, τ') are constant and (l, k, v) depends on the security level. In the setup phase, Server generates the initial index I_0, the permanent key $\mathbf{S}'$, the session key $\mathbf{S}_0''$ and its corresponding $\text{P}_0 \leftarrow \mathbf{S}_0''\mathbf{S}_0''^{+\dagger}$ and other public parameters; and set them into a tag non-volatile memory and into the reader. Note that, we use different secret keys for entities. For instance, T_{id} is shared among three entities of the protocol. In contrast, each tag has 2 secrets ($\mathbf{S}'$,$\mathbf{S}''$) and each reader has 1 secret (P) respectively to share with the server. However, for any time instance i a tuple $[\text{I}_i, \text{T}_{id}, \mathbf{S}'_{i-1}, \mathbf{S}', \mathbf{S}''_i, \text{P}_{i-1}, \text{P}_i, r_i]$ needs to be stored in the back-end database of the server while a reader needs to memorize $[\text{P}_{i-1}, \text{P}_i, \text{P}_i^{-1}]$.

For *tag* authentication, a tag holds $\mathbf{S}''_i$ and I_i that have been derived from the previous $(i-1)$ successful sessions.

- Reader: Generate a random binary v-bit challenge string s, and sends it to a tag.
- Tag: Check the *hamming weight* of the string s and generate a k-bit noise vector e from Bernoulli distribution $\mathbf{Ber}_{k\tau}$, a random v-bit challenge string s' with hamming weight $v/2$. Next a k-bit commitment string r on the message s is generated as $r := \mathbf{S}_i \cdot (\text{T}_{id} \parallel s) \oplus e$. Note that, $\mathbf{S}_i$ consists of 2 keys: the permanent key $\mathbf{S}'_i$ and the session key $\mathbf{S}''_i$.
  In addition, σ_i is generated by using the key $\mathbf{S}''_i$ to demonstrate the authenticity of the message s and to give an impression to the reader that it was created by its known tag. In order to extinguish brute-force searching at the server end, an index I_i is maintained and updated each time ($\text{I}_{i+1} \leftarrow r$) by the tag. Finally, the tag forwards $(\text{I}_i, r, \sigma_i, s')$ to the reader.

$^\dagger\mathbf{S}_0''^+$ is the pseudo-inverse of the matrix $\mathbf{S}_0''$ by following the algorithm in [7]

Reader $(\mathbf{P_i}, \mathbf{T_{id}})$	**Tag** $(\mathbf{I_i}, \mathbf{T_{id}}, \mathbf{S_i}', \mathbf{S_i}'')$
$s \in_R \mathbb{Z}_2^v$; s.t. $\mathbf{w}(s) = v/2$	
$\xrightarrow{s}$	
	If $\mathbf{w}(s) \neq v/2$ **return**;
	$e \in_R \mathbf{Ber}_{k\tau}^k$
	$\mathbf{S_i} = \mathbf{S_i}' \parallel \mathbf{S_i}'' \in \mathbb{Z}_2^{k\times(l+v)}$
	$r := \mathbf{S_i} \cdot (\mathbf{T_{id}} \parallel s) \oplus e$
	$\sigma_\mathbf{i} = \mathbf{S_i}'' \cdot s$
	$s' \in_R \mathbb{Z}_2^v$ s.t., $\mathbf{w}(s') = v/2$
	$\underleftarrow{(\mathbf{I_i}, r, \sigma_i, s')}$

- Reader: The reader conveys the messages it received from the tag. But before forwarding, it apparently verifies the tag with σ_i, whether it is generated from the challenge s. Note that $\mathrm{P}_i\sigma_i = \mathbf{S}_i''\mathbf{S}_i''^{+}\mathbf{S}_i''s = \mathbf{S}_i''s = \sigma_i$. Subsequently, it also checks the hamming weight of s'.
- Server: First search the database with I_i in order to find out a tuple [I_i,T_{id},$\mathbf{S}_i''$,r_{i-1},$\mathbf{S}_{i-1}''$]. Note that $(r_{i-1},\mathbf{S}_{i-1}'')$ would be stored to resist synchronization attack. However, searching with index I_i might fail sometimes e.g., due to synchronization attack etc. In that case, server could apply brute-force searching method[‡] targeting to explore the previous transaction parameters: $(\mathbf{S}_{i-1}'', r_{i-1})$. Then, given a commitment r on a message s sent by the reader, it accepts the commitment if and only if: $\mathbf{w}(\mathbf{S}_i \cdot (\mathrm{T}_{id} \parallel s) \oplus r) \stackrel{?}{=} \lfloor k\tau' \rceil$ and $\mathbf{w}(s') \stackrel{?}{=} v/2$ where s' is the new challenge (commitment) message for the server. Consequently, it accepts the tag, update the index to I_{i+1} and enter *server/reader* authentication phase.

For *server authentication*, it has secret: (T_{id},$\mathbf{S}_i'$,$\mathbf{S}_i''$) and (T_{id},P_i) respectively shared with the tag and the reader, where except (T_{id},$\mathbf{S}_i'$), the rest of the parameters would have been derived directly from the previous $(i-1)^{th}$ successful authentication session.

- Server: First generate a non singular binary matrix Q to update session key $\mathbf{S}_{i+1}''$ as $[Q{\cdot}\mathbf{S}_i'']$ for the next $i+1$ session and compute pseudo inverse-matrix $\mathbf{S}_{i+1}''^{+}$, and P_{i+1} as $\mathbf{S}_{i+1}''{\cdot}\mathbf{S}_{i+1}''^{+}$. In order to send the new session key $\mathbf{S}_{i+1}''$ to the tag and blinding the matrix Q, P_i' is computed by $\mathrm{P}_i \cdot Q$ which is actually equivalent to $\mathbf{S}_i\mathbf{S}_i^{+}Q$.
  Subsequently, a k-bit commitment r' on s' will be generated with a view to authenticate server to the tag: $r' := \mathbf{S}_i \cdot (\mathrm{T}_{id} \parallel s') \oplus e'$, where e' is a k-bit randomly generated noise vector.
  After this, P_i'' is generated in order to update P_i at the reader, where Q^{-1} is used to randomize P_{i+1}. Finally, compute s'' from P_i'': $s'' := \mathrm{P}_i'' \cdot (\mathrm{T}_{id} \parallel s') \oplus e'$ for authenticating server to the reader. Subsequently, the communication string $(\mathrm{P}_i', \mathrm{P}_i'', r', s'')$ is forwarded to the reader.

[‡]Server can search [$\mathrm{I}_i \stackrel{?}{=} r_{i-1}$] the database with previous index stored for $(i-1)^{th}$ session.

Server $(\mathbf{I_i}, \mathbf{T_{id}}, \mathbf{S_i}', \mathbf{S_i}'', \mathbf{P_i})$	**Reader** $(\mathbf{P_i}, \mathbf{T_{id}})$
	If $(\mathbf{P_i} \cdot \sigma_i \neq \sigma_i \bigvee \mathbf{w}(s') \neq v/2)$ **return;**
	$\underleftarrow{(\mathbf{I_i}, r, s, s')}$
Lookup $\mathbf{T_{id}}$ by using I_i:	
Direct match: **If** $(\mathbf{I} \neq \mathbf{I_i})$ then	
Brute-force search:	
$\exists$ $(\mathbf{T_{id}}, \mathbf{S_i}''$ or $\mathbf{S_{i-1}''})$ that satisfies:	
$\mathbf{S_i} = \mathbf{S_i}' \parallel \mathbf{S_i}''$	
If $\mathbf{w}\big((\mathbf{S_i} \cdot (\mathbf{T_{id}} \parallel s) \oplus r) \neq k.\tau'$	
$\bigvee \mathbf{w}(s') \neq v/2\big)$	
return;	
$\mathbf{I_{i+1}} = r$	
Generate non-singular $\mathbf{Q} \in_R \mathbb{Z}_2^{k\times k}$	
$\mathbf{S_{i+1}''} = \mathbf{Q} \cdot \mathbf{S_i}'' \in \mathbb{Z}_2^{k\times v}$	
where *rank*$(\mathbf{S_{i+1}''}) = v$	
$\mathbf{S_{i+1}''}^{+} := (\mathbf{S_{i+1}''}^{\mathbf{T}}\mathbf{S_{i+1}''})^{-\mathbf{1}}\mathbf{S_{i+1}''}^{\mathbf{T}} \in \mathbb{Z}_2^{v\times k}$	
$\mathbf{P_{i+1}} := [\mathbf{S_{i+1}''}] \cdot [\mathbf{S_{i+1}''}]^{+} \in \mathbb{Z}_2^{k\times k}$	
$\mathbf{P_i}' := \mathbf{P_i} \cdot \mathbf{Q} \in \mathbb{Z}_2^{k\times k}$	
$\mathbf{S_i} = \mathbf{S_i}' \parallel \mathbf{S_i}''$	
$e' \in_R \mathbf{Ber}_{k\tau}^k$;	
$r' := \mathbf{S_i} \cdot (\mathbf{T_{id}} \parallel s') \oplus e'$	
$\mathbf{P_i''} := \mathbf{Q^{-1}} \cdot \mathbf{P_{i+1}} \in \mathbb{Z}_2^{k\times k}$	
$s'' := \mathbf{P_i''} \cdot (\mathbf{T_{id}} \parallel s') \oplus e'$	
$\xrightarrow{(\mathbf{P_i}', \mathbf{P_i''}, \mathbf{r}', s'')}$	

- Reader: First check the *hamming weight*: $\mathbf{w}(\mathrm{P}_i'' \cdot (\mathrm{T}_{id} \parallel s') \oplus s'') \stackrel{?}{=} \lfloor k\tau' \rceil$. It ensures P_i'', consequently, (Q,P_i') to be generated by the server; and hence, server is authenticated. If any of the parameters is replicated during transmission the above equation will not hold. Then the reader updates P_i by using Hill deciphering technique: $\mathrm{P_i'P_i^{-1}P_i''} = \mathrm{P_i}\ Q\ \mathrm{P_i^{-1}Q^{-1}P_{i+1}} = \mathrm{P_{i+1}}$. Note that $\mathrm{P_i^{-1}}$ can be precomputed and stored in the reader for efficiency.

Reader $(\mathbf{P_i}, \mathbf{T_{id}})$	**Tag** $(\mathbf{I_i}, \mathbf{T_{id}}, \mathbf{S_i}', \mathbf{S_i}'')$
If $\mathbf{w}\big(\mathbf{P_i}'' \cdot (\mathbf{T_{id}} \parallel s') \oplus s''\big) \neq k.\tau'$	
return;	
$\mathbf{P_{i+1}} := \mathbf{P_i'} \cdot \mathbf{P_i^{-1}} \cdot \mathbf{P_i''} \in \mathbb{Z}_2^{k\times k}$	
$(\mathbf{P_i}', r')$	
$\longrightarrow$	
	If $\mathbf{w}(\mathbf{S_i} \cdot (\mathbf{T_{id}} \parallel s') \oplus r') \neq k.\tau'$ **return**;
	$\mathbf{S_{i+1}''} = (\mathbf{P_i}' \cdot \mathbf{S_i''}) \in \mathbb{Z}_2^{k\times v}$
	if *rank*$(\mathbf{S_{i+1}''}) \neq v$ **return**;
	$\mathbf{I_{i+1}} = r$

For *reader authentication*, it has *shared secret* T_{id} to the tag. It is quite certain that the reader would forward the protocol message (P_i', r') to the tag if it could verify the hamming weight equation $\mathbf{w}(\cdot) \stackrel{?}{=} \lfloor k\tau' \rceil$ successfully.

- Tag: Verify the commitment r' on the message s' by

checking the *hamming weight* of $(\mathbf{S_i} \cdot (\mathbf{T_{id}} \parallel s') \oplus r')$ is exactly $\lfloor k\tau' \rceil$. If the check passes, accept the reader as well as the server and update the session key to $\mathbf{S''_{i+1}}$ [i.e., $\mathbf{S''_{i+1}} = \mathbf{P_i}' \cdot \mathbf{S''_i} = \mathbf{S''_i S''^{+}_i S''_i Q} = \mathbf{Q S''_i}$][§], the session index to $\mathrm{I}_{i+1} = r$. However, if the check fails, tag's session key remains unchanged.

Note that, in the protocol, session keys are generated and updated at i^{th} instance by the server and later followed by the reader and the tag. To be precise, session key is updated in each transaction of the protocol: inside the tag $\mathbf{S}''_{i+1}$ by randomizing the former key $\mathbf{S}''_i$ with Q, and inside the reader P_{i+1} by secure Hill cipher.

IV. Security Analysis

A. Commitment Scheme

A commitment scheme should satisfy *three* security properties: correctness, prefect hiding and binding. Our constructing satisfies the following security properties:

- *Correctness*: $\mathbf{Ver}(ck, m, c, d)$ should result to 1 if the inputs are computed by an honest party, such that,
$$\Pr[\mathbf{Ver}(ck, m, c, d) = 1;\ ck \leftarrow \mathbf{KGen}(1^l), m \in \mathcal{M}, (c, d) \leftarrow \mathbf{Com}(m, ck)] = 1$$
- *Computation hiding*: Receiving a commitment c to a message m should give no information to the receiver about m. A commitment c computationally hides the committed message with overwhelming probability over the choice of ck, s.t.,
$$\Pr[ck \leftarrow \mathbf{KGen}(1^l); \forall\ m, m' \in \mathcal{M} \wedge (c, d) \leftarrow \mathbf{Com}(m, ck), (c', d') \leftarrow \mathbf{Com}(m', ck) : c = c'] = 1/2$$
- *Perfect binding*: It means that the *sender* cannot cheat in the second phase and sending a different commitment key ck' causes the commitment to open to a different message m'. That is, with overwhelming probability over the choice of the commitment key $ck \leftarrow \mathbf{KGen}(1^l)$, no commitment c can be opened in two different ways, s.t.,
$$\Pr[(\mathbf{Ver}(ck, m, c, d) = 1) \wedge (\mathbf{Ver}(ck, m', c, d') = 1) : m \neq m'] \leq \epsilon$$

In order to ensure the commitment scheme is *hard* enough, the length of the parameter l should be chosen carefully. Although the length of the challenging messages ($|s| = |s'| = v$) can be chosen arbitrarily, but for efficiency reasons it is better to choose the same size as l. In our protocol, we consider $k = v + l$ s.t., $v = l$, where k would be large enough to make the commitment scheme accomplished computationally hiding and perfectly binding with high probability over the choice of secret matrix $\mathbf{S}$. Note that *binding* property is ascertained by large distance of the code generated by the random matrix $\mathbf{S}''$, while the hiding property directly from the LPN assumption that outputs pseudo random string r or r'.

Theorem 1. *Let decisional exact* LPN_x *be hard under* $\tau \in]0, 1/4[, (k, l, v) \in \mathbb{Z}$, *and* $k = \mathcal{O}(l + v)$. *And for any* $\mathbf{S} \in_R \mathbb{Z}_2^{k\times(l+v)}$ *such that,* $\mathbf{w}(\boldsymbol{S}\cdot x) > 2\lfloor k\tau \rceil$, *where* $x \in_R \mathbb{Z}_2^{l+v}$. *Then the commitment scheme used in the protocol is perfectly binding and computationally hiding.*

[§]From the properties of pseudo-inverse matrix ($AA^{+}A = A$).

Proof: Assume $[(\mathbf{T}_i, s_i)$ for $i = 1, 2]$ be two different openings for a commitment r. Then, $e_i = r \oplus \mathbf{S} \cdot (\mathbf{T}_i \parallel s_i)$, and norm of e_i for $i = 1, 2$ is at most $\lfloor k\tau \rceil$. Therefore, $e_1 \oplus e_2 = \mathbf{S} \cdot (\mathbf{T}_1 \parallel s_1 \oplus \mathbf{T}_2 \parallel s_2)$ and $\mathbf{w}(e_1 \oplus e_2) \leq \mathbf{w}(e_1) + \mathbf{w}(e_2) \leq 2\lfloor k\tau \rceil$ which contradicts our initial assumption $\mathbf{w}(S \cdot x) > 2\lfloor k\tau \rceil$, thus, satisfies *perfect binding* property. On the other hand, it would appear that we have

$$r = \mathbf{S}' \cdot \mathbf{T} \oplus e \oplus \mathbf{S}'' \cdot s$$

Since $\mathbf{S}' \cdot \mathbf{T} \oplus e$ is pseudorandom from the exact LPN_x assumption, r is also pseudorandom. Thus, distribution of r is computationally indistinguishable and hence, satisfies *computational hiding* property. □

Theorem 1.1. *The commitment protocol from LPN described in Fig. 1. is computationally indistinguishable.*

Proof: If a commitment c computationally hides the committed message with overwhelming probability, the distributions of the commitments are computationally indistinguishable. From Theorem 1. we conclude that *decisional exact* LPN_x is perfectly computationally hiding. Let a prover and verifier share a common input y and the prover has a private secret input x. Therefore, for a binary relation $\mathcal{R}$ such that $(x, y) \in \mathcal{R}$. Then For every potentially malicious (Q, t)-adversary $\mathcal{A}$, there exists a PPT simulator V^*, that takes y as an input, but its output is indistinguishable from an honest prover's conversations. In [28], authors describe an efficient simulator for indistinguishability game, where for each challenge c outputs an accepting protocol transcript the distribution of which is computationally indistinguishable from real protocol transactions with an honest prover for challenge c. For more detail clarification, we refer to the respected literature. However, due to the fact that bernoulli random noise might exceed the acceptable threshold, false rejection and false acceptance probability will be:

$$\mathrm{P}_{FA} = \sum_{i=0}^{\tau k} \binom{k}{i}\ 2^k \text{ and } \mathrm{P}_{FR} = \sum_{i=\tau k+1}^{k} \binom{k}{i} \tau^i (1-\tau)^{(k-i)}$$

B. Pseudo-random matrix

We followed the security analysis in [16], where it is claimed that, having known the messages $\mathbf{X}\mathbf{X}^{+}\mathbf{Q} \in \mathbb{Z}_2^{k\times k}$, it is impossible to recover the secrets $\mathbf{X} \in \mathbb{Z}_2^{k\times v}$, or $\mathbf{Q} \in \mathbb{Z}_2^{k\times k}$.

However, to ascertain security, we need to ensure that $k \gg v$, that can be obtained with $k = \Theta(v + l)$. So, we let $|v| = |l|$ to ensure a large value of k.

Theorem 2: *If* X *is pseudo-invertible then its pseudo-inverse matrix* X^{+} *is unique*.

Proof: Assume, as contraposition, that Y, Z be two pseudo-inverse matrices of X. Therefore, from the property of pseudo-inverse matrix we have $\mathrm{XYX} = \mathrm{X}$ and $\mathrm{XZX} = \mathrm{X}$. It appears that $(\mathrm{XYX})^{\mathrm{T}} = \mathrm{X^T Y^T X^T} = \mathrm{X^T} = \mathrm{X^T Z^T X^T} = (\mathrm{XZX})^{\mathrm{T}}$. Similarly, $\mathrm{YXY} = \mathrm{Y}$ and $\mathrm{ZXZ} = \mathrm{Z}$. Thus,

$\mathrm{XY} = (\mathrm{XY})^{\mathrm{T}} = \mathrm{Y^T X^T} = \mathrm{Y^T}(\mathrm{X^T Z^T X^T}) = (\mathrm{XY})^{\mathrm{T}}$
$(\mathrm{XZ})^{\mathrm{T}} = \mathrm{XYXZ} = \mathrm{XZ}$, that implies $\mathrm{YX} = \mathrm{ZX}$.

Finally, we conclude with $\mathrm{Y} = \mathrm{YXY} = \mathrm{ZXY} = \mathrm{ZAZ} = \mathrm{Z}$ that contradicts the initial assumption. Hence, pseudo-inverse matrix exists uniquely. □

C. Secure Hill Cipher

The security of the ordinary Hill cipher relies on the rank of Key matrix $rank(K)$. However, Hill cipher succumbs to the most popular *Chosen Plaintext Attack* (CPA) that is in effect a linear transformation on the message space.

Theorem 3. *Hill cipher used in the protocol described in Fig. 1. can resist CPA attack.*

Proof: We use the matrix P_i as the secret symmetric key for the Hill cipher and show that P_i is the only matrix that can decrypt the cipher P''_i correctly. We use non-singular matrix $Q \in \mathbb{Z}_2^{k\times k}$ as the *permutation matrix* in the scheme while P_{i+1} is the *message* to transfer from the server to the reader. We could consider a special case: $Q^{-1} = Q^T$ when $QQ^T = Q^TQ = I$ where I is the identity matrix. For contradiction, suppose there is a non-singular matrix G exists, such that G=P_i. In that case for every valid (P''_i,P_{i+1}, Q) there exist, $\mathrm{G}^{-1}\mathrm{P}''_i\mathrm{P}'_i$ =P_{i+1}. This clearly concludes that whatever Q is, we have G=P_i. This should also hold for Q such that $QG = Q\mathrm{P}_i$ =P'_i, but that is not possible. So the only matrix that can decrypt successfully is P_i^{-1} that contradicts our assumption on G. Since CPA attack enquires k-pairs of plaintext-ciphertext pairs,using a linear transformation by a fixed matrix leads to linear dependency that results weak security. In our scheme, both Q and P_i is refreshed in each session. It is like *one time one key matrix* for each block ciphering where the key has been derived from the preceding key matrix i.e., P_{i+1} $\leftarrow$$\mathrm{P}_i$. More concisely, we use two different matrices: one is to randomize P_{i+1} by permutation matrix Q, another is to convey Q. However, commitment s'' is generated on the message s' by the commitment key P''_i from LPN. Therefore, reader can verify the commitment and hence the permutation matrix Q.

Let rewrite the ciphertext $\mathrm{P}''_i \in \mathbb{Z}_2^{k\times k}$ as: P''_i =$\mathrm{P}_i\mathrm{P}'^{-1}_i\mathrm{P}_{i+1}$ such that, Y = HZX $(mod\ 2)$ for simplicity. Since Q is refreshed at each transaction, the equation can be written as follows:

$$\begin{aligned} Y_0 &= HZ_0X_0 \ (mod\ 2)\\ Y_1 &= X_0Z_1X_1 \ (mod\ 2)\\ Y_2 &= X_1Z_2X_2 \ (mod\ 2)\\ &\vdots\\ Y_k &= Y_{k-1}Z_kX_k \ (mod\ 2) \end{aligned}$$

It can be clearly seen from the above equations: although the attacker knows k-pairs of (Y X), k equations cannot be used to solve a $k \times k$ non-singular matrix P_i at any time instance i that resist CPA attack.

Let a valid ciphertext-plaintext pair (P''_i,P_{i+1}) with a permuting matrix Q yield a set of key matrices G_q. Then the number of solution matrices for G_q is $2^{k(k-rank(\mathrm{P}_{i+1}))}$. Although the knowledge of all valid pair (P''_i,P_{i+1}) is sufficient to determine P_{i+1}, but it demands exponential time/memory considering the size of the set G_q. Therefore, the probability that a key matrix $G \in G_q$ decrypts correctly a randomly and uniformly chosen pair (P''_i,P_{i+1}) is negligible $(1/2^{k(k+1)})$. In the optimal case, this probability is $1/2^{k^2}$ where a non-trivial permutation matrix is used. □

D. Secure Exact LPN

Proposition 1. *The hardness of decisional* LPN_x *is polynomially related to that of search* LPN_τ.

Proof: Hardness of the LPN_x problem holds assuming the hardness of the standard LPN_τ problem; the reduction is based on the Goldreich-Levin theorem described in [24]. Note that if the security of the scheme be considered on the standard LPN assumption in a provable manner, there is no efficient attacks against LPN_x than against LPN_τ. However, if the loss in the reduction is taken into account, it might result in large parameters. The security of the commitment scheme is directly based on the standard LPN_τ. Actually it replaces the LPN_τ assumption with an assumption where the upper bound on the weight of the error vector is fixed, i.e., $\lfloor k\tau \rceil$, thus removes the completeness error. In [28], authors show a protocol for proving knowledge of committed values whose security relies directly on the standard decisional LPN_τ assumption. However, the protocol has a soundness or knowledge error 4/5, and thus requires running the protocol roughly twice in order to achieve the same knowledge error. Interested readers are referred to [28], for further clarification and proof of the theorem.

E. Man-in-the Middle Attack

The most sophisticated and realistic attack in an RFID system is the Man-in-the Middle (MIM) attack. Our protocol is MIM-secure against an active attack from several assumptions i.e, the exact LPN, secure Hill cipher and pseudo-inverse matrix properties. In case of tag-reader, the authentication tags $(\gamma_1, \gamma_2) \leftarrow [(\mathrm{I}_i, r, \sigma_i, s'), (\mathrm{P}', r')]$ is MIM-free: $\gamma_1 = (s, \sigma : f_{k_1}(s))$, $\gamma_2 = (s', r' : \mathbf{S} \cdot \bar{f}_{k_2}(s') \oplus e')$ where $(f_{k_1}, \bar{f}_{k_2})$ are secret key derivation functions which uniquely encode challenges resp. s and s' according to the keys (k_1, k_2) where we use resp. $\mathbf{S}''$ and $(\mathbf{S}, \mathbf{T}_{id})$ as the secret keys (k_1, k_2). The main technical difficulty to build a secure MIM-free authentication from LPN is to make sure the secret key k_i does not leak from verification queries. Since we randomize $\mathbf{S}''$, and hence $\mathbf{S}$ at every protocol session i and Theorem 1.1. at page 5 shows that protocol transcripts are computationally indistinguishable from the exact LPNx assumption, the tag-reader communication is MIM-secure. On the other hand, reader-server authentication tag $(\gamma_3, \gamma_4) \leftarrow [(\mathrm{I}_i, r, s'), (\mathrm{P}', \mathrm{P}'', r', s'')]$ is MIM-free from exact LPN_x (γ_3 likewise γ_2) and secure Hill cipher assumption. Let $\gamma_4 = (\mathrm{P}', \mathrm{P}'' : \hat{f}_{k_3}(\mathrm{P}', \mathrm{P}''))$ be an authentication tag for Hill cipher, where $\hat{f}_{k_3}(\cdot)$ is the secret key derivation function with the secret key $k_3 \leftarrow \mathrm{P}^{-1}$. Since the variation of Hill Cipher used in this protocol can resist Chosen plaintext attack as described in Theorem 3. at page 6 and we update $\mathbf{P}$ at each protocol session i, the reader-server communication is MIM-secure. In addition, we use a pseudo-random matrix as blinding factor that is secure under pseudo-inverse matrix properties. Therefore, even if the adversary compromises T_{id}, it cannot generate $\mathbf{S}''$ and hence $\mathbf{S}$ for any subsequent sessions using only T_{id}.

V. Privacy

In order to define privacy, we analyzed our protocol according to the privacy framework based on *zero-knowledge* (ZK)

formulation described in [33]. 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 mutual authentication protocol follows ($\pi = 3 \ s.t. \ \lambda = 1$) the same framework. Due to space constraint, we refer to the definitions of generic oracles from [33].

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 $\mathbf{Exp}_{\mathcal{A}}^{ZK}(\hat{T})$ 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*$_{\hat{\mathcal{A}}}$)
  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 is negligible that defined as

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

Theorem 4. *From the exact* LPN *problem, the protocol described in Fig. 1 satisfies ZK-privacy.*

Proof: Due to lack of space, we remove the proof of the above theorem. That will appear in the full version.

Theorem 5. *An RFID protocol described in* Fig. 1. *is forward (resp., backward)-ZK private.*

Proof: ZK-privacy allows to give the secrets to the adversary $\mathcal{A}$ at the end of the experiment. Let a pair (k^f, s^f) be a final key (k) and internal state (s) of a challenged tag T_c from the initial (k^0, s^0). Then the protocol is forward (resp., backward)-ZK private if any PPT distinguisher $\mathcal{D}$ cannot distinguish $(k^f, s^f, c, T_c view_{\mathcal{A}}(\kappa, l))$ from $(k^f, s^f, c, T_c, sview_{Sim}(\kappa, l))$ after the oracle **Ukey**(.) is run by $\hat{\mathcal{A}}_2$. Note that T_c should not be in the oracle table D (related to **DTag**(.)) before the experiment $\mathbf{Exp}^{ZK}(\hat{\mathcal{T}})$ ends. However, *forward (resp., backward)-ZK privacy* cannot be achieved if $\mathcal{A}$ has corrupted the challenging tag T_c before the experiment finishes. □

TABLE II
TAG RESOURCES AND SECURITY COMPARISON WITH HB FAMILY

Scheme	Storage	Computation (major)	Authentication	Security achieved	Hardware (gates)
HB-MP [11]	2 S	1 LPN	tag	5,6,8	≈ 1600
HB-MP$^+$ [21]	2 S	1 LPN,1 HASH	tag	1,5,6,8	≈ 3500
GHB# [34]	2 S	1 LPN	tag	1,5,6,9	≈ 1600
[20]	1 S	1 SC	mutual	2,4*,7,8	≈ 2000
F-HB [18]	1 I , 1 S	1 PRNG,2 LPN	mutual	$1, 2, 4^*, 5, 6, 9$	≈ 3500
[25]	1 I, 1 S	1 SLPN,1 P	mutual	1,2,3*,4,5,6,7,8	≈ 1600
ours	1 I, 2 S	1 LPNx,1 P, 1 H	Full mutual	1,2,3*,4,5,6,7,8	≈ 2000

where SC:= Stream Cipher; S:= Secret key; I:= Index; H:= Hill cipher; PRNG:= Psudo Random Number Generator; P:= Pseudo Inverse Matrix; LPN:= Learning parity from noise SLPN:= Subset LPN; LPNx := exact LPN
Security attributes: MIM attack(1), Forward Security (2), Backward Security (3), Reduced Backward Security (3*), High Privacy (4), Limited Privacy (4*) Tag tracking (5), De-synchronization (6), Replay attack (7), DoS (8).

VI. COMPARISON AND PERFORMANCE

Computation Requirement: We focus on tag, which is the computationally weakest. Most of the expensive computations will be performed at the server site. The *exact* version of the LPN problem used in the protocol is of independent interest as this assumption removes the completeness error [28]. Setting $v = l$ in the public parameters, it results $k = \theta(v + l) = \theta(v)$ and commitment scheme requires $2^{\theta(v/\ log\ v)}$ time. Thus, commitment proof is quasi-linear in the length of the committed messages.

Major protocol operations regarding the tag include *one* LPN problem generation and checking and *two* binary linear matrix multiplications. As bitwise XOR, matrix multiplication, and calculating the hamming weight $w(\cdot)$ are all binary operations, they can easily be implemented using bit-by-bit serialization to save hardware gates.

In order to compute a Hill ciphertext with randomized permutation need $2k$ vector products over $\mathbb{Z}_2$ If the vectors are stored in *words*, the vector product can be simply reduced to a *logical* AND (&) and *parity check* operations. Therefore, $\sum_{i=1}^{k} a_i b_i (mod\ 2)$ is equivalent to $a\&b$ that needs only $12k$ operations [30]. In decryption case (in the reader), we need $3k$ vector products over $\mathbb{Z}_2$ and an inverse operation that can be pre-computed to enhance efficiency. That's why, we need k^3 (multiplication) $+(k^3 - k^2)$ (addition) over $\mathbb{Z}_2$.

Storage Requirement: All the parties in the protocol need to store the public parameters. However, a tag needs to store only 2 secret keys and an index for the session $(k \cdot l + k \cdot v + k)$ bits, a reader requires to store a tag identifier and 1 secret key $(k^2 + l)$ bits while the server needs to maintain a database for all the tags (for session i and $i - 1$) with index, tag identifier and 3 secret keys $(2k \cdot l + 2k \cdot v + 2k^2 + 2k + l)$ bits for each tag. Consequently, storage requirement for the tag and the reader

can be expressed by $\mathcal{O}(1)$ while that is $\mathcal{O}(n)$ for the server such that n is the number of tags in an RFID system.

Communication complexity: The protocol requires $(k^2 + 2v + 4k)$ bits in the tag-reader communication and $(2k^2 + 2k + 3v)$ bits in the reader-server communication. There is a natural trade-off between the communication cost and key size. For any constant c $(1 \leq c \leq k)$, the communication cost can be reduced by a factor of c by increasing the key size with the same factor.

In Table II, we show a comparative study on some general attributes e.g., storage consumption, major computations, authentication party, achieved security, approximate hardware cost etc., between our protocol and several HB-like and non-HB protocols. It appears that although the tag's hardware cost of the proposed protocol is optimal, it achieves most common security requirements and uniquely *full mutual authentication* properties from exact LPN assumption.

VII. Conclusion

This paper presents a novel hardware-friendly RFID authentication protocol based on a commitment scheme from the exact LPN problem that can meet the hardware constraints of the EPC Class-1 generation-2 tags. In comparison to other protocols as described in Table 2, it requires less hardware and has achieved major security attributes. The protocol is also compliant to *ZK-private* privacy settings. Moreover, this is the first protocol that allows mutual authentication for the whole system i.e., tag, reader and server from the LPN problem. Furthermore, security and privacy proofs are given in the standard model that uses indistinguishability as basic privacy notion. Note that the proposed protocol can be easily utilized for other popular security protocols of RFID application s,t., ownership transfer, Supply chain management etc.

References

[1] E. Welbourne, L. Battle, G. Cole . Building the Internet of Things Using RFID: The RFID Ecosystem Experience. Internet Computing, IEEE , vol.13, no.3, pp.48-55, 2009.

[2] Ari Juels and Stephen A. Weis. Authenticating pervasive devices with human protocols. In Victor Shoup, editor, CRYPTO 2005, volume 3621 of LNCS, pages 293-308. Springer, August 2005.

[3] Ahamed, Sheikh Iqbal, Farzana Rahman, and Endadul Hoque. ERAP: ECC based RFID authentication protocol. Future Trends of Distributed Computing Systems, 2008. FTDCS'08. 12th IEEE International Workshop on. IEEE, 2008.

[4] Batina, Lejla, et al. An elliptic curve processor suitable for RFID tags. International Association for Cryptologic Research ePrint Archive, 2006.

[5] N. W. Lo, K. Yeh, and C. Y. Yeun. New mutual agreement protocol to secure mobile RFID-enabled devices. Information Security Technical Report 13.3, pp. 151-157, 2008.

[6] N. J. Hopper and M. Blum. Secure human identification protocols. Advances in Cryptology - ASIACRYPT 2001, Lecture Notes in Computer Science, Vol. 2248, Springer, pp. 52-66, 2001.

[7] Golub, Gene, W. Kahan. Calculating the singular values and pseudo-inverse of a matrix. Journal of the Society for Industrial & Applied Mathematics, Series B: Numerical Analysis 2.2: pp. 205-224, 1965.

[8] Jonathan Katz, Ji Sun Shin, and Adam Smith. Parallel and concurrent security of the HB and HB+ protocols. Journal of Cryptology, 23(3):402-421, 2010.

[9] Henri Gilbert, Matt Robshaw, and Herve Sibert. An active attack against HB+ - a provably secure lightweight authentication protocol. Cryptology ePrint Archive, Report 2005/237, 2005.

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

[11] Jorge Munilla and Alberto Peinado. HB-MP: A further step in the HB-family of lightweight authentication protocols. Computer Networks, 51(9):2262-2267, 2007.

[12] Henri Gilbert, Matthew J. B. Robshaw, and Yannick Seurin. Good variants of HB+ are hard to find. In Gene Tsudik, editor, FC 2008, volume 5143 of LNCS, pp. 156-170. Springer, 2008.

[13] Henri Gilbert, Matthew J. B. Robshaw, and Yannick Seurin. HB++: Increasing the security and efficiency of HB+. In Nigel P. Smart, editor, EUROCRYPT 2008, volume 4965 of LNCS, pp. 361-378. Springer, 2008.

[14] Khaled Ouafi, Raphael Overbeck, and Serge Vaudenay. On the security of HB# against a man-in-the-middle attack. In Josef Pieprzyk, editor, ASIACRYPT 2008, volume 5350 of LNCS, pages 108-124. Springer, 2008.

[15] Moore, E. H. On the reciprocal of the general algebraic matrix. Bulletin of the American Mathematical Society 26 (9), 394-395, 1920.

[16] Thuc, D.N., Hue, T.B.P., Van, H.D. An Efficient Pseudo Inverse Matrix-Based Solution for Secure Auditing. IEEE-RIVF, pp. 712 2010.

[17] Jens Hermans, Andreas Pashalidis, Frederik Vercauteren, Bart Preneel. A New RFID Privacy Model, ESORICS 2011.

[18] Cao, X , ONeill, M. (2011). F-HB: An Efficient Forward Private Protocol. Workshop on Lightweight Security and Privacy: Devices, Protocols and Applications(Lightsec), 2011.

[19] T. V. Le, M. Burmester, and B. de Medeiros. Universally Composable and Forward-secure RFID Authentication and Authenticated Key Exchange. ACM Symposium on InformAtion, Computer and Communications Security (ASIACCS), 2007.

[20] O. Billet, J. Etrog and H. Gilbert. Lightweight Privacy Preserving Authentication for RFID Using a Stream Cipher. International Workshop on Fast Software Encryption (FSE), 2010.

[21] Xuefei Leng, Keith Mayes, Konstantinos Markantonakis. HB-MP+ Protocol: An Improvement on the HB-MP Protocol. IEEE International Conference on RFID. pp. 118-124, 2008.

[22] G. Tsudik, Ya-trap: Yet another trivial RFID authentication protocol, in PerCom Workshops, pp. 640-643, 2006.

[23] L. He, S. Jin, T. Zhang, and N. Li, An enhanced 2-pass optimistic anonymous rfid authentication protocol with forward security, in WiCOM, pp. 1-4, 2009.

[24] B. Applebaum, Y. Ishai, E. Kushilevitz. Cryptography with Constant Input Locality. Journal of Cryptology 22(4), 429469, 2009.

[25] MSI Mamun, A. Miyaji, M. Rahman. A Secure and Private RFID Authentication Protocol under SLPN Problem. NSS2012, LNCS 7645, pp. 476-489, 2012.

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

[27] C. Yu Ng, W. Susilo, Y. Mu, and R. Safavi-Naini. Practical RFID Ownership Transfer Scheme. Journal of Computer Security - Special Issue on RFID System Security, 2010.

[28] A. Jain, S. Krenn, K. Pietrzak, A. Tentes. Commitments and Efficient Zero-Knowledge Proofs from Learning Parity with Noise. ASIACRYPT 2012, LNCS,Volume 7658, pp 663-680, 2012.

[29] S. Hill, Cryptography in an Algebraic Alphabet, The American Mathematical Monthly Vol.36, pp. 306-312, 1929.

[30] S. Saeednia, How to Make the Hill Cipher Secure, Cryptologia, Vol.24, No.4, pp. 353-360, 2000.

[31] E. Murray. Hill Ciphers and Modular Linear Algebra. Mimeographed notes,University of Massachusetts, 1998. (accessed: 26 Nov, 2012) (www.apprendre-en-ligne.net/crypto/hill/Hillciph.pdf)

[32] M. Toorani, A. Falahati. A Secure Variant of the Hill Cipher. Proceedings of the 14th IEEE Symposium on Computers and Communications (ISCC'09), pp. 313316, 2009.

[33] R. H. Deng, Y. Li, M. Yung, and Y. Zhao. A new framework for RFID Privacy. in Proceedings of the 15th European Symposium on Research in Computer Security (ESORICS 10), vol. 6345 of LNCS, pp. 118, Springer, 2010.

[34] Rizomiliotis, Panagiotis, and Stefanos Gritzalis. GHB#: a provably secure HB-like lightweight authentication protocol. Applied Cryptography and Network Security. Springer,DEAKIN 2012.