A secure and private RFID authentication protocol under SLPN problem

Mohammad S. I. Mamun, Atsuko Miyaji, and Mohammad S. Rahman

Japan Advanced Institute of Science and Technology (JAIST)
Ishikawa, Japan.
{mamun, miyaji, mohammad}@jaist.ac.jp

Abstract. Authentication is one of the prominent features of RFID system. As wireless link between the tag and the reader in an RFID system is vulnerable against active adversary, ample research has been done in this area. In this paper[1], we present a novel, efficient and privacy preserving mutual authentication protocol of HB-family to meet the demand of low-cost tags. It is composed of Subspace Learning Parity from Noise problem (SLPN) and pseudo-inverse matrix properties; both of them significantly minimize the cost in terms of computation and hardware requirements. We compare our protocol with other existing HB and non-HB authentication protocols according to their construction and achievements of security and privacy attributes.

1 Introduction

RFID system simplifies data acquisition and management with an automated identification of an object to which the tag is attached. A number of authentication schemes have been proposed targeting privacy, security, efficiency and performance issues. Majority of the authentication protocols in this area use symmetric key ciphers, as asymmetric key ciphers are too expensive for a compact hardware like RFID tag. For example, RSA require more than 30,000 gates, which is too expensive for low-cost tag where maximum 2,000 gates out of 10,000 gates are available for security purpose [1].

LPN problem is a light-weight provably-secure cryptographic scheme which was first introduced in 2001 by Hopper & Blum [3]. LPN based authentication is not only *theoretically secure* in terms of provable security, but also provides better *efficiency* than classical symmetric ciphers that are not related to hard problems. There has been a large body of research on HB protocol that outputs protocols such as HB^+, HB^{++}, $HB^\#$, HB-MP, $HB\text{-}MP^+$, HB^* etc.[5–9, 11, 22]. Unfortunately all of them were later shown insecure or susceptible to particular attacks [10, 11]. In [2], authors propose an authentication protocol based on Subspace LPN (SLPN) problem with tight security reduction which is as efficient as the previous HB-family but has twice the key length; in addition their proof

[1] Research supported by Graduate Research Program (GRP), JAIST foundation grants.

works in a quantum setting which leads the protocol secure against quantum adversaries.

To the best of our knowledge, the latest addition to HB-family for RFID authentication is F-HB, where authors use 2 LPN problem as their basic computation [17]. We carefully observe that the Toeplitz matrix multiplication (EX-OR operation) for multiple bit LPN problem and MAC generation in the main protocol of [17] are not consistent with matrix size, although authors did not clarify specific matrix size in operation; and threshold value for LPN problem is not specified concretely. Moreover, in the last protocol transcripts, where a tag's secret key is updated, *if-checking* is not consistent and is not based on LPN problem; but an EX-OR vector computation. Unlike [17], our protocol follows SLPN based problem for tag authentication where secret key is not a vector but a binary matrix. In addition, we introduce pseudo-inverse matrix for updating the secret key of the tag and apply SLPN problem for both the tag and the reader authentication. As a consequence, our proposed protocol is more robust against quantum adversaries while efficient like previous HB-protocol family.

The rest of this paper is organized as follows. Section 2 introduces notations and assumptions used in this paper and other useful definitions related to basic primitives and security notions. The proposed protocol is described in Section 3. In Section 4, all achieved security and privacy attributes are discussed in detail with their proof; while Section 5 covers the analysis and comparison results. Finally, Section 6 concludes this paper.

2 Preliminaries

In this section, we first briefly introduce the notations used in the paper in Table 1. Then we discuss some inevitable assumptions followed by useful definitions for primitives and security notions.

2.1 Assumption

RFID system described in this paper consists of a single legitimate reader and a set of tags (EPC global Class 1 generation 2). Reader is connected to the back-end server that stores all the relevant data including tag database. Each tag has its unique identification T_{id} and session key S_i. T_{id} is used as the shared secret key between the tag and the reader.

The authentication protocol is an interactive protocol executed between *tags/prover* and a *reader/verifier* where both of them are *probabilistic polynomial time* (PPT) algorithm. All communications between the server and the reader are assumed to be secure and over authentic channel. For simplicity, we consider reader and server as identical. Throughout the paper, we use the term reader and server interchangeably. A tag is not tamper-resistant device; so its session key S_i is refreshed after each session completes successfully. For updating key, tag authenticates the reader first. Adversary cannot compromise reader/server and it cannot corrupt the tag until it compromises both T_{id} and S_i at a time. However,

Table 1. Notations used in the paper

λ	security parameter
$\mathbb{Z}_p$	set of integers modulo an integer $p \geq 1$
$l \in \mathbb{N}$	length of the secret key
$n \in \mathbb{N}$	number of parallel repetitions $n \leq l/2$
T_{id}	$2l$ bit EPC or unique ID of a tag
I_i	nIndex of the tag during time period i
P_i	$l \times l$ bit matrices as session key for the reader during time period i
S_i	$l \times n$ bit matrices as session key between the reader and the tag during time period i
s	$2l$ bit vector random binary number generated by reader.
s'	$2l$ bit vector random binary number generated by tag
$w(s)$	Hamming weight of the vector s
τ	Parameter of the Bernoullli error distribution $\mathbf{Ber}_\tau$ where $\tau \in]0, 1/2[$
τ'	Authentication verifier acceptance threshold (Tag/Reader) where $\tau' = 1/4 + \tau/2$
e	n bit vector from Bernoullli distribution $\mathbf{Ber}_\tau$ with parameter τ ; $Pr[e = 1] = \tau$
$[Q]$	$l \times n$ bit randomly generated non-singular binary matrices by reader
$[S]^T$	transpose of matrices $[S]$ i.e., $T : \mathbb{Z}_2^{n \times l} \rightarrow \mathbb{Z}_2^{l \times n}$
$[P]^+$	pseudo-inverse of a matrices $[P]$
$(x_{\downarrow} y)$	the vector derived from x by deleting all the bits $x[i]$ where $y[i] = 0$
$\oplus, \|$	bitwise XOR operation and concatenation of two vectors respectively

if both of the secret keys are exposed at a time, the adversary can trace the tag for certain period i until the next authentication cycle starts. We assume tag binary identification T_{id} is unique within an RFID system. To avoid exhaustive database search at the reader hash-index (I) is used. Database at the server associates the tag index with other tag related data e.g., T_{id}, S_i, P_i etc.

2.2 Definitions for primitives

Definition 1. *The **LPN problem** is to distinguish from random binary vectors*[2] *with a random "secret" vector. Let* $[R] \in_R \mathbb{Z}_2^{l \times n}$, $s \in_R \mathbb{Z}_2^n$, τ *be noise parameters, and* $e \in \mathbb{Z}_2$ *selected from* Ber_τ *s.t.* $w(e) \leq \tau l$. *Given* $r = ([R]^T \cdot s) \oplus e \in \mathbb{Z}_2^l$, *finding* $s' \in_R \mathbb{Z}_2^n$ *such that* $w([R]^T \cdot s') \oplus r) \leq \tau l$ *is denoted by the distribution* $\mathcal{D}_{\tau,l}(s')$. *The* $LPN_{\tau,l}$ *problem is to distinguish an oracle returning samples from* $\mathcal{D}_{\tau,l}(s')$, *or an oracle returning uniform samples* U_l.

Definition 2. *The **Subspace LPN (SLPN)** problem is defined as a biased half-space distribution where adversary can ask not only with secret "s" but also with* $r'.s \oplus e'$; *where* $\mathbf{e}', \mathbf{r}'$ *can be adaptively chosen with sufficient* $rank(r')$. *Let* $s \in \mathbb{Z}_2^l$ *and* $l, n \in \mathbb{Z}$ *where* $n \leq l$. *Decisional Subspace LPN problem (SLPN) is* (t, Q, ϵ)*-hard such that*

[2] result of noisy inner products of vectors

$$Adv_{\mathcal{A}}^{\mathbf{SLPN}}(\tau, l, n) = Pr[LPN_{\tau,l,n}(s, \cdot, \cdot) = 1] - Pr[U_l : LPN_{1/2}(\cdot, \cdot) = 1] \leq \epsilon$$

Definition 3. *The **Subset LPN problem** (**SLPN***) is defined as a weaker version of $SLPN$ problem where the adversary cannot ask for all inner products with $r' \cdot s \oplus e'$; for any $rank(r') \geq n$ but only with **subset** of s. Let $(l, n, v) \in \mathbb{Z}$ where $n \leq l$ and $w(v) \geq n$ where v can be adaptively chosen. Hence, $LPN^*_{\tau,l,n}(s, v)$ samples are of the form $([R]^T{}_{\downarrow} v \cdot s_{\downarrow} v) \oplus e$ and $LPN_{1/2}(v)$ takes v as input and output a sample of U_l. $SLPN^*$ problem is (t, Q, ϵ)-hard such that*

$$\mathbf{Adv}_{\mathbf{A}}^{\mathbf{SLPN}^*}(\tau, l, n) = Pr[LPN^*_{\tau,l,n}(s, \cdot) = 1] - Pr[U_l : LPN_{1/2}(\cdot) = 1] \leq \epsilon$$

Definition 4. *In linear algebra, a **pseudo-inverse** A^+ of a matrix A is a generalization of the inverse matrix. The most widely known and popular pseudo-inverse is the **Moore-Penrose** pseudo-inverse, which was independently described by E. H. Moore [12]. An algorithm for generating pseudo-random matrix on non-singular matrix $\mathbb{Z}_2$ is given in [13]. However, the matrix A is the unique matrix that satisfies the following properties: $AA^+A = A$, $A^+AA^+ = A^+$, $(A^+A)^T = A^+A$, $(A^+)^+ = A$, $(A^T)^+ = (A^+)^T$, $(A^+)^T = (A^T)^+$, $(AA^+)^T = AA^+$ where $T : \mathbb{Z}_2^{n \times l} \to \mathbb{Z}_2^{l \times n}$, $A^+ = (A^TA)^{-1}A^T$, and $A^+ = A^T(AA^T)^{-1}$ where $row(A)$ and $col(A)$ are linearly independent.*

2.3 Definitions for security notions

Definition 5. *A protocol is secure against **passive attacks**, if there exists no PPT adversary $\mathcal{A}$ that can forge the verifying entity with non-negligible probability by observing any number of interactions between the tag and reader.*

Definition 6. *A protocol (t, Q, ϵ) is called secure against **active attacks**, if there exists no PPT adversary $\mathcal{A}$, usually called $(\mathbf{Q}, \mathbf{t})$ **adversary**, running in time t and making Q queries to the honest prover, has probability more than ϵ. Adversary $\mathcal{A}$ runs in two stages: First, it observes and interrupts all the interactions between the target tag T and legitimate reader with concurrent executions according to the defined security. Then it is allowed only one time to convince the reader. Note that, this time $\mathcal{A}$ is not allowed to continue his attacks in time instance t; but can utilize several discrete or successive time period.*

Definition 7. *In **man-in-the-middle** (MIM) attack, adversary $\mathcal{A}$ is allowed to maintain connections with both the tag and the reader, making the tag believe that they are talking directly to the reader over a secure connection, when in fact the entire communication is controlled by $\mathcal{A}$. Then $\mathcal{A}$ interacts with the reader to authenticate. The goal of the attacker $\mathcal{A}$ is to authenticate successfully in Q rounds. $\mathcal{A}$ is successful iff it gets accept response from all Q rounds.*

Definition 8. ***Forward security** property means that even if the adversary obtains the current secret key, it cannot derive the keys used for past time periods.*

Definition 9. ***Backward security*** *is the opposite to the forward security. If the adversary can explore the secret of the tag at time i, it cannot be traced in future using the same secret. In other words, exposure of a tag's secret should not reveal any secret information regarding the future of the tag. But if an adversary is allowed to obtain full access to the tag's secret and thus can trace the target tag at least during the current session of authentication immediately following the attack, its not making any sense to perfect security in practice. Therefore, it is impossible to provide backward security for RFID-like device practically.*

Definition 10. ***Tracking a tag*** *refers the attacker could guess the tag identity or link multiple authentication sessions of the same tag. In our protocol, adversary cannot recover S_i or any other information identifying that particular tag.*

Definition 11. *In* ***De-synchronization attack****, adversary aims to disrupt the key update leaving the tag and the reader in a desynchronized state and renders future authentication impossible.*

Definition 12. ***Denial of Service*** *(DoS) is an attempt to make a tag unavailable to its intended users. DoS resistance capability of the protocol is infinite as tag updates the key after reader authentication is successful.*

Definition 13. ***Tag cloning*** *entails that the data on a valid tag is scanned and copied by a malicious RFID reader and later the copied data will be embedded onto a fake tag.*

Definition 14. *In* ***replay attack****, an adversary reuses the communication scripts from the former sessions to perform a successful authentication between each tag and their reader.*

Definition 15. *An RFID system is said to be (Q, t, ϵ)* ***strong private****, if there exist no (Q, t) adversary $\mathcal{A}$ who can break its strong privacy with advantage $Adv^b_{\mathcal{A}}(k) \geq \epsilon$.*

3 Construction

We adopt the idea of key-insulation to slightly twist our 3-round mutual authentication protocol described in Fig. 1. Protocol allows significantly less computations to a tag. On the other hand, the most expensive computations of the protocol are handled by the reader. We use only random generation, bitwise XOR and matrix multiplication as tag operation. Protocol uses $(\lambda, \tau, \tau', n, l)$ as public parameters, where (τ, τ') are constant while (l, n) depends on the security parameter λ. For initialization, server generates initial index I_0, session key S_0 and its corresponding P_0 and other public parameters; and set necessary data into tag non-volatile memory. Note that, we use *matrix* as a secret, not a *vector*. Therefore, for each tag, there is a tuple $[I_i, T_{id}, S_{i-1}, S_i, P_{i-1}, P_i]$ to be stored in the back-end database of the server at any time instance i.

For *tag* authentication, let a tag have S_i and I_i, which have been derived from the previous $(i-1)$ successful authentication sessions.

Reader $(I_i, T_{id} \in \mathbb{Z}_2^{2l}, \mathbf{S_i} \in \mathbb{Z}_2^{l\times n}, \mathbf{P_i} \in \mathbb{Z}_2^{l\times l})$	**Tag**$(I_i, T_{id} \in \mathbb{Z}_2^{2l}, \mathbf{S_i} \in \mathbb{Z}_2^{l\times n})$
$s \in_R \mathbb{Z}_2^{2l}$; where $\mathbf{w}(s) = l$	
$\xrightarrow{\mathbf{s}}$	
	if $\mathbf{w}(s) \neq l$ **return**;
	$\mathbf{e} \in_R \mathbf{Ber}_\tau^n$;
	$\mathbf{r} := [\mathbf{S_i}]^{\mathbf{T}}.\ (T_{id\downarrow} s) \oplus \mathbf{e}$
	$s' \in_R \mathbb{Z}_2^{2l}$; where $\mathbf{w}(s') = l$
	$I_{i+1} = r$
	$\xleftarrow{(\mathbf{I_i}, \mathbf{s'}, \mathbf{r})}$
Lookup T_{id} by using hash-table index:	
Direct match: $I = I_i$; if (not found) then	
Brute-force search: find an entry $[I_i, T_{id}, \mathbf{S_{i-1}}, \mathbf{S_i}, \mathbf{P_{i-1}}, \mathbf{P_i}]$	
s.t., $\exists$ ($\mathbf{S_i}$ or, $\mathbf{S_{i-1}}$), for which the following satisfies:	
If $\mathbf{w}([\mathbf{S_i}]^{\mathbf{T}}.(T_{id\downarrow} s) \oplus \mathbf{r}) > n.\tau'$ **return**;	
Else	
$I_{i+1} = r$	
if $\mathbf{w}(s') \neq l$ **return**;	
Generate non-singular $\mathbf{[Q]} \in_R \mathbb{Z}_2^{l\times l}$	
$[\mathbf{S_{i+1}}] = [\mathbf{Q}].[\mathbf{S_i}] \in \mathbb{Z}_2^{l\times n}$	
where $rank(\mathbf{S_{i+1}}) = n$	
Compute $\mathbf{P_{i+1}} = [\mathbf{S_{i+1}}][\mathbf{S_{i+1}}]^+ \in \mathbb{Z}_2^{l\times l}$	
where $[\mathbf{S_{i+1}}]^+ = ([\mathbf{S_{i+1}}]^{\mathbf{T}}[\mathbf{S_{i+1}}])^{-\mathbf{1}}[\mathbf{S_{i+1}}]^{\mathbf{T}} \in \mathbb{Z}_2^{n\times l}$	
$\mathbf{P_i}' = [\mathbf{P_i}][\mathbf{Q}] \in \mathbb{Z}_2^{l\times l}$;	
$\mathbf{e}' \in_R \mathbf{Ber}_\tau^n$;	
$\mathbf{r}' := [\mathbf{S_i}]^{\mathbf{T}}.\ (T_{id\downarrow} s') \oplus \mathbf{e}'$	
$\xrightarrow{(\mathbf{P_i}', \mathbf{r}')}$	
	if $\mathbf{w}([\mathbf{S_i}]^{\mathbf{T}}.\ (T_{id\downarrow} s') \oplus \mathbf{r}') > n.\tau'$ **return**; else **accept**
	$\mathbf{S_{i+1}} = (\mathbf{P_i}'.\mathbf{S_i}) \in \mathbb{Z}_2^{l\times n}$
	if $rank([\mathbf{S_{i+1}}]) \neq n$ **return**;

Fig. 1. RFID Authentication Protocol

- Reader: Generate a random binary challenge string s, and sends it to a tag.
- Tag: Check the *hamming weight* of the string s and generate an n-bit noise vector $\mathbf{e}$, a random $2l$-bit challenge string s' for a reader with hamming weight l. Next an n-bit LPN problem is computed as $\mathbf{r} := [\mathbf{S_i}]^{\mathbf{T}} \cdot (T_{id\downarrow} s) \oplus \mathbf{e}$. To eliminate brute-force searching at the server end, maintain an index I_i

and send it to the reader. Finally, update index I_{i+1} to r and send $(I_i, s', \mathbf{r})$ to the server.

– Reader: First search database to find a tuple $[I_i, T_{id}, S_{i-1}, S_i, P_{i-1}, P_i]$ with index I sent by the server. But searching might fail sometimes e.g., due to synchronization attack etc. If it fails, then apply brute-force method targeting to explore $\mathbf{S_i}$ or $\mathbf{S_{i-1}}$ such that it satisfies LPN problem: $\mathbf{w}([\mathbf{S_i}]^{\mathbf{T}} \cdot (T_{id\downarrow}s) \oplus \mathbf{r}) \leq n \cdot \tau']$, or $[\mathbf{w}([\mathbf{S_{i-1}}]^{\mathbf{T}} \cdot (T_{id\downarrow}s) \oplus \mathbf{r}) \leq n \cdot \tau']$. If the brute-force method passes, it accepts the tag, update the index to I_{i+1} and enter *reader authentication* phase.

For *reader authentication*, it has secret S_i, P_i and other public parameters which has been derived from previous $(i-1)$ successful authentication sessions.

– Reader: First test whether *hamming weight* of s' is exactly l. Then generate a non singular binary matrix Q to update session key S_{i+1} as $[Q \cdot S_i]$ and compute pseudo inverse-matrix S_{i+1}^{+}, and P_{i+1} as $[S_{i+1} \cdot S_{i+1}^{+}]$. To send the new session key S_{i+1} to the tag and blinding the matrix Q, P_i' is computed by $[P_i \cdot Q]$ which is actually equivalent to a binary matrix $[S_i S_i^{+} Q]$. Assume the adversary cannot reveal S_i from P_i' in polynomial time. Next, for reader authentication, generate an n-bit noise vector e' and compute multiple bit LPN problem as $\mathbf{r'} := [\mathbf{S_i}]^{\mathbf{T}} \cdot (T_{id\downarrow}s') \oplus \mathbf{e'}$. Finally answer the tag with string $(\mathbf{P_i}', \mathbf{r'})$.
– Tag: Check the *hamming weight* of $([\mathbf{S_i}]^{\mathbf{T}} \cdot (T_{id\downarrow}s') \oplus \mathbf{r'}) \leq n \cdot \tau'$ where $(n \cdot \tau')$ is the predefined accepted threshold value for the LPN problem. If this check passes, accept the reader and update session key $\mathbf{S_{i+1}}$ by $[(\mathbf{P_i}' \cdot \mathbf{S_i}) = (\mathbf{S_i S_i^{+} Q} \cdot \mathbf{S_i}) = (\mathbf{S_i Q})]$ where $[\mathbf{S_i S_i^{+} S_i} = \mathbf{S_i}]$[3]. However, if the check fails, tag's session key remains unchanged.

4 Security Analysis

4.1 SLPN problem

We use the proof method similar to that described in [2] as Theorem 1. follows. Even though protocol in our model and that in [2] are different, a similar proof can be used as both are based on the $SLPN^*$ problem. Hardness of $SLPN^*$ can be defined using an indistinguishability game. More formally, security of the proof is based on the computational indistinguishability of the two oracles $SLPN^*$ and uniform distribution U_{2l}. From the protocol description it can be found that noise is a vector rather than a single bit; and the secret is not a vector but a pseudo-random matrix.

Theorem 1. *For any constant $\gamma > 0$, let $n = l/2 + \gamma$. If the $SLPN^*$ problem is (t, Q, ϵ)-hard then the authentication protocol from Figure 1. is (t', Q, ϵ')-secure against active adversaries, where the constants $(c_\gamma, c_\tau > 0)$ depend only on γ and τ respectively.*

[3] From the properties of pseudo-inverse matrix.

$$t' = t - poly(Q, l) \quad \epsilon' = \epsilon + Q.2^{-c_\gamma.l} + 2^{-c_\tau.n} = \epsilon + 2^{-\theta(n)}$$

The protocol has completeness error $2^{-c_\tau.n}$

4.2 Man-in-the Middle Attack

The most sophisticated and realistic attack in RFID system is Man-in-the Middle (MIM) attack. Our protocol is MIM-secure against active attack from SLPN assumption. Note that, first the reader authenticates the tag and then vice versa. In case of tag authentication, it runs a two-round MIM-secure authentication protocol where *reader* chooses a random variable as challenge, and *tag* returns the response according to the challenge. Authentication tag $\gamma = (S, r : S^T f_k(s) \oplus e)$ where $f_k(s)$ is the secret key derivation function which uniquely encodes challenge s according to k by selecting l bits from the key[4] k. The main technical difficulty to build a secure MIM-free authentication from LPN is to make sure the secret key k does not leak from verification queries. In [2], they use randomize-mapping function $f_k(s) = (k \downarrow s : \mathbb{Z}_2^{2l} \to \mathbb{Z}_2^l)$ for some random s and proved that if LPN is hard then the construction is MIM-secure. We have twisted a little the original idea. In our construction, we remain both S and k secret which enhances security. We use EX-OR operation for hiding s' using T_{id} as key. Note that, XOR cipher is vulnerable to frequency analysis; therefore even if adversary compromises T_{id}, it cannot generate S_i for any subsequent sessions using only T_{id}. In the third phase of the protocol, we introduce pseudo-random matrix as blinding factor to transfer new session key S_{i+1} which is secure from pseudo-random matrix property assumption.

4.3 Pseudo-random matrix

We followed security analysis in [13] where they claim that, having known the messages $XX^+Q \in \mathbb{Z}_2^{l \times l}$, it is impossible to recover the secrets $X \in \mathbb{Z}_2^{l \times n}$, or $Q \in \mathbb{Z}_2^{l \times l}$. Given $XX^+Q \in \mathbb{Z}_2^{l \times l}$, suppose that $rank(X) = r$, and

$$X^+X = \begin{pmatrix} I^{r \times r} & 0 \\ 0 & 0 \end{pmatrix} \Rightarrow X^+XQ = \begin{pmatrix} Q^{r \times r} & 0 \\ 0 & 0 \end{pmatrix}$$

where $I^{r \times r}$ is an *Identity matrix* and $Q^{r \times r}$ is the left upper sub-matrix of Q. Then the probability of an adversary to determine the correct Q is $2^{-(l-r)n}$. To ensure security we need to ensure $l \gg r$ which can be obtained by $l > n$. In our authentication protocol, we let $n \leq l/2$ to ensure large value of l.

4.4 Forward Security

For each operation, tag uses session key S_i and reader also use its corresponding P_i for verification of authentication tags. At the end of each valid session, (S_i, P_i) is updated with random matrix and previous key is deleted permanently in the tag. We say that even if S_i is exposed during authentication session i by the attacker, tag's privacy is fully guaranteed for $(i-1)$ period.

[4] We use T_{id} as the secret key k

4.5 Backward Security

Typical RFID tags and their reader communicate only for a short period of time because of power constraint of a tag. Thus, either we restrict the adversary in a way that it can obtain neither T_{id} nor S_i at any time instance i, or there should exist some non-empty gap between the time of a reveal query and the attack, while tag is not accessible by the adversary. This entails the adversary miss the protocol transcripts needed to update the compromised secret key and hence our protocol claims **reduced** backward security.

4.6 Tracking a tag

Protocol can resist tracking the tag due to the following reason: it refreshes the random vector (s, s', e, e'), updates the keys (P_i, S_i) while assumptions like SLPN problem, pseudo-random matrix makes the protocol indistinguishable from the adversarial perspective.

4.7 De-synchronization attack

We introduce indexing of the tag to get rid of the attack. When the reader and the tag maintain synchronization, searching hash table becomes very fast with *direct match* technique. However, synchronization attack may take place in the third protocol transcript from the reader to the tag; while the tag may not receive (p', r') to update its shared key. In the later case, brute-force search will be used for successful authentication. Though it incurs worse performance, but after successful authentication synchronization would be recovered.

4.8 Tag cloning

We use two different keys T_{id} and S_i for the tag. Therefore, even if the tag is cloned by malicious reader, we assume either of the keys is not compromised. For instance, an EPC generation 2 allows a password-enabled *secure state* configuration that prevents anyone from reading or writing onto a tag memory bank. Let T_{id} be stored in a password protected memory bank. Moreover, tag is not allowed to update it's key S_i until it authenticates the reader. This verification thwarts the cloning attack as well.

4.9 Replay Attack

Assuming that the random challenges sent by the reader and the tag are same in two different sessions, an adversary can launch *replay attack* by snooping the random numbers; but in our protocol, the reader queries the tag each time with a new random challenge s and then tag queries reader with random s', I_i. So, it is very unlikely to find a match between a pair of (I_i, t, r) from two different sessions of the same tag.

4.10 Privacy

First, we analyze our protocol using the *privacy model* in [15].

Theorem 2. *If the encryption in the protocol described in Fig. 1. is indistinguishable then the protocol is strong private for narrow adversaries.*

Proof: Given an adversary $\mathcal{A}$ that wins the privacy game with non-negligible advantage, we consider another adversary $\mathcal{B}$ that can break the *indistinguishaibility* game with non-negligible advantage described in *section* 4.1. The adversary $\mathcal{B}$ runs the adversary $\mathcal{A}$ to answer queries with the following exceptions:

- S, T_{id} are two different keys of the indistinguishaibility game.
- *SendTag* $(vtag, s)_b$: By retrieving the tag T_i and T_j references from the table D using virtual tag $vtag$; it generates two references $m_0 = \mathbf{w}([\mathbf{S_i}]^{\mathbf{T}}.(T_{i\downarrow}s) \oplus \mathbf{r}) > n.\tau'$ and $m_1 = \mathbf{w}([\mathbf{S_j}]^{\mathbf{T}}.(T_{j\downarrow}s) \oplus \mathbf{r}) > n.\tau'$. The references m_0, m_1 are sent to the indistinguishability oracle of SLPN problem which returns whether *hamming weight* satisfies $w \leq n.\tau'$ under one of the references .
- $\mathcal{B}$ cannot query for *Result*() oracle.

At the end of the game, $\mathcal{B}$ outputs according to $\mathcal{A}$'s guess. Hence, $\mathcal{B}$ is perfectly simulated for $\mathcal{A}$. If $\mathcal{A}$ breaks privacy then $\mathcal{B}$ wins indistinguishibility game; but indistinguishibility with only one call to the oracle is equivalent to indistinguishibility with multiple calls to the oracle that proves the *narrow privacy* of the protocol. □

In [16], authors have categorized RFID authentication protocols into *four* types according to their constructions and distinguished *eight* privacy levels by their natures on accessing *Corrupt*() oracle in the strategies of the adversary and whether *Result*() oracle is used or not.

- *Nil*: No privacy protection at all.
- *Weak*: Adversary has access to all oracles except *Corrupt* (T_i).
- *Forward* : Adversary has access to *Corrupt*(T_i) but other oracles are not allowed as *Corrupt*(T_i) oracles are accessed.
- *Destructive* : No restriction on accessing other oracles after *Corrupt* (T_i), but T_i is not allowed to use again.
- *Strong* : It is the strongest defined privacy level with no restrictions.

Each of these levels has its *narrow* counterpart to restrict the access of *Result*() oracle.

Our protocol belongs to *Type 2a* for construction where the shared key S_i has been updated just after the reader is authenticated. We now redefine our protocol privacy according to the model described in [16].

Without reader authentication, any adversary can keep querying a tag with any compatible reader until it is desynchronized with legitimate reader. Therefore, the tag's secret can only be desynchronized by one update. As the reader has both the keys S_i and S_{i-1}, in case of tag failure to update its shared key S_i, the reader can still try to authenticate the victim using the previous key S_{i-1} in

the next protocol conversation. Thus, it provides *weak* privacy to the protocol construction. Let an adversary $\mathcal{A}$ try to send authentication transcripts to the tag by blocking a valid reader authentication message, or by intercepting the tag in an online attack. This causes the tag to be in DoS attack or in a deadlock condition as it cannot update the key without reader authentication.

Theorem 3. *Protocol described in Fig. 1. is weak non-narrow privacy preserved.*

Due to lack of space, we remove the proof of the above theorem. That will appear in the full version. However, this narrow-forward privacy level attack can be reduced if tag accepts any value to update the key, but it is not practical; as in that case reader authentication message can be easily forged.

Table 2. Tag Resources and Security Comparison with HB family and Others

Scheme	**Storage**	**Computation** (major)	**Authentication**	**Security achieved**	**Hardware** (gates)
[2]	1 S	1 LPN	tag	1,4*	$\approx$ 1600
HB [3]	1 S	1 LPN	tag		$\approx$ 1600
HB$^+$ [6]	2 S	2 LPN	tag	7	$\approx$ 1600
HB-MP[8]	2 S	1 LPN	tag	5,6,7,9	$\approx$ 1600
HB-MP$^+$ [22]	2 S	1 LPN,1 HASH	tag	1,5,6,7,9	$\approx$ 3500
F-HB [17]	1I , 1 S	1 PRNG,2 LPN	mutual	$1, 2, 4^*, 5, 6, 7, 9$	$\approx$ 3500
ours	**1 I, 1 S**	**1 LPN,1 PIM**	**mutual**	1,2,3*,4,5,6,7,8,9	$\approx$ **1600**
[21]	1 S	1 PRF,1 HASH	tag	2,4,6,8,9	$\approx$ 6000
[18]	1I ,1 S	1 PRF	mutual	2,4*,6,8,9	$\approx$ 6000
[19]	1 S	1 PRNG,1 UH	tag	2,4,9	$\approx$ 3500
[20]	1 S	1 SC	mutual	2,4*,8,9	$\approx$ 2000
[23]	1 S ,2 TS	1 HASH	tag	4^*	$\approx$ 1500
[24]	1 S , 1 TS, 1 RN	2 HASH	mutual	$4^*, 8, 9$	$\approx$ 1500
[25]	1 RN,1 C, 1 TS,1 S	3 HASH	mutual	$2, 4^*, 6, 8, 9$	$\approx$ 1500

where SC:= Stream Cipher; S:= Secret key; C:= Counter; I:= Index;
PRNG:= Psudo Random Number Generator; UH:= Universal Hash;
PIM:= Pseudo Inverse Matrix; LPN:= Learning parity from noise
TS:= Time Stamp; RN:= random number;
Security attributes: Man-in-the Middle attack(1), Forward Security (2), Backward Security (3), Reduced Backward Security (3*), High Privacy (4), Limited Privacy (4*) Tag tracking (5), De-synchronization (6), Tag Cloning (7), Replay attack (8), DoS (9).

5 Comparison and Performance analysis

In case of tag, protocol operations include *two* random binary vector generation, *one* SLPN problem, *one* EX-OR operation, *three* binary linear matrix multiplication. For computation, we only consider SLPN problem and assume rest of the operations (e.g., calculation hamming weight) be trivial in terms of computational complexity. The protocol is roughly as efficient as HB^+ protocol with just twice the key length. Since it is a reduction of LPN to SLPN problem, the protocol is secure against quantum adversaries assuming LPN is secure against such adversaries. There is a natural trade-off between communication cost and key size. For any constant c $(1 \leq c \leq n)$, communication cost can be reduced by a factor of "c" by increasing the key size with the same factor.

Major computations of the proposed authentication scheme on the tag include linear binary matrix multiplication and LPN problem. And in case of storage, only a secret key and an index for the key. As bitwise XOR, matrix multiplication, hamming weight $w()$ and $(a_{\downarrow}b)$ are all binary operation, they can be easily implemented using bit-by-bit serialization to save hardware gates. In e-STREAM project, PRNG operation needs only 1,294 gates using Grain-v1[4]. The cost of a LPN problem and storing index and secret key may not be greater than PRNG and should be less than CRC. However, LPN problem can be implemented using a linear feedback shift register (LFSR) (for Transpose matrix), a 1-bit multiplier plus 1-bit accumulator (for binary multiplication), XOR gates (for $\oplus$ operation) 1-bit counter(for hamming weight) and a 1-bit comparator (for $a_{\downarrow}b$ operation). Thus, to achieve λ-bit security level; the overall hardware cost of the proposed protocol for the above mentioned functions on a tag is no more than 1600 gates including the cost of non-volatile memory to store secret key, index value and protocol intermediate values; and the protocol is suitable for Class-1 Generation-2 EPC tags where PRNG and CRC are used as hardware.

We demonstrate 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 in Table 2. It appears that although tag's hardware cost of the proposed protocol is optimal, yet it achieves most common security requirements. Additionally, it achieves $O(1)$ time complexity during synchronized state that resists brute-force searching in each authentication session. Alternatively, hardware cost of the reader is very expensive for the purpose of complex computing[5] resulting in reduced computing in tag and hence hardware. Besides that, hash-indexed searching technique at the reader where all the data related to a certain tag are stored efficiently as *index* reduces exhaustive database search at the reader end. As a consequence, in an RFID system with *remote authentication*[6], reader can use this index in *batch mode* operation to aggregate responses from several tags together to reduce communication cost between the reader and

[5] Searching database, Generating pseudo-random matrix are the most complex part of the protocol

[6] Tag readers are portable and server access is costly

the server where each tag contains unique index within the reader's *field of view* at a specific time instance.

6 Conclusion

This paper presents a novel hardware-friendly RFID authentication protocol based on SLPN 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 *strong private for narrow adversaries* privacy settings. Moreover, scalability of the protocol can be realized in synchronized mode and desynchronized mode that ensures infinite DoS resistance. Security and privacy can be protected as long as we allow adversary not to cope with both tag ID and the secret key simultaneously. In addition, the security and privacy proof follows standard model that uses indistinguishability as basic privacy notion. Our future research will focus on how to reduce the communication cost between the reader and server, assuming the wireless link between them is insecure, to figure a realistic privacy-preserving RFID environment.

References

1. 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.
2. K. Pietrzak, E. Kiltz, D. Cash, A. Jain, D. Venturi., Efficient authentication from Hard learning problem, Eurocrypt 2011, LNCS 6632, pp 7-26,2011.
3. N. J. Hopper and M. Blum, Secure human identification protocols, Advances in Cryptology - ASIACRYPT 2001, Lecture Notes in Computer Science, Vol. 2248, Springer, 2001, pp. 52-66.
4. Cid, C. Robshaw, M. (2009). The eSTREAM Portfolio 2009 Annual Update. July 2009. Available from http://www.ecrypt.eu.org/stream/.
5. 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, July 2010
6. 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. http://eprint.iacr.org/.
7. Julien Bringer, H. Chabanne, and Emmanuelle Dottax. HB++: a lightweight authentication protocol secure against some attacks. In SecPerU, pages 28-33, 2006.
8. 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.
9. 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, pages 156-170. Springer, January 2008.
10. 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, pages 361-378. Springer, April 2008.

11. 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, December 2008.
12. Moore, E. H. (1920). On the reciprocal of the general algebraic matrix, Bulletin of the American Mathematical Society 26 (9): 394-395. doi:10.1090/S0002-9904-1920-03322-7
13. Thuc, D.N., Hue, T.B.P., Van, H.D.: An Efficient Pseudo Inverse Matrix-Based Solution for Secure Auditing. IEEE-RIVF, 7-12 (2010); ISBN: 978-1-4244-8072-2
14. Y. Dodis, J. Katz, S. Xu, M. Yung. Key-Insulated Public Key Cryptosystems. In Eurocrypt 2002, volume 2332 of LNCS, pages 65-82. Springer, 2002.
15. Jens Hermans, Andreas Pashalidis, Frederik Vercauteren, Bart Preneel. A New RFID Privacy Model ESORICS 2011.
16. Ching Yu Ng, Willy Susilo, Yi Mu, and Reihaneh Safavi-Naini. New Privacy Results on Synchronized RFID Authentication Protocols against Tag Tracing. In Michael Backes and Peng Ning, editors, ESORICS, volume 5789 of Lecture Notes in Computer Science, pages 321-336. Springer, 2009.
17. Cao, X , ONeill, M. (2011). F-HB: An Efficient Forward Private Protocol. Workshop on Lightweight Security and Privacy: Devices, Protocols and Applications (Lightsec2011), March 14-15, 2011, Istanbul, Turkey.
18. 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), March 2007.
19. C. Berbain, O. Billet, J. Etrog and H. Gilbert, An Efficient Forward Private RFID Protocol, ACM Conference on Computer and Communications Security (CCS), November 2009.
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), February 2010
21. G. Avoine and P. Oechslin, A Scalable and Provably Secure Hash-Based RFID Protocol, IEEE International Workshop on Pervasive Computing and Communication Security, March 2005.
22. Xuefei Leng, Keith Mayes, Konstantinos Markantonakis: HB-MP+ Protocol: An Improvement on the HB-MP Protocol. IEEE International Conference on RFID, 2008: 118-124, April 2008.
23. G. Tsudik, Ya-trap: Yet another trivial rfid authentication protocol, PerCom Workshops, 2006, pp. 640-643.
24. C. Chatmon, T. van Le, and M. Burmester, Secure anonymous rfid authentication protocols, Computer & Information Sciences, Florida AM University, Tech. Rep., 2006.
25. L. He, S. Jin, T. Zhang, and N. Li, An enhanced 2-pass optimistic anonymous rfid authentication protocol with forward security, in WiCOM, 2009, pp. 1-4.