A privacy-preserving efficient RFID authentication protocol from SLPN assumption

Mohammad Saiful Islam Mamun* and Atsuko Miyaji

Japan Advanced Institute of Science and Technology (JAIST),
1-1, Asahidai, Nomi, Ishikawa 923-1292, Japan
E-mail: mamun@jaist.ac.jp
E-mail: miyaji@jaist.ac.jp
*Corresponding author

Abstract: This paper presents an authentication protocol of RFID system where both the tag and the reader are authenticated mutually. Optimal performance requirement, considering storage and computation constraints of low-cost tags, keeping security and privacy policies intact are some major challenges in recent research in this area. We propose a secure and private mutual authentication protocol of HB-family to meet the demand of low-cost tags. It is composed of subspace learning parity from noise (SLPN) problem and pseudo-inverse matrix properties where both of them significantly reduce the cost in terms of computation and hardware requirements. In addition, we compare our protocol with other existing HB-like and ordinary RFID authentication protocols according to their construction primitives and security and privacy achievements.

Keywords: RFID; mutual authentication; privacy; subspace learning parity from noise; SLPN problem; pseudo-inverse matrix.

Reference to this paper should be made as follows: Mamun, M.S.I. and Miyaji, A. (xxxx) 'A privacy-preserving efficient RFID authentication protocol from SLPN assumption', *Int. J. Computational Science and Engineering*, Vol. X, No. Y, pp.000–000.

Biographical notes: Mohammad Saiful Islam Mamun is a Doctoral candidate at Information Security Lab, School of Information Science, JAIST, Japan. He received his BSc (Hons.) in Computer Science and Engineering from Dhaka University, Bangladesh and MS in Information and Communication System Security from Royal Institute of Technology (KTH), Sweden in 2005 and 2008, respectively. His primary research interests include applied cryptography and information system security.

Atsuko Miyaji received her BSc, MSc and Dr. Sci. in Mathematics from Osaka University, Osaka, Japan in 1988, 1990, and 1997, respectively. She joined Panasonic Co., Ltd. from 1990 to 1998 and engaged in research and development for secure communication. She has been a Professor at the Japan Advanced Institute of Science and Technology (JAIST) since 2007. Her research interests include the application of number theory into cryptography and information security. She is a member of the International Association for Cryptologic Research, the Information Processing Society of Japan, and the Mathematical Society of Japan.

1 Introduction

Tag authentication is an indispensable approach to prevent an RFID tag from impersonation. In particular, tag authentication is more significant since tags are much vulnerable to counterfeit than readers. However, mutual authentication protocols add an additional protection for the RFID system in the protocol construction to safeguard the query is, in fact, coming from a legitimate reader.

Unlikability or untraceability, sometimes referred to interchangeable with same meaning, conveys the property that an adversary cannot distinguish whether two events occurring in an RFID system are related to the same tag or not. In addition, anonymity is another indispensable security property that assure the inability to identify a tag within an RFID system. This definition can be framed in terms of unlinkability by saying that a tag is anonymous in any transactions between the reader provided that adversary cannot link the tag to a transaction. In order to provide aforementioned security properties, ample research has been done in this area targeting enhanced privacy, security and performance issues. Since asymmetric key ciphers are too expensive for a compact hardware such as low-cost RFID tag, majority of the authentication protocols use *symmetric key* as secret. 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 the purpose of security (Juels and Weis, 2005).

The LPN problem is a light-weight provably-secure cryptographic scheme which was first introduced in 2001 by Hopper and Blum (2001). 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-MP^+, HB^*, etc. (Katz et al., 2010; Gilbert et al., 2005, 2008a; Bringer et al., 2006; Munilla and Peinado, 2007; Ouafi et al., 2008; Leng et al., 2008). Unfortunately, all of them later shown to be insecure or susceptible to particular attacks (Gilbert et al., 2008b; Ouafi et al., 2008). In Pietrzak et al. (2011), authors propose an authentication protocol based on the subspace learning parity from noise (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 works in quantum setting, which leads the protocol to be secure against quantum adversaries.

To the best of our knowledge, the latest addition to the HB-family for RFID authentication is F-HB, where authors use two LPN problems as their basic computation (Cao and ONeill, 2011). We carefully observe that the Toeplitz matrix multiplication (EX-OR operation) for the multiple bit LPN problem and MAC generation in the main protocol of Cao and ONeill (2011) are not consistent with matrix size, although the authors did not clarify the specific matrix size in operation; and the 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 the LPN problem; but an EX-OR vector computation. Unlike Cao and ONeill (2011), our protocol follows the SLPN-based problem for tag authentication, where the 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 to the SLPN problem for both the tag and the reader authentication. As a consequence, our proposed protocol is more robust against quantum adversaries while been efficient like the previous HB-protocol family.

The rest of this paper is organised 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 corresponding 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

We assume the RFID system described in this paper consist of a single legitimate reader and a set of tags (EPC global Class-1 Generation-2). The reader is connected to the backend server that stores all the relevant data including the tag database. Initially, the reader generates and set T_{id} and *public parameters* depending on security parameter λ. 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 are *probabilistic polynomial time* (PPT) algorithms. All communications between the server and the reader are assumed to be secure and over an authentic channel. For simplicity, we consider the reader and server as identical. Throughout the paper, we use the term reader and server interchangeably. A tag is not a tamper-resistant device; so its session key S_i is refreshed after each session is completed successfully. For updating the key, the tag authenticates the reader first. An adversary cannot compromise the reader/server and cannot corrupt the tag until it compromises both T_{id} and S_i at the same time. However, if both of the secret keys are exposed at a time, the adversary can trace the tag for a certain period i until the next authentication cycle starts. We assume tag binary identification T_{id} is unique within an RFID system. To avoid an 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.

Table 1 Notations used in this paper

λ	Security parameter
$\mathbb{Z}_p$	Set of integers modulo an integer $p \geq 1$
$l \in N$	Length of the secret key
$n \in N$	Number of parallel repetitions $n \leq l/2$
T_{id}	$2l$ bit EPC or unique ID of a tag
I_i	n index 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 the reader.
s'	$2l$ bit vector random binary number generated by the tag
$w(s)$	Hamming weight of the vector s
τ	Parameter of the Bernoullli error distribution Ber_τ 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 Ber_τ with parameter τ; $\Pr[e = 1] = \tau$
$[Q]$	$l \times n$ bit non-singular binary matrices randomly generated by the 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

2.2 Definitions for primitives

Definition 1: A protocol is called (t, Q, ϵ)-hard if there exist a PPT adversary $\mathcal{A}$, usually called (Q, t)-adversary that makes Q-queries in running time t to the honest prover, has an advantage at most ϵ,

$$\Pr\left|[\mathcal{A} \text{ succeeds}] - 1/2\right| \le \epsilon$$

Definition 2: Let $[R] \in_R \mathbb{Z}_2^{l\times n}$, $s \in_R \mathbb{Z}_2^n$, τ be the noise parameters, and $e \in \mathbb{Z}_2$ be selected from Ber_τ s.t., $w(e) \le \tau l$. Given $r = ([R]^T \cdot s) \otimes e \in \mathbb{Z}_2^l$, we denote **LPN**($\tau$, l)(s) for the distribution $\mathbb{Z}_2^l \leftarrow \mathbb{Z}_2^l \times \mathbb{Z}_2$.

The decisional *LPN problem* is (t, Q, ϵ)-hard to distinguish uniform random binary vectors[1] U_l from **LPN**(τ, l)(s) with random secret $s \in \mathbb{Z}_2^n$,

$$Adv_{\mathcal{A}}^{\mathbf{LPN}}(\tau, l) = \Pr\left[s \in \mathbb{Z}_2^l : LPN_{\tau,l}(s) = 1\right] - \Pr\left[U_l : LPN_{1/2} = 1\right] \le \epsilon$$

Definition 3: The *SLPN* problem is defined as a biased half-space distribution where the 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 $n \le l$. The decisional SLPN problem is (t, Q, ϵ)-hard such that,

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

Definition 4: The *subset LPN problem* (*SLPN**) is defined as a weaker version to *SLPN* problem where the adversary cannot ask for all inner products with $r' \cdot s \oplus e'$; for any rank(r') $\ge n$ but only with *subset* of s. Let $(l, n, v) \in \mathbb{Z}$ where $n \le l$ and $w(v) \ge n$ where v can be adaptively chosen. Hence, $\mathrm{LPN}^*_{\tau,l,n}(s, v)$ samples are of the form $([R]^T{}_{\downarrow} v \cdot s_{\downarrow} v) \oplus e$ and $\mathrm{LPN}_{1/2}(v)$ takes v as input and output a sample of U_l. The SLPN* problem is (t, Q, ϵ)-hard such that,

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

Definition 5: In linear algebra, a *pseudo-inverse* A^+ of a matrix A is a generalisation of the inverse matrix. The most widely known and popular pseudo-inverse is the *MoorePenrose* pseudo-inverse, which was independently described by Moore (1920). An algorithm for generating pseudo-random matrix on non-singular matrix $\mathbb{Z}_2$ is given in Thuc et al. (2010). However, the matrix A is the unique matrix that satisfies the following properties:

- $AA^+A = A$
- $A^+AA^+ = A^+$
- $(A^+A)^T = A^+A$
- $(A^+)+ = A$
- $(A^T)^+ = (A^+)^T$
- $(AA^+)^T = AA^+$ where $T : \mathbb{Z}_2^{n\times l} \to \mathbb{Z}_2^{l\times n}$
- $A^+ = (A^TA)^{-1}A^T$, such that *col*(A) is linearly independent
- $A^+ = A^T(AA^T)^{-1}$, s.t. *row*(A) is linearly independent.

2.3 Definitions for security notions

Definition 6: 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 7: A (t, Q, ϵ)-hard protocol is called secure against active attacks where the 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 utilise several discrete or successive time period.

Definition 8: In the *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 if and only if it gets accept response from all Q rounds.

Definition 9: The *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 10: The *backward security* is 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, it does not make any sense to perfect security in practice. Therefore, it is impossible to provide backward security for an RFID-like device practically.

Definition 11: Tracking a tag refers the attacker could guess the tag identity or link multiple authentication sessions of the same tag. In our protocol, the adversary cannot recover S_i or any other information identifying that particular tag.

Definition 12: In *de-synchronisation attack*, the adversary aims to disrupt the key update, leaving the tag and the reader in a desynchronised state and renders future authentication impossible.

Definition 13: 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 14: 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 15: In the *replay attack*, an adversary reuses the communication scripts from the former sessions to perform a successful authentication between each tag and their reader.

Definition 16: An *RFID* system, is said to unconditionally provide privacy notion *X*, if and only if for all adversaries $\mathcal{A}$ of type *X*, it holds that $Adv_{\mathcal{A}}^{X}(\lambda) = 0$. In case of computational privacy, it is $Adv_{\mathcal{A}}^{X}(\lambda) \leq \epsilon$ for all PPT adversaries $\mathcal{A}$ (Hermans et al., 2011).

Definition 17: 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 $Adv_{\mathcal{A}}^{b}(k) \geq \epsilon$.

3 Construction

We adopt the idea of key-insulation to slightly twist our three-round mutual authentication protocol described in Figure 1. The 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. The 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 initialisation, the server generates the initial index I_0, the session key S_0 and its corresponding P_0 and other public parameters; and set the necessary data into a 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*.

Figure 1 RFID authentication protocol

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 **[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**;

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: 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 **e**, a random 2*l*-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 synchronisation 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 $[\mathbf{w}([\mathbf{S_{i-1}}]^{\mathbf{T}} \cdot (T_{id} \downarrow s) \oplus \mathbf{r}) \leq \mathrm{n} \cdot \tau']$. 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 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 $[\mathbf{S_i S_i^{+} S_i} = \mathbf{S_i}]$. [2] However, if the check fails, tag's session key remains unchanged.

Note that, in the protocol, session key generated by the reader is used by the tag. To be precise, session key S_{i+1} is generated from the former key S_i and random matrix $[Q]$. Sending S_{i+1} as plain text is not secure since $[S_{i+1}]$ will act as the next session key between the tag and the reader. Therefore, random matrices $[S_{i+1}]$ is sent with encryption to the tag. We first use $[Q]$ for randomising S_i and then pseudo-random matrix computation for blinding the matrix $[S_i]$. However, a tag's session key is updated each time period i by computing S_{i+1} from simple decryption using pseudo-inverse matrix properties. More precisely, tag's session key is not updated until a successful reader authentication.

Hash-table lookup: An appropriate lookup hash-function can offer efficient database searching. In our protocol, *index* is updated in both the tag and the reader, as the transaction becomes successful. This demands an efficient hash-table that provide $O(1)$ query, insertion and deletion operations at high loads.[3] We suggest segmented hash table architecture described in Kumar and Crowley (2005), that provides high collision resistance and comparatively low search cost in *worst case* performance. A traditional hash table maps the key, e.g., *index* into a single hash bucket, whereas N-segmented hash table maps into N potential buckets. Therefore, a table with capacity m has equally sized logical segments containing m/N buckets. Here, the hash function is defined as $\mathcal{H}: I \rightarrow \{0, 1, \ldots, m/N - 1\}$ where I is the index space of size n. Let *linear chaining* be used as *searching* technique, then average and worse search time will be $\Theta(1 + \alpha)$ and $\Theta(\log n/\log\log n)$ respectively, where $\alpha = n/m$. To ensure $O(1)$ searches, they utilise N-independent bloom filter[4] to achieve low false positive rates.

4 Security analysis

4.1 SLPN problem

We use a proof method similar to that described in Pietrzak et al. (2011) as Theorem 1 follows. Even though the protocol in our model and that in Pietrzak et al. (2011) are different, a similar proof can be used as both are based on the SLPN* problem. The hardness of SLPN* can be defined using an indistinguishability game. More formally, the 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 $d = l/(2 + \gamma)$. If the SLPN*$(s,\cdot)$ problem is (t, nQ, 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.

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

The protocol has completeness error $2^{-c_\tau . n}$ where $c_\tau > 0$.

Theorem 2: Let an oracle be $\mathcal{O}$ which is either an SLPN*$(s, \cdot)$ oracle or $U_{2l}(\cdot)$ defined in Definition 4. Let $\mathcal{B}$ be a simulator that uses (t, Q, ϵ)-adversary $\mathcal{A}$ such that:

$$\Pr\left|\mathcal{B}^{SLPN^*(s,\cdot)} = 1\right| \geq \epsilon - Q.\alpha'_{l,d} \text{ and}$$
$$\Pr\left|\mathcal{B}^{U_{2l}(\cdot)} = 1\right| \leq \alpha''_{\tau',n}$$

where

$$\alpha'_{l,n} \leftarrow \Pr\left|\left(w(l) < w(d)\right)\right| \leq 2^{-c_\gamma . l}$$

and

$$\alpha''_{\tau',n} \leftarrow \Pr\left|\left(w(r) \leq n \cdot \tau' : r \in_R \mathbb{Z}_2^n\right)\right| \leq 2^{-c_\gamma . n}$$

Therefore, $\mathcal{B}$ can distinguish between two oracles SLPN*(s, ·) and $U_{2l}(\cdot)$ with advantage $\epsilon - Q \cdot \alpha'_{l,d} - \alpha''_{\tau',n}$. Now we can upper bound the gap between two probability that $\mathcal{B}$ outputs:

$$\left|\Pr\left[\mathcal{B}^{SLPN^*(s,\cdot)} = 1\right] - \Pr\left[\mathcal{B}^{U_{2l}(\cdot)} = 1\right]\right| \leq Q.\alpha'_{l,d}$$

This implies the probability of success of the simulator $\mathcal{B}$, and hence the adversary $\mathcal{A}$, in the indistinguishability game.

Interested readers are referred to Pietrzak et al. (2011), for further clarification and proof of the theorem.

4.2 MIM attack

The most sophisticated and realistic attack in an RFID system is the MIM attack. Our protocol is MIM-secure against an active attack from the 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 the *reader* chooses a random variable as challenge, and tag returns the response according to the challenge. The 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[5] 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 Pietrzak et al. (2011), they use randomise-mapping function $f_k(s) = (k \downarrow s : \mathbb{Z}_2^{2l} \rightarrow \mathbb{Z}_2^l)$ for some random s and prove 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, that enhances security. We use an EX-OR operation for hiding s' using T_{id} as key. Note that, the XOR cipher is vulnerable to frequency analysis; therefore, even if the 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 a pseudo-random matrix as blinding factor to transfer the new session key S_{i+1}, which is secure from the pseudo-random matrix property assumption.

4.3 Pseudo-random matrix

We followed the security analysis in Thuc et al. (2010), where it is claimed 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 that an adversary determines the correct Q is $2^{-(l-r)n}$. To ensure security, we need to ensure that $l >> r$, which can be obtained with $l > n$. In our authentication protocol, we let $n \leq l/2$ to ensure a large value of l.

4.4 Forward security

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

4.5 Backward security

Typical RFID tags and their reader communicate only for a short period of time because of the power constraint of a tag. Thus, either we restrict the adversary in such 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 the 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 the SLPN problem, the pseudo-random matrix makes the protocol indistinguishable from the adversarial perspective.

4.7 De-synchronisation attack

We introduce indexing of the tag to get rid of the attack. When the reader and the tag maintain synchronisation, searching hash table becomes very fast with *direct match* technique. However, synchronisation 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. Although it yields worse performance, but after successful authentication synchronisation 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 a 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, the tag is not allowed to update the 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 the same in two different sessions, an adversary can launch a *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 the tag queries the 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

We define oracles according to the following:

- *CTag* (ID) → T_{id}: On input of a tag identifier, this oracle registers the new tag to the reader/server and return a reference T_{id} to resist duplicate *IDs*.
- *Launch*()→ π, s: This oracle launches a new protocol by returning a session identifier π and first transcript s by the reader to ensure reader-initiated protocol.
- *DTag* (T_i, T_j)$_b$ → *vtag*: On input of a tag reference (T_i, T_j), this oracle generates a virtual tag reference *vtag* and stores the triple (*vtag*, T_i, T_j) in a table D, provided that none of the (T_i, T_j) are already referenced in the table. Depending on the value of the random bit b by the challenger, *vtag* either refers to T_i or T_j.
- *Free*(*vtag*)$_b$: On input of *vtag*, b, it erases the volatile memory of the tag T_i(b = 0) or T_j(b = 1) and removes the entry (*vtag*, T_i, T_j) from D.
- *SendTag* (*vtag*, s)$_b$ → t': On input of *vtag*, this oracle sends s to either T_i(b = 0) or T_j(b = 1). It returns the reply t' of the tag or ⊥.
- *UKey*(S_i) → S_{i+1}: A tag key update oracle performed on the tag side which takes S_i as input and outputs an updated key S_{i+1}.
- *SReader*(π, s') → s'': On input of (π, s'), this oracle sends s' to the reader in session π and returns the reply s'' of the reader or ⊥.
- *Result* (π): This oracle returns either 1 or 0 on successful authentication of a tag. But If the session π is not finished, or there exists no session π it returns ⊥.
- *Corrupt* (T_i): On input of T_i, this oracle returns the non-volatile internal state of T_i. Note that, corruption is done w.r.t. tag, not the *vtag*. Therefore, the adversary is forced to corrupt tags T_i that are currently not drawn.

First, we analyse our protocol using the *privacy model* in Hermans et al. (2011). Where challenger runs the $\mathbf{Exp}^b_{\mathcal{A}}(S)$ experiments with the above oracles.

- $b \in_R \{0, 1\}$
- SetupReader (1^{λ})
- $b' \leftarrow \mathcal{A}^{CTag, Launch, DTag, Free, STag, SReader, Result}()$
- return ($b' == b$).

We assume that $\mathcal{A}$ queries the challenger with $\mathbf{Exp}^b_{\mathcal{A}}(S)$ experiments a number of times and hence guess bit b' and wins the privacy game if and only if ($b' == b$). The advantage of the adversary to win is defined as

$$\mathbf{Adv}^b_{\mathcal{A}}(k) = \left|\Pr\left[\mathbf{Exp}^0_{\mathcal{A}}(k)\right] + \Pr\left[\mathbf{Exp}^1_{\mathcal{A}}(k)\right] - 1\right|$$

The reader sends out a random vector s and the tag computes the protocol transcript from the challenge s, combined with shared key k_i and (e, [R]). The reader decrypts the tag's reply and verifies whether it gets right e under the shared key k in the database. In the second phase, it encrypts the random matrix [Q] with the session key P_i and computes the protocol transcript from the challenge vector s' sent from the tag under the shared secret key k_i. Tag can decrypt the matrix [Q] with session key S_i and verify e' under the shared secret key K_i and MAC value s''.

Theorem 3: If the encryption in the protocol described in Figure 1 is indistinguishable then the protocol is strong private for narrow adversaries.

Proof: We analyse our protocol using the *privacy model* in Hermans et al. (2011). Given an adversary $\mathcal{A}$ that wins the privacy game with non-negligible advantage, we consider another adversary $\mathcal{B}$ that can break the *indistinguishability* 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 indistinguishability 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 the hamming weight satisfies $w \le n.\tau'$ 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 the privacy, then $\mathcal{B}$ wins the indistinguishability game; but indistinguishability 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 Ng et al. (2009), the authors have categorised 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 Ng et al. (2009).

Without reader authentication, any adversary can keep querying a tag with any compatible reader until it is desynchronised with a legitimate reader. Therefore, the tag's secret can only be desynchronised 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 of the tag in an online attack. This causes the tag to be in a DoS attack or in a deadlock condition, as it cannot update the key without reader authentication.

Theorem 4: The protocol described in Figure 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. We can reduce the protocol to *narrow-forward* privacy level by two ways. Firstly, by *reduced backward security*, where we restrict the adversary in such a way that 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; which means the adversary misses the protocol transcripts needed to update the compromised secret key (Song and Mitchell, 2008). Secondly, note that *Corrupt*(·) oracle operates w.r.t. *a tag* not with a *virtual tag vtag*, which means adversary is forced to corrupt tags T_i that are currently not drawn. Therefore, after single *Corrupt*(·) oracle, henceforth adversary is allowed to use *DrawTag*(·,·) oracle. Of course, here adversary is not allowed to access *Result*(·) oracle.

Theorem 5: Considering aforementioned assumptions (*Reduced Backward security or disallowing Result*(·) oracle), the protocol described in Figure 1 is semi-forward narrow privacy preserved. □

5 Comparison and performance analysis

In order to support dynamic scalability, the proposed protocol requires to search and store the *lookup hash table* for each transaction, based on the index value in online, to retrieve the corresponding data in the hash-table. However, the data can be pre-computed in the hash-table either in off-line or dynamically in online.

In case of the tag, protocol operations include two random binary vector generation, *one* SLPN problem, *one* EX-OR operation, and *three* binary linear matrix multiplications. For computation, we only consider the SLPN problem and assume the rest of the operations (e.g., calculation hamming weight) to be trivial in terms of computational complexity. The protocol is roughly as efficient as the HB^+ protocol with just twice the key length. Since it is a reduction of the LPN to the SLPN problem, the protocol is secure against quantum adversaries, assuming LPN is secure against such adversaries. There is a natural trade-off between the communication cost and key size. For any constant c ($1 \leq c \leq n$), the 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 the LPN problem. And, in case of storage, only a secret key and an index for the key. As bitwise XOR, matrix multiplication, the hamming weight $w(\cdot)$ and $(a_{\downarrow}b)$ are all binary operation, they can easily be implemented using bit-by-bit serialisation to save hardware gates. In the e-STREAM project, the PRNG operation needs only 1,294 gates to achieve 80-bit security level using Grain-v1 (Cid and Robshaw, 2009). A PRNG requires a linear feedback shift register (LFSR) structure to compute, so LPN problem can share the same LFSR. s' can be deduced from the state variable of PRNG. The cost of a LPN problem and of storing the index and secret key may not be greater than that of a PRNG, and should be less than that of a CRC as well. However, the LPN problem can be implemented using an 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 a λ-bit security level, the overall hardware cost of the proposed protocol for the above mentioned functions on a tag is no more than 1,600 gates, including the cost of non-volatile memory to store the secret key, the 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.

Table 2 Tag resources and security comparison with HB family and others

Scheme	*Storage*	*Computation (major)*	*Authentication*	*Security achieved*	*Hardware (gates)*
Pietrzak et al. (2011)	1S	1 LPN	Tag	1, 4*	≈ 1,600
HB (Hopper and Blum, 2001)	1S	1 LPN	Tag		≈ 1,600
HB$^+$ (Gilbert et al., 2005)	2S	2 LPN	Tag	7	≈ 1,600
HB-MP (Munilla and Peinado, 2007)	2S	1 LPN	Tag	5, 6, 7, 9	≈ 1,600
HB-MP$^+$ (Leng et al., 2008)	2S	1 LPN, 1 HASH	Tag	1, 5, 6, 7, 9	≈ 3,500
F-HB (Cao and ONeill, 2011)	1*I*, 1S	1 PRNG, 2 LPN	Mutual	1, 2, 4*, 5, 6, 7, 9	≈ 3,500
Ours	1*I*, 1S	1 LPN, 1 PIM	Mutual	1, 2, 3*, 4, 5, 6, 7, 8, 9	≈ 1,600
Avoine and Oechslin (2005)	1S	1 PRF, 1 HASH	Tag	2, 4, 6, 8, 9	≈ 6,000
Le et al. (2007)	1*I*, 1S	1 PRF	Mutual	2, 4*, 6, 8, 9	≈ 6,000
Berbain et al. (2009)	1S	1 PRNG, 1 UH	Tag	2, 4, 9	≈ 3,500
Billet et al. (2010)	1S	1 SC	Mutual	2, 4*, 8, 9	≈ 2,000
Tsudik (2006)	1S, 2TS	1 HASH	Tag	4*	≈ 1,500
Chatmon et al. (2006)	1S, 1TS, 1RN	2 HASH	Mutual	4*, 8, 9	≈ 1,500
He et al. (2009)	1RN, 1C, 1TS, 1S	3 HASH	Mutual	2, 4*, 6, 8, 9	≈ 1,500

Notes: Where SC = stream cipher; S = secret key; C = counter; I = index; PRNG = pseudo random number generator; UH = universal hash; PIM = pseudo inverse matrix; LPN = learning parity from noise; TS = time stamp; RN = random number. *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-synchronisation (6), tag cloning (7), replay attack (8), DoS (9).

In Table 2, 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. Additionally, it achieves $O(1)$ time complexity during the synchronised state that resists brute-force searching in each authentication session. Alternatively, hardware cost of the reader is expensive for the purpose of complex computing[6], that results in reduced computing in tag and hence hardware cost. Besides that, the hash-indexed searching technique at the reader, where all the data related to certain tags are stored efficiently as *index*, reduces an exhaustive database search at the reader end. As a consequence, in an RFID system with *remote authentication*[7], reader can use this *index* in *batch mode* operation to aggregate responses from several tags together, that reduces the communication cost between the reader and the server, where each tag contains unique index within the reader's *field of view* at a specific time instance.

6 Conclusions

This paper presents a novel hardware-friendly RFID authentication protocol based on the 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 *semi forward for narrow adversaries* privacy settings. Moreover, scalability of the protocol can be realised best in synchronised and desynchronised modes that ensures infinite DoS resistance. Security and privacy can be protected as long as we allow an adversary not to cope with both tag ID and the secret key simultaneously. In addition, the security and privacy proof follows the 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.

Acknowledgements

Research has been partially supported by Graduate Research Programme (GRP), JAIST foundation grants and NTT C&C grants no. 24.016.

References

Avoine, G. and Oechslin, P. (2005) 'A scalable and provably secure hash-based RFID protocol', *IEEE International Workshop on Pervasive Computing and Communication Security*, March.

Berbain, C., Billet, O., Etrog, J. and Gilbert, H. (2009) 'An efficient forward private RFID protocol', *ACM Conference on Computer and Communications Security (CCS)*, November.

Billet, O., Etrog, J. and Gilbert, H. (2010) 'Lightweight privacy preserving authentication for RFID using a stream cipher', *International Workshop on Fast Software Encryption (FSE)*, February.

Bringer, J., Chabanne, H. and Dottax, E. (2006) 'HB++: a lightweight authentication protocol secure against some attacks', in *SecPerU*, pp.28–33.

Cao, X. and ONeill, M. (2011) 'F-HB: an efficient forward private protocol', *Workshop on Lightweight Security and Privacy: Devices, Protocols and Applications (Lightsec2011)*, 14–15 March, Istanbul, Turkey.

Chatmon, C., van Le, T. and Burmester, M. (2006) 'Secure anonymous RFID authentication protocols', Computer & Information Sciences, Florida AM University, Tech. Rep.

Cid, C. and Robshaw, M. (2009) *The eSTREAM Portfolio 2009 Annual Update*, July [online] http://www.ecrypt.eu.org/stream/D.SYM.3-v1.1.pdf (accessed October 2012).

Gilbert, H., Robshaw, M. and Sibert, H. (2005) 'An active attack against HB+ – a provably secure lightweight authentication protocol', *Cryptology ePrint Archive*, Report 2005/237.

Gilbert, H., Robshaw, M.J.B. and Seurin, Y. (2008a) 'Good variants of HB+ are hard to find', in Tsudik, G. (Ed.): *FC 2008, LNCS*, Springer, January, Vol. 5143, pp.156–170.

Gilbert, H., Robshaw, M.J.B. and Seurin, Y. (2008b) 'HB: increasing the security and efficiency of HB+', in Smart, N.P. (Ed.): *EUROCRYPT 2008, LNCS*, Springer, April, Vol. 4965, pp.361–378.

He, L., Jin, S., Zhang, T. and Li, N. (2009) 'An enhanced 2-pass optimistic anonymous RFID authentication protocol with forward security', in *WiCOM*, pp.1–4.

Hermans, J., Pashalidis, A., Vercauteren, F. and Preneel, B. (2011) 'A new RFID privacy model', *ESORICS 2011*.

Hopper, N.J. and Blum, M. (2001) 'Secure human identification protocols', *Advances in Cryptology – ASIACRYPT 2001, Lecture Notes in Computer Science*, Springer, Vol. 2248, pp.52–66.

Juels, A. and Weis, S.A. (2005) 'Authenticating pervasive devices with human protocols', in Shoup, V. (Ed.): *CRYPTO 2005, LNCS*, Springer, August, Vol. 3621, pp.293–308.

Katz, J., Shin, J.S. and Smith, A. (2010) 'Parallel and concurrent security of the HB and HB+ protocols', *Journal of Cryptology*, July, Vol. 23, No. 3, pp.402–421.

Kumar, S. and Crowley, P. (2005) 'Segmented hash: an efficient hash table implementation for high performance networking subsystems', *ANCS'05 Proceedings of the 2005 ACM Symposium on Architecture for Networking and Communications Systems*, pp.91–103.

Le, T.V., Burmester, M. and de Medeiros, B. (2007) 'Universally composable and forward-secure RFID authentication and authenticated key exchange', *ACM Symposium on InformAtion, Computer and Communications Security (ASIACCS)*, March.

Leng, X., Mayes, K. and Markantonakis, K. (2008) 'HB-MP+Protocol: an improvement on the HB-MP protocol', *IEEE International Conference on RFID*, April, pp.118–124.

Moore, E.H. (1920) 'On the reciprocal of the general algebraic matrix', *Bulletin of the American Mathematical Society*, Vol. 26, No. 9, pp.394–395, DOI: 10.1090/S0002-9904-1920-03322-7.

Munilla, J. and Peinado, A. (2007) 'HB-MP: a further step in the HB-family of lightweight authentication protocols', *Computer Networks*, Vol. 51, No. 9, pp.2262–2267.

Ng, C.Y., Susilo, W., Mu, Y. and Safavi-Naini, R. (2009) 'New privacy results on synchronized rfid authentication protocols against tag tracing', in Backes, M. and Ning, P. (Eds.): *ESORICS, Lecture Notes in Computer Science*, Springer, Vol. 5789, pp.321–336.

Ouafi, K., Overbeck, R. and Vaudenay, S. (2008) 'On the security of HB# against a man-in-the-middle attack', in Pieprzyk, J. (Ed.): *ASIACRYPT 2008, LNCS*, Springer, December, Vol. 5350, pp.108–124.

Pietrzak, K., Kiltz, E., Cash, D., Jain, A. and Venturi, D. (2011) 'Authentication from hard learning problem', *Eurocrypt 2011, LNCS*, Vol. 6632, pp.7–26.

Song, B. and Mitchell, C.J. (2008) 'RFID authentication protocol for low-cost tags', *The ACM Conference on Wireless Network Security, WiSec*, ACM Press.

Thuc, D.N., Hue, T.B.P. and Van, H.D. (2010) 'An efficient pseudo inverse matrix-based solution for secure auditing', *IEEE-RIVF*, Vol. 712, ISBN: 978-1-4244-8072-2.

Tsudik, G. (2006) 'Ya-trap: yet another trivial RFID authentication protocol', in *PerCom Workshops*, pp.640–643.

Notes

1 Result of noisy inner products of vectors.

2 From the properties of pseudo-inverse matrix.

3 To provide scalability.

4 An on-chip predictive filter that supports space-efficient membership queries.

5 We use T_{id} as the secret key k.

6 Searching the database and generating a pseudo-random matrix are the most complex part of the protocol.

7 Tag readers are portable and server access is costly.