Peer-to-Peer Networking and Applications
https://doi.org/10.1007/s12083-020-01007-z

A lightweight multi-party authentication in insecure reader-server channel in RFID-based IoT

Mohammad Mamun[1] · Atsuko Miyaji[2] · Rongxing Luv[3] · Chunhua Su[4]

Received: 28 January 2020 / Accepted: 26 September 2020

Abstract

The rapid proliferation of Radio Frequency Identification (RFID) tags in the past decade has made tremendous impact on our daily lives. As part of Internet of Things (IoT), RFID technology ensures an efficient, secure and reliable system to identify tagged objects in supply chain environment such as manufacturing, automotive and healthcare. Several lightweight authentication solutions have been proposed to satisfy optimal security and privacy features of RFID communication. Hopper-Blum (HB) family of protocols that rely on the hard problem of Learning Parity with Noise (LPN) is a series of lightweight authentication protocol used to identify RFID tags. Our study shows that recent RFID authentication protocols from HB family that mostly focus on two party authentication such as tag-reader authentication, in general, cannot be applied directly to a three party authentication such as tag-reader-server authentication. In contrast to typical RFID authentication system, we consider the channel between the reader and back-end server insecure. We focus HB protocol and its variants and propose a modified protocol where the entire system is authenticated under LPN-based scheme.

Keywords Mutual authentication · HB-family protocol · Pseudo-inverse matrix · Key-evolving signature

1 Introduction

As a part of IoT sensors, RFID-based technology is not just a futuristic vision- but rather a technology that is being deployed successfully in applications ranging from aviation system, manufacturing, smart applications to healthcare and safety system [4, 5]. RFID based ecosystem evolves as an example of P2P-based IoT that provides a full communication infrastructure by enabling data to be shared between an end-user client device such as smart phone and an RFID-enabled IoT device [31]. In contrast to classical IoT solutions such as database-driven IoT where all data travels through a centralized server e.g. a cloud database, P2P IoT connection allows sharing data directly between the client device and IoT device. Database server can be used in P2P-based IoT only for *initiating secure connection* such as authenticating devices. Once the connection is established, the communication is transmitted to the client and IoT devices. P2P based IoT model is therefore comply with the General Data Protection Regulation (GDPR) guideline that includes restrictions for IoT data stored on the vendor's servers [32].

However, in a heterogeneous RFID enabled IoT system such as P2P-based IoT system where devices belong to different security realms do not have direct authentication relationships, enabling reliable secure communication is a tremendous technological challenge. For example, US hospitals use RFID-enabled bracelets (RFID reader) connected to central monitoring system (server) to monitor bandages and other supplies and to track patients movement around

✉ Mohammad Mamun
mohammad.mamun@nrc-cnrc.gc.ca

Atsuko Miyaji
miyaji@comm.eng.osaka-u.ac.jp

Rongxing Lu
rlu1@unb.ca

Chunhua Su
chsu@u-aizu.ac.jp

1 Digital Technology, National Research Council of Canada/Government of Canada, Fredericton, Canada

2 Osaka University, Osaka, Japan

3 Faculty of Computer Science, University of New Brunswick, Fredericton, Canada

4 The University of Aizu, Fukushima, Japan

Published online: 21 November 2020

the facility [29]. A leading U.S. healthcare provider *Permanent Capitalist* uses RFID medical device tracking system to manage more than 140 thousands electronic medical devices and supplies in order to automate the role of healthcare practitioners and ensure timely service [30]. This paper focuses on the fast and secure connection establishment in a P2P-based IoT environment where a database server and RFID reader are involved initially for authentication and session key management. A similar approach has been proposed in [33] where authors propose a P2P based RFID network architecture and a mutual authentication protocol using asymmetric encryption algorithm such as RSA.

Based on the comprehensibility and scarce resources such as memory of a tag RFID protocols can be classified into heavyweight, lightweight, ultra-lightweight protocol [2]. Lightweight protocol construction utilizes pseudo random number generator, one-way hash function. While ultra lightweight protocols mainly use bitwise operation for tag computation. HB family protocols belong to between ultra-lightweight category as the tag computation algorithm in HB protocol exhibits low cost bitwise operation following noise vector generation from Bernoulli distribution. Mutual authentication adds an additional protection for an RFID system 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*. 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 be secure [1, 3, 9, 10, 18, 27]. We believe that this is the first mutual authentication protocol in the HB family that considers the communication channel between the reader and server is insecure.

Limited processing and storage capability of traditional RFID tags limit the use of strong cryptographic tools such as RSA, ECC [6, 7]. For example, EPC Class-1 Gen-2 allows only 2500 gates for security operations that resists standard cryptographic techniques such as RSA to implement [6]. Similarly, stronger security primitives such as SHA-1, MD5 require 16000-20000 gates, elliptic curve cryptography (ECC) require approximately 15000 gates [7]. However, HB protocols that require a few thousand gates for implementation make them an attractive option for securing low cost EPC tags [21].

Since its first introduction in 2001 [8], HB-family protocols (HB^{+}, HB^{++}, $HB^{\#}$, HB-MP, HB-MP^{+}, HB^{*}, F-HB etc. [9–13, 15, 19, 27]) have been used in numerous applications such as lightweight crypto system, symmetric encryption due to efficacy of algorithm, robust security against quantum algorithm.

Our contribution: We extend the security scope of a variant of HB-protocol in [17] and [21] by adding some non-linear components without significant increase of its complexity. We use the commitment scheme presented in [22] for authentication where security is based on the hardness of exact LPN and improve exiting commitment scheme by incorporating properties of pseudo-inverse matrix based short signature. In addition to the privacy notion that captures the privacy of both tag-reader and reader-server transactions, our protocol considers only the back-end server to be a trusted entity in the network. We demonstrate forward privacy under zero-knowledge (ZK) indistinguishable notion and also provide security proof under standard model. To the best of our knowledge, this is the first RFID authentication protocol construction in HB-family where both tag/reader and reader/server channels are assumed to be insecure. It is intended to raise interests in other IoT security applications such as *authorization recovery, ownership transfer, controlled delegation* etc.

2 Models and building blocks

In this section, we formalize background, system model, security notions in this paper, and identify our building blocks.

2.1 Types of RFID authentication

Authentication protocols in an RFID system can be roughly categorized into three types See Fig. 1.

- *Tag authentication* refers to the process of identifying a tag based on some secret where authentication merely ensures that the tag is which it claims to be.
- *Mutual authentication* (also called two-way authentication) refers to two parties (tag and reader) authentication each other at the same time. Communication channel between the reader and server is assumed to be secure in this case.
- *Multi-party mutual authentication* refers to all parties (tag, reader and server) authentication where entities authenticate each others at the same time. In our work, we focus on the multi-party authentication between the tags and the server through readers where both the communication channels tags-readers and readers-server are considered insecure. In order to simplify the description of our proposed scheme, we consider only one RFID-reader.

2.2 System model

We envision a P2P based IoT environment where an end-user client device needs to establish a direct secure connection initially to an RFID-enabled IoT device. Our proposed

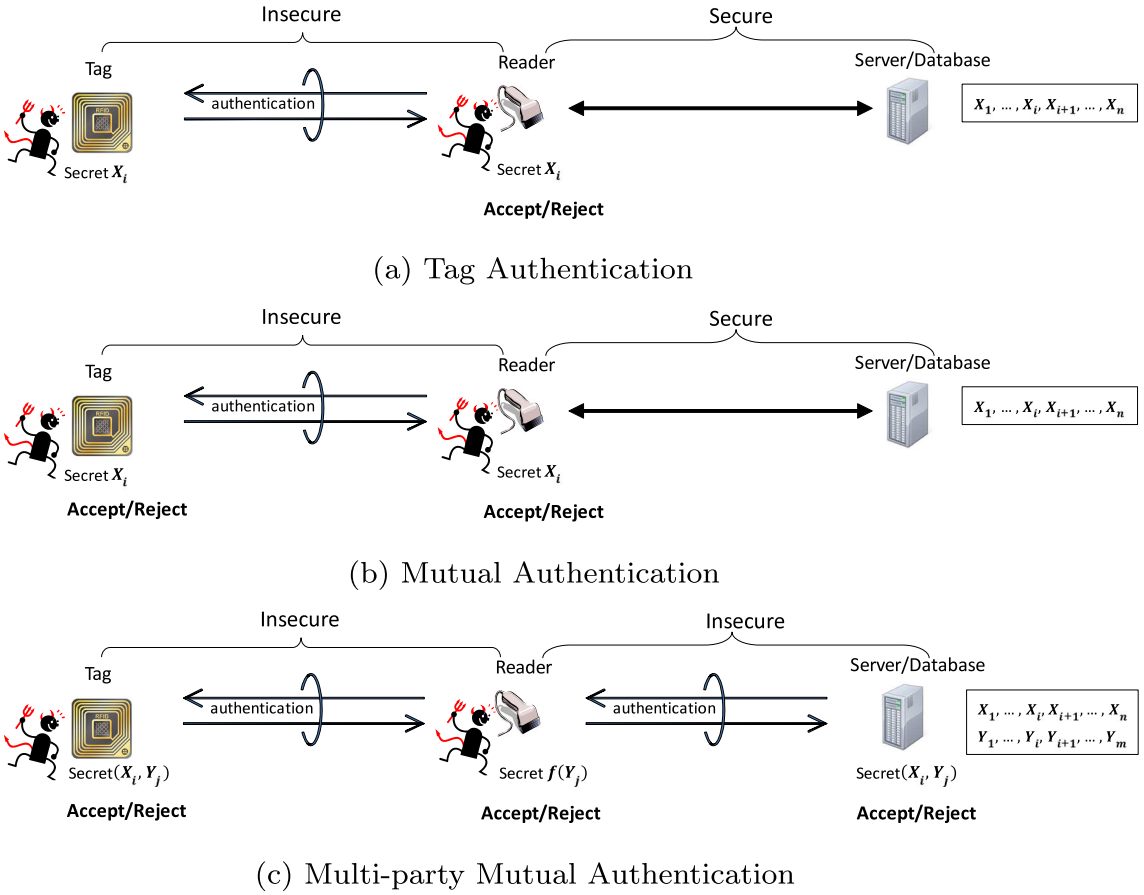


Fig. 1 Types of RFID Authentication

protocol can be used to mutually authenticate different parties such as database server, Reader, and generate session keys for further communication. Once the connection is established (authenticated), the communication including session keys is handed over to the end-user client and the IoT device where the server is not involved any more. This enables the end- user client in a P2P IoT solution to get full control of the data while maintaining low latency.

As shown in Fig. 2, the system mainly contains three entities: a database server, a set of RFID readers, and tags.

– *Server*. A database-driven IoT server offers authentication service for RFID tags. Communication channel between the server and the reader is assumed to be wireless and insecure. Authentication data travels to a centralized server that processes the authentication data

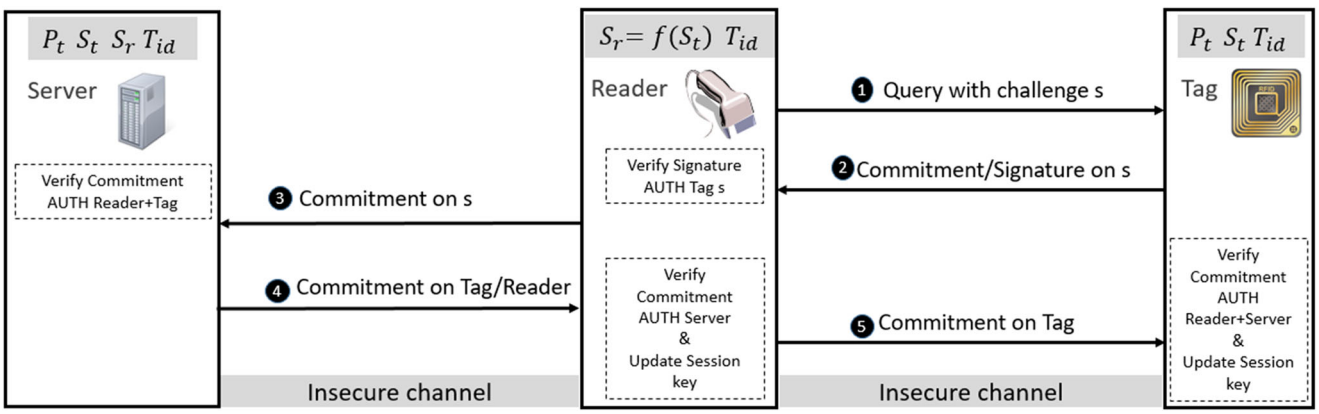


Fig. 2 System model [Permanent Key (P_t), Session Key (S_t)]

and stores session key for the next connection. It helps establish secure connection between the client device and the RFID-enabled IoT device indirectly. Once the connection is established, computation and network resources are owned by the client devices. The server is not be willing to disclose permanent secret key of an RFID tag to the clients. Thus, it authenticates all the parties such as readers, tags, and sends the session keys with identity information to the authenticated clients. Since the server is not involved in any business logic but secure connection mediation, hybrid solution like this massively reduces complexity and helps developing P2P applications using traditional P2P paradigm.

– *RFID Readers*. RFID readers are portable devices that can be connected wirelessly to the server and use radio frequency waves to transmit signals between themselves and the tags. Since the communications between the tag and the reader is wireless, it can be easily eavesdropped. Therefore, wireless channel is always considered to be insecure and prone to malicious attacks by counterfeit or unauthorized readers. The readers have sufficient resources to process and store strong cryptographic tools and data in order to resist adversarial attacks. Readers connected to the back-end server store all the secrets related to the tags.
– *RFID Tags*. RFID tags, consisting of an embedded transmitter and non-volatile storage, are mainly used to allow everyday objects interact and report their status to the reader. Each tag authenticates not only its licit reader but also the legitimate server. Having limited computing power and memory resources make RFID tags an easy target of attackers and authentication protocols a major source of vulnerabilities. Since an RFID tag is not tamper-resistant, session keys are updated at each new transaction. Database at the server binds tag indexes to the session keys.

2.3 Design goal

Based on the system model described in Section 2.2, our goal is to develop an HB-like, privacy-preserving lightweight multi-party authentication protocol in P2P IoT system. The following are three major objectives that need to be achieved.

– *Efficiency*. Considering real-time secure low latency connection requirement for the P2P-based IoT smart solution, the proposed authentication system should be efficient in communication and computation. In addition, resource-constraint and cost-efficient RFID tags have limited storage capacity. In an authentication scheme, achieving all the desired security and privacy goals while satisfying efficiency is challenging. Efficiency of our authentication protocol is comparable to the HB-family of lightweight authentication protocol.
– *Privacy*. Anonymity is commonly used as a privacy measure against attacks. our authentication scheme, we consider Zero-Knowledge (ZK) based privacy, a variant of indistinguishability (IND) based privacy that assures that the replies from two different tags are computationally indistinguishable without the knowledge of internal states. What's more, if an attacker knows the ID of the tag, it cannot determine whether a tag was involved in any past (backward privacy) or future (forward privacy) protocol transaction it may have recorded. However, privacy can be guaranteed as long as no tag is compromised before it is activated. This is because the adversary can differentiate tag's reply based on the past experiment.
– *Security*. In P2P IoT system, tags' ID and permanent keys used for devices are required to be confidential to the clients as it is a natural approach to defend counterfeiting attack to the device identification. In our system, authenticated clients are allowed to make use of the temporary session keys generated at each successful protocol exchange. Only the clients who have registered to the system and gained the session keys, can get the access to the RFID-enabled IoT device. Moreover, an attacker cannot learn about the ID of a tag by following the protocol transactions between a tag, a reader, and a server. We discuss the protocol's resistance to man-in-the-middle attack, tag tracking attack, dysynchronisation attack, traceability, and compromising. Nonetheless, the well-known DoS attacks were not taken into account in this work.

2.4 Building blocks

LPN problem Let for a noise-parameter $\tau \in]0, 1/2[$, *Bernoulli distribution* Ber_τ output 1 with probability τ and 0 with probability $(1-\tau)$. For $k \in \mathbb{N}$, the *decisional*-LPN problem is (q, t, ϵ)-hard if for any distinguisher $\mathcal{D}$ running in time t

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

where $\mathsf{x} \in_R \mathbb{Z}_2^k$, $\mathsf{A} \in_R \mathbb{Z}_2^{q \times k}$, $\mathsf{r} \in_R \mathbb{Z}_2^q$, and $\mathsf{e} \in \mathsf{Ber}_\tau^q$. Subsequently, the *search*-LPN problem is (q, t, ϵ)-hard if for every $\mathcal{D}$ running in time t

$$\mathsf{Pr}[\mathcal{D}(\mathsf{A}, \mathsf{A.x} \oplus \mathsf{e}) = x] \leq \epsilon \tag{2}$$

Let $y = \mathsf{A.x} \oplus \mathsf{e}$, then *computational*-LPN problem is to compute x and e from a given (A, y) pair. Note that, in

the standard definition of the LPN problem, the error vector $\mathsf{e} \in \mathbb{Z}_2^q$, from Bernoulli distribution Ber_τ with parameter $0 < \tau < 1/2$ yields the *expected* Hamming weight to be $q\tau$. However, in case of *exact*-LPN (LPN_x) in [22], the problem is defined exactly like ordinary LPN except that the Hamming weight of the error vector is defined exactly $\lfloor q\tau \rceil$. That means, e is chosen independently and identically from $\mathsf{Ber}_{\lfloor q\tau \rceil}$.

Commitment scheme [22] It is a two-phase protocol between a *sender* and a *receiver* where the sender holds a message m. In the first phase, the sender 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:

- $\mathsf{KeyGen}(l) \rightarrow ck$: On input 1^l, the key generation algorithm KGen output a commitment key ck.
- $\mathsf{Com}(m, ck) \rightarrow (c, d)$: The commitment algorithm 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).
- $\mathsf{Ver}(m, ck, c, d) \rightarrow$ (1 or 0): The verification algorithm takes commitment key ck, a message m, a commitment-opening pair (c, d), and output 1 (success) or 0 (reject).

A commitment scheme should satisfy the following *three* security properties:

- *Correctness*: $\mathsf{Ver}(ck, m, c, d)$ should result to 1 if the inputs are computed by an honest party, such that,

$$\begin{aligned}\Pr[\mathsf{Ver}(ck, m, c, d) = 1;\ ck \leftarrow \mathsf{KGen}(1^l),\\ m \in \mathcal{M}, (c, d) \leftarrow \mathsf{Com}(m, ck)] = 1\end{aligned} \tag{3}$$

- *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.,

$$\begin{aligned}\Pr[ck \leftarrow \mathsf{KGen}(1^l); \forall\, m, m' \in \mathcal{M} \wedge (c, d) \leftarrow \mathsf{Com}(m, ck),\\ (c', d') \leftarrow \mathsf{Com}(m', ck) : c = c'] = 1/2\end{aligned} \tag{4}$$

- *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 $ck \leftarrow \mathsf{KGen}(1^l)$, no commitment c can be opened in two different ways, s.t.,

$$\Pr[(\mathsf{Ver}(ck, m, c, d) = 1) \wedge (\mathsf{Ver}(ck, m', c, d') = 1) : m \neq m'] \leq \epsilon \tag{5}$$

Pseudo-inverse matrix A pseudo-invertible matrix X has its unique inverse. If Y, Z be two pseudo-inverse matrices of X, then the following occurs.

$$XYX = X \text{ and } XZX = X \tag{6}$$

$$Y = YXY = ZXY = ZXZ = Z \tag{7}$$

$$(\mathrm{XYX})^{\mathrm{T}} = \mathrm{X}^{\mathrm{T}}\mathrm{Y}^{\mathrm{T}}\mathrm{X}^{\mathrm{T}} = \mathrm{X}^{\mathrm{T}} = \mathrm{X}^{\mathrm{T}}\mathrm{Z}^{\mathrm{T}}\mathrm{X}^{\mathrm{T}} = (\mathrm{XZX})^{\mathrm{T}} \tag{8}$$

$$\begin{aligned}XY &= (XY)^T = Y^T X^T = Y^T (X^T Z^T X^T)\\ &= (XY)^T (XZ)^T = XYXZ = XZ\end{aligned} \tag{9}$$

Following the security analysis in [14], it is hard to recover $\mathbf{X} \in \mathbb{Z}_2^{k \times v}$ or $\mathbf{Q} \in \mathbb{Z}_2^{k \times k}$ from the message $m \leftarrow \mathbf{X}\mathbf{X}^{+}\mathbf{Q} \in \mathbb{Z}_2^{k \times k}$ when $k \gg v$ which can be easily obtained if $k = \Theta(v + l)$.

Key Evolving Signature It is a stateful signature scheme where a key-pair's lifespan is split into several time spans. A key updating algorithm is invoked at the end of each time period in order to update secret keys. The end of the time period can be determined by the time or occurrence such as the number of signatures a secret key can be used for. For example, in our scheme, a period ends after signing one message and the key update algorithm takes place after every successful authentication. A quadruple algorithms KeyG, KeyU, Sign, Ver is called key evolving signature scheme if it satisfies the following:

- $\mathsf{KeyG}(l) \rightarrow (sk_0, vk_0)$: On input of the security parameter l, it outputs an initial signing key sk_0 and a verification key vk_0.
- $\mathsf{KeyU}(k, i) \rightarrow k_{i+1}$: On input of a key k and an index i, this algorithm outputs the key k_{i+1} for the next time period given the key k is a valid key for time period i. Otherwise it returns null.
- $\mathsf{Sign}(m, sk, i) \rightarrow (\sigma, i)$: On input of a message $m \in \mathcal{M}$, a signature key sk, and an index i, this algorithm outputs the signature (σ, i) if sk is a valid key for time period i, it returns null otherwise.
- $\mathsf{Ver}(vk, m, (\sigma, i))) \rightarrow$ (1 or 0): It's a deterministic algorithm that on input of a verification key vk , message m, and a signature (σ, i) outputs $\mathsf{1}$ ($\mathsf{success}$)

if σ is a valid signature on m for time period i and 0 (reject) otherwise.

3 Protocol construction

We propose a *multi-party* authentication protocol that attains mutual authentication not only in the tag-reader channel, but also in the reader-server channel. It provides secure multi-party− tag, reader, and server communication. We mainly incorporate two cryptographic building blocks− LPN based commitment scheme, and a pseudo inverse matrix based key evolving signature into an authentication protocol. A skeleton of the protocol is given in Fig. 2.

3.1 Notation

Table 1 summarizes the notations used in this work.

3.2 Predeployment phase

The server is the trusted entity in our model and sets up the system, that takes a security parameter κ, and runs a generation algorithm $\mathtt{Gen}(\kappa)$ for LPN-based commitment scheme. The outcome of the generation algorithm is the public parameters (τ, k, l, v, τ') (see the definition in Table 1). Given the security parameter l, the server runs key generation algorithm $\mathsf{KeyG}(l)$ to generate a signing key S_0'' and corresponding verification key P_0, that is, the initial session key pair for the tag−reader $(\mathrm{S}_0'', \mathrm{P}_0)$.

$$P_0 \leftarrow \mathbf{S}_0'' \cdot \mathbf{S}_0''^{+} \tag{10}$$

where $\mathbf{S}_i''^{+}$ is the pseudo-inverse of the matrix $\mathbf{S}_i''$. In the setup phase, the server also initiates each tag with an index I_i, a permanent key S'. Note that the session keys such as P_i, S_i'' and tag index I_i are updated at i^{th} instance by the server and later followed by the reader and the tag.

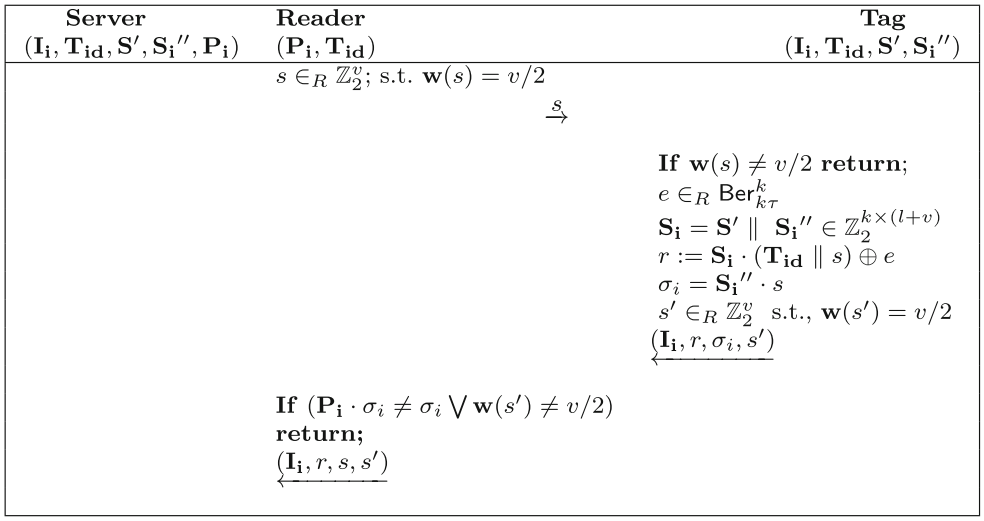
3.3 Step-1: Tag-Reader Interaction (Fig. 3)

– *Reader*: The reader holds the tag id T_{id} and session/verification key P_i either from the initial setup or derived from the previous $(i-1)^{th}$ successful session. In order to initiate protocol, the reader generates a ran-

Table 1 Definition of notations used in the protocol

Notation	Definition
$\mathbb{Z}_p$	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
$\mathbf{S}'$	$k \times l$ bit matrices as the permanent *secret* commitment key between the server and the tag
$\mathbf{S_i''}$	$k \times v$ 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 Ber_τ 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 $\mathsf{Ber}_{k\tau}^k$ 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
$\mathbf{[S]^T}$	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., $\mathbf{A}^{(+/-)} : \mathbb{Z}_2^{k\times v} \rightarrow \mathbb{Z}_2^{v\times k}$
$\mathbf{a} \cdot \mathbf{b}$	Inner-product of two vectors (or matrices) $\mathbf{a}$ and $\mathbf{b}$.
$\oplus, \|$	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., $a < x < b$

Fig. 3 Tag-Reader Interaction



dom binary v-bit challenge string s, and send it to the tag.

– *Tag*: The tag holds a unique identifier T_{id}, a permanent key S_i', S_i'' and I_i from the initial setup phase or derived from the previous $(i-1)^{th}$ successful session. Check the *hamming weight* of the string s and generate a k-bit noise vector e from Bernoulli distribution $\mathsf{Ber}_{k\tau}$, a random v-bit challenge string s' with hamming weight $v/2$. Next, a k-bit commitment string r on the message s, that is, a commitment-opening pair (r, s) is generated.

$$r := \mathbf{S}_i \cdot (\mathrm{T}_{id} \parallel s) \oplus e \tag{11}$$

Note that, a new session key $\mathbf{S}_i$ is generated by combining two keys– a permanent key $\mathbf{S}_i'$ and a session key $\mathbf{S}_i''$.

$$\mathbf{S_i} = \mathbf{S'} \parallel \mathbf{S_i}'' \in \mathbb{Z}_2^{k\times(l+v)} \tag{12}$$

To demonstrate the authenticity of the challenge message s, a lightweight and key-evolving signature σ_i is generated by using the signing/session key S_i'' on the message s.

$$\sigma_i = \mathbf{S_i}'' \cdot s \tag{13}$$

The tag generates a random v-bit binary challenge string s' for the commitment algorithm to be executed in the server. Finally, the tag forwards the protocol message (I_i, r, σ_i, s') to the corresponding reader.

– *Reader*: The reader conveys the protocol messages it received from the tag to the server if it can successfully verifies the signature σ_i by using the verification key P_i for the i^{th} session. That is, before forwarding the protocol message (I_i, r, s, s') to the server, it verifies the sender tag σ_i whether it has been generated from the message/challenge s initiated by the reader itself and proves the hamming weight of s' as follows:

$$P_i\sigma_i = (\mathbf{S}_i''\mathbf{S}_i''^{+})(\mathbf{S}_i''s) = \mathbf{S}_i''s = \sigma_i \tag{14}$$

3.4 Step-2: Reader-server Interaction (Fig. 4)

– Server: The server holds (T_{id}, S_i', S_i'') and (T_{id}, P_i) shared with the tag and reader respectively. After receiving the protocol message from the reader, it first searches the database with I_i to find out a tuple $[I_i, T_{id}, \mathbf{S}_i'', r_{i-1}, \mathbf{S}_{i-1}'']$. However, searching may fail due to synchronization attack. In that case, the server can try with previous index $[I_i \stackrel{?}{=} r_{i-1}]$ stored for $(i-1)^{th}$ session targeting previous transaction parameters: $(\mathbf{S}_{i-1}'', r_{i-1})$ to recover. Given a commitment message r on a message s forwarded by the reader, it verifies the commitment if and only if the following holds, and hence *authenticates the tag*.

$$\begin{aligned} \mathbf{w}(\mathbf{S}_i \cdot (\mathrm{T}_{id} \parallel s) \oplus r) \stackrel{?}{=} \lfloor k\tau' \rceil \mathbf{w}(s') \stackrel{?}{=} v/2 \\ \mathbf{w}(s') \stackrel{?}{=} v/2 \end{aligned} \tag{15}$$

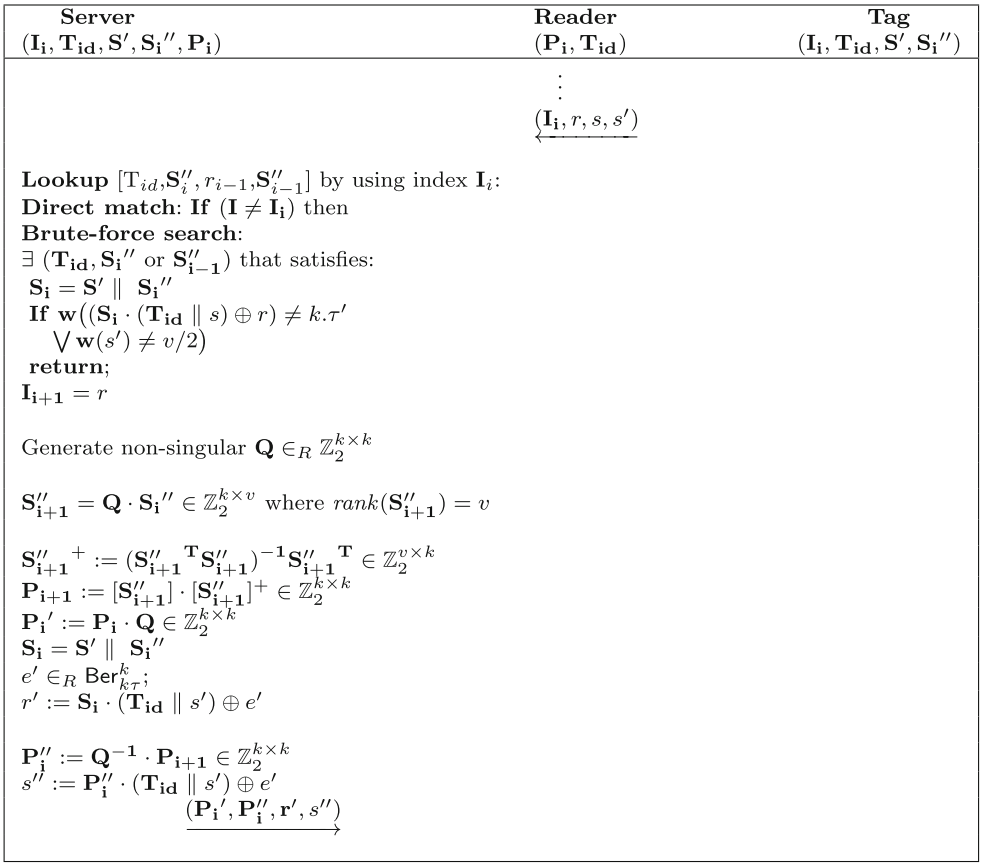
Consequently, it updates the index to I_{i+1} with r for the next session.

The server then generates a non singular binary matrix Q to update the signature/session key for the time period i to $\mathbf{S}_{i+1}''$ as part of key-evolving signature scheme. Verification/session key for the reader P_{i+1} and pseudo inverse-matrix $\mathbf{S}_{i+1}''^{+}$ can be derived from $\mathbf{S}_{i+1}''$.

$$\begin{aligned} \mathbf{S}_{i+1}'' &= [\mathbf{Q} \cdot \mathbf{S}_i''] \\ \mathbf{S_{i+1}}^{+} &:= (\mathbf{S_{i+1}}^{\mathbf{T}}\mathbf{S_{i+1}''})^{-1}\mathbf{S_{i+1}}^{\mathbf{T}} \\ \mathbf{P_{i+1}} &:= [\mathbf{S_{i+1}''}] \cdot [\mathbf{S_{i+1}''}]^{+} \end{aligned} \tag{16}$$

In order to securely share the new session key for the reader, the server uses the blinding matrix Q and

Fig. 4 Reader-Server Interaction



compute

$$\begin{aligned} \mathbf{P}'_i &:= \mathbf{P}_i \cdot \mathbf{Q} \\ \mathbf{P}''_{\mathbf{i}} &:= \mathbf{Q}^{-\mathbf{1}} \cdot \mathbf{P_{i+1}} \end{aligned} \tag{17}$$

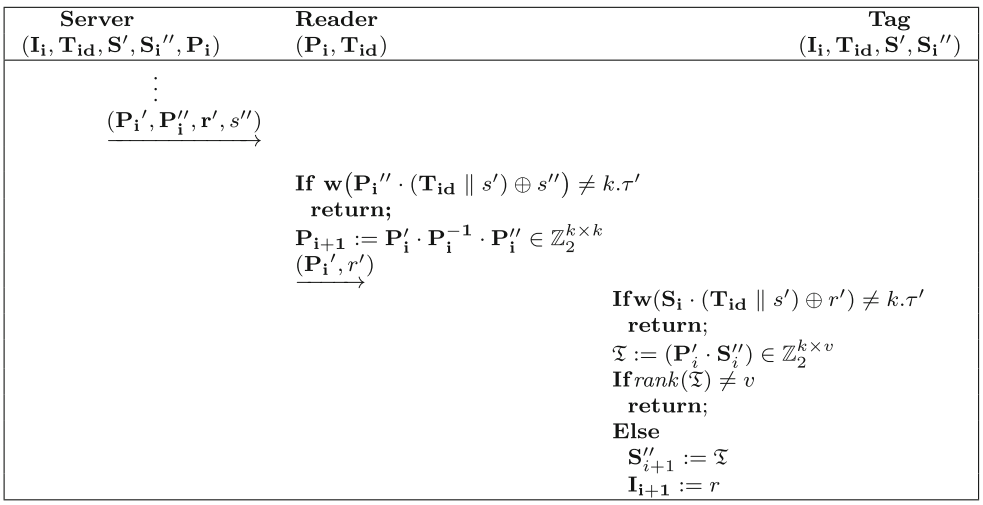
Subsequently, commitment messages r' and s'' are generated with a view to authenticate the server by the tag and the reader respectively.

$$\begin{aligned} r' &:= \mathbf{S}_i \cdot (\mathbf{T}_{id} \parallel s') \oplus e' \\ s'' &:= \mathbf{P}''_{\mathbf{i}} \cdot (\mathbf{T_{id}} \parallel s') \oplus e' \end{aligned} \tag{18}$$

where e' is a k-bit randomly generated noise vector.

Finally, the protocol message $(\mathrm{P}'_i, \mathrm{P}''_i, r', s'')$ is forwarded to the reader.

Fig. 5 Reader-Tag Interaction



3.5 Step-3: Reader-tag interaction (Fig. 5)

– Reader: Given a commitment-opening pair (s'', s') forwarded by the server, it verifies the commitment to be successful, and hence *authenticates the server*.

$$\mathbf{w}(\mathbf{P}_i'' \cdot (\mathbf{T}_{id} \parallel s') \oplus s'') \stackrel{?}{=} \lfloor k\tau' \rceil \tag{19}$$

As the reader verifies the commitment message successfully, it updates it's verifier/session key P_i and forwards the protocol message (P_i', r') to the tag.

$$\mathbf{P}_{i+1} := \mathbf{P_i'}\mathbf{P_i^{-1}}\mathbf{P_i''} = (\mathbf{P_i}\ \mathbf{Q})\ \mathbf{P_i^{-1}}(\mathbf{Q^{-1}}\mathbf{P_{i+1}}) \tag{20}$$

– Tag: After a successful verification of the commitment r' on the message s', it accepts the reader as well as the server. That is all the parties are now mutually authenticated.

$$\mathbf{w}(\mathbf{S_i} \cdot (\mathbf{T_{id}} \parallel s') \oplus r') \stackrel{?}{=} \lfloor k\tau' \rceil \tag{21}$$

The tag then updates the signing/session key $\mathbf{S''_{i+1}}$ for the next time period $i+1$. However, in order to prevent brute-force searching at the server end, an index I_i is maintained and updated at each session by the tag.

$$\mathbf{S}_{i+1}'' = \mathbf{P}_i' \cdot \mathbf{S}_i'' = \mathbf{S}_i''\mathbf{S}_i''^{+}\mathbf{S}_i''\mathbf{Q} = \mathbf{Q}\mathbf{S}_i'' \tag{22}$$

$$\mathrm{I}_{i+1} \leftarrow r \tag{23}$$

4 Security analysis

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.*

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 2 *The commitment scheme described in the protocol 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 $(x, y) \in \mathcal{R}$ and 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. For more detail clarification, we refer to the [22] where authors describe an efficient simulator for indistinguishability game. □

Proposition 1 *The exact version of search* LPN_x *is hard if and only if standard search* LPN_τ *is hard .*

Proof Let an adversary $\mathcal{A}$ find out the secret x with advantage ϵ for LPN_x where the error vector e', sampled in LPN_τ, has weight $\lfloor k\tau \rceil$ with probability at least $1/\sqrt{k}$ such that

$$\Pr[\mathcal{A}(A, A.x \oplus e') = x] \leq \epsilon/\sqrt{k}$$

where $e' \in \mathsf{Ber}^k_{\lfloor k\tau \rceil}$. It is not hard to see that error distribution on the above case is exactly same as that of *exact search* LPN_x. For a detail proof see [22] □

Proposition 2 *The hardness of decisional* LPN_x *is polynomially related to that of search* LPN_τ.
Proof: The standard search and decision LPN_τ *are equivalent. However, reduction from the search to decision incurs the number of samples k in the decision-*LPN_τ *to be larger than that in the search-*LPN_τ [26, 27]. *However, Hardness of the* LPN_x *problem holds assuming the hardness of the standard* LPN_τ *problem, where the reduction is based on*

the Goldreich-Levin theorem described in [20]. *Note that if security of the scheme is 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* [22], *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* [22], *for further clarification and proof of the theorem.*

Theorem 3 *If Binary Matrix Factorization (BMF) is a hard problem, then construction of the lightweight stateful signature is existentially unforgeable under a one-time chosen message attack.*

Proof Let $\mathcal{A}$ be a PPT adversary such that $(pk,\ sk) \leftarrow \mathsf{KGen}(1^k)$; $(m, \sigma) \leftarrow \mathcal{A}^{\mathsf{Sign}_s k(\cdot)}(pk)$ where $\mathcal{A}$ is allowed only a single query to its signing oracle. Given the public key $pk = X^+X$, $\mathcal{A}$ needs to find an X^+ such that $f(X^+) = X^+X$ where X^+ is the unique pseudo-inverse of a matrix X. □

Let m' be the message that $\mathcal{A}$ queries to its signing oracle then the signature scheme is assumed to be forged on the event: $\mathrm{Vrfy}(m, \sigma) = 1$ and $m \neq m'$. In a certain experiment $\mathrm{Exp}_{\mathcal{A},\Pi}(1^k)$ of the signature scheme Π, success probability can be defined:

$\mathsf{Succ}_{\mathcal{A},\Pi}(k) := \mathsf{Pr}[(m, \sigma) \leftarrow \mathrm{Exp}_{\mathcal{A},\Pi}(1^k) : \mathrm{Vrfy}(m, \sigma) = 1 \text{ and } m \neq m']$

Since BMF is hard, $\mathcal{A}$ runs the experiment $\mathrm{Exp}_{\mathcal{A},\Pi}(1^k)$ by choosing a random $X^+ \in \mathbb{Z}_2^{m\times n}$ where m is the rank of matrix X^+. Then setting $pb := X^+X$, $\mathcal{A}$ will try to output forgery. Since pb is updated at every state by using a random matrix M. Given $XX^+M \in \mathbb{Z}_2^{n\times n}$, suppose that $rank(X^+) = r$, and $I^{r\times r}$ is an *Identity matrix* s.t.,

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

Then the probability of $\mathcal{A}$ to determine the correct M is $2^{-(n-r)m}$ [21] where $n \gg r$ such that $n \gg m$. Hence the probability that $\mathcal{A}$ outputs forgery is at least $\mathrm{Succ}_{A,\Pi}(k)/2^{-(n-r)m}$. Since sk is updated at each protocol transaction $\mathrm{Succ}_{A,\Pi}(k)/2^{-(n-r)m} \leq neg(k)$. We conclude that $\mathrm{Succ}_{A,\Pi}(k)$ is negligible. It is worth mentioning that the signature scheme is might be insecure if the sk is used to sign more than one message. Adversary $\mathcal{A}$ who obtains several signature w.r.t the same pk might learn the entire sk by using Gaussian elimination method.

Proposition 3 *Protocol can resist Man-in-the Middle (MIM) attack.*

Proof The most sophisticated and realistic attack in an RFID system is the MIM attack. Our protocol is MIM-secure against an active attack from several assumptions i.e, the exact LPN, a lightweight signature from 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 2. shows that protocol transcripts are computationally indistinguishable from the exact $\mathsf{LPN}x$ assumption, the tag-reader communication is MIM-secure. □

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}.

Proposition 4 *Protocol can resist tag tracking attack.*

Proof Let a tag's responses to a certain reader be linkable to each other, or distinguisahble from other tags. As a result, location of the tag can be tracked by a malicious adversary, called unauthorized *tag tracking*. This yields a serious threat to the privacy that can be avoided if the protocol responses appear to an attacker is random and uniformly distributed. Our authentication scheme is *tracking resistant* under the security of exact-LPN, the pseudo-randomness of LPN. We consider tag tracking in the context of the server-tag communication. Note that under the decisional LPN assumption $(s, \mathbf{S}.(\mathbf{T} \parallel s) \oplus e)$ samples are pseudorandom. Even if an adversary has access to the protocol message (s, r) or (s', r'), session key $\mathbf{S_i}$ cannot be deduced efficiently. Because the success probability of guessing the correct secret key is negligible ($\approx 2^{k\times(l+v)}$) even without considering the permanent secret $\mathbf{T}$. In order to resist *side channel* and *replay* attack, we use random nonce (s, s') in tag-server communication. In addition, since $\mathbf{S}''$ is refreshed at each protocol session, even if an attacker replays the previous session message in an attempt to eavesdrop necessary number of outputs of the tag to break security, authentication will fail as each $\mathbf{S}''$ is used

only once. Note that the tag-reader communication is not important to consider for tag tracking, since the shared secret for the LPN problem (tag-server communication) is not identical to that of tag-reader. However, in case of tag-reader communication, we use a stateful signature scheme where signing key is refreshed at each protocol session. This will also help the protocol transaction to be unlinkable. □

Proposition 5 *Protocol can resist desynchronization attack.*

Proof Desynchronization is a kind of Denial of Service (DoS) attack where the tag and reader/back-end server cannot recognize each other (due to adversarial interruption or impediment) and finally the tag gets disabled. If tag authentication involve random initialization from the tag, it can cause replay attack and hence yields desynchronisation in the tag-server communication. Therefore, our protocol has been initialized by the reader. An adversary can produce a valid protocol message only if it successfully discovers the shared secrets among the entities. Since the tag is assumed to be uncorrupted and the LPN and BMF problems are NP-hard, a PPT adversary cannot learn the secrets. The only way to desynchronize the tag is to block the last message of the protocol and to update the shared session secret key ($\mathbf{S_i''}$) (contrary to the server). More precisely, let the tag $\mathcal{T}$ and server $\mathcal{S}$ share a secret key $\mathbf{S}$. $\mathcal{T}$ requires to prove its status to $\mathcal{S}$. Either $\mathcal{T}$ or $\mathcal{S}$ knows the authentication is successful. Assume that $\mathcal{S}$ knows the authentication result, while $\mathcal{T}$ is unaware and desynchronization occurs. In order to address this problem, we propose to preserve the session key ($\mathbf{S_{i-1}''}$) used in the previous successful authentication session in the server. If the server fails to match the current index $\mathbf{I}_i$, and consequently the current session key ($\mathbf{S_i''}$), it will try with ($\mathbf{S_{i-1}''}$). □

However, the adversary cannot execute the same attack twice consecutively. First, the tag could not be totally desynchronized, since it has another permanent key ($\mathbf{S}'$). Secondly, the server immediately re-synchronizes the key in the next consecutive session by *brute-force* searching (seek and match the previous session key $\mathbf{S_{i-1}''}$).

5 Privacy

Privacy models for RFID authentication protocols can be divided into four categories indistinguishability (IND) based privacy, unpredictability based privacy, Zero-Knowledge (ZK) based privacy, simulation (SIM) based privacy, universal composability based privacy. Relationship among them has been addressed [28] such as ZK-privacy is equivalent to IND-privacy. If the protocol message is not publicly verifiable, IND-privacy is also equivalent to SIM-privacy.

Since LPN-based authentication uses shared secret key (not public verifiable) and protocol adapts mutual authentication, we analyzed our protocol based on ZK-privacy framework described in [24]. We assume that the protocol is always initiated by the reader, transaction messages do not disclose any secret, and the protocol produces $\pi \leftarrow 2\lambda+1$ ($\pi = 3$ s.t $\lambda = 1$) transaction rounds.

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 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. $\hat{\mathcal{A}}$ outputs an arbitrary tags $C \subseteq \hat{\mathcal{T}}$ called *clean tags*. Let $\hat{\mathcal{A}}$ be composed of a pair of adversaries $(\hat{\mathcal{A}}_1, \hat{\mathcal{A}}_2)$ and their corresponding simulators $(\mathsf{Sim}_1, \mathsf{Sim}_2)$ for $\mathbf{Exp}^{\mathsf{ZK}}_{\hat{\mathcal{A}}}(\hat{T})$ experiments.

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

- Initialize RFID system, the reader $\mathcal{R}$, the tag set $\hat{\mathcal{T}}$ (s.t., $|\hat{\mathcal{T}}| = l$) by **SetupTag**$(\cdot)$
- let $\mathcal{O} \leftarrow$ Launch, Dtag, STag, SReader, Ukey, Corrupt
- Real: $(\mathcal{T}, st) \leftarrow \hat{\mathcal{A}}_1^{\mathbf{DTag}}(\mathcal{R}, \hat{\mathcal{T}}, \mathsf{Pub}_T)$
  Simulation: $(\mathcal{T}, st) \leftarrow \mathsf{Sim}_1^{\mathbf{DTag}}(\mathcal{R}, \hat{\mathcal{T}}, \mathsf{Pub}_T)$ 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: $sview \leftarrow \mathsf{Sim}_2^{\mathcal{O}}(\mathcal{R}, \hat{\mathcal{T}}, st)$
- Real: output $(c, view_{\hat{A}})$ Simulation: output $(c, sview_{Sim})$

We assume that $\hat{\mathcal{A}}$ queries the challenger with $\mathbf{Exp}^{\mathsf{ZK}}(\hat{\mathcal{T}})$ in the *real* 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 ZK-**privacy.** RFID Authentication protocol described in Fig. 1 satisfies the ZK-privacy model if for any PPT adversary $\hat{\mathcal{A}}$ (resp. PPT simulator Sim), $\mathbf{Adv}^{\mathsf{ZK}}_{\hat{\mathcal{A}}}(\kappa, \hat{T})$ is negligible.

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

Theorem 4 *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 (st) of a challenged tag T_c from the initial (k^0, st^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**$(\cdot)$ is run by $\hat{\mathcal{A}}_2$. Note that T_c should not be in the oracle table D (related to **DTag**$(\cdot)$) before the experiment $\mathbf{Exp}^{\mathsf{ZK}}(\hat{T})$ ends. However, *forward* (resp., *backward*)-ZK privacy cannot be achieved if $\mathcal{A}$ has corrupted the challenging tag T_c before the experiment finishes. □

6 Evaluation

As the back-end server is scalable and has sufficient computation and storage resources, most of the expensive computations of the protocol has been held at the server site. Tags are the most resource constraint and computationally weak entity in the system. Therefore, we focus mainly on the computation and storage of RFID tags. In order for a cryptographic scheme to be eligible for RFID tag, it must be executable into a tag, computationally feasible, having less storage requirement and communication cost. HB-family protocol for RFID tag identification is well-known for it's simplicity, effectiveness, and lightweight compared to most other authentication schemes. LPN-problem, the foundation of HB-family protocol is NP-hard and a low cost solution for authentication. The security of the problem instances depends on the key length and the noise level. This paper proposes a new mathematical structures that supports LPN problem to achieve more functionality and stronger authentication scheme. We analyze the performance of our protocol by comparing with some well-known HB-family protocols. Table 2 presents a comparative study between our protocol and several popular HB-family protocols based on some common attributes such as storage consumption, major computations, authentication party, achieved security, approximate hardware cost etc.

Computation Requirement: Major protocol operations on the tag include *two* LPN, *two* binary matrix multiplications for signature generation and updating session key. A very recent work in [34], authors claim solving an LPN instance with complexity less than 2^{80} operations is in expectation. This is why they recommend the HB variants to retain the minimum security parameter size to 80-bit. Authors in [35] presented an LPN-based scheme similar to us using Verilog HDL and Virtex-5 FPGA board for hardware design, and Xilinx ISE 14.5 for implementation. Assuming our scheme achieve the same 80-bit security as [35], choosing public parameters' $(\tau, l, v, k, \tau', e) := (0.08, 80, 83, 163, 0.29, 13)$ would require .53 *W* as *power* and 3.789 *ns* as *execution time*. More clearly we assume an LPN instance of dimension $k = 163$, where we allow at most 2^{80} received samples, that is, at most around 2^{80} vectors of length 163 to be stored in tag memory. The upper-bounded Bernoulli noise is $\tau = .08$.

In [23], authors present an efficient matrix multiplication and vector addition for lattice-based cryptography. Using ARM NEON intrinsic functions and Raspberry Pi 3 Model B as hardware, they described the fastest method and time for multiplying two matrices $(\mathbb{Z}_2^{m\times n} \times \mathbb{Z}_2^{n\times l})$ is 93.91 *ms* where $(m, n, l) := (1024, 536, 256)$.

Storage Requirement: Tag stores two secret keys and an index for each session which requires $(k \cdot l + k \cdot v + k)$ bits. A reader needs $(k^2 + l)$ bits to store a tag identifier and one secret key.

Table 2 Tag resources and security comparison with HB family

Scheme	Storage	Computation	Authentication (major)	Security achieved	Hardware (gates)
HB [8]	1 S	1 LPN	tag	7	≈ 1600
HB$^+$ [6]	2 S	1 LPN	tag	0,7	≈ 1600
HB-MP [11]	2 S	1 LPN	tag	0,5,6, 7	≈ 1600
HB-MP$^+$ [19]	2 S	1 LPN, 1 HASH	tag	0,1,5,6,7	≈ 3500
GHB# [25]	2 S	1 LPN	tag	0,1,5,6,7	≈ 1600
F-HB [17]	1 I , 1 S	1 PRNG,2 LPN	mutual	0, 1, 2, 4*, 5, 6, 7	≈ 3500
SLPN [21]	1 I, 1 S	1 SLPN,1 P	mutual	0,1,2,3,4†,5,6,7	≈ 1600
Tree-HB$^+$ [16]	2S, BM	2 LPN	mutual	0,1,2,3†,5,6,7	≈ 2000
ours	1 I, 2 S	2 LPNx, 1 LS	multi-party	0,1,2,3,4,5,6,7	≈ 2000

SC:= Stream Cipher; S:= Secret key; I:= Index; LS:= lightweight signature; PRNG:= Pseudo Random Number Generator; BM:= Binary Matrix P:= Pseudo Inverse Matrix; LPN:= Learning parity from noise SLPN:= Subset LPN; LPNx := exact LPN Security attributes: Active attack (0), MIM attack (1), Forward Security (2), Reduced Backward Security (3), Backward Security (3†), ZK-privacy (4), IND-privacy (4†), UNP-privacy (4*) Tag tracking (5), De-synchronization (6), Replay attack (7)

Storage requirement for the tag and reader can be expressed by $\mathcal{O}(1)$ and $\mathcal{O}(n)$ where n is the number of tags in an RFID system.

Communication Complexity : We estimate the total amount of messages sent in our protocol to approximate average transmission time. The protocol requires $(k^2+2k+4v)$ bits in the tag-reader communication and $(2k^2+2k+3v)$ bits in the reader-server communication. The ISO 14443 standard that specify transmission protocols for proximity cards for identification defines bit rate as 106 kbps to 848 kbps for the reader to card communication. Based on this fact, the required transmission time for reader-tag communication would be *0.31* seconds considering protocol message 27.1k bit where ($k = 163$, $v = 80$) to achieve 80-bit security.

7 Conclusion

Authentication protocols using postquantum cryptographic system such HB-family protocols are presently developed to deal with the security threats in the presence of quantum computing technology. This paper presents a HB-like hardware-friendly RFID authentication protocol based on LPN-based commitment scheme and key-evolving signature system. Such operations are proven fairly lightweight to be implemented into low cost RFID tags. In comparison to other HB-family protocols, it achieves the desired security and privacy properties and uniquely multi-party authentication properties from LPN assumption. Proposed authentication framework can suitably be applied to any RFID-based IoT solutions such as smart home/hospital, smart cloud, ownership transfer, supply chain management.

References

1. Wu F, Xu L, Kumari S, Li X, Das AK, Shen J (2018) A lightweight and anonymous RFID tag authentication protocol with cloud assistance for e-healthcare applications. J Ambient Intell Human Comput 9(4):919–930
2. Zheng L, Xue Y, Zhang L, Zhang R (2017) Mutual Authentication Protocol for RFID based on ECC. In: IEEE International Conference on Computational Science and Engineering (CSE) and IEEE International Conference on Embedded and Ubiquitous Computing (EUC), vol 2. IEEE, pp 320–323
3. Gope P, Lee J, Quek TQS (2018) Lightweight and practical anonymous authentication protocol for RFID systems using physically unclonable functions. IEEE Trans Inf Forensic Secur 13.11:2831–2843
4. BD D, Al-Turjman F, Mostarda L (2019) A Hash-Based RFID Authentication Mechanism for Context-Aware Management in IoT-Based Multimedia Systems. Sensors 19.18:3821
5. Welbourne E, Battle L, Cole G, Gould K, Rector K, Raymer S, Balazinska M, Borriello G (2009) Building the internet of things using RFID: the RFID ecosystem experience. IEEE Internet Comput 13(3):48–55
6. Juels A, Weis SA (2005) Authenticating pervasive devices with human protocols. In: Annual international cryptology conference. Springer, Berlin, pp 293–308
7. Batina L et al (2006) An elliptic curve processor suitable for RFID tags. International Association for Cryptologic Research ePrint Archive
8. Hopper NJ, Blum M (2001) Secure human identification protocols. In: International conference on the theory and application of cryptology and information security. Springer, Berlin, pp 52–66
9. Gilbert H, Robshaw M, Sibert H (2005) An active attack against HB+ - a provably secure lightweight authentication protocol: An active attack against HB+ - a provably secure lightweight authentication. Cryptology ePrint Archive Report 2005/237
10. Bringer J, Chabanne H, Dottax E (2006) HB++: A lightweight authentication protocol secure against some attacks. In: SecPerU, pp 28–33
11. Munilla J, Alberto P (2007) HB-MP A Further step in the HB-family of lightweight authentication protocols. Comput Netw 51(9):2262–2267
12. Gilbert H, Robshaw MJB, Seurin Y (2008) Good variants of HB+ are hard to find. In: Tsudik G (ed) FC 2008, volume 5143 of LNCS. Springer, pp 156–170
13. Ouafi K, Overbeck R, Vaudenay S (2008) On the security of HB# against a man-in-the-middle attack. In: International Conference on the Theory and Application of Cryptology and Information Security. Springer, Berlin, pp 108–124
14. Nguyen TD, Pham HTB, Van HD (2010) An efficient Pseudo Inverse matrix-based solution for secure auditing. In: IEEE RIVF International Conference on Computing & Communication Technologies, Research, Innovation, and Vision for the Future (RIVF). IEEE, pp 1–6
15. MSI Mamun, Miyaji A (2014) A privacy-preserving efficient RFID authentication protocol from SLPN assumption. International Journal of Computational Science and Engineering (IJCSE), Special Issue on Converged Networks, Technologies and Applications, Inderscience Publishers, vol 9
16. Li X, Xu J, Zhang Z (2019) Revisiting the Security of Qian others.'s Revised tree-LSHB$^+$ Protocol. Wirel Person Commun 106(2):321–343
17. Cao X, O'Neill M (2011) F-HB: An efficient forward private protocol. In: 2011 Workshop on Lightweight Security & Privacy: Devices, Protocols, and Applications. IEEE, pp 53–60
18. Billet O, Etrog J, Gilbert H (2010) Lightweight privacy preserving authentication for RFID using a stream cipher. In: International Workshop on Fast Software Encryption. Springer, Berlin, pp 55–74
19. Leng X, Mayes K, Markantonakis K (2008) HB-MP+ protocol: An improvement on the HB-MP protocol. In: 2008 IEEE international conference on RFID. IEEE, pp 118–124
20. Applebaum B, Ishai Y, Kushilevitz E (2009) Cryptography with constant input locality. J Cryptol 22(4):429–469
21. Mamun MSI, Miyaji A, Rahman M (2012) A Secure and Private RFID Authentication Protocol under SLPN Problem. NSS2012, LNCS 7645, pp 476–489
22. Jain A, Krenn S, Pietrzak K, Tentes A (2012) Commitments and Efficient Zero-Knowledge Proofs from Learning Parity with Noise. ASIACRYPT 2012, LNCS 7658, pp 663–680
23. Park T, Seo H, Kim J, Park H, Kim H (2018) Efficient parallel implementation of matrix multiplication for Lattice-Based cryptography on modern ARM processor. Security and Communication Networks

24. Deng RH, Li Y, Yung M, Zhao Y (2010) A new framework for RFID privacy. In: European Symposium on Research in Computer Security. Springer, Berlin, pp 1–18
25. Rizomiliotis P, Gritzalis S (2012) GHB#: A provably secure HB-like lightweight authentication protocol. In: International Conference on Applied Cryptography and Network Security. Springer, Berlin, pp 489–506
26. Micciancio D, Mol P (2011) Pseudorandom Knapsacks and the Sample Complexity of LWE Search-to-Decision Reductions. CRYPTO 2011, volume 6841 of LNCS. Springer, pp 465–484
27. Katz J, Shin JS, Smith A (2010) Parallel and concurrent security of the HB and HB+ protocols. J Cryptol 23(3):402–421
28. Moriyama D, Matsuo SI, Ohkubo M (2012) Relations among notions of privacy for RFID authentication protocols. In: Computer Security– ESORICS. Springer LNCS, pp 661–678
29. IoT and RFID: What's the Connection?. RFID journal expert views. https://rfidjournal.com/iot-and-rfid-whats-the-connection, Accessed 7 July 2020
30. The Role of RFID Technology and Internet of Things in Healthcare. Digitium IoT Software, https://digiteum.com/rfid-technology-internet-of-things. Accessed 10 July 2020
31. Jo S, Lee J, Han J, Ghose S (2020) P2P Computing for intelligence of things. Peer-to-Peer Netw Appl 13(2):575–578
32. Get smarter about the many benefits of P2P IoT. Nabto Inc. P2P IoT solution. https://www.nabto.com/p2p-explainer/, Accessed 10 July 2020
33. He XU, WANG SP, WANG RC, WANG ZQ (2011) Efficient P2P-based mutual authentication protocol for RFID system security of EPC network using asymmetric encryption algorithm. J China Univ Posts Telecommun 18:40–47
34. Guo Q, Johansson T (2020) Carl löndahl Solving LPN Using Covering Codes. J Cryptol 33.1:1–33
35. Bagadia K, Urbi Chatterjee DBR, Mukhopadhyay D, Chakraborty RS (2019) Revisiting the Security of LPN Based RFID Authentication Protocol and Potential Exploits in Hardware Implementations. In: International Conference on Security, Privacy, and Applied Cryptography Engineering. Springer, Cham, pp 214–230

Mohammad Mamun Associate Research officer, Cybersecurity National Research Council of Canada / Government of Canada.

Atsuko Miyaji Graduate School of Engineering, Osaka University, Division of Electrical, Electronic and Information Engineering, Department of Information and Communications Technology, Cyber Security Engineering area.

Rongxing Luv IEEE Senior Member, Vice-Chair (Publication) of IEEE ComSoc CIS-TC (Communications and Information Security Technical Committee), Associate Professor, Faculty of Computer Science.

Chunhua Su Information Security Laboratory Division of Computer Science.