Cryptanalysis of Izza et al.'s Protocol: An Enhanced Scalable and Secure RFID Authentication Protocol for WBAN Within An IoT Environment

Atakan Arslan[a,*], Muhammed Ali Bingöl[b]

[a] *TÜBİTAK BİLGEM-UEKAE, Gebze, Kocaeli, Turkey*
[b] *De Montfort University University, Cyber Technology Institute, Leicester, UK*

Abstract

Most recently, Izza et al. propose a new ECC-based RFID authentication protocol by showing the vulnerabilities of Naeem's protocol. They claim that their scheme provides security and privacy. However, we assert that their protocol does not satisfy privacy including anonymity, untraceability, forward and backward secrecy on the contrary of their claim. We also argue that the scheme suffers from availability problems.

Keywords: RFID, Protocol, Privacy, Security, ECC

1. Introduction

Security and privacy concerns are becoming more serious in our daily life with internet of things (IoT) paradigm. Rapid development of technology and getting cheaper mobile devices are accompanying IoT. Today everybody has become sensitive to their privacy much more than before and people's lives are getting more digitalized day by day. In the future, security and privacy surely will be still one of the essential concerns in the digitized age.

Radio Frequency Identification (RFID) is quite popular technology in IoT and has many application areas in everyday life such as healthcare, payment, access control, supply chain management, etc. systems. Nowadays, many RFID authentication protocols have been proposed to mitigate the security and privacy issues by using Elliptic Curve Cryptography (ECC) [2, 3, 4, 5, 6, 7, 8]. Very recently, Izza et al. [1] propose an ECC-based RFID authentication protocol for especially Wireless Body Area Networks (WBANs) to protect the patients' private information. They point out that the scalability, security and privacy problems of Naeem et al.'s scheme [6]. They attack their scheme and show its

*Corresponding author. E-mail: atknarsln@gmail.com
Email addresses: atknarsln@gmail.com (Atakan Arslan), muhammed.bingol@dmu.ac.uk (Muhammed Ali Bingöl)

security and privacy vulnerabilities. Izza et al. extend and enhance the Naeem et al.'s scheme.

Izza et al. [1] claim that their improved scheme achieves both scalability, security and privacy requirements for RFID systems. They present the security analysis of their protocol and state that their scheme provides tag anonymity, untraceability, backward and forward secrecy in their paper. However, we realize that their scheme does not satisfy these privacy properties. Therefore, we will show the vulnerabilities of their scheme. Moreover, we claim that their scheme does not also achieve availability due to suffering from synchronization issues. In this paper, we also enhance the protocol by proposing solutions to overcome the privacy weaknesses.

The organization of this paper is as follows. In Section 2, description of Izza et al.'s protocol will be present. In Section 3, the security and privacy vulnerabilities of Izza et al.'s protocol will be shown. In this section, the enhancements to provide security and privacy by mitigating the vulnerabilities, and will be explained in detail. Finally, Section 4 will conclude the paper.

2. Izza et al.'s Protocol Description

We present the overview of Izza et al.'s scheme (IBD21) in Figure 2 and Figure 3. In IBD21, there are three major phases: (i) initialization and registration phase, (ii) authentication phase, (iii) digital signature and data transmission phase [1]. In the initialization and registration phase, the registrations of users, tags, reader and the medical server (MS) are completed (see Figure 2). In authentication and data transmission phases depicted in Figure 3, the tags, the reader and MS mutually authenticate with each others and then, the data of tag is exchanged. According to IBD21, the channels between tag-reader and reader-MS is insecure. We stick to IBD21 notations to avoid from the possible confusions (see in Figure 1).

Notation	Meaning
P	Elliptic curve base point
$h(.)$	One-way hashing function
NM	Network manager
U_j	Network users
y_j	User public key
x_j	User secret key
α	NM secret key
β	NM public key
P_{r_R}/P_{r_S}	Reader / server private key
P_{u_R}/P_{u_S}	Reader / server public key
n	Number of tags
ID_{T_i}	The ith tag identity
PID_{T_i}	The ith tag's pseudo identity
ID_R	The reader's identity
PID_R	The reader's pseudo identity
ID_S	The server's identity
SK_{TR}/SK_{RT}	Shared session key between Tag and Reader
$E_{SK}(.)/D_{SK}(.)$	Message Encryption / Decryption
m_i	Message from the ith tag
$(.)_x$	The x coordinate of a given point

Figure 1: The notations of IB21 [1].

The reader starts the authentication phase with transmitting a nonce R_{r_1} to the tag T_i. When the tag receives the nonce, the tag firstly picks a random number t_1 and computes C_1 and R_{t_1}. Then the tag initializes $PID_{T_{inew}} = h\left(PID_{T_{iold}}||init\right)$ and calculates C_2. The tag sends the messages $[C_1, C_2, T_1]$ to the reader back, where T_1 denotes the current timestamp.

Upon receiving the messages, the reader first checks the elapsed time. If the elapsed time is smaller than ΔT, the reader does not abort the session. Later, the reader extracts the pseudo identifier of the tag, $PID_{T_{inew}}$ by using its private key and searches in its own database. If the reader finds, the tag is authenticated. Later on, the reader communicates with the MS. In this communication, the reader computes the message N_2 with initializing $PID_{R_{new}} = h\left(PID_{R_{old}}||init\right)$, where $init$ is a random value selected by the MS and $init$ is also inserted in both memory of the reader and the tag in initialization phase.

The messages $[N_2, R_{r_1}, T_2]$ is sent by the reader. The MS responses with the messages $[N_3, S_1, T_3]$ to the reader after authenticating the reader. The reader takes the messages checks the time interval and authenticates the MS. After the successful authentication, the reader computes the message $C_3 = h\left(ID_{T_i}, T_3, T_4\right) + PID_{R_{new}}$ by using the previous initialization of the pseudo identifier $PID_{R_{new}} = h\left(PID_{R_{old}}||init\right)$ and computes the message C_4.

After all, the reader sends the messages $[C_3, C_4, T_3, T_4]$ and updates the pseudo identifiers of the tag and itself. Since tag receives the messages of the reader, the tag also verifies the time interval and authenticates the reader. The authentication phase is completed with generating an ephemeral shared session key S_{TR} by the tag.

In the data transmission phase, the tag encrypts a message m_i with S_{TR}, and transmits the messages $[m_i, T_5]$ to the reader. Once the reader gets the messages, the reader obtains m_i with using its own session key S_{RT}. Later on, the reader shares the same message with the MS by using elliptic curve digital signature with message recovery (ECDSMR) mechanism.

We present the updating process of PID_R parameters on the MS for IBD21 during protocol sessions in Figure 4 to clarify our attacks. Figure 4 depicts how to updating the parameters session by session. The reader and the MS have ${}^{s_0}PID_{R_{old}}$ and ${}^{s_0}PID_{R_{new}}$ identifiers after the initialization phase (the initial session s_0). After s_0, the entities update the identifiers with status of the previous session synchronization. For instance, if the synchronization is provided, the related entity executes the computations shown in Figure 4 with respect to the statement "$sync$".

User U_j (Reader/Server)	Secure channel	Network Manager NM
Selects $c_j \in [1, n-1]$ calculates $d_j = c_j P$		
	$\xrightarrow{d_j, ID_j}$	Chooses $k_j \in [1, n-1]$ and computes $y_j = k_j P + d_j$ $z_j = k_j + ((y_j)_x + ID_j)\alpha \mod n$
computes $x_j = z_j + c_j \mod n$ checks $x_j.P = y_j + ((y_j)_x + ID_j)\beta$	$\xleftarrow{y_j, z_j}$	

Figure 2: Initialization phase of IB21 [1].

Tag $P_{u_R}, ID_{T_i}, PID_{T_{iold}}, n, P, P_{u_S}$	Insecure channel	Reader $P_{r_R}, P_{u_R}, ID_{T_i}, PID_{T_i old}, PID_{T_i new}$, $ID_R, PID_{R_{old}}, ID_S, n, P, P_{u_S}$	Insecure channel	Medical Server $P_{u_R}, ID_R, ID_S, PID_{R_{old}}, PID_{R_{new}}$, n, P, P_{r_S}, P_{u_S}
2- Generates t_1 and computes $C_1 = t_1.P$ $R_{t_1} = t_1.P_{u_R}$	$\xleftarrow{R_{r_1}}$	**1-** Generates r_1 and computes $R_{r_1} = r_1.P$		
Initializes $PID_{T_{inew}} = h(PID_{T_{iold}} \| init)$ $C_2 = PID_{T_{inew}} + h((R_{t_1})_x \| (R_{r_1})_x \| (C_1)_x \| T_1)$	$\xrightarrow{C_1, C_2, T_1}$	**3-** $\|T_2 - T_1\| < \Delta T$ Calculates $R^*_{t_1} = C_1.P_{r_R}$ $PID^*_{T_i} = C_2 - h((R^*_{t_1})_x \| (R_{r_1})_x \| (C_1)_x \| T_1)$ Checks $PID^*_{T_i}$ in database		
			$\xrightarrow{N_2, R_{r_1}, T_2}$	**4-** $\|T_3 - T_2\| < \Delta T$ Computes $N^*_1 = R_{r_1}.P_{r_S}$ $PID^*_R = N_2 - h((R_{r_1})_x \| ID_R \| (N^*_1)_x \| T_2)$ Checks PID^*_R in database Generates s_1 and computes $S_1 = s_1.P$ Calculates $R_{s_1} = s_1.P_{u_R}$ $N_3 = h((R_{s_1})_x \| PID^*_R \| T_2 \| T_3) + ID_S$ If $PID^*_R = PID_{R_{old}}$ Updates $\begin{cases} PID_{R_{old}} \leftarrow PID_{R_{old}}, \\ PID_{R_{new}} \leftarrow h(PID_{R_{old}} \| (N_1)_x) \end{cases}$
		5- $\|T_4 - T_3\| < \Delta T$ $R^*_{s_1} = S_1.P_{r_R}$ Calculates $ID^*_S = N_3 - h((R^*_{s_1})_x \| PID_{R_{new}} \| T_2 \| T_3)$ Checks ID^*_S in database Computes $C_3 = h(ID_{T_i} \| T_3 \| T_4) + PID_{R_{new}}$ $C_4 = h((R^*_{t_1})_x \| PID_{R_{new}} \| (R_{r_1})_x \| T_4)$ If $PID^*_{T_i} = PID_{T_{iold}}$ Updates $\begin{cases} PID_{T_{iold}} \leftarrow PID_{T_{iold}}, \\ PID_{T_{inew}} \leftarrow h(PID_{T_{iold}} \| (R_{t_1})_x) \end{cases}$ Else if $PID^*_{T_i} = PID_{T_{inew}}$ Updates $\begin{cases} PID_{T_{iold}} \leftarrow PID_{T_{inew}}, \\ PID_{T_{inew}} \leftarrow h(PID_{T_{inew}} \| (R_{t_1})_x) \end{cases}$ Updates $PID_{R_{new}} \leftarrow h(PID_{R_{new}} \| (N_1)_x)$ *Generates the shared session key* $SK_{RT} = h(ID_{T_i} \| PID_{T_{inew}} \| (r_1.C_1)_x)$	$\xleftarrow{T_3, N_3, S_1}$	Else if $PID^*_R = PID_{R_{new}}$ Updates $\begin{cases} PID_{R_{old}} \leftarrow PID_{R_{new}}, \\ PID_{R_{new}} \leftarrow h(PID_{R_{new}} \| (N_1)_x) \end{cases}$
6- $\|T_5 - T_4\| < \Delta T$ $PID^*_{R_{new}} = C_3 - h(ID_{T_i} \| T_3 \| T_4)$ $C^*_4 = h((R_{t_1})_x \| PID^*_{R_{new}} \| (R_{r_1})_x \| T_4)$ $C^*_4 \stackrel{?}{=} C_4$ Updates $PID_{T_{inew}} \leftarrow h(PID_{T_{inew}} \| (R_{t_1})_x)$ *Generates the shared session key* $SK_{TR} = h(ID_{T_i} \| PID_{T_{inew}} \| (t_1.R_{r_1})_x)$	$\xleftarrow{C_4, C_3, T_3, T_4}$			
7-Generates a message m_i $M_i = E_{SK}(m_i)$	$\xrightarrow{M_i, T_5}$	**8-** $\|T_6 - T_5\| < \Delta T$ $m_i = D_{SK}(M_i)$ Makes $r_0 = 0$ Selects a random number $k \in [1, n-1]$ Calculates $r_i = m_i + h(r_{i-1} \oplus (k(y_S + ((y_S)_x + ID_S)\beta))_x) \mod n$ $r = h(r_1 \| r_2 \| r_3 \| ... \| r_n)$ $z = k - rx_R \mod n$	$\xrightarrow{r, z, r_1, r_2, r_3, ..., r_n, T_6}$	**9-** $\|T_7 - T_6\| < \Delta T$ $r^* = h(r_1 \| r_2 \| r_3 \| ... \| r_n)$ $r \stackrel{?}{=} r^*$ $m_i = r_i - h(r_{i-1} \oplus (zP + r(y_R + ((y_S)_x + ID_R)\beta)x_S)_x) \mod n$

Figure 3: Authentication and data transmission phase of IB21 [1].

sessions	$PID_{R_{old}}$	$PID_{R_{new}}$
initial: s_0	${}^{s_0}PID_{R_{old}}$	${}^{s_0}PID_{R_{old}}$
s_1	sync: ${}^{s_1}PID_{R_{new}} = {}^{s_0}PID_{R_{new}}$ desync: ${}^{s_1}PID_{R_{new}} = {}^{s_0}PID_{R_{old}}$	${}^{s_1}PID_{R_{new}} = h({}^{s_0}PID_{R_{old}}\|\|init)$ sync: ${}^{s_1}PID_{R_{new}} = h({}^{s_1}PID_{R_{new}}\|\|({}^{s_1}R_{t_1})_x)$ desync: ${}^{s_1}PID_{R_{new}} = h({}^{s_0}PID_{R_{old}}\|\|({}^{s_1}R_{t_1})_x)$
s_2	sync: ${}^{s_2}PID_{R_{new}} = {}^{s_1}PID_{R_{new}}$ desync: ${}^{s_2}PID_{R_{new}} = {}^{s_1}PID_{R_{old}}$	${}^{s_2}PID_{R_{new}} = h({}^{s_1}PID_{R_{old}}\|\|init)$ sync: ${}^{s_2}PID_{R_{new}} = h({}^{s_2}PID_{R_{new}}\|\|({}^{s_2}R_{t_1})_x)$ desync: ${}^{s_2}PID_{R_{new}} = h({}^{s_1}PID_{R_{old}}\|\|({}^{s_2}R_{t_1})_x)$
...	...	...
s_j	sync: ${}^{s_j}PID_{R_{new}} = {}^{s_{j-1}}PID_{R_{new}}$ desync: ${}^{s_j}PID_{R_{new}} = {}^{s_{j-1}}PID_{R_{old}}$	${}^{s_j}PID_{R_{new}} = h({}^{s_{j-1}}PID_{R_{old}}\|\|init)$ sync: ${}^{s_j}PID_{R_{new}} = h({}^{s_j}PID_{R_{new}}\|\|({}^{s_j}R_{t_1})_x)$ desync: ${}^{s_j}PID_{R_{new}} = h({}^{s_{j-1}}PID_{R_{old}}\|\|({}^{s_j}R_{t_1})_x)$
...	...	...

Figure 4: The updating process of PID_R parameter on the reader for IBD21.

3. Vulnerabilities of Izza et al.'s Protocol

Izza et al. claim that IBD21 provides backward and forward secrecy property. If an RFID scheme provides backward and forward secrecy, or sometimes called privacy, it means that an adversary cannot distinguish a tag with using future or previous protocol transactions even though she knows all data stored in the tag. In other words, the adversary obtains whole internal knowledge of a tag but she cannot trace the tag and ruin the privacy. Therefore, it can be said that the protocol satisfies backward and forward privacy/secrecy property. Izza et al. also claim that IBD21 provides achieve availability properties. However, we realize that IBD21 needs small amendments to provide synchronizations. Moreover, we present forward/backward secrecy attacks under the assumption that IBD21 is resistant to synchronization problems. In addition to this, we show the vulnerabilities Izza et al.'s protocol in terms of tag anonymity and untraceability contrarily their claim. Finally, in this section we propose some enhancements to mitigate their vulnerabilities.

3.1. Forward/Backward Secrecy Attacks On Izza et al.'s Protocol

Both *backward* and *forward* privacy are the essential security requirements for an RFID scheme [9]. In the RFID literature, *forward privacy* and *backward privacy* property are sometimes called *backward untraceability* and *forward untraceability*, respectively. The notion forward and backward privacy imply the untraceability of a legitimate tag in an RFID system by an adversary with helping of aforementioned tag information. An adversary can obtain the internal data of a legitimate tag via several ways [9]: tampering/corrupting, having ownership transfer of the tag, etc.

We present below definitions to clearly explain our attacks in the same language with the literature.

Let $f_{Adv}\left(\phi_{t_i}^{\boldsymbol{T}}, \phi^{public}, \boldsymbol{sp}_t\right) \rightarrow out$ be function that takes whole internal knowledge $\phi_{t_i}^{\boldsymbol{T}}$ (e.g. identity numbers, secret keys, public keys) of a legitimate tag $\boldsymbol{T}$ at time t_i, public known parameters ϕ^{public} of an RFID scheme and the set of valid session parameters $\boldsymbol{sp}_t$ of all executed sessions in the scheme until the time

t as inputs and outputs the probability of $\boldsymbol{Adv}$ to successfully trace $\boldsymbol{T}$, where $0 \leq out \leq 1$.

Definition 3.1. ***(Backward Untraceability / Forward Privacy).*** *An RFID scheme provides* backward untraceability *property, if* $f_{Adv}\left(\phi_{t_i}^{\boldsymbol{T}}, \phi^{public}, \boldsymbol{sp}_t\right)$ *is negligible for all probabilistic polynomial time (PPT)* $\boldsymbol{Adv}$*, where* $t < t_i$*.*

Definition 3.2. ***(Forward Untraceability / Backward Privacy).*** *An RFID scheme provides* Forward untraceability *property, if* $f_{Adv}\left(\phi_{t_i}^{\boldsymbol{T}}, \phi^{public}, \boldsymbol{sp}_t\right)$ *is negligible for all probabilistic polynomial time (PPT)* $\boldsymbol{Adv}$*, where* $t > t_i$*.*

To show our attacks on IBD21, let a PPT $\boldsymbol{Adv}$ attacks on the following simple architecture of Izza et al.'s RFID system. Let we say that there are two legitimate tags called $\boldsymbol{T_a}$ and $\boldsymbol{T_b}$ and a legitimate reader $\boldsymbol{R}$ in this system. $\boldsymbol{R}$ executes several sessions with randomly selected a tag $\boldsymbol{T_\gamma}$ in a time interval, where $\gamma \in_R \{a, b\}$ and $Pr(\gamma = a) = Pr(\gamma = b)$. $\boldsymbol{Adv}$ can eavesdrop the session parameters transmitted within each IBD21 transactions. Izza et al. claim that IBD21 provides forwards and backward secrecy. If the authors' claim is valid, $\boldsymbol{Adv}$ never distinguish the tags by using the scheme session parameters, although she obtains the whole internal knowledge of only one tag. Formally, the adversary can perform the following attack.

Theorem 3.1. *IBD21 does not provide forward secrecy.*

Proof. Let Adversary $\boldsymbol{Adv}$ plays a security game as below.

1. $\boldsymbol{Adv}$ records the parameters of two consecutive protocol sessions s_j, s_{j+1} executing between the reader $\boldsymbol{R}$ and $\boldsymbol{T_\gamma}$.
   (a) s_j, s_{j+1} include the following set of protocol transaction parameters: $\boldsymbol{sp} : [R_{r_1}, C_1, C_2, C_3, C_4, T_1, T_2, T_3, T_4, T_5, N_2, N_3, S_1, M_i]$**.**
   (b) Let ${}^{s_j}C_1$ denotes the parameter C_1 of the session s_j for $\boldsymbol{T_\gamma}$.
2. Later on, $\boldsymbol{Adv}$ arbitrarily selects a tag, called $\boldsymbol{T_a}$ and obtains the internal knowledge of $\boldsymbol{T_a}$, called $\phi^{\boldsymbol{T_a}}$.
   Hence, $\boldsymbol{Adv}$ knows $\phi^{\boldsymbol{T_a}}$: $\left[ID_{T_a}, PID_{T_{a^{old}}}, n, P, P_{u_R}, P_{u_S}, init\right]$.
3. $\boldsymbol{Adv}$ calculates $\xi_j : {}^{s_j}C_3 - h\left(ID_{T_a} || {}^{s_j}T_3 || {}^{s_j}T_4\right)$ as ${}^{s_j}PID_{R_{a^{new}}}$ and $\xi_{j+1} : {}^{s_{j+1}}C_3 - h\left(ID_{T_a} || {}^{s_{j+1}}T_3 || {}^{s_{j+1}}T_4\right)$ as ${}^{s_{j+1}}PID_{R_{a^{new}}}$.
4. $\boldsymbol{Adv}$ knows that ${}^{s_{j+1}}PID_{R_{a^{old}}} = {}^{s_j}PID_{R_{a^{new}}}$ from the scheme description and computes
   ${}^{s_{j+1}}PID_{R_{a^{new}}} = h\left({}^{s_j}PID_{R_{a^{new}}} || init\right)$.
5. Therefore, $\boldsymbol{Adv}$ checks $\xi_{j+1} \stackrel{?}{=} h\left(\xi_j || init\right)$. If the verification is succeeded, $\boldsymbol{Adv}$ claims that $\boldsymbol{T_\gamma}$=$\boldsymbol{T_a}$ else she claims that $\boldsymbol{T_\gamma}$=$\boldsymbol{T_b}$.

The success probability of this adversary is 1 and she wins the game. This means that $\boldsymbol{Adv}$ has stored some past messages. When she gets the internal parameters of the tag, she can check the relationship of tag identity ID_T with the transmitted messages. Therefore, this scheme does not provide forward secrecy (backward untraceability).

□

Theorem 3.2. *IBD21 does not provide backward secrecy.*

Proof. Let Adversary ***Adv*** plays a security game as below.

1. ***Adv*** arbitrarily selects a tag, called $\boldsymbol{T_a}$ and obtains the internal knowledge of $\boldsymbol{T_a}$, called $\phi^{\boldsymbol{T_a}}$.***Adv*** frees $\boldsymbol{T_a}$.

   Hence, ***Adv*** knows $\phi^{\boldsymbol{T_a}}$: $\left[ID_{T_a}, PID_{T_{a^{old}}}, n, P, P_{u_R}, P_{u_S}, init\right]$.
2. Later on, ***Adv*** records the parameters of two consecutive protocol sessions s_j, s_{j+1} executing between the reader $\boldsymbol{R}$ and $\boldsymbol{T_\gamma}$.
   (a) s_j, s_{j+1} include the following set of protocol transaction parameters: $\boldsymbol{sp}: [R_{r_1}, C_1, C_2, C_3, C_4, T_1, T_2, T_3, T_4, T_5, N_2, N_3, S_1, M_i]$**.**
   (b) Let ${}^{s_j}C_1$ denotes the parameter C_1 of the session s_j for $\boldsymbol{T_\gamma}$.
3. ***Adv*** calculates $\xi_j : {}^{s_j}C_3 - h\left(ID_{T_a}||{}^{s_j}T_3||{}^{s_j}T_4\right)$ as ${}^{s_j}PID_{T_{a^{new}}}$ and $\xi_{j+1} : {}^{s_{j+1}}\ C_3 - h\left(ID_{T_a}||{}^{s_{j+1}}T_3||{}^{s_{j+1}}T_4\right)$ as ${}^{s_{j+1}}PID_{T_{a^{new}}}$.
4. ***Adv*** knows that ${}^{s_{j+1}}PID_{R_{a^{old}}} = {}^{s_j}PID_{R_{a^{new}}}$ from the scheme description and computes

   ${}^{s_{j+1}}PID_{R_{a^{new}}} = h\left({}^{s_j}PID_{R_{a^{new}}}||init\right)$.
5. Therefore, ***Adv*** checks $\xi_{j+1} \stackrel{?}{=} h\left(\xi_j||init\right)$. If the verification is succeeded, ***Adv*** claims that $\boldsymbol{T_\gamma}$=$\boldsymbol{T_a}$ else she claims that $\boldsymbol{T_\gamma}$=$\boldsymbol{T_b}$.

The success probability of this adversary is 1 and she wins the game. This means that ***Adv*** gets the internal parameters of the tag and then she records future sessions so she can check the relationship of obtained tag identity ID_T with the transmitted messages. Therefore, this scheme does not provide backward secrecy (forward untraceability).

□

3.1.1. Enhancements For IBD21:

Obliviously seen in above attacks, binding the long term identity ID_{T_i} of tag T_i to the $PID_{T_{i^{old}}}$ causes privacy weaknesses in IBD21. To prevent the above attacks, we propound that the computation of the message C_3 and the pseudo identifier $PID_{T_{i^{new}}}$ should be redesign as below:

$$C_3 = h\left(\left(R^*_{t_1}\right)_x||ID_{T_i}||T_3||T_4\right)$$
$$PID_{T_{i^{new}}} = h\left(PID_{T_{i^{old}}}||PID_{T_{i^{old}}}||\left(R^*_{t_1}\right)\right).$$

The above solution improves the scheme to provide forward and backward secrecy requirements. This enhancement also prevents an adversary can reveal the identity of the tag and breaches its anonymity so the improved scheme achieves privacy.

3.2. Synchronizations Problems in IBD21

We realize that Izza et al.'s scheme [1] cannot provide availability property due to the fact that the scheme suffers from synchronization issues. The old values of the pseudo identities $PID_{T_{i^{old}}}$ and $PID_{R^{old}}$ are not updated on the tag and reader side, respectively. Therefore, synchronization of the scheme never occurs. Even if the scheme is not under any denial of service attack, the scheme does not provide authentications between tag and reader because of using $PID_{T_{i^{old}}}$ and $PID_{R^{old}}$ values.

We think that the authors forgot to explain the update mechanism on both the tag and reader sides. Two small amendments are crucial for the scheme to prevent synchronization problems. The same update operations of $PID_{T_{i^{old}}}$ on the reader side might be used for tags. Similarly, the parameter of $PID_{R^{old}}$ might be updated with the same mechanism used on the MS side. These small amendments will prevent synchronization problems in the scheme. In fact, we assume that the same process is executed with the tag side and the reader for our cryptanalysis mentioned above.

3.3. Anonimity Problems in IBD21

We also realize that IBD21 has several serious security and privacy vulnerabilities. In the above attacks, we show that an PPT adversary can trace the tags if she obtains their internal knowledge values. We encountered the weaknesses of IBD21 when we question that what the adversary can do if she does not obtains the internal information of a tag. We claim that an adversary can threaten security and privacy of the scheme by revealing the long term tag identity ID_T.

Let $g : (\psi_j, ID_{T_i}) \rightarrow \xi_{i,j}$, where ψ_j denotes the j^{th} session parameters $[^{s_j}C_3, ^{s_j}T_3, ^{s_j}T_4]$ such that $\xi_{i,j} = {}^{s_j}C_3 - h\left(ID_{T_i}||^{s_j}T_3||^{s_j}T_4\right)$.

For example, the reader executes j^{th} session with tag T_a and $j+1^{th}$ session with tag T_b. ***Adv*** will be successful if she finds the equality $g\left(\psi_{j+1}, ID_{T_b}\right) = h\left(g\left(\psi_j, ID_{T_a}\right)||init\right)$ or equivalently $\xi_{b,j+1} = h\left(\xi_{a,j}||init\right)$.

Theorem 3.3. *IBD21 does not provide tag anonymity.*

Proof. Adversary ***Adv*** does the following attack:

1. ***Adv*** records a session s_j between a tag T and reader R, where s_j j^{th} session of the scheme.
2. ***Adv*** generates the following lists for ID_{T_i}, $\forall i \in [1, L]$ by computing $g\left(\psi_j, ID_{T_i}\right)$.
   (a) $List_0 : ID_{T_i}$,
   (b) $List_{1,j} : \xi_{i,j} = {}^{s_j}C_3 - h\left(ID_{T_i}||^{s_j}T_3||^{s_j}T_4\right)$,
   (c) $List_{2,j} : h\xi_{i,j} = h\left(\xi_{i,j}||init\right)$.
3. ***Adv*** *records* the next session and generates $List_{1,j+1}$and $List_{2,j+1}$ by computing $g\left(\psi_{j+1}, ID_{T_i}\right)$.
4. ***Adv*** compares two lists: $List_{2,j}$ and $List_{1,j+1}$. She takes an element of $List_{2,j}$ and searches it within the list $List_{1,j+1}$. Whenever she finds the match, she wins with revealing a least one corresponding tag identity so

terminates recording and searching procedures. If reader executes two consequent sessions with different tags, ***Adv*** obtains two different tag identities.

The adversary disclosures the identity of a tag with a non-negligible probability in a polynomial time. Once, adversary reveals the identity of a tag, she breaches tag privacy and she also can trace the tag. □

The success probability $prb_{\boldsymbol{Adv}}$ of the adversary depends on L which the searching space of tag identities and the number of different tags interacting with the reader during K sessions. The growth in number of sessions between tags and reader will increase her success probability. If the searching space of the adversary covers all identities of tags involved in the scheme s.t. $L = n$ and n denotes the number of tags, she definitely wins the game with $prb_{\boldsymbol{Adv}} = 1$. If all tags does not interact with the reader, $prb_{\boldsymbol{Adv}} = y/n$, where y denotes the number of different tags involved in during all recorded sessions by adversary.

Adv computes $2LK$ number of hash values and $K - 1$ numbers of sorting lists with length of L. Hence, the adversary has $\mathcal{O}(LK)$ computation complexity and $\mathcal{O}(KLlog(L))$ searching complexity. If all tags interacts with the reader and the adversary searches for all their identities, two consequent sessions are enough for her attack so she can disclosures at least one tag identity with $\mathcal{O}(n)$ computation complexity and $\mathcal{O}(nlog(n))$ searching complexity.

We examine IBD21 under the assumption of $ID_{T_i} \in \mathbb{N}$ and $ID_{T_i} \in_R \{1, n\}$. Actually, this assumptions says that the identity selection space equals the number of tags in the RFID system. As a matter of fact, the privacy of an RFID scheme should not stand to the selection space size of the tag identities.

For instance, let there are 8×10^9 number of tags in the system as many as roughly the world population [10]. The adversary can compute approximately 23 GH/s for SHA-256 (see hashCat benchmarks [11]). Therefore, the adversary can reveal a least one tag identity in less than a couple of seconds by only recording two consequent sessions.

We claim that the enhancements mention in Section 3.1.1 strengthen IBD21 and this improved scheme provides privacy and security. Using ephemeral $R^*_{t_1}$ ensures freshness and increases randomness for each session in the extended scheme. Hence, the adversary will face high searching complexity for each session due to increased randomness to break the privacy of the plan and her success probability will be non-negligible.

4. Conclusions

The proposed protocol by Izza et al. [1] suffers particularly from the existing relation between the message C_3 and the long-term identity of a tag ID_T. Therefore, IBD21 does not achieve security and privacy including tag anonymity, forward secrecy, backward secrecy. Furthermore, the scheme has some synchronization problems due to the lack of updating mechanism for pseudo identities.

In this paper, we show our attacks on the scheme and point out the synchronization problems of the scheme and we enhance IBD21 to overcome the availability, security, and privacy issues.

References

[1] S. Izza, M. Benssalah, K. Drouiche, An Enhanced Scalable and Secure RFID Authentication Protocol for WBAN Within An IoT Environment, Journal of Information Security and Applications 58 (2021) 102705. URL: https://www.sciencedirect.com/science/article/pii/S2214212620308516. doi:https://doi.org/10.1016/j.jisa.2020.102705.

[2] P. Alexander, R. Baashirah, A. Abuzneid, Comparison and Feasibility of Various RFID Authentication Methods Using ECC, Sensors 18 (2018) 2902.

[3] G. Liu, H. Zhang, F. Kong, L. Zhang, A Novel Authentication Management RFID Protocol Based on Elliptic Curve Cryptography, Wireless Personal Communications 101 (2018) 1445–1455. URL: https://doi.org/10.1007/s11277-018-5771-9. doi:10.1007/s11277-018-5771-9.

[4] A. A. Alamr, F. Kausar, J. Kim, C. Seo, A Secure ECC-Based RFID Mutual Authentication Protocol for Internet of Things, The Journal of Supercomputing 74 (2018) 4281–4294. doi:https://doi.org/10.1007/s11227-016-1861-1.

[5] D. Kumar, H. S. Grover, Adarsh, A Secure Authentication Protocol for Wearable Devices Environment Using ECC, Journal of Information Security and Applications 47 (2019) 8–15. URL: https://www.sciencedirect.com/science/article/pii/S2214212618303727. doi:https://doi.org/10.1016/j.jisa.2019.03.008.

[6] M. Naeem, S. A. Chaudhry, K. Mahmood, M. Karuppiah, S. Kumari, A Scalable and Secure RFID Mutual Aauthentication Protocol Using ECC for Internet of Things, International Journal of Communication Systems 33 (2020) e3906. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/dac.3906. doi:https://doi.org/10.1002/dac.3906. arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1002/dac.3906, e3906 dac.3906.

[7] V. Kumar, M. Ahmad, D. Mishra, S. Kumari, M. K. Khan, RSEAP: RFID Based Secure and Efficient Authentication Protocol for Vehicular Cloud Computing, Vehicular Communications 22 (2020) 100213. URL: https://www.sciencedirect.com/science/article/pii/S2214209619302608. doi:https://doi.org/10.1016/j.vehcom.2019.100213.

[8] A. Arslan, S. Kardaş, S. A. Çolak, S. Ertürk, Are RNGs Achilles' Heel of RFID Security and Privacy Protocols?, Wireless Personal Communications 100 (2018) 1355–1375. URL: https://doi.org/10.1007/s11277-018-5643-3. doi:10.1007/s11277-018-5643-3.

[9] C. H. Lim, T. Kwon, Strong and Robust RFID Authentication Enabling Perfect Ownership Transfer, in: P. Ning, S. Qing, N. Li (Eds.), Information and Communications Security, Springer Berlin Heidelberg, Berlin, Heidelberg, 2006, pp. 1–20.

[10] Worldometers, Current World Population, https://www.worldometers.info/world-population/, 2021. [Online; accessed on 17 April 2021].

[11] Jeremi M Gosney, 8x Nvidia GTX 1080 Hashcat Benchmarks, https://gist.github.com/epixoip/a83d38f412b4737e99bbef804a270c40, 2021. [Online; accessed on 17 April 2021].