 doi:10.20944/preprints201907.0298.v1

Peer-reviewed version available at *Sensors* **2019**, *19*, 4752; doi:10.3390/s19214752

Article

Securing IoT Based RFID Systems: A Robust Authentication Protocol Using Symmetric Cryptography

Khwaja Mansoor [1,‡], Anwar Ghani [2,‡], Shehzad Ashraf Ch. [2,‡], Shahaboddin Shamshirband [3,‡,*], Shahbaz Ahmed Khan Ghayyur [2,‡], and Ely Salwana [4,‡]

1 College of Signals, National University Science & Technology Islambad; kh.mansoorulhassan@gmail.com
2 Department of Computer Science & Software Engineering, International Islamic University Islamabad; anwar.ghani@ii.edu.pk, shahzad@iiu.edu.pk, shahbaz@iiu.edu.pk
3 Department for Management of Science and Technology Development, Ton Duc Thang University, Viet Nam; shahab1396@gmail.com
4 Institute of Visual Informatics, Universiti Kebangsaan Malaysia Bangi, Selangor, Malaysia; elysalwana@ukm.edu.my
* Correspondence: shahab1396@gmail.com
‡ These authors contributed equally to this work.

Abstract: Radio Frequency Identification (RFID) devices use radio waves to relay identifying information to an electronic reader using low-cost RFID Tag. RFID is expected to replace the conventional bar-code identification system due to its advantage like real-time recognition of a considerable number of objects. However, in RFID systems an attacker can get the tag that may lead to various security threats, and the limited computation power of RFID tags can cause delay. Lightweight authentication protocols proposed using cryptographic algorithms (one-way hash function, symmetric key encryption/decryption, and exclusive-OR) in order to cope with these problems. One such lightweight cryptographic protocol has been presented by Gope and Hwang using RFID systems. However, it analyzed in this article that their protocol is infeasible and vulnerable to Collision Attack, Denial-of-service (DoS), and Stolen verifier Attacks. A realistic, lightweight authentication protocol has been presented in this article to ensure protection against the mentioned attacks for IoT based RFID system. The proposed protocol has been formally analyzed using BAN logic and ProVerif as well as also analyzed informally using security requirement. The results show that the proposed protocol outperforms the existing protocols not only in security enhancements but also in terms of computation and communication complexity. Furthermore, the proposed protocol has also been analyzed for storage complexity.

Keywords: authentication protocol; IoT Security; RFID security; symmetric cryptography;

---

1. Introduction

Internet of things (IoT) is based on the Internetwork connectivity of daily use objects. It consists devices(such as sensors) with Internet connectivity capable of communication and interaction with others and be controlled and monitored remotely [1,2]. The conventional bar code system suffer from various issues like line of sight communication, scanning one at time, prone to physical damage, and limited storage of information, is rapidly being replaced by the emerging IoT based RFID systems. Due to the scares resource of RFID Tags, an RFID based could be an attractive target for attacker. Therefore,

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 26 July 2019 doi:10.20944/preprints201907.0298.v1

Peer-reviewed version available at Sensors 2019, 19, 4752; doi:10.3390/s19214752

it is very important to pay a special attention to the security and authentication of such systems. The overall RFID system architecture has been depicted in Figure 1.

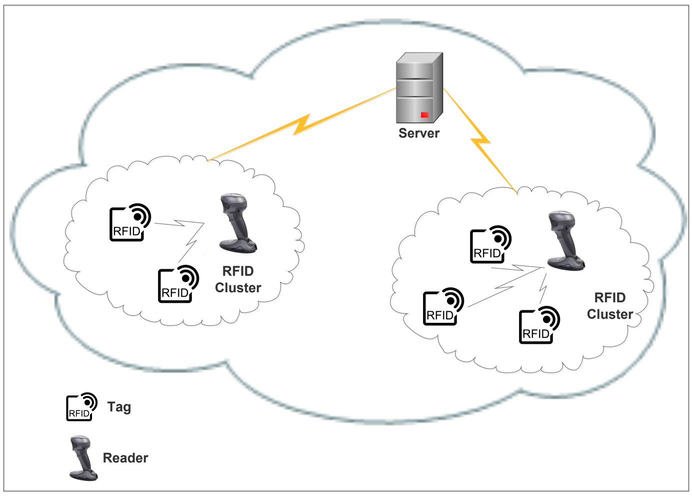

Figure 1. RFID System Architecture

RFID technology is most simple form of pervasive sensor networks that are commonly used for identification of different physical objects [3,4]. It uses a two way radio signals receiver and transmitter called reader or the interrogators. Radio signals is sent to the tag attached to the physical object and the interrogator or reader is responsible to respond [5,6]. RFID reader device is a network connected device (mobile or fixed) along with an antenna which is responsible to transmit power, data as well as commands to the objects tags [7–9]. RFID reader device is an access point for the RFID-Tags clusters that is responsible for the availability of the tag information to different System application. Tags are of two types, passive tags in which energy is transferred using radio frequency from reader device and active tags they have internal energy source like power batteries. Different features are described in Table 1.

Table 1. RFID-Tag Features

Features	Passive Tags	Active Tags
Data Storage	128 bytes	128 bytes
Tag Power	Energy transferred through Radio Frequency from Reader	Internal source to Tag
Tag Battery	No	Yes
Availability of Source Power	Only in range of Radar	Continuous
Signal Strength required to Tag	Very High	Very Low
Range	Upto 3-5 M	Upto 100 M
Multiple Tag Reading	less then thousand Tags within 3 M of Reader range	More then 1000 tags recognized upto 100mph

RFID technology is system having identification device which use small radio signals for object tracking and identification purpose. RFID tagging system contain the tags, read and write device and the system application for data processing, transmission and collection of data. RFID technology has a very broad application domain and can be used in high value innovative solutions assisting in the areas of IT Asset Tracking, Race Timing, E-Passport, Transportation, Payments, Human Implants, Supply-Chain-Management, Fleet and Asset-Management, Security Access-Control, E-Commerce, and Traffic Analysis and Management[8,10–13].

A low cost method to identify objects is to used RFID device that uses radio waves. These devices relays identifying information to an electronic reader using RFID Tags [2]. However, reliable identification of various objects is a great challenge especially if there are various object present at the same time. In some applications, it is very important to identify objects for example, vehicle tracking system, entry and exit of a free way, and smart parking etc. Vehicle tracking is specifically crucial in security and law and order applications [10]. Therefore, an authentication process suitable for RFID based system is very crucial to ensure authenticity of the information exchanged between different RFID enabled devices [8].

As IoT based RFID systems remains an emerging technology consisting of limited resource devices, it is vulnerable to various security threats like data threats, authentication, and security threats [14–16]. Due to the broad application domain, RFID must be more secure and protected to meet the industrial standards. For reliable security of RFID, different types of features in security protocols should be considered; 1) security schemes of RFID must provide user friendliness, 2) user anonymity must be ensured to protect the identity of legitimate users, 3) there should be a provision for change moreover, update of tag data at any time securely, 4) ensure backward and forward secrecy in RFID tag reading-writing process, 5) protection of an RFID system from malicious system users to prevent insider attacks, 6) having capabilities to endure replay attacks, 7) having capabilities to withstand the impersonation and forgery attacks, 8) having capabilities to achieve secure mutual authentication, and 8) having capabilities to endure man-in-middle attacks.

Different technological solutions are available to overcome the issues of reliable identification of different objects, however, all such technological solutions have their specific limitations. For example, using bar code for identification of objects is the most prevalent technology used today. However, reading the bar code required manual and physical contact in the line of sight between the bar code reading device and the tag [13,17]. It makes the bar code based process slow and inefficient in real time applications.

Different secure authentication protocols proposed by different researchers to achieve the above mentioned Security features. However, the security of RFID system is still inadequate and facing various security issues like spoofing, RFID counterfeiting, tracking, sniffing, denial of service, repudiation, and replay attacks [18]. Recently, a lightweight cryptographic protocol has been presented by Gope and Hwang [19] using RFID systems. However, after analysis it is clear that their protocol is unfeasible and vulnerable to Collision Attack, Denial-of-service (DoS), and Stolen verifier Attacks.

In this article, an improved and robust authentication protocol based on symmetric cryptography has been presented for IoT based RFID system to overcome the issues in Gope and Hwang [19] protocol. The general contributions of this article include the following:

- To perform cryptanalysis of the baseline protocol for possible security loopholes and vulnerabilities.
- To propose an improved authentication protocol for the same scenario to overcome the security issues of the baseline protocol.
- To formally and informally analyze the proposed protocol for possible security lapses.
- To compare the proposed protocol in terms of the security requirements with existing protocols. Furthermore, to comparatively analyze the proposed protocol with the existing protocol in terms of computation and communication cost (complexity).

Rest of the article organized in various sections. In section 2 a brief overview of the literature has been presented followed by a detailed review of Gope and Hwang [19] that is base of the proposed protocol which the base of this research work. Section 3 presents the proposed protocol whereas section 4 presents the detail security analysis of the proposed protocol. Section 5 presents a comparative analysis of the proposed protocol with existing protocols and finally section 6 concludes the article.

2. Literature Survey

In literature, authentication in RFID based system has been rigorously investigated. Recently, various anonymous mutual authentication security schemes/protocols have been proposed for RFID Systems [11,12,19–31]. The protocols in literature have been categorized into non-public key-cryptosystem (NPKC) and public-key-crypto system (PKC) security protocol. Also, other researchers came up with Asymmetric/public key cryptosystem (PKC) solution based on elliptic curve cryptosystem (ECC). Due to the modular exponential operations having very high computation cost, the elliptic curve cryptosystem (ECC) based security protocols are infeasible for low-powered RFID systems. However, using the symmetric key cryptosystem (NPKC) solution – hash based RFID Systems may be a feasible solution due to their equitable computational overhead.

In 2005, the authors in [23] proposed a symmetric public key crypto (PKC) based protocol for RFID Systems that was mainly designed using exclusive-OR and a one-way Hashing function. The authors claimed that their protocol addresses various security issues faced by the RFID-Tag System. Unfortunately, their scheme is vulnerable to different security threats including forgery attack, man-in-the-middle attack, and traceability issue. subsequently, various security protocols including; [26], [30], and [21] were designed using hash function encryption, and exclusive-OR encryption. However, they proved to be exposed to various security attacks like DoS attack, forgery attack [28].

Yang et al. [23] in 2005 presented a mutual authentication protocol for RFID-System which utilizes a hash function and symmetric-key cryptography. The protocol successfully achieved low-cost mutual authentication between FRID-Tags and RFID-Server. This mutual authentication between an RFID-Tag and a server have been achieved in a total of six messages. Message 1 and 2 are used for sharing a secret key between a Tag and a Server while the messages 3 to 6 are responsible for session-specific parameters. Two new and secure shared keys used between two participants for each session. The correctness of the proposed protocol has been verified using GNY logic [32]. Furthermore, the authors claimed that their protocol is secured and resistant to all known attacks. However, their protocol still has various vulnerabilities like exposure to forgery attack, man-in-the-middle attack, and traceability problems.

Tan et al. [30] proposed a server-less authentication scheme for RFID-System in 2008. In the scheme, they put forward a secure searching technique for Tag. In their scheme authentication occurred in between only two participants Tag and Reader. A trusted certificate authority is used to deploy Tags and authorize reader devices. Communication takes place on secure channel among certificate authority, Tag and RFID reader in deploying Phase. While mutual authentication is done over a public channel between RFID-Tag and RFID Reader only. However, their scheme has scalability issue and also vulnerable to different attacks such as forgery attacks, untraceability, de-synchronization, and DoS attacks.

Cai et al. [21] proposed a security scheme for RFID System. They improved a secure extension of the previous security schemes for RFID Systems. The designed architecture was based on XOR operation and Hash encryption to overcome the RFID-Tag impersonation attack. RFID-Reader impersonation attack and resist adversary to manipulate legitimate RFID-Tag, and Reader. They came up with the solution of ownership transfer issue of RFID tags moreover, preserve synchronization issue among RFID-Tags and server. Five communication messages used by Cai et al. scheme to complete mutual authentication Between participants. They claim that their security scheme resists server impersonation attack, Tag impersonation attack, and de-synchronization attack. However, their scheme cannot resist forwarding secrecy and DoS attack. .

Authors in [29] put forward an authentication protocol designed for RFID based systems. The main focus of their proposal is to overcome brute force attack and to keep retrieval cost minimum. The authors claimed that their proposed protocol is resistant to different attacks like DoS attack, replay attack, man-in-the-middle attack and successfully achieved mutual authentication between RFID-Tag and the server. They also claimed that their scheme preserves Confidentiality, indistinguishability, forward secrecy and mutual authentication.

Gope and Hwang [19] performed cryptanalysis of [31] security protocol and proved that it is vulnerable to reader impersonation attack, DoS attack, and Tag impersonation attack. Gope and Hwang presented a lightweight authentication protocol. This article examined and found that their scheme contains some vulnerabilities like Collision Attack, Stolen verifier Attack and Denial-Of-Service (DoS). Furthermore, Gope and Hwang presented their protocol and claimed to be a more secure scheme then [31] scheme.

Table 2. Notation Guide

Notations	Description
T	RFID-tag
R	Reader Device
S	Database Server System
ID_{T-i}	ith Tag identity
AID_T	One-time Tag alias identity
SID	Shadow identity
R_j	jth Reader identity
N_t	Tag Random number
N_r	Reader Random number
K_{ts}	Shared key of server and tag
K_{emg}	Shared emergency key of server and tag
K_{rs}	server and tag secret key
Tr_{seq}	Track sequence number (used by both S and T)
r_j	Randomly derived from Shadow-ID and Emergency Key
$h(.)$	Hash function
$\oplus$	The exclusive XOR operation
$\|\|$	concatenation

The Gope and Hwang [19] protocol has been investigated in the following subsection in detail and its cryptanalysis has been performed to check it for vulnerabilities. From now on this protocol will be referred to as the baseline protocol. Different symbols used in designing the proposed protocol has been presented in Table 2.

2.1. Adversarial Model

The proposed protocol has been designed keeping in mind the following adversarial model where assumptions may be common as pointed out in [33]. As per mentioned adversary $\mathcal{A}$, the following assumptions are considered.

1. The public channel is under full control of $\mathcal{A}$, so that the $\mathcal{A}$ has the ability to intercept, reply, modify, revert or even resend fresh fabricated message.
2. $\mathcal{A}$ has the capability to extract the RFID-Tag by power analysis of the information or leak the same information.
3. There is a chance that $\mathcal{A}$ might be an unauthorized user or a malicious user of the system.
4. The insider has access to the identities of legitimate users and server information as these are public and available to the system's users.
5. The servers are assumed to be under protection, however, $\mathcal{A}$ cannot break the protection of server as well as the system.
6. The $\mathcal{A}$ does not have any access to the secret key of the server.

2.2. Review of Baseline Protocol

The proposed baseline protocol designed for RFID has three main entities; 1) Database Server, 2) Reader Device, and 3) RFID Tags. The network layout of RFID System divided into several RFID clusters. Every cluster consists of a Reader and many Tags. Tags can shift from one cluster to another. Every RFID Reader of the cluster is required to authenticate RFID Tags through Database server where the RFID-Tags need to be registered. Each cluster and database server share a symmetric key Krs [19]. Gope and Hwang [19] authentication scheme consist of two main phases; 1) Tag Registration Phase, 2) Tags Authentication Phase.

2.3. Baseline Protocol Registration Phase

Step-1: Every RFID Tag submit its ID_{T_i} to the RFID Database S.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 26 July 2019 doi:10.20944/preprints201907.0298.v1

Peer-reviewed version available at *Sensors* **2019**, *19*, 4752; doi:10.3390/s19214752

Step-2: Database Server generates random number n_s and computes $K_{ts} = h(ID_{T_i} \| n_s \oplus ID_s)$.

Step-3: S generates set of unlikeable shadow ID_s, and $SID = \{sid_1, sid_2 \cdots\}$, where the $sid_j \in SID$. Database Server computes $sid_j = h(ID_{T_i} \| r_j \| K_{ts})$. Then it generates a set of emergency keys $K_{emg} = \{k_{emg_1}, k_{emg_2} \cdots\}$, each of the key corresponding to $sid_j \in SID$ where each $k_{emg_i} \in K_{emg}$. S then computes $k_{emg_i} = h(ID_{T_i} \| sid_j \| r_j)$.

Step-4: The Database Server S generates a 32-bit random sequence number Tr_{seq}. Then S generates random number m and matches it with Tr_{seq}, $Tr_{seq} = m$, and send the Tr_{seq} to the RFID Tag through Reader R by maintaining the copy of Tr_{seq} in its database for speeding up the authentication process.

Step-5: The RFID Server S authenticates the validity of RFID Tag ID_{T_i} based on TR_{seq}. If TR_{seq} does not have a match within the record of RFID Server S, then the Server S does not authenticate RFID Tag and terminate the process. In this case the RFID Tag ID_{T_i} will use one of its fresh pair of the emergency key $k_{emg_j} \in K_{emg}$ and shadow ID $sid_j \in SID$. The used pair of shadow ID and emergency ID (SID, K_{emg}) must be deleted from both, the Database Server S and the RFID Tag ID_{t_i}. Database Server S again updates and send $\{K_{ts}, (SID, K_{emg}), Tr_{seq}, h(.)\}$ through secure channel for further communication. The registration phase of the baseline protocol has be shown in Figure 2.

$\mathcal{T}agID_{T_j}$		$\mathcal{D}atabaseServer(S)$
Identity: ID_{T_i}		
	$M=\{ID_{T_i}\}$ $\longrightarrow$	
		Generate: n_s Generate random number: m Set: $Tr_{seq} = m$ Compute:$K_{ts} = h(ID_{T_i} \|\| n_s) \oplus ID_s$ $sid_j = h(ID_{T_i} \|\| r \|\| K_{ts})$ $emg_j = h(ID_{T_i} \|\| sid_j \|\| r_j$ $sid_j \in SID, K_{emgj} \in K_{emg}$ Store: $\{ID_{T_i}, K_{ts}, Tr_{seq}, (SID, K_{emg})\}$
	$\longleftarrow$ $M=\{ID_{T_i}, K_{ts}, Tr_{seq}, (SID, K_{emg})\}$	
Store: $\{ID_{T_i}, K_{ts}, Tr_{seq}, (SID, K_{emg})\}$		

Figure 2. Gope-Hwang's proposed registration scheme

2.4. Baseline Protocol Tag Authentication Phase

After the registration of RFID Tags with the Database Server S, mutual authentication of RFID Tags with the Database Server Sthrough Reader R starts that is described in the following Steps.

Step-1: RFID Tag ID_{T_i} generates random number N_t, then derives $AID_T = h(ID_{T_i} \| K_{ts} \| N_t \| Tr_{seq})$, $N_x = K_{ts} \oplus N_t$ and computes $V_1 = h(AID_{ti} \| K_{ts} \| N_x \| R_i)$. RFID Tag sends a message request as M_{A_1} to the Reader device R_i and the Reader R_i also receives a recently used sequence number from the Server S for mutual authentication. In the case of synchronization loss, the RFID Tag uses one of its fresh pair (sid_j, K_{emg_j}). Subsequently, assign to the sid_j as AID_T and then k_{emg_j} as K_{ts}. RFID Tag sends a message to the Reader R as $M_{A_1} : \{AID_T, N_x, Tr_{seq}, V_1\}$.

Step-2: Upon receiving request from RFID Tag, reader of the cluster (in which Tag is located) generates random number N_r and computes $N_y = K_{rs} \oplus N_r$, $V_2 = h(M_{A_1} \| N_r \| K_{rs})$. Reader R_i sends a message to the Database Server S for verification. $M_{A_2} : R_i \rightarrow S\{N_y, R_i, V_2, M_{A_1}\}$.

Step-3: When the Database server S receives a request from the Reader R, first it validate track sequence number Tr_{seq} by computing $V_1 = h(AID_T \| K_{ts} \| N_x \| R_i)$. Database Server S then derives

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 26 July 2019 doi:10.20944/preprints201907.0298.v1

Peer-reviewed version available at Sensors 2019, 19, 4752; doi:10.3390/s19214752

$TagID_{T_i}$	$ReaderR_i$	$DatabaseServer(S)$
Generate: N_t Compute:$N_x = K_{ts} \oplus N_t$ $AID_T = h(ID_{Ti}\|\|K_{ts}\|N_t\|Tr_{seq})$ $V_1 = h(ID_T\|\|K_{ts}\|N_x\|R_i)$ or $sid_j \in SID, K_{emgj} \in K_{emg}$ $AID_T = sid_j, K_{ts} = K_{emg}$		
$\xrightarrow{M_{A_1}=\{,AID_T,N_x,Tr_{seq},V_1\}}$		
	Generate: N_r Decieve: $N_y = K_{rs} \oplus N_r$ Computes: $V_2 = \{M_{A_1}\|N_r\|K_{rs}\}$ $\xrightarrow{M_{A_2}=\{N_y,R_i,V_2,M_{A_1}\}}$	
		Verify: Tr_{seq} Derive:$N_t = K_{ts} \oplus N_x, N_r = K_{rs} \oplus N_y$ Compute and verify:$V_2?, V_1?, AID_T?$ Generate: m Compute: $Tr_{seq_{new}} = m$ $Tr = h(K_{ts}\|ID_{T_i}\|N_t) \oplus Tr_{seq_{new}}$ $V4 = h(Tr\|K_{ts}\|ID_{T_i}\|N_t)$ $V3 = h(R_i\|N_r\|K_{rs})$ Update: $K_{ts_{new}} = h(K_{ts}\|ID_{T_i}\|Tr_{seq_{new}})$ $Tr_{seq} = Tr_{seq_{new}}, K_{ts} = K_{ts_{new}}$ *or* Generate $K_{ts_{new}}$ Compute $x = h(ID_{T_i}\|k_{emg_j}) \oplus K_{ts_{new}}$ Generate $K_{ts} = K_{ts_{new}}$
	$\xleftarrow{M_{A_3}=\{Tr,V_3,V_4,x(ifreq.)\}}$	
	Compute and Check: $V_3^* \stackrel{?}{=} h(R_i\|N_r\|K_{rs})$	
$\xleftarrow{M_{A_4}=\{Tr,V_4,x(ifreq.)\}}$		
Compute and Verify: $V4^* \stackrel{?}{=} h(T_r\|K_{ts}\|ID_{T_i}\|N_t)$ Compute and update: $Tr_{seq_{new}} = h(K_{ts}\|ID_{T_i}\|N_t) \oplus T_r$ $K_{ts_{new}} = h(K_{ts}\|ID_{T_i}\|Tr_{seq_{new}}$ $Tr_{seq} = Tr_{seq_{new}}, K_{ts} = K_{ts_{new}}$ *Or* $K_{ts_{new}} = h(ID_{T_i}\|k_{emg_j}) \oplus x,$ $K_{ts} = K_{ts_{new}}$		

Figure 3. Gope-Hwang's proposed authentication scheme

$N_t = K_{ts} \oplus N_x$ and verify AID_T. Upon successful verification of AID_T, the Database Server S generates a random number m and assigns it to $T_{r_{seq}} = m$. It also computes $T_r = h(K_{ts}\|ID_{T_i}\|N_t) \oplus T_{r_{seq}}$, $V_4 = h(T_r\|K_{ts}\|ID_{T_i}\|N_t)$, $V_3 = h(R_i\|N_r\|K_{rs})$ to creates a Message M_{A_3} and send it to the Reader R. Finally, the Database Server S computes $K_{Ts_{new}} = h(K_{ts}\|ID_{T_i}\|Tr_{seq_{new}})$ and updates $K_{Ts_{new}}$ and $Tr_{seq_{new}}$. In case the message M_{A_1} dose not contain $T_{r_{seq}}$, then the Database Server S randomly generates a new shared key $K_{TS_{new}}$ using the emergency key K_{emg_j} and real identity of the RFID Tag ID_{T_i}. $x = K_{ts_{new}} \oplus h(ID_{T_i}\|K_{emg_j})$ is computed and x is sent within the message M_{A_4}. Where V_4 is calculated as $V_4 = h(N_t\|T_r\|x\|K_{emg_j})$. $M_{A_3} : S \rightarrow R_i : \{T_r, V_3, V_4, x_{(ifreq.)}\}$.

Step-4 The Reader R receives M_{A_3} and computes $h(R_i\|N_r\|K_{rs})$ and validates if it is equal to V_3. If it is equal RFID Reader R sends M_{A_4} to the Tag. Contrarily, the Reader R terminates the established connection. RFID Tag receives the message M_{A_4} and computes $h(Tr\|K_{ts}\|ID_{T_i}\|N_t)$ and verifies whether it is equal to V_4 or not. RFID Tag derives $K_{ts_{new}} = h(K_{ts}\|ID_{T_i}\|Tr_{seq_{new}})$ and stores $K_{ts} = K_{ts_{new}}$, $Tr_{seq} = Tr_{seq_{new}}$ for future communication. $M_{A_4} : R_i \rightarrow ID_{T_i} : \{T_r, V_4, x_{(ifreq.)}\}$. The detailed stepwise representation of the baseline protocol has been shown in Figure 3.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 26 July 2019 doi:10.20944/preprints201907.0298.v1

Peer-reviewed version available at Sensors 2019, 19, 4752; doi:10.3390/s19214752

2.5. Cryptanalysis of Baseline Protocol

The baseline protocol [19] provides authentication for RFID system in a distributed environment, and the entire protocol is based on the verification of Track sequence number. In the registration phase, the Database Server generates an auto-generated a random number, save it in its database and sends its copy to the RFID Tag through a secure channel.During the authentication phase, the RFID Tag sends $M_{A_1} = \{AID_T, N_x, Tr_{seq}, V_1\}$ to the Reader (R_i) .

The Database Server receives the message M_{A_2} from the reader that contains different information of a particular RFID Tag like $M_{A_2} = \{M_{A_1}, N_y, R_i, V_2\}$ for authentication. The first step of the Database Server (S) for authenticating the particular RFID Tag is to check its Tr_{seq} number. If the received Tr_{seq} number of RFID Tag does not matches, the authentication process terminates, otherwise it undergoes further computation. Here the proposed protocol in [19] is totally dependent on the Tr_{seq} number and the Tr_{seq} number is generated randomly.

2.5.1. Vulnerable to Collision Attack

In baseline protocol [19] verification is done in the very first step of the Login phase. If collision occurs in generating the Tr_{seq} for different RFID Tag, there is no provision for handling such situation. The Database Server can cannot handle such situation. Therefore, it will rather terminate the authentication process.

2.5.2. Vulnerable to Stolen Verifier Attack

An adversary A can steal verified data from the server during the current or any past authentication session. The verified data does not have the secret keys with XOR or any other encryption. An Adversary is capable of generating communication data using the stolen data and resend it to the server. If this operation has been successful, the Adversary impersonates a legal user in the next communication and authentication section. In Gope and Hwang [19] scheme Tr_{seq} number is saved unencrypted on the server's verifier table. It leads to the exposure of verifier table to become a victim of Stolen Verifier Attack.

2.5.3. Vulnerable to Denial-Of-Service (DoS) Attack

In baseline proposal [19]an adversary can launch DoS attack by continuously generating 32 bits random Tr_{seq} numbers and send it to the Database Server. It will keep the Database Server busy in verification of dummy random numbers thus avoiding it to serve a legitimate request.

3. Proposed Scheme

The proposed protocol for RFID Tag System consists of three main entities; 1) Database Server, 2) Reader Device, and 3) RFID Tags. RFID system is divided into different clusters where every cluster consists of RFID Reader and many tags. RFID Tags can move from one cluster to another. Every Reader of the cluster is required to authenticate RFID Tags through the Database server where RFID Tags need to register itself. Each cluster and the Database Server share a key K_ts in a symmetric way. The proposed mutual authentication protocol consists of two main phases: 1) Tags Registration Phase, 2) Tags Authentication Phase.

3.1. Tags Registration Phase

In this phase, the Database Server issues security credentials to RFID Tags using a secure channel. This process takes place in the following steps.

Step-1: Every RFID Tag submits its ID_{T_i} to the RFID Server S. $M = \{ID_{T_i}\}$

Step-2: The Database Server Generates a random number n_s and computes $K_{ts} = h(ID_{T_i} \| n_s \oplus ID_s)$.

Step-3: The Database Server S also computes one-time alias Tag identity AID_T by encrypting it with the Secret Key of Database Server as $AID = E_{s_x}(ID_{T_i} \| r_{T_i})$. Here r_i is a random number.

Step-4: The Database Server S authenticates the RFID Tag ID_{T_i} based on AID_T in Tags authentication Phase by checking if a request is valid or not.

Step-5: The Database Server stores and sends a message M to the RFID Tag through a secure channel. $M = \{ID_{T_i}, K_{ts}, AID\}$. Upon receiving the message from database server S, the RFID Tag stores the information in its memory.

The detailed steps of the proposed registration phase have been shown in Figure 4.

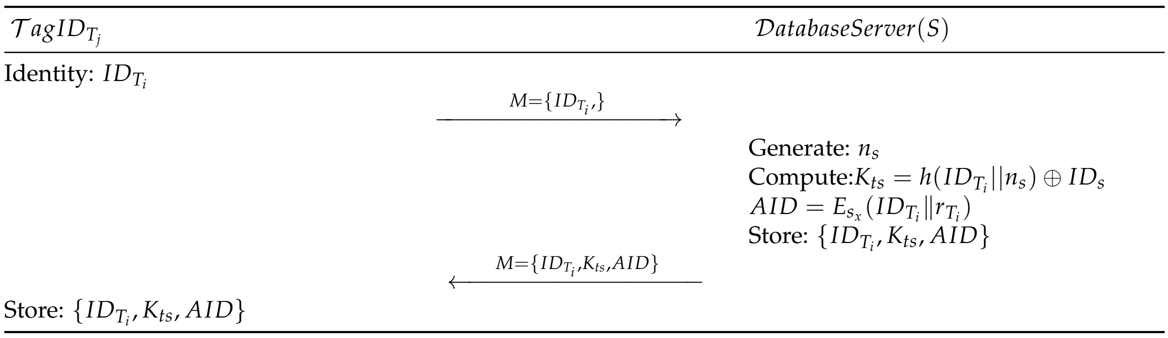

Figure 4. Registration phase of the proposed protocol

3.2. Tags Authentication Phase

Once registered with the Database Server S, the RFID Tags undergoes a mutual authentication process with the Database Server S through the Reader R. This process described in the following Steps.

Step-1: RFID Tag ID_{T_i} generates a random number N_t, then derives $N_x = K_{ts} \oplus N_t$ and $V_1 = h(AID_t \| K_{ts} \| N_x \| R_i)$. RFID Tag makes a request through the message $M_{A_1} : \{AID_T, N_x, T_1, V_1\}$ to Reader device R_i for mutual authentication.

Step-2: Upon receiving the request from RFID Tag, reader of the cluster (in which Tag is located) first verifies and checks the freshness of the message as $(T_2 - T_1) \leq \Delta T$. Then the Reader generates a random number N_r and computes $N_y = K_{rs} \oplus N_r$, $V_2 = h(M_{A_1} \| N_r \| K_{rs} \| T_2)$. Reader R_i sends a message to the Database Server S for verification. $M_{A_2} : R_i \rightarrow S\{N_y, R_i, V_2, M_{A_1}, T_2\}$.

Step-3: When the Database Server S receives the request from the Reader R, first it verifies $(T_3 - T_2) \leq \Delta T$, then derives $N_t = K_{ts} \oplus N_x$ and $N_r = K_{rs} \oplus N_y$. Further, the Database Server computes and verifies $V_1 = h(AID_T \| K_{ts} \| N_x \| R_i)$, $V_2 = h(M_{A_1} \| N_r \| K_{rs} \| T_2)$. The Database Server S then verifies AID_{T_i} by decrypting it as $AID_{T_i} = D_{S_x}(ID_{T_i} \| r_i)$. After verification of AID_T the Database Server computes $V3 = h(R_i \| N_r \| K_{rs} \| T_3)$ and $V4 = h(K_{ts} \| ID_{T_i} \| N_t \| T_3)$. Upon successful verification the server then updates $AID_{T_{i(new)}} = E_{S_x}(ID_{T_i} \| r_{i_{(new)}})$ and derives $Z_T = AID_{T_{new}} \oplus K_{Ts}$ and creates a message M_{A_3} and sends it to the Reader R. $M_{A_3} : S \rightarrow R_i : \{V_3, V_4, Z_T, T_3\}$

Step-4: The Reader R receives M_{A_3} and checks the freshness of the message as $(T_4 - T_3) \leq \Delta T$ and computes $h(R_i \| N_r \| K_{rs})$ and verifies if it is equal to V_3. If true then RFID Reader R sends M_{A_4} to RFID

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 26 July 2019 doi:10.20944/preprints201907.0298.v1

Peer-reviewed version available at *Sensors* **2019**, *19*, 4752; doi:10.3390/s19214752

Tag otherwise the Reader R terminates the established connection. $M_{A_4} : R_i \rightarrow ID_{T_i} : \{V_4, T_4, Z_T\}$

Step-5: The RFID Tag receives a message M_{A_4} and checks its freshness. After that the message is verified as $V4^* \stackrel{?}{=} h(K_{ts} \| ID_{T_i} \| N_t)$. After that RFID Tag computes and updates $AID_{T_i(new)} = (Z_T \oplus K_{Ts})$, $AID_{T_i} = AID_{T_i(new)}$ and save the information for next authentication process. Else freshness check get unsuccessful then Tag will not update AID_{T_i} as $AID_{T_i(new)}$.

Steps of the proposed authentication protocol have been shown in Figure 5.

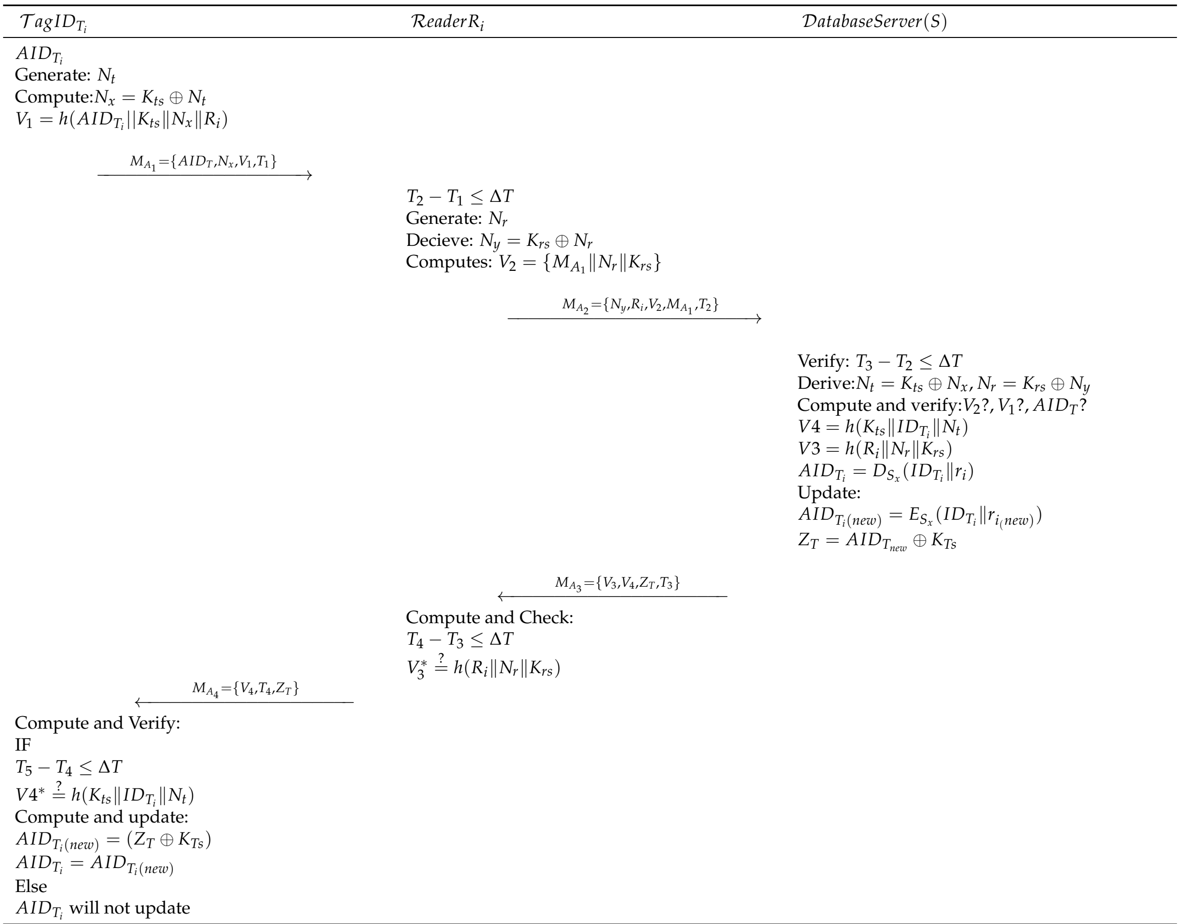

Figure 5. Proposed authentication protocol

4. Security Analysis

In this section, the security features and robustness of the improved proposed authentication protocol has been analyzed in the light of the adversarial model that has been discussed in subsection I-A. Here, the security strength of the proposed protocol against all known security attacks that an adversary can launch examined. This task has established in multiple steps; 1) Formal analysis, 2) Informal analysis, and 3) Comparative analysis. Formal analysis of the proposed protocol has been performed using two methods; a) BAN logic, and b) ProVerif simulation. Informal analysis performed by launching various attacks on the proposed protocol and analyzing it for possible loopholes. Finally, the proposed protocol has also been comparatively analyzed for computation and communication complexity (cost or time) with existing state-of-the-art protocols.

4.1. Security Analysis with BAN-logic

Burrows Abadi-Needham (BAN) logic consists of a set of rules that can be used to analyzed information exchange protocols. It specifically determines if a the information exchanged in a protocol

 doi:10.20944/preprints201907.0298.v1

Peer-reviewed version available at Sensors 2019, 19, 4752; doi:10.3390/s19214752

Table 3. BAN logic Notations

Notations	Description
$P\| \equiv X$	P Believes that X
$P \triangleleft X$	P Sees that X
$P\| \sim X$	P once said X
$P \Rightarrow X$	P have total jurisdiction on X
$\#(X)$	X is updated and fresh
(X, Y)	X, Y is component of formula(X,Y)
$< X >_Y$	X is combine with Y
$(X)_K$	Hash of message X using a key K
$P \overset{K}{\longleftrightarrow} Q$	P and Q share key K for communication
$AIDT_i$	$AIDT_i$ is one time session key
$\frac{P\|\equiv P \overset{K}{\longleftrightarrow} Q.p \triangleleft < X >_K}{P\|\equiv Q\|\sim X}$	Message-Meaning rule
$\frac{P\|\equiv\#(X)}{P\|\equiv\#(X,Y)}$	Freshness-conjuncatenation rule
$\frac{P\|\equiv\#(X), P\|\equiv Q\|\sim X}{P\|\equiv Q\|\equiv X}$	Nonce-verification rule
$\frac{P\|\equiv Q\Rightarrow X, P\|\equiv Q\|\equiv X}{P\|\equiv X}$	Jurisdiction rule
$P\| \equiv X$	P believes X

is resistant against eavesdropping and trustworthy and secured. The mutual authentication of the proposed protocol has been checked using the BAN logic [34]. Different rules of BAN logic including idealized form, assumptions, and proofs are shown in Table 3.

To analyzed the security a protocol using BAN logic, different goals have to be determined. In case of the proposed protocol, eight different goals have been determine based on BAN Logic. These goals have been shown in the following list.

- Goal 1: $R_i| \equiv Tag \overset{AID_T}{\longleftrightarrow}$ Ri
- Goal 2: $R_i| \equiv Tag| \equiv Tag \overset{AID_T}{\longleftrightarrow}$ Ri
- Goal 3: $S_j| \equiv R_i \overset{AID_t}{\longleftrightarrow}$ Sj
- Goal 4: $S_j| \equiv R_i| \equiv R_i \overset{AID_T}{\longleftrightarrow}$ Sj
- Goal 5: $R_i| \equiv S_j \overset{AID_T}{\longleftrightarrow}$ Ri
- Goal 6: $Ri| \equiv S_j| \equiv S_j \overset{AID_T}{\longleftrightarrow}$ Ri
- Goal 7: $Tag| \equiv R_i \overset{AID_T}{\longleftrightarrow}$ Tag
- Goal 8: $Tag| \equiv R_i| \equiv R_i \overset{AID_T}{\longleftrightarrow}$ Tag

The achieve the goals listed above, the security analysis using BAN logic has been divided into two parts; part1 and part2. Part1 shows the idealized form of the protocol that has been proved in Part3 whereas Part2 uses assumptions to analyzed the proposed protocol.

Part1: The idealized form for the proposed protocol has been discussed as follows:

- M1: Tag →Ri: AID_T,N_x $:< N_t >_{K_{ts}}, V1, T1$
- M2: R_i →Sj: M1,N_y $:< N_r >_{K_{rs}}, R_i, V2, T2,$
- M3: S_j →Ri: V3,V4,Z_t $:< AIDT_i >^{*}_{K_{ts}}, T3$
- M4: $R_i \rightarrow Tag : V4, T4, Z_t :< AIDT_i >_{K_{ts}}$

Part2: The assumptions used for analyzing the proposed protocol using BAN logic have been shown below:

- A1: $Tag| \equiv \#(N_t)$
- A2: $R_i| \equiv \#(N_r)$
- A3: $S_j| \equiv \#(AIDTi)(r_i)$
- A4: $R_i| \equiv S_j \Rightarrow r_i$
- A5: $R_i| \equiv Tag \Rightarrow N_t$
- A6: $S_j| \equiv R_i \Rightarrow N_r$
- A7: $S_j| \equiv Tag \Rightarrow N_t$
- A8: $Tag| \equiv S_j \Rightarrow r_i$
- A9: $Tag| \equiv R_i \Rightarrow N_r$

Part 3: Analysis of Idealized form of the proposed protocol that has been derived on the basis of BAN logic assumptions and rules are described as follow:

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 26 July 2019 doi:10.20944/preprints201907.0298.v1

Peer-reviewed version available at *Sensors* **2019**, *19*, 4752; doi:10.3390/s19214752

M1: Tag →Ri: $AIDT_i$,N_x $:< N_t >_{K_{ts}}, T1$ is time-stamp of Tag
Using seeing rule, the following can be achieved

- S1: $R_i \lhd AIDT_i, SID, N_x :< N_t >_{K_{ts}}, T1$

According to message-meaning rule and S1, the following can be obtained

- S2: $R_i | \equiv Tag | \sim N_t$

Using Freshness-conjuncatenation rule and S2 to achieve the following

- S3: $R_i | \equiv Tag | \equiv N_t$

Using jurisdiction rule and S3, the following can be achieved

- S4:$R_i | \equiv N_t$

Using S4 and session key rule, the following can be achieved

- S5: $R_i | \equiv Tag \overset{AIDT_i}{\longleftrightarrow}$ Ri **(Goal 1)**

Using nonce-verification rule, the following is obtained

- S6: $R_i | \equiv Tag | \equiv Tag \overset{AIDT_i}{\longleftrightarrow}$ Ri **(Goal 2)**

M2: $R_i \to S_j : M1, N_y :< N_r >_{K_{rs}}, T2, V2$. Where, $T2$ is time-stamp of R_i
By using the seeing rule, we achieve

- S7: $S_j \lhd M1, N_y :< N_r >_{K_{rs}}, T2, V2$

By message-meaning rule and S7, the following can be achieved

- S8: $S_j | \equiv R_i | \sim N_r$

By Freshness-conjuncatenation rule and S8, the following can be computed

- S9: $S_j | \equiv R_i | \equiv N_r$

By applying the jurisdiction rule and S9, the following can be obtained

- S10:$S_j | \equiv N_r$

Using S10 and the SK rule, the following can achieved

- S11: $S_j | \equiv R_i \overset{AIDT_i}{\longleftrightarrow}$ Sj **(Goal 3)**

Using nonce-verification rule and S11, the following can be achieved

- S12: $S_j | \equiv R_i | \equiv R_i \overset{AIDT_i}{\longleftrightarrow}$ Sj. **(Goal 4)**

M3: S_j →Ri: $V3, V4, Zt < AIDT_{i_{new}} >^*_{K_{ts}}, T3$,$T3$ is time-stamp of S_j
By seeing-rule, the following can be achieved

- S13: $R_i \lhd V3, V4, Zt < AIDT_{i_{new}} >^*_{K_{ts}}, T3$

By message-meaning rule and S13, the following can be obtained

- S14: $R_i | \equiv S_j | \sim AIDT_{i_{new}}$

By S14 and the Freshness-conjuncatenation rule, the following can achieved

- S15: $R_i | \equiv S_j | \equiv AIDT_{i_{new}}$

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 26 July 2019 doi:10.20944/preprints201907.0298.v1

Peer-reviewed version available at Sensors 2019, 19, 4752; doi:10.3390/s19214752

By the assumption S15 and jurisdiction rule, the following can be achieved

- S16:$R_i| \equiv AIDT_{i_{new}}$

Using S16 and session-key rule, the following can be achieved

- S17: $R_i| \equiv S_j \stackrel{AIDT_{i_{new}}}{\longleftrightarrow}$ Ri. **(Goal 5)**

And applying nonce-verification rule, the following can be computed

- S18: $R_i| \equiv S_j| \equiv S_j \stackrel{AIDT_{i_{new}}}{\longleftrightarrow}$ Ri. **(Goal 6)**

M4: $R_i \rightarrow Tag : V4, Zt < AIDT_{i_{new}} >_{K_{ts}}, T4$, $T4$ is timestamp of R_i

Using seeing rule, the following can be computed

- S19: $Tag \lhd V4, Zt < AIDT_{i_{new}} \geq >_{ts}, T4$

Using message-meaning rule and S19, the following is achieved

- S20: $Tag| \equiv R_i| \sim AIDT'_{i_{new}}$

Using S20 and Freshness-conjuncatenation rule, the following can be obtained

- S21: $Tag| \equiv R_i| \equiv AIDT_{i_{new}}$

Using the jurisdiction rule and S21, the following can be achieved

- S22: $Tag| \equiv AIDT_{i_{new}}$

Using session-key rule, the following can be obtained

- S23: $Tag| \equiv R_i \stackrel{AIDT_{i_{new}}}{\longleftrightarrow} Tag$ **(Goal 7)**

Finally, using nonce-verification rule, the following can be achieved that also the final goal of the proposed protocol.

- S24: $Tag| \equiv R_i| \equiv R_i \stackrel{AIDT_{i_{new}}}{\longleftrightarrow} Tag$ **(Goal 8)**

Consequently, using the BAN logic it has been shown that Tag, R_i and S_j achieve mutual authentication successfully and securely attain the session key agreement.

4.2. Security Analysis with ProVerif

ProVerif software uses automated reasoning to test the security features of cryptographic and authentication protocols. It specifically checks the reachability, correspondence, and observational equivalence and supports primitive cryptographic operations [35]. The supported cryptographic functions include message authentication code MAC, digital signatures, encryption/decryption, elliptic curve cryptographic function hash functions and different other functions [36]. Furthermore, it is widely accepted tool for verification of security protocol in the research community.

In the proposed authentication protocol two main channels have been considered for communication, i.e., ChSec: Private channel, and ChPub: public channel. ChSec is a secure channel used for registration of RFID Tag IDT_i with RFID Server S_i through RFID Reader deviceR_i. ChPub is a public and insecure channel between RFID Tag IDT_i, RFID-Reader device R_i and RFID Server S_i. ChPub is a public and insecure channel between RFID Tag IDT_i, RFID-Reader device R_i and RFID Server S_i. ChPub used for authenticating RFID Tag IDT_i with the RFID Server S_i. All the participant including RFID Tag IDT_i, Reader R_i and Server S_i compute and verify session Key IDT_i. N_t, N_r and r_i are different nonce generated by each participant. K_ts and K_rs shared keys among the participants. Different function including XOR, Hash, Concatenation encryption and decryption function declared as constructors. The results obtained from ProVerif prove the correctness of the proposed protocol and the code is given below.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 26 July 2019 doi:10.20944/preprints201907.0298.v1

Peer-reviewed version available at Sensors 2019, 19, 4752; doi:10.3390/s19214752


```
(* --------- Channels ----------*)
free ChSec:channel [private].  (*secure channel between Tag and S*)
free ChPub:channel.                                  (*public channel between Tag,R and S*)
(*----------- Constants and Variables ---------*)
free IDTi :bitstring.
free IDs :bitstring.
free Ri:bitstring.
free Kts : bitstring [private].
(*========Constructors=======*)
fun h(bitstring):bitstring.
fun Inverse(bitstring):bitstring.
fun Concat(bitstring,bitstring):bitstring.
fun XOR(bitstring,bitstring):bitstring.
fun enc(bitstring,bitstring): bitstring.
fun dec(bitstring,bitstring): bitstring.
(*======Equations=======*)
equation forall a:bitstring; Inverse(Inverse(a))=a.
equation forall a:bitstring, b:bitstring; XOR(XOR(a,b),b)=a.
equation forall x: bitstring, y: bitstring; dec(enc(x,y),y) = x.
equation forall x: bitstring, y: bitstring; enc(dec(x,y),y) = x.
```

In ProVerif, the Tag process starts with the registration phase, where the Tag selects its identity ID_{Ti} and sends it to the server. The detailed code of this process is given as under.

```
(*---------------RFID TAG---------------------*)
(*=====*registration*======*)
let pTag=
(*     Registration   *)
out(ChSec,(IDTi));
in (ChSec,(IDTi:bitstring,Kts:bitstring,AIDTi:bitstring));
(*-------------TAG login--------------*)
event start_Tag(IDTi);
new Nt:bitstring;
new T1:bitstring;
let Nx=XOR(Kts,Nt) in
let V1=h(Concat(AIDTi, (Kts,Nx, Ri)))in
out(ChPub,(AIDTi,Nx,V1,T1));
in (ChPub,(V4:bitstring,T4:bitstring,Zt:bitstring));
let V4=h(Concat(Kts,(IDTi, Nt))) in
if V4=h(Concat(Kts,(IDTi, Nt))) then
let xAIDTinew = XOR(Zt,Kts) in
let AIDTi = AIDTinew in
event end_Tag(IDTi)
else
0.

(*--------------------RFID-Reader---------------------*)
let pR=
event start_R(Ri);
new Nr:bitstring;
new T2:bitstring;
new Krs:bitstring;
new Dj:bitstring;
new Ts:bitstring;
new MA1:bitstring;
in (ChPub,(xxAIDTi:bitstring,Nx:bitstring,V1:bitstring,T1:bitstring));
```

```
let Ny=XOR(Krs,Nr) in
let V2=Concat(MA1, (Nr,Krs)) in
out(ChPub,(Ny,Ri,V2,MA1,T2));
in (ChPub,(V3:bitstring,V4:bitstring,Zt:bitstring,T3:bitstring));
new ZT:bitstring;
new T4:bitstring;
if V3=h((Concat(Ri,(Nr,Krs)))) then
out(ChPub,(V4,T4,ZT));
event end_R(Ri)
else
0.

(*----------------------------RFID Server------------------------*)
let pS=
(*--------Registration phase--------*)
in (ChSec,(IDTi:bitstring));
new ns:bitstring;
new rti:bitstring;
new S:bitstring;
let Kts=XOR(h(Concat(IDTi, ns)),IDs) in
(*let AIDTi=enc(Concat(IDTi,rti),S) in*)
out(ChSec,(IDTi,Kts));
(*--------login-authentication-------------------*)
event start_S(IDs);
in (ChPub,(Ny:bitstring,Ri:bitstring,V2:bitstring,MA1:bitstring,T2:bitstring));
new Nx: bitstring;
new Krs: bitstring;
let Nt=Concat(Kts, Nx)in
let Nr=Concat(Krs, Ny)in
new T3: bitstring;
let V4= h(Concat(Kts,( IDTi,Nt))) in
let V3=h(Concat(Ri,( Nr, Krs))) in
new ri: bitstring;
let AIDTi=dec(Sx,(Concat(IDTi, ri))) in
new rinew: bitstring;
let AIDTinew= enc(Concat(IDTi,rinew),Sx)in
let ZT =XOR(AIDTinew, Kts) in
out(ChPub,(V3,V4,ZT,T3));
event end_S(IDs)
else
0.
```

The proposed protocol has been executed in parallel as shown below:

```
process ((!pS) | (!pR) | (!pTag) )
```

Authentication properties are verified for the proposed protocol with the help of following Queries:

```
(*-------queries------*)
free AIDTinew:bitstring [private].
query attacker(AIDTinew).
query id:bitstring; inj-event(end_Tag(IDTi)) ==> inj-event(start_Tag(IDTi)).
query id:bitstring; inj-event(end_R(Ri)) ==> inj-event(start_R(Ri)).
query id:bitstring; inj-event(end_S(IDs)) ==> inj-event(start_S(IDs)).
```

Six different events have been used in implementation of the proposed protocol for code including the *Tag*'s event (begin/end), Reader's *R* event (begin/end) and Server's *S* event (begin/end).

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 26 July 2019 doi:10.20944/preprints201907.0298.v1

Peer-reviewed version available at Sensors 2019, 19, 4752; doi:10.3390/s19214752


```
(*======*Events*======*)
event start_Tag(bitstring).
event end_Tag(bitstring).
event start_R(bitstring).
event end_R(bitstring).
event start_S(bitstring).
event end_S(bitstring).
```

After the compilation of the proposed protocol, ProVerif code the following result has been obtained:

```
1-- Query inj-event(end_S(IDs[])) ==> inj-event(start_S(IDs[]))
Completing...
Starting query inj-event(end_S(IDs[])) ==> inj-event(start_S(IDs[]))
RESULT inj-event(end_S(IDs[])) ==> inj-event(start_S(IDs[])) is true.
2-- Query inj-event(end_R(Ri[])) ==> inj-event(start_R(Ri[]))
Completing...
Starting query inj-event(end_R(Ri[])) ==> inj-event(start_R(Ri[]))
RESULT inj-event(end_R(Ri[])) ==> inj-event(start_R(Ri[])) is true.
3-- Query inj-event(end_Tag(IDTi[])) ==> inj-event(start_Tag(IDTi[]))
Completing...
Starting query inj-event(end_Tag(IDTi[])) ==> inj-event(start_Tag(IDTi[]))
RESULT inj-event(end_Tag(IDTi[])) ==> inj-event(start_Tag(IDTi[])) is true.
4-- Query not attacker(AIDTinew[])
Completing...
Starting query not attacker(AIDTinew[])
RESULT not attacker(AIDTinew[]) is true.
```

On the basis of the above description of results 1, 2, and 3, all the three main processes of the proposed protocol have successfully been started and terminated. Result 4 shows that the session key AID_{T_i} is safe from any adversary attack. Using ProVerif, it has been shown that the proposed protocol for RFID system preserves the secrecy and attains secure authentication.

4.3. Informal Security Analysis

The proposed protocol for RFID System has been analyzed for security loopholes against all known attacks. The list of attacks has been shown as follows:

1. Mutual Authentication between Tag and Server
2. Anonymity
3. Untraceability
4. Backward/Forward secrecy
5. Scalability
6. Collision attack
7. Denial of Service (DoS) attack
8. Replay attacks
9. Location tracking attack
10. Impersonation attack (Forgery attack)
11. Stolen-verifier attack

The proposed protocol has been analyzed against each of the above listed attacks. A brief description of each is presented in the following subsections.

4.3.1. Mutual Authentication Between Tag And Server

In the proposed authentication protocol, RFID server authenticates RFID Tag by verifying a one time alias AID_{T_i} and V_1 in the message M_1. Only legitimate RFID Tag can make a valid request message M_2 in M_1. On other side, RFID Tag can authenticate the legitimacy of the server using parameters V_4 and message M_3 in M_4. This way the proposed protocol achieves mutual authentication property.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 26 July 2019 doi:10.20944/preprints201907.0298.v1

Peer-reviewed version available at Sensors 2019, 19, 4752; doi:10.3390/s19214752

4.3.2. Anonymity

One of the basic principles of security is that an authentication protocol must not reveal the identity information of any participant (user or device) to an adversary. Anonymity is an essential factor of a secure protocol. Secure scheme guards personal information of a user so that an adversary or intruder cannot access any information that may lead to a security breach of the system. In the proposed protocol strong anonymity has been achieved. In the registration phase, RFID Tag registered itself with the Server S through RFID-reader using a secure channel. $M = \{ID_{T_i}, K_{ts}, AID\}$.

In Login and authentication phase of the proposed protocol, message $M_{A_1} = \{AID_{T_i}, N_x, V_1, T_1\}$ has been sent to the server S using public channel. Here if an adversary gets the message M_1 still the adversary cannot know identity of the RFID Tag, because AID_{T_i} is a one-time alias identity of the Tag. The original identity is kept encrypted in AID_{T_i} and can only be decrypted by the server using a shared secret Key K_{ts}. Thus, an adversary cannot reveal the RFID Tag's actual identity and hence the proposed protocol achieves anonymity.

4.3.3. Traceability

A genuinely secure protocol must not reveal any identifying information of the participants to an illegitimate user. The identifying information may lead to the traceability of the RFID Tag. The proposed protocol does not reveal any Login information of the current of or any previous sessions that lead to a security attack on RFID system. It is achieved through the use of different random numbers at different levels like N_t, N_r, r_i. Furthermore, a new one-time-alias identity for the RFID Tag AID_{Ti} has been used. Making it impossible for an adversary to guess any random number and launch an attack on the RFID system. Consequently, it can be been claimed that the proposed protocol makes RFID Tag untraceable.

4.3.4. Backward/Forward Secrecy

It is essential for security protocols that the information transmitted in a session must not be compromised as well as traced or used by an adversary to create vulnerabilities in the current, previous or future authentication session between RFID Tag and RFID Server S. In the proposed protocol it has been assumed that even if an attacker gets messages M_1, M_2, it still must not allow the attacker to extract any useful information that can be used to compromise next authentication session and create any vulnerability in the RFID System. It has been ensured through the use of encrypted AID_{Ti} that has been updated for every new session. In this way the proposed protocol for RFID System guarantees backward and forward secrecy.

4.3.5. Scalability

In the proposed protocol for RFID System, the RFID Server S does not perform an exhaustive process to authenticate any RFID-Tag. Instead the RFID-Server S process AID_{Ti} to validate the RFID Tag and responds back quickly to the RFID Tag. This makes the proposed protocol more scalable.

4.3.6. Collision Attack

If RFID-Tags share the same credentials for authentication to access the RFID server may leave the protocol vulnerable to collision attack. In the proposed protocol every RFID Tag uses different parameters i.e N_y, R_i, V_2, M_{A_1} for authentication that makes it impossible for collision attack to take place.

4.3.7. Denial Of Service (DoS) Attack

Since the protocol is not based on any random key that is responsible for authentication or verification of the RFID Tag. Rather it is based on AID_{Ti} that is well encrypted and updated for every transaction. Therefore, the proposed scheme resists any DoS attack.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 26 July 2019 doi:10.20944/preprints201907.0298.v1

Peer-reviewed version available at Sensors 2019, 19, 4752; doi:10.3390/s19214752

4.3.8. Replay Attacks

In a replay attack the attacker may delay or repeats the transmitted information for authentication with the server S. The proposed protocol for RFID systems has three participants including; Tag, Reader, and Server. For authentication, four messages are exchanged among each pair i.e, $\{M_1, M_2, M_3, M_4\}$ using public channel. Having access to the messages, an adversary A attempts to launch a replay attack form RFID-Tag to Server S. However, this attempt will fail as every message has been sent with a fresh time-stamp T. In case the time-stamp is invalid, the adversary A request will be rejected each time. Furthermore, if an adversary A can not compute other parameters of the message, the adversary still cannot launch the attack as all message parameters are updated for every new session by the participants of RFID System. Therefore, the proposed protocol for RFID systems is resistant to replay attack.

4.3.9. Location Tracking Attack

As the real identity of the RFID Tag is not sent directly in the message for authentication between the RFID-Tag and Server S, but it has been sent in an encrypted form that only the server can decrypt using its secret key. Moreover, the messages exchanged among the participants are constantly updated on every new session that provides unpredictability. Hence, an adversary cannot find location and any attempts of finding the location will ultimately fail.

4.3.10. Impersonation Attacks(Forgery Attacks)

An adversary A may intercept the messages of the previous legitimate RFID Tag and modify that for authentication with the RFID Server S. In this case the adversary A needs to make a valid message request that includes different parameters like $N_y, R_i, V_2, M_{A_1}, AID_{Ti}$. To do so the adversary A must compute AID_{Ti} that is well encrypted and impossible to be computed or forged. Moreover, the adversary A also needs different other parameters and timestamps to put valid request for authentication as legitimate RFID Tag. It is impossible for the adversary A without knowing the actual parameters of the Message used for authentication and hence the adversary A cannot prove his legitimacy as RFID Tag to the RFID Server S. Resultantly, the proposed protocol for RFID System resists any forgery attack.

4.3.11. Stolen-Verifier Attacks

The proposed protocol resists stolen-verifier-attack. All the verification and validation keys are stored encrypted in the RFID Database Server S. If the data and keys are stolen form the RFID database Server S, still the adversary A cannot decrypt and extract them. Also, the adversary A cannot alter or modify the original data saved in the RFID Database Server S. Hence, the proposed protocol resists any stolen-verifier attack.

5. Comparative Analysis

This section presents a comprehensive comparative analysis of the proposed protocol with the existing protocols. Firstly, the proposed protocol has been compared with the existing protocols in terms of security requirements. Secondly, a comparison of the proposed protocol with existing protocols based on computation cost(running time or execution time) and thirdly, a comparison based on communication cost has been presented. Furthermore, the proposed protocol has also been analyzed for storage complexity. Each of these comparison has been elaborated in the following subsections one by one.

5.1. Security Requirements

Security requirements are the features expected from an authentication protocol. Every authentication protocol must be able to ensure these features or requirements. By these requirements,

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 26 July 2019 doi:10.20944/preprints201907.0298.v1

Peer-reviewed version available at Sensors 2019, 19, 4752; doi:10.3390/s19214752

the proposed protocol compared with the existing protocols. Following is the list of features/ requirements considered for comparative analysis.

- SR1: Mutual authentication.
- SR2: Tag untraceability.
- SR3: Tag anonymity.
- SR4: Backward/Forward secrecy.
- SR5: Scalability.
- SR6: Collision attacks.
- SR7: DoS attacks.
- SR8: Replay attacks.
- SR9: Location tracking attack.
- SR10: Forgery attack.
- SR11: Stolen-verifier attacks.

The proposed protocol compared with existing protocols from the above-listed requirements.The existing protocols considered for comparison proposed in various articles including [19,21,23,30, 31].Note that SR is a security requirement, so SR1 is the security requirement number 1 in the list and so on. Comparison based on the security requirements shown in Table 4.

Table 4. Security requirements table

Requirements	Yang et al. [23]	Tan et al. [30]	Cai et al. [21]	Cho et al. [31]	Gope et al. [19]	Proposed Scheme
SR1	×	×	✓	✓	✓	✓
SR2	×	×	×	✓	✓	✓
SR3	×	×	✓	×	✓	✓
SR4	×	✓	×	✓	✓	✓
SR5	×	×	×	×	✓	✓
SR6	×	×	×	✓	×	✓
SR7	✓	×	✓	✓	×	✓
SR8	✓	✓	✓	✓	✓	✓
SR9	✓	✓	✓	✓	✓	✓
SR10	✓	✓	✓	✓	✓	✓
SR11	✓	✓	✓	✓	×	✓

✓: Yes provides, ×: Does not provide

It is clear from Table 4 that the proposed protocol is successful in providing all the requirements in comparison to the existing protocols.

5.2. Computation Cost Analysis

The computation complexity/ cost is basically the time required by a protocol to execute one time. It is computed by multiplying the significant operations by their respective frequencies and then sum the costs of all operations. Some operation with significantly low execution time for example the concatenations, and XOR operations have been neglected [37]. So the computation cost is computed only in terms of operations with significant execution time requirements for example cryptographic functions and operations that the Tag, R_i and S_j need to execute. Operations at three main participants Tag, R_i and S_j have been considered. The list of operation with description has been listed as follows;

- CC:Computation cost
- T_h: CC of single hash function;
- T_{me}: CC of modular exponentiation;
- T_{se}: CC of symmetric encryption;
- T_{sd}: CC of symmetric decryption;
- CC_{mbu}: Computation-cost of Tag
- CC_{HA} : Computation-cost of $Reader R_i$
- CC_{FA} : Computation-cost of $Server S$
- CC_{total} : Total computation-cost.

The proposed protocol has been compared with the protocols presented recently in [19,21,23,30,31]. The detailed comparative analysis has been presented in Table 5. From the table it can be seen that the

Table 5. Comparison of computation cost and running time

Computation Cost	Yang et al. [23]	Tan et al. [30]	Cai et al. [21]	Cho et al. [31]	Gope and Hwang [19]	Proposed Scheme
CC_{Tag}	$2T_h$	$2T_h$	$4T_h$	$3T_h$	$5T_h$	$2T_h + 1T_{se}$
CC_{R_i}	$3T_h$	$2T_h$	$2T_h$	$2T_h$	$2T_h$	$1T_h$
CC_S	$5T_h$	$3T_h$	$6T_h$	$5T_h$	$7T_h$	$3T_h + 1T_{se}$
CC_{Total}	$10T_h$	$7T_h$	$12T_h$	$10T_h$	$14T_h$	$6T_h + 2T_{se}$
CC_{Time}	0.023ms	0.0161ms	0.0276ms	0.023ms	0.0322ms	0.023ms

computation cost of protocol in [23] is $2T_h + 3T_h + 5T_h$, that in total is equal to $10T_h$. The computational cost of protocol presented in [30] is $2T_h + 2T_h + 3T_h$, that is equal to $7T_h$. Similarly, the computation cost of the protocol presented in [21] is $4T_h + 2T_h + 6T_h$ that in total is $12T_h$, and the computation cost of Gope and Hwang protocol presented in [19] is $5T_h + 2T_h + 7T_h$ in total that is equal to $14T_h$. While the computation cost of the proposed protocol uses $2T_h + 1T_{se} + 1T_h + 3T_h + 1T_{se}$ hash function, and in total the computational cost of the proposed is equal to $6T_h$.

The calculated running time is recorded using an Intel Pentium dual-core PC with processor 2.20 GHz (E2200) possessing 2048 MB of RAM. The operating system is a 32-bit Ubuntu version 12.04.1 is utilized. Running time of the protocols is in milliseconds (ms). The total time for all protocols in Table 5 have also been computed in millisecond (ms) following the Kilinc and Yanik [38] experiments. According to [38], a single T_h cost $0.0023ms$, Tecmp cost $2.226ms$, Tecpa cost $0.0288ms$, T_{me} cost $3.8500ms$. Therefore, based on [38], the total computational time of the proposed protocol is $0.023ms$, whereas the total cost of the protocol presented in [23] is $0.0230ms$, the cost of protocol in [30] is proximately $0.0161ms$, the cost of the proposal in [19] is $0.0322ms$, and the proposal in [21] takes a total of $0.0276ms$. It has been observed that the proposed protocol is very efficient and outperforms some of the existing protocols in terms of computation cost. In some cases, however, it achieves either less efficiency or equal efficiency to some existing protocols. In such case, it must also be noted that the existing protocol though having equal or more effective than the proposed protocol, but at the same time those protocols are vulnerable to various security threats/attacks whereas the proposed protocol is secured against all known attacks. This comparative analysis has been depicted in Figure 6.

It can be clearly observed from Figure 6 that the proposed protocol out performs the baseline protocol in terms of computation cost. The proposed protocol not only overcomes the security shortcomings of the baseline protocol but also it is 28.57% more efficient computationally in comparison to the baseline protocol.

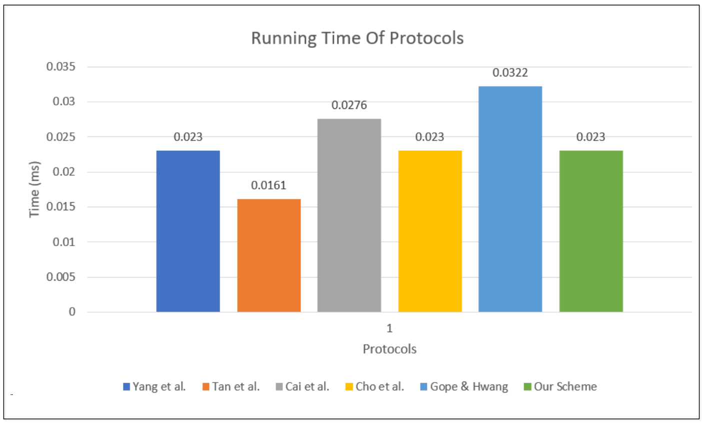

Figure 6. Running Time of Proposed Scheme

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 26 July 2019 doi:10.20944/preprints201907.0298.v1
Peer-reviewed version available at Sensors 2019, 19, 4752; doi:10.3390/s19214752

5.3. Communication and Storage Cost Analysis

Communication cost presented in term of the total number of messages exchanged, or the total number of bits exchanged during one transaction of the protocol. For communication cost of the proposed protocol again the three main participants; RFID Tag, Database Server, and the Reader have been considered. All the messages exchanged among the three participants during one transaction of the protocol have been considered. In the proposed protocol, the RFID *Tag* transmits four parameters in M_1 to RFID Reader R_i that is equal to $416bits$ and receives $384bits$ from the RFID reader R_i. Similarly, the RFID Reader R_i transmits $736bits$ and receives $416bits$ from RFID Server S while RFID Server S transmits $416bits$ and receives $736bits$. The communication cost of security protocol has been computed considering a total number of exchanged messages or the total number of bit in both directions. As in terms of messages, a total of four messages have been exchanged during a successful authentication by the proposed protocol among all participants. The communication cost comparison of the proposed protocol with other existing protocols presented in Table 6.

The storage cost is represented by length Value L. The proposed protocol uses SHA-1 hash function to implement $h(.)$, then each of the length value is 160-bit long. In the proposed scheme each of the Tag stored ID_{T_i}, K_{ts}, AID parameters. Therefore, the cost of storage in the Tag is $3L$. Whereas on server side $ID_{T_i}, K_{ts}, AID_{new}, AID_{old}$ are being stored, hence the storage cost on server side is $4L$ per Tag.

Table 6. Communication Cost of Proposed and other Protocols

Schemes	Tag	Reader	Server	Total Bits	Messages
Yang et al. [23]	256	512	640	1408	5
Tan et al. [30]	896	768	768	2432	4
Cai et al. [21]	256	544	256	1056	5
Cho et al. [31]	512	512	256	1280	5
Gope and Hwang [19]	416	1180	288	1888	4
Proposed Protocol	416	736	416	1568	4

It can be clearly observed from the table that the proposed protocol performs better than most of the existing protocols. However, some of the existing protocol also have the same communication cost as the proposed protocol. However, it must be remembered that the proposed protocol protects against all known attacks with the same communication cost whereas the existing protocols are still vulnerable to various security threats. The behavior in terms of the communication cost of protocol along with the existing protocols has been depicted in Figure 7.

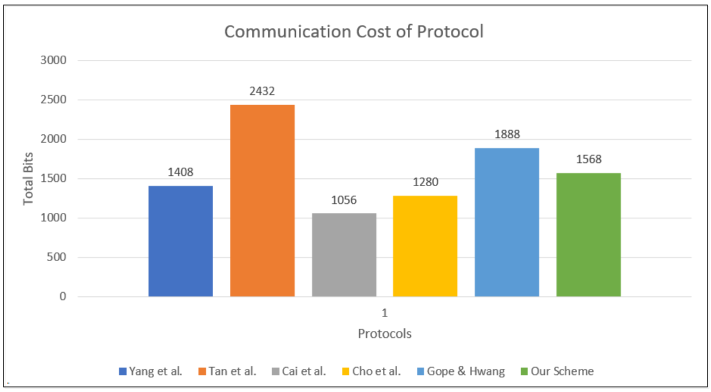

Figure 7. Communication Cost

Figure 7 indicates that the proposed protocol is efficient than the baseline protocol in terms of communication cost. Specifically, the proposed protocol not only overcomes the security flaws of the baseline protocol but also achieves 16.94% efficiency in terms of the number of bits exchanged during one transaction of the protocol.

6. Conclusion

In this article, a secure authentication protocol for IoT based RFID-System has been presented that resists all known security attacks and specifically those attacks that were successful against the baseline (Gope and Hwang) protocol. A detailed cryptanalysis of the baseline protocol has been presented proving its security vulnerabilities. The proposed lightweight protocol addresses the vulnerabilities of the baseline protocol and has been proved to be robust, realistic and anonymous authentication protocol for IoT based RFID-Systems. The proposed protocol has been formally analyzed using BAN logic and ProVerif to prove message freshness property and security of session key. Furthermore, it has also been comparatively analyzed for with existing protocol using the security requirements. The protocol has also been analyzed for communication cost, Computation cost, and storage cost and can be argued as an efficient, realistic, and enhanced authentication protocol in comparison to the baseline protocol. In the future, the proposed protocol may be investigated and analyzed to be deployed in other IoT based systems like Wireless Sensor Network, Wireless Health care systems and other similar fields.

References

1. Gope, P.; Hwang, T. BSN-Care: a secure IoT-based modern healthcare system using body sensor network. *IEEE Sensors Journal* **2016**, *16*, 1368–1376.
2. Gope, P.; Amin, R.; Islam, S.H.; Kumar, N.; Bhalla, V.K. Lightweight and privacy-preserving RFID authentication scheme for distributed IoT infrastructure with secure localization services for smart city environment. *Future Generation Computer Systems* **2018**, *83*, 629–637.
3. Peris-Lopez, P.; Hernandez-Castro, J.C.; Estevez-Tapiador, J.M.; Ribagorda, A. Lightweight cryptography for low-cost RFID tags. *Security in RFID and Sensor Networks* **2016**, pp. 121–150.
4. Gope, P.; Amin, R.; Islam, S.H.; Kumar, N.; Bhalla, V.K. Lightweight and privacy-preserving RFID authentication scheme for distributed IoT infrastructure with secure localization services for smart city environment. *Future Generation Computer Systems* **2018**, *83*, 629–637.
5. Kitsos, P. *Security in RFID and sensor networks*; CRC Press, 2016.
6. Hsu, C.H.; Wang, S.; Zhang, D.; Chu, H.C.; Lu, N. Efficient identity authentication and encryption technique for high throughput RFID system. *Security and Communication Networks* **2016**, *9*, 2581–2591.
7. Simon, P.M.G.; Riggert, E.F.; Trivelpiece, S.E. System and method for reading RFID tags across a portal, 2016. US Patent 9,519,811.
8. Wu, F.; Xu, L.; Kumari, S.; Li, X.; Das, A.K.; Shen, J. A lightweight and anonymous RFID tag authentication protocol with cloud assistance for e-healthcare applications. *Journal of Ambient Intelligence and Humanized Computing* **2018**, *9*, 919–930.
9. Sidorov, M.; Ong, M.T.; Vikneswaran, R.; Nakamura, J.; Ohmura, R.; Khor, J.H. Ultralightweight Mutual Authentication RFID Protocol for Blockchain Enabled Supply Chains. *IEEE Access* **2019**.
10. Noman, A.T.; Hossain, S.; Islam, S.; Islam, M.E.; Ahmed, N.; Chowdhury, M.M. Design and Implementation of Microcontroller Based Anti-Theft Vehicle Security System using GPS, GSM and RFID. 2018 4th International Conference on Electrical Engineering and Information & Communication Technology (iCEEiCT). IEEE, 2018, pp. 97–101.
11. Liao, Y.P.; Hsiao, C.M. A secure ECC-based RFID authentication scheme integrated with ID-verifier transfer protocol. *Ad Hoc Networks* **2014**, *18*, 133–146.
12. Kim, H. RFID mutual authentication protocol based on synchronized secret. *International Journal of Security and Its Applications* **2013**, *7*, 37–50.
13. Cha, J.R.; Kim, J.H. Novel anti-collision algorithms for fast object identification (RFID) system. Parallel and Distributed Systems, Proceedings. 11th International Conference. IEEE, 2005, Vol. 2, pp. 63–67.
14. El Beqqal, M.; Azizi, M. Classification of major security attacks against RFID systems. International Conference on Wireless Technologies, Embedded and Intelligent Systems (WITS). IEEE, 2017, pp. 1–6.

15. Tewari, A.; Gupta, B. Cryptanalysis of a novel ultra-lightweight mutual authentication protocol for IoT devices using RFID tags. *The Journal of Supercomputing* **2017**, *73*, 1085–1102.
16. Ayaz, U.; Haq, T.A.; Taimour, S.; Mansoor, K.; Mahmood, S. An Enhanced Biometric Based RFID Authentication Scheme Defending Against Illegitimate Access. 14th International Conference on Emerging Technologies (ICET). IEEE, 2018, pp. 1–6.
17. Kolhe, P.; Dharaskar, R.; Tharkar, M.; Joshi, S.; Desai, S.; Dapoli, B. Information technology tool in library barcode & Radio Frequency Identification RFID. *Int. J. Innov. Sci. Eng. Technol* **2016**, *3*, 81–6.
18. Sidorov, M.; Ong, M.T.; Sridharan, R.V.; Nakamura, J.; Ohmura, R.; Khor, J.H. Ultralightweight Mutual Authentication RFID Protocol for Blockchain Enabled Supply Chains. *IEEE Access* **2019**, *7*, 7273–7285. doi:10.1109/ACCESS.2018.2890389.
19. Gope, P.; Hwang, T. A realistic lightweight authentication protocol preserving strong anonymity for securing RFID system. *Computers & Security* **2015**, *55*, 271–280.
20. Burmester, M.; De Medeiros, B.; Motta, R. Robust, anonymous RFID authentication with constant key-lookup. Proceedings of the 2008 ACM symposium on Information, computer and communications security. ACM, 2008, pp. 283–291.
21. Cai, S.; Li, Y.; Li, T.; Deng, R.H. Attacks and improvements to an RIFD mutual authentication protocol and its extensions. Proceedings of the second ACM conference on Wireless network security. ACM, 2009, pp. 51–58.
22. Gaubatz, G.; Kaps, J.P.; Ozturk, E.; Sunar, B. State of the art in ultra-low power public key cryptography for wireless sensor networks. Pervasive Computing and Communications Workshops, PerCom Workshops. Third IEEE International Conference on. IEEE, 2005, pp. 146–150.
23. Yang, J.; Park, J.; Lee, H.; Ren, K.; Kim, K. Mutual authentication protocol. Workshop on RFID and lightweight crypto, 2005.
24. Kang, S.Y.; Lee, I.Y. A Study on low-cost RFID system management with mutual authentication scheme in ubiquitous. Asia-Pacific Network Operations and Management Symposium. Springer, 2007, pp. 492–502.
25. Lee, L.S.; Fiedler, K.D.; Smith, J.S. Radio frequency identification (RFID) implementation in the service sector: A customer-facing diffusion model. *International Journal of Production Economics* **2008**, *112*, 587–600.
26. Qingling, C.; Yiju, Z.; Yonghua, W. A minimalist mutual authentication protocol for RFID system & BAN logic analysis. International Colloquium on Computing, Communication, Control, and Management, CCCM. IEEE, 2008, Vol. 2, pp. 449–453.
27. Zhou, S.; Zhang, Z.; Luo, Z.; Wong, E.C. A lightweight anti-desynchronization RFID authentication protocol. *Information Systems Frontiers* **2010**, *12*, 521–528.
28. Piramuthu, S. RFID mutual authentication protocols. *Decision Support Systems* **2011**, *50*, 387–393.
29. Safkhani, M.; Peris-Lopez, P.; Hernandez-Castro, J.C.; Bagheri, N. Cryptanalysis of the Cho et al. protocol: A hash-based RFID tag mutual authentication protocol. *Journal of Computational and Applied Mathematics* **2014**, *259*, 571–577.
30. Tan, C.C.; Sheng, B.; Li, Q. Secure and serverless RFID authentication and search protocols. *IEEE Transactions on Wireless Communications* **2008**, *7*, 1400–1407.
31. Cho, J.S.; Jeong, Y.S.; Park, S.O. Consideration on the brute-force attack cost and retrieval cost: A hash-based radio-frequency identification (RFID) tag mutual authentication protocol. *Computers & Mathematics with Applications* **2015**, *69*, 58–65.
32. Gong, L.; Needham, R.; Yahalom, R. Reasoning about belief in cryptographic protocols. Proceedings of IEEE Computer Society Symposium on Research in Security and Privacy. IEEE, 1990, pp. 234–248.
33. Chaudhry, S.A.; Naqvi, H.; Farash, M.S.; Shon, T.; Sher, M. An improved and robust biometrics-based three factor authentication scheme for multiserver environments. *The Journal of Supercomputing* **2015**, pp. 1–17.
34. Kyntaja, T. A logic of authentication by Burrows, Abadi and Needham. *Science Helsinki University of Technology, Tehran. http://www. tml. tkk. fi/Opinnot/Tik-110.501/1995/ban. html* **1995**.
35. Blanchet, B.; others. Modeling and verifying security protocols with the applied pi calculus and ProVerif. *Foundations and Trends® in Privacy and Security* **2016**, *1*, 1–135.
36. Lumini, A.; Nanni, L. An improved biohashing for human authentication. *Pattern recognition* **2007**, *40*, 1057–1065.
37. Zhang, G.; Fan, D.; Zhang, Y.; Li, X.; Liu, X. A privacy preserving authentication scheme for roaming services in global mobility networks. *Security and Communication Networks* **2015**, *8*, 2850–2859.

38. Kilinc, H.H.; Yanik, T. A survey of SIP authentication and key agreement schemes. *IEEE Communications Surveys & Tutorials* **2014**, *16*, 1005–1023.