

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 Islamabad; kh.mansoorulhassan@gmail.com

² Department of Computer Science & Software Engineering, International Islamic University Islamabad; anwar.ghani@iiu.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 Internet 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 scarce resource of RFID Tags, an RFID based could be an attractive target for attacker. Therefore,

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

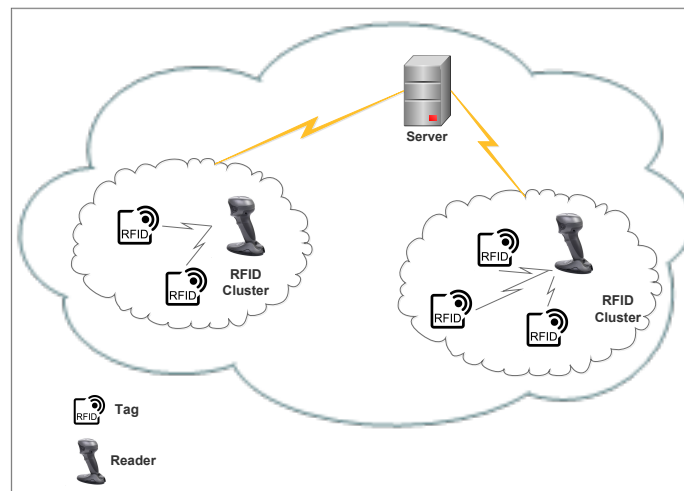


Figure 1. RFID System Architecture

27 RFID technology is most simple form of pervasive sensor networks that are commonly used for
 28 identification of different physical objects [3,4]. It uses a two way radio signals receiver and transmitter
 29 called reader or the interrogators. Radio signals is sent to the tag attached to the physical object and
 30 the interrogator or reader is responsible to respond [5,6]. RFID reader device is a network connected
 31 device (mobile or fixed) along with an antenna which is responsible to transmit power, data as well as
 32 commands to the objects tags [7-9]. RFID reader device is an access point for the RFID-Tags clusters
 33 that is responsible for the availability of the tag information to different System application. Tags are
 34 of two types, passive tags in which energy is transferred using radio frequency from reader device and
 35 active tags they have internal energy source like power batteries. Different features are described in
 36 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

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

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

50 based system is very crucial to ensure authenticity of the information exchanged between different
51 RFID enabled devices [8].

52 As IoT based RFID systems remains an emerging technology consisting of limited resource
53 devices, it is vulnerable to various security threats like data threats, authentication, and security threats
54 [14–16]. Due to the broad application domain, RFID must be more secure and protected to meet the
55 industrial standards. For reliable security of RFID, different types of features in security protocols
56 should be considered; 1) security schemes of RFID must provide user friendliness, 2) user anonymity
57 must be ensured to protect the identity of legitimate users, 3) there should be a provision for change
58 moreover, update of tag data at any time securely, 4) ensure backward and forward secrecy in RFID
59 tag reading-writing process, 5) protection of an RFID system from malicious system users to prevent
60 insider attacks, 6) having capabilities to endure replay attacks, 7) having capabilities to withstand the
61 impersonation and forgery attacks, 8) having capabilities to achieve secure mutual authentication, and
62 8) having capabilities to endure man-in-middle attacks.

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

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

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

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

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

91 2. Literature Survey

92 In literature, authentication in RFID based system has been rigorously investigated. Recently,
93 various anonymous mutual authentication security schemes/protocols have been proposed for
94 RFID Systems [11,12,19–31]. The protocols in literature have been categorized into non-public
95 key-cryptosystem (NPKC) and public-key-crypto system (PKC) security protocol. Also, other
96 researchers came up with Asymmetric/public key cryptosystem (PKC) solution based on elliptic
97 curve cryptosystem (ECC). Due to the modular exponential operations having very high computation

98 cost, the elliptic curve cryptosystem (ECC) based security protocols are infeasible for low-powered
99 RFID systems. However, using the symmetric key cryptosystem (NPKC) solution – hash based RFID
100 Systems may be a feasible solution due to their equitable computational overhead.

101 In 2005, the authors in [23] proposed a symmetric public key crypto (PKC) based protocol
102 for RFID Systems that was mainly designed using exclusive-OR and a one-way Hashing function.
103 The authors claimed that their protocol addresses various security issues faced by the RFID-Tag
104 System. Unfortunately, their scheme is vulnerable to different security threats including forgery attack,
105 man-in-the-middle attack, and traceability issue. subsequently, various security protocols including;
106 [26], [30], and [21] were designed using hash function encryption, and exclusive-OR encryption.
107 However, they proved to be exposed to various security attacks like DoS attack, forgery attack [28].

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

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

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

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

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

Table 2. Notation Guide

Notations	Description
T	RFID-tag
R	Reader Device
S	Database Server System
ID_{T-i}	i th Tag identity
AID_T	One-time Tag alias identity
SID	Shadow identity
R_j	j th 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

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

151 2.1. Adversarial Model

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

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

165 2.2. Review of Baseline Protocol

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

173 2.3. Baseline Protocol Registration Phase

174 **Step-1:** Every RFID Tag submit its ID_{T_i} to the RFID Database S.
 175

176 **Step-2:** Database Server generates random number n_s and computes $K_{ts} = h(ID_{T_i} || n_s \oplus ID_s)$.
 177
 178 **Step-3:** S generates set of unlikeable shadow ID_s , and $SID = \{sid_1, sid_2 \dots\}$, where the
 179 $sid_j \in SID$. Database Server computes $sid_j = h(ID_{T_i} || r_j || K_{ts})$. Then it generates a set of emergency
 180 keys $K_{emg} = \{k_{emg_1}, k_{emg_2} \dots\}$, each of the key corresponding to $sid_j \in SID$ where each $k_{emg_i} \in K_{emg}$.
 181 S then computes $k_{emg_i} = h(ID_{T_i} || sid_j || r_j)$.
 182
 183 **Step-4:** The Database Server S generates a 32-bit random sequence number Tr_{seq} . Then S
 184 generates random number m and matches it with Tr_{seq} , $Tr_{seq} = m$, and send the Tr_{seq} to the RFID Tag
 185 through Reader R by maintaining the copy of Tr_{seq} in its database for speeding up the authentication
 186 process.
 187
 188 **Step-5:** The RFID Server S authenticates the validity of RFID Tag ID_{T_i} based on Tr_{seq} . If Tr_{seq}
 189 does not have a match within the record of RFID Server S , then the Server S does not authenticate
 190 RFID Tag and terminate the process. In this case the RFID Tag ID_{T_i} will use one of its fresh pair
 191 of the emergency key $k_{emg_j} \in K_{emg}$ and shadow ID $sid_j \in SID$. The used pair of shadow ID and
 192 emergency ID (SID, K_{emg}) must be deleted from both, the Database Server S and the RFID Tag ID_{T_i} .
 193 Database Server S again updates and send $\{K_{ts}, (SID, K_{emg}), Tr_{seq}, h(\cdot)\}$ through secure channel for
 194 further communication. The registration phase of the baseline protocol has be shown in Figure 2.

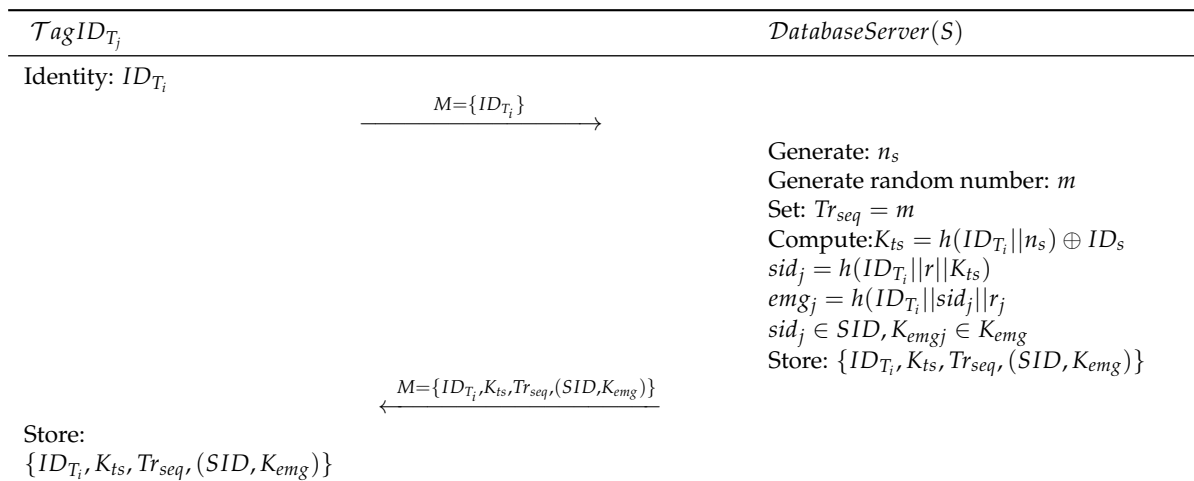


Figure 2. Gope-Hwang's proposed registration scheme

195 2.4. Baseline Protocol Tag Authentication Phase

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

198 **Step-1:** RFID Tag ID_{T_i} generates random number N_t , then derives $AID_T = h(ID_{T_i} || K_{ts} || N_t || Tr_{seq})$,
 199 $N_x = K_{ts} \oplus N_t$ and computes $V_1 = h(AID_{T_i} || K_{ts} || N_x || R_i)$. RFID Tag sends a message request as M_{A_1}
 200 to the Reader device R_i and the Reader R_i also receives a recently used sequence number from the
 201 Server S for mutual authentication. In the case of synchronization loss, the RFID Tag uses one of its
 202 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
 203 a message to the Reader R as $M_{A_1} : \{AID_T, N_x, Tr_{seq}, V_1\}$.

204 **Step-2:** Upon receiving request from RFID Tag, reader of the cluster (in which Tag is located)
 205 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
 206 message to the Database Server S for verification. $M_{A_2} : R_i \rightarrow S\{N_y, R_i, V_2, M_{A_1}\}$.

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

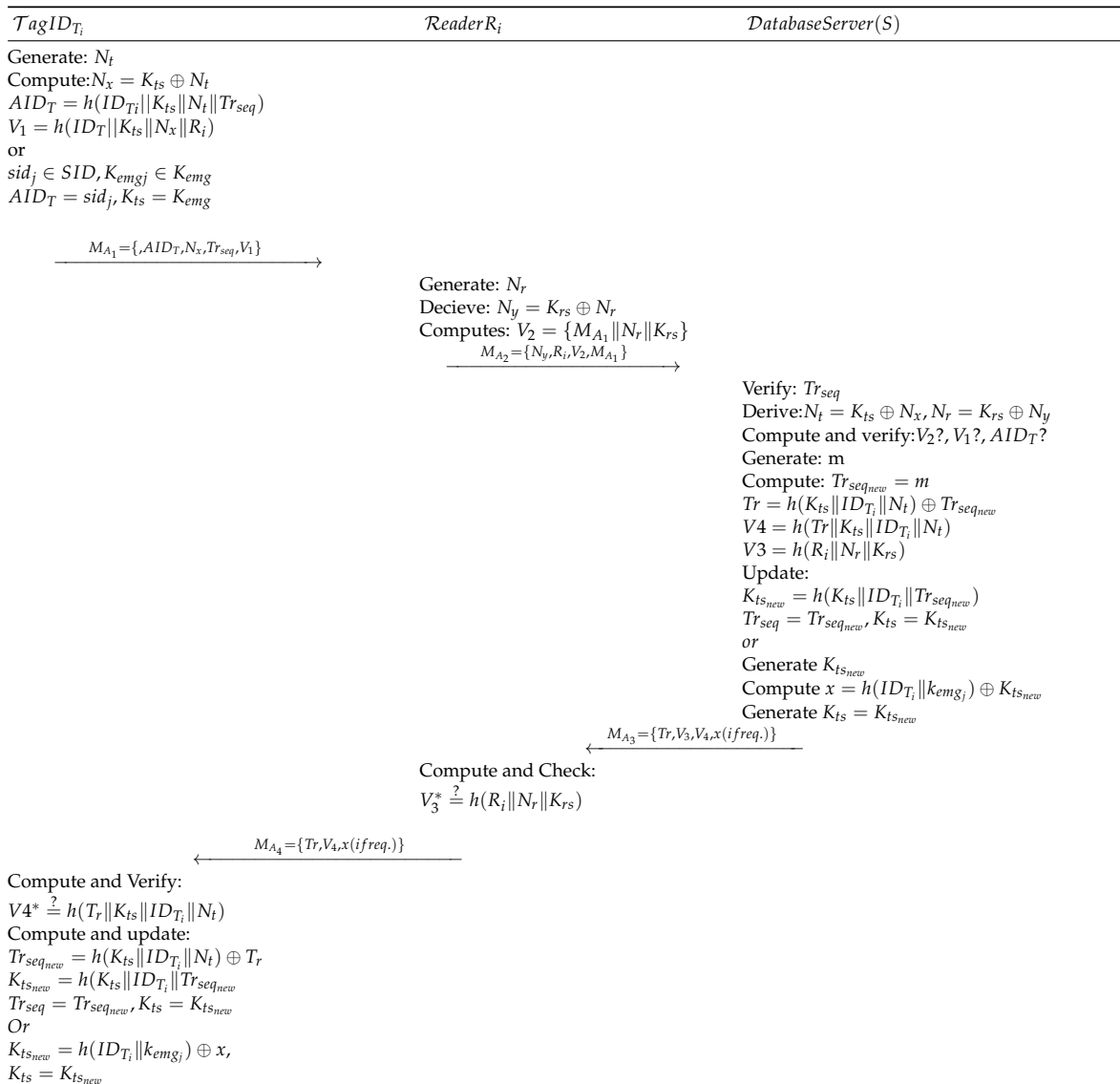


Figure 3. Gope-Hwang's proposed authentication scheme

209 $N_t = K_{ts} \oplus N_x$ and verify AID_T . Upon successful verification of AID_T , the Database Server S
 210 generates a random number m and assigns it to $Tr_{seq} = m$. It also computes $Tr = h(K_{ts} || ID_{T_i} || N_t) \oplus Tr_{seq}$,
 211 $V_4 = h(Tr || K_{ts} || ID_{T_i} || N_t)$, $V_3 = h(R_i || N_r || K_{rs})$ to creates a Message M_{A_3} and send it to the Reader R.
 212 Finally, the Database Server S computes $K_{ts_{new}} = h(K_{ts} || ID_{T_i} || Tr_{seq_{new}})$ and updates $K_{ts_{new}}$ and $Tr_{seq_{new}}$.
 213 In case the message M_{A_1} dose not contain Tr_{seq} , then the Database Server S randomly generates
 214 a new shared key $K_{ts_{new}}$ using the emergency key K_{emg_j} and real identity of the RFID Tag ID_{T_i} .
 215 $x = K_{ts_{new}} \oplus h(ID_{T_i} || K_{emg_j})$ is computed and x is sent within the message M_{A_4} . Where V_4 is calculated
 216 as $V_4 = h(N_t || Tr || x || K_{emg_j})$. $M_{A_3} : S \rightarrow R_i : \{Tr, V_3, V_4, x(ifreq.)\}$.

217 **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
 218 V_3 . If it is equal RFID Reader R sends M_{A_4} to the Tag. Contrarily, the Reader R terminates the
 219 established connection. RFID Tag receives the message M_{A_4} and computes $h(Tr || K_{ts} || ID_{T_i} || N_t)$ and
 220 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
 221 $K_{ts} = K_{ts_{new}}, Tr_{seq} = Tr_{seq_{new}}$ for future communication. $M_{A_4} : R_i \rightarrow ID_{T_i} : \{Tr, V_4, x(ifreq.)\}$. The
 222 detailed stepwise representation of the baseline protocol has been shown in Figure 3.

223 2.5. Cryptanalysis of Baseline Protocol

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

229 The Database Server receives the message M_{A_2} from the reader that contains different information
 230 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
 231 Server (S) for authenticating the particular RFID Tag is to check its Tr_{seq} number. If the received Tr_{seq}
 232 number of RFID Tag does not matches, the authentication process terminates, otherwise it undergoes
 233 further computation. Here the proposed protocol in [19] is totally dependent on the Tr_{seq} number and
 234 the Tr_{seq} number is generated randomly.

235 2.5.1. Vulnerable to Collision Attack

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

240 2.5.2. Vulnerable to Stolen Verifier Attack

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

248 2.5.3. Vulnerable to Denial-Of-Service (DoS) Attack

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

252 3. Proposed Scheme

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

260 3.1. Tags Registration Phase

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

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

264

265 **Step-2:** The Database Server Generates a random number n_s and computes $K_{ts} =$
 266 $h(ID_{T_i} \| n_s \oplus ID_s)$.

267
 268 **Step-3:** The Database Server S also computes one-time alias Tag identity AID_T by encrypting it
 269 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.

270
 271 **Step-4:** The Database Server S authenticates the RFID Tag ID_{T_i} based on AID_T in Tags
 272 authentication Phase by checking if a request is valid or not.

273
 274 **Step-5:** The Database Server stores and sends a message M to the RFID Tag through a secure
 275 channel. $M = \{ID_{T_i}, K_{ts}, AID\}$. Upon receiving the message from database server S , the RFID Tag
 276 stores the information in its memory.

277
 278 The detailed steps of the proposed registration phase have been shown in Figure 4.

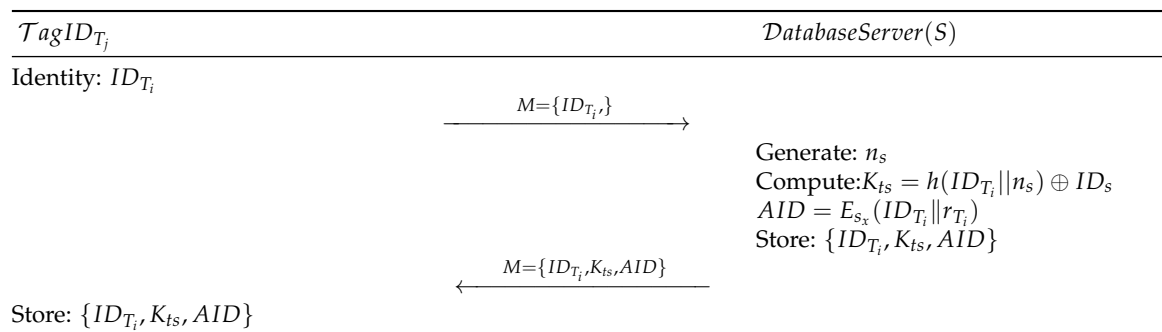


Figure 4. Registration phase of the proposed protocol

279 3.2. Tags Authentication Phase

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

283 **Step-1:** RFID Tag ID_{T_i} generates a random number N_t , then derives $N_x = K_{ts} \oplus N_t$ and
 284 $V_1 = h(AID_t \| K_{ts} \| N_x \| R_i)$. RFID Tag makes a request through the message $M_{A_1} : \{AID_T, N_x, T_1, V_1\}$
 285 to Reader device R_i for mutual authentication.

286
 287 **Step-2:** Upon receiving the request from RFID Tag, reader of the cluster (in which Tag is located)
 288 first verifies and checks the freshness of the message as $(T_2 - T_1) \leq \Delta T$. Then the Reader generates
 289 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
 290 message to the Database Server S for verification. $M_{A_2} : R_i \rightarrow S\{N_y, R_i, V_2, M_{A_1}, T_2\}$.

291
 292 **Step-3:** When the Database Server S receives the request from the Reader R , first it verifies
 293 $(T_3 - T_2) \leq \Delta T$, then derives $N_t = K_{ts} \oplus N_x$ and $N_r = K_{rs} \oplus N_y$. Further, the Database Server
 294 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
 295 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
 296 Server computes $V_3 = h(R_i \| N_r \| K_{rs} \| T_3)$ and $V_4 = h(K_{ts} \| ID_{T_i} \| N_t \| T_3)$. Upon successful verification
 297 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
 298 a message M_{A_3} and sends it to the Reader R . $M_{A_3} : S \rightarrow R_i : \{V_3, V_4, Z_T, T_3\}$

299
 300 **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
 301 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

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

303

304 **Step-5:** The RFID Tag receives a message M_{A_4} and checks its freshness. After that the message
 305 is verified as $V_4^* \stackrel{?}{=} h(K_{ts} \| ID_{T_i} \| N_i)$. After that RFID Tag computes and updates $AID_{T_i(new)} =$
 306 $(Z_T \oplus K_{Ts})$, $AID_{T_i} = AID_{T_i(new)}$ and save the information for next authentication process. Else
 307 freshness check get unsuccessful then Tag will not update AID_{T_i} as $AID_{T_i(new)}$.

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

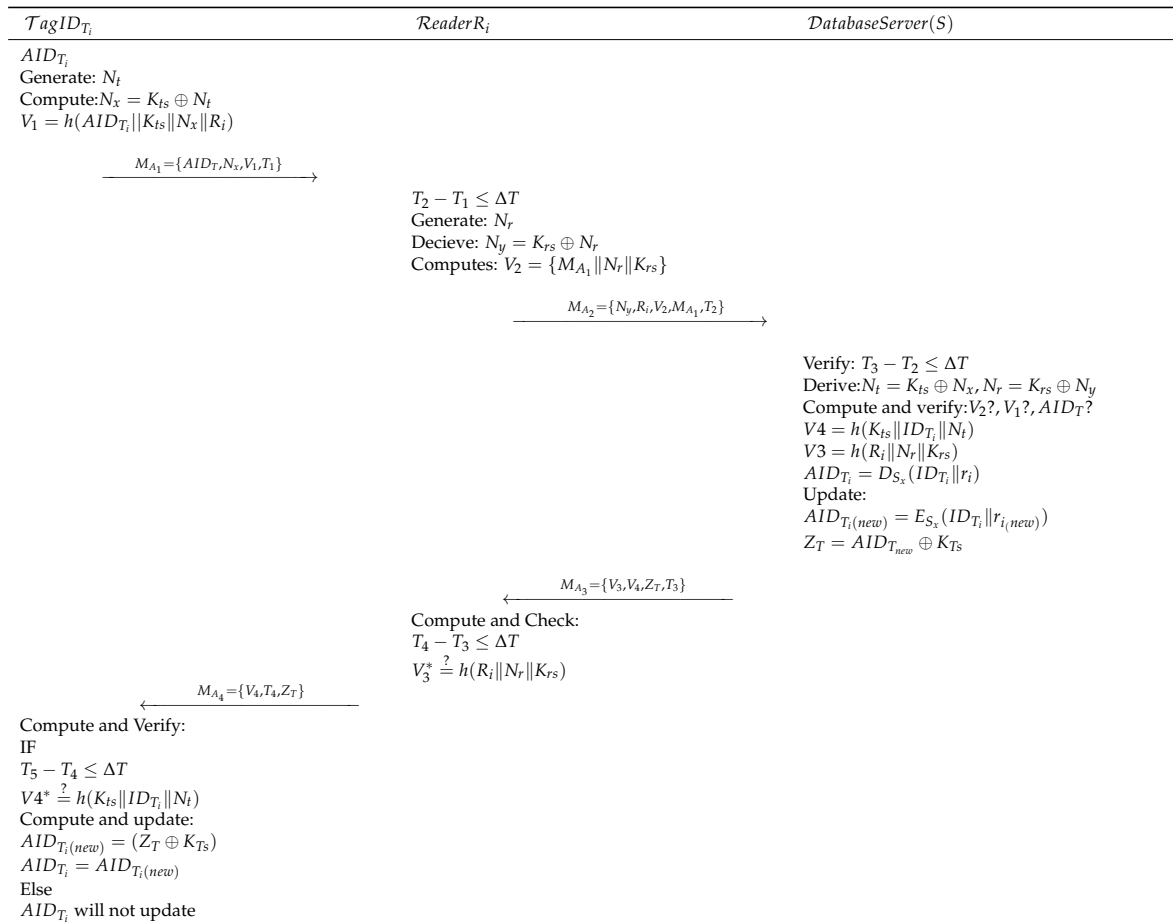


Figure 5. Proposed authentication protocol

309 4. Security Analysis

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

319 4.1. Security Analysis with BAN-logic

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

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)
$\langle X \rangle_Y$	X is combine with Y
$(X)_K$	Hash of message X using a key K
$P \xleftrightarrow{K} Q$	P and Q share key K for communication
$AIDT_i$	$AIDT_i$ is one time session key
$\frac{P \equiv P \xleftrightarrow{K} Q, P \triangleleft \langle X \rangle_K}{P \equiv Q \sim X}$	Message-Meaning rule
$\frac{P \equiv \#(X)}{P \equiv \#(X, Y)}$	Freshness-conjunction 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

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

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

- | | | | |
|-----|--|-----|--|
| 328 | • Goal 1: $R_i \equiv Tag \xleftrightarrow{AIDT_i} Ri$ | 332 | • Goal 5: $R_i \equiv S_j \xleftrightarrow{AIDT_i} Ri$ |
| 329 | • Goal 2: $R_i \equiv Tag \equiv Tag \xleftrightarrow{AIDT_i} Ri$ | 333 | • Goal 6: $R_i \equiv S_j \equiv S_j \xleftrightarrow{AIDT_i} Ri$ |
| 330 | • Goal 3: $S_j \equiv R_i \xleftrightarrow{AIDT_i} S_j$ | 334 | • Goal 7: $Tag \equiv R_i \xleftrightarrow{AIDT_i} Tag$ |
| 331 | • Goal 4: $S_j \equiv R_i \equiv R_i \xleftrightarrow{AIDT_i} S_j$ | 335 | • Goal 8: $Tag \equiv R_i \equiv R_i \xleftrightarrow{AIDT_i} Tag$ |

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

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

- 340 • M1: $Tag \rightarrow Ri: AIDT_i, N_x : \langle N_t \rangle_{K_{ts}}, V1, T1$
 341 • M2: $R_i \rightarrow Sj: M1, N_y : \langle N_r \rangle_{K_{rs}}, R_i, V2, T2,$
 342 • M3: $S_j \rightarrow Ri: V3, V4, Z_t : \langle AIDT_i \rangle_{K_{ts}}, T3$
 343 • M4: $R_i \rightarrow Tag : V4, T4, Z_t : \langle AIDT_i \rangle_{K_{ts}}$

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

- | | | | |
|-----|---|-----|---|
| 346 | • A1: $Tag \equiv \#(N_t)$ | 351 | • A6: $S_j \equiv R_i \Rightarrow N_r$ |
| 347 | • A2: $R_i \equiv \#(N_r)$ | 352 | • A7: $S_j \equiv Tag \Rightarrow N_t$ |
| 348 | • A3: $S_j \equiv \#(AIDT_i)(r_i)$ | 353 | • A8: $Tag \equiv S_j \Rightarrow r_i$ |
| 349 | • A4: $R_i \equiv S_j \Rightarrow r_i$ | 354 | • A9: $Tag \equiv R_i \Rightarrow N_r$ |
| 350 | • A5: $R_i \equiv Tag \Rightarrow N_t$ | | |

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

357 M1: $Tag \rightarrow Ri: AIDT_i, N_x :< N_t >_{K_{ts}}, T1$ is time-stamp of Tag

358 Using seeing rule, the following can be achieved

359 • S1: $R_i \triangleleft AIDT_i, SID, N_x :< N_t >_{K_{ts}}, T1$

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

361 • S2: $R_i | \equiv Tag | \sim N_t$

362 Using Freshness-conjunction rule and S2 to achieve the following

363 • S3: $R_i | \equiv Tag | \equiv N_t$

364 Using jurisdiction rule and S3, the following can be achieved

365 • S4: $R_i | \equiv N_t$

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

367 • S5: $R_i | \equiv Tag \xleftrightarrow{AIDT_i} Ri$ (**Goal 1**)

368 Using nonce-verification rule, the following is obtained

369 • S6: $R_i | \equiv Tag | \equiv Tag \xleftrightarrow{AIDT_i} Ri$ (**Goal 2**)

370 M2: $R_i \rightarrow S_j : M1, N_y :< N_r >_{K_{rs}}, T2, V2$. Where, $T2$ is time-stamp of R_i

371 By using the seeing rule, we achieve

372 • S7: $S_j \triangleleft M1, N_y :< N_r >_{K_{rs}}, T2, V2$

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

374 • S8: $S_j | \equiv R_i | \sim N_r$

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

376 • S9: $S_j | \equiv R_i | \equiv N_r$

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

378 • S10: $S_j | \equiv N_r$

379 Using S10 and the SK rule, the following can be achieved

380 • S11: $S_j | \equiv R_i \xleftrightarrow{AIDT_i} S_j$ (**Goal 3**)

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

382 • S12: $S_j | \equiv R_i | \equiv R_i \xleftrightarrow{AIDT_i} S_j$. (**Goal 4**)

383 M3: $S_j \rightarrow Ri: V3, V4, Zt < AIDT_{i_{new}} >_{K_{ts}}^*, T3, T3$ is time-stamp of S_j

384 By seeing-rule, the following can be achieved

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

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

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

388 By S14 and the Freshness-conjunction rule, the following can be achieved

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

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

$$391 \quad \bullet \text{ S16: } R_i | \equiv AIDT_{i_{new}}$$

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

$$393 \quad \bullet \text{ S17: } R_i | \equiv S_j \xleftrightarrow{AIDT_{i_{new}}} R_i. \text{ (Goal 5)}$$

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

$$395 \quad \bullet \text{ S18: } R_i | \equiv S_j | \equiv S_j \xleftrightarrow{AIDT_{i_{new}}} R_i. \text{ (Goal 6)}$$

$$396 \quad \text{M4: } R_i \rightarrow Tag : V4, Zt < AIDT_{i_{new}} >_{K_{ts}}, T4, T4 \text{ is timestamp of } R_i$$

397 Using seeing rule, the following can be computed

$$398 \quad \bullet \text{ S19: } Tag \triangleleft V4, Zt < AIDT_{i_{new}} \geq_{ts}, T4$$

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

$$400 \quad \bullet \text{ S20: } Tag | \equiv R_i | \sim AIDT'_{i_{new}}$$

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

$$402 \quad \bullet \text{ S21: } Tag | \equiv R_i | \equiv AIDT_{i_{new}}$$

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

$$404 \quad \bullet \text{ S22: } Tag | \equiv AIDT_{i_{new}}$$

405 Using session-key rule, the following can be obtained

$$406 \quad \bullet \text{ S23: } Tag | \equiv R_i \xleftrightarrow{AIDT_{i_{new}}} Tag \text{ (Goal 7)}$$

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

$$409 \quad \bullet \text{ S24: } Tag | \equiv R_i | \equiv R_i \xleftrightarrow{AIDT_{i_{new}}} Tag \text{ (Goal 8)}$$

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

412 4.2. Security Analysis with ProVerif

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

419 In the proposed authentication protocol two main channels have been considered for
420 communication, i.e., ChSec: Private channel, and ChPub: public channel. ChSec is a secure channel
421 used for registration of RFID Tag IDT_i with RFID Server S_i through RFID Reader device R_i . ChPub is
422 a public and insecure channel between RFID Tag IDT_i , RFID-Reader device R_i and RFID Server S_i .
423 ChPub is a public and insecure channel between RFID Tag IDT_i , RFID-Reader device R_i and RFID
424 Server S_i . ChPub used for authenticating RFID Tag IDT_i with the RFID Server S_i . All the participant
425 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
426 are different nonce generated by each participant. K_{ts} and K_{rs} shared keys among the participants.
427 Different function including XOR, Hash, Concatenation encryption and decryption function declared
428 as constructors. The results obtained from ProVerif prove the correctness of the proposed protocol and
429 the code is given below.

```

430 (* ----- Channels -----*)
431 free ChSec:channel [private]. (*secure channel between Tag and S*)
432 free ChPub:channel. (*public channel between Tag,R and S*)
433 (*----- Constants and Variables -----*)
434 free IDTi :bitstring.
435 free IDs :bitstring.
436 free Ri:bitstring.
437 free Kts : bitstring [private].
438 (*=====Constructors=====*)
439 fun h(bitstring):bitstring.
440 fun Inverse(bitstring):bitstring.
441 fun Concat(bitstring,bitstring):bitstring.
442 fun XOR(bitstring,bitstring):bitstring.
443 fun enc(bitstring,bitstring): bitstring.
444 fun dec(bitstring,bitstring): bitstring.
445 (*=====Equations=====*)
446 equation forall a:bitstring; Inverse(Inverse(a))=a.
447 equation forall a:bitstring, b:bitstring; XOR(XOR(a,b),b)=a.
448 equation forall x: bitstring, y: bitstring; dec(enc(x,y),y) = x.
449 equation forall x: bitstring, y: bitstring; enc(dec(x,y),y) = x.

```

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

```

452 (*-----RFID TAG-----*)
453 (*====*registration*====*)
454 let pTag=
455 (* Registration *)
456 out(ChSec,(IDTi));
457 in (ChSec,(IDTi:bitstring,Kts:bitstring,AIDTi:bitstring));
458 (*-----TAG login-----*)
459 event start_Tag(IDTi);
460 new Nt:bitstring;
461 new T1:bitstring;
462 let Nx=XOR(Kts,Nt) in
463 let V1=h(Concat(AIDTi, (Kts,Nx, Ri)))in
464 out(ChPub,(AIDTi,Nx,V1,T1));
465 in (ChPub,(V4:bitstring,T4:bitstring,Zt:bitstring));
466 let V4=h(Concat(Kts,(IDTi, Nt))) in
467 if V4=h(Concat(Kts,(IDTi, Nt))) then
468 let xAIDTinew = XOR(Zt,Kts) in
469 let AIDTi = AIDTinew in
470 event end_Tag(IDTi)
471 else
472 0.

473 (*-----RFID-Reader-----*)
474 let pR=
475 event start_R(Ri);
476 new Nr:bitstring;
477 new T2:bitstring;
478 new Krs:bitstring;
479 new Dj:bitstring;
480 new Ts:bitstring;
481 new MA1:bitstring;
482 in (ChPub,(xAIDTi:bitstring,Nx:bitstring,V1:bitstring,T1:bitstring));

```

```

483 let Ny=XOR(Krs,Nr) in
484 let V2=Concat(MA1, (Nr,Krs)) in
485 out(ChPub, (Ny,Ri,V2,MA1,T2));
486 in (ChPub, (V3:bitstring,V4:bitstring,Zt:bitstring,T3:bitstring));
487 new ZT:bitstring;
488 new T4:bitstring;
489 if V3=h((Concat(Ri, (Nr,Krs)))) then
490 out(ChPub, (V4,T4,ZT));
491 event end_R(Ri)
492 else
493 0.

(*-----RFID Server-----*)
494 let pS=
495 (*-----Registration phase-----*)
496 in (ChSec, (IDTi:bitstring));
497 new ns:bitstring;
498 new rti:bitstring;
499 new S:bitstring;
500 let Kts=XOR(h(Concat(IDTi, ns)),IDs) in
501 (*let AIDTi=enc(Concat(IDTi,rti),S) in*)
502 out(ChSec, (IDTi,Kts));
503 (*-----login-authentication-----*)
504 event start_S(IDs);
505 in (ChPub, (Ny:bitstring,Ri:bitstring,V2:bitstring,MA1:bitstring,T2:bitstring));
506 new Nx: bitstring;
507 new Krs: bitstring;
508 let Nt=Concat(Kts, Nx) in
509 let Nr=Concat(Krs, Ny) in
510 new T3: bitstring;
511 let V4= h(Concat(Kts, ( IDTi,Nt))) in
512 let V3=h(Concat(Ri, ( Nr, Krs))) in
513 new ri: bitstring;
514 let AIDTi=dec(Sx, (Concat(IDTi, ri))) in
515 new rnew: bitstring;
516 let AIDTinew= enc(Concat(IDTi,rnew),Sx) in
517 let ZT =XOR(AIDTinew, Kts) in
518 out(ChPub, (V3,V4,ZT,T3));
519 event end_S(IDs)
520 else
521 0.
522

```

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

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

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

```

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

```

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

```

535 (*====*Events*====*)
536 event start_Tag(bitstring).
537 event end_Tag(bitstring).
538 event start_R(bitstring).
539 event end_R(bitstring).
540 event start_S(bitstring).
541 event end_S(bitstring).
542 After the compilation of the proposed protocol, ProVerif code the following result has been obtained:
543 1-- Query inj-event(end_S(IDs[])) ==> inj-event(start_S(IDs[]))
544 Completing...
545 Starting query inj-event(end_S(IDs[])) ==> inj-event(start_S(IDs[]))
546 RESULT inj-event(end_S(IDs[])) ==> inj-event(start_S(IDs[])) is true.
547 2-- Query inj-event(end_R(Ri[])) ==> inj-event(start_R(Ri[]))
548 Completing...
549 Starting query inj-event(end_R(Ri[])) ==> inj-event(start_R(Ri[]))
550 RESULT inj-event(end_R(Ri[])) ==> inj-event(start_R(Ri[])) is true.
551 3-- Query inj-event(end_Tag(IDTi[])) ==> inj-event(start_Tag(IDTi[]))
552 Completing...
553 Starting query inj-event(end_Tag(IDTi[])) ==> inj-event(start_Tag(IDTi[]))
554 RESULT inj-event(end_Tag(IDTi[])) ==> inj-event(start_Tag(IDTi[])) is true.
555 4-- Query not attacker(AIDTnew[])
556 Completing...
557 Starting query not attacker(AIDTnew[])
558 RESULT not attacker(AIDTnew[]) is true.

```

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

563 4.3. Informal Security Analysis

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

- | | | | |
|-----|--|-----|---|
| 566 | 1. Mutual Authentication between Tag and | 573 | 6. Collision attack |
| 567 | Server | 574 | 7. Denial of Service (DoS) attack |
| 568 | 2. Anonymity | 575 | 8. Replay attacks |
| 569 | 3. Untraceability | 576 | 9. Location tracking attack |
| 570 | 4. Backward/Forward secrecy | 577 | 10. Impersonation attack (Forgery attack) |
| 571 | 5. Scalability | | 11. Stolen-verifier attack |

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

580 4.3.1. Mutual Authentication Between Tag And Server

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

586 4.3.2. Anonymity

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

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

599 4.3.3. Traceability

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

608 4.3.4. Backward/Forward Secrecy

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

617 4.3.5. Scalability

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

621 4.3.6. Collision Attack

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

626 4.3.7. Denial Of Service (DoS) Attack

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

630 4.3.8. Replay Attacks

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

641 4.3.9. Location Tracking Attack

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

647 4.3.10. Impersonation Attacks(Forgery Attacks)

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

657 4.3.11. Stolen-Verifier Attacks

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

663 5. Comparative Analysis

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

671 5.1. Security Requirements

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

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

- | | | | |
|-----|----------------------------------|-----|----------------------------------|
| 676 | • SR1: Mutual authentication. | 682 | • SR7: DoS attacks. |
| 677 | • SR2: Tag untraceability. | 683 | • SR8: Replay attacks. |
| 678 | • SR3: Tag anonymity. | 684 | • SR9: Location tracking attack. |
| 679 | • SR4: Backward/Forward secrecy. | 685 | • SR10: Forgery attack. |
| 680 | • SR5: Scalability. | 686 | • SR11: Stolen-verifier attacks. |
| 681 | • SR6: Collision attacks. | | |

687 The proposed protocol compared with existing protocols from the above-listed requirements. The
688 existing protocols considered for comparison proposed in various articles including [19,21,23,30,
689 31]. Note that SR is a security requirement, so SR1 is the security requirement number 1 in the list and
690 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

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

693 5.2. Computation Cost Analysis

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

- 701 • CC: Computation cost
- 702 • T_h : CC of single hash function;
- 703 • T_{me} : CC of modular exponentiation;
- 704 • T_{sc} : CC of symmetric encryption;
- 705 • T_{sd} : CC of symmetric decryption;
- 706 • CC_{mbu} : Computation-cost of Tag
- 707 • CC_{HA} : Computation-cost of $Reader R_i$
- 708 • CC_{FA} : Computation-cost of $Server S$
- 709 • CC_{total} : Total computation-cost.

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

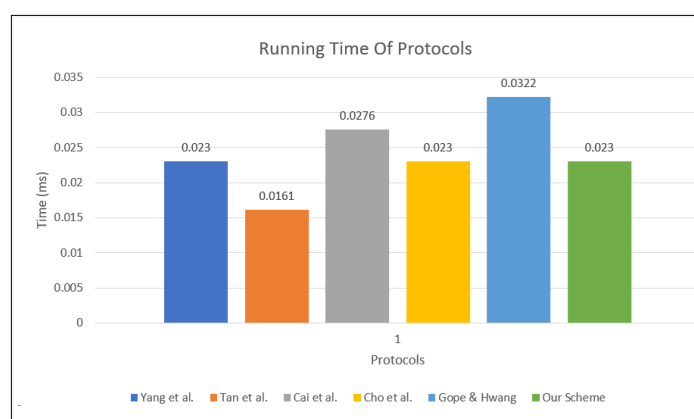
Table 5. Comparison of computation cost and running time

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

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

718 The calculated running time is recorded using an Intel Pentium dual-core PC with processor 2.20
 719 GHz (E2200) possessing 2048 MB of RAM. The operating system is a 32-bit Ubuntu version 12.04.1
 720 is utilized. Running time of the protocols is in milliseconds (ms). The total time for all protocols in
 721 Table 5 have also been computed in millisecond (ms) following the Kilinc and Yanik [38] experiments.
 722 According to [38], a single T_h cost $0.0023ms$, T_{comp} cost $2.226ms$, T_{cpa} cost $0.0288ms$, T_{me} cost $3.8500ms$.
 723 Therefore, based on [38], the total computational time of the proposed protocol is $0.023ms$, whereas
 724 the total cost of the protocol presented in [23] is $0.0230ms$, the cost of protocol in [30] is proximately
 725 $0.0161ms$, the cost of the proposal in [19] is $0.0322ms$, and the proposal in [21] takes a total of $0.0276ms$.
 726 It has been observed that the proposed protocol is very efficient and outperforms some of the existing
 727 protocols in terms of computation cost. In some cases, however, it achieves either less efficiency or
 728 equal efficiency to some existing protocols. In such case, it must also be noted that the existing protocol
 729 though having equal or more effective than the proposed protocol, but at the same time those protocols
 730 are vulnerable to various security threats/attacks whereas the proposed protocol is secured against all
 731 known attacks. This comparative analysis has been depicted in Figure 6.

732 It can be clearly observed from Figure 6 that the proposed protocol out performs the baseline
 733 protocol in terms of computation cost. The proposed protocol not only overcomes the security
 734 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

736 5.3. Communication and Storage Cost Analysis

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

749 The storage cost is represented by length Value L . The proposed protocol uses SHA-1 hash
 750 function to implement $h(\cdot)$, then each of the length value is 160-bit long. In the proposed scheme each
 751 of the Tag stored ID_{T_i}, K_{ts}, AID parameters. Therefore, the cost of storage in the Tag is $3L$. Whereas on
 752 server side $ID_{T_i}, K_{ts}, AID_{new}, AID_{old}$ are being stored, hence the storage cost on server side is $4L$ per
 753 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

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

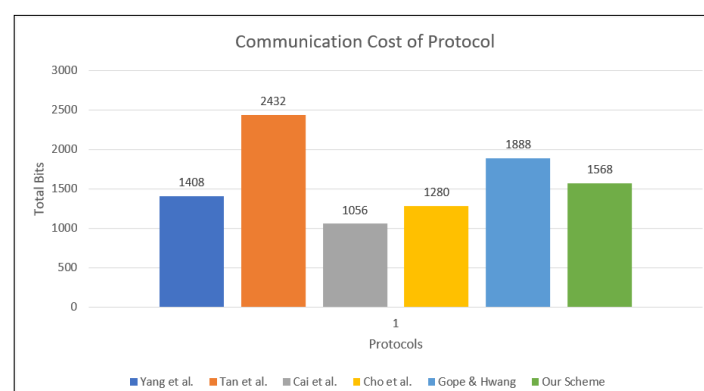


Figure 7. Communication Cost

760 Figure 7 indicates that the proposed protocol is efficient than the baseline protocol in terms of
 761 communication cost. Specifically, the proposed protocol not only overcomes the security flaws of the

762 baseline protocol but also achieves 16.94% efficiency in terms of the number of bits exchanged during
763 one transaction of the protocol.

764 6. Conclusion

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

778 References

- 779 1. Gope, P.; Hwang, T. BSN-Care: a secure IoT-based modern healthcare system using body sensor network.
780 *IEEE Sensors Journal* **2016**, *16*, 1368–1376.
- 781 2. Gope, P.; Amin, R.; Islam, S.H.; Kumar, N.; Bhalla, V.K. Lightweight and privacy-preserving RFID
782 authentication scheme for distributed IoT infrastructure with secure localization services for smart city
783 environment. *Future Generation Computer Systems* **2018**, *83*, 629–637.
- 784 3. Peris-Lopez, P.; Hernandez-Castro, J.C.; Estevez-Tapiador, J.M.; Ribagorda, A. Lightweight cryptography
785 for low-cost RFID tags. *Security in RFID and Sensor Networks* **2016**, pp. 121–150.
- 786 4. Gope, P.; Amin, R.; Islam, S.H.; Kumar, N.; Bhalla, V.K. Lightweight and privacy-preserving RFID
787 authentication scheme for distributed IoT infrastructure with secure localization services for smart city
788 environment. *Future Generation Computer Systems* **2018**, *83*, 629–637.
- 789 5. Kitsos, P. *Security in RFID and sensor networks*; CRC Press, 2016.
- 790 6. Hsu, C.H.; Wang, S.; Zhang, D.; Chu, H.C.; Lu, N. Efficient identity authentication and encryption
791 technique for high throughput RFID system. *Security and Communication Networks* **2016**, *9*, 2581–2591.
- 792 7. Simon, P.M.G.; Riggert, E.F.; Trivelpiece, S.E. System and method for reading RFID tags across a portal,
793 2016. US Patent 9,519,811.
- 794 8. Wu, F.; Xu, L.; Kumari, S.; Li, X.; Das, A.K.; Shen, J. A lightweight and anonymous RFID tag authentication
795 protocol with cloud assistance for e-healthcare applications. *Journal of Ambient Intelligence and Humanized*
796 *Computing* **2018**, *9*, 919–930.
- 797 9. Sidorov, M.; Ong, M.T.; Vikneswaran, R.; Nakamura, J.; Ohmura, R.; Khor, J.H. Ultralightweight Mutual
798 Authentication RFID Protocol for Blockchain Enabled Supply Chains. *IEEE Access* **2019**.
- 799 10. Noman, A.T.; Hossain, S.; Islam, S.; Islam, M.E.; Ahmed, N.; Chowdhury, M.M. Design and Implementation
800 of Microcontroller Based Anti-Theft Vehicle Security System using GPS, GSM and RFID. 2018 4th
801 International Conference on Electrical Engineering and Information & Communication Technology
802 (iCEEICT). IEEE, 2018, pp. 97–101.
- 803 11. Liao, Y.P.; Hsiao, C.M. A secure ECC-based RFID authentication scheme integrated with ID-verifier transfer
804 protocol. *Ad Hoc Networks* **2014**, *18*, 133–146.
- 805 12. Kim, H. RFID mutual authentication protocol based on synchronized secret. *International Journal of Security*
806 *and Its Applications* **2013**, *7*, 37–50.
- 807 13. Cha, J.R.; Kim, J.H. Novel anti-collision algorithms for fast object identification (RFID) system. Parallel
808 and Distributed Systems, Proceedings. 11th International Conference. IEEE, 2005, Vol. 2, pp. 63–67.
- 809 14. El Beqqal, M.; Azizi, M. Classification of major security attacks against RFID systems. International
810 Conference on Wireless Technologies, Embedded and Intelligent Systems (WITS). IEEE, 2017, pp. 1–6.

- 811 15. Tewari, A.; Gupta, B. Cryptanalysis of a novel ultra-lightweight mutual authentication protocol for IoT
812 devices using RFID tags. *The Journal of Supercomputing* **2017**, *73*, 1085–1102.
- 813 16. Ayaz, U.; Haq, T.A.; Taimour, S.; Mansoor, K.; Mahmood, S. An Enhanced Biometric Based RFID
814 Authentication Scheme Defending Against Illegitimate Access. 14th International Conference on Emerging
815 Technologies (ICET). IEEE, 2018, pp. 1–6.
- 816 17. Kolhe, P.; Dharaskar, R.; Tharkar, M.; Joshi, S.; Desai, S.; Dapoli, B. Information technology tool in library
817 barcode & Radio Frequency Identification RFID. *Int. J. Innov. Sci. Eng. Technol* **2016**, *3*, 81–6.
- 818 18. Sidorov, M.; Ong, M.T.; Sridharan, R.V.; Nakamura, J.; Ohmura, R.; Khor, J.H. Ultralightweight Mutual
819 Authentication RFID Protocol for Blockchain Enabled Supply Chains. *IEEE Access* **2019**, *7*, 7273–7285.
820 doi:10.1109/ACCESS.2018.2890389.
- 821 19. Gope, P.; Hwang, T. A realistic lightweight authentication protocol preserving strong anonymity for
822 securing RFID system. *Computers & Security* **2015**, *55*, 271–280.
- 823 20. Burmester, M.; De Medeiros, B.; Motta, R. Robust, anonymous RFID authentication with constant
824 key-lookup. Proceedings of the 2008 ACM symposium on Information, computer and communications
825 security. ACM, 2008, pp. 283–291.
- 826 21. Cai, S.; Li, Y.; Li, T.; Deng, R.H. Attacks and improvements to an RFID mutual authentication protocol and
827 its extensions. Proceedings of the second ACM conference on Wireless network security. ACM, 2009, pp.
828 51–58.
- 829 22. Gaubatz, G.; Kaps, J.P.; Ozturk, E.; Sunar, B. State of the art in ultra-low power public key cryptography for
830 wireless sensor networks. Pervasive Computing and Communications Workshops, PerCom Workshops.
831 Third IEEE International Conference on. IEEE, 2005, pp. 146–150.
- 832 23. Yang, J.; Park, J.; Lee, H.; Ren, K.; Kim, K. Mutual authentication protocol. Workshop on RFID and
833 lightweight crypto, 2005.
- 834 24. Kang, S.Y.; Lee, I.Y. A Study on low-cost RFID system management with mutual authentication scheme in
835 ubiquitous. Asia-Pacific Network Operations and Management Symposium. Springer, 2007, pp. 492–502.
- 836 25. Lee, L.S.; Fiedler, K.D.; Smith, J.S. Radio frequency identification (RFID) implementation in the service
837 sector: A customer-facing diffusion model. *International Journal of Production Economics* **2008**, *112*, 587–600.
- 838 26. Qingling, C.; Yiju, Z.; Yonghua, W. A minimalist mutual authentication protocol for RFID system & BAN
839 logic analysis. International Colloquium on Computing, Communication, Control, and Management,
840 CCCM. IEEE, 2008, Vol. 2, pp. 449–453.
- 841 27. Zhou, S.; Zhang, Z.; Luo, Z.; Wong, E.C. A lightweight anti-desynchronization RFID authentication
842 protocol. *Information Systems Frontiers* **2010**, *12*, 521–528.
- 843 28. Piramuthu, S. RFID mutual authentication protocols. *Decision Support Systems* **2011**, *50*, 387–393.
- 844 29. Safkhani, M.; Peris-Lopez, P.; Hernandez-Castro, J.C.; Bagheri, N. Cryptanalysis of the Cho et al. protocol:
845 A hash-based RFID tag mutual authentication protocol. *Journal of Computational and Applied Mathematics*
846 **2014**, *259*, 571–577.
- 847 30. Tan, C.C.; Sheng, B.; Li, Q. Secure and serverless RFID authentication and search protocols. *IEEE*
848 *Transactions on Wireless Communications* **2008**, *7*, 1400–1407.
- 849 31. Cho, J.S.; Jeong, Y.S.; Park, S.O. Consideration on the brute-force attack cost and retrieval cost: A hash-based
850 radio-frequency identification (RFID) tag mutual authentication protocol. *Computers & Mathematics with*
851 *Applications* **2015**, *69*, 58–65.
- 852 32. Gong, L.; Needham, R.; Yahalom, R. Reasoning about belief in cryptographic protocols. Proceedings of
853 IEEE Computer Society Symposium on Research in Security and Privacy. IEEE, 1990, pp. 234–248.
- 854 33. Chaudhry, S.A.; Naqvi, H.; Farash, M.S.; Shon, T.; Sher, M. An improved and robust biometrics-based three
855 factor authentication scheme for multiserver environments. *The Journal of Supercomputing* **2015**, pp. 1–17.
- 856 34. Kyntaja, T. A logic of authentication by Burrows, Abadi and Needham. *Science Helsinki University of*
857 *Technology, Tehran*. <http://www.tml.tkk.fi/Opinnot/Tik-110.501/1995/ban.html> **1995**.
- 858 35. Blanchet, B.; others. Modeling and verifying security protocols with the applied pi calculus and ProVerif.
859 *Foundations and Trends® in Privacy and Security* **2016**, *1*, 1–135.
- 860 36. Lumini, A.; Nanni, L. An improved biohashing for human authentication. *Pattern recognition* **2007**,
861 *40*, 1057–1065.
- 862 37. Zhang, G.; Fan, D.; Zhang, Y.; Li, X.; Liu, X. A privacy preserving authentication scheme for roaming
863 services in global mobility networks. *Security and Communication Networks* **2015**, *8*, 2850–2859.

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