Research Article

Cloud-Based RFID Mutual Authentication Protocol without Leaking Location Privacy to the Cloud

Qingkuan Dong, Jiaqing Tong, and Yuan Chen

State Key Laboratory of Integrated Services Networks, Xi\'an Jiaotong University, Xi\’an, Shaanxi 710071, China

Correspondence should be addressed to Jiaqing Tong; tjql9901228@sina.cn

Received 30 April 2015; Revised 4 July 2015; Accepted 26 July 2015

Academic Editor: Zongming Fei

Copyright © 2015 Qingkuan Dong et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

With the rapid developments of the IoT (Internet of Things) and the cloud computing, cloud-based RFID systems attract more attention. Users can reduce their cost of deploying and maintaining the RFID system by purchasing cloud services. However, the security threats of cloud-based RFID systems are more serious than those of traditional RFID systems. In cloud-based RFID systems, the connection between the reader and the cloud database is not secure and cloud service provider is not trusted. Therefore, the users have to encrypt their data stored in the cloud database to prevent the leakage of privacy. In addition, the reader's location privacy should be protected to avoid its leak to the cloud provider. In this paper, a cloud-based RFID mutual authentication protocol without leaking location privacy to the cloud is proposed. It provides real-time mutual authentication between the reader and the tag and protects the reader's location privacy by introducing the location privacy cloud. Compared with traditional backend-server-based schemes and serverless schemes, the proposed scheme has obvious advantages in deployment cost, scalability, real-time authentication, and the tag's computational complexity.

1. Introduction

RFID (radio frequency identification) is a key technology of the IoT for identifying the objects in a noncontact way. It is widely used in the fields of manufacture, retail, medical treatment, transportation, tracking, and location because the RFID tag is low in price, small in size, and easy to take. Besides, massive tags can be read simultaneously compared with bar codes. However, once the object is labeled a tag, the data privacy and the owner's location privacy would be threatened. So the owner's location privacy and security protection are the prerequisites for popularizing the RFID technology.

The traditional RFID system is composed of tags, readers, and a backend database, as seen in Figure 1. The reader activates the tag by sending the RF signals to communicate and exchange information with it in a noncontact way and submits the relevant data to the backend database. There are a lot of authentication schemes under this architecture [1–3]. These authentication schemes always assume that there is a secure backend server and the link between the reader and the backend server is reliable. For instance, Wei et al. [1] proposed a mutual authentication protocol based on hash function and Dong et al. [2] proposed a mutual authentication protocol based on SHA-3. In their schemes, the backend server needs to search the matching records by computing hash function; the computing ability of the backend server will be the bottleneck of the system. He et al. [3] proposed an ECC-based authentication scheme in which the tag needs to compute scalar multiplication over the elliptic curve, so it does not satisfy the requirements of the lightweight tag. What is worse is that the backend-server-based architecture limits the mobility of the reader and the cost of deploying and maintaining the backend server is high.

The serverless architecture consists of three kinds of entities: readers, tags, and a Certificate Authority (CA). Readers authenticate tags via the help of online CA, as seen in Figure 2. Each tag registers in the CA and each authorized reader downloads the Access List (AL) from the CA through a secure channel during the initialization process. For example, Lee et al. [4] proposed a serverless RFID authentication and search protocol. Hoque et al. [5] proposed enhancing privacy
(iii) The tag calculates $H(R || T || Nr)$ as a response and a random nonce $Nr$ as its challenge to the reader.

(iv) The reader verifies $H(R || T || Nr)$, sends queries to the cloud, and checks answers, until finding the last valid record, assuming its SID is $M$. Then the reader computes $H(R || T || M')$ and $E(R || T' || M')$ to notify the cloud to begin the update, where $M' = M + 1$.

In this paper, a lightweight cloud-based RFID mutual authentication protocol without leaking location privacy to the cloud is proposed. A global encrypted hash table (EHT) corresponding to the encrypted RFID tags' information is stored in the cloud database. It provides real-time mutual authentication between the reader and the tag and protects the reader's location privacy by introducing the location privacy cloud which is not able to read the RFID data. Compared with traditional backend-server-based schemes and serverless schemes, the proposed scheme has obvious advantages in deployment cost, scalability, real-time authentication, and the tag's computational complexity.

The rest of the paper is organized as follows. Section 2 reviews the cloud-based RFID authentication schemes. The proposed authentication framework is given in Section 3. In Section 4, the proposed cloud-based RFID mutual authentication protocol is given. In Section 5, the security and the performance of the proposed protocol are analyzed. Section 6 concludes the paper.

2. Related Work

In this section, we will briefly review the cloud-based RFID authentication schemes and point out some disadvantages of them. Usually, the cloud-based RFID system is composed of a cloud server (cloud database), readers, and tags, where the reader is able to move and the RFID data is stored in the cloud database. There are several cloud-based RFID systems.

Xie et al. [12] proposed a cloud-based RFID authentication protocol. They used a Virtual Private Network (VPN) agency to guarantee the reliable connection between the reader and the cloud database. However, the cost of deploying and maintaining the VPN agency is high and this architecture is not suitable for SMEs (Small and Medium Enterprises). If the VPN agency is not maintained by the user, it is likely to expose the business information of the enterprise to the VPN agency. The protocol works as follows:

(i) The tag sends $H(R || T || S)$ and a request to the reader.

(ii) The reader reads the cipher text $E(R || T || S)$ indexed by $H(R || T || S)$ from the cloud, decrypts it, and gets $T$ and $S$. Then the reader generates a random number $Nr$ as a challenge to the tag.

(iii) The tag calculates $H(R || T || Nr)$ as a response and a random nonce $Nr$ as its challenge to the reader.

(iv) The reader verifies $H(R || T || Nr)$, sends queries to the cloud, and checks answers, until finding the last valid record, assuming its SID is $M$. Then the reader computes $H(R || T || M')$ and $E(R || T' || M')$ to notify the cloud to begin the update, where $M' = M + 1$. 
(v) The cloud adds the new record into the EHT, computes \( H(R \parallel T \parallel M') \oplus H(E(R \parallel T \parallel M')) \), and sends it to the reader.

(vi) The reader confirms the updating is successful and computes \( H(R \parallel T \parallel Nt) \oplus M' \) and \( H(T \parallel R \parallel M') \) to notify the tag to begin the update.

(vii) The tag calculates \( H(R \parallel T \parallel Nt) \) XOR with the received \( H(R \parallel T \parallel Nt) \oplus M' \) to obtain \( M' \), and then it calculates and verifies \( H(T \parallel R \parallel M') \).

The weakness of Wei’s cloud-based RFID authentication protocol was pointed out by Abughazalah et al. [13]. The authors find that Wei’s authentication protocol suffers from reader impersonate attack and tag’s location tracking attack. And the authors proposed a secure improved cloud-based RFID authentication protocol. They assume that the communication channel between the reader and the cloud server is secure, but this assumption is not suitable for the mobile readers.

Lin et al. [14] proposed a cloud-based authentication protocol for RFID supply chain systems. In their scheme, the system is composed of tags, readers, a cloud server, and a trust party. The authentication protocol cannot resist old key compromise attack. Its key update used a simple XOR operation, \( Key_{\text{new}} = Key \oplus h(r_T) \), where \( h() \) is a hash function and \( r_T \) is a random number. When the adversaries obtained the old key, they can impersonate and track the key update. Therefore, this scheme does not achieve the forward security.

Furthermore, all the above schemes do not protect the reader’s location privacy; the leakage of the reader’s location privacy will expose the business information of the company.

3. Our Authentication Framework

In this section, we describe our authentication framework. It provides the location privacy protection for the reader to access the cloud database.

3.1. System Components. The framework is illustrated in Figure 3.

**Tag.** The tag is used to label the object. Its identity, \( id_T \), and secret key, \( Key \), are stored in the tag, and it is able to generate random numbers and conduct the bitwise XOR and hash operation.

**Reader.** The reader can move freely. In the reader, its identity \( id_R \) and the key \( kRC \) shared by the reader and the cloud server are stored. The reader is able to generate random numbers and conduct the bitwise XOR, hash operation, and symmetric encryption and decryption.

**Location Privacy Cloud.** Location privacy cloud is an infrastructure of IoT. It is maintained by a trusted third party or an organization that provides privacy service like the public platform of IoT. It is composed of several public access points; one of them is called edge access point which connects to the cloud database directly, and the others are called general access points. The edge access point needs to support random numbers generation and symmetric encryption and decryption.

**Cloud Server.** The cloud server needs to provide service on demand. It stores the new EHT pairs and the old EHT pairs in the cloud database. The cloud server needs to support query and update of the RFID tag records and have the ability to calculate message authentication code (MAC).

3.2. The Location Privacy Mechanism of the Reader to the Cloud Database. In the following, we suppose the user \( U_A \) is a small company and \( U_A \) manages the sales of goods using RFID technology. In order to save maintenance costs, \( U_A \) purchases cloud database service from the cloud service provider and applies for accessing the cloud database via the location privacy cloud in which there are some general access points that provide public access services for RFID readers.
Generally, a company has one or more mobile readers to identify their tags with the help of the cloud database. The mobile reader belongs to the user and its IP address is denoted by IP\(_2\). When the user purchases cloud database service, the cloud server shares a key kRC with the reader. And the reader needs to store the IP address IP\(_{\text{CloudA}}\) of the cloud server. The cloud database which provides cloud service for the user U\(_A\) is called CloudA, and the IP address is IP\(_{\text{CloudA}}\).

In order to protect the location privacy of the reader, we design the following message transmission mechanism between the reader and the cloud database.

### 3.2.1. The Reader Sends Message \(m_1\) to the Cloud Server

As shown in Figure 3, assume that the reader registers to the general access point AP\(_1\) (its corresponding IP Address is IP\(_1\)); it will send the message \(m_1\) to the cloud server by IP packets, where each IP packet is like the structure shown in Figure 4.

Under this structure, the reader’s packet can be denoted by IP\(_{\text{CloudA}}\) || IP\(_1\) || IP\(_2\) || … || \(m_1\). When AP\(_1\) receives the packet, it adds its IP address IP\(_1\) to the loose source route field of the packets, namely, IP\(_{\text{CloudA}}\) || IP\(_1\) || IP\(_1\) || … || \(m_1\), and then AP\(_1\) sends the packets to the cloud server by other general access points. When the packet routes to the edge access point AP\(_2\), AP\(_2\) adds its IP address IP\(_2\) to the loose source route field of the packets, namely, IP\(_{\text{CloudA}}\) || IP\(_1\) || IP\(_2\) || … || \(m_1\). Then AP\(_2\) generates a random number \(r\) and encrypts IP\(_1\) and IP\(_2\), respectively. Hence, the packet is \(E(r || IP_{\text{CloudA}} || E(r || IP_1) || IP_2 || … || m_1)\). If the reader registers to the edge access point AP\(_2\) directly, in order to protect the location privacy, AP\(_2\) will construct the packets like \(E(r || IP_{\text{CloudA}} || E(r || IP_2) || IP_2 || … || m_1)\). Finally, AP\(_2\) sends packets to the CloudA.

### 3.2.2. The Cloud Server Sends Message \(m_2\) to the Reader

Firstly, the cloud server sends the response packet IP\(_{\text{CloudA}}\) || \(E(r || IP_{\text{CloudA}} || E(r || IP_1) || IP_2 || … || m_2)\) to AP\(_2\), where \(E(r || IP_1)\) may be \(E(r || IP_2)\) if the reader registers to the edge access point AP\(_2\) directly. AP\(_2\) retrieves the loose source route field and destination IP\(_1\) by decrypting \(E(r || IP_1)\), \(E(r || IP_2)\) and then sends the packet to the mobile reader through the registered general access point AP\(_1\).

Through the above process, the location privacy of the mobile reader is protected because the cloud server cannot obtain the information which the routing packet comes from. So it provides the location privacy protection transmission for the reader to access the cloud database.

---

**4. The Proposed Cloud-Based RFID Mutual Authentication Protocol**

The location privacy mechanism provides a privacy channel for the mobile reader to the cloud server. We propose an RFID mutual authentication protocol based on the above channel.

#### 4.1. Notations

The notations are listed in the Notations.

#### 4.2. Initial Phase

The tag stores its identity, id\(_T\), and secret key, Key, and calculates \(H(\text{Key} \parallel \text{id}_T)\). The cloud database stores the EHT \{\(H_{\text{old}}(\text{id}_T \parallel \text{Key})\), \(E_{\text{old}}(\text{id}_T \parallel \text{Key})\), \{\(H_{\text{new}}(\text{id}_T \parallel \text{Key})\), \(E_{\text{new}}(\text{id}_T \parallel \text{Key})\}\} in the cloud database. And the old record and the new record are set to the same value. The reader shares the key kRC with the cloud server.

#### 4.3. The Authentication Phase

The proposed RFID mutual authentication is illustrated in Figure 5. The authentication process is as follows.

1. \(R \rightarrow T\): Request \||\ r_1\.
   - The mobile reader generates a random number \(r_1\) and sends Request \||\ r_1\ to the tag.
2. \(T \rightarrow R\): \(r_2\ ||\ H'(\text{Key} \parallel \text{id}_T)\ ||\ M_1\.
   - The tag generates a random number \(r_2\), calculates \(H'(\text{Key} \parallel \text{id}_T)\) and \(M_1 = H(\text{Key} \parallel \text{id}_T \parallel r_1)\), and sends \(r_2\ ||\ H'(\text{Key} \parallel \text{id}_T)\ ||\ M_1\) to the reader.
3. \(R \rightarrow C\): \(U_A\ ||\ id_A\ ||\ H'(\text{Key} \parallel \text{id}_T)\ ||\ MAC_{kRC}\.
   - The reader stores \(M_1\ and\ sends\ \ U_A\ ||\ id_A\ ||\ H'(\text{Key} \parallel \text{id}_T)\ ||\ MAC_{kRC}\) to the cloud server.

\(U_A\ denotes\ the\ identity\ of\ the\ user\ with\ which\ the\ cloud\ server\ only\ needs\ to\ search\ for\ the\ data\ belonging\ to\ the\ user,\ which\ is\ able\ to\ improve\ the\ efficiency\ of\ the\ searching\ operation.\ MAC_{kRC}\ denotes\ the\ message\ authentication\ code\ with\ the\ shared\ key\ kRC.

4. \(C \rightarrow R\): \(E_k(\text{id}_T)\ ||\ MAC_{kRC}\.
   - If \(MAC_{kRC}\ is\ authenticated\ by\ the\ cloud\ with\ kRC,\ the\ cloud\ server\ extracts\ \(H'(\text{Key} \parallel \text{id}_T)\ and\ searches\ the\ record\ stored\ in\ the\ cloud\ database.\ The\ results\ include\ three possible cases as follows.

**Case 1.** There is no matching record; the authentication is failed.

**Case 2.** If \(H(\text{Key} \parallel \text{id}_T)\ equals \(H'(\text{Key} \parallel \text{id}_T)\), it indicates that both the cloud server and the tag carried out the update normally last time. Then the cloud server sends \(E_k(\text{Key} \parallel \text{id}_T) || MAC_{ARC}\) to the reader.

**Case 3.** If \(H(\text{Key}_{\text{old}} \parallel \text{id}_T) equals \(H'(\text{Key} \parallel \text{id}_T)\), it indicates that the cloud server carried out the update normally, while the tag did not renew its key last time. Then the cloud server sends \(E_k(\text{Key}_{\text{old}} \parallel \text{id}_T) || MAC_{ARC}\) to the reader.

5. \(R \rightarrow C\): \(H(\text{Key}_{\text{new}} \parallel \text{id}_T) || E_k(\text{Key}_{\text{new}} \parallel \text{id}_T) || H(\text{Key} \parallel \text{id}_T) || E_k(\text{Key} \parallel \text{id}_T) || MAC_{ARC}\.
   - The reader decrypts \(E_k(\text{Key} \parallel \text{id}_T)\ to get Key and id\(_T\), and it checks whether \(H(\text{Key} \parallel \text{id}_T \parallel r_1)\ equals \(M_1\). If it succeeds, the tag is authenticated by the reader. Then it calculates \(Key_{\text{new}} = H(\text{Key} \parallel r_1 \parallel r_2)\), \(H(\text{Key}_{\text{new}} \parallel \text{id}_T)\),
5. Analysis of the Security Goals.

Our protocol provides mutual authentication between the tag and the reader, which achieves the following security goals.

Tag Tracking. In each time of the authentication, the tag and reader generate random numbers \(r_2\) and \(r_1\), respectively. The tag’s responses are changed in each session using the updated tag’s secret and fresh random numbers. After receiving the request, Request \(\parallel r_1\), the tag computes \(M_1 = H(\text{Key} \parallel id_T \parallel r_1)\) and sends \(r_2, M_1, H(\text{Key} \parallel id_T \parallel r_2)\) to the reader; thus the attacker obtains new different responses every time when he/she eavesdrops on a session. So he/she cannot track the tag.

Mutual Authentication. Mutual authentication means that the tag and the reader could authenticate each other in the execution of the authentication. In the proposed scheme, the reader authenticates the tag by checking the correctness of \(M_1\) and the tag authenticates the reader by checking the correctness of \(H(id_T \parallel \text{Key}_{new})\).

Forward Security. Forward security means that an adversary cannot obtain previous key by an entity even if he/she compromises it. In the proposed protocol, the Key begins the update by computing hash function of the random numbers and \(\text{Key}_{old}\), and the adversary cannot impersonate the process of the update. So the adversary cannot figure out the previous Key even if he/she intercepts a lot of messages.

Location Privacy of the Reader. In our scheme, we introduce a location privacy cloud which can protect the reader’s location information from leaking to the cloud server and the adversary.

Replay Attack. Having intercepted previous communication, the attacker can replay the same message of the receiver or the sender to pass the verification of the system. However, the random numbers and the secret key are different in each
Table 1: Security comparisons.

| Feature                              | [12] | [13] | [14] | The proposed protocol |
|--------------------------------------|------|------|------|-----------------------|
| Tag tracking                         | No   | Yes  | Yes  | Yes                   |
| Mutual authentication                | Yes  | Yes  | Yes  | Yes                   |
| Forward security                     | Yes  | Yes  | No   | Yes                   |
| Location privacy of the reader       | No   | No   | No   | Yes                   |
| Replay attack                        | Yes  | Yes  | Yes  | Yes                   |
| Desynchronization attack             | Yes  | Yes  | Yes  | Yes                   |

Table 2: Performance comparisons.

| Feature                             | [12]          | [13]          | [14]          | The proposed protocol |
|-------------------------------------|---------------|---------------|---------------|-----------------------|
| Tag's computation cost              | $4h + 1r + 1\text{xor}$ | $5h + 1r + 2\text{xor}$ | $4h + 2r + 7\text{xor}$ | $3h + 1r + 1\text{xor}$ |
| Tag's storage spaces                | $3L$          | $2L$          | $4L$          | $2L$                  |
| Tag's communication message         | $3l$          | $3l$          | $3l$          | $3l$                  |
| Efficiency of the cloud database    | Low           | Low           | Medium        | High                  |
| Deployment cost                     | High          | Medium        | Medium        | Low                   |

$h$: the computing cost of a hash function; $r$: the computing cost of generating a random number; $L$: the length of the tag's identity and secret; $l$: the length of the hash output; xor: the computing cost of a bitwise XOR operation.

Session and there is no regular pattern to follow for the adversary. The replay attack will fail in the proposed protocol.

Desynchronization Attack. During a successful tag query, both the reader and the tag are in synchronization. In fact, the adversary can hamper the communication between the reader and the tag so that the keys stored in the database and the tag will be out of sync. In the proposed protocol, the cloud database stores \{${H_{\text{old}}(\text{id}_T \parallel \text{Key})}$, $E_{\text{old}}(\text{id}_T \parallel \text{Key})$\}, \{${H_{\text{new}}(\text{id}_T \parallel \text{Key})}$, $E_{\text{new}}(\text{id}_T \parallel \text{Key})$\} pairs. So when the adversary interrupts the update process of the session, it will recover to synchronization in the next run.

5.2. The Formal Security Verification under the AVISPA Tool. In our analysis, we have used a formal verification tool called AVISPA (Automated Validation of Internet Security Protocols and Applications) [15]. It is considered the best suitable one for verifying security properties [16]. It provides a modular and expressive formal language for specifying protocols and their security properties and integrates different backends that implement a variety of automatic protocol analysis techniques. AVISPA requires the protocol specification to be written in HLPSL (High Level Protocol Specification Language), which is then provided as an input to the tool. User establishes a secure analysis model by inputting participant identification, operating environment, attacker's ability, and goals of the protocol.

We describe the interactive process of the proposed protocol in HLPSL and define the ability of the adversary. The specification of the proposed protocol contains three basic roles: the tag, the reader, and the cloud database. Each role contains a list of global and local variables and it defines transitions that usually describe the receipt of a message and the sending of a reply. Once roles are defined, we have added the protocol session, the environment section, and the list of declarations goals for security properties. The adversary can obtain the hash function and all the random numbers in our definition.

We verify the security of the proposed protocol and the result shows that it is safe. The HLPSL code and the result are shown in the appendix.

5.3. Security Comparisons. Security comparisons between the proposed protocol and the other schemes are listed in Table 1.

5.4. Performance and Cost Comparisons. We analyze the performance of our method in terms of tag’s computation cost, tag’s storage spaces, tag’s communication message, the efficiency of the cloud database, and so forth. Performance comparisons between the proposed protocol and the other existing schemes are listed in Table 2.

By analyzing and comparing these schemes in security and performance, our method not only meets the security requirements, but also has advantages in deployment cost, tag’s computation cost, and the efficiency of the cloud database.

6. Conclusion

In this paper, we proposed an RFID mutual authentication protocol based on location privacy cloud. Our scheme not only protects the reader’s location privacy, but also has advantages in security and performance. Compared with the existing cloud-based RFID systems, our proposed scheme is more efficient in terms of communication overhead and memory requirement while offering higher level of security. We proved that our proposed scheme is secure against the relevant attacks and also ensures a higher security level and good performance compared with the existing similar
HLPSL code:

// Definition of the tag
role role_Aa(Aa:agent,Bb:agent,R2:text,H:function,Key:text,SND,RCV:channel(dy))
played_by_Aa
def=
local
  State:nat,R1:text,Keynew:text
init
  State:=0
transition
  1. State=0 ∧ RCV(R1') =|> State' :=1 ∧ SND(R1'.R2.H(Key.Aa).H(Key.Aa.R1'))
  6. State=1 ∧ RCV(exp(H(Key,R2).Keynew').H(Aa.Keynew')) =|> State' :=2
endrole

// Definition of the reader
role role_Bb(Bb:agent,Cc:agent,R1:text,H:function,K:symmetric_key,KRC:symmetric_key,SND,RCV:channel(dy))
played_by_Bb
def=
local
  State:nat,MAC:text,Keynew:text,Aa:agent,R2:text,Key:text
init
  State:=0
transition
  1. State=0 ∧ RCV(start) =|> State' :=1 ∧ SND(R1)
  2. State=1 ∧ RCV(R1.R2'.H(Key'.Aa').H(Key'.Aa'.R1')) =|> State' :=2 ∧
     witness(Bb,Aa,auth,2,Key') ∧ MAC' :=new() ∧ SND(Bb.R1.H(Key'.Aa').{MAC'}_KRC)
  4. State=2 ∧ RCV(R1.[Key.Aa]_K.[MAC]_KRC) =|> State' :=3 ∧ Keynew' :=new() ∧
     SND(R1.H(Keynew'.Aa).{Keynew'.Aa}_K.[MAC]_KRC) ∧ SND(exp(H(Key,R2).Keynew').H(Aa.Keynew'))
endrole

// Definition of the cloud server
role role_Cc(Cc:agent,Bb:agent,KRC:symmetric_key,K:symmetric_key,H:function,SND,RCV:channel(dy))
played_by_Cc
def=
local
  State:nat,Key:text,MAC:text,R1:text,Aa:agent,Keynew:text
init
  State:=0
transition
  3. State=0 ∧ RCV(Bb.R1'.H(Key'.Aa').{MAC'}_KRC) =|> State' :=1 ∧ SND(R1'.[Key'.Aa']_K.[MAC']_KRC)
  5. State=1 ∧ RCV(R1.H(Keynew'.Aa).{Keynew'.Aa}_K.[MAC]_KRC) =|> State' :=2
endrole

session1(R1:text,Aa:agent,R2:text,Key:text,Cc:agent,Bb:agent,KRC:symmetric_key,K:symmetric_key,H:function)
def=
local
  SND3,RCV3,SND2,RCV2,SND1,RCV1:channel(dy)
composition
  role_Cc(Cc,Bb,KRC,K,H,SND3,RCV3) ∧ role_Bb(Bb,Cc,R1,H,K,KRC,SND2,RCV2) ∧
  role_Aa(Aa,Bb,R2,H,Key,SND1,RCV1)
endrole

role environment()
def=
const
hash_0:function, k:symmetric_key, reader:agent, key:text, tag:agent, r1:text, r2:text, cloudDatabase:agent, kRC:symmetric_key, h:function, auth_1:protocol_id, auth_2:protocol_id

// Attacker's knowledge
intruder_knowledge = [h, r1, r2]

composition

session(r1, tag, r2, key, cloudDatabase, reader, kRC, k, h)

endrole

// Goals
goal
authentication_on auth_1
authentication_on auth_2
endgoal

evironment()
[10] L. Chu and S.-J. Wu, “An integrated building fire evacuation system with RFID and cloud computing,” in Proceedings of the 7th International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIHMS '11), pp. 17–20, IEEE, Dalian, China, October 2011.

[11] D. Guinard, C. Floerkemeier, and S. Sarma, “Cloud computing, REST and mashups to simplify RFID application development and deployment,” in Proceedings of the 2nd International Workshop on the Web of Things (WoT '11) in Conjunction with the 9th International Conference on Pervasive Computing, San Francisco, Calif, USA, June 2011.

[12] W. Xie, L. Xie, C. Zhang, Q. Zhang, and C. Tang, “Cloud-based RFID authentication,” in Proceedings of the IEEE International Conference on RFID, pp. 168–175, May 2013.

[13] S. Abughazalah, K. Markantonakis, and K. Mayes, “Secure improved cloud-based RFID authentication protocol,” in Data Privacy Management, Autonomous Spontaneous Security, and Security Assurance, pp. 147–164, Springer International Publishing, 2015.

[14] I.-C. Lin, H.-H. Hsu, and C.-Y. Cheng, “A cloud-based authentication protocol for RFID supply chain systems,” Journal of Network and Systems Management, 2014.

[15] The AVISPA Project, http://www.avispa-project.org/.

[16] K. Vishesh and A. Verma, “Formal verification of authenticated AODV protocol using AVISPA,” International Journal of Computer Applications, vol. 50, no. 19, pp. 38–43, 2012.