sensors

MDPI

Article

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

Khwaja Mansoor [1,†], Anwar Ghani [2,†], Shehzad Ashraf Chaudhry [3,†], Shahaboddin Shamshirband [4,5,*,†], Shahbaz Ahmed Khan Ghayyur [2,†] and Amir Mosavi [6,7,†]

1 Department of Computer Science, Air University Islamabad, Islamabad 44000, Pakistan; kh.mansoorulhassan@gmail.com
2 Department of Computer Science & Software Engineering, International Islamic University Islamabad, Islamabad 44000, Pakistan; anwar.ghani@iiu.edu.pk (A.G.); shahbaz.ahmed@iiu.edu.pk (S.A.K.G.)
3 Department of Computer Engineering, Faculty of Engineering and Architecture, Istanbul Gelisim University, Istanbul 34310, Turkey; Shahzad@iiu.edu.pk
4 Department for Management of Science and Technology Development, Ton Duc Thang University, Ho Chi Minh City, Viet Nam
5 Faculty of Information Technology, Ton Duc Thang University, Ho Chi Minh City, Viet Nam
6 Faculty of Health, Queensland University of Technology, Victoria Park Road, Kelvin Grove, QLD 4059, Australia; amir.mosavi@kvk.uni-obuda.hu
7 Kando Kalman Faculty of Electrical Engineering, Obuda University, 1034 Budapest, Hungary
* Correspondence: shahaboddin.shamshirband@tdtu.edu.vn
† These authors contributed equally to this work.

Received: 22 July 2019; Accepted: 12 September 2019; Published: 1 November 2019

Abstract: Despite the many conveniences of Radio Frequency Identification (RFID) systems, the underlying open architecture for communication between the RFID devices may lead to various security threats. Recently, many solutions were proposed to secure RFID systems and many such systems are based on only lightweight primitives, including symmetric encryption, hash functions, and exclusive *OR* operation. Many solutions based on only lightweight primitives were proved insecure, whereas, due to resource-constrained nature of RFID devices, the public key-based cryptographic solutions are unenviable for RFID systems. Very recently, Gope and Hwang proposed an authentication protocol for RFID systems based on only lightweight primitives and claimed their protocol can withstand all known attacks. However, as per the analysis in this article, their protocol is infeasible and is vulnerable to collision, denial-of-service (DoS), and stolen verifier attacks. This article then presents an improved realistic and lightweight authentication protocol to ensure protection against known attacks. The security of the proposed protocol is formally analyzed using Burrows Abadi-Needham (BAN) logic and under the attack model of automated security verification tool ProVerif. Moreover, the security features are also well analyzed, although informally. The proposed protocol outperforms the competing protocols in terms of security.

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

1. Introduction

Since its inception, the Internet of Things (IoT) is an emerging idea and is defined as, "A system of interrelated computing devices, mechanical and digital machines, objects, animals, or people that are provided with unique identifiers (UIDs) and the ability to transfer data over a network without requiring human-to-human or human-to-computer interaction" [1]. The devices are equipped with

internet and are capable of communicating with other devices, and such systems are administered and monitored remotely [2,3]. The IoT assimilates heterogeneity of networks, such as smart cities, sensor networks, smart grids, Radio Frequency Identification (RFID), and transportation and parking systems. The RFID is also on its way to replace conventional bar code systems, as the latter have limitations, including line of sight communication, very limited storage capacity, and prone to physical damage. The overall RFID system architecture is depicted in Figure 1.

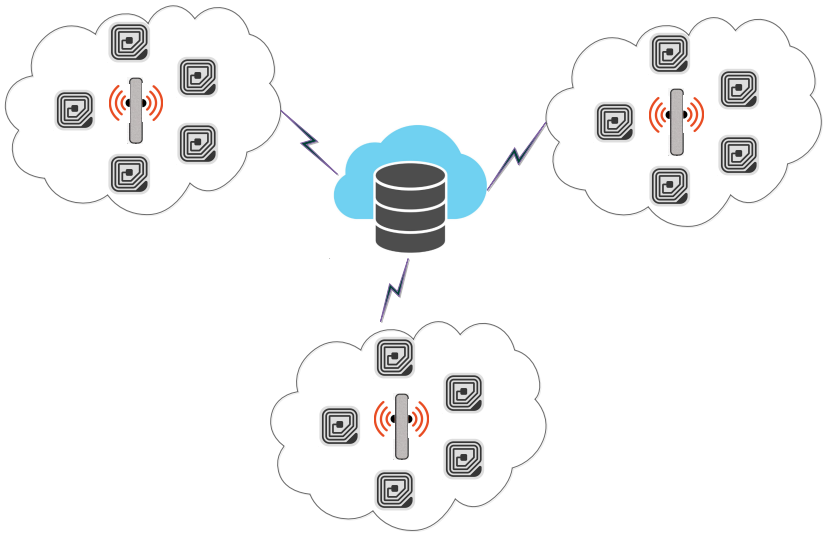

Figure 1. Radio Frequency Identification (RFID) System Architecture.

RFID is simplest form of pervasive sensor networks and is commonly used for identification of physical objects [4,5]. Systems based on RFID consist of a tag, which is equipped with a transceiver to send and receive radio signals from connected devices [6,7]. The RFID Reader is the other device which acts as an access point and can receive and send messages to transceivers. The Reader is also responsible for the availability of tag information at application level [8–10]. RFID tags can be passive, as well as active. Table 1 summarizes the features of passive and active tags.

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 100 mph

RFID systems are typically used for object tracking and identification purposes. The system application accessed through Reader can perform data processing for onward usage in a range of applications like: Asset Tracking, Race Timing, E-Passport, Transportation, Payments, Human Implants, Supply-Chain-Management, Fleet and Asset-Management, Security Access-Control, E-Commerce, and Traffic Analysis and Management [9,11–14]. The IoT-enabled RFID system facilitates all such systems without any physical exposure and in bulk. However, such facilities come with security threats because of the underlying wireless media used for the communication between the tag and the Reader [15–17]. To make an RFID system acceptable and meet industrial standards, the following security features should be considered during the design phase of RFID security schemes:

1. The security scheme should preserve user privacy and anonymity.
2. The scheme should ensure forward and backward secrecy.

3. The scheme should prevent insider attacks and replay attacks.
4. The system should have capabilities to withstand impersonation and forgery attacks.
5. The system should provide mutual authentication and thwart man in middle attack.
6. The system should be user-friendly and should have the provision of updation and alteration of tag data at any time.

Various authentication protocols have been proposed for securing RFID systems [3,12,13,17–33]. Some of these protocols are based on public key infrastructure (PKI) [12,18,19,32,33]. Due to the resource-constrained nature of RFID, the protocols based on PKI are unenviable. Some of the schemes have been proposed on lightweight cryptographic primitives. However, many such schemes based on merely the lightweight primitives were proved as insecure [23,27,29–31].

In 2005, Yang et al. proposed an RFID authentication protocol based on only exclusive-OR (XOR) and hash functions [23]. Some other protocols were also proposed in [21,26,30], using only lightweight hash, XOR and/or symmetric encryption. Despite their [21,23,26,30] claims to provide flawless security, Piramuthu [28] proved that the protocols in [23,30] are vulnerable to replay attack, the protocol in [26] is vulnerable to impersonation of the tag and the Reader, and protocol proposed by Cai et al. [30] is vulnerable to denial-of-service (DoS) and impersonation of tag. Cho et al. [31] then proposed another hash-based protocol for securing RFID. However, Safkhani et al. [29], through cryptanalysis, proved that Cho et al.'s protocol is insecure against DoS, as well as impersonation attacks. Another authentication protocol for securing RFID systems using only symmetric key operations was proposed by Ayaz et al. [17]. However, in their protocol [17], the authentication is performed on the basis of biometrics verification. Such biometric verification may not be desirable in many scenarios, like anti-counterfeiting of life saving drugs, recording and counting number of specific goods moving in and out of a store, etc.

1.1. Motivations and Contributions

Quite recently, Gope and Hwang [3] argued that the existing protocols [21,23,26,29–31] based on hash functions are impractical. Then Gope and Hwang presented a new lightweight authentication protocol using only hash functions. They claimed to avoid all known attacks while maintaining efficiency. However, in this paper, we show that the protocol of Gope and Hwang is vulnerable to collision, DoS, and stolen verifier attacks. Moreover, this article presents an improved and robust protocol using only lightweight symmetric cryptography primitives for IoT-based RFID systems to resist all known attacks. The general contributions of this article include:

- Cryptanalysis of the baseline [3] protocol.
- Proposed an improved authentication protocol using only lightweight symmetric key primitives to overcome the security issues of the baseline protocol.
- Performed formal and informally security analysis of the proposed protocol.
- Solicited the comparison of the proposed protocol with related existing protocols with respect to security features.
- Accomplished the comparison of the proposed protocol with related existing protocols with respect to performance, including communication, as well as computation complexity.

1.2. Adversarial Model

The proposed protocol is designed keeping in mind the following adversarial model where common assumptions as pointed out in [34] are made. The following assumptions are considered as the capabilities of the adversary $\mathcal{A}$.

1. The public channel is under full control of $\mathcal{A}$, so that the $\mathcal{A}$ can intercept, revert, modify, replay, or even send a fresh fabricated message.
2. $\mathcal{A}$ has the capability to extract some of the information of the tag by power analysis. However, shared key of the tag and Server is secret and is inaccessible to any adversary.

3. $\mathcal{A}$ can be any deceitful tag or an outsider of the system.
4. The database attached to the Server is inaccessible, and no adversary $\mathcal{A}$ can access the private key of the Server.

1.3. Road Map

The rest of the article is organized in various sections. In Section 1.4, a brief overview of the protocol of Gope and Hwang [3] is presented. Section 2 presents the proposed protocol, whereas Section 3 presents the detail security analysis of the proposed protocol. Section 4 presents the comparative analysis of the proposed protocol with existing protocols, and, finally, Section 5 concludes the article.

1.4. Review of Baseline Protocol

This section first reviews the baseline protocol of Gope and Hwang's [3] and then performs its cryptanalysis. Table 2 presents some of the notations used in the baseline protocol. The proposed protocol designed for RFID consists three main entities: (1) Database Server, (2) Reader Device, and, (3) RFID tags. The network layout of the RFID System divided into several RFID clusters. Every cluster consists of a Reader and many tags. Tags can shift from one cluster to another. Every Reader of the cluster authenticates the registered tags through the Database Server. Each Reader and Database Server share a symmetric key K_{rs} [3]. Gope and Hwang's [3] authentication scheme consists of two main phases: (1) tag Registration Phase and (2) tag Authentication Phase.

Table 2. Notation Guide.

Notations	Description
T	RFID-tag
R	Reader Device
S	Database Server System
ID_{T_i}	ith tag identity
AID_T	One-time tag alias identity
SID	Shadow identity
R_j	jth Reader identity
N_t	tag Random number
N_r	Reader Random number
K_{ts}	Shared key of Server and tag
K_{emg}	Shared emergency key of Server and tag
K_{rs}	Server and Reader shared 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

1.5. Baseline Protocol Tag Registration Phase

The following steps, as shown in Figure 2, are performed for tag registration:

Step BLR 1: $\mathbf{tag_i} \xrightarrow{\mathbf{ID_{T_i}}} \mathbf{S}$

Each tag (tag_i) submits ID_{T_i} to the Server S.

Step BLR 2: $\mathbf{S} \xrightarrow{\mathbf{M}} \mathbf{tag_i} : \langle \mathbf{M} = \{\mathbf{K_{ts}}, (\mathbf{SID}, \mathbf{K_{emg}}), \mathbf{Tr_{seq}}, \mathbf{h}(.)\}\rangle$

S generates random number n_s and computes $K_{ts} = h(ID_{T_i} \| n_s \oplus ID_s)$. S then generates a set of unlikeable shadow identities ID_s, and $SID = \{sid_1, sid_2 \cdots\}$, where the $sid_j \in SID$. S computes $sid_j = h(ID_{T_i} \| r_j \| K_{ts})$. Further, S generates a set of emergency keys $K_{emg} = \{k_{emg_1}, k_{emg_2} \cdots\}$, each of the keys corresponding to specific $sid_j \in SID$, where each $k_{emg_i} \in K_{emg}$. S then computes $k_{emg_i} = h(ID_{T_i} \| sid_j \| r_j)$. Then S generates a 32-bit random sequence number Tr_{seq} and random

number m and matches it with Tr_{seq}, $Tr_{seq} = m$. S then sends the Tr_{seq} to the tag_i through Reader R_i by maintaining the copy of Tr_{seq} in its database for speeding up the authentication process. S authenticates the validity of RFID tag ID_{T_i} based on TR_{seq}. If TR_{seq} does not have a match within the record of S, it terminates the process. In this case, the RFID tag ID_{T_i} will use one of its fresh pair of the emergency key $k_{emg_j} \in K_{emg}$ and shadow ID $sid_j \in SID$. The used pair of shadow ID and emergency ID (SID, K_{emg}) must be deleted from both, the Database Server S and the RFID tag ID_{t_i}. Database Server S again updates and send $\{K_{ts}, (SID, K_{emg}), Tr_{seq}, h(.)\}$ through a secure channel for further communication.

Step BLR 3: tag_i, upon receiving message from S, stores $\{ID_{T_i}, K_{ts}, (SID, K_{emg}), Tr_{seq}, h(.)\}$ in its memory.

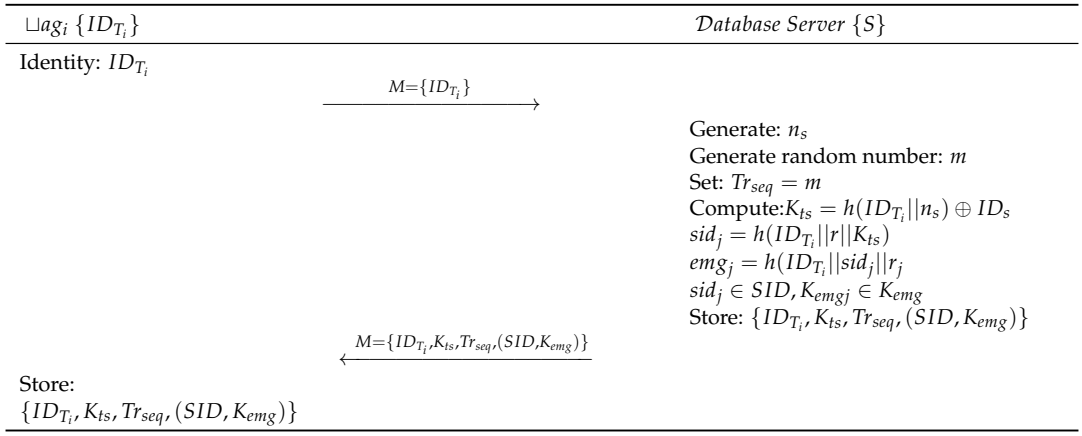

Figure 2. Gope-Hwang's proposed registration scheme.

1.6. Baseline Protocol Tag Authentication Phase

The registered tag initiates the authentication process, as shown in Figure 3, and is detailed as follows:

Step BLA 1: $\mathbf{tag_i} \xrightarrow{\mathbf{M_{A_1}}} \mathbf{R_i} : \langle \mathbf{M_{A_1}} = \{\mathbf{AID_T}, \mathbf{N_x}, \mathbf{Tr_{seq}}, \mathbf{V_1}\} \rangle$

tag_i with identifier ID_{T_i} generates random number N_t, and derives $AID_T = h(ID_{T_i} \| K_{ts} \| N_t \| Tr_{seq})$, $N_x = K_{ts} \oplus N_t$. The tag then computes $V_1 = h(AID_{ti} \| K_{ts} \| N_x \| R_i)$ and sends message request as M_{A_1} to the Reader device R_i. R_i also receives a recently used sequence number from S for mutual authentication. In the case of synchronization loss, the tag uses one of its fresh pair (sid_j, K_{emg_j}). Subsequently, it is assigned to the sid_j as AID_T and then k_{emg_j} as K_{ts}. tag_i sends M_{A_1} to the Reader R_i.

Step BLA 2: $\mathbf{R_i} \xrightarrow{\mathbf{M_{A_2}}} \mathbf{S} : \langle \mathbf{M_{A_2}} = \{\mathbf{N_y}, \mathbf{R_i}, \mathbf{V_2}, \mathbf{M_{A_1}}\} \rangle$

Upon receiving request from tag_i, Reader R_i of the i^{th} cluster (in which tag_i is located) generates random number N_r and computes $N_y = K_{rs} \oplus N_r$, $V_2 = h(M_{A_1} \| N_r \| K_{rs})$. R_i then sends M_{A_2} to S for verification.

Step BLA 3: $\mathbf{S} \xrightarrow{\mathbf{M_{A_3}}} \mathbf{R_i} : \left\langle \mathbf{M_{A_3}} = \{\mathbf{T_r}, \mathbf{V_3}, \mathbf{V_4}, \mathbf{x_{(ifreq.)}}\} \right\rangle$

When S receives a request from R_i, first it validates the track sequence number Tr_{seq} by computing $V_1 = h(AID_T \| K_{ts} \| N_x \| R_i)$. S then derives $N_t = K_{ts} \oplus N_x$ and verifies AID_T. Upon successful verification of AID_T, S generates a random number m and assigns it to $T_{r_{seq}} = m$. S also computes $T_r = h(K_{ts} \| ID_{T_i} \| N_t) \oplus T_{r_{seq}}$, $V_4 = h(T_r \| K_{ts} \| ID_{T_i} \| N_t)$, $V_3 = h(R_i \| N_r \| K_{rs})$ to create a message M_{A_3} and the S sends M_{A_3} to R_i. Finally, S computes $K_{Ts_{new}} = h(K_{ts} \| ID_{T_i} \| Tr_{seq_{new}})$ and updates $K_{Ts_{new}}$ and $Tr_{seq_{new}}$. In case the message M_{A_1} does not contain $T_{r_{seq}}$, then S randomly generates a new shared key $K_{TS_{new}}$ using the emergency key K_{emg_j} and real identity of the tag ID_{T_i}. Then $x = K_{ts_{new}} \oplus h(ID_{T_i} \| K_{emg_j})$ is computed and x is sent with the message M_{A_3}, where V_4 is calculated as $V_4 = h(N_t \| T_r \| x \| K_{emg_j})$.

Step BLA 4: $\mathbf{R_i} \xrightarrow{\mathbf{M_{A_4}}} \mathbf{tag_i} : \left\langle \mathbf{M_{A_4}} = \{\mathbf{T_r}, \mathbf{V_4}, \mathbf{x_{(ifreq.)}}\} \right\rangle$

R_i receives M_{A_3} and computes $h(R_i \| N_r \| K_{rs})$, and validates if it is equal to V_3. Upon successful validation, R_i sends M_{A_4} to tag_i. Contrarily, the Reader R_i terminates the session.

Step BLA 5: tag_i, on receiving M_{A_4}, computes $h(Tr \| K_{ts} \| ID_{T_i} \| N_t)$ and verifies its equality with V_4. Upon success, tag_i derives $K_{ts_{new}} = h(K_{ts} \| ID_{T_i} \| Tr_{seq_{new}})$ and stores $K_{ts} = K_{ts_{new}}$, $Tr_{seq} = Tr_{seq_{new}}$ for future communication.

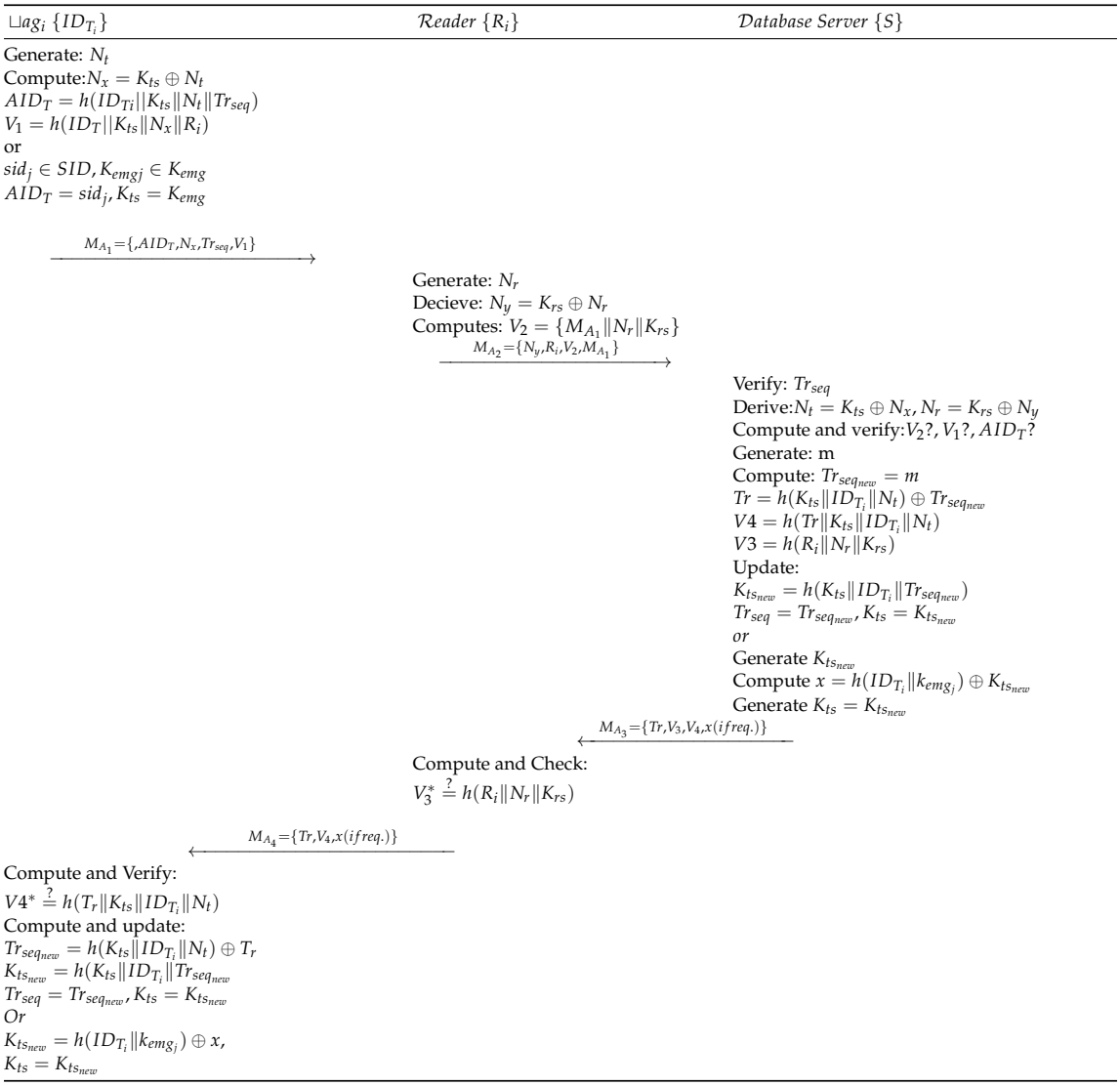

Figure 3. Gope-Hwang's proposed authentication scheme.

1.7. Cryptanalysis of Baseline Protocol

The following subsections show that the baseline protocol is vulnerable to: (1) Collision, (2) Stolen Verifier, and (3) DoS Attacks.

1.7.1. Vulnerable to Collision Attack

The correctness of the baseline protocol [3] depends on Track sequence number Tr_{seq}, generated randomly during registration and saved in the database, as well as in the tag's memory. This number Tr_{seq} is sent in authentication request $M_{A_1} = \{AID_T, N_x, Tr_{seq}, V_1\}$ by the tag and then, upon reception of M_{A_1}, the Reader R_i sends $M_{A_2} = \{M_{A_1}, N_y, R_i, V_2\}$ to the Database Server. S verifies the legitimacy of Tr_{seq} by comparing it with the one stored in its database. The randomness can cause two or more

track sequence numbers to have the same value (collision), and there is no mechanism to handle such collisions. Then the process will terminate abnormally, and the legitimate tag will be deprived of its right to authentication and access.

1.7.2. Vulnerable to Stolen Verifier Attack

Considering the common adversarial modal, as mentioned in Section 1.2, an adversary can steal the verifier table stored unencrypted on server. $\mathcal{A}$ based on the track sequence number Tr_{seq} and the public request from any of the previous session $M_{A_1} : \{AID_T, N_x, Tr_{seq}, V_1\}$ and $M_{A_2} : \{N_y, R_i, V_2, M_{A_1}\}$ can then generate login request using the previous session's N_x, N_y and the stolen new Tr_{seq}. The request will pass the authentication test as all values are valid. Hence baseline protocol is also vulnerable to stolen verifier attack.

1.7.3. Vulnerable to DoS Attack

In the baseline proposal [3], an adversary can launch a DoS attack by continuously generating 32 bits random Tr_{seq} numbers and send it to the Database Server. It will keep S busy in verifying dummy random numbers, thus restricting S to serve a legitimate request.

2. Proposed Scheme

Like the baseline protocol, the proposed protocol for RFID consists of three main entities: (1) Database Server, (2) Reader Device, and (3) RFID tags. The network layout of the RFID System is divided into several RFID clusters. Every cluster consists of a Reader and many tags. Tags can shift from one cluster to another. Every Reader of the cluster authenticates the registered tags through the Database Server. Each Reader and Database Server share a symmetric key K_{rs}. Proposed improved authentication scheme consists of two main phases; (1) tag Registration Phase, (2) tags Authentication Phase.

2.1. Tags Registration Phase

The following steps, as shown in Figure 4, are performed for tag registration:

Step PTR 1: $\mathbf{tag_i} \xrightarrow{\mathbf{ID_{T_i}}} \mathbf{S}$

Each tag submits ID_{T_i} to the Server S.

Step PTR 2: $\mathbf{S} \xrightarrow{\mathbf{M}} \mathbf{tag_i} : \langle \mathbf{M} = \{\mathbf{ID_{T_i}}, \mathbf{K_{ts}}, \mathbf{AID}\} \rangle$

S generates a random number n_s and computes $K_{ts} = h(ID_{T_i} \| n_s \oplus ID_s)$. S generates r_i randomly and computes one-time alias tag_i's identity $AID = E_{s_x}(ID_{T_i} \| r_{T_i})$ by encrypting it with the Secret Key s_x of S. S authenticates tag_i based on AID_T in authentication phase by checking if a request is valid or not. S stores and sends M to the RFID tag through a secure channel.

Step PTR 3: Upon receiving the message from S, tag_i stores the information $M = \{ID_{T_i}, K_{ts}, AID\}$ in its memory.

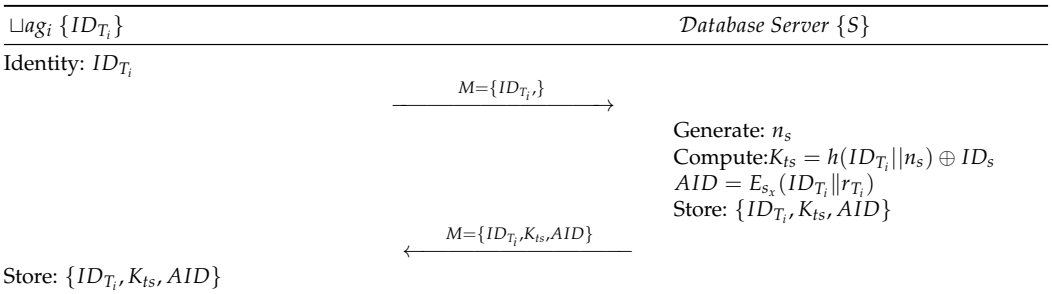

Figure 4. Registration phase of the proposed protocol.

2.2. Tags Authentication Phase

The registered tag initiates the authentication process, as shown in Figure 5, and is detailed as follows:

Step PTA 1: $\mathbf{tag_i} \xrightarrow{\mathbf{M_{A_1}}} \mathbf{R_i} : \langle \mathbf{M_{A_1}} = \{\mathbf{AID_T}, \mathbf{N_x}, \mathbf{T_1}, \mathbf{V_1}\}\rangle$

RFID tag with identifier ID_{T_i} generates a random number N_t and derives $N_x = K_{ts} \oplus N_t$ and $V_1 = h(AID_t \| K_{ts} \| N_x \| R_i)$. The tag then initiates an authentication request request by sending M_{A_1} to R_i.

Step PTA 2: $\mathbf{R_i} \xrightarrow{\mathbf{M_{A_2}}} \mathbf{S} : \langle \mathbf{M_{A_2}} = \{\mathbf{N_y}, \mathbf{R_i}, \mathbf{V_2}, \mathbf{M_{A_1}}, \mathbf{T_2}\}\rangle$

Upon receiving the request from the tag, Reader R_i of the i^{th} cluster (in which tag is located) first verifies the timestamp freshness as $(T_2 - T_1) \leq \Delta T$. R_i generates a random number N_r and computes $N_y = K_{rs} \oplus N_r$, $V_2 = h(M_{A_1} \| N_r \| K_{rs} \| T_2)$. R_i sends M_{A_2} to the S for verification.

Step PTA 3: $\mathbf{S} \xrightarrow{\mathbf{M_{A_3}}} \mathbf{R_i} : \langle \mathbf{M_{A_3}} = \{\mathbf{V_3}, \mathbf{V_4}, \mathbf{Z_T}, \mathbf{T_3}\}\rangle$

When S receives the request from R_i, first it verifies $(T_3 - T_2) \leq \Delta T$, then derives $N_t = K_{ts} \oplus N_x$ and $N_r = K_{rs} \oplus N_y$. Further, S computes and verifies $V_1 = h(AID_T \| K_{ts} \| N_x \| R_i)$, $V_2 = h(M_{A_1} \| N_r \| K_{rs} \| T_2)$. Then S verifies AID_{T_i} by decrypting it as $AID_{T_i} = D_{S_x}(ID_{T_i} \| r_i)$. Upon successful verification, S computes $V3 = h(R_i \| N_r \| K_{rs} \| T_3)$ and $V4 = h(K_{ts} \| ID_{T_i} \| N_t \| T_3)$. S then updates $AID_{T_{i(new)}} = E_{S_x}(ID_{T_i} \| r_{i_{(new)}})$ and computes $Z_T = AID_{T_{new}} \oplus K_{Ts}$. S, finally, sends M_{A_3} to R_i.

Step PTA 4: $\mathbf{R_i} \xrightarrow{\mathbf{M_{A_4}}} \mathbf{tag_i} : \langle \mathbf{M_{A_4}} = \{\mathbf{V_4}, \mathbf{T_4}, \mathbf{Z_T}\}\rangle$

Upon receiving M_{A_3}, R_i checks freshness of the timestamp $(T_4 - T_3) \leq \Delta T$. R_i computes $h(R_i \| N_r \| K_{rs})$ and verifies its equality with the received V_3. Upon success, R_i sends M_{A_4} to tag_i. Otherwise, R_i terminates the session.

Step PTA 5: Upon receiving M_{A_4}, tag_i first checks freshness of the timestamp and upon success verifies the message $V4^* \stackrel{?}{=} h(K_{ts} \| ID_{T_i} \| N_t)$. Then tag_i computes and updates $AID_{T_{i(new)}} = (Z_T \oplus K_{Ts})$, $AID_{T_i} = AID_{T_{i(new)}}$ and saves the information for the next authentication process.

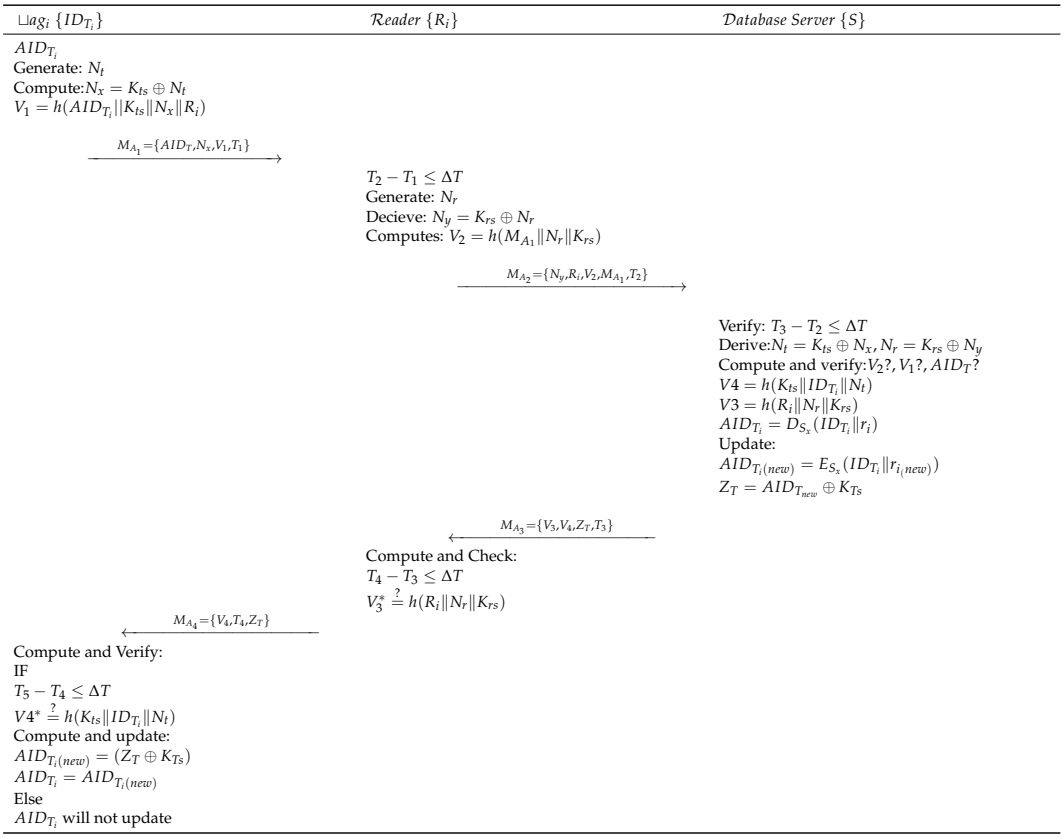

Figure 5. Proposed authentication protocol.

3. Security Analysis

In this section, the security analysis of the proposed protocol under the adversarial model briefed in Section 1.2 is performed. The task is accomplished by formal analysis under Burrows Abadi-Needham (BAN) logic, and informal security features are explained. Moreover, the robustness of the proposed protocol is also analyzed through the automated tool, ProVerif—a widely accepted simulation tool for verification of the security of authentication protocols [35–43].

3.1. BAN Logic-Based Formal Security Analysis

BAN logic consists of a set of rules that can be used to analyzed information exchange protocols. It specifically determines if the information exchanged in a protocol is resistant against eavesdropping and is trustworthy and secured. The mutual authentication of the proposed protocol has been checked using the BAN logic [44]. Different rules of BAN logic, including idealized form, assumptions, and proofs, are shown in Table 3.

Table 3. BAN logic Notations.

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

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

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

To achieve the goals listed above, the security analysis using BAN logic has been divided into three parts. Part1 shows the idealized form of the protocol and is proved in Part3, whereas Part2 uses assumptions to analyzed the proposed protocol.

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

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

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

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

Part 3: Analysis of Idealized form of the proposed protocol that has been derived on the basis of BAN logic assumptions and rules is described as follows:
M1: $tag \rightarrow$Ri: $AIDT_i$,$N_x :< N_t >_{K_{ts}}$, $T1$ is time-stamp of tag. Using the seeing rule, the following can be achieved:

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

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

- S2: $R_i| \equiv tag| \sim N_t$.

Using the freshness-conjuncatenation rule and S2 will achieve the following:

- S3: $R_i| \equiv tag| \equiv N_t$.

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

- S4: $R_i| \equiv N_t$.

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

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

Using the nonce-verification rule, the following is obtained:

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

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

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

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

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

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

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

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

- S10: $S_j| \equiv N_r$.

Using the S10 and the SK rule, the following can achieved:

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

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

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

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

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

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

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

By S14 and the freshness-conjuncatenation rule, the following can achieved:

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

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

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

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

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

Applying nonce-verification rule, the following can be computed:

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

M4: $R_i \rightarrow tag : V4, Zt < AIDT_{i_{new}} >_{K_{ts}}, T4$, $T4$ is timestamp of R_i.
Using the seeing rule, the following can be computed:

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

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

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

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

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

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

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

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

- S23: $tag| \equiv R_i \overset{AIDT_{i_{new}}}{\longleftrightarrow} tag$ **(Goal 7)**.

Finally, using the nonce-verification rule, the following can be achieved, which is also the final goal of the proposed protocol:

- S24: $tag| \equiv R_i| \equiv R_i \overset{AIDT_{i_{new}}}{\longleftrightarrow} tag$ **(Goal 8)**.

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

3.2. Security Analysis with ProVerif

Based on applied π calculus, ProVerif uses automated reasoning to test the security features of authentication protocols. Specifically, ProVerif can verify the reachability, correspondence, and observational equivalence, as well as secrecy properties. ProVerif supports primitive cryptographic operations [45], including MAC, digital signatures, encryption/decryption, elliptic curve operations, hash, and other functions [46]. The steps of the proposed scheme, as illustrated in Section 2 and shown in Figures 4 and 5, are simulated in ProVerif. The formal security validation model of ProVerif consists of three phases: (1) Declaration, as coded in Figure 6A, declares the constants, names, variablesm, and cryptographic function, (2) Process part, as shown in Figure 6B, defines the three processes, each for tag, Reader, and Server, and (3) Main, as implemented in Figure 6C, simulates the actual protocol.

```
(******************** Channels ******************)
free ChSec:channel [private].   (*secure channel*)
free ChPub:channel.
                    (*public channel*)
(******** Constants and Variables ********)
free IDTi :bitstring.
free IDs :bitstring.
free Ri:bitstring.
free Kts : bitstring [private].
(*************** Constructors *************)

fun h(bitstring):bitstring.
fun Inverse(bitstring):bitstring.
fun Concat(bitstring,bitstring):bitstring.
fun XOR(bitstring,bitstring):bitstring.
fun enc(bitstring,bitstring): bitstring.
fun dec(bitstring,bitstring): bitstring.
(*************** Equations ***************)
equation forall a:bitstring; Inverse(Inverse(a))=a.
equation forall a:bitstring, b:bitstring; XOR(XOR(a
    ,b),b)=a.
equation forall x: bitstring, y: bitstring; dec(enc
    (x,y),y) = x.
equation forall x: bitstring, y: bitstring; enc(dec
    (x,y),y) = x.
```

(A) Declarations

```
(******************** Events ********************)
event start_Tag(bitstring).
event end_Tag(bitstring).
event start_R(bitstring).
event end_R(bitstring).
event start_S(bitstring).
event end_S(bitstring).
(********** Process Replication ***************)
process ((!pS) | (!pR) | (!pTag) )
(******************** Queries *******************)
free AIDTinew:bitstring [private].
query attacker(AIDTinew).
query id:bitstring; inj event(end_Tag(IDTi)) ==>
    inj event(start_Tag(IDTi)).
query id:bitstring; inj event(end_R(Ri)) ==> inj
    event(start_R(Ri)).
query id:bitstring; inj event(end_S(IDs)) ==> inj
    event(start_S(IDs)).
```

(C) Main

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

(D) Simulation Results

```
(******************Processes ******************)
(*================RFID Tag================*)
(******** Registration ********)
let pTag=
out(ChSec,(IDTi));
in (ChSec,(IDTi:bitstring,Kts:bitstring,AIDTi:
    bitstring));
(**********Tag login**********)
event start_Tag(IDTi);
new Nt:bitstring;
new T1:bitstring;
let Nx=XOR(Kts,Nt) in
let V1=h(Concat(AIDTi, (Kts,Nx, Ri)))in
out(ChPub,(AIDTi,Nx,V1,T1));
in (ChPub,(V4:bitstring,T4:bitstring,Zt:bitstring))
    ;
let V4=h(Concat(Kts,(IDTi, Nt))) in
if V4=h(Concat(Kts,(IDTi, Nt))) then
let xAIDTinew = XOR(Zt,Kts) in
let AIDTi = AIDTinew in
event end_Tag(IDTi)
else 0.
(*================RFID Reader==============*)
let pR=
event start_R(Ri);
new Nr:bitstring;
new T2:bitstring;
new Krs:bitstring;
new Dj:bitstring;
new Ts:bitstring;
new MA1:bitstring;
in (ChPub,(xxAIDTi:bitstring,Nx:bitstring,V1:
    bitstring,T1:bitstring));
let Ny=XOR(Krs,Nr) in
let V2=Concat(MA1, (Nr,Krs)) in
out(ChPub,(Ny,Ri,V2,MA1,T2));
in (ChPub,(V3:bitstring,V4:bitstring,Zt:bitstring,
    T3:bitstring));
new ZT:bitstring;
new T4:bitstring;
if V3=h((Concat(Ri,(Nr,Krs)))) then
out(ChPub,(V4,T4,ZT));
event end_R(Ri)
else 0.
(*================RFID Server==============*)

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

(B) Processes

Figure 6. ProVerif Simulation.

Simulation of three processes executed in parallel is performed, along with six events to validate the reachability properties of three processes. Finally, four queries are implemented. The results are shown in Figure 6D. Based on the above description of results 1, 2, and 3, all three original processes of the proposed protocol successfully started and terminated. Result 4 shows that the session tag identity AID_{T_i} is safe from any adversary attack. Therefore, the proposed protocol possesses correctness and provides tag secrecy.

3.3. Informal Security Analysis

The proposed protocol for RFID System is analyzed for security loopholes against the known attacks in the following subsections.

3.3.1. Mutual Authentication Between Tag And Server

The RFID Server authenticates RFID tag by verifying a one time alias AID_{T_i} and $V_1 = h(ID_T||K_{ts}||N||R_i)$ in the message M_1. Only a legitimate RFID tag can form a valid request message M_1, including both these parameters, as valid AID_{T_i} is only known to legal tag; moreover, ID_T, K_{ts} are known to the legal tag only. On other side, the RFID tag can authenticate the legitimacy of the Server using parameters V_4 and message M_3 in M_4. This way, the proposed protocol achieves mutual authentication property.

3.3.2. Anonymity

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

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

3.3.3. Traceability

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

3.3.4. Backward/Forward Secrecy

It is essential for security protocols that the information transmitted in a session is not compromised, as well as traced or used by an adversary to create vulnerabilities in the current, previous, or future authentication session between the RFID tag and RFID Server S. In the proposed protocol, even if the identity ID_T or alias identity are lost, it does not affect previous or next sessions. It is ensured through the use of encrypted AID_{Ti}, which is updated in every new session. In this way, the proposed protocol for the RFID System guarantees backward and forward secrecy.

3.3.5. Scalability

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

3.3.6. Collision Attack

If RFID-tags share the same credentials for authentication to access the RFID Server, the protocol may be left vulnerable to a collision attack. In the proposed protocol, every RFID tag uses different

parameters, i.e., $\{N_y, R_i, V_2, M_{A_1}\}$, for authentication that makes it impossible for collision attack to take place.

3.3.7. DoS Attack

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

3.3.8. Replay Attacks

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

3.3.9. Location Tracking Attack

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

3.3.10. Impersonation Attacks (Forgery Attacks)

An adversary A may intercept the messages of the previous legitimate RFID tag and modify that for authentication with the RFID Server S. In this case, the adversary A needs to make a valid message request that includes different parameters, like $N_y, R_i, V_2, M_{A_1}, AID_{Ti}$. To do so, the adversary A must compute AID_{Ti} that is well encrypted and impossible to be computed or forged. Moreover, the adversary A also needs different other parameters and timestamps to put a valid request for authentication as a legitimate RFID tag. It is impossible for the adversary A without knowing the actual parameters of the Message used for authentication, hence leaving the adversary A unable to prove its legitimacy as an RFID tag to the RFID Server S. Reluctantly, the proposed protocol for RFID System resists any forgery attack.

3.3.11. Stolen-Verifier Attacks

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

4. Comparative Analysis

This section presents a comprehensive comparative analysis of the proposed protocol with the existing protocols [3,21,23,30,31], as these schemes are based on lightweight symmetric key primitives. Hence, they are eligible for a fair comparison with the proposed scheme. Firstly, the proposed protocol is compared with the existing protocols in terms of security requirements. Secondly, a comparison of

the proposed protocol with existing protocols based on computation cost (running time or execution time) is given, and thirdly, a comparison based on communication cost is presented. Furthermore, the proposed protocol is analyzed for storage complexity. Please note that we have selected the schemes based on lightweight symmetric key primitives and also that have been published recently. Each of these comparisons has been elaborated in the following subsections, one-by-one.

4.1. Security Requirements

Security requirements are the features expected from an authentication protocol. Every authentication protocol must be able to ensure these features or requirements. By these requirements, the proposed protocol is compared with the existing protocols. Following is the list of features/requirements considered for comparative analysis.

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

Table 4 shows the security requirements comparison of the proposed protocol with existing symmetric key-based protocols [3,21,23,30,31].

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. [3]	Proposed Scheme
SR1	×	×	✓	✓	✓	✓
SR2	×	×	×	✓	✓	✓
SR3	×	×	✓	×	✓	✓
SR4	×	✓	×	✓	✓	✓
SR5	×	×	×	×	✓	✓
SR6	×	×	×	✓	×	✓
SR7	✓	×	✓	✓	×	✓
SR8	✓	✓	✓	✓	✓	✓
SR9	✓	✓	✓	✓	✓	✓
SR10	✓	✓	✓	✓	✓	✓
SR11	✓	✓	✓	✓	×	✓

✓: Yes provides, ×: Does not provide.

The insecurities of the existing schemes [3,21,23,30,31] are well defined in Section 1 and are replicated in Table 4. The security requirements in Table 4 show that only the proposed protocol provides all security features.

4.2. Computation Cost Analysis

This section describes the computation cost analysis of the proposed protocol with existing related protocols [3,21,23,30,31]. For analysis purposes, the following notations are introduced:

- CC: Computation cost;
- T_h: CC of single hash function;
- T_{se}: CC of symmetric encryption/decryption.

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 [3]	Proposed Scheme
CC_{tag}	$2T_h$	$2T_h$	$4T_h$	$3T_h$	$5T_h$	$2T_h$
CC_{R_i}	$3T_h$	$2T_h$	$2T_h$	$2T_h$	$2T_h$	$2T_h$
CC_S	$5T_h$	$3T_h$	$6T_h$	$5T_h$	$7T_h$	$4T_h + 2T_{se}$
CC_{Total}	$10T_h$	$7T_h$	$12T_h$	$10T_h$	$14T_h$	$8T_h + 2T_{se}$
CC_{Time}	0.023 ms	0.0161 ms	0.0276 ms	0.023 ms	0.0322 ms	0.0276 ms

Table 5 shows the computation cost analysis. The protocol presented in [23] incurs $2T_h$,$3T_h$, and $5T_h$, for each tag, Reader and Server, respectively, making its total computation cost $10T_h$. Similarly, the computation cost of protocol presented in [30] is $2T_h$, $2T_h$, and $3T_h$, respectively, for each participant, totaling it to $7T_h$. The protocol presented in [21] requires $4T_h$, $2T_h$, and $6T_h$ for each tag, Reader, and Server, respectively, totaling it to $12T_h$. The computation cost of the protocol of Gope and Hwang [3] is $5T_h$, $2T_h$, and $7T_h$, respectively, for each participant totaling it to $14T_h$. In comparison, the tag in proposed protocol uses $2T_h$, the Reader uses $2T_h$, and the Server requires $4T_h + 2T_{se}$, so in total the computation cost of the proposed protocol is equal to $8T_h + 2T_{se}$. Considering the experiment of Kilinc and Yanik [47], the computation time of T_h is 0.0023 ms, whereas the computation time to calculate T_{se} is 0.0046 ms. The experiment was performed on a Ubuntu system with an Intel dual-core Pentium processor with specifications, including 2.20 GHz, 2048 MB processor and Ram, respectively. The total computation time of the proposed protocol is 0.0276 ms, whereas the total cost of the protocol presented in [23] is 0.0230 ms, the cost of protocol in [30] is approximately 0.0161 ms, the cost of the proposal in [3] is 0.0322 ms, and the proposal in [21] takes a total of 0.0276 ms. Although the proposed protocols incur a slightly higher computation cost as compared with [23,30], it provides less computation cost when compared with the baseline [3] and provides the same computation cost as compared to the protocol presented in [21]. Moreover, the proposed protocol is the only protocol that provides resistance against all known attacks. The results presented in Table 5 are visualized in Figure 7.

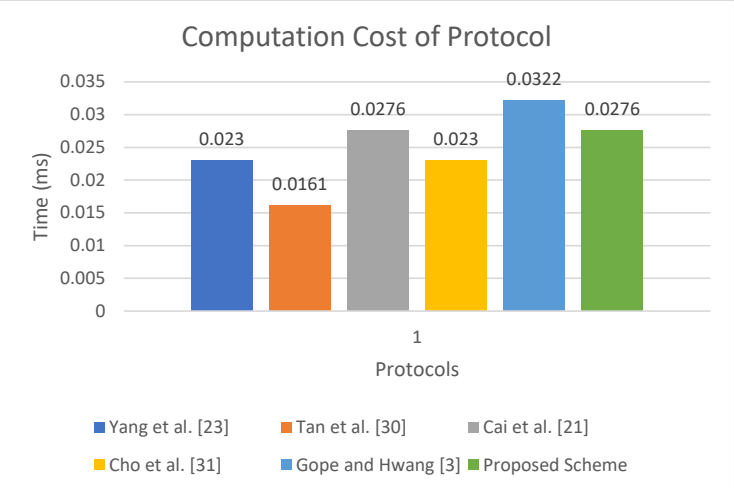

Figure 7. Running Time of Proposed Scheme.

4.3. Communication and Storage Cost Analysis

Communication cost is presented in terms of the total number of messages exchanged and total number of bits exchanged during one transaction of the protocol. In the proposed protocol, tag_i transmits four parameters in M_1 to R_i carrying 416 *bits* and receives 384 *bits* from R_i. Similarly, R_i transmits 736 *bits* and receives 416 *bits* from S, while S transmits 416 *bits* and receives 736 *bits*. The communication cost comparison of the proposed protocol with other existing protocols is presented in Table 6. The storage cost is represented by length Value L, the proposed protocol uses $SHA-1$ hash function to implement $h(.)$; for simplicity, each of the length values is considered as 160-*bit* long.

In the proposed scheme, each tag stores ID_{T_i}, K_{ts}, AID parameters. Therefore, the cost of storage in the tag is $3L$, whereas on the Server side, $ID_{T_i}, K_{ts}, AID_{new}, AID_{old}$ are being stored; hence, the storage cost on the Server side is $4L$ per tag.

Table 6. Communication Cost of Proposed and other Protocols.

Schemes	tag	Reader	Server	Total Bits	Messages
Yang et al. [23]	256	512	640	1408	5
Tan et al. [30]	896	768	768	2432	4
Cai et al. [21]	256	544	256	1056	5
Cho et al. [31]	512	512	256	1280	5
Gope and Hwang [3]	416	1180	288	1888	4
Proposed Protocol	416	736	416	1568	4

The proposed protocol incurs less communication cost as compared with the protocols of [3,30], whereas it has more communication cost when compared with others [21,23,31]. However, only the proposed protocol provides required security. The results presented in Table 6 are visualized in Figure 8.

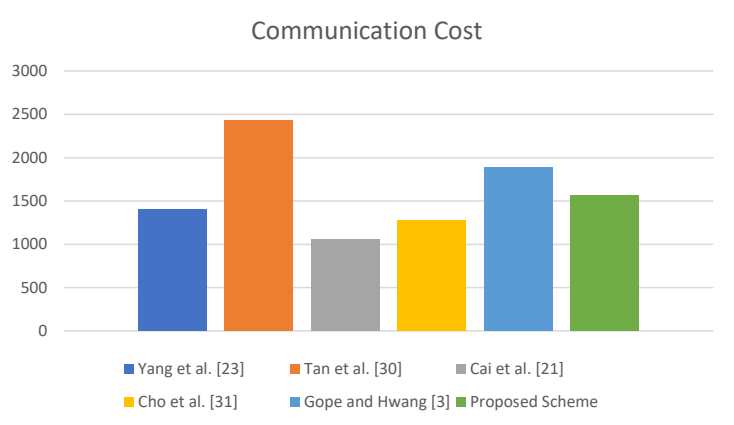

Figure 8. Communication Cost.

Table 6 indicates that the proposed protocol is more efficient than the baseline protocol in terms of communication cost. Specifically, the proposed protocol not only overcomes the security flaws of the baseline protocol but also achieves 16.94% efficiency in terms of the number of bits exchanged during one transaction of the protocol.

5. Conclusions

In this article, cryptanalysis of a recent authentication protocol by Gope and Hwang has been presented, and it has been proved that their protocol has some weaknesses against collision, stolen verifier, and DoS attacks. An improved scheme using only lightweight primitives is proposed to resist all known attacks. The security of the proposed scheme has been thoroughly analyzed informally, as well as formally, using BAN logic. Moreover, the scheme is simulated in automated applied π calculus-based tool ProVerif. The simulation also backs the formal and informal security analysis. Although the proposed scheme incurs some extra computation and communication cost as compared with some existing related protocols, only the proposed protocol resists all known attacks and is more suitable for practical IoT-based scenarios.

Author Contributions: For research articles with several authors, conceptualization, K.M., S.A.C. and A.G.; methodology, K.M. and S.A.C.; software, K.M.; validation, A.G., S.S., A.M., and S.A.K.G.; formal analysis, S.A.C. and A.G.; investigation, K.M.; resources, S.S. and A.M.; data curation, K.M. and A.G.; writing–original draft preparation, K.M. and A.G.; writing–review and editing, S.A.C. and A.G.; visualization, A.G.; supervision, A.G. and S.A.C.; project administration, A.G. and S.A.C.; funding acquisition, S.S. and A.M.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare that they have no conflict of interest.

References

1. Rouse, M. Internet of Things (IoT). Available online: https://internetofthingsagenda.techtarget.com/definition/Internet-of-Things-IoT (accessed on 3 September 2019).
2. Gope, P.; Hwang, T. BSN-Care: A secure IoT-based modern healthcare system using body sensor network. *IEEE Sens. J.* **2016**, *16*, 1368–1376. [CrossRef]
3. Gope, P.; Hwang, T. A realistic lightweight authentication protocol preserving strong anonymity for securing RFID system. *Comput. Secur.* **2015**, *55*, 271–280. [CrossRef]
4. Peris-Lopez, P.; Hernandez-Castro, J.C.; Estevez-Tapiador, J.M.; Ribagorda, A. Lightweight cryptography for low-cost RFID tags. In *Security in RFID and Sensor Networks*; CRC Press: London, UK, 2016; pp. 121–150.
5. Gope, P.; Amin, R.; Islam, S.H.; Kumar, N.; Bhalla, V.K. Lightweight and privacy-preserving RFID authentication scheme for distributed IoT infrastructure with secure localization services for smart city environment. *Future Gener. Comput. Syst.* **2018**, *83*, 629–637. [CrossRef]
6. Kitsos, P. *Security in RFID and Sensor Networks*; CRC Press: New York, NY, USA, 2016.
7. Hsu, C.H.; Wang, S.; Zhang, D.; Chu, H.C.; Lu, N. Efficient identity authentication and encryption technique for high throughput RFID system. *Secur. Commun. Netw.* **2016**, *9*, 2581–2591. [CrossRef]
8. Simon, P.M.G.; Riggert, E.F.; Trivelpiece, S.E. System and Method for Reading RFID Tags Across a Portal. U.S. Patent 9,519,811, 13 December 2016.
9. Wu, F.; Xu, L.; Kumari, S.; Li, X.; Das, A.K.; Shen, J. A lightweight and anonymous RFID tag authentication protocol with cloud assistance for e-healthcare applications. *J. Ambient Intell. Humanized Comput.* **2018**, *9*, 919–930. [CrossRef]
10. Sidorov, M.; Ong, M.T.; Vikneswaran, R.; Nakamura, J.; Ohmura, R.; Khor, J.H. Ultralightweight Mutual Authentication RFID Protocol for Blockchain Enabled Supply Chains. *IEEE Access* **2019**, *7*, 7273–7285. [CrossRef]
11. Noman, A.T.; Hossain, S.; Islam, S.; Islam, M.E.; Ahmed, N.; Chowdhury, M.M. Design and Implementation of Microcontroller Based Anti-Theft Vehicle Security System using GPS, GSM and RFID. In Proceedings of the 2018 4th International Conference on Electrical Engineering and Information & Communication Technology (iCEEiCT), Dhaka, Bangladesh, 13–15 September 2018; pp. 97–101.
12. Liao, Y.P.; Hsiao, C.M. A secure ECC-based RFID authentication scheme integrated with ID-verifier transfer protocol. *Ad Hoc Netw.* **2014**, *18*, 133–146. [CrossRef]
13. Kim, H. RFID mutual authentication protocol based on synchronized secret. *Int. J. Secur. Its Appl.* **2013**, *7*, 37–50.
14. Cha, J.R.; Kim, J.H. Novel anti-collision algorithms for fast object identification (RFID) system. In Proceedings of the 11th International Conference on Parallel and Distributed Systems, Washington, DC, USA, 20–22 July 2005; Volume 2; pp. 63–67.
15. El Beqqal, M.; Azizi, M. Classification of major security attacks against RFID systems. In Proceedings of the International Conference on Wireless Technologies, Embedded and Intelligent Systems (WITS), Fez, Morocco, 19–20 April 2017; pp. 1–6.
16. Tewari, A.; Gupta, B. Cryptanalysis of a novel ultra-lightweight mutual authentication protocol for IoT devices using RFID tags. *J. Supercomput.* **2017**, *73*, 1085–1102. [CrossRef]
17. Ayaz, U.; Haq, T.A.; Taimour, S.; Mansoor, K.; Mahmood, S. An Enhanced Biometric Based RFID Authentication Scheme Defending Against Illegitimate Access. In Proceedings of the 14th International Conference on Emerging Technologies (ICET), Islamabad, Pakistan, 21–22 November 2018; pp. 1–6.
18. Zhao, Z. A secure RFID authentication protocol for healthcare environments using elliptic curve cryptosystem. *J. Med. Syst.* **2014**, *38*, 46. [CrossRef]

19. Farash, M.S.; Nawaz, O.; Mahmood, K.; Chaudhry, S.A.; Khan, M.K. A provably secure RFID authentication protocol based on elliptic curve for healthcare environments. *J. Med. Syst.* **2016**, *40*, 165. [CrossRef] [PubMed]
20. Burmester, M.; De Medeiros, B.; Motta, R. Robust, anonymous RFID authentication with constant key-lookup. In Proceedings of the 2008 ACM symposium on Information, computer and communications security, Tokyo, Japan, 18–19 March 2008; pp. 283–291.
21. Cai, S.; Li, Y.; Li, T.; Deng, R.H. Attacks and improvements to an RIFD mutual authentication protocol and its extensions. In Proceedings of the second ACM conference on Wireless network security, Zurich, Switzerland, 16–19 March 2009; pp. 51–58.
22. Gaubatz, G.; Kaps, J.P.; Ozturk, E.; Sunar, B. State of the art in ultra-low power public key cryptography for wireless sensor networks. In Proceedings of the Third IEEE International Conference on Pervasive Computing and Communications Workshops, PerCom Workshops, Kauai Island, HI, USA, 8–12 March 2005; pp. 146–150.
23. Yang, J.; Park, J.; Lee, H.; Ren, K.; Kim, K. Mutual authentication protocol. In Proceedings of the Workshop on RFID and lightweight crypto, Graz, Austria, 14–15 July 2005.
24. Kang, S.Y.; Lee, I.Y. A Study on low-cost RFID system management with mutual authentication scheme in ubiquitous. In Proceedings of the Asia-Pacific Network Operations and Management Symposium, Sapporo, Japan, 10–12 October 2007; pp. 492–502.
25. Lee, L.S.; Fiedler, K.D.; Smith, J.S. Radio frequency identification (RFID) implementation in the service sector: A customer-facing diffusion model. *Int. J. Prod. Econ.* **2008**, *112*, 587–600. [CrossRef]
26. Qingling, C.; Yiju, Z.; Yonghua, W. A minimalist mutual authentication protocol for RFID system & BAN logic analysis. In Proceedings of the International Colloquium on Computing, Communication, Control, and Management, CCCM, Guangzhou, China, 3–4 August 2008, Volume 2; pp. 449–453.
27. Zhou, S.; Zhang, Z.; Luo, Z.; Wong, E.C. A lightweight anti-desynchronization RFID authentication protocol. *Inf. Syst. Front.* **2010**, *12*, 521–528. [CrossRef]
28. Piramuthu, S. RFID mutual authentication protocols. *Decis. Support Syst.* **2011**, *50*, 387–393. [CrossRef]
29. Safkhani, M.; Peris-Lopez, P.; Hernandez-Castro, J.C.; Bagheri, N. Cryptanalysis of the Cho et al. protocol: A hash-based RFID tag mutual authentication protocol. *J. Comput. Appl. Math.* **2014**, *259*, 571–577. [CrossRef]
30. Tan, C.C.; Sheng, B.; Li, Q. Secure and serverless RFID authentication and search protocols. *IEEE Trans. Wirel. Commun.* **2008**, *7*, 1400–1407. [CrossRef]
31. Cho, J.S.; Jeong, Y.S.; Park, S.O. Consideration on the brute-force attack cost and retrieval cost: A hash-based radio-frequency identification (RFID) tag mutual authentication protocol. *Comput. Math. Appl.* **2015**, *69*, 58–65. [CrossRef]
32. Naeem, M.; Chaudhry, S.A.; Mahmood, K.; Karuppiah, M.; Kumari, S. A scalable and secure RFID mutual authentication protocol using ECC for Internet of Things. *Int. J. Commun. Syst.* **2019**. [CrossRef]
33. Zhang, Z.; Qi, Q. An efficient RFID authentication protocol to enhance patient medication safety using elliptic curve cryptography. *J. Med. Syst.* **2014**, *38*, 47. [CrossRef]
34. Chaudhry, S.A.; Naqvi, H.; Farash, M.S.; Shon, T.; Sher, M. An improved and robust biometrics-based three factor authentication scheme for multiserver environments. *J. Supercomput.* **2018**, *74*, 3504–3520. [CrossRef]
35. Asgari, H.; Haines, S.; Rysavy, O. Identification of Threats and Security Risk Assessments for Recursive Internet Architecture. *IEEE Syst. J.* **2018**, *12*, 2437–2448. [CrossRef]
36. Abbasinezhad-Mood, D.; Nikooghadam, M. An Anonymous ECC-Based Self-Certified Key Distribution Scheme for the Smart Grid. *IEEE Trans. Ind. Electron.* **2018**, *65*, 7996–8004. [CrossRef]
37. Tan, H.; Ma, M.; Labiod, H.; Boudguiga, A.; Zhang, J.; Chong, P.H.J. A Secure and Authenticated Key Management Protocol (SA-KMP) for Vehicular Networks. *IEEE Trans. Veh. Technol.* **2016**, *65*, 9570–9584. [CrossRef]
38. Chaudhry, S.A.; Kim, I.L.; Rho, S.; Farash, M.S.; Shon, T. An improved anonymous authentication scheme for distributed mobile cloud computing services. *Cluster Comput.* **2019**, *22*, 1595–1609. [CrossRef]
39. Roy, S.; Chatterjee, S.; Das, A.K.; Chattopadhyay, S.; Kumari, S.; Jo, M. Chaotic Map-Based Anonymous User Authentication Scheme With User Biometrics and Fuzzy Extractor for Crowdsourcing Internet of Things. *IEEE Internet Things J.* **2018**, *5*, 2884–2895. [CrossRef]

40. Jiang, Q.; Zeadally, S.; Ma, J.; He, D. Lightweight three-factor authentication and key agreement protocol for internet-integrated wireless sensor networks. *IEEE Access* **2017**, *5*, 3376–3392. [CrossRef]
41. Mahmood, K.; Naqvi, H.; Alzahrani, B.A.; Mehmood, Z.; Irshad, A.; Chaudhry, S.A. An ameliorated two-factor anonymous key exchange authentication protocol for mobile client-server environment. *Int. J. Commun. Syst.* **2018**, *31*, e3814. [CrossRef]
42. Xu, Z.; Xu, C.; Chen, H.; Yang, F. A lightweight anonymous mutual authentication and key agreement scheme for WBAN. *Concurr. Comput. Pract. Exp.* **2019**, *31*, e5295. [CrossRef]
43. Xie, Q.; Hwang, L. Security enhancement of an anonymous roaming authentication scheme with two-factor security in smart city. *Neurocomputing* **2019**, *347*, 131–138. [CrossRef]
44. Kyntaja, T. *A logic of authentication by Burrows, Abadi and Needham*; Science Helsinki University of Technology: Tehran, Iran. Available online: http://www.tml.tkk.fi/Opinnot/Tik-110.501/1995/ban.html (accessed on 13 July 2019).
45. Blanchet, B. Modeling and verifying security protocols with the applied pi calculus and ProVerif. *Found. Trends Privacy Secur.* **2016**, *1*, 1–135. [CrossRef]
46. Lumini, A.; Nanni, L. An improved biohashing for human authentication. *Pattern Recognit.* **2007**, *40*, 1057–1065. [CrossRef]
47. Kilinc, H.H.; Yanik, T. A survey of SIP authentication and key agreement schemes. *IEEE Commun. Surv. Tutor.* **2014**, *16*, 1005–1023. [CrossRef]