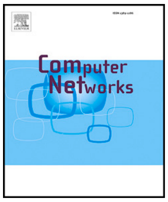
Computer Networks 220 (2023) 109469

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet



SAPWSN: A Secure Authentication Protocol for Wireless Sensor Networks

Foroozan Ghosairi Darbandeh [a], Masoumeh Safkhani [a,b,*]

[a] Computer Engineering Department, Shahid Rajaee Teacher Training University, Tehran, 16788-15811, Iran
[b] School of Computer Science, Institute for Research in Fundamental Sciences (IPM), P.O. Box 19395-5746, Tehran, Iran

ARTICLE INFO

Keywords:
WSN
EPC Class 1 Gen-2
Authentication
Privacy
Backward security contradiction attack
The Scyther Compromise-0.9.2 tool

ABSTRACT

With the advancement of RFID systems, there is a need for secure RFID authentication that can provide security against a variety of attacks, so designing a perfectly safe protocol has become a security challenge. The most remarkable security challenges may be information leakage, traceability, and tag impersonation. Several researchers have attempted to address this security demand by proposing ultra-lightweight solutions that use only very low-cost operations such as bit-wise operations. However, approximately all of the presented previous ultra-lightweight authentication schemes are vulnerable to a variety of attacks. For this purpose, Chiou and Chang recently proposed an EPC Class 1 Gen-2-based RFID authentication protocol and claimed it is resistant against replay attacks and also other known active and passive attacks. They also stated that their proposed protocol does not require features such as a secure channel, time parameters, or virtual IDs. In this paper, we will investigate the security of the Chiou and Chang authentication scheme and demonstrate that it is completely insecure. Specifically, we will present the security faults of this scheme. In addition, we will present an enhanced protocol called SAPWSN. The proposed protocol presents precise authentication and highly secure transfers. We demonstrate the proposed protocol's security in the formal and informal methods. In the formal method, we use the Compromise version of Scyther tool.

1. Introduction

In recent years, the Internet of Things (IoT) has gained prominence as a potential communication paradigm with a wide range of applications, including smart cities, smart homes, and intelligent transportation systems [1]. As the number of IoT devices grow, the associated IoT challenges are expected to grow as well. The IoT field faces different security challenges including data confidentiality, data authenticity, availability of the services, and privacy of the entities that are part of the IoT networks [2]. Radio frequency identification (RFID) is one of the fundamental technologies of IoT. In the internet of things system, RFID is a non-contact communication technology with automatic identification, which can effectively transmit and exchange data to achieve the purpose of secure authentication [3]. RFID uses radio waves for short-range communication to provide contactless and automatic object identification [4]. RFID system is made up of three parts: an RFID tag or smart label, an RFID reader, and a back-end database. Since RFID systems communicate via a public and insecure channel, various attacks such as replay attacks, man in the middle attacks, impersonation attacks, and so on constantly threaten the security of these systems.

RFID tags are now so common in our world that almost everyone comes into contact with them on a daily basis without even realizing it. RFID systems have a wide range of applications, particularly in health-care, agriculture, transportation, and industry. The advantages of RFID technology are the ability to use on objects with rough surfaces, read and write competencies without Line of Sight (LoS), and identify various RFID tags simultaneously. All these advantages make RFID an improved technology in comparison to the customary barcode system [5]. The increasing use of this technology, attention to the importance of the safety of sent data, and the increased likelihood of various attacks on it cause to have increased the significance of its security, so RFID security and privacy concerns have progressively become a stumbling block to the ongoing growth of the internet of things. The security issues can be handled by different cryptographic, authentication, authorization, and privacy solutions.

Designing secure and efficient authentication protocols has recently received many researchers' attention. Designed authentication protocols for RFID systems should take into account the limited capabilities of RFID tags, in addition to the ability to resist various attacks and threats. In RFID systems, mutual authentication is required for tags and readers, so only an authenticated reader can access the information of its corresponding tag.

[*] Corresponding author at: Computer Engineering Department, Shahid Rajaee Teacher Training University, Tehran, 16788-15811, Iran.
E-mail addresses: Sh.ghosairi@gmail.com (F.G. Darbandeh), Safkhani@sru.ac.ir (M. Safkhani).

https://doi.org/10.1016/j.comnet.2022.109469
Received 27 January 2022; Received in revised form 11 November 2022; Accepted 14 November 2022
Available online 18 November 2022

1.1. Related work

In recent years, the design of sensors has made significant progress that causes to consume less energy. As a result, RFID technology is now widely used in a variety of applications. Because RFID tags always have resource constraints and play different roles in different environments, thus the authentication mechanisms are required to be lightweight and flexible [6]. Security is a very important and significant issue that has been worked out thus far, and a number of security schemes on this important issue have been proposed. Unfortunately, many recent authentication schemes to secure IoT-based systems either were proved vulnerable to different attacks or were preyed on inefficiencies [7]. Some of the presented methods have flawed design as a result of an overemphasis on privacy and anonymity besides performance efficiency.

Molnar et al. in [8] developed a hash-based authentication scheme in 2004 that could use a method that divides computing costs evenly among nodes. Jules presented an authentication scheme in 2003 [9], which Rhee et al. later demonstrated it is vulnerable to attacks such as spoofing and replay attacks, and then presented an authentication scheme based on random numbers and one-way hash functions [10].

Juels in [11] suggested the "Yoking Proof" technique, which utilized hash functions and message authentication codes, but the proposed scheme was not secure against replay attacks and chosen-plaintext attacks [12].

Wong et al. presented a "hash-lock" concept [13], but this system also had numerous security flaws against various assaults [12].

Weis et al. in [14] created a security protocol based on hash functions and pseudo random number generators (PRNG). Tewari and Gupta [15] presented an ultra-lightweight mutual authentication mechanism for IoT devices utilizing RFID tags in 2016, but Wang et al. discovered that this system is still subject to secret disclosure attacks [16]. Cho et al. in [17] published a scheme that did not require a secure channel and instead depended on hash functions. However, because of the desynchronizing between tag and reader in this scheme, it was vulnerable to impersonation attacks [18].

In 2021, Shariq and Singh presented a revolutionary vector-space-based lightweight RFID authentication technique for an IoT context [19]. In the presented protocol, the authors have integrated the concept of vector space, basis, and linear mapping and have reduced the computational cost associated with the authentication process. This protocol has been proposed considering only the passive tags.

Mamun et al. presented a protocol in 2021 that is a multi-party authentication protocol and has provided mutual authentication not only in the tag-reader channel but also in the reader–server channel [20]. This protocol has used postquantum cryptographic systems such as Hopper–Blum (HB) authentication protocols based on Learning Parity with Noise (LPN) which is used to identify RFID tags. Authors claimed that their protocol could present the required security and privacy properties, as well as distinctive multi-party authentication properties, compared to other HB-family protocols in a better way.

Doss et al. published an authentication scheme for wireless/mobile RFID systems [21], however Chiou and Chang [22] demonstrated that the scheme was vulnerable to replay attacks and presented another protocol. The Chiou and Chang's protocol, which is based on EPC Class-1 Gen-2 requirements, does not need a secure channel.

In this paper, we show that the Chiou and Chang's protocol has several security flaws, and then we present SAPWSN, an amended version of the Chiou and Chang's scheme that addresses all of its security flaws.

1.2. Forward/Backward secrecy importance

Backward/Forward secrecy in security protocols is of great interest in any field, as will be discussed further below. Ren and Xu in [23] proposed a mutual authentication protocol for low-cost RFID systems and demonstrated that their protocol achieves backward secrecy without any assumptions and forward secrecy with an assumption.

In the field of VANET authentication protocols, Li et al. in [24] presented a lightweight authentication protocol for VANETs, which Shamshad et al. in [25] later demonstrated lacks the provision of backward/forward secrecy. They also proposed solutions to correct this flaw.

In the field of key distribution mechanisms, Hwang et al. in [26] proposed a group key exchange method, which was later proven by Lee et al. in [27] that Hwang et al.'s protocol does not preserve forward/backward secrecy.

Backward and forward security is so important that in [28], Kumar and Om stated that the security of symmetric key-based protocols is dependent on a long-term shared secret key, and this dependency introduces a threat to forward/backward secrecy. As a result, they proposed a conditional privacy-preserving and de-synchronization-resistant authentication protocol for VANETs that takes an efficient approach to forward/backward secrecy. Gebremichael et al. in [29] presented a Quantum-Safe Group Key Establishment Protocol from Lattice Trapdoors which also includes a mechanism for session key generation in a forward and backward secrecy preserving manner. The security feature of providing backward/forward security is so important that it has been the focus of researchers in the design of authentication protocols in the field of Global Mobility Network (GLOMONET). In [30], it is said that in authentication protocols for GLOMONET it is necessary to provide security issues such as resistance against all kind of impersonation attacks, backward/ forward secrecy, and stolen smart card attacks. Gope et al. in [31] presented a symmetric key based authentication protocol for GLOMONET that later Roy and Bhattacharya in [32] demonstrated that their protocol is vulnerable against privileged insider attack, offline password guessing attack, stolen smart card attack, session key compromised attack, unverified login phase and forward secrecy contradiction attack. Also in the field of Machine Type Communication Devices (MTCD) handover authentication protocols the backward/forward secrecy is also important. Yan and Ma in [33], stated the existing solutions for handover scenarios have several security problems in terms of the failure of forward secrecy and lack of mutual authentication. Therefore, they proposed an efficient authentication protocol for a group of Machine Type Communication Devices (MTCD)s in all handover scenarios.

1.3. Adversary model

Messages in our proposed strategy are sent over safe and insecure channels, giving the attacker the ability to attack via messages transferred via the insecure channels. The used adversary model's detailed assumptions are defined as follows:

- An attacker can eavesdrop, delete, intercept, and insert exchanged messages over an insecure channel.
- If an attacker intercepts a message, s/he has the ability to modify, delete, resend, and reroute it.
- The attacker can pose as a legitimate entity and receive messages from other protocol participants.
- In this paper, we assume that the public has access to a description of the protocol steps.

1.4. Main contributions

The session key revelation is probable and may causes too serious problems for the information which has been transferred based on that session key. However, it should not affect the transferred messages using other session keys. Due to the fact, that the RFID technology is used in most critical applications of IoT, Device to Device (D2D) communication, healthcare services, Wireless Body Area Networks (WBAN), Vehicular Ad hoc Networks (VANETs) and etc., so proposed RFID

Table 1
Notations used in this paper.

Notation	Description
t_X	The current time of X
T_{th_i}	The ith time threshold
n, m	The random numbers are generated by the reader and the tag, respectively
K_{RT} and K_{RT_B}	The reader's current and previous secret keys which are shared with the tag
K_{RS} and K_{RS_B}	The reader's current and previous secret keys which are shared with the server
K_{ST} and K_{ST_B}	The server's current and previous secret keys which are shared with the tag
$h(.)$	A one-way hash function
ID_x and ID_{x_B}	The current and previous identifier of x
$SID_x^{(S)}$	The pseudonym of x in the database of server
$SIDB_x^{(S)}$	The previous pseudonym of x in the database of server
$PRNG$	The pseudo random number generator
pt_R	The previous time of reader
$A \overset{?}{=} B$	Examining the equality or non-equality of two expressions A and B
s	The shared secret between tag, reader and server in the SAPWSN
r	The random number generated by the reader in the SAPWSN

authentication schemes inherently require anonymity and complete security. The presented protocols must guarantee that the security of past and future session keys are protected when the long term key is exposed, in other words, providing Backward Security in RFID's authentication protocols is vital. This concept has been considered on recent related works also, e.g. [34–37,37,38]. However, in this paper we show that the Chiou and Chang's protocol does not provide this aspect of security, so it has a big security hole. We also tried to present a more secure scheme that almost provides complete security to use in critical usages and also proved our claim by the Compromise version of Scyther tool, an automatic security protocols verification tool. Furthermore, in this paper, we proved the Chiou and Chang protocol has other security weaknesses and we also showed through the Compromise version of Scyther tool, the Chiou and Chang's protocol does not provide the security claims of $Weakagree$, $Niagree$, $Alive$, and $Secrecy$ which will be defined later. So, the contributions of this paper are summarized into several folds:

- Security analysis of the Chiou and Chang authentication protocol;
- Amending the security flaws of the Chiou and Chang authentication protocol that led to the creation of SAPWSN, a new secure authentication scheme.
- Proving the proposed protocol's security (i.e. SAPWSN) both in the formal way (using the Compromise version of Scyther tool) and in an informal way.

1.5. Paper organization

Section 2 of this paper reviews Chiou and Chang's EPC Class 1 Gen-2 authentication scheme. Section 3 discusses the security analysis of this protocol and identifies its security flaws. In Section 4, we introduce SAPWSN, an amended version of the protocol. Section 5 evaluates the SAPWSN's security and finally Section 7 concludes the paper with concluding remarks and suggestions for future works.

2. Review of Chiou and Chang's authentication protocol

Chiou and Chang in [22] developed an authentication scheme based on virtual identities and temporal parameters that did not require using a secure channel in 2018. They stated that their scheme presented privacy and authentication for mobile RFID systems.

Table 1 shows the used notations in the Chiou and Chang authentication scheme and the scheme's steps are illustrated in Fig. 1. This scheme runs as described below:

1. **Initialization Phase:** In this phase, the server S generates the system's initial parameters as follows:

   (a) The server:

   i. generates two random numbers p and q in such a way the size of them be 512 bits or more;
   ii. calculates $n = p.q$ and saves n into the memory of the tag and reader.

   (b) Each tag selects a unique identifier ID_T and the reader selects ID_R.
   (c) Two 1024-bit random numbers K_{ST} and K_{SR} are produced as common keys where K_{ST} is a shared key between the server and the tag and K_{SR} is a shared key between the server and the reader.
   (d) SID_T and SID_R are respectively unique identifiers of the tag and reader.
   (e) The server computes following values:
   $SIDB_T^{(S)} \leftarrow SID_T^{(S)}$
   $SIDB_T^{(R)} \leftarrow SID_T^{(R)}$
   $K_{ST_B} \leftarrow K_{ST}$
   $K_{SR_B} \leftarrow K_{SR}$
   $pt_R \leftarrow 0$
   and also stores the values $\{p, q, n, ID_T, ID_R, K_{ST_B}, K_{ST}, K_{SR_B}, K_{SR}, SIDB_T^{(S)}, SID_T^{(S)}, SIDB_R^{(S)}, SID_R^{(S)}, pt_R\}$, in its memory.
   (f) $\{ID_T, K_{ST}, SID_T, n\}$ are stored in the tag and $\{ID_R, K_{SR}, SID_R, n\}$, are stored in the reader.

2. **Verification Phase**

This step of the protocol, runs as follows:

1. The reader extracts t_R and sends t_R to the tag.
2. When the tag receives t_R, it:

   - determines whether $t_R > 0$ is or not, if so, calculates $x = K_{ST} \oplus t_R$ and $x' = (x)^2$ mod n;
   - and sends $\{SID_T, x'\}$ to the reader.

3. Once the message is delivered to the reader, it:

   - calculates $y = K_{SR} \oplus t_R$ and $y' = (y)^2$ mod n;
   - and sends $\{SID_T, SID_R, x', y', t_R\}$ to the server.

4. When the message is received by the server, it:

   - checks whether $T_{th_1} < t_S - t_R \leq T_{th_2}$ is or not. If it is true, then searches its database for $SID_T^{(S)}$ and $SID_R^{(S)}$ or $SIDB_T^{(S)}$ and $SIDB_R^{(S)}$;
   - extracts pt_R, K_{ST} and K_{SR} or K_{ST_B} and K_{SR_B}.
   - checks whether $t_R \neq pt_R$ is valid or not. If it is true, solves the equation $y' = (y)^2$ mod n by using Chinese Reminder Theorem (CRT) with p, q and obtains y_j which equals with $K_{SR} \bigoplus t_R$. If y_j is found, the reader is verified.
   - Likewise, finds the value of x_i equals to $K_{ST} \bigoplus t_R$ by solving the equation $x' = (x)^2$ mod n, if it is so, then the tag is authenticated.
   - calculates $ACK_1 = PRNG(x) \bigoplus PRNG(y+1)$ and $ACK_2 = PRNG(y)$ and sends $\{ACK_1, ACK_2\}$ to the reader.
   - If $SID_R = SID_R^{(S)}$ then lets $SIDB_R^{(S)} \leftarrow SID_R$, $SID_R^{(S)} \leftarrow PRNG(y \bigoplus SIDB_R)$, $K_{SR_B} \leftarrow K_{SR}$ and $K_{SR} \leftarrow (K_{SR})^2$ mod n.

Server $(q,p,ID_R,ID_T,K_{SR},K_{ST},$ $K_{SR_B}, K_{ST_B}, SID_R^{(s)},SID_T^{(s)},$ $, SIDB_T^{(s)}, SIDB_R^{(s)}, pt_R)$		**Reader** (ID_R,K_{SR},n,SID_R)		**Tag** (ID_T,K_{ST},n,SID_T)
		Extracts t_R	$\xrightarrow{t_R}$	
				Checks $t_R > 0$ if valid $x = K_{ST} \oplus t_R$ $x' = x^2 \ mod \ n$
			$\xleftarrow[,x']{SID_T}$	
		$y = K_{SR} \oplus t_R$ $y' = y^2 \ mod \ n$		
	$\xleftarrow[,y',x',t_R]{SID_R,SID_T}$			
Extracts t_s checks $T_{th_1} < t_s - t_r \le T_{th_2}$ searches for $SID_T^{(s)}, SID_R^{(s)}$ extracts pt_R, K_{ST}, K_{SR} or searches for $SIDB_T^{(s)},SIDB_R^{(s)}$ extracts pt_R, K_{ST_B}, K_{SR_B} checks $pt_R \neq t_R$, if valid, solves $y' = y^2 \ mod \ n$ using CRT with p,q and gets y_j checks $y_j \overset{?}{=} K_{SR} \oplus t_R$ if valid, user is legitimate. solves $x' = x^2 \ mod \ n$ using CRT with p,q and gets x_i checks $x_i \overset{?}{=} K_{ST} \oplus t_R$ if valid, tag is legitimate. $ACK_1 = PRNG(x)\oplus$ $PRNG(y+1)$ $ACK_2 = PRNG(y)$				
	$\xrightarrow[ACK_2]{ACK_1,}$			
		Checks $ACK_2 \overset{?}{=}$ $PRNG(y)$, if ok server is legitimate. $ACK = ACK_1\oplus$ $APRNG(y+1)$		
			$\xrightarrow{ACK}$	
				Checks $ACK \overset{?}{=}$ $PRNG(x)$, if valid server and reader are legitimate.
checks $SID_R \overset{?}{=} SID_R^S$, if valid $SIDB_R^{(s)} = SID_R$ $SID_R = PRNG(y \oplus SIDB_R)$ $K_{SR_B} = K_{SR}, K_{SR} = K_{SR}^2 \ mod \ n$ $SID_T^{(S)} \overset{?}{=} SID_T$, if valid $SIDB_T^{(S)} = SID_T$ $SID_T^{(S)} = PRNG(y \oplus SIDB_T)$ $K_{ST_B} = K_{ST}$, $K_{ST} = K_{ST}^2 \ mod \ n$ $pt_R = t_R$		$K_{SR} = K_{SR}^2 \ mod \ n$ $SID_R =$ $PRNG(y \oplus SID_R)$		$K_{ST} = K_{ST}^2 \ mod \ n$ $SID_T =$ $PRNG(x \oplus SID_T)$

Fig. 1. Chiou and Chang protocol for WSN [22].

- If $SID_T = SID_T^{(S)}$ then lets $K_{ST_B} \leftarrow K_{ST}$, $K_{ST} \leftarrow (K_{ST})^2 \ mod \ n$, $SIDB_T^{(S)} \leftarrow SID_T$, $SID_T^{(S)} \leftarrow PRNG(y \bigoplus SIDB_T)$ and $pt_R \leftarrow t_R$.

5. When the message is received by the reader, it:

   - checks whether $ACK_2 \overset{?}{=} PRNG(y)$ is or not, if it is true, the server is authenticated.
   - calculates $ACK = ACK_1 \bigoplus PRNG(y{+}1)$ and sends $\{ACK\}$ to the tag and lets $K_{SR} \leftarrow (K_{SR})^2 \ mod \ n$ and $SID_R \leftarrow PRNG(y \bigoplus SID_R)$.

6. Upon receiving the message, the tag:

   - checks whether $ACK \overset{?}{=} PRNG(x)$, if it is true, it authenticates both the reader and the server, then lets $K_{ST} \leftarrow (K_{ST})^2 \ mod \ n$ and $SID_T \leftarrow PRNG(x \bigoplus SID_T)$.

3. Security analysis of Chiou and Chang scheme

In this section, we will demonstrate that the Chiou and Chang scheme [22] does not ensure backward security. We evaluated the protocol's security using both formal and informal methods.

3.1. Informal method

3.1.1. Backward security

A protocol's backward security arises when an attacker who knows the prior transaction information unable to acquire critical information about future transactions.

This rule is incorrect in the Chiou and Chang scheme, because if the attacker reveals the present session secret key, s/he could easily compute the secret keys of the future sessions. In general, if the adversary can gain the secret key of a session, s/he may also discover the secret key of subsequent sessions, as follows:

1. **Learning Phase:** During this phase of the attack, the adversary eavesdrops on one run of protocol and records the protocol messages, including: $\{SID_T, SID_R, x', y', t_R\}$ and $\{ACK_1 = PRNG(x) \bigoplus PRNG(y+1), ACK_2 = PRNG(y)\}$.
2. **Secret Disclosure Phase:** It is assumed, that the adversary has the current session shared keys throughout this phase of the attack. Furthermore, because the attacker knows the value of t_R and these facts $x = K_{ST} \bigoplus t_R$ and $y = K_{SR} \bigoplus t_R$, as a result, the attacker can easily acquire the values of x and y.
   However, if the attacker knows the values of y', y, x', and x, as well as the information that $y' = (y)^2 \ mod \ n$ and $x' = (x)^2 \ mod \ n$, the attacker can calculate the value of n.
   Since the shared keys have been updated to $K_{ST} \leftarrow (K_{ST})^2 \ mod \ n$ and $K_{SR} \leftarrow (K_{SR})^2 \ mod \ n$, and moreover the attacker knows the value of the current session keys as well as the value of n, the attacker may simply obtain the new values of the shared keys. As a result, the protocol lacks backward secrecy.

3.2. Formal method

Cremers [39] created Scyther, a formal security protocol verification tool, based on the perfect cryptography assumption. The Scyther tool is one of the most powerful tools for security protocol testing, falsification, and analysis, on the assumption an attacker cannot decrypt any communication without the symmetric encryption key. This tool can determine the security requirements and vulnerabilities of a protocol. The security goals in each application are defined as three important principles of confidentiality, integrity and availability. In the Scyther, the designers have used these three important principles in the form of two features, $Secrecy$ and $Authentication$, with the following definitions:

$Secrecy$: $Secrecy$ states that certain confidential and secret information will not be revealed to the adversary, even if we transmit this data through an insecure channel. Different forms of $Secrecy$ can be defined with subtle differences. For an example, a security claim written as $claim(R, Secret, S)$, which executes in the role R, taking the expression S as the secret parameter. This claim states whether for all executions of the protocol role, the statement S remains secret, i.e. it remains unknown to the adversary or not.

$Authentication$: The most studied security feature in the field of security protocol analysis is $Authentication$. However, contrary to the claim of $Secrecy$, there is no general consensus on the meaning of $Authentication$. In fact, as shown by Lowe in [40], there is a hierarchy of authentication features. $Authentication$ focuses on the fact that the implementation of a protocol role actually guarantees that there is at least one communication partner in the network. In most cases, we want to establish a stronger objective, for example, that the intended partner is aware of our communication and that a protocol is being implemented and that messages have been exchanged as expected. These hierarchies are described under the properties of $Aliveness$, $Synchronization$ and $Agreement$ in the Scyther tool which in detail are explained in Table 2. As can be seen in Table 2, different types of security features that actually provide the same three main goals of confidentiality, availability and integrity are reviewed in the Scyther tool.

The Chiou and Chang scheme is specified using the Security Protocol Description Language (SPDL). Fig. 2 also depicts the Compromise version of Scyther tool's attacks for Chiou and Chang's protocol.

As shown in Fig. 2, the Chiou and Chang protocol does not satisfy the security claims of $Weakagree$, $Niagree$, $Alive$, and $Secrecy$. Table 2 gives a detailed description of these security claims.

4. Improved protocol: SAPWSN

In this section, we will propose SAPWSN to improve the security of the Chiou and Chang's protocol. The protocol 's procedure is divided into three phases: Initialization, Registration, and Verification.

4.1. Initialization phase

The server:

1. Produces ID_T and ID_R, two unique identifiers for the legal tag and valid reader, respectively.
2. Produces three distinct 128-bit secret keys between the server and the tag, namely K_{ST}, the server and the reader, namely K_{RS}, and the reader and the tag, namely K_{RT}.
3. The data that is maintained in the server, tag, and reader are $\{ID_{T_B}, ID_T, ID_R, ID_{R_B}, K_{ST}, K_{ST_B}, K_{RS}, K_{RS_B}, s\}$, $\{ID_T, K_{ST}, K_{RT}, ID_R, s\}$ and $\{ID_R, K_{RS}, K_{RT}, K_{RS_B}, ID_T, s\}$ respectively.
4. Sets the timer of the reader and the server to synchronize them.

4.2. Registration phase

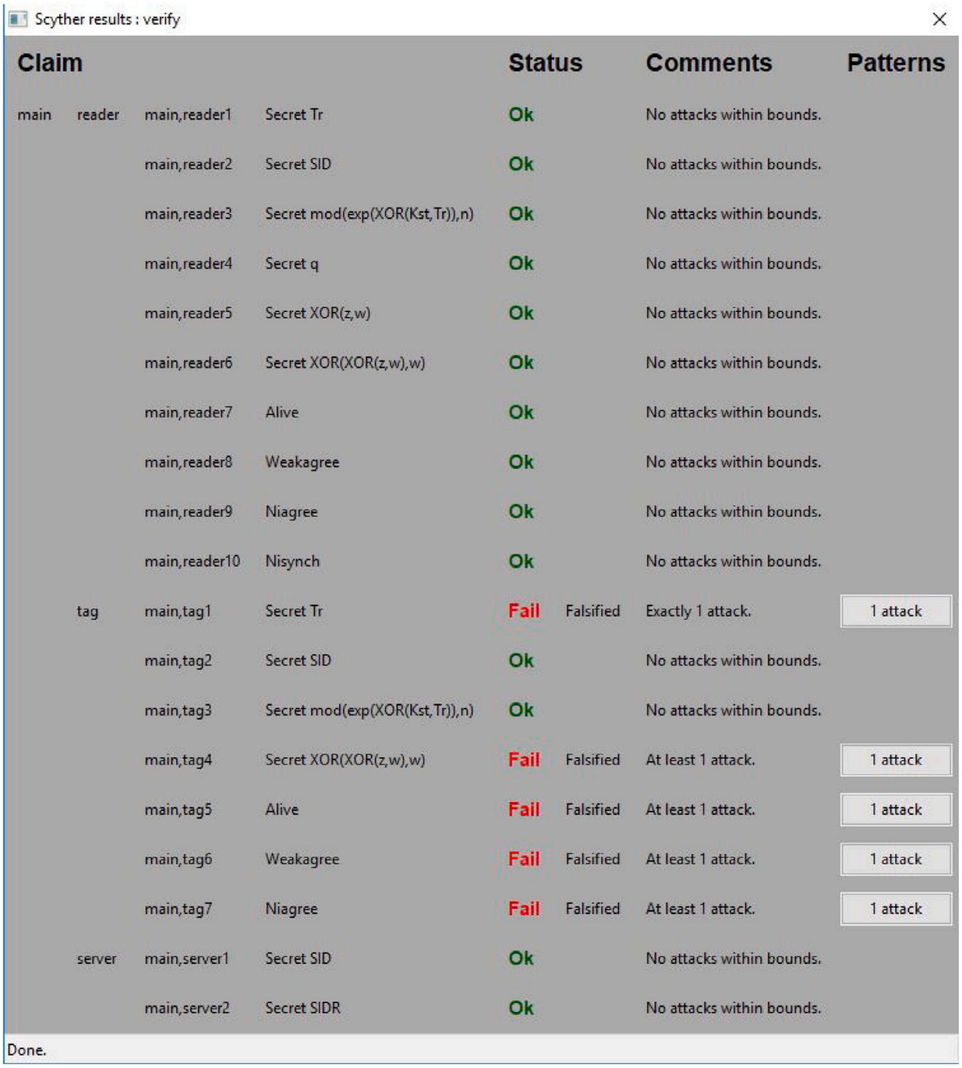
During this phase, entities attempt to generate and preserve a safe value that is employed to produce the following secret keys. The secure channel transmits data depending on the processes outlined below (see Fig. 3):

1. The reader chooses a random number r and calculates $s = h(r \parallel ID_R)$. The reader transmits ID_R and s to the tag through a secure channel.
2. Once the message is received by the tag, it saves the values s and ID_R before sending ID_T to the reader.
3. The reader stores ID_T and sends s, ID_R, and ID_T to the server.
4. The server stores s, ID_R, and ID_T in its memory.

4.3. Verification phase

The steps outlined in Fig. 4 are performed to accomplish mutual authentication in the proposed protocol.

1. The reader begins this step as follows:
   - extracts current time as t_R;
   - generates a random number n;
   - computes $x = n \bigoplus K_{RT}$;
   - sends message $\{t_R, n, h(x \parallel t_R)\}$ to the tag;
2. When the message is received by the tag, it:
   - first checks $t_R > 0$, if valid, computes $x' = n \bigoplus K_{RT}$ and $h(x' \parallel t_R)$, then checks whether $h(x' \parallel t_R) \overset{?}{=} h(x \parallel t_R)$ is valid or not, if it is valid, generates another random number m and computes $y = h(m \| K_{RT} \| n)$ and $z = h(n \| m \| K_{ST})$.
   - sends the message $\{m, y, z\}$ to the reader.
3. When the message is delivered to the reader, it:
   - computes $y' = h(m \| K_{RT} \| n)$;
   - checks whether $y' \overset{?}{=} y$ is valid or not, if it is valid, computes $w = h(n \parallel m \parallel t_R \parallel K_{SR})$;
   - and sends message $\{z, m, n, w, t_R\}$ to the server.
4. After the server receives the message, it:
   - extracts current time t_S;
   - checks whether $t_S - t_R \leq T_{th}$ is valid or not, if it is valid, it computes $z' = h(n \| m \| K_{ST})$.



Claim				Status		Comments	Patterns
main	reader	main,reader1	Secret Tr	Ok		No attacks within bounds.	
		main,reader2	Secret SID	Ok		No attacks within bounds.	
		main,reader3	Secret mod(exp(XOR(Kst,Tr)),n)	Ok		No attacks within bounds.	
		main,reader4	Secret q	Ok		No attacks within bounds.	
		main,reader5	Secret XOR(z,w)	Ok		No attacks within bounds.	
		main,reader6	Secret XOR(XOR(z,w),w)	Ok		No attacks within bounds.	
		main,reader7	Alive	Ok		No attacks within bounds.	
		main,reader8	Weakagree	Ok		No attacks within bounds.	
		main,reader9	Niagree	Ok		No attacks within bounds.	
		main,reader10	Nisynch	Ok		No attacks within bounds.	
	tag	main,tag1	Secret Tr	Fail	Falsified	Exactly 1 attack.	1 attack
		main,tag2	Secret SID	Ok		No attacks within bounds.	
		main,tag3	Secret mod(exp(XOR(Kst,Tr)),n)	Ok		No attacks within bounds.	
		main,tag4	Secret XOR(XOR(z,w),w)	Fail	Falsified	At least 1 attack.	1 attack
		main,tag5	Alive	Fail	Falsified	At least 1 attack.	1 attack
		main,tag6	Weakagree	Fail	Falsified	At least 1 attack.	1 attack
		main,tag7	Niagree	Fail	Falsified	At least 1 attack.	1 attack
	server	main,server1	Secret SID	Ok		No attacks within bounds.	
		main,server2	Secret SIDR	Ok		No attacks within bounds.	

Done.

Fig. 2. Security verification results of the Chiou and Chang protocol through the Scyther tool.

Server		Reader		Tag
		Generates a random r $s = h(r \Vert ID_R)$		
			$\xrightarrow[SecureChannel]{ID_R,s}$	Stores s, ID_R
			$\xleftarrow[SecureChannel]{ID_T}$	
		Stores ID_T		
	$\xleftarrow[SecureChannel]{s,ID_T,ID_R}$			
Stores s, ID_T, ID_R				

Fig. 3. The Registration phase of SAPWSN.

Table 2
Security claims of the Scyther tool.

Claims	Description
Secrecy	implies that no specific confidential information is exposed to the attacker, even if this data is transferred over an insecure channel.
Authentication	Authentication is often concerned with ensuring that the network has at least one communication partner by completing a protocol role.
Aliveness	According to the definition, when an agent executes a role specification up to the claim event and believes he is communicating with a trusted agent, the intended communication partner has actually executed an event.
Synchronization	It necessitates a higher level of verification. *Synchronization* necessitates that the communication partner sends all incoming messages and receives all sent messages. This requirement is consistent with the requirement that the actual message exchange occur exactly as specified in the protocol description. Synchronization criteria ensure that the protocol behaves in accordance with predefined explanations even when an adversary is present.
Agreement	Another authentication criterion that focuses on the agreement reached between the protocol's parties is *Agreement*. The *Agreement* criterion is based on the idea that after running the protocol, the parties agree on the values of certain variables. *Agreement* is defined as a criterion that requires the message's content to follow the message sent in accordance with the protocol's rules. As a result, the content of the variables will be accurate as defined by the protocol after the protocol is executed. It is not possible to change the message's content from this perspective. If there is a change in the message, the recipient will notice it.
Weakagree	As a result of this criterion, the communication partners must ensure that they are communicating with each other in order to prevent one of them from being fabricated by the adversary.
Nisynch	Non-injective synchronization, as defined in [40], means that receiving and sending events are carried out by roles, in the order specified, and with the primary content in question.
Niagree	Non-injective agreement on messages as defined in [40] means that the sender and receiver agree on the secret values exchanged, and the results of the analysis justify the validity of this claim.
Empty	This claim has not been verified and is simply ignored. This claim holds true only if the Scyther tool is used as a backend for other security verification tools.
Reachable	When this claim is generated, the Scyther tool checks to see if it can be materialized at all. This claim is true if there is a way for it to occur. This claim is actually inserted when the Scyther tool check mode is used and can be useful for checking any obvious errors in the protocol specification.

- checks whether $z' \overset{?}{=} z$ is valid or not, if it is, validates the tag, then computes $w' = h(n \parallel m \parallel t_R \parallel K_{RS})$.
- determines whether $w' \overset{?}{=} w$ is valid or not, if it is, authenticates the reader.
- computes $ACK_1 = h(t_S \parallel m \parallel K_{ST} \parallel ID_T) \bigoplus h(n \parallel K_{RS})$ and $ACK_2 = h(t_S \parallel n \parallel K_{RS} \parallel ID_R \parallel ACK_1 \parallel ID_T)$;
- Lets $ID_{R_B} \leftarrow ID_R$ and $ID_R \leftarrow h(ID_R \parallel n \parallel K_{RS})$ $K_{RS_B} \leftarrow K_{RS}$ and $K_{RS} \leftarrow h(K_{RS} \parallel ID_R \parallel n \parallel s)$.
- Lets $K_{ST_B} \leftarrow K_{ST}$, $K_{ST} \leftarrow h(K_{ST} \parallel ID_T \parallel m \parallel s)$, $ID_{T_B} \leftarrow ID_T$, $ID_T \leftarrow h(ID_T \parallel m \parallel K_{ST})$.
- and sends message $\{ACK_1, ACK_2, t_S\}$ to the reader.

5. When the reader gets the server's message, it:

    - checks whether $t_S - t_R \leq T_{th}$ is true or not, if it is true, computes $ACK_2' = h(t_S \parallel n \parallel K_{RS} \parallel ID_R \parallel ACK_1 \parallel ID_T)$ and checks whether $ACK_2' \overset{?}{=} ACK_2$ is true or not, if it is true, authenticates the server.
    - computes $ACK = ACK_1 \bigoplus h(n \parallel K_{RS})$;
    - sends message $\{ACK, t_S\}$ to the tag.
    - and updates its secret records ID_R as $h(ID_R \parallel n \parallel K_{RS})$, K_{RS} as $h(K_{RS} \parallel ID_R \parallel n \parallel s)$, K_{RT_B} as K_{RT} and K_{RT} as $h(K_{RT} \parallel n \parallel m \parallel ID_T \parallel s)$.

6. Once the tag receives ACK, it:

    - checks whether $t_S - t_R \leq T_{th}$ is valid or not, if it is valid, computes $ACK' = h(t_S \parallel m \parallel K_{ST} \parallel ID_T)$ and checks whether $ACK' \overset{?}{=} ACK$ is true or not, if it is true, authenticates the server and the reader.
    - sets the timer of the reader and the server to hold them in sync.
    - and updates its secret records ID_T as $h(ID_T \parallel m \parallel K_{ST})$, K_{RT} as $h(K_{RT} \parallel n \parallel m \parallel ID_T \parallel s)$ and K_{ST} as $h(K_{ST} \parallel ID_T \parallel m \parallel s)$.

5. Security analysis of the SAPWSN protocol

To demonstrate that our proposed protocol, i.e. SAPWSN, has the needed security features, we first conduct an informal validation and then a formal verification using the Compromise version of Scyther tool.

5.1. Informal verification

- **Resistance against Secret Disclosure Attacks**
  Given that we utilized the hash function in the structure of the SAPWSN protocol's messages, the attacker cannot obtain secret values such as K_{RS}, K_{RT} and K_{ST}, and none of the reader's and tag's other secret values. As a result, the SAPWSN protocol is immune from secret disclosure attacks.
- **Resistance against Impersonation Attacks**
  In the SAPWSN protocol, a wicked user cannot easily masquerade as another authorized user. Because the attacker needs to know the K_{RT}, K_{RS} and K_{ST}, and since the hash function has been

Server $(K_{ST}, K_{RS}, ID_R, ID_T, K_{ST_B}, K_{RS_B}, ID_{R_B}, ID_{T_B}, s)$		**Reader** $(K_{RT}, ID_R, K_{RS}, K_{RT_B}, ID_T, s)$		**Tag** $(ID_T, K_{RT}, K_{ST}, ID_R, s)$
		Extracts t_R, generates n, $x = n \oplus K_{RT}$		
			$\xrightarrow[h(x\|t_R)]{t_R,n,}$	
				Checks $t_R > 0$, if ok $x' = K_{RT} \oplus n$ checks $h(x\|t_R) \overset{?}{=} h(x'\|t_R)$ if ok, generates m computes $y = h(m\|K_{RT}\|n)$ and $z = h(n\|m\|K_{ST})$
			$\xleftarrow{m,y,z}$	
		$y' = h(m\|K_{RT}\|n)$, checks $y \overset{?}{=} y'$, if ok, $w = h(n\|m\|t_R\|K_{SR})$		
	$\xleftarrow[m,n,z]{w,t_R}$			
Extracts t_S checks $t_S - t_R \leq T_{th}$, $z' = h(n\|m\|K_{ST})$ checks $z \overset{?}{=} z'$, if ok, tag is legitimate, $w' = h(n\|m\|t_R\|K_{RS})$, checks $w \overset{?}{=} w'$ if ok, user is legitimate, $ACK_1 = h(t_S\|m\|K_{ST}\|ID_T) \oplus h(n\|K_{RS})$ $ACK_2 = h(t_S\|n\|K_{RS}\|ID_R \|ACK_1\|ID_T)$				
	$\xrightarrow[ACK_2,t_S]{ACK_1}$			
		Checks $t_S - t_R \leq T_{th}$ $ACK_2' = h(t_S\|n\|K_{RS}\|ID_R \|ACK_1\|ID_T)$ checks $ACK_2 \overset{?}{=} ACK_2'$ to authenticate server computes $ACK = ACK_1 \oplus h(n\|K_{RT})$		
			$\xrightarrow{ACK,t_S}$	
				Verifies $t_S - t_R \leq T_{th}$ computes $ACK' = h(t_S\|m\|K_{ST}\|ID_T)$ checks $ACK \overset{?}{=} ACK'$ to authenticate server and reader.
$K_{ST_B} = K_{ST}$ $K_{ST} = h(K_{ST}\|ID_T\|m\|s)$, $K_{RS_B} = K_{RS}$, $K_{RS} = h(K_{RS}\|ID_R\|n\|s)$, $ID_{T_B} = ID_T$, $ID_T = h(ID_T\|m\|K_{ST})$, $ID_{R_B} = ID_R$, $ID_R = h(ID_R\|n\|K_{SR})$		$K_{RS} = h(K_{RS}\|ID_R\|n\|s)$, $K_{RT_B} = K_{RT}$, $K_{RT} = h(K_{RT}\|n\|m\|ID_T\|s)$, $ID_R = h(ID_R\|n\|K_{SR})$		$K_{ST} = h(K_{ST}\|ID_T\|m\|s)$, $K_{RT} = h(K_{RT}\|n\|m\|ID_T\|s)$, $ID_T = h(ID_T\|m\|K_{ST})$

Fig. 4. The process of SAPWSN: the proposed protocol to be employed in WSN.

utilized in the structure of the protocol's messages, we can claim it is impossible to obtain these values in the protocol; therefore, the SAPWSN protocol can able to be safe against all kinds of impersonation threats. More precisely, to impersonate the tag, the adversary should compute the valid $y = h(m \parallel K_{RT} \parallel n)$ and $z = h(n \parallel m \parallel K_{ST})$ while n is a fresh challenge sent by the reader and K_{ST} and K_{RT} are secret parameters which is not feasible. To impersonate the reader to the server, the adversary should compute the valid $w = h(n \parallel m \parallel t_R \parallel K_{SR})$ while t_R is a fresh timestamp and out of the adversary's control and K_{SR} is a secret parameter which is not feasible. On the other hand, to impersonate the server, the adversary should successfully compute $ACK_1 = h(t_S \parallel m \parallel K_{ST} \parallel ID_T) \oplus h(n \parallel K_{RS})$ and $ACK_2 = h(t_S \parallel n \parallel K_{RS} \parallel ID_R)$ which is not feasible. Finally, to impersonate the reader to the tag, the adversary should send a valid $ACK = ACK_1 \oplus h(n \parallel K_{RT})$ to the tag that is a factor of the secret K_{RT} and other session dependent parameters such as m. All in all, the adversary cannot impersonate any entity of the protocol.

- **Resistance against Replay Attacks**
  The timestamp system and verifying transmission delay time are utilized in the SAPWSN protocol. Moreover, the tag and reader are involved in the randomization of protocol messages. As a result, the attacker cannot transmit to an arbitrary entity the sent prior messages in the protocol, then claim that these messages have been sent by a legit entity. As a result, the SAPWSN protocol is resistant to replay attacks.
- **Resistance against Desynchronization Attack**
  In a desynchronization attack, the attacker sends bogus messages to one or both sides of an active connection in a wireless sensor network in order to mismatch the secret values. As a consequence, the parties will be unable to identify each other in future interactions.

  The exchanged values in the messages in the SAPWSN protocol are protected using the hash function, so the attacker cannot simply modify these values without the communication sides being realized. If the values are changed, the message receiver will discover this by comparing the received amounts.
- **Backward Security**
  Backward security of a protocol is achieved when an attacker has a key of a protocol session but is unable to obtain the next session key. As mentioned above, in our proposed protocol, the key to each session is created using secret values such as s. Therefore, revealing a session key does not imperil other keys in subsequent sessions. Precisely, the keys in the SAPWSN protocol are updated as $K_{RS} = h(K_{RS} \parallel ID_R \parallel n \parallel s)$, $K_{RT} = h(K_{RT} \parallel n \parallel m \parallel ID_T \parallel s)$ and $K_{ST} = h(K_{ST} \parallel ID_T \parallel m \parallel s)$, which are also dependent on the updated parties identifiers and secret values. As a result, in the proposed protocol, divulging the current session key will not reveal the key of the next session.
- **Forward Security**
  In this case, it is anticipated that if an attacker obtains the key of a protocol session in any way, s/he will be unable to determine the key of previous sessions using that key. The keys of the protocol

sessions, i.e. $K_{RS} = h(K_{RS} \parallel ID_R \parallel n \parallel s)$, $K_{RT} = h(K_{RT} \parallel n \parallel m \parallel ID_T \parallel s)$ and $K_{ST} = h(K_{ST} \parallel ID_T \parallel m \parallel s)$, are made through hash functions with secret values in the SAPWSN protocol, so that the disclosure of one key does not endanger the key of the previous sessions.

- **Robustness**
  In seminal works, Anderson and Needham [41] and Abadi and Needham [42] provided robustness principles for designing cryptographic protocols. Following the first principle in [42], every message should say exactly what it means: the message's interpretation should be based solely on its content. It should be possible to write a simple English sentence describing the content; however, if a suitable formalism is available, that is also acceptable. It is clear that all the transferred messages in the SAPWSN fulfill this principle because we have clearly defined each part of the transferred messages. The third principle contends that if the identity of a principal is critical to the meaning of a message, the principal's name should be mentioned explicitly in the message. However, implicit ID compromises the traceability property and hence we included the IDs implicitly to avoid such attack, i.e. $ACK_1 = h(t_S \parallel m \parallel K_{ST} \parallel ID_T) \oplus h(n \parallel K_{RS})$ and $ACK_2 = h(t_S \parallel n \parallel K_{RS} \parallel ID_R \parallel ACK_1 \parallel ID_T)$. Furthermore, the seventh principle states that in order to a predictable quantity to be effective, it must be protected so that an intruder cannot simulate a challenge and then replay a response. The only predictable value in the SAPWSN is the timestamp, which is protected by a keyed hash. The second principle of [41] states that the same key should not be used for two different purposes (such as signing and decryption), and that different runs of the same protocol should be distinguished from one another. In the SAPWSN, we do not use encryption and also different sessions are distinguished by different timestamps and the contributed nonces by the protocol's entities. Hence, the SAPWSN meets this principle also.
- **No Key Control**
  The computed session keys, in each session of SAPWSN, are $K_{RS} = h(K_{RS} \parallel ID_R \parallel n \parallel s)$, $K_{ST} = h(K_{ST} \parallel ID_T \parallel m \parallel s)$ and $K_{RT} = h(K_{RT} \parallel n \parallel m \parallel ID_T \parallel s)$. It is clear each session key contains an equal share of both participants' session parameters as well as their private keys. As a result, none of the participants has any control over the formation of the session key, and the SAPWSN satisfies the No Key Control (NKC) property.

5.2. Formal verification

In this section, we will use the Compromise version of Scyther tool to verify the security of the SAPWSN protocol. The Scyther tool has two versions, Standard and Compromise, the difference between these two versions is that in the Standard model, the adversary model is the Dolev–Yao adversary model [43], and in the Compromise version, in addition to the Dolev–Yao adversary model, is also possible to check advanced security properties such as backward secrecy and forward secrecy. Fig. 5 shows the settings page in the Compromise version of the Scyther tool, which is easy to see where it is possible to check attacks in which it is assumed that long term keys or session keys or random numbers are exposed.

We employed *Scyther Compromise-0.9.2* tool because we needed to prove that our protocol provides both backward and forward security. All available protocols in the Standard version are supported by this version of the Scyther. To confirm that the proposed protocol's backward and forward security is accomplished, in the first run of Scyther codes, we choose the "after (PFC) option", and in the second run, we select the "Session Key Reveal" option, so we present Scyther results in these two conditions.

In the first subsection, we will study the claim events and security attributes in the Scyther tool, and in the second subsection, we will evaluate the security and accuracy of the SAPWSN protocol using the Scyther Compromise-0.9.2 tool.

Table 3
Reader role definitions of the SAPWSN protocol in the Scyther tool.

Role	Code

Public	const XOR : Function;
	var n : Nonce;
	var m: Nonce;
	usertype Timestamp;
	secret s;
	hashfunction h1;

Reader	fresh n:Nonce;
	secret ID_r, ID_t;
	secret K_{rt}, K_{st}, K_{sr};
	fresh T_r : Timestamp;
	var Ts:Timestamp;
	var m:Nonce;
	macro x = XOR(n, K_{rt});
	macro y=h1(x, T_r);
	$send_1(reader, tag, T_r, n, y)$;
	$recv_2(tag, reader, m, h1(m, K_{rt}, n), h1(n, m, K_{st}))$;
	claim(reader,Secret,h1(m, K_{rt},n));
	claim(reader,Secret,h1(n,m,Kst));
	$send_3(reader, server, h1(n, m, K_{st}), h1(n, m, T_r, K_{sr}), n, m, T_r)$;
	$recv_4(server, reader, Ts, XOR(h1(m, Kst, IDt, Ts), h1(n, Ksr)),;$
	$h1(Ts, n, Ksr, IDr, XOR(h1(m, Kst, IDt, Ts), h1(n, Ksr)), IDt))$;
	claim$(reader, Secret, XOR(h1(m, Kst, IDt, Ts), h1(n, Ksr)))$;
	//Server is legitimate
	claim (reader,Secret, $XOR(h1(m, K_{st}, ID_t, Ts), h1(n, K_{sr}))$;
	$send_5(reader, tag, Ts, XOR(XOR(h1(m, K_{st}, ID_t, Ts), h1(n, K_{sr})), h1(n, K_{sr})))$;
	claim(reader,Alive);
	claim(reader,Weakagree);
	claim(reader,Niagree);
	claim(reader,Nisynch);
	claim (reader,Secret, $h1(n, m, K_{st})$;
	claim (reader,Secret, $XOR(XOR(h1(m, K_{st}, ID_t, Ts), h1(n, K_{sr})), h1(n, K_{sr}))$;
	};

Table 4
Tag role definitions of the SAPWSN protocol in the Scyther tool.

Role	Code

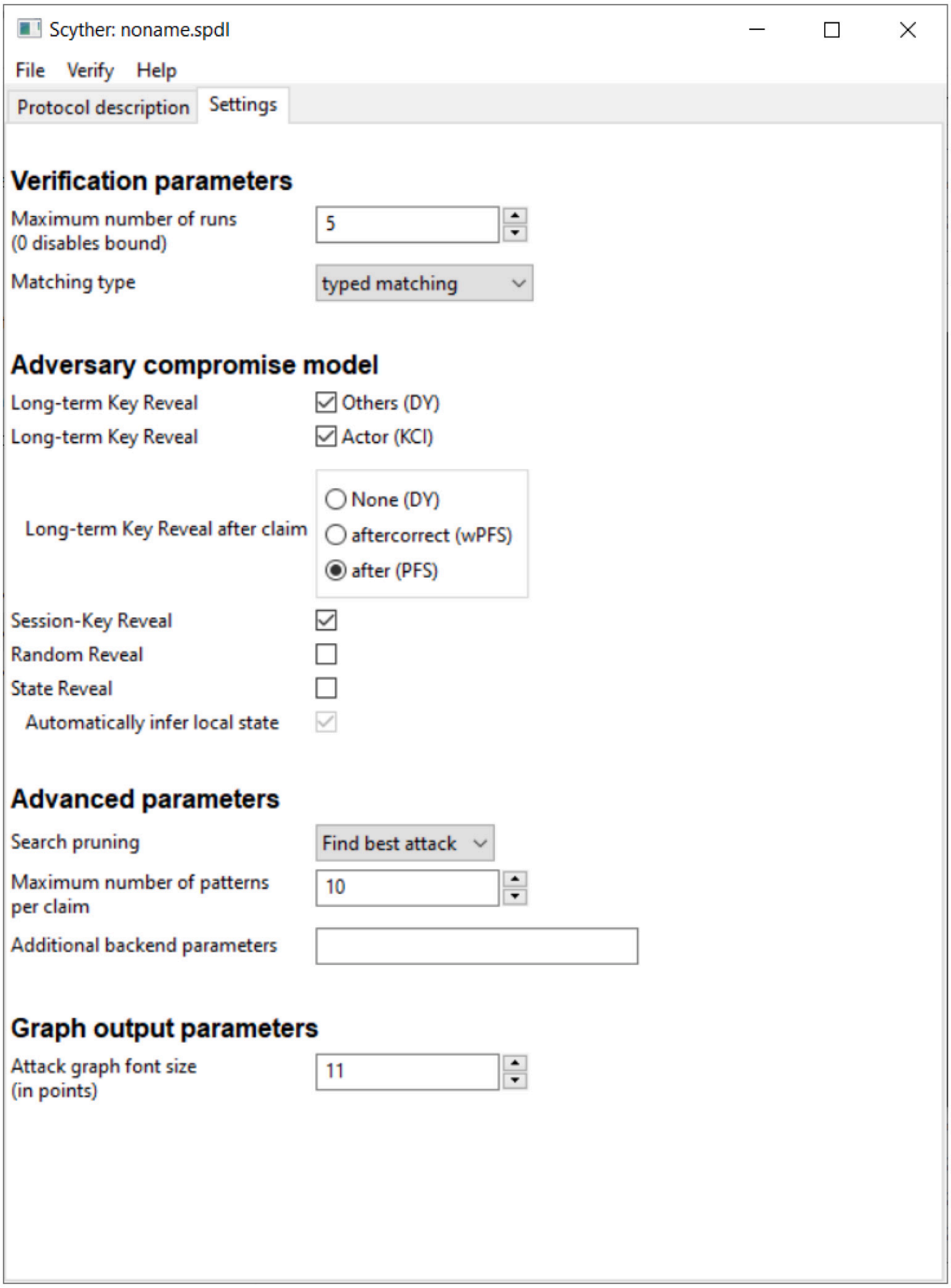
Tag	var n:Nonce;
	fresh m:Nonce;
	secret ID_t;
	secret K_{rt}, K_{st}, K_{sr};
	var T_r, T_s:Timestamp;
	$recv_1(reader, tag, T_r, n, y)$;
	claim(tag,Secret,n);
	claim(tag,Secret,y);
	$claim(tag, Secret, T_r)$;
	$send_2(tag, reader, m, h1(m, K_{rt}, n), h1(n, m, K_{st}))$;
	$recv_5(reader, tag, Ts, XOR(XOR(h1(T_s, m, K_{st}, ID_t), h1(n, K_{sr})), h1(n, K_{sr})))$;
	$claim(tag, Secret, XOR(XOR(h1(T_s, m, K_{st}, ID_t), h1(n, K_{sr})), h1(m, K_{sr})))$;
	claim(tag,Alive);
	claim(tag,Weakagree);
	claim(tag,Niagree);
	claim(tag,Nisynch);
	claim(tag,Secret,h1(m, K_{rt},n));
	claim(tag,Secret,h1(n,m, K_{st}));
	};

5.2.1. The scyther tool's security claim events and security properties

In role specifications of one security protocol in the Scyther tool, claim events are used to simulate the required security attributes. The Scyther tool has a number of predetermined claim types. (see Table 2.)

As shown in Tables 3, 4, and 5, we defined all parties of the proposed protocol in separated roles.

The Scyther tool results for the SAPWSN protocol are depicted in Figs. 6 and 7. Figs. 8 and 9 also show the security of the SAPWSN protocol against all the Scyther tool attacks scenarios while selecting the "Session Key Reveal" option. It is easy to see that the security evaluation results of the proposed protocol in all different cases show that the proposed protocol is secure against all possible attacks.



Verification parameters

Maximum number of runs (0 disables bound) 5

Matching type typed matching

Adversary compromise model

Long-term Key Reveal ☑ Others (DY)

Long-term Key Reveal ☑ Actor (KCI)

Long-term Key Reveal after claim ○ None (DY) ○ aftercorrect (wPFS) ● after (PFS)

Session-Key Reveal ☑

Random Reveal □

State Reveal □

Automatically infer local state ☑

Advanced parameters

Search pruning Find best attack

Maximum number of patterns per claim 10

Additional backend parameters

Graph output parameters

Attack graph font size (in points) 11

Fig. 5. The setting page in the Scyther Compromise-0.9.2 tool.

6. Comparison

We compared our proposed protocol to the most recent similar schemes in terms of security features, computational costs, execution times, communication cost and storage cost in this section. It worth noting that the phases of authentication and updating the secret values is taken into account in the comparison section calculations.

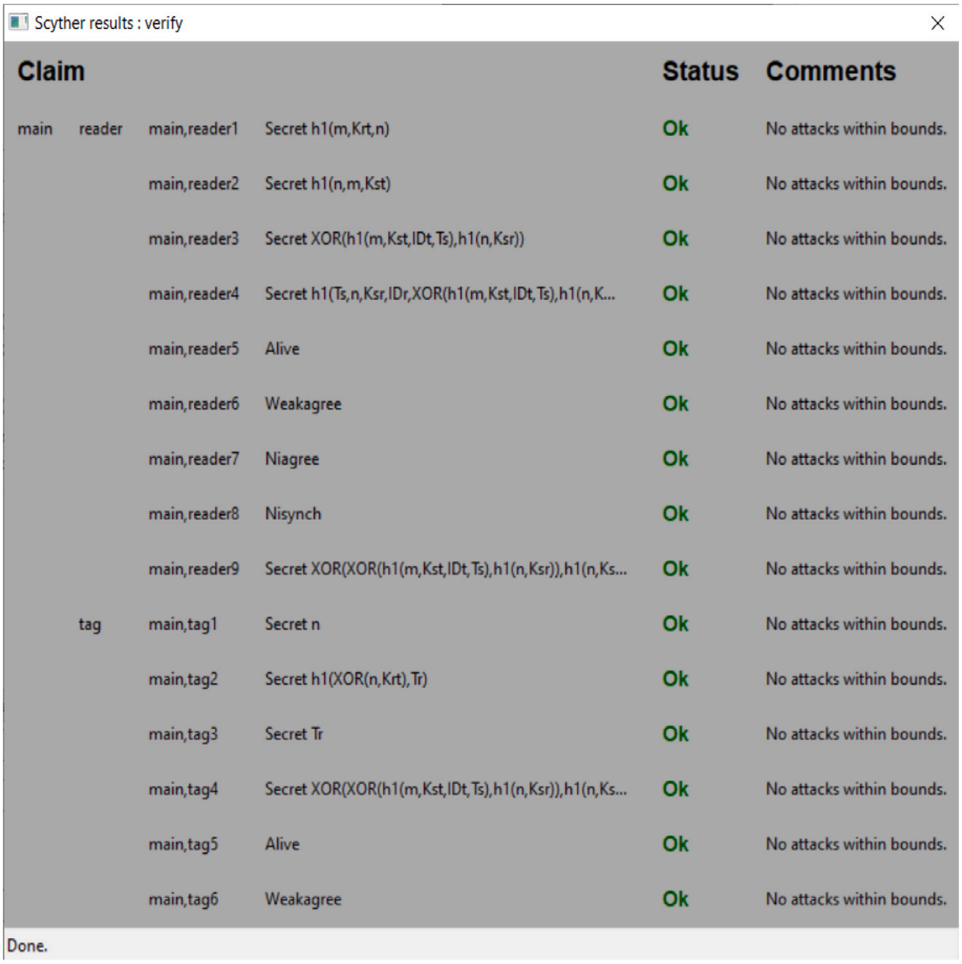
6.1. Comparison of security features

Table 6 compares the security features of our improved protocol to those of the most recent and similar schemes. As can be observed, all of these protocols except ours are vulnerable to one or more threats, but we are attempting to develop a protocol that is resistant to all known active and passive attacks. Sections 5.1 and 5.2 evaluate the security of our improved protocol.

6.2. Comparison of computational cost

The type of cryptographic operation used in the protocol influences its security and execution time. Some designers use lightweight operations such as XOR, Rotation, AND, and OR operations instead of traditional encryption functions to drastically reduce protocol execution time. However, almost all of these schemes have been shown to be vulnerable to various attacks, so we excluded them from the comparison.

In Table 7 and graphically in Fig. 10, we compared the type and the number of operations used in SAPWSN with those of other protocols. In Table 8, the execution time of a modular squaring operation is



```
Scyther results : verify

Claim                                                                          Status  Comments

main  reader  main,reader1  Secret h1(m,Krt,n)                                 Ok      No attacks within bounds.
              main,reader2  Secret h1(n,m,Kst)                                 Ok      No attacks within bounds.
              main,reader3  Secret XOR(h1(m,Kst,IDt,Ts),h1(n,Ksr))             Ok      No attacks within bounds.
              main,reader4  Secret h1(Ts,n,Ksr,IDr,XOR(h1(m,Kst,IDt,Ts),h1(n,K...  Ok  No attacks within bounds.
              main,reader5  Alive                                              Ok      No attacks within bounds.
              main,reader6  Weakagree                                          Ok      No attacks within bounds.
              main,reader7  Niagree                                            Ok      No attacks within bounds.
              main,reader8  Nisynch                                            Ok      No attacks within bounds.
              main,reader9  Secret XOR(XOR(h1(m,Kst,IDt,Ts),h1(n,Ksr)),h1(n,Ks...  Ok  No attacks within bounds.
      tag     main,tag1     Secret n                                           Ok      No attacks within bounds.
              main,tag2     Secret h1(XOR(n,Krt),Tr)                           Ok      No attacks within bounds.
              main,tag3     Secret Tr                                          Ok      No attacks within bounds.
              main,tag4     Secret XOR(XOR(h1(m,Kst,IDt,Ts),h1(n,Ksr)),h1(n,Ks...  Ok  No attacks within bounds.
              main,tag5     Alive                                              Ok      No attacks within bounds.
              main,tag6     Weakagree                                          Ok      No attacks within bounds.
Done.
```

Fig. 6. Security verification results of the SAPWSN protocol using the Scyther Compromise-0.9.2 tool (selecting "after(PFC)" option).

Table 5
Server role definitions of the SAPWSN protocol in the Scyther tool.

Role	Code

Server	var T_r:Timestamp;
	fresh T_s:Timestamp;
	var n:Nonce;
	var m:Nonce;
	secret ID_r, ID_t;
	secret K_{rt}, K_{st}, K_{sr};
	$recv_3(reader, server, h1(n, m, K_{st}), h1(n, m, T_r, K_{sr}), n, m, T_r)$;
	claim(server,Secret,m);
	claim(server,Secret,n);
	claim(server,Secret, T_r);
	claim(server,Secret, $h1(n, m, K_{st})$);
	//Tag is Legitimate
	claim(server,Secret, $h1(n, m, T_r, K_{sr})$);
	//User is Legitimate
	$send_4(server, reader, T_s, XOR(h1(m, K_{st}, ID_t, T_s), h1(n, K_{sr}))$,
	$h1(T_s, n, K_{sr}, ID_r$,
	$XOR(h1(m, K_{st}, ID_t, T_s), h1(n, K_{sr})), ID_t))$;
	claim(server,Alive);
	claim(server,Weakagree);
	claim(server,Niagree);
	claim(server,Nisynch);
	claim(server,Secret, $XOR(h1(T_s, m, K_{st}, ID_t), h1(n, K_{sr}))$);
	};

Table 6
Security comparison of the SAPWSN protocol with other similar protocols.

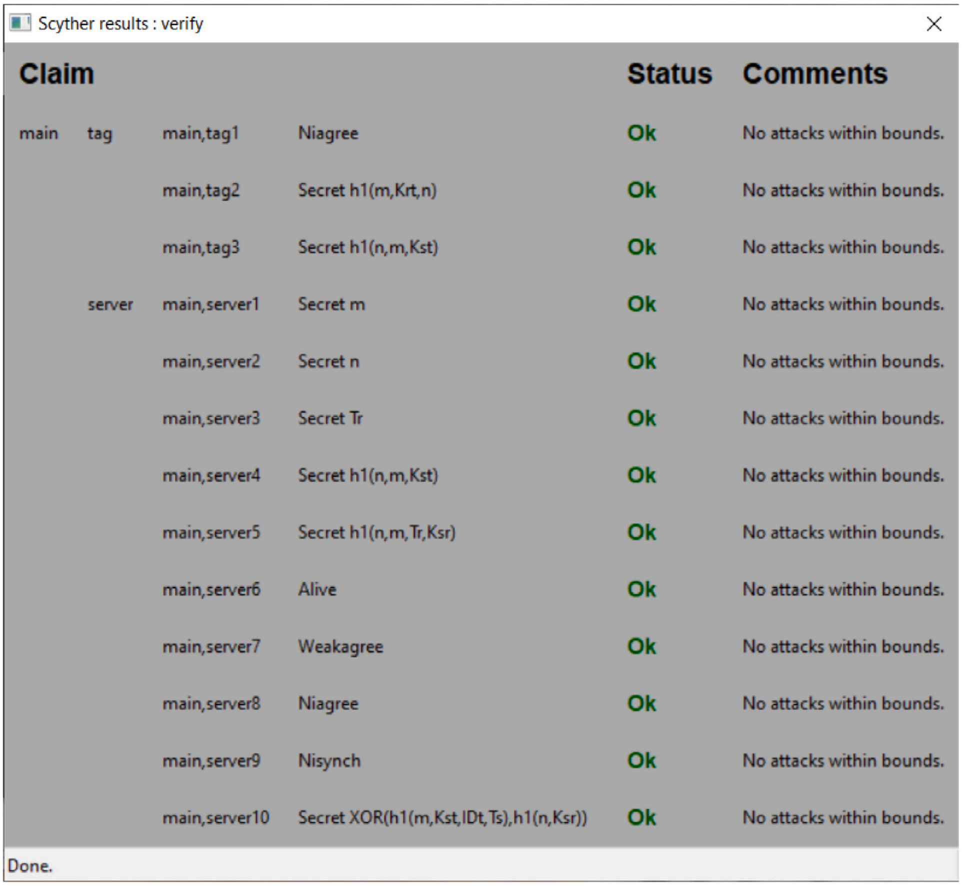
Protocol	A1	A2	A3	A4
Cho et al. [17]	No	No	Yes [21]	Yes [21]
Doss et al. [21]	Yes	No [22]	Yes	Yes
Chiou and Chang [22]	Yes	No(in this paper)	Yes	Yes
Yeh et al. [44]	Yes	No [45]	No [45]	Yes
SAPWSN	Yes	Yes	Yes	Yes

A1: Secret Disclosure Attack Resistance; A2: Backward/Forward Secrecy
A3: Impersonation Attack Resistance, A4: Desynchronization Attack Resistance

represented as T_s, the execution time of square root modular solving is represented as T_r, the execution time of a pseudo random number generation operation is denoted as T_p, and the execution time of a hash operation is indicated as T_h. According to [46], T_s = 1.896 ms, T_r = 3.481 ms, T_p = 0.021 ms, T_h = 0.253 ms, and also because the execution time of the exclusive or operation (Xor) and concatenation operation is negligible, they are ignored at this step. Fig. 11 graphically compares the proposed protocol to other recent similar protocols in terms of the execution time. Based on the results of the calculations presented in Table 8 and Fig. 11, the Cho et al. [17] and SAPWSN protocols have the shortest execution times, but because the Cho et al. [17] protocol is insecure against secret disclosure attack and lacks backward/forward secrecy, we can conclude that our proposed protocol is superior in terms of both security features and execution time.

6.3. Comparison of communication cost

Another criterion for comparing security protocols with each other is the communication costs in the protocol. Communication costs mean the number of bits exchanged during the execution of the protocol

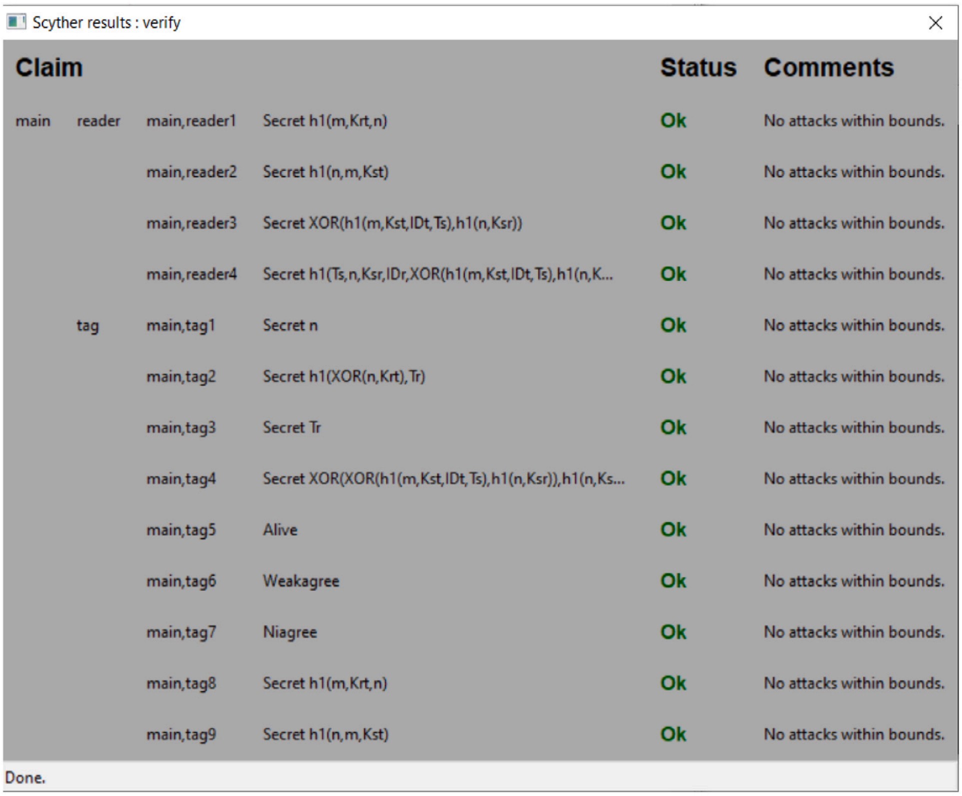


Scyther results : verify

Claim				Status	Comments
main	tag	main,tag1	Niagree	Ok	No attacks within bounds.
		main,tag2	Secret h1(m,Krt,n)	Ok	No attacks within bounds.
		main,tag3	Secret h1(n,m,Kst)	Ok	No attacks within bounds.
	server	main,server1	Secret m	Ok	No attacks within bounds.
		main,server2	Secret n	Ok	No attacks within bounds.
		main,server3	Secret Tr	Ok	No attacks within bounds.
		main,server4	Secret h1(n,m,Kst)	Ok	No attacks within bounds.
		main,server5	Secret h1(n,m,Tr,Ksr)	Ok	No attacks within bounds.
		main,server6	Alive	Ok	No attacks within bounds.
		main,server7	Weakagree	Ok	No attacks within bounds.
		main,server8	Niagree	Ok	No attacks within bounds.
		main,server9	Nisynch	Ok	No attacks within bounds.
		main,server10	Secret XOR(h1(m,Kst,IDt,Ts),h1(n,Ksr))	Ok	No attacks within bounds.

Done.

Fig. 7. Continuation of security verification results of the SAPWSN protocol using the Scyther Compromise-0.9.2 tool (selecting "after(PFC)" option).

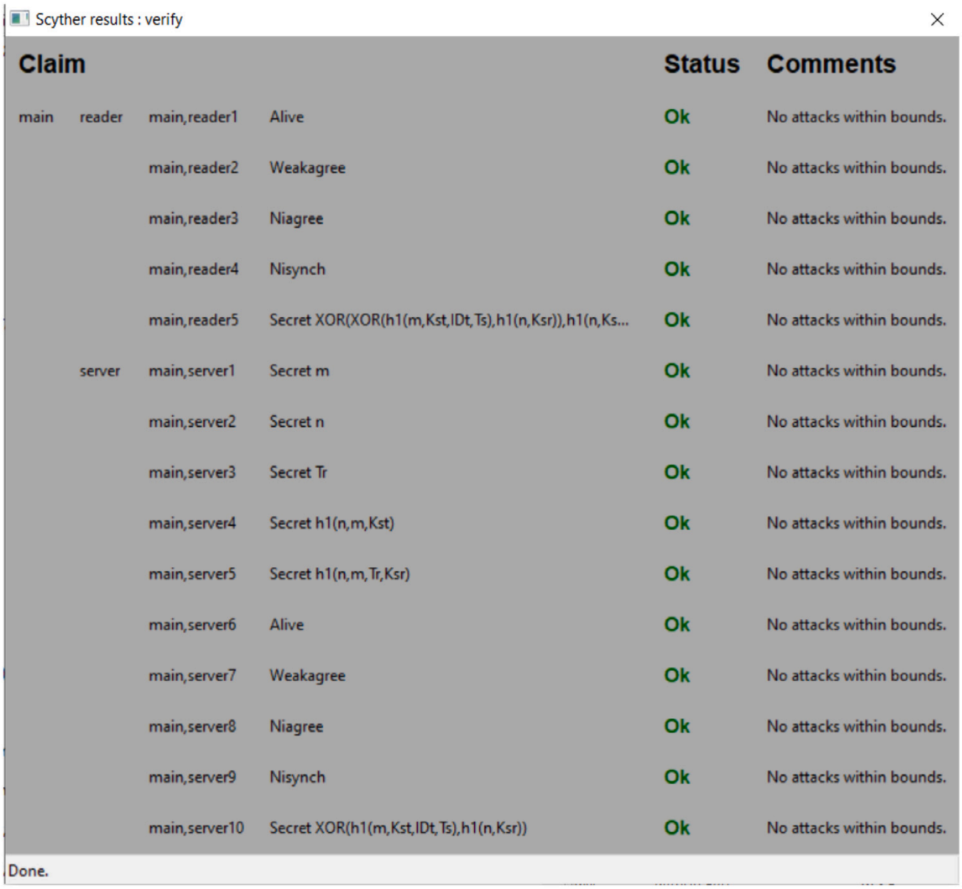


Scyther results : verify

Claim				Status	Comments
main	reader	main,reader1	Secret h1(m,Krt,n)	Ok	No attacks within bounds.
		main,reader2	Secret h1(n,m,Kst)	Ok	No attacks within bounds.
		main,reader3	Secret XOR(h1(m,Kst,IDt,Ts),h1(n,Ksr))	Ok	No attacks within bounds.
		main,reader4	Secret h1(Ts,n,Ksr,IDr,XOR(h1(m,Kst,IDt,Ts),h1(n,K...	Ok	No attacks within bounds.
	tag	main,tag1	Secret n	Ok	No attacks within bounds.
		main,tag2	Secret h1(XOR(n,Krt),Tr)	Ok	No attacks within bounds.
		main,tag3	Secret Tr	Ok	No attacks within bounds.
		main,tag4	Secret XOR(XOR(h1(m,Kst,IDt,Ts),h1(n,Ksr)),h1(n,Ks...	Ok	No attacks within bounds.
		main,tag5	Alive	Ok	No attacks within bounds.
		main,tag6	Weakagree	Ok	No attacks within bounds.
		main,tag7	Niagree	Ok	No attacks within bounds.
		main,tag8	Secret h1(m,Krt,n)	Ok	No attacks within bounds.
		main,tag9	Secret h1(n,m,Kst)	Ok	No attacks within bounds.

Done.

Fig. 8. Security verification results of the SAPWSN protocol using the Scyther Compromise-0.9.2 tool (selecting "Session Key Reveal" option).



Scyther results : verify

Claim				Status	Comments
main	reader	main,reader1	Alive	Ok	No attacks within bounds.
		main,reader2	Weakagree	Ok	No attacks within bounds.
		main,reader3	Niagree	Ok	No attacks within bounds.
		main,reader4	Nisynch	Ok	No attacks within bounds.
		main,reader5	Secret XOR(XOR(h1(m,Kst,IDt,Ts),h1(n,Ksr)),h1(n,Ks...	Ok	No attacks within bounds.
	server	main,server1	Secret m	Ok	No attacks within bounds.
		main,server2	Secret n	Ok	No attacks within bounds.
		main,server3	Secret Tr	Ok	No attacks within bounds.
		main,server4	Secret h1(n,m,Kst)	Ok	No attacks within bounds.
		main,server5	Secret h1(n,m,Tr,Ksr)	Ok	No attacks within bounds.
		main,server6	Alive	Ok	No attacks within bounds.
		main,server7	Weakagree	Ok	No attacks within bounds.
		main,server8	Niagree	Ok	No attacks within bounds.
		main,server9	Nisynch	Ok	No attacks within bounds.
		main,server10	Secret XOR(h1(m,Kst,IDt,Ts),h1(n,Ksr))	Ok	No attacks within bounds.

Done.

Fig. 9. Continuation of security verification results of the SAPWSN protocol using the Scyther Compromise-0.9.2 tool (selecting "Session Key Reveal" option).

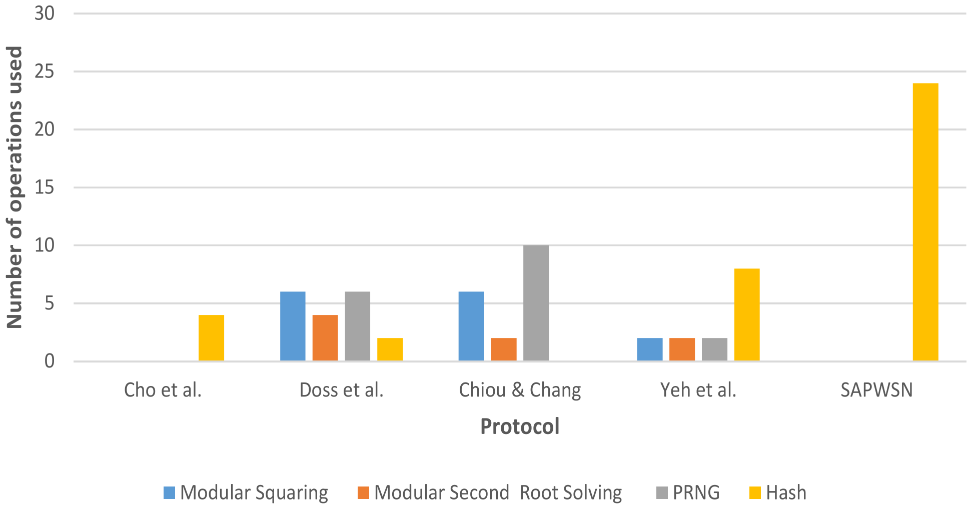


Fig. 10. Comparison of the protocols based on the number of used operations.

between different parties. For this purpose, the length of the parameters used in the protocol is needed, which we have shown in Table 9. After that, we calculated the total communication cost for the considered protocols based on the bit, so by comparing the total communication cost of protocols presented in Table 10, it can be concluded that the SAPWSN protocol can both provide higher security and after the Cho et al. protocol has the best computation cost compared to other protocols. Fig. 12 graphically compares the proposed protocol to other recent similar protocols in terms of the communication cost.

6.4. Comparison of storage cost

Storage costs are usually used to compare different protocols. Storage cost usually refers to the number of bits that the limited party in the protocol, such as a tag or sensor, must store for the correct implementation of the protocol. Usually, because other components involved in the protocol such as servers, gateways or readers do not have many restrictions, they are not considered in these storage costs.

In this section, using the length of the parameters shown in Table 9, the storage cost of the proposed protocol is compared with other

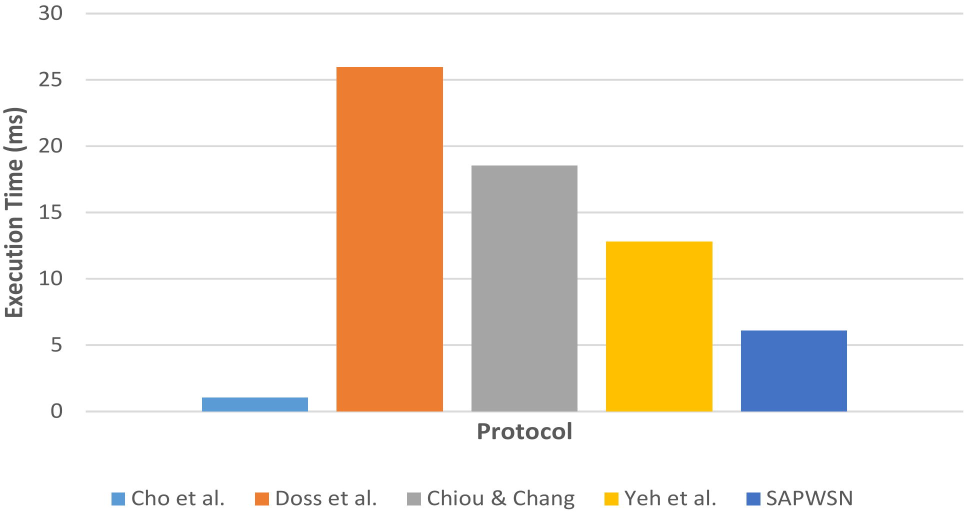


Fig. 11. Comparison of the execution time of protocols.

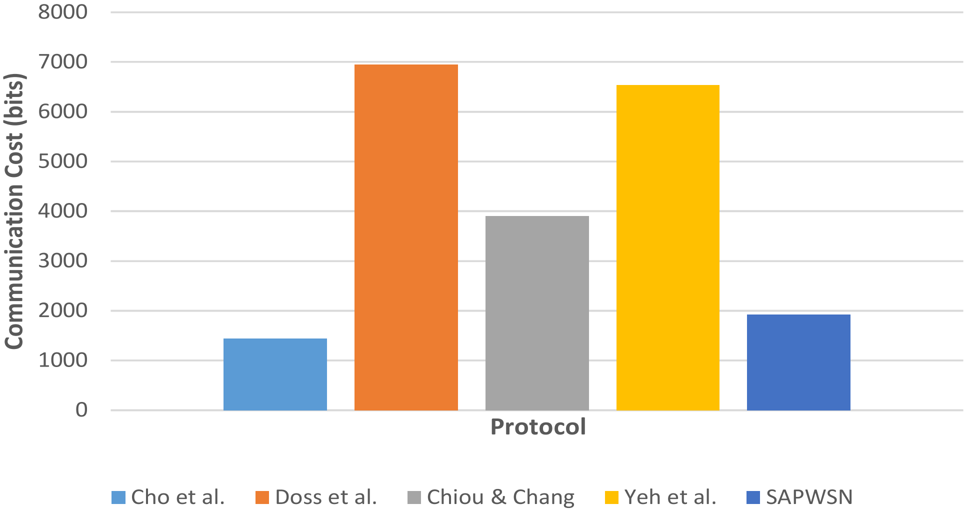


Fig. 12. Communication cost comparison of the SAPWSN protocol with other similar protocols.

Table 7
Comparison of the SAPWSN protocol with other similar protocols based on the type and number of operations where SRMS: Square Root Modular Solving.

Protocol	Squaring	SRMS	PRNG	Hash
Cho et al. [17]	–	–	–	4
Doss et al. [21]	6	4	6	2
Chiou and Chang [22]	6	2	10	–
Yeh et al. [44]	2	2	2	8
SAPWSN	–	–	–	24

Table 8
Execution time comparison of the SAPWSN protocol with other similar protocols (in milliseconds).

Protocol	Computational cost	Execution time (ms)
Cho et al. [17]	$4T_h$	1.012
Doss et al. [21]	$6T_s + 4T_r + 6T_p + 2T_h$	25.932
Chiou and Chang [22]	$6T_s + 2T_r + 10T_p$	18.548
Yeh et al. [44]	$2T_s + 2T_r + 2T_p + 8T_h$	12.82
SAPWSN	$24T_h$	6.072

protocols in Table 11 and Fig. 13. As can be seen, the storage cost in the proposed protocol is lower compared to other protocols after the Cho et al.'s protocol and the proposed protocol has not imposed more storage cost on the sensors to create security.

7. Conclusion

In this paper, we evaluated the security of the Chiou and Chang EPC Class 1 Gen-2 authentication protocol and demonstrated its flaws. To be more specific, we revealed security flaws in the protocol that potentially reveal all of the secret information. The main disadvantage of this protocol is that it is vulnerable to backward security contradiction attack. We also presented a novel way to amend the Chiou and Chang authentication protocol's flaws. Furthermore, we demonstrated by both informal and formal methods that our proposed protocol is secure against the attack mentioned in this paper, as well as all other known active and passive attacks. We used the Compromise version of Scyther tool which is one of the formal methods for assessing the security of all kinds of security protocols.

Table 9
The length of the protocols' parameters based on bit.

Parameter	Symbol	Length(bit)
Hash function	l_h	160 bits
CRC and PRNG	l_c	128 bits
Random numbers	l_r	128 bits
Data	l_d	40 bits
Encrypted/Decryption	l_e	128 bits
Message "hello"	l_m	40 bits
Secret values, modules and random numbers in the quadratic residue-based protocols	l_n	1024 bits
Secret values in the non-quadratic residue based protocols	l_k	128 bits
ID, SID	l_{ID}	128 bits
Time	l_t	32 bits
EPC	l_{epc}	96 bits

Table 10
Comparison of communication cost.

Protocol	Tag	Reader	Server	Total (bit)
Cho et al. [17]	$l_h + l_r$	$2l_h + 4l_r$	$l_d + l_h + l_r$	1448 bits
Doss et al. [21]	$2l_n$	$l_m + 2l_r + l_h + l_c + 4l_n$	$l_h + l_c$	6956 bits
Chiou and Chang [22]	$l_c + l_n$	$3l_c + 2l_n + 2l_t$	$2l_c$	3904 bits
Yeh et al. [44]	$3l_n + 3l_h$	$l_m + 3l_h + 2l_n + 2l_r$	l_h	6536 bits
SAPWSN	$l_r + 2l_h$	$3l_t + 4l_h + 3l_r$	$l_t + 2l_h$	1920 bits

Table 11
Storage cost comparison of the SAPWSN protocol with other similar protocols.

Protocol	Tag	Total
Cho et al. [17]	$l_{ID} + l_K$	256 bits
Doss et al. [21]	$l_r + l_{ID} + l_h + l_n$	1440 bits
Chiou and Chang [22]	$2l_{ID} + l_k + l_n$	2304 bits
Yeh et al. [44]	$l_r + l_{ID} + l_h + l_n$	2336 bits
SAPWSN	$2l_{ID} + 2l_k + l_h$	672 bits

Although the performed analysis shows that the proposed scheme outperforms the most recent related works, any cryptographic protocol should not be trusted unless its security is independently evaluated, which is left as a future work for interested researchers. In addition, to provide a fair comparison for different protocols, we did not consider control unit costs in our cost analysis, instead focusing on the number of calls to cryptographic primitives. However, each platform has its own recruitment process. As a result, another research direction is to create prototypes based on the proposed protocol and test its efficiency in various platforms, such as microcontrollers and FPGAs. From the standpoint of design strategy, it is worth noting that WSN sensors are dispersed throughout the field, so an adversary could always capture them and read their memory directly or using side-channel attacks. In that case, cloning a sensor might be possible. Designers commonly use Physically unclonable functions (PUFs) as a device fingerprint to avoid such attacks in the literature. Although this approach has its own set of advantages and disadvantages, it is intriguing to consider adapting it to the proposed protocol in this study.

CRediT authorship contribution statement

Foroozan Ghosairi Darbandeh: Conception and design of study, Acquisition of data, Analysis and/or interpretation of data, Software, Writing – original draft, Writing – review & editing. **Masoumeh Safkhani:** Conception and design of study, Acquisition of data, Analysis and/or interpretation of data, Validation, Supervision, Formal analysis, Writing – original draft, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No data was used for the research described in the article.

Acknowledgment

This work was supported by Shahid Rajaee Teacher Training University.

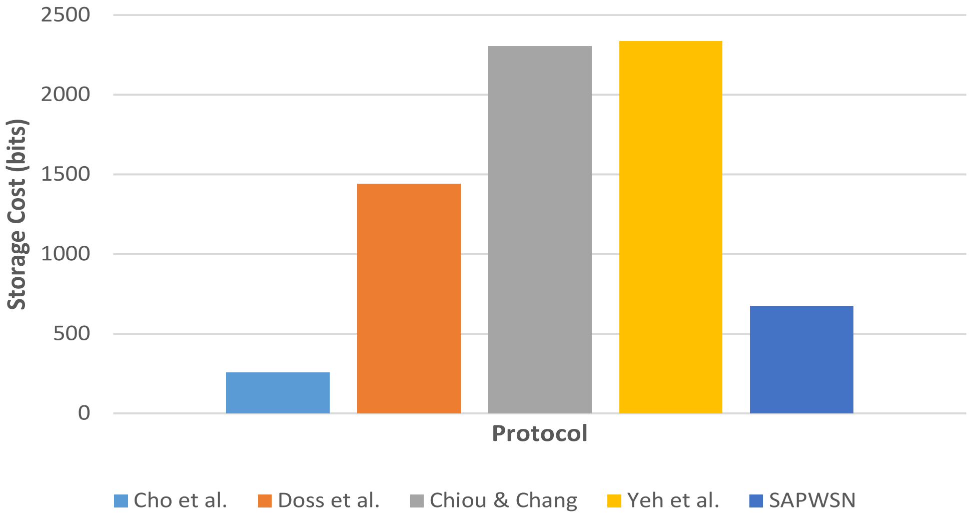


Fig. 13. Comparison of the protocols based on their storage cost.

References

[1] Waleed Kareem Ahmed, Rana Saad Mohammed, Lightweight authentication methods in IoT: Survey, in: 2022 International Conference on Computer Science and Software Engineering (CSASE), IEEE, 2022, pp. 241–246.

[2] Adil Adeel, Mazhar Ali, Abdul Nasir Khan, Tauqeer Khalid, Faisal Rehman, Yaser Jararweh, Junaid Shuja, A multi-attack resilient lightweight IoT authentication scheme, Trans. Emerg. Telecommun. Technol. 33 (3) (2022) e3676.

[3] Xueping Zhu, Yanxing Li, Yuan Lei, A forwarding secrecy based lightweight authentication scheme for intelligent logistics, in: 2020 IEEE International Conference on Advances in Electrical Engineering and Computer Applications (AEECA), IEEE, 2020, pp. 356–360.

[4] Xingmiao Wang, Kai Fan, Kan Yang, Xiaochun Cheng, Qingkuan Dong, Hui Li, Yintang Yang, A new RFID ultra-lightweight authentication protocol for medical privacy protection in smart living, Comput. Commun. 186 (2022) 121–132.

[5] V. Haribaabu, Jospeh James, Selvakumara Samy, Nilesh Singh, Aparna Upadhyay, The lightweight algorithm for secure RFID authentication system, in: Journal of Physics: Conference Series, 2007, IOP Publishing, 2021, 012038.

[6] Alsaify Baha'A, Dale R. Thompson, Abdallah Alma'aitah, Jia Di, Using dummy data for RFID tag and reader authentication, Digit. Commun. Netw. (2021).

[7] Shehzad Ashraf Chaudhry, Azeem Irshad, Khalid Yahya, Neeraj Kumar, Mamoun Alazab, Yousaf Bin Zikria, Rotating behind privacy: an improved lightweight authentication scheme for cloud-based IoT environment, ACM Trans. Int. Technol. (TOIT) 21 (3) (2021) 1–19.

[8] David Molnar, David Wagner, Privacy and security in library RFID: Issues, practices, and architectures, in: Proceedings of the 11th ACM Conference on Computer and Communications Security, ACM, 2004, pp. 210–219.

[9] Ari Juels, Ravikanth Pappu, Squealing euros: Privacy protection in RFID-enabled banknotes, in: International Conference on Financial Cryptography, Springer, 2003, pp. 103–121.

[10] Keunwoo Rhee, Jin Kwak, Seungjoo Kim, Dongho Won, Challenge-response based RFID authentication protocol for distributed database environment, in: International Conference on Security in Pervasive Computing, Springer, 2005, pp. 70–84.

[11] Ari Juels, Yoking-proofs for RFID tags, in: Pervasive Computing and Communications Workshops, 2004. Proceedings of the Second IEEE Annual Conference on, IEEE, 2004, pp. 138–143.

[12] Yalin Chen, Jue-Sam Chou, Hung-Min Sun, A novel mutual authentication scheme based on quadratic residues for RFID systems, Comput. Netw. 52 (12) (2008) 2373–2380.

[13] Kirk H.M. Wong, Patrick C.L. Hui, Allan C.K. Chan, Cryptography and authentication on RFID passive tags for apparel products, Comput. Ind. 57 (4) (2006) 342–349.

[14] Ronggao Zhang, An enhanced lightweight authentication protocol for low-cost RFID systems, in: Electronic Information and Communication Technology (ICEICT), IEEE International Conference on, IEEE, 2016, pp. 29–33.

[15] Aakanksha Tewari, B.B. Gupta, Cryptanalysis of a novel ultra-lightweight mutual authentication protocol for IoT devices using RFID tags, J. Supercomput. 73 (3) (2017) 1085–1102.

[16] King-Hang Wang, Chien-Ming Chen, Weicheng Fang, Tsu-Yang Wu, On the security of a new ultra-lightweight authentication protocol in IoT environment for RFID tags, J. Supercomput. 74 (1) (2018) 65–70.

[17] Jung-Sik Cho, Sang-Soo Yeo, Sung Kwon Kim, Securing against brute-force attack: A hash-based RFID mutual authentication protocol using a secret value, Comput. Commun. 34 (3) (2011) 391–397.

[18] Gaurav Kapoor, Selwyn Piramuthu, Vulnerabilities in Chen and Deng's RFID mutual authentication and privacy protection protocol, Eng. Appl. Artif. Intell. 24 (7) (2011) 1300–1302.

[19] Mohd Shariq, Karan Singh, A novel vector-space-based lightweight privacy-preserving RFID authentication protocol for IoT environment, J. Supercomput. 77 (8) (2021) 8532–8562.

[20] Mohammad Mamun, Atsuko Miyaji, Rongxing Luv, Chunhua Su, A lightweight multi-party authentication in insecure reader-server channel in RFID-based IoT, Peer-To-Peer Netw. Appl. 14 (2) (2021) 708–721.

[21] Robin Doss, Saravanan Sundaresan, Wanlei Zhou, A practical quadratic residues based scheme for authentication and privacy in mobile RFID systems, Ad Hoc Netw. 11 (1) (2013) 383–396.

[22] Shin-Yan Chiou, Shan-Yen Chang, An enhanced authentication scheme in mobile RFID system, Ad Hoc Netw. 71 (2018) 1–13.

[23] Xueping Ren, Xianghua Xu, A mutual authentication protocol for low-cost RFID system, in: 2010 IEEE Asia-Pacific Services Computing Conference, IEEE, 2010, pp. 632–636.

[24] Xiong Li, Tian Liu, Mohammad S. Obaidat, Fan Wu, Pandi Vijayakumar, Neeraj Kumar, A lightweight privacy-preserving authentication protocol for VANETs, IEEE Syst. J. 14 (3) (2020) 3547–3557.

[25] Salman Shamshad, Muhammad Asad Saleem, Mohammad S. Obaidat, Usman Shamshad, Khalid Mahmood, Muhammad Faizan Ayub, On the security of a lightweight privacy-preserving authentication protocol for VANETs, in: 2021 International Conference on Artificial Intelligence and Smart Systems, ICAIS, IEEE, 2021, pp. 1766–1770.

[26] Jung Yeon Hwang, Su-Mi Lee, Dong Hoon Lee, Scalable key exchange transformation: from two-party to group, Electron. Lett. 40 (12) (2004) 1.

[27] Jung-San Lee, Chin-Chen Chang, Kuo-Jui Wei, Provably secure conference key distribution mechanism preserving the forward and backward secrecy, Int. J. Netw. Secur. 15 (5) (2013) 405–410.

[28] Pankaj Kumar, Hari Om, A conditional privacy-preserving and desynchronization-resistant authentication protocol for vehicular ad hoc network, J. Supercomput. (2022) 1–32.

[29] Teklay Gebremichael, Mikael Gidlund, Gerhard P. Hancke, Ulf Jennehag, Quantum-safe group key establishment protocol from lattice trapdoors, Sensors 22 (11) (2022) 4148.

[30] Jihyeon Ryu, Hakjun Lee, Youngsook Lee, Dongho Won, SMASG:secure mobile authentication scheme for global mobility network, IEEE Access 10 (2022) 26907–26919.

[31] Prosanta Gope, S.K. Hafizul Islam, Mohammad S. Obaidat, Ruhul Amin, Pandi Vijayakumar, Anonymous and expeditious mobile user authentication scheme for GLOMONET environments, Int. J. Commun. Syst. 31 (2) (2018) e3461.

[32] Prasanta Kumar Roy, Ansuman Bhattacharya, Secure and authentic anonymous roaming service, Wirel. Pers. Commun. (2022) 1–21.

[33] Xiaobei Yan, Maode Ma, A privacy-preserving handover authentication protocol for a group of MTC devices in 5G networks, Comput. Secur. 116 (2022) 102601.

[34] Jian Shen, Ziyuan Gui, Xiaofeng Chen, Jun Zhang, Yang Xiang, Lightweight and certificateless multi-receiver secure data transmission protocol for Wireless Body Area networks, IEEE Trans. Dependable Secur. Comput. 19 (3) (2022) 1464–1475.

[35] Jian Shen, Ziyuan Gui, Xiaofeng Chen, Jun Zhang, Yang Xiang, Lightweight and certificateless multi-receiver secure data transmission protocol for Wireless Body Area networks, IEEE Trans. Dependable Secur. Comput. 19 (3) (2022) 1464–1475.

[36] Omar Basem, Abrar Ullah, Hani Ragab Hassen, Stick: an end-to-end encryption protocol tailored for social network platforms, IEEE Trans. Dependable Secur. Comput. (2022) 1.

[37] L. Ellen Funderburg, Im-Yeong Lee, Efficient short group signatures for conditional privacy in vehicular Ad Hoc networks via ID caching and timed revocation, IEEE Access 9 (2021) 118065–118076.

[38] Zaher Haddad, Mohamed Baza, Mohamed M.E.A. Mahmoud, Waleed Alasmary, Fawaz Alsolami, Secure and efficient AKA scheme and uniform handover protocol for 5G network using blockchain, IEEE Open J. Commun. Soc. 2 (2021) 2616–2627.

[39] Cas J.F. Cremers, The Scyther tool: Verification, falsification, and analysis of security protocols, in: International Conference on Computer Aided Verification, Springer, 2008, pp. 414–418.

[40] Gavin Lowe, A hierarchy of authentication specifications, in: Proceedings 10th Computer Security Foundations Workshop, IEEE, 1997, pp. 31–43.

[41] Ross J. Anderson, Roger M. Needham, Robustness principles for public key protocols, in: Don Coppersmith (Ed.), Advances in Cryptology - CRYPTO '95, 15th Annual International Cryptology Conference, Santa Barbara, California, USA, August 27-31, 1995, Proceedings, in: Lecture Notes in Computer Science, Vol. 963, Springer, 1995, pp. 236–247.

[42] Martín Abadi, Roger M. Needham, Prudent engineering practice for cryptographic protocols, IEEE Trans. Softw. Eng. 22 (1) (1996) 6–15.

[43] Danny Dolev, Andrew Yao, On the security of public key protocols, IEEE Trans. Inform. Theory 29 (2) (1983) 198–208.

[44] Tzu-Chang Yeh, Chien-Hung Wu, Yuh-Min Tseng, Improvement of the RFID authentication scheme based on quadratic residues, Comput. Commun. 34 (3) (2011) 337–341.

[45] Robin Doss, Wanlei Zhou, Saravanan Sundaresan, Shui Yu, Longxiang Gao, A minimum disclosure approach to authentication and privacy in RFID systems, Comput. Netw. 56 (15) (2012) 3401–3416.

[46] Farokhlagha Moazami, Masoumeh Safkhani, $TBGODP^{+}$: improvement of TB-GODP, a time bound group ownership delegation protocol, J. Ambient Intell. Humaniz. Comput. 13 (6) (2022) 3283–3302.

Foroozan Ghosairi Darbandeh received her M.Sc. in computer engineering from Computer Engineering Department, Shahid Rajaee Teacher Training University, Tehran, Iran. Her research interests include Wireless Sensor Networks Security and Software Engineering.

Masoumeh Safkhani received the Ph.D. degree in Electrical Engineering from the Iran University of Science and Technology, in 2012, with the security analysis of RFID protocols as her major field. She is currently an Associate Professor with the Computer Engineering Department, Shahid Rajaee Teacher Training University, Tehran, Iran. Her current research interests include the security analysis of lightweight and ultra-lightweight protocols, targeting constrained environments, such as RFID, the IoT, VANET, and WSN. She is the author/coauthor of over 70 technical articles in information security and cryptology in major international journals and conferences.