World Academy of Science, Engineering and Technology
International Journal of Computer and Systems Engineering
Vol:18, No:7, 2024

Tag Impersonation Attack on Ultra-Lightweight Radio Frequency Identification Authentication Scheme

Reham Al-Zahrani, Noura Aleisa

***Abstract*—**The proliferation of Radio Frequency Identification (RFID) technology has raised concerns about system security, particularly regarding tag impersonation attacks. Regarding RFID systems, an appropriate authentication protocol must resist active and passive attacks. A tag impersonation occurs when an adversary's tag is used to fool an authenticating reader into believing it is a legitimate tag. The paper thoroughly analyses the security of the Efficient, Secure, and Practical Ultra-Lightweight RFID Authentication Scheme (ESRAS). It examines the protocol within the context of RFID systems and focuses specifically on its vulnerability to tag impersonation attacks. The Scyther tool is utilized to assess the protocol's security, providing a comprehensive evaluation of ESRAS's effectiveness in preventing unauthorized tag impersonation.

***Keywords*—**RFID, radio frequency identification, impersonation attack, authentication, ultra-lightweight protocols, security.

I.Introduction

OVER the past few years, the demand for true end-to-end visibility and traceability has continued to grow, owing to increasing consumer demand for transparency and a need for security, accuracy, and auditability. Tracking products and assets is essential to effective operations and supply chain management. Physical objects can be identified and positioned, and their status - at the moment and point of use - provides highly desired supply chain visibility to businesses. The ability to track raw materials, work-in-progress, and finished products can assist retailers and manufacturers in reducing counterfeit products, controlling inventory effectively, responding to changes in demand, and planning the supply chain more effectively. Several technologies have been developed for tracking products in recent years, including quick response codes (QR codes), RFID, and the Internet of Things (IoT). One can track goods in real time by combining satellite navigation and telematics systems [1].

Technology based on RFID is becoming increasingly prevalent, especially in supply chain management, transportation, payment, passport systems, and smart communities, including universities, hospitals, and libraries [2].

RFID system uses radio communications to identify physical objects [3]. In study conducted by Ibrahim [4], he focuses on the development of a small tag for identifying the correct object. There has been considerable discussion regarding the importance of this technology in recent decades. RFID system comprises three main components: a server, a reader, and a tag [5]. The tags can be classified into three types, depending on the power supply system: active, semi-active, and passive tags. Active RFID tags require internal batteries to power the electronic components and create a reply signal to the reader. The semi-active (also termed semi-passive) tag requires no batteries other than powering the microchip circuit board. Using the reader's radio signal, the tag harvests energy to generate a reply signal. Passive tags obtain their energy from the reader [4]. Computationally intensive algorithms do not protect these tags to provide privacy and security. It is recommended that the target protocol be composed of a few computationally efficient primitives to achieve a low manufacturing cost [6].

When RFID systems use non-secure transmission channels, communication between the reader and the tag can be compromised by several attacks. The purpose of these threats is to provide an attacker with the possibility of intercepting this communication or obtaining confidential data to impersonate one of the legitimate parties [7].

Possible attacks against an RFID system can be:

- *Tracking attack:* RFID tags are commonly targeted by tracking tags. Consequently, all tags contain unique identifiers. Using a malicious reader, it is possible to obtain strong tracking information by simply reading a tag attached to an individual or an object. A fixed tag's identifier can be read by an attacker using many readers. By combining tag identifiers with personal information, this attack will be exacerbated [8].
- Denial-of-service (DoS) attack: Readers and tags are unable to communicate with each other because the adversary overloads them with requests or overburdens the communication channels with requests [9].
- Desynchronization attack: A reader and tag are desynchronized by the attacker.
- Man-in-the-middle attack: The message flow is being controlled by the attacker.
- Impersonation attack: The attacker forged an authenticated tag and acted as if it were legitimate.
- Cloning attack: Readers are fooled into believing that they are receiving data from legitimate tags by the attacker.
- Disclosure attack: The attacker compromises all of the tag's confidential data.
- Eavesdropping: Communication channels are eavesdropped upon by the attacker.
- Replay attack: Data are intercepted and re-transmitted by the attacker, possibly as part of a masquerade attack through packet substitution [4].

Reham Al-Zahrani and Noura Aleisa are with College of Computing and Informatics, Saudi Electronic University, Saudi Arabia (e-mail: n.aleisa@se.edu.seu).

Impersonation attacks operate on the principle of obtaining reader information or tag information to create an enemy entity. The enemy entity is subsequently posed as a legitimate entity to communicate [7].

Several measures protect RFID device operations and communications. These include mutual authentication, confidentiality, indistinguishability, forward security, and desynchronization resilience. Mutual authentication is a process by which the reader and tag verify each other's identity. In order for data to be considered confidential, these should be transmitted using authentication or encryption, to prevent data disclosure to third parties. A property of indistinguishability is that the transmitted data do not allow the individual to be tracked. All data sent should be different from what was sent previously, to ensure this property. Concerning forward secrecy, the information transmitted in the current session cannot be used to reveal the information transmitted in previous sessions. Desynchronization resilience refers to the ability to withstand an adversary's attempt to desynchronize a tag and a reader. In order for a tag and reader to be desynchronization resilient, the shared confidential data must be identical on both devices [9].

RFID systems are vulnerable to attacks if they are not authenticated. Once the server has validated the identity of the RFID tag, it begins to trust it. The reader can access authenticated tags once they have been authenticated. After the variety of security threats that could be associated with RFID tags, it is essential that authentication protocols, regardless of their class, address all or most of these issues [4]. However, RFID tags should also be considered as having limited capabilities [10].

The Efficient, Secure, and Practical Ultra-Lightweight RFID Authentication Scheme (ESRAS) is a newly proposed ultra-lightweight rank operation that provides high security at low cost, as described within the literature review below. Following the ESRAS analysis concerning its security and performance, it was found to withstand several known attacks - such as disclosure, desynchronization, and tag tracking attacks [11].

The research contribution of this study was to evaluate the effectiveness and resistance of the ESRAS Scheme against RFID physical attacks - such as tag impersonation attacks – through two differing methods (with/without authentication), formally employing a security analysis protocol tool and followed by outcome comparative analyses.

This study assumed that ESRAS schemes can resist impersonation attacks. To validate this hypothesis, it was evaluated using the Scyther tool.

A. Problem Statement

This study focused on authentication protocol concerning RFID systems, between tag and reader, validating the security of ESRAS scheme (ultra-lightweight protocol) - as proposed recently by [11] against tag impersonation attacks – through the employment of the Scyther tool.

B. Research Aim

Authentication is one of the security concerns of RFID tag attacks. This research aimed to validate the security of recently proposed authentication protocols of ESRAS schemes for RFID tags, based upon impersonation attacks. Considering previous literature findings, the paper examines the properties and features associated with the identified technology. It explores the security properties, with a particular emphasis on ensuring their integrity. By building upon the existing body of knowledge, the paper aims to enhance our understanding of the technology's capabilities and potential implications for security.

C. Research Question

To meet the aim, this investigation formulated the research question:

- Can the ultra-lightweight authentication protocols of ESRAS schemes resist tag impersonation attacks?

D. Research Objectives

- Review the physical attack upon the RFID system.
- Investigate recently proposed schemes for authentication protocols.
- Perform the tag impersonation attack.
- Analyze the effectiveness of ESTAS schemes with/without authentication.
- Verification under two differing adversary models and comparative analyses of evaluation outcomes.

II. Literature Review

RFID refers to wireless communication technology widely utilized across various applications, including healthcare, asset tracking, IoT, supply chain management, and anti-counterfeiting systems. RFID has the advantages of automation, real-time tracking, and cost-effectiveness in asset tracking systems. However, like other technologies, RFID tags are vulnerable to various security threats, including physical attacks. This can lead to data theft, unauthorized access, and other security breaches.

Consequently, securing RFID tags from physical threats is essential to ensure the integrity and confidentiality of the data stored on such tags. It further explains the concept of each attack and how it can be prevented.

This literature review focuses on attacks upon RFID tags, specifically concerning tag cloning, replay, and DoS attacks. It will discuss differing proposed RFID countermeasures based on their security properties - including confidentiality, integrity, and availability.

A. Confidentiality

Confidentiality in RFID is one of the security concerns to prevent any unauthorized access to stored data, or during transmission, while simultaneously rendering it accessible to those with authorized access. Consequently, cryptographic techniques such as encryption and hashing need to be implemented. This article focused on one specific attack and countermeasures to prevent it [12].

Replay Attacks: Replay attacks are Channel attacks on RFID tags, where attackers intercept and replay legitimate RFID signals to gain unauthorized access. According to [13], an anonymous authentication protocol is lightweight and practical for RFID systems by applying physically unclonable functions (PUFs) to prevent replay attacks. PUFs are security primitives based on hardware that employs the intrinsic device's physical properties to generate unique cryptography keys. However, several limitations exist for the scheme that was developed in 2022, although this group has proved that protocol/system security presently runs with high reliability. Notwithstanding, since the proposed scheme can be used in a wide range of scenarios, and since the device's application environment is complex, other security threats could arise in practical applications, and more practical engineering applications are lacking to evaluate scheme reliability.

Alternatively, [14] provides countermeasures for replay attack, that include timestamping, and dynamic updating of information one-time password (OTP). Another effective approach for defeating replay attacks is storing the message hash code at the receiving end. If the message is new, its hash code will not be found at the receiving end, though if the message is replayed, its hash code will be found at the receiving end and consequently discarded.

In [12], the proposed solution states two possible methods; either by a timestamp, or by prevention using the nonce option, where the values generated will be used only once, and hence the identity cannot be spoofed.

B. Integrity

The integrity of data stored on RFID tags is another essential security property that needs to be protected, including preventing unauthorized data manipulation.

Data integrity can be compromised if the received data are not identical to the sent or if the data become deleted while travelling on network/s. Consequently, security must be ensured against tampering, as well as data deletion. Integrity must be ensured at multiple levels in a system, either by proper encryption or access control, although specific integrity attacks require special solutions. Tag cloning attacks can affect system integrity [12], and are discussed below.

Tag Cloning: One of the most common physical attacks on RFID tags poses a significant threat to RFID system security. Similar to generic cloning attacks, tag cloning attacks are security threats in which the attacker creates a clone of either a RFID tag or an RFID reader, impersonating the genuine device/s and gaining access to restricted data, area, systems, and user accounts by using the cloned tag's credentials. Data encryption and authentication protocols are the most common and effective countermeasures [15].

One study [3] proposed a lightweight mutual authentication scheme for RFID systems with resource-constrained tags. This novel mechanism provides authenticity for low-cost RFID systems, fulfilling EPC Gen2, and providing a novel pseudorandom number generator (PRNG), whose design is based on nonlinear filtering of a linear-feedback shift register (LFSR). Both the reader and tag can use the proposed method to authenticate each other mutually and establish a shared-session secret key. It assumes the reader is linked through a secure communication channel to a back-end server with a database. Both systems meet all practical requirements of low-cost RFID, including security properties such as confidentiality, and mutual tag authentication.

Consequently, both schemes are immune against known attacks on PRNGs and authentication schemes. In [3], researchers provided several topics for further research, such as developing formal security proofs for protocols. Furthermore, the authentication scheme could be adapted for employment whenever the channel between the back-end server and the reader is insecure.

Reference [4] examined and compared recently proposed RFID authentication protocols. This investigation concluded the requirement for additional/stronger ultra-lightweight authentication protocols. It is thus necessary to propose a robust authentication protocol with an integrated approach to deal with all - or at least most – threats, apart from other mentioned protocols.

Reference [16] proposed desynchronization attack to a traceability attack and an enhanced version of the Rabin public key-based protocol to provide a secure authentication between the tag and reader. Consequently, using the Scyther tool, the study evaluated the security of the proposed protocol. The security analysis demonstrated that the enhanced protocol provides the desired security against differing attacks, such as traceability, impersonation, and desynchronization attacks. Moreover, the proposed attack was successful on the hash-based and Rabin public key-based protocols.

A comparison of security among various ultra-lightweight authentication schemes, provided in [11], revealed that several schemes were unsatisfactory, with no discussion of mutual authentication. This paper offered an efficient and secure ultra-lightweight RFID authentication scheme (ESRAS) that defies possible security attacks.

The article [17] proposed an authentication scheme for passive RFID tags and their readers: Decoy-Based Authentication (DBA). The experimental results revealed that tags could be authenticated with 100% accuracy. Since RFID readers can also be compromised, this research group developed a technique to authenticate the RFID reader using decoys. The experimental results [10] showed that 100% of the available readers could be authenticated, also identifying which reader was compromised, based upon responses sent to the backend server. The research provides a future study to plan further evaluation of the proposed method, using empirical data instead of simulated data, which will enhance the reliability of the proposed approach.

C. Availability

Within the RFID system, the authentication and key agreement procedure run continuously between the RFID tag and the RFID back-end database server. In most authentication methods, the shared confidential data between the RFID tag and the RFID back-end database server must be updated to achieve accessibility. However, security risks such as DoS or de-synchronization attacks could disrupt this process. The RFID system's efficiency could be harmed due to such concerns. Consequently, when designing an authentication protocol, this issue should be considered [18].

DoS Attacks: can disrupt the authorized tags and readers communication. In these attacks, the adversary intentionally floods the server with multiple signals in the form of responses, making the system inaccessible for further communications. DoS can cause a tag modification attack by allowing the attacker to modify the Electronic Product Code (EPC) data on RFID tags into a random number not recognized by the reader [7].

In [7], the study proposed a novel, secure error correction code (ECC)-based RFID authentication protocol. The proposed protocol is performed against server spoofing, tag impersonation, position tracking, and replay attacks, with the ability to provide mutual authentication. Results indicated that the proposed protocol could be practically implemented within RFID environments, in order to improve reliability and security. Concerning future research, an RFID tag architecture can be implemented using the proposed protocol and applying an ECC cryptosystem secured against side-channel attacks.

Finally, based on previous literature reviews, most studies require additional and more robust ultra-lightweight authentication protocols. It is thus necessary to propose a robust authentication protocol. We have conducted individual investigations into all recent articles concerning RFID authentication protocols. The findings demonstrated proposed schemes for differing methods to examine the efficiency of authentication protocols against physical attacks. Several articles referred to selected limitations in their study concerning the effectiveness of the suggested protocol should the communication channels be insecure. Furthermore, a comparison between the proposed authentication protocols could be developed. Moreover, several physical attacks that were not implemented or evaluated could be launched against existing authentication protocols.

III.Methods and Materials

A. Research Design and Methodology Structure

The research methodology was tailored to quantitative research. This study used data collection to formally answer the research question concerning evaluating the ESRAS scheme effectiveness proposed by [11] against tag impersonation attacks. This was performed through a security analysis protocol tool, such as the Scyther tool, with two differing methods (with/without authentication), followed by a comparative analysis of outcomes for potentially validating the research hypothesis/research question (Fig. 1).

B. Preliminaries

Notations are used throughout this article, following the same notation from the study by Shariq and colleagues [11], as described in Table I.

Adversary Model: Defining the potential capabilities of an adversary is the purpose of an adversary model. Various adversary models are available, such as the Dolev and Yoa (DY) model and the extended Canetti-Krawczyk model (eCK) model [19].

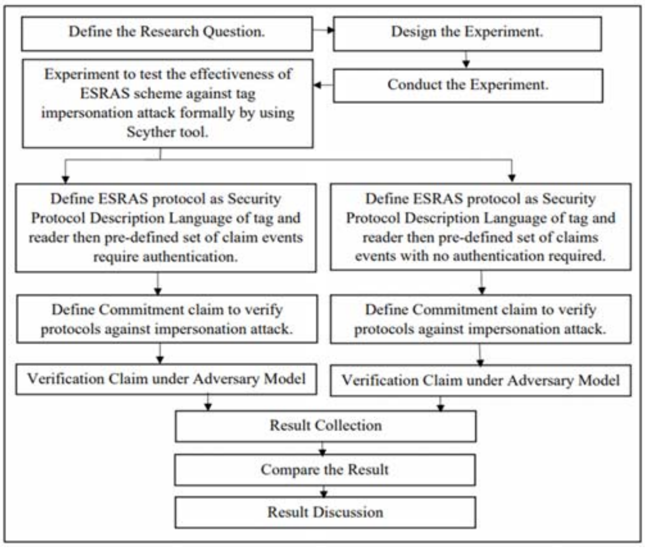

Define the Research Question.
Design the Experiment.
Experiment to test the effectiveness of ESRAS scheme against tag impersonation attack formally by using Scyther tool.
Conduct the Experiment.
Define ESRAS protocol as Security Protocol Description Language of tag and reader then pre-defined set of claim events require authentication.
Define ESRAS protocol as Security Protocol Description Language of tag and reader then pre-defined set of claims events with no authentication required.
Define Commitment claim to verify protocols against impersonation attack.
Define Commitment claim to verify protocols against impersonation attack.
Verification Claim under Adversary Model
Verification Claim under Adversary Model
Result Collection
Compare the Result
Result Discussion

Fig. 1 Research Methodology Structure

TABLE I
Notations and Their Descriptions [11]

Notations	Description
T, R, BS	RFID tag, reader, back-end server
ID	Static identification number for each RFID tag
IDS	Index pseudonym stored in tag and database
R_1, R_2	Pseudo random numbers generated at reader
K_1, K_2	Pre-shared secret keys of tags shared with the back-end server
$Rank(X, Y)$	Rank operation between strings X and Y
$Rank(X \text{ or } Y)$	Number of 1's presents in string X or Y
$nullity(X \text{ or } Y)$	Number of 0's presents in string X or Y
$Rot(X, Y)$	Circular left rotation of X by $rank(Y)$
lsb	Least significant bit
msb	Most significant bit
T_h	Threshold used to limit individual substring size
$\oplus$	Bitwise XOR operator
? =	Comparison operator

C. Experimental Design and Procedure

Experimental Process: This study provides the experimental protocol/process to verify the research hypothesis/research question. Scyther tool, installed on the Windows™ Operating System, provides a graphical user interface to analyze the effectiveness of the ESRAS scheme by implementing a simulated impersonation attack targeted at the tag and reader in RFID system (with/without authentication), followed by comparing the result status.

Scheme: The proposed scheme has used two circular left and right rotation operations, Rot (X, Y), where rotates string X by rank(y) mod L bits. L is the bit length of string X. X or Y rank is represented as the number of 1's appearing in string X or Y, and *nullity* is represented as the number of 0's appearing in the string. Rank operations are computationally complex, depending on the threshold value, Th. In the case of a low value for Th, the confusion will be more significant. Therefore, this study suggests a more considerable value for Th (> 5) [11].

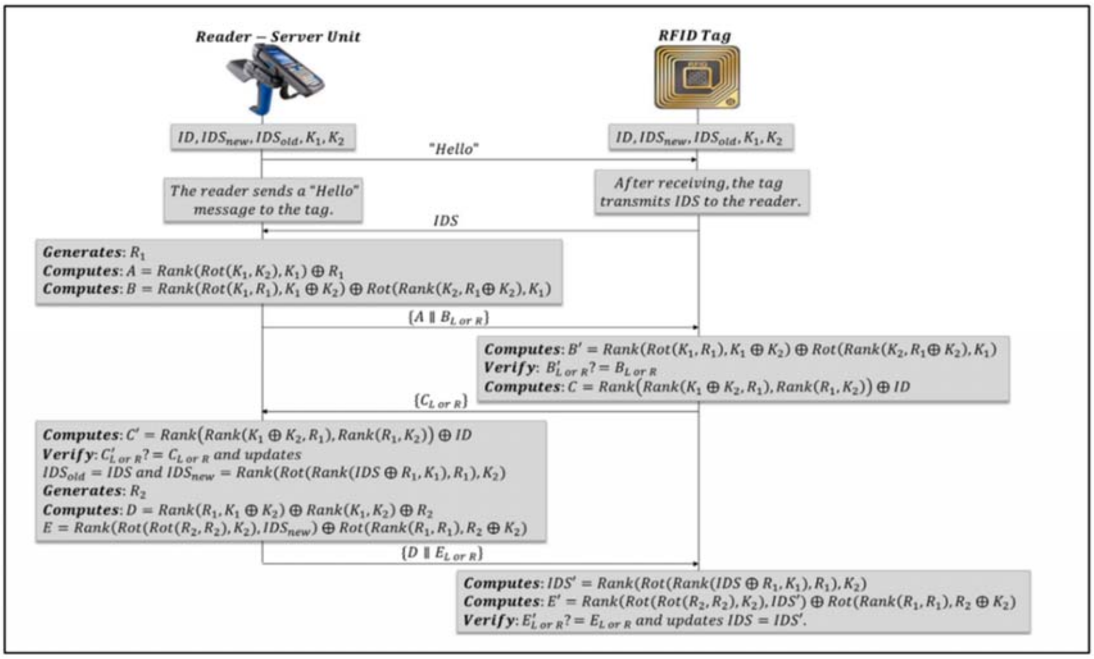

Reader – Server Unit
RFID Tag
$ID, IDS_{new}, IDS_{old}, K_1, K_2$
$ID, IDS_{new}, IDS_{old}, K_1, K_2$
"Hello"
The reader sends a "Hello" message to the tag.
After receiving, the tag transmits IDS to the reader.
IDS
Generates: R_1
Computes: $A = Rank(Rot(K_1, K_2), K_1) \oplus R_1$
Computes: $B = Rank(Rot(K_1, R_1), K_1 \oplus K_2) \oplus Rot(Rank(K_2, R_1 \oplus K_2), K_1)$
$\{A \parallel B_{L\ or\ R}\}$
Computes: $B' = Rank(Rot(K_1, R_1), K_1 \oplus K_2) \oplus Rot(Rank(K_2, R_1 \oplus K_2), K_1)$
Verify: $B'_{L\ or\ R} ?= B_{L\ or\ R}$
Computes: $C = Rank(Rank(K_1 \oplus K_2, R_1), Rank(R_1, K_2)) \oplus ID$
$\{C_{L\ or\ R}\}$
Computes: $C' = Rank(Rank(K_1 \oplus K_2, R_1), Rank(R_1, K_2)) \oplus ID$
Verify: $C'_{L\ or\ R} ?= C_{L\ or\ R}$ and updates
$IDS_{old} = IDS$ and $IDS_{new} = Rank(Rot(Rank(IDS \oplus R_1, K_1), R_1), K_2)$
Generates: R_2
Computes: $D = Rank(R_1, K_1 \oplus K_2) \oplus Rank(K_1, K_2) \oplus R_2$
$E = Rank(Rot(Rot(R_2, R_2), K_2), IDS_{new}) \oplus Rot(Rank(R_1, R_1), R_2 \oplus K_2)$
$\{D \parallel E_{L\ or\ R}\}$
Computes: $IDS' = Rank(Rot(Rank(IDS \oplus R_1, K_1), R_1), K_2)$
Computes: $E' = Rank(Rot(Rot(R_2, R_2), K_2), IDS') \oplus Rot(Rank(R_1, R_1), R_2 \oplus K_2)$
Verify: $E'_{L\ or\ R} ?= E_{L\ or\ R}$ and updates $IDS = IDS'$.

Fig. 2 Proposed ESRAS scheme [11]

The researcher [11] assumes that an adversary can imitate a genuine tag or reader and that messages exchanged among RFID components can be tampered with, modified, intercepted, added, and deleted by an adversary.

ESRAS' two authentication phases are illustrated in Fig. 2 [11].

- Initialization Phase

Each tag has a unique static identification number ID and an index identification number IDS.

There are two secret keys, K_1 and K_2, stored for each tag and a new IDS_{new} and old IDS_{old} pseudonym in the back-end server. A reader consists of two PRNGs (·).

- Authentication Phase

Step1. M1 → R initiates an authentication session by sending a ‘‘*Hello*’’ message to T.

Step2. M2 → T sends an index pseudonym IDS to R.

Step3. M3 → To retrieve the tags' secrets from the database, R uses the IDS. The reader generates a pseudo-random number R1, which is used to compute A and B when a match is found in the database.

Step4. T authenticates R as a legitimate reader and calculates the response message C, CR and CL.

Step5. M5 → To authenticate T as a legitimate tag in the database, the reader calculates a local value of C ′ and verifies whether C ′ L or R = CL or R. If so, the reader updates T's index pseudonyms IDS_{old} and IDS_{new}. Subsequently, the reader generates L-bit pseudo-random number R2 and sends response messages to the tag.

Step6. The tag extracts $R2$ from D, computes IDS′ and the local value of E', and subsequently verifies whether $E\ '\ L\ or\ R\ ?= EL\ or\ R$, as a result of authentication. Finally, the tag updates the reader's pseudonym in the index as a legitimate reader.

Tag Impersonation Attack: A security threat which can impersonate a legitimate tag without any knowledge of the secret values ID and K. To impersonate the tag T, the attacker A follows the steps listed below and as depicted in Fig. 3.

Step1. A eavesdrops one protocol execution between the reader R and T, and stores all transferred values between R and T. Those values include R1, A and B.

Step2. Upon the next protocol round, when R sends $Hello$ and $R1$ to the tag, the attacker responds with B′ or B to compute C, then authenticates R as a legitimate reader.

Step3. The reader sends C′ to the attacker and verifies the authenticate T as a legitimate tag. Hence, following the above attack, the reader authenticates the adversary as a legitimate tag.

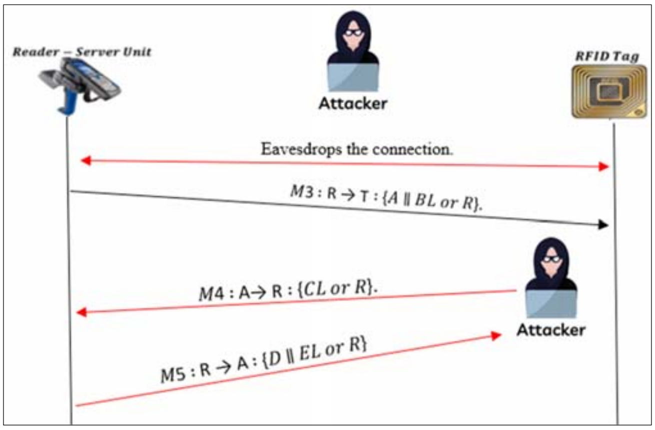


Fig. 3 Tag Impersonation Attack

Verification of Experimental Result: In [11], the study claims that the tag is anonymous to the attacker because IDS and the shared secret keys $K1$ and $K2$ are updated on every successful execution of the protocol. The update process also includes random numbers. In this section, our study evaluated the security of the proposed scheme for resisting a tag impersonation attack as a formal analysis employing the Scyther tool.

Scyther is a tool for verifying and falsifying security protocols. It is an automated simulation tool with a graphical user interface incorporating the Scyther command line tool and a Python scripting interface. The Scyther tool accepts protocol descriptions and optimal parameters as inputs and produces a summary report and graph for each attack. Security Protocol Description Language is the language used to describe protocols [20].

There are three methods to use the tool: to verify the security claims stated in the protocol description, to generate and verify security claims for a protocol automatically, and to analyze the protocol by performing a complete characterization [21].

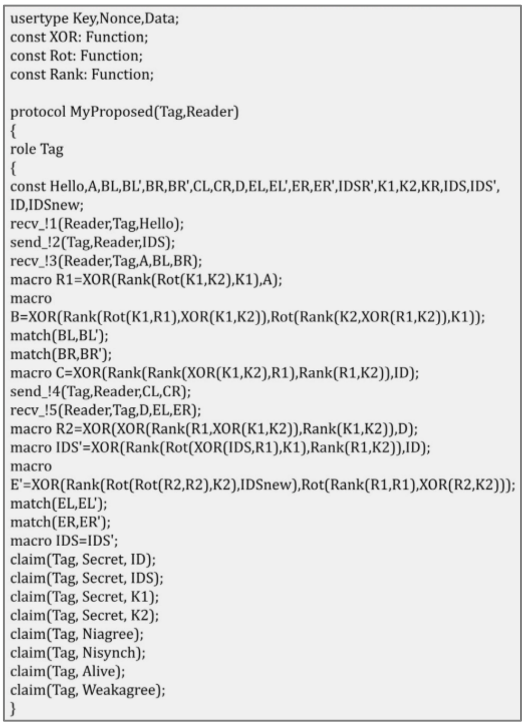
- *Verification claim:* Scyther verifies or falsifies the security properties of the system [20].
- *Automatic claims:* Scyther can automatically generate security claims if the protocol specification contains no security claims. At the end of each role authentication, 19 claims are added, claiming that the supposed communication partners must have adhered to the protocol as anticipated. Secrecy claims are added for all locally generated variables and values (nonces). Scyther identically analyzes this enhanced protocol description as in the preceding example. Consequently, users can rapidly assess the properties of a protocol [21].
- *Characterization:* It is possible to characterize each protocol role. Following Scyther analysis of the protocol, it provides a finite representation of all traces that contain an execution of the protocol role [20].

The ESRAS protocol is described in this study as SPDL (Security Protocol Description Language) to write the tag (Fig. 4) and the reader (Fig. 5) where a set of claim events in Scyther needs to be defined.

In addition, a sequence of events can include sending or receiving the data. The ESRAS Scheme tag and reader roles are defined in two ways:

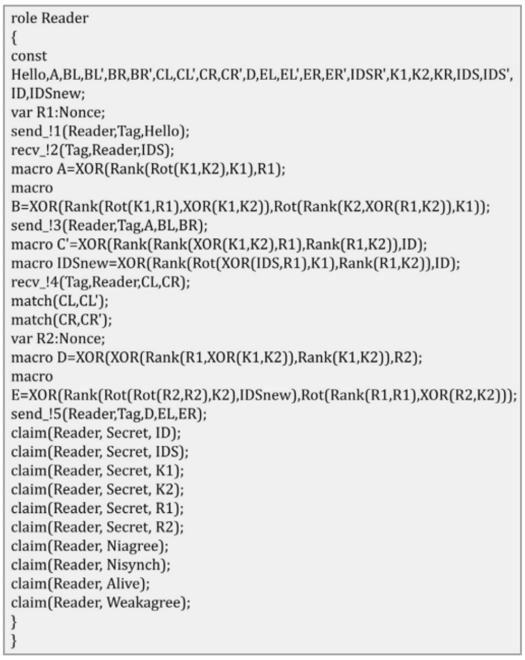
- *With Authentication:* The authentication phases define the claim to specify security requirements. For instance, Alive is a form of authentication, Commitment to verify protocols against impersonation attacks, Nisynch to ensure communication between tag and reader, and Secret, which means unknown to an adversary.
- *Without Authentication:* These phases define the claim to specify the security requirement, claim commitment, Nisynch and Secret.

Consequently, the analysis was to be performed under two differing adversary models. The result was compared to ability levels for resisting tag impersonation attacks.



```
usertype Key,Nonce,Data;
const XOR: Function;
const Rot: Function;
const Rank: Function;

protocol MyProposed(Tag,Reader)
{
role Tag
{
const Hello,A,BL,BL',BR,BR',CL,CR,D,EL,EL',ER,ER',IDSR',K1,K2,KR,IDS,IDS',
ID,IDSnew;
recv_!1(Reader,Tag,Hello);
send_!2(Tag,Reader,IDS);
recv_!3(Reader,Tag,A,BL,BR);
macro R1=XOR(Rank(Rot(K1,K2),K1),A);
macro
B=XOR(Rank(Rot(K1,R1),XOR(K1,K2)),Rot(Rank(K2,XOR(R1,K2)),K1));
match(BL,BL');
match(BR,BR');
macro C=XOR(Rank(Rank(XOR(K1,K2),R1),Rank(R1,K2)),ID);
send_!4(Tag,Reader,CL,CR);
recv_!5(Reader,Tag,D,EL,ER);
macro R2=XOR(XOR(Rank(R1,XOR(K1,K2)),Rank(K1,K2)),D);
macro IDS'=XOR(Rank(Rot(XOR(IDS,R1),K1),Rank(R1,K2)),ID);
macro
E'=XOR(Rank(Rot(Rot(R2,R2),K2),IDSnew),Rot(Rank(R1,R1),XOR(R2,K2)));
match(EL,EL');
match(ER,ER');
macro IDS=IDS';
claim(Tag, Secret, ID);
claim(Tag, Secret, IDS);
claim(Tag, Secret, K1);
claim(Tag, Secret, K2);
claim(Tag, Niagree);
claim(Tag, Nisynch);
claim(Tag, Alive);
claim(Tag, Weakagree);
}
```

Fig. 4 SPDL specification for the Tag role [11]



```
role Reader
{
const
Hello,A,BL,BL',BR,BR',CL,CL',CR,CR',D,EL,EL',ER,ER',IDSR',K1,K2,KR,IDS,IDS',
ID,IDSnew;
var R1:Nonce;
send_!1(Reader,Tag,Hello);
recv_!2(Tag,Reader,IDS);
macro A=XOR(Rank(Rot(K1,K2),K1),R1);
macro
B=XOR(Rank(Rot(K1,R1),XOR(K1,K2)),Rot(Rank(K2,XOR(R1,K2)),K1));
send_!3(Reader,Tag,A,BL,BR);
macro C'=XOR(Rank(Rank(XOR(K1,K2),R1),Rank(R1,K2)),ID);
macro IDSnew=XOR(Rank(Rot(XOR(IDS,R1),K1),Rank(R1,K2)),ID);
recv_!4(Tag,Reader,CL,CR);
match(CL,CL');
match(CR,CR');
var R2:Nonce;
macro D=XOR(XOR(Rank(R1,XOR(K1,K2)),Rank(K1,K2)),R2);
macro
E=XOR(Rank(Rot(Rot(R2,R2),K2),IDSnew),Rot(Rank(R1,R1),XOR(R2,K2)));
send_!5(Reader,Tag,D,EL,ER);
claim(Reader, Secret, ID);
claim(Reader, Secret, IDS);
claim(Reader, Secret, K1);
claim(Reader, Secret, K2);
claim(Reader, Secret, R1);
claim(Reader, Secret, R2);
claim(Reader, Niagree);
claim(Reader, Nisynch);
claim(Reader, Alive);
claim(Reader, Weakagree);
}
}
```

Fig. 5 SPDL specification for the Reader role [11]

D. *Experimental Result*

Analytical results were comparatively assessed between with- and without-authentication methods. The Scyther tool displayed the result status for each pre-defined set of claim events. If the result status is "ok" and no possible attacks within bounds exist, this would reveal that the proposed ESRAS scheme is highly resistant to tag impersonation attacks.

This experiment is reproducible by using differing security analysis protocol tools or implementing another kind of non-tested attack dose. Moreover, the experimental result is reproducible to validate that the ESRAS scheme is not vulnerable to impersonation attacks in the future, or in the case that the result of this study would not be resistant to tag impersonation attacks, to improve the ESRAS scheme.

IV. Implementation

This section describes the implementation of the study to evaluate the effectiveness of ESRAS against tag impersonation attack through formally employing the Scyther tool.

The impersonation attack's concept is that the attacker impersonates a legitimate tag by using the leaked sensitive data or by performing replay attacks. The experiment has three dimensions to assess ESRAS scheme strength:

- Define ESRAS protocol as SPDL of tag and reader.
- Assess resistance of the ESRAS protocol against tag impersonation attack by using claim commitment with/ without authentication claim.
- Verification of security protocol with/without authentication claim under the extended Canetti–Krawczyk (eCK) adversary model and under the Dolev and Yoa (DY) adversary model.

A. Installation

The experimental configuration is simulated using Scyther installed on the Windows™ operating system by following three requirements:

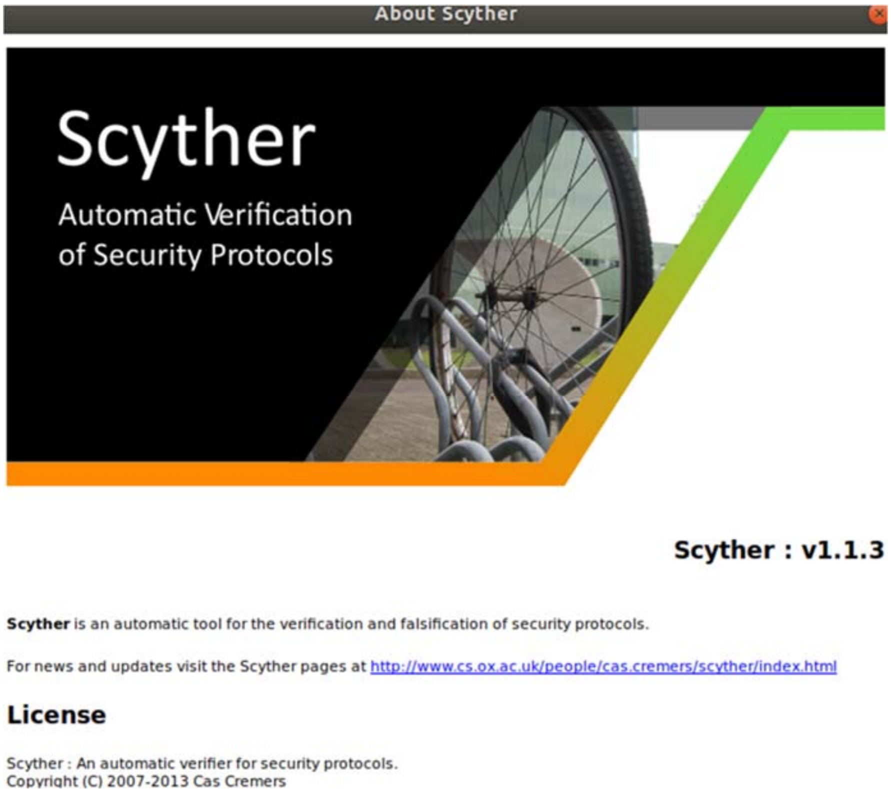
- Install the GraphViz library from http://www.graphviz.org/
- Install Python, since Scyther does not support Python 3. Therefore, selecting the latest production release of Python 2 is recommended.
- Install wxPython libraries from http://downloads.sourceforge.net/wxpython/wxPython2.8-win32-unicode-2.8.12.1-py27.exe

B. Run Scyther Tool

Scyther is initiated by executing the scyther-gui.py program in the Scyther directory. The program launches two windows: the main window (in which files are edited), and the 'about' window, which depicts information on the tool. The main window should appear as in Fig. 6.

C. Scyther Tool Verification

Scyther is a widely accepted tool that automatically verifies the security of cryptographic protocols using the Dolev-Yao adversary model (DY) [16]. Communication parties in the DY threat model are considered honest and are capable of running multiple sessions with each other. The communication channel is completely insecure and completely under the adversary's control [19]. Consequently, an adversary can eavesdrop, tamper, delete, or modify messages on the Scyther communication channel and learn from them.



Scyther : v1.1.3

Scyther is an automatic tool for the verification and falsification of security protocols.

For news and updates visit the Scyther pages at http://www.cs.ox.ac.uk/people/cas.cremers/scyther/index.html

License

Scyther : An automatic verifier for security protocols.
Copyright (C) 2007-2013 Cas Cremers

Fig. 6 Scyther main window

Concerning authentication and key agreement protocols, the extended Canetti-Krawczyk (eCK) model is the most widely accepted. In this adversary model, the attacker can compromise the PRNG and obtain access to the secret randomness of the session.

Furthermore, it is assumed that an adversary can compromise and gain access to the session. It is also possible for an attacker to gain access to long-term keys [19]. Overall, eCK security models are widely used as security arguments for authenticated key exchange protocols to detect confidential data leakage, including long-term private keys and session-specific state information [22]. Therefore, when analyzing protocols, it is not required to formalize an adversary's capabilities [23]. The Scyther tool is designed for protocol validation, presentation, analysis, specification, and derivation. By defining protocol behavior classes, Scyther allows security problems to be identified through straightforward formalization and verification [24]. Scyther simulates the ESRAS protocol using a syntax similar to C/JAVA programming (albeit case-insensitive). Roles are defined as a sequence of events, including declarations, which consist of events representing data transmission and reception, as well as claims [23]. In protocol verification claims, Scyther can verify or falsify security attributes. During the protocol verification process, Scyther generates attack graphs for unsafe protocols and displays one attack graph for each claim [24]. It consequently performs the necessary settings in the Scyther tool's settings, such as setting a maximum number of runs, determining search pruning, and determining the maximum number of patterns per claim [16]. This can be changed in the Settings tab of the main window [23].

D. Scyther Settings

The parameters employed in this study are listed in Table II.

TABLE II
SCYTHER PARAMETER SETTINGS FOR THIS STUDY

Verification parameters	Scyther Parameters
Maximum number of Runs	5
Matching Types	Typed Matching
Search pruning	Find best attack
Maximum number of patterns per claim	10

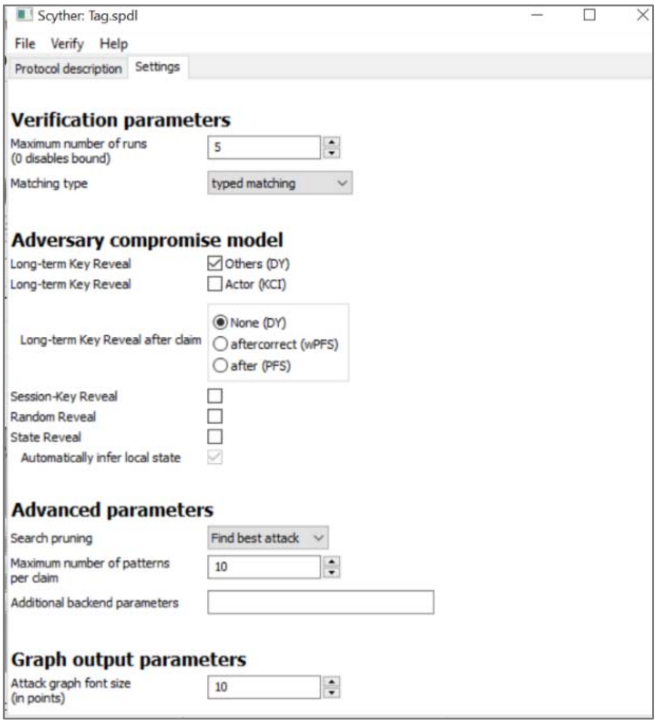
1. Case 1. The eCK Adversary Model

In evaluating the ESRAS scheme, the eCK adversary model is considered a stricter and more relevant adversary model [19]. Consequently, the ESARS scheme was formally validated under the eCK adversary model, as shown in Fig. 7.

2. Case 2. Doled and Yoa Adversary Models

The ESRAS scheme was also formally validated under Dolev and Yoa (DY) adversary, as shown in Fig. 8. This adversary model uses a long-term key reveal LKR after correction of wPFS. Weak perfect forward security (wPFS) assumes that the adversary is not actively involved in selecting messages during a session [25]. This ensures that the adversary cannot insert fake messages and learn the key to the involved agents during protocol execution. Protocols that satisfy secrecy properties relating to adversaries that can use LKRafter-correct are deemed to satisfy weak perfect forward secrecy (wPFS) [26].

To run the protocol code, click ‘verify’, followed by ‘verify protocol’, as shown in Fig. 9. The Scyther tool evaluates the security feature as ‘OK’ if it cannot find the attack, and if it finds the attack, it fails that feature and displays the flow chart for the attack scenario.



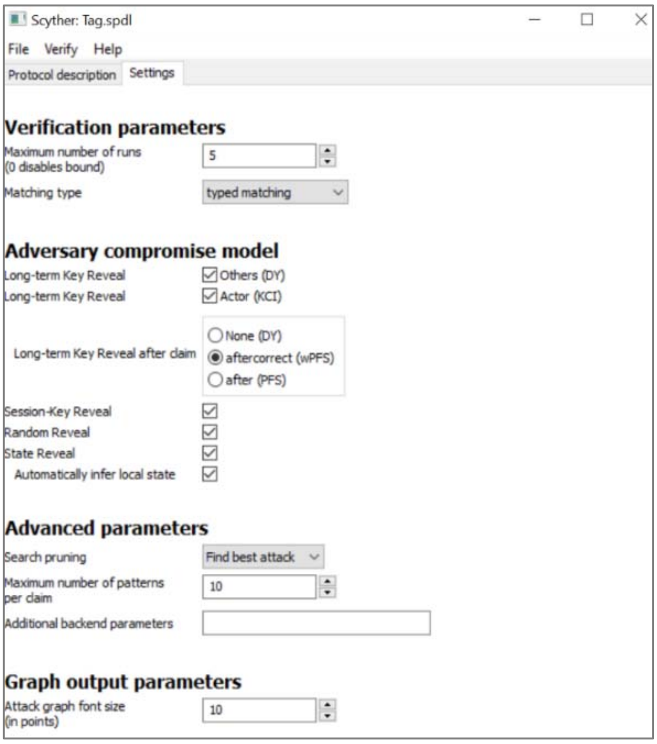
Verification parameters
Maximum number of runs (0 disables bound): 5
Matching type: typed matching

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): 10

Fig. 7 Scyther settings for the eCK Adversary Model



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): 10

Fig. 8 Scyther settings for the DY Adversary Model

E. Assessing ESRAS Scheme with Authentication

1. Role Definitions

Role definitions includes sequences of events, declarations, claim, send, or receive events that define the behaviors and interactions within a system or framework. The communication behaviors of roles are defined within the curly brackets following the corresponding role Tag and role Reader commands. These roles have the Reader role (Fig. 10) and Tag role (Fig. 11), respectively.

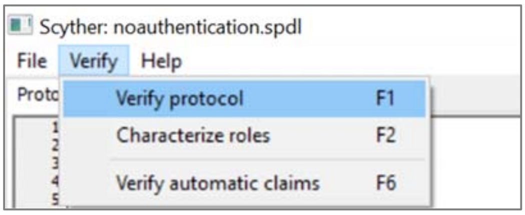
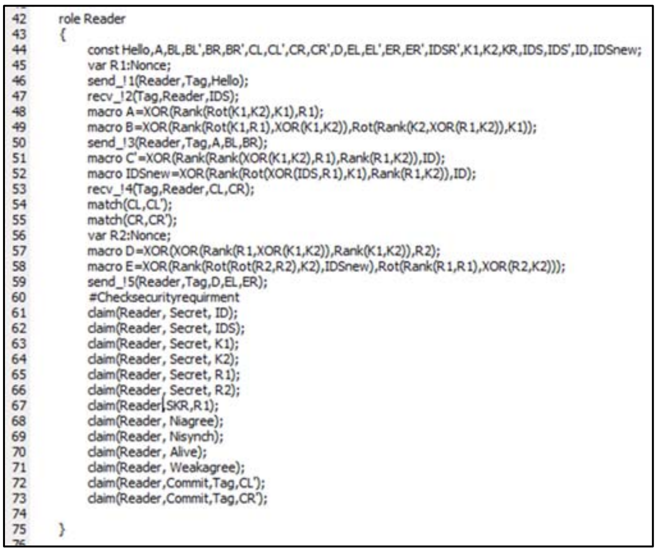


Fig. 9 Verification protocol

2. Claim Events

Security requirements of the protocol were added, using the claim to specify security requirements for both roles (Tag and Reader), share the secret goal over the secret values to the adversary ID, IDS, and K1 (Fig. 12), and secret values of reader ID, IDS, K1, K2, R1, and R2 (Fig. 13).

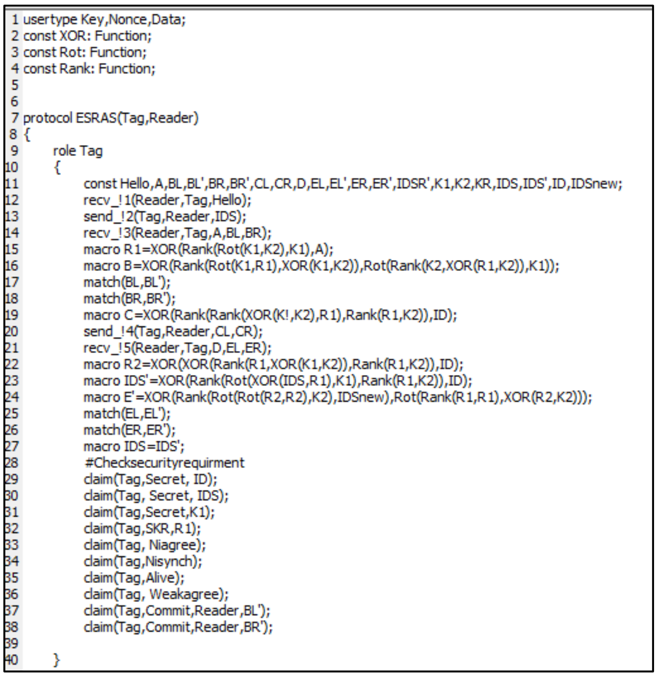
Alive, *Nisynch*, *Weakagree*, and *commitment* were also (as required) specified in the claim. As a form of authentication, Alive is intended to ensure that the intended communication party has completed certain activities. The term Commitment refers to the promise made by a communication partner to another party as well as the use of commitment to verify protocols against impersonation attacks.



```
    role Reader
    {
        const Hello,A,BL,BL',BR,BR',CL,CL',CR,CR',D,EL,EL',ER,ER',IDSR',K1,K2,KR,IDS,IDS',ID,IDSnew;
        var R1:Nonce;
        send_!1(Reader,Tag,Hello);
        recv_!2(Tag,Reader,IDS);
        macro A=XOR(Rank(Rot(K1,K2),K1),R1);
        macro B=XOR(Rank(Rot(K1,R1),XOR(K1,K2)),Rot(Rank(K2,XOR(R1,K2)),K1));
        send_!3(Reader,Tag,A,BL,BR);
        macro C'=XOR(Rank(Rank(XOR(K1,K2),R1),Rank(R1,K2)),ID);
        macro IDSnew=XOR(Rank(Rot(XOR(IDS,R1),K1),Rank(R1,K2)),ID);
        recv_!4(Tag,Reader,CL,CR);
        match(CL,CL');
        match(CR,CR');
        var R2:Nonce;
        macro D=XOR(XOR(Rank(R1,XOR(K1,K2)),Rank(K1,K2)),R2);
        macro E=XOR(Rank(Rot(Rot(R2,R2),K2),IDSnew),Rot(Rank(R1,R1),XOR(R2,K2)));
        send_!5(Reader,Tag,D,EL,ER);
        #Checksecurityrequirment
        claim(Reader, Secret, ID);
        claim(Reader, Secret, IDS);
        claim(Reader, Secret, K1);
        claim(Reader, Secret, K2);
        claim(Reader, Secret, R1);
        claim(Reader, Secret, R2);
        claim(Reader,SKR,R1);
        claim(Reader, Niagree);
        claim(Reader, Nisynch);
        claim(Reader, Alive);
        claim(Reader, Weakagree);
        claim(Reader,Commit,Tag,CL');
        claim(Reader,Commit,Tag,CR');

    }
```

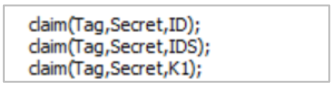
Fig. 10 Reader role

Nisynch refers to messages sent by one communication partner (Tag) and received by another (Reader) [23]. The *Weakagree* protocol is secure against impersonation attacks (see Fig. 14) [27].



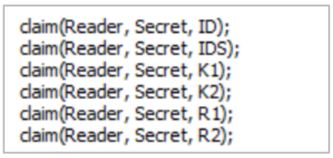
```
usertype Key,Nonce,Data;
const XOR: Function;
const Rot: Function;
const Rank: Function;


protocol ESRAS(Tag,Reader)
{
    role Tag
    {
        const Hello,A,BL,BL',BR,BR',CL,CR,D,EL,EL',ER,ER',IDSR',K1,K2,KR,IDS,IDS',ID,IDSnew;
        recv_!1(Reader,Tag,Hello);
        send_!2(Tag,Reader,IDS);
        recv_!3(Reader,Tag,A,BL,BR);
        macro R1=XOR(Rank(Rot(K1,K2),K1),A);
        macro B=XOR(Rank(Rot(K1,R1),XOR(K1,K2)),Rot(Rank(K2,XOR(R1,K2)),K1));
        match(BL,BL');
        match(BR,BR');
        macro C=XOR(Rank(Rank(XOR(K!,K2),R1),Rank(R1,K2)),ID);
        send_!4(Tag,Reader,CL,CR);
        recv_!5(Reader,Tag,D,EL,ER);
        macro R2=XOR(XOR(Rank(R1,XOR(K1,K2)),Rank(R1,K2)),ID);
        macro IDS'=XOR(Rank(Rot(XOR(IDS,R1),K1),Rank(R1,K2)),ID);
        macro E'=XOR(Rank(Rot(Rot(R2,R2),K2),IDSnew),Rot(Rank(R1,R1),XOR(R2,K2)));
        match(EL,EL');
        match(ER,ER');
        macro IDS=IDS';
        #Checksecurityrequirment
        claim(Tag,Secret, ID);
        claim(Tag, Secret, IDS);
        claim(Tag,Secret,K1);
        claim(Tag,SKR,R1);
        claim(Tag, Niagree);
        claim(Tag,Nisynch);
        claim(Tag,Alive);
        claim(Tag, Weakagree);
        claim(Tag,Commit,Reader,BL');
        claim(Tag,Commit,Reader,BR');

    }
```

Fig. 11 Tag role



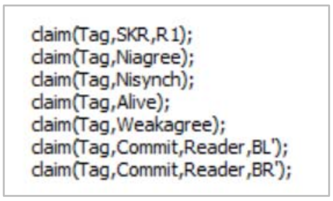
```
claim(Tag,Secret,ID);
claim(Tag,Secret,IDS);
claim(Tag,Secret,K1);
```

Fig. 12 Claim Secret for Tag



```
claim(Reader, Secret, ID);
claim(Reader, Secret, IDS);
claim(Reader, Secret, K1);
claim(Reader, Secret, K2);
claim(Reader, Secret, R1);
claim(Reader, Secret, R2);
```

Fig. 13 Claim secret for Reader



```
claim(Tag,SKR,R1);
claim(Tag,Niagree);
claim(Tag,Nisynch);
claim(Tag,Alive);
claim(Tag,Weakagree);
claim(Tag,Commit,Reader,BL');
claim(Tag,Commit,Reader,BR');
```

Fig. 14 Claim security requirement

3. Formalization of Security Requirements

There are two communication parties, Tag and Reader. ESRAS protocols are claimed to satisfy the following security requirements:

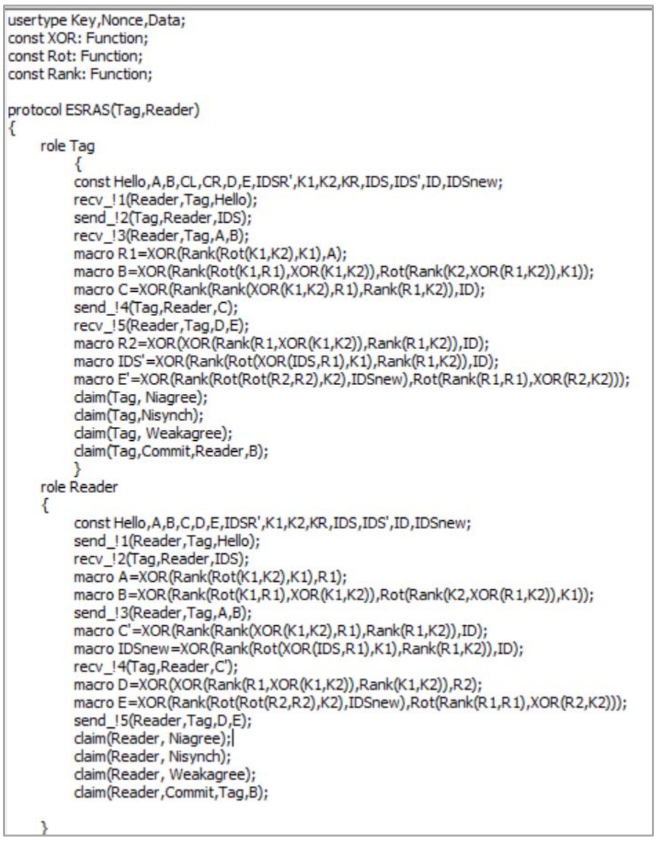
a. *Mutual authentication:* Communication parties use authentication to exchange messages with the intended recipients. Mutual authentication refers to authentication achieved by both parties in a communication. An *Alive* claim must confirm that the communication is deriving from the intended party rather than another [23].

b. *Secure against impersonation attacks:* Impersonating a legitimate communication party is known as Tag Impersonation. The security requirement can be derived from mutual authentication, and *Commitment* claims are used to verify protocols against tag impersonation attacks [23].

c. *Secure against passive adversaries:* An adversary who eavesdrops on a communication channel analyses these messages and attempts to learn as much as possible. A passive adversary has limited capabilities compared to an active adversary. It cannot delete or insert messages into the communication channel; its primary goal is to gain helpful information from messages that have been eavesdropped. The most useful information in the ESRAS protocol of Type tag is R1, which can be expressed using claim (Tag, SKR, R1) [23].

F. Assessing ESRAS Scheme with No Authentication

This section describes the ESRAS protocol as SPDL to initiate a role without a match event. These roles have Tag and Reader roles, as shown in Fig. 15. Moreover, a claim is employed to specify security requirements: *Nisynch, Niagree, Weakagree* and *Commitment*.



```
usertype Key,Nonce,Data;
const XOR: Function;
const Rot: Function;
const Rank: Function;

protocol ESRAS(Tag,Reader)
{
      role Tag
            {
            const Hello,A,B,CL,CR,D,E,IDSR',K1,K2,KR,IDS,IDS',ID,IDSnew;
            recv_!1(Reader,Tag,Hello);
            send_!2(Tag,Reader,IDS);
            recv_!3(Reader,Tag,A,B);
            macro R1=XOR(Rank(Rot(K1,K2),K1),A);
            macro B=XOR(Rank(Rot(K1,R1),XOR(K1,K2)),Rot(Rank(K2,XOR(R1,K2)),K1));
            macro C=XOR(Rank(Rank(XOR(K1,K2),R1),Rank(R1,K2)),ID);
            send_!4(Tag,Reader,C);
            recv_!5(Reader,Tag,D,E);
            macro R2=XOR(XOR(Rank(R1,XOR(K1,K2)),Rank(R1,K2)),ID);
            macro IDS'=XOR(Rank(Rot(XOR(IDS,R1),K1),Rank(R1,K2)),ID);
            macro E'=XOR(Rank(Rot(Rot(R2,R2),K2),IDSnew),Rot(Rank(R1,R1),XOR(R2,K2)));
            claim(Tag, Niagree);
            claim(Tag,Nisynch);
            claim(Tag, Weakagree);
            claim(Tag,Commit,Reader,B);
            }
      role Reader
      {
            const Hello,A,B,C,D,E,IDSR',K1,K2,KR,IDS,IDS',ID,IDSnew;
            send_!1(Reader,Tag,Hello);
            recv_!2(Tag,Reader,IDS);
            macro A=XOR(Rank(Rot(K1,K2),K1),R1);
            macro B=XOR(Rank(Rot(K1,R1),XOR(K1,K2)),Rot(Rank(K2,XOR(R1,K2)),K1));
            send_!3(Reader,Tag,A,B);
            macro C'=XOR(Rank(Rank(XOR(K1,K2),R1),Rank(R1,K2)),ID);
            macro IDSnew=XOR(Rank(Rot(XOR(IDS,R1),K1),Rank(R1,K2)),ID);
            recv_!4(Tag,Reader,C');
            macro D=XOR(XOR(Rank(R1,XOR(K1,K2)),Rank(K1,K2)),R2);
            macro E=XOR(Rank(Rot(Rot(R2,R2),K2),IDSnew),Rot(Rank(R1,R1),XOR(R2,K2)));
            send_!5(Reader,Tag,D,E);
            claim(Reader, Niagree);
            claim(Reader, Nisynch);
            claim(Reader, Weakagree);
            claim(Reader,Commit,Tag,B);

      }
```

Fig. 15 SPDL specification for Tag and Reader role with no Authentication

V. Results

A. ESRAS Scheme with Authentication

- *Case 1. Verification Under the eCK adversary model:* According to the specified security requirements (Fig. 15), and Scyther setting (Fig. 8), the Scyther verification result is depicted in Fig. 16, which indicated that there was no attack possible within bounds. Thus, the ESRAS protocol is deemed safe under tag impersonation attacks.
- *Case 2. Verification under Dolev and Yoa adversary models:* According to specified security requirements (Fig. 15) and the Scyther setting (Fig. 8), the Scyther verification result is depicted in Fig. 17, indicating that no attack was possible within bounds. Consequently, the ESRAS protocol is deemed safe under a tag impersonation attack.

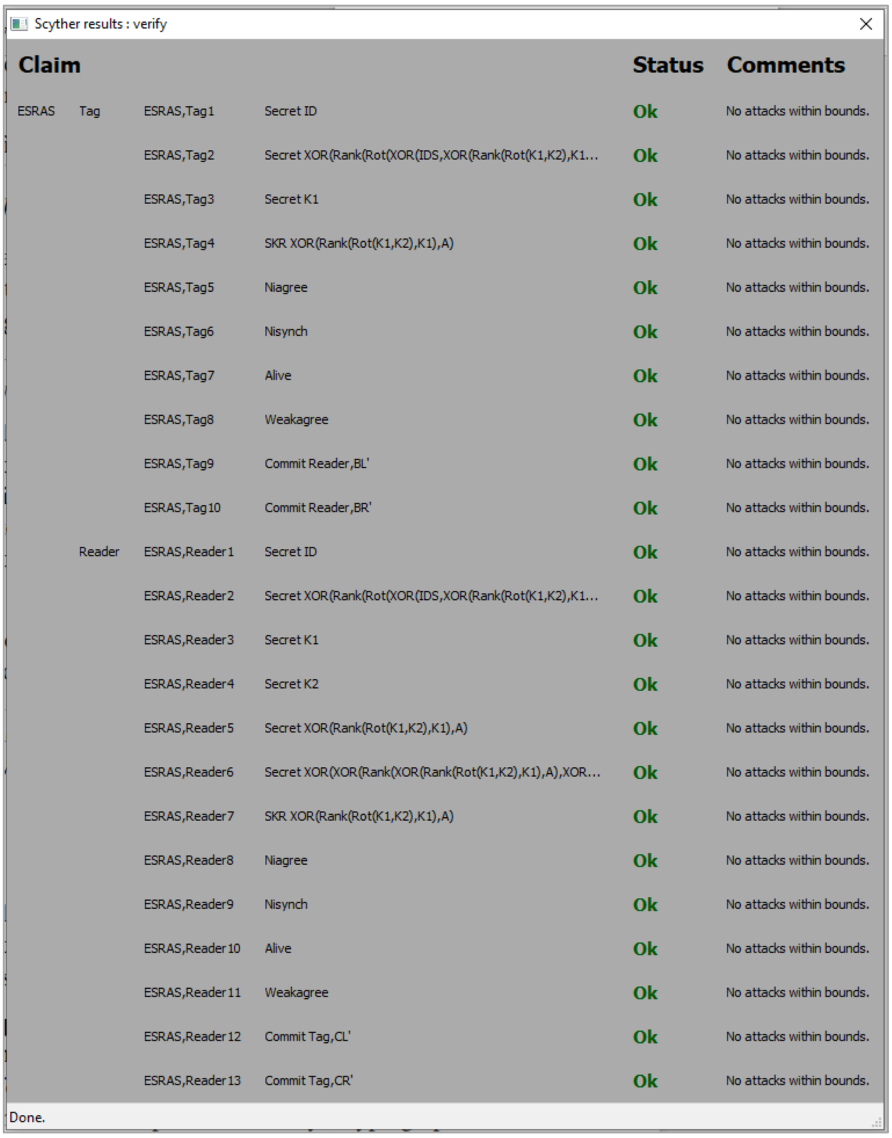
B. ESRAS Scheme without Authentication

- *Case 1. Under the eCK adversary model:* According to specified security requirements (Fig. 15) and Scyther setting (Fig. 8), the Scyther verification result is depicted in Fig. 18, indicating that no attack was possible within bounds. Thus, the ESRAS protocol was deemed safe under tag impersonation attacks.
- *Case 2. Dolev and Yoa adversary models:* According to specified security requirements (Fig. 15) and the Scyther settings (Fig. 8), the Scyther verification result is shown in Fig. 19, indicating that the ESRAS scheme under the Dolev and Yoa adversary models is not resilient to tag impersonation attack.

VI. Discussion

This study assumed that ESRAS schemes could resist the impersonation attack, and the research contribution was to evaluate the effectiveness and resistance of the ESRAS Scheme against tag impersonation attacks in two different approaches (with/without authentication), formally using a security analysis protocol tool, and comparatively analyzing outcomes. During the implementation process, the study defined parameters and applied different adversary models (with authentication/without authentication). With the authentication process, the study defined a set of claim events requiring authentication and a Commitment claim to verify protocols against impersonation attacks. In this case, under the eCK and the DY adversary models, the ESRAS Scheme can resist tag impersonation attacks. This means that the protocol (with an authentication process) effectively prevents active and passive attacks. The authentication process renders it much more challenging for an attacker to perform an attack, and it provides a good level of security for most applications.

Without authentication, in the case of an eCK adversary model, the ESRAS Scheme also resists tag impersonation attacks. This is the case since the eCK adversary model utilizes a challenge-response mechanism to verify a tag's identity. A challenge-response mechanism involves the reader sending a challenge to the tag, and the tag responds with a response calculated using the tag's private key. The reader can verify the response using the tag’s public key. A challenge-response mechanism renders it more challenging for an attacker to impersonate a legitimate tag. It would be necessary the attacker must know the tag's private key to generate a valid response [28].



Claim				Status	Comments
ESRAS	Tag	ESRAS,Tag1	Secret ID	Ok	No attacks within bounds.
		ESRAS,Tag2	Secret XOR(Rank(Rot(XOR(IDS,XOR(Rank(Rot(K1,K2),K1...	Ok	No attacks within bounds.
		ESRAS,Tag3	Secret K1	Ok	No attacks within bounds.
		ESRAS,Tag4	SKR XOR(Rank(Rot(K1,K2),K1),A)	Ok	No attacks within bounds.
		ESRAS,Tag5	Niagree	Ok	No attacks within bounds.
		ESRAS,Tag6	Nisynch	Ok	No attacks within bounds.
		ESRAS,Tag7	Alive	Ok	No attacks within bounds.
		ESRAS,Tag8	Weakagree	Ok	No attacks within bounds.
		ESRAS,Tag9	Commit Reader,BL'	Ok	No attacks within bounds.
		ESRAS,Tag10	Commit Reader,BR'	Ok	No attacks within bounds.
	Reader	ESRAS,Reader1	Secret ID	Ok	No attacks within bounds.
		ESRAS,Reader2	Secret XOR(Rank(Rot(XOR(IDS,XOR(Rank(Rot(K1,K2),K1...	Ok	No attacks within bounds.
		ESRAS,Reader3	Secret K1	Ok	No attacks within bounds.
		ESRAS,Reader4	Secret K2	Ok	No attacks within bounds.
		ESRAS,Reader5	Secret XOR(Rank(Rot(K1,K2),K1),A)	Ok	No attacks within bounds.
		ESRAS,Reader6	Secret XOR(XOR(Rank(XOR(Rank(Rot(K1,K2),K1),A),XOR...	Ok	No attacks within bounds.
		ESRAS,Reader7	SKR XOR(Rank(Rot(K1,K2),K1),A)	Ok	No attacks within bounds.
		ESRAS,Reader8	Niagree	Ok	No attacks within bounds.
		ESRAS,Reader9	Nisynch	Ok	No attacks within bounds.
		ESRAS,Reader10	Alive	Ok	No attacks within bounds.
		ESRAS,Reader11	Weakagree	Ok	No attacks within bounds.
		ESRAS,Reader12	Commit Tag,CL'	Ok	No attacks within bounds.
		ESRAS,Reader13	Commit Tag,CR'	Ok	No attacks within bounds.

Done.

Fig. 16 Scyther verification results under eCK Adversary Model with Authentication

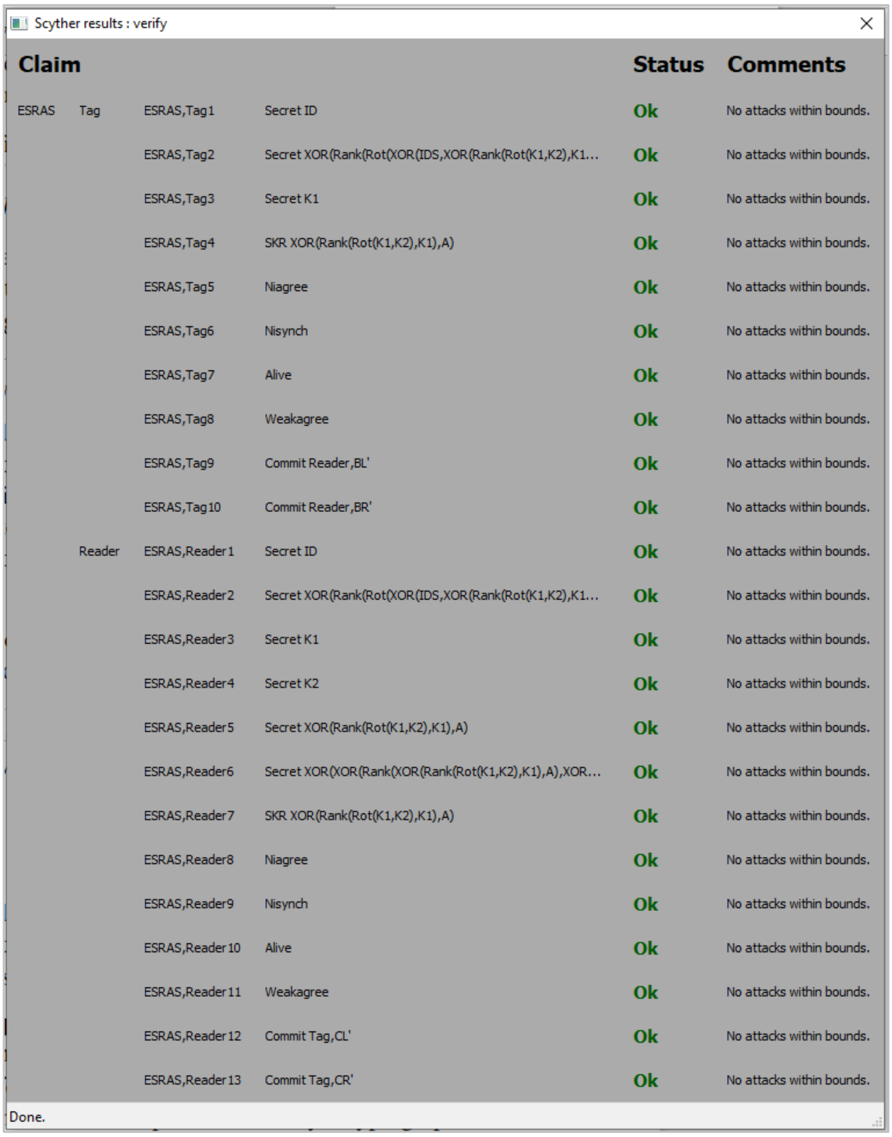
However, under the DY adversary model, this system is not resistant to tag impersonation attacks. Therefore, several attacks - including tag impersonation attacks - are possible with this protocol. There is no authentication mechanism used in the protocol to verify tag identity. An attacker can impersonate a legitimate tag and access its associated resources. Consequently, an attacker can send a message to the reader, claiming to be from a legitimate tag, and the reader will accept it without verifying the tag's identity. Furthermore, the protocol does not encrypt the messages sent between the tag and the reader. An attacker can eavesdrop on the messages between the tag and reader and determine the values of the cryptographic keys used to protect the protocol.

According to the Scyther analysis, the ESRAS protocol is insecure under the DY adversary model. The protocol is, therefore, vulnerable to several attacks, including tag impersonation attacks.

Furthermore, when Scyther reports 'No attack within bounds', no attacks involving five runs or less exist. However, there might exist attacks that involve six runs or more. For some protocols, increasing the maximum number of runs can lead to complete results (i.e., finding an attack or being sure there is no attack). However, for other protocols, the result will always be 'No attack within bounds'. Note that the verification time typically grows exponentially concerning the maximum number of runs [23].

A. Comparison Analysis

In this section we compare the proposed protocol [11] for different attacks against [29]-[31] ultra-lightweight protocols. After reading and comparing the research papers we will get results that are shown in Table III, together with our protocol, which has greater strength than these protocols.

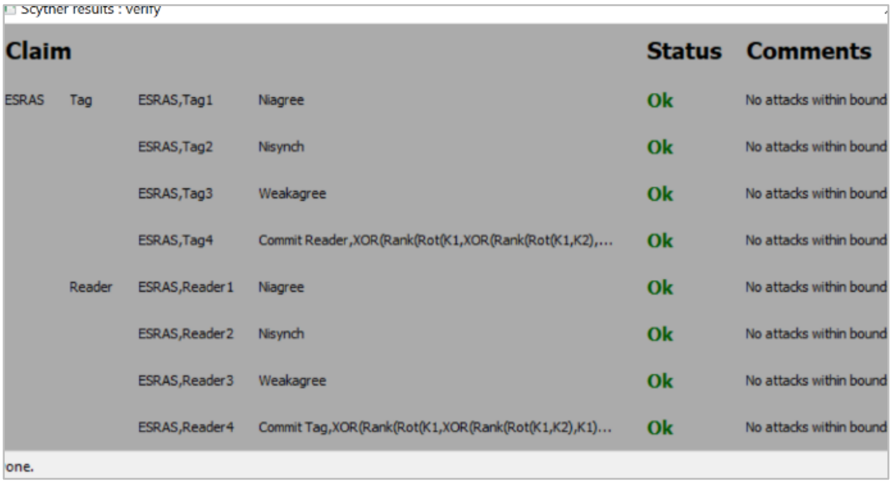


Scyther results : verify

Claim				Status	Comments
ESRAS	Tag	ESRAS,Tag1	Secret ID	Ok	No attacks within bounds.
		ESRAS,Tag2	Secret XOR(Rank(Rot(XOR(IDS,XOR(Rank(Rot(K1,K2),K1...	Ok	No attacks within bounds.
		ESRAS,Tag3	Secret K1	Ok	No attacks within bounds.
		ESRAS,Tag4	SKR XOR(Rank(Rot(K1,K2),K1),A)	Ok	No attacks within bounds.
		ESRAS,Tag5	Niagree	Ok	No attacks within bounds.
		ESRAS,Tag6	Nisynch	Ok	No attacks within bounds.
		ESRAS,Tag7	Alive	Ok	No attacks within bounds.
		ESRAS,Tag8	Weakagree	Ok	No attacks within bounds.
		ESRAS,Tag9	Commit Reader,BL'	Ok	No attacks within bounds.
		ESRAS,Tag10	Commit Reader,BR'	Ok	No attacks within bounds.
	Reader	ESRAS,Reader1	Secret ID	Ok	No attacks within bounds.
		ESRAS,Reader2	Secret XOR(Rank(Rot(XOR(IDS,XOR(Rank(Rot(K1,K2),K1...	Ok	No attacks within bounds.
		ESRAS,Reader3	Secret K1	Ok	No attacks within bounds.
		ESRAS,Reader4	Secret K2	Ok	No attacks within bounds.
		ESRAS,Reader5	Secret XOR(Rank(Rot(K1,K2),K1),A)	Ok	No attacks within bounds.
		ESRAS,Reader6	Secret XOR(XOR(Rank(XOR(Rank(Rot(K1,K2),K1),A),XOR...	Ok	No attacks within bounds.
		ESRAS,Reader7	SKR XOR(Rank(Rot(K1,K2),K1),A)	Ok	No attacks within bounds.
		ESRAS,Reader8	Niagree	Ok	No attacks within bounds.
		ESRAS,Reader9	Nisynch	Ok	No attacks within bounds.
		ESRAS,Reader10	Alive	Ok	No attacks within bounds.
		ESRAS,Reader11	Weakagree	Ok	No attacks within bounds.
		ESRAS,Reader12	Commit Tag,CL'	Ok	No attacks within bounds.
		ESRAS,Reader13	Commit Tag,CR'	Ok	No attacks within bounds.

Done.

Fig. 17 Scyther verification results under DY Adversary Model with Authentication

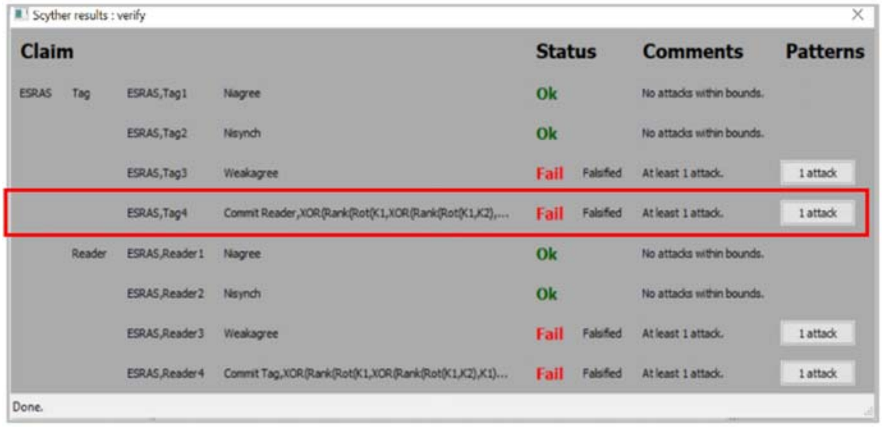


Scyther results : verify

Claim				Status	Comments
ESRAS	Tag	ESRAS,Tag1	Niagree	Ok	No attacks within bound
		ESRAS,Tag2	Nisynch	Ok	No attacks within bound
		ESRAS,Tag3	Weakagree	Ok	No attacks within bound
		ESRAS,Tag4	Commit Reader,XOR(Rank(Rot(K1,XOR(Rank(Rot(K1,K2),...	Ok	No attacks within bound
	Reader	ESRAS,Reader1	Niagree	Ok	No attacks within bound
		ESRAS,Reader2	Nisynch	Ok	No attacks within bound
		ESRAS,Reader3	Weakagree	Ok	No attacks within bound
		ESRAS,Reader4	Commit Tag,XOR(Rank(Rot(K1,XOR(Rank(Rot(K1,K2),K1)...	Ok	No attacks within bound

one.

Fig. 18 Scyther verification results Under eCK Adversary Model without Authentication



Scyther results : verify

Claim				Status		Comments	Patterns
ESRAS	Tag	ESRAS,Tag1	Niagree	Ok		No attacks within bounds.	
		ESRAS,Tag2	Nisynch	Ok		No attacks within bounds.	
		ESRAS,Tag3	Weakagree	Fail	Falsified	At least 1 attack.	1 attack
		ESRAS,Tag4	Commit Reader,XOR(Rank(Rot(K1,XOR(Rank(Rot(K1,K2),...	Fail	Falsified	At least 1 attack.	1 attack
	Reader	ESRAS,Reader1	Niagree	Ok		No attacks within bounds.	
		ESRAS,Reader2	Nisynch	Ok		No attacks within bounds.	
		ESRAS,Reader3	Weakagree	Fail	Falsified	At least 1 attack.	1 attack
		ESRAS,Reader4	Commit Tag,XOR(Rank(Rot(K1,XOR(Rank(Rot(K1,K2),K1)...	Fail	Falsified	At least 1 attack.	1 attack

Done.

Fig. 19 Scyther Verification Results Under DY Adversary Model without Authentication

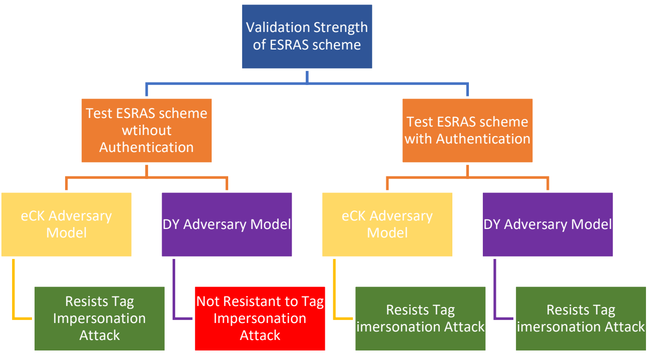

Validation Strength of ESRAS scheme
Test ESRAS scheme wtihout Authentication
Test ESRAS scheme with Authentication
eCK Adversary Model
DY Adversary Model
eCK Adversary Model
DY Adversary Model
Resists Tag Impersonation Attack
Not Resistant to Tag Impersonation Attack
Resists Tag imersonation Attack
Resists Tag imersonation Attack

Fig. 20 Experiment implementation and results

TABLE III
COMPARISON OF ULTRA- LIGHTWEIGHT AUTHENTICATION PROTOCOLS IN TERMS OF SECURITY THREATS

Security Threat	[29]	[30]	[31]	[11]
Impersonation attack	✓			✓
Tracking attack			✓	✓
Mutual authentication	✓	✓		✓
De-synchronization attacks			✓	✓

VII.CONCLUSIONS AND FUTURE RESEARCH

Across numerous applications on a global scale, RFID systems offer tremendous technical potential and profitable opportunities. It is important to note that RFID is the key technology behind the IoT, which enables real-time information to be transmitted between objects without manual intervention. Security is a significant concern in RFID systems, and this study examined authentication protocols between tags and readers. Consequently, this study reviewed physical attacks that could affect the RFID system and investigated the recently proposed authentication protocol, ESRAS.

This study provided formal validation for the effectiveness of the ESRAS Scheme by using the Scyther tool against physical tag impersonation attacks. This study assumed the hypothesis and employed the experiment as a data collection method for implementation. This assessment was performed when concerned with/without the authentication process, under the eCK adversary and DY adversary models, placing several parameters in the Scyther settings.

With authentication, the Scyther results under both eCK and DY adversary models demonstrated that the ESRAS scheme has no possible attacks within bounds. This case accomplished this study's hypothesis and answered the research question.

However, without authentication, the Scyther results under the DY adversary model demonstrated that the ESRAS scheme did not resist tag impersonation attacks. This confirms the importance of authentication as a security requirement to ensure the integrity and confidentiality of RFID components.

The summary of this study's experimental implementation and results are depicted in Fig. 20. Future research will evaluate such findings and (considering the Scyther tool) settings be optimized parameters (as defined in Table II), consequently verifying ESRAS protocol. In addition, future research can define the ESRAS protocol, perform automated security,y claim verification and compare the result with this study's findings.

Otherwise, one can analyse the security protocol using another tool, such as Casper FDR and ProVerif, and consider introducing an adversary model to validate the effectiveness of the ESRAS Scheme against several attacks.

Furthermore, this study did not consider multiple tag or reader environments. Hence, assuming that multiple tag environments communicate with the reader, future research could validate if the ESRAS scheme provides a security protocol for multiple tags. Moreover, future research can include developing authentication protocols more resistant to side-channel attacks. A side-channel attack can extract information from an RFID system, such as the secret key used for authentication. To ensure that the system is more secure, authentication protocols. They should be designed so that they are more resilient to side-channel attacks.

Furthermore, it is necessary to develop additional adversary models. There is a need to develop realistic attack models for RFID systems that consider the practical limitations of attackers. It would be possible for researchers to develop more effective authentication protocols if adversary models were more realistic.

There is a need to consider the impact of hardware constraints on protocol security. The processing power and memory of RFID tags and readers are limited, which can compromise the protocol's security. When the tag does not have enough memory to store the session key, the protocol may be susceptible to attack.

Overall, we believe that such technology will gradually be implemented within practical applications for the safety of all parties involved in RFID-based transactions.

References

[1] J. &. W. Y. Davies, "Physically unclonable functions (PUFs): a new frontier in supply chain product and asset tracking.," IEEE Engineering Management Review, vol. 49, no. 2, pp. 116-125, 2021.

[2] J. H. I. W. Y. M. I. S. M. K. &. R. M. G. Khor, "Security problems in an RFID system," Wireless Personal Communications, vol. 59, pp. 17-26, 2011.

[3] P. C.-G. C. &. M.-G. J. Caballero-Gil, "RFID authentication protocol based on a novel EPC Gen2 PRNG," arXiv preprint arXiv, p. 2208.05345, 2022.

[4] A. &. D. G. Ibrahim, "Review of different classes of RFID authentication protocols," Wireless Networks, vol. 25, pp. 961-974, 2019.

[5] Z. &. M. K. Bilal, "Ultra-lightweight mutual authentication protocols: Weaknesses and countermeasures." in 2013 International Conference on Availability, Reliability and Security, 2013.

[6] M. B. N. N. M. L. Y. &. C. Q. Safkhani, "Tag impersonation attack on two RFID mutual authentication protocols," in 2011 Sixth International Conference on Availability, Reliability and Security, 2011.

[7] S. K. Y. B. V. K. Y. A. A. &. H. B. Gabsi, "Novel ECC-based RFID mutual authentication protocol for emerging IoT applications.," IEEE access, vol. 9, pp. 130895-130913, 2021.

[8] H. A. &. K. D. Abdul-Ghani, "A comprehensive study of security and privacy guidelines, threats, and countermeasures: An IoT perspective," Journal of Sensor and Actuator Networks, vol. 8, no. 2, p. 22, 2019.

[9] S. &. R. B. Azad, "A lightweight protocol for RFID authentication," in 2019 IEEE Asia-Pacific Conference on Computer Science and Data Engineering (CSDE), 2019.

[10] B. &. M. C. J. Song, "RFID authentication protocol for low-cost tags," in Proceedings of the first ACM conference on Wireless network security, 2008.

[11] M. S. K. L. C. C. M. &. K. T. Shariq, "ESRAS: An efficient and secure ultra-lightweight RFID authentication scheme for low-cost tags," Computer Networks, vol. 217, p. 109360, 2022.

[12] M. J. D. W. M. H. B. Z. S. S. M. &. A. M. S. Imdad, "Internet of things (IoT); security requirements, attacks and counter measures," Indonesian Journal of Electrical Engineering and Computer Science, vol. 18, no. 3, pp. 1520-1530, 2020.

[13] Y. Z. Y. C. W. T. Z. &. H. Z. An, "A lightweight and practical anonymous authentication protocol based on bit-self-test PUF," Electronics, vol. 11, no. 5, p. 772, 2022.

[14] A. K. &. P. B. D. K. Singh, "Security Attacks on RFID and their Countermeasures," In Computer Communication, Networking and IoT: Proceedings of ICICC 2020, pp. 509-518, 2021.

[15] S. M. S. L. C. &. F. C. Miniaoui, "Comparing cyber physical systems with RFID applications: common attacks and countermeasure challenges," International Journal of Business Information Systems, vol. 40, no. 4, pp. 540-559, 2022.

[16] M. A. O. H. A. S. H. T. C. B. N. K. S. &. H. B. Hosseinzadeh, "An enhanced authentication protocol for RFID systems," IEEE Access, pp. 126977-126987, 2020.

[17] A. T. D. R. A. A. &. D. J. Baha'A, "Using dummy data for RFID tag and reader authentication.," Digital Communications and Networks, vol. 8, no. 5, pp. 804-813, 2022.

[18] V. K. R. K. A. A. K. V. C. Y. C. &. C. C. C. Kumar, "RAFI: Robust authentication framework for IoT-based RFID infrastructure," Sensors, vol. 22, no. 9, p. 3110, 2022.

[19] U. T. A. G. S. Y. A. R. N. R. &. G. F. W. Iqbal, "A Novel Secure Authentication Protocol for IoT and Cloud Servers," Wireless Communications and Mobile Computing, 2022.

[20] N. G. N. &. L. P. Kahya, "Formal analysis of PKM using scyther tool," in 2012 International Conference on Information Technology and e-Services, 2012.

[21] C. J. Cremers, "The Scyther Tool: Verification, Falsification, and Analysis of Security Protocols: Tool Paper," in Computer Aided Verification: 20th International Conference, CAV 2008 Princeton, NJ, USA, July 7-14, 2008 Proceedings 20, 2008.

[22] Z. Yang, "Efficient eck-secure authenticated key exchange protocols in the standard model," in Information and Communications Security: 15th International Conference, 2013.

[23] H. O. V. A. &. P. A. Yang, "Verifying Group Authentication Protocols by Scyther," J. Wirel. Mob. Networks Ubiquitous Comput. Dependable Appl., vol. 7, no. 2, pp. 3-19, 2016.

[24] Y. K. J. D. D. G. A. P. V. Y. I. &. P. G. Ko, "Drone secure communication protocol for future sensitive applications in military zone," Sensors, vol. 21, no. 6, p. 2057, 2021.

[25] H. Huang, "An eCK-Secure One Round Authenticated Key Exchange Protocol with Perfect Forward Security," J. Internet Serv. Inf. Secur., pp. 32-43, 2011.

[26] D. &. C. C. Basin, "Degrees of security: Protocol guarantees in the face of compromising adversaries.," in Computer Science Logic: 24th International Workshop, CSL 2010, 19th Annual Conference of the EACSL, Brno, Czech Republic, 2010.

[27] A. K. M. M. P. P. K. K. K. G. S. &. L. M. Yadav, "LEMAP: A lightweight EAP based mutual authentication protocol for IEEE 802.11 WLAN," in ICC 2022-IEEE International Conference on Communications, 2022.

[28] A. Sarr, "Authenticated key agreement protocols: security models, analyses, and designs," (Doctoral dissertation, Université Joseph-Fourier-Grenoble I).2010, .

[29] S. Ç. S. B. M. A. K. M. S. D. H. &. L. A. Kardaş, "k-strong privacy for radio frequency identification authentication protocols based on physically unclonable functions," Wireless Communications and Mobile Computing, vol. 15, no. 18, pp. 2150-2166, 2015.

[30] A. &. G. B. B. ewari, "Cryptanalysis of a novel ultra-lightweight mutual authentication protocol for IoT devices using RFID tags," The Journal of Supercomputing, vol. 73, pp. 1085-1102, 2017.

[31] H. W. G. S. J. &. H. Z. Luo, "SLAP: Succinct and Lightweight Authentication Protocol for low-cost RFID system," Wireless Networks, vol. 24, pp. 69-78, 2018.