Computational Sciences and Engineering 1(2) (2021) 109-121

University of Guilan

Computational Sciences and Engineering

journal homepage: https://cse.guilan.ac.ir/

A Modified Continuous Lightweight Authentication to Increase the Information Security on Internet of Things

Rahim Asghari [a,*], Frus Munif [b], Reza Semyari [c]

[a] Faculty of Electrical and Computer engineering, Malek-Ashtar University of Technology, Tehran.
[b] Faculty of Electrical and Computer engineering, Malek-Ashtar University of Technology, Tehran.
[c] Faculty of Electrical and Computer engineering, Malek-Ashtar University of Technology, Tehran.

ARTICLE INFO

Article history:
Available online 11 May 2021

Keywords:
Internet of Things
Security
Lightweight Authentication Protocol
RFID

ABSTRACT

The Internet of Things is an emerging paradigm that will change the way we interact with objects and computers in the future. It envisions a global network of devices interacting with each other, over the Internet, to perform a useful action. Firstly, we provided the overview of the Internet of Things and then the relevant technologies that can help in the large-scale development of Internet of Things, then the security issues in Internet of Things and its challenging.

Secondly, we analyzed some of the lightweight authentication protocol in Internet of Things based on different techniques such as RFID Authentication and Continuous Authentication to evaluate their vulnerability. Finally, we proposed the solution for one of RFID authentication protocol by using physically unclonable functions. In this protocol, the valid authentication time period is proposed to enhance robustness of authentication between Internet of Things devices and used the authentication token to authenticate the message which transmits from sensor node to the gateway and at the end the security analysis is conducted to evaluate the security strength of the proposed protocol.

[*] Corresponding author.
E-mail addresses: meisam.mathhome@gmail.com (R. Asghari)

http://dx.doi.org/10.22124/cse.2021.19589.1015

1. Introduction

Internet of Things is a new revolution of the Internet. Objects make themselves recognizable and they obtain intelligence by making or enabling context related decisions thanks to the fact that they can communicate information about themselves. They can access information that has been aggregated by other things, or they can be components of complex services. This transformation is concomitant with the emergence of cloud computing capabilities and the transition of the Internet towards IPv6 with an almost unlimited addressing capacity [1,2]. Internet of Things faces challenges that need to be overcome to ensure that the technology is successfully deployed on a large scale the challenge in regard to security is of particular importance, as Internet of Things technology is designed to privately collect information about the environment in which it is residing in at the moment[3].

Providing security to internet of things is far more complicated as compared to Internet security. Because internet of things is a combination of different networks, that not only the security issues related to the mobile network, the sensor network and the Internet. However, problems such as privacy protection, heterogeneous network, authentication, access control, management, etc. arise because of the integration of different networks. Therefore, you must perform a solution for each security issue.

The technologies for the Internet of Things such as sensor networks, RFID, M2M, mobile Internet, semantic data integration, semantic search, IPv6, etc[4]. Can be grouped into three categories:

a) Technologies that enable “things” to acquire information.
b) Technologies that enable “things” to process information.
c) Technologies to improve security and privacy.

The first two categories can be jointly understood as functional building blocks required building “intelligence” into “things”, which are indeed the features that differentiate the internet of things from the usual internet. The third category is not a functional but rather factual requirement, without which the penetration of the internet of things would be severely reduced [5, 6].

We can divide IOT communication into three categories that Interacting through internet (communication channel):

a) People to people.
b) People to machine (things).
c) Machine (things) to machine (things).

Machine to Machine (M2M) communication provides each component (machine) with access to the Internet, leading to the evolution of the internet of things technology. The internet of things, which can be regarded as an enhanced version of M2M communication technology, was proposed to realize intelligent thing-to-thing communications by utilizing Internet connectivity. In the internet of things,

“things” are generally heterogeneous and resource constrained. In addition, such things are connected with each other over low-power and lousy networks (LLNs) [7, 8].

The goal of internet of things is to allow thing device to communicate anytime, anyplace, with anything and anyone using any path/network and any communication service using the Internet protocol Figure 1.

In this environment, security is a critical element that is necessary to enable various types of applications and services. Various types of authentication technologies and session-key distribution/agreement technologies have been proposed for security services under Machine-to-Machine (M2M) or sensor-network environments [9].

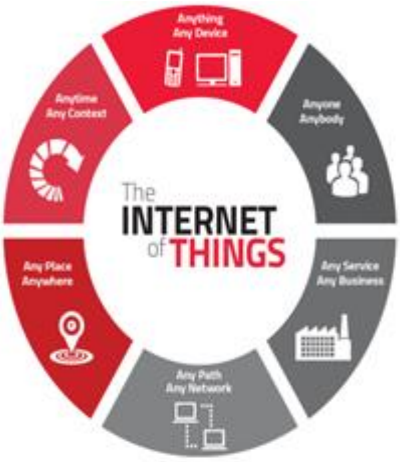

Figure.1. Internet of Things Concepts

Internet of things is an interesting field Furthermore, there has been quite a bit of research into internet of things, it's possible uses and the security and the privacy aspects of internet of things. This thesis will focus on the authentication aspect of internet of things. Authentication is important since the majority of communications will occur without user interaction. One of the aims of this thesis is to perform a literature survey of the state of security in regard to internet of things. This is done to get a better overview of what has been done, and then proposed a lightweight authentication protocol for internet of things, this lightweight protocol proposed to authenticate the legitimacy of a peer device when a message needs to transmit to the peer device[10].

2. Light weight Authentication

Security professionals try to protect their environments as effectively as possible. These actions can also be described as protecting confidentiality, integrity, and availability (CIA), or maintaining CIA. CIA is shown in figure 2.

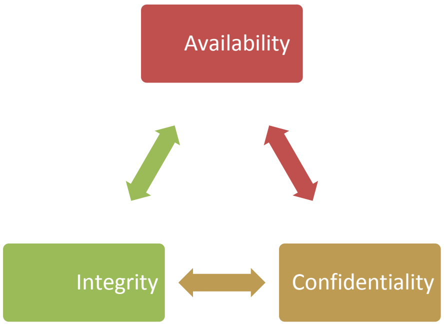

Figure.2. the Security Requirements Triad

The purpose of information security is to protect an organization's valuable resources, such as information, hardware, and software. Through the selection and application of appropriate safeguards, security helps an organization to meet its business objectives or mission by protecting its physical and financial resources, reputation, legal position, employees, and other tangible and intangible assets[11]. Figure3, show Information security includes the broad areas of information security management, computer and data security, and network security[12].

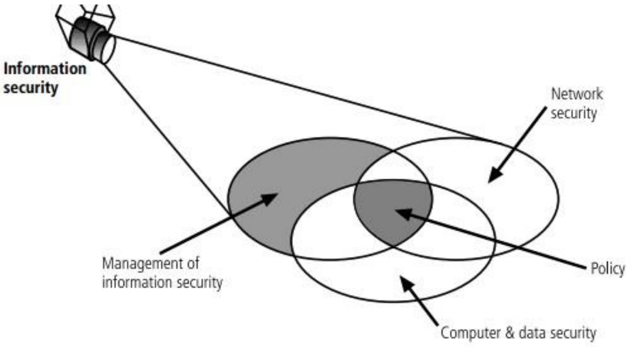

Figure.3. Components of information security [13]

Authentication is the process of determining whether someone or something is actually who they claim to be, and not a malicious user pretending to be someone they are not.

For internet of things, authentication is important since the majority of communications will occur without user interaction. Additionally, the ability to ensure that correct devices, sensors, and users have the right to access the network of resources and information is an important security concern. It is also crucial to ensure that information, commands, and requests are received from the correct devices [14].

2.1 Review on the Fan Authentication Protocol

Fan et al. authentication protocol is shown in figure 4 and in this protocol the three Components of it can execute the cro (·), Rot (·), PRNG (·) and pre-share the K_i [14].

The details of the protocol are as following:

Step 1: In Reader, before communication to tag the reader generate random number N_R, initializes the information of Query and send N_R and Query to the tag .

Step 2: The tag obtains N_R, and sets the value of Mark to "00" , which indicates a new session starts. Then the tag computes cro(RID$\oplus$TID, K_i) and sends it as well as N_T to the reader.

Step 3: After receiving the message, the reader obtains N_T and sends the message that received and N_R to the server directly.

Step 4: After receiving the message, the server obtains N_R and N_T, then use the value cro(RID$\oplus$TID, K_i) that received to search the matching index content in IDT. If it can find a match, it indicates that the last session has been done correctly and the current session is executable. Then the server generates a random number N_S and sends cro(RID $\oplus$ TID, $K_i \oplus N_S$) and Rot(K_i $\oplus$ TID, K_i $\oplus$ RID)‖$N_S \oplus K_i$ to the reader. Otherwise the authentication fails and the protocol will be terminated.

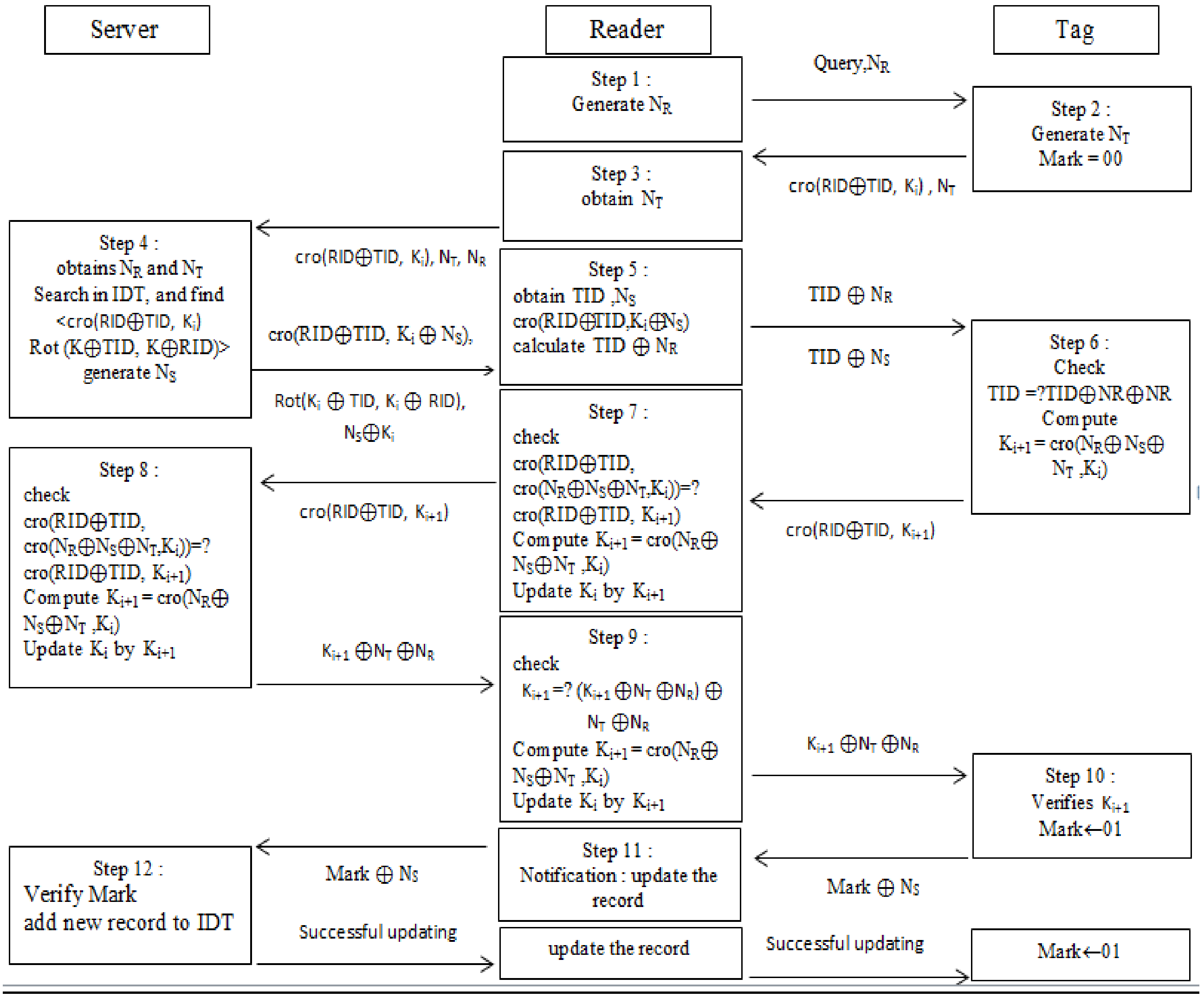
Server
Reader
Tag
Step 1 : Generate N_R
Query,N_R
Step 2 : Generate N_T Mark = 00
cro(RID$\oplus$TID, K_i) , N_T
Step 3 : obtain N_T
cro(RID$\oplus$TID, K_i), N_T, N_R
Step 4 : obtains N_R and N_T Search in IDT, and find <cro(RID$\oplus$TID, K_i) Rot (K$\oplus$TID, K$\oplus$RID)> generate N_S
cro(RID$\oplus$TID, $K_i \oplus N_S$),
Rot(K_i $\oplus$ TID, K_i $\oplus$ RID), $N_S \oplus K_i$
Step 5 : obtain TID ,N_S cro(RID$\oplus$TID,$K_i \oplus N_S$) calculate TID $\oplus$ N_R
TID $\oplus$ N_R
TID $\oplus$ N_S
Step 6 : Check TID =?TID$\oplus$NR$\oplus$NR Compute K_{i+1} = cro($N_R \oplus N_S \oplus$ N_T ,K_i)
Step 7 : check cro(RID$\oplus$TID, cro($N_R \oplus N_S \oplus N_T$,$K_i$))=? cro(RID$\oplus$TID, K_{i+1}) Compute K_{i+1} = cro($N_R \oplus$ $N_S \oplus N_T$,K_i) Update K_i by K_{i+1}
cro(RID$\oplus$TID, K_{i+1})
cro(RID$\oplus$TID, K_{i+1})
Step 8 : check cro(RID$\oplus$TID, cro($N_R \oplus N_S \oplus N_T$,$K_i$))=? cro(RID$\oplus$TID, K_{i+1}) Compute K_{i+1} = cro($N_R \oplus$ $N_S \oplus N_T$,K_i) Update K_i by K_{i+1}
$K_{i+1} \oplus N_T \oplus N_R$
Step 9 : check K_{i+1} =? ($K_{i+1} \oplus N_T \oplus N_R$) $\oplus$ $N_T \oplus N_R$ Compute K_{i+1} = cro($N_R \oplus$ $N_S \oplus N_T$,K_i) Update K_i by K_{i+1}
$K_{i+1} \oplus N_T \oplus N_R$
Step 10 : Verifies K_{i+1} Mark←01
Step 11 : Notification : update the record
Mark $\oplus$ N_S
Mark $\oplus$ N_S
Step 12 : Verify Mark add new record to IDT
Successful updating
update the record
Successful updating
Mark←01

Figure.4. Fan et al. authentication protocol.

Step 5: check TID and obtain N_S in the reader. According to the hamming weight W (K ⊕ TID) of the rotation operation and $K_i \oplus K_i \oplus TID$, TID is obtained, and at the same time, N_S is gotten through the XOR operation of conducting $K_i \oplus K_i \oplus N_S$ Then verify the value Cro (RID⊕ TID, $K_i \oplus N_S$) by comparing the received value with the calculated value in local. If OK, calculate TID⊕ N_R and TID⊕ N_S , and send them to the tag.

Step 6: After receiving the message, the tag obtains NS and then check TID by doing an XOR operation between TID⊕N_R⊕N_R. at this time the three random numbers are acquired, they are N_T, N_R, N_S and we can start to update the key K_i as K_{i+1} = cro($N_R \oplus N_S \oplus N_T$, K_i) and sends it to the reader involved in the message cro(RID⊕TID, K_{i+1}).

Step 7: Upon receiving the message cro(RID⊕TID, K_{i+1}), the reader executes the cro(RID⊕TID, cro($N_R \oplus N_S \oplus N_T$, K_i)), then compare it with cro(RID⊕TID, K_{i+1}) that received , If they are equal, the K in the reader updates to be K_{i+1} (new key value), then and sends it to the server by the message cro(RID⊕TID, K_{i+1}).

Step 8: The server receives the message and execute the same comparison in the before step and if they are equal ,the server will update key with new value , and then computes the message ($K_{i+1} \oplus N_T \oplus N_R$) and sends it to the reader.

Step 9: the reader verifies K_{i+1} and sends the message $K_{i+1} \oplus N_T \oplus N_R$ to the tag for the same verification process .

Step 10: Tag verifies K_{i+1}, If verification is correct, Mark is set to be "01", indicating the synchronization about K is completed. Then the tag computes Mark ⊕ N_S and sends it to the server through the reader.

Step 11: The tag send to server (Mark ⊕ N_s) and the server obtain the Mark and check it , if its value is "01", the server knows that new K is about consistency, and a new record {Cro (RID ⊕TID , K_{i+1}) , Rot (K_{i+1} ⊕ TID, K_{i+1} ⊕ RID)} will be generated and added to the IDT.

Step 12: Now, the tag sets Mark = 10, indicating the authentication protocol is completed.

2.2 Security analysis of the Fan et al. protocol

Fan et al. protocol is vulnerable to several attacks such as secret disclosure, anonymity and reader impersonation.

2.2.1 Secret disclosure attack

Because the attacker can by eavesdropping the messages of Steps 1, 2 and 9 which are respectively N_R, N_T and $K_{i+1} \oplus N_T \oplus N_R$, because the random number transmitted between Reader and Tag as plaintext without any cryptographic operation. So, the attacker can obtain the new session key k_{i+1} from the equation $Ki_{+1} = (K_{i+1} \oplus N_T \oplus N_R) \oplus N_R \oplus N_T$.

2.2.2 Attack on the anonymity

Because the attacker can by eavesdropping the messages of Steps 1 and 5 which are respectively the N_R and $TID \oplus N_R$. So the attacker can obtain the TID from equation $TID = (TID \oplus N_R) \oplus N_R$ and hack the anonymity of the target tag.

2.2.3 Reader impersonation attack

An impersonation attack is an attack in which an adversary successfully assumes the identity of one of the legitimate parties in a system or in a communications protocol. The goal of a strong identification or entity authentication protocol is to make small the probability of this attack. IN this protocol assume that the attacker has already done previous attacks and obtained the tag's identification TID and the tag's current key K_i. We describe this attack against Fan et al. protocol as follows:

- The adversary send start protocol and send random number N_{A1} to tag.
- Tag generate random number N_T and set mark 00 then send COR $(RID \oplus TID \oplus K_i) \| N_T$ to adversary.
- Adversary generate new random number N_{A2} and sent to tag the $(TID \oplus N_{A1}, TID \oplus N_{A2})$.
- Tag obtain N_{A2} and verifies $TID = TID \oplus N_{A1} \oplus N_{A1}$, and authenticate the adversary, then update the key with new $K_{i+1} = cro(N_{A1} \oplus N_{A2} \oplus N_T, K_i)$ and send to adversary $cro(RID \oplus TID, K_{i+1})$.
- Adversary can obtain the N_T at step 3 of protocol .So, Adversary uses N_{A1}, N_{A2}, N_T and K_i to computes $K_{i+1} = cro(N_{A1} \oplus N_{A2} \oplus N_T, K_i)$, then send to tag $(K_{i+1} \oplus N_T \oplus N_{A1})$.
- Upon receiving the message, the tag verifies K_{i+1} and sets Mark = 01, indicating the synchronization of K is completed . Then the tag computes Mark $\oplus N_{A2}$ and sends it to the adversary.
- The adversary informs the tag that the updating is successful.
- The tag sets Mark = 10 and authentication protocol is completed.

In this attack the adversary does not send any information to the server and the steps that executed are steps which transmitted message between tag and reader.

3. Proposed Authentication Protocol

Continuous authentication provides fast and simple authentication for frequent message transmission in short time intervals and by using the short time intervals for authentication, the authentication between internets of thing devices becomes secure and robust.

This section shows the approaches that proposed for authentication in internets of thing. We will first show the proposed Solution for the Fan et al. protocol by using the PUF to solve the problems which we presented in the previous section.

In the previous section we have seen some of Weaknesses in Fan et al. protocol. The main idea that we have to think about is how to solve these weaknesses. So, we can use the PUF to solve this problem and we explain the solution as below:

- Tag and Reader must be registered into backend server.
- At the registration stage all of the tag, reader, and server generate their challenges (C_S,C_R,C_T), and send these challenges to the tag Via a secure channel, then tag produces the response (R_S, R_R, R_T) by using embedded function PUF, then the server store the pair of (C, R) in secure table. As shown in Table 1.
- By using the pair (C, R), we do not need to use the random variable.

Table 1. Challenges, response Table

	challenges	**response**
server	C_S	R_S
reader	C_R	R_R
tag	C_T	R_T

3.1 Proposed Solution Steps

In this section, steps of proposed Solution protocol are showed.

Step1: The reader starts the protocol by it initializes the information of Query and retrieve its challenge C_R then send the Query as well as C_R to the tag.

Step2: The tag obtains C_R , and sets the value of Mark to "00" , which indicates a new session starts. Then the tag computes:

- R_T = PUF(C_T).
- R_R = PUF (C_R).
- cro (RID $\oplus$ TID , K_i).
- Then send cro (RID$\oplus$TID, K_i) , C_T,H($R_T$$\oplus$ R_R) to the reader.

Step3: After the reader received the message from the tag, the reader obtains the tag challenge C_T, and then send the received message and C_R to the server.

Step4: When the server received the message. It obtains the challenges for tag and reader. Then, the server used C_R ,C_T which obtained to retrieve the corresponding response and the server compare check :

IF (H ($R_T \oplus R_R)_{Retrieved}$ = H($R_T \oplus R_R)_{Received}$).

Which indicates that the server retrieves the responses correctly and the tag and the reader have been registered on the server before? Otherwise, the protocol will be terminated. After this verification, the server employs cro (RID$\oplus$TID, K_i) to find the corresponding index content in the IDT. If it can find a match, it indicates that the last session has been done correctly and the current session is executable. Then the server retrieves its challenge C_S and send cro(RID$\oplus$TID, $K_i \oplus C_S$), Rot($K_i \oplus$ TID, $K_i \oplus$ RID), $C_S \oplus K_i$, H($R_S \oplus R_R$) to the reader. Otherwise the authentication fails and the protocol will be terminated.

Step5: When the reader receives the message, the reader obtains TID according to hamming weight W ($K_i \oplus$ TID) of rotation operation and $K_i \oplus K_i \oplus$ TID. Then, the reader obtains C_S through the XOR operation of conducting $K_i \oplus K_i \oplus$ CS . After the reader obtained TID and C_S, then verify the value of cro(RID$\oplus$TID, $K_i \oplus C_S$) by comparing the received value with calculated value in local. If ok, calculate the TID$\oplus C_R \oplus K_i$, TID $\oplus C_S$ and send it with H($R_S \oplus R_R$) to tag.

Step6: In the tag side, after receiving the message, TID is checked by conducting the XOR operation between TID $\oplus C_R \oplus K_i$ that received and C_R , K_i obtained before.

TID =TID $\oplus C_R \oplus K_i \oplus C_R \oplus K_i$.

Then the tag obtain C_S by conducting the XOR operation TID $\oplus C_S \oplus$ TID, after that, the tag calculate the response of server challenge C_S by using PUF function as R_S = PUF (C_S) and use this R_S to check if

H($R_S \oplus R_R$)= H($R_S \oplus R_R)_{received}$

Then the tag authenticates the server and the reader and update the secret key K_i as follow : K_{i+1} = cro($R_R \oplus R_S \oplus R_T$,K_i) .

Finally, the tag send cro (RID$\oplus$TID, K_{i+1}), $K_{i+1} \oplus C_S$ to the reader.

Step7: When the reader receives the message from tag, it the obtain new value of key by performing XOR operation K_{i+1}= $K_{i+1} \oplus C_S$ with C_S which obtained before . And then send cro(RID$\oplus$TID, K_{i+1}) to the server.

Step8: When the server receives the message, it uses the stored response to calculate the new key value and to check if the value of cro operation with the new key calculated is equal to the value of cro operation which received. Then, if the verification occurs, the server computes $K_{i+1} = cro(R_R \oplus R_S \oplus R_T, K_i)$, and update K_i value with this K_{i+1} value.

Finally, the server sends $K_{i+1} \oplus R_T \oplus R_R$ and $H(K_{i+1} \oplus C_S \oplus C_R)$ to the reader.

Step9: The reader receives the message from the server. Then, the reader uses the value of K_{i+1} which obtained in step 7 to check if he hash function of the received message is equal to hash function of the message which calculates by local variables as IF $H(K_{i+1} \oplus C_S \oplus C_R)_{calculated} = H(K_{i+1} \oplus C_S \oplus C_R)_{received}$.

Then, if the verification occurs, the reader update K_i value with K_{i+1} value and send $K_{i+1} \oplus R_T \oplus R_R$ to the tag.

Step10: When the tag received the message, it retrieves the K_{i+1} value from this message as follow: $K_{i+1} = K_{i+1} \oplus R_T \oplus R_R \oplus R_T \oplus R_R$.

The tag verifies if this K_{i+1} is equal to K_{i+1} which computed before, if the verification occurs, the tag set the $mark$ equal 01 and this means that the synchronization about K is completed. Finally, the tag sends Mark $\oplus$ RS to the server.

Step11: The reader sends the received message to the server.

Step12: The server receives the message and obtains the $mark$ by performing the XOR operation on the received message and then verifies if the $mark$ is equal to 01 that indicate the new key is of consistency. Then the server generates the new record {Cro (RID $\oplus$TID, K_{i+1}), Rot ($K_{i+1} \oplus$ TID, $K_{i+1} \oplus$ RID)} and add this record to the IDT. Finally, the server tells the reader and the tag that the record was updated successfully, and then the tag sets the mark is equal to 10, indicating the authentication protocol is completed. We show the steps of proposed Solution protocol in Figure 5.

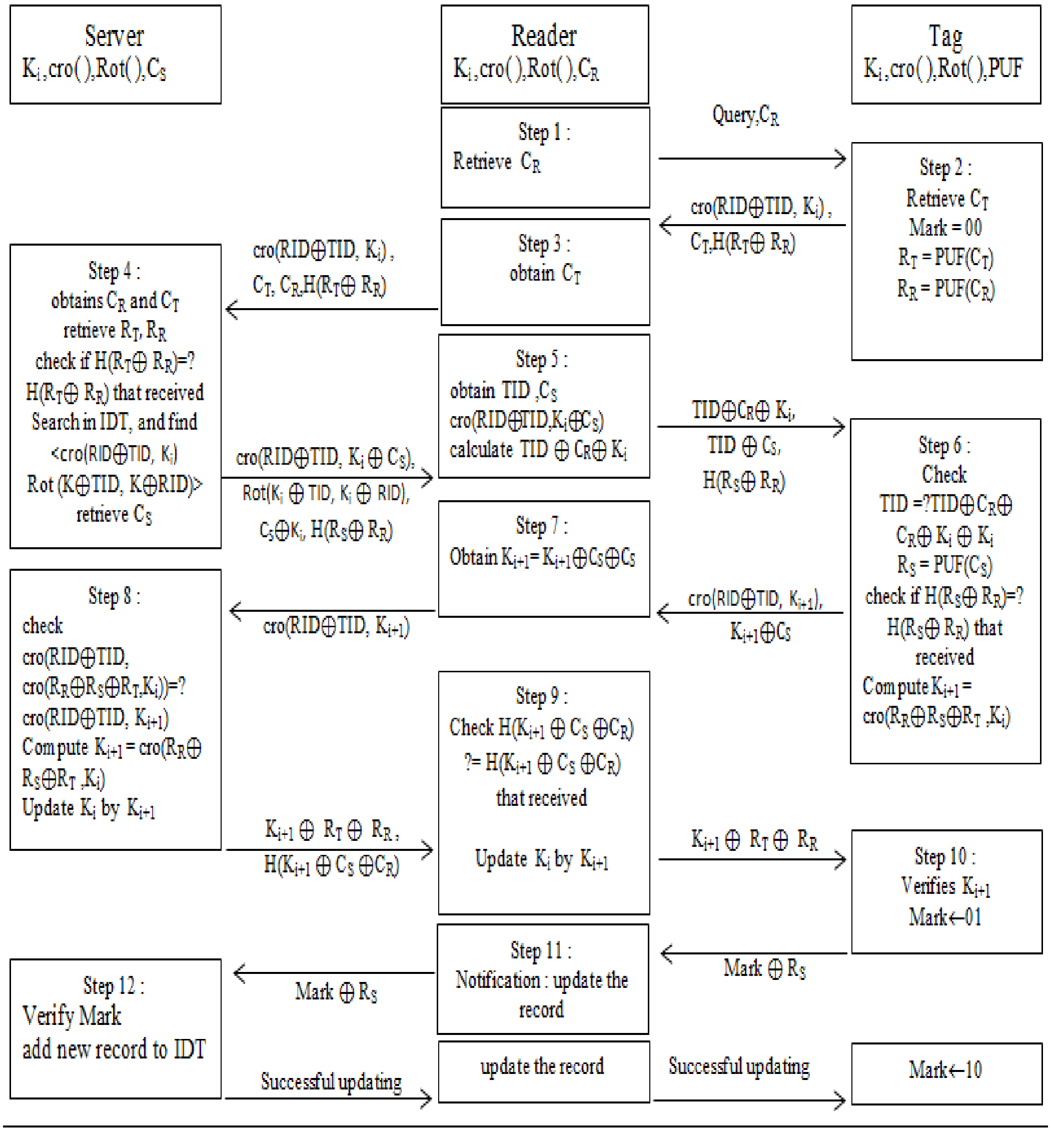
Server
K_i,cro( ),Rot( ),C_S
Reader
K_i,cro( ),Rot( ),C_R
Tag
K_i,cro( ),Rot( ),PUF
Step 1 : Retrieve C_R
Query,C_R
Step 2 : Retrieve C_T Mark = 00 R_T = PUF(C_T) R_R = PUF(C_R)
cro(RID⊕TID, K_i) , C_T,H(R_T⊕ R_R)
Step 3 : obtain C_T
cro(RID⊕TID, K_i) , C_T, C_R,H(R_T⊕ R_R)
Step 4 : obtains C_R and C_T retrieve R_T, R_R check if H(R_T⊕ R_R)=? H(R_T⊕ R_R) that received Search in IDT, and find <cro(RID⊕TID, K_i) Rot (K⊕TID, K⊕RID)> retrieve C_S
cro(RID⊕TID, K_i ⊕ C_S), Rot(K_i ⊕ TID, K_i ⊕ RID), C_S⊕K_i, H(R_S⊕ R_R)
Step 5 : obtain TID ,C_S cro(RID⊕TID,K_i⊕C_S) calculate TID ⊕ C_R⊕ K_i
TID⊕C_R⊕ K_i, TID ⊕ C_S, H(R_S⊕ R_R)
Step 6 : Check TID =?TID⊕C_R⊕ C_R⊕ K_i ⊕ K_i R_S = PUF(C_S) check if H(R_S⊕ R_R)=? H(R_S⊕ R_R) that received Compute K_{i+1} = cro(R_R⊕R_S⊕R_T ,K_i)
Step 7 : Obtain K_{i+1}= K_{i+1}⊕C_S⊕C_S
cro(RID⊕TID, K_{i+1}), K_{i+1}⊕C_S
cro(RID⊕TID, K_{i+1})
Step 8 : check cro(RID⊕TID, cro(R_R⊕R_S⊕R_T,K_i))=? cro(RID⊕TID, K_{i+1}) Compute K_{i+1} = cro(R_R⊕ R_S⊕R_T ,K_i) Update K_i by K_{i+1}
Step 9 : Check H(K_{i+1} ⊕ C_S ⊕C_R) ?= H(K_{i+1} ⊕ C_S ⊕C_R) that received Update K_i by K_{i+1}
K_{i+1} ⊕ R_T ⊕ R_R , H(K_{i+1} ⊕ C_S ⊕C_R)
K_{i+1} ⊕ R_T ⊕ R_R
Step 10 : Verifies K_{i+1} Mark←01
Mark ⊕ R_S
Step 11 : Notification : update the record
Mark ⊕ R_S
Step 12 : Verify Mark add new record to IDT
update the record
Successful updating
Successful updating
Mark←10

Figure. 5. Proposed Solution for the Fan et al. protocol.

3.2 Security Analysis For Proposed Solution

We will check if the proposed solution resistant for the vulnerabilities that shown in the original protocol.

3.2.1 Secret disclosure attack

If the attacker eavesdroppers on the messages of Steps 1, 2 and 9, the attacker will not be able to retrieve any information useful for computing K_{i+1}, because these steps contain C_R, C_T, $K_{i+1} \oplus R_T \oplus R_R$.

3.2.2 Attack on the anonymity

If the attacker eavesdroppers on the messages of Steps 1 and 5 , attacker will not be able to retrieve any information useful for computing TID, because these steps contain C_R, TID $\oplus C_R \oplus K_i$, TID $\oplus C_S$, and the attacker cannot retrieve K_i or C_S.

3.2.3 Reader impersonation attack

The attacker can do impersonation attack should the attacker has already done previous attacks and obtained the tag identification TID and the tag current key K_i. And in this proposed solution the attacker cannot obtain any of TID or K_i . So , the attacker cannot perform an impersonation attack.

3.2.4 Replay attack

Obtaining current session messages does not benefit the attacker in the next session, because in this proposed solution the protocol parties select their challenges at the beginning of protocol session and this challenges and the secret key will be changed after all authentication session. So the attacker cannot use the previous message in the current session.

3.2.5 Synchronization attack

In proposed solution and the original protocol, the session key number K is updated orderly and its consistency is ensured by verifying the validity by using the mark flag which consists of two bits and used for signing the current system synchronization status. This flag has three status values as follow:

- "00" indicate to establishing session, which means that a new session is started.

- "01" indicate the consistency of K has been completed between the protocol parties.

- "10" indicate that the synchronization is complete.

As a result, this proposed solution is resistance to Synchronization attack.

3.2.6 Security Performance Comparison

The security performance comparison between the fan et al. protocol and the proposed solution based on resistance of attacks kind, Table 2. show this comparison in which "√" means the corresponding property is satisfied while "×" means the corresponding property is not satisfied.

Table 2. Security Performance Comparison

Authentication protocols	*Mutual authentication*	*Tag anonymity*	*Resistance to Secret disclosure*	*Resistance to reader impersonation*	*Resistance to replay attack*	*Resistance to synchronization attack*
Fan et al. protocol	√	×	×	×	√	√
proposed Solution	√	√	√	√	√	√

3.2.7 Computation Cost Comparison

For computation cost comparison between the fan et al. protocol and proposed solution, should define the computational operations which, used in the protocol. In the fan et al. protocol uses these operations (Rot, cro, PRGN, XOR), and the proposed solution uses the same operations, but adds these operations to it (PUF, HASH). So, the fan et al. protocol achieves better computation cost than proposed solution, but this proposed solution achieves more security.

4. Conclusion

Internet of things technology is designed to privately collect information about the environment in which resides in at the moment. The most important security requirements in the internet of things include secure booting, authentication, access control, data integrity and privacy. First, we conducted the security analysis for the Fan et al. protocol to detect weaknesses in this protocol. And provide a proposed solution to this vulnerability by using PUF. Second, we proposed a lightweight continuous authentication protocol for internet of things.

5. References

[1]. Keyur Patel, K, Sunil Patel, M, (2016) “Internet of Things-IOT: Definition, Characteristics, Architecture, Enabling Technologies, Application & Future Challenges”, IJESC.

[2]. Atzori, L, Iera, A, Morabito, G, (2010) “The Internet of Things: A survey”, Elsevier.

[3]. Subha, R, (2017), “Biometrics in Internet of Things (IoT) Security”, IJERGS.

[4]. Naveed Aman, M, Chaing Chua, K, & Sikdar, B, (2017), “A Light-Weight Mutual Authentication Protocol for IoT Systems”, IEEE.

[5]. Braeken, A,(2018), “PUF Based Authentication Protocol for IoT”, mdpi/journal/ symmetry.

[6]. El-hajj, M, Fadlallah, A, Chamoun, M and Serhrouchni, A,(2019), “A Survey of Internet of Things (IoT) Authentication Schemes”, mdpi/journal/sensors.

[7]. Xu, H, Ding, J, Li, P, Zhu, F, & Wang, R, (2018), “A Lightweight RFID Mutual Authentication Protocol Based on Physical Unclonable Function”, Sensors.

[8]. Jaiswal, S, Gupta, D, (2017), “Security Requirements for Internet of Things (IoT)”, Springer.

[9]. Conti, M, Dehghantanha, A, Franke, F, Watson, S, (2018), “Internet of Things security and forensics: Challenges and opportunities”, Elsevier.

[10]. Kaur, S, Singh, I, (2016), “A Survey Report on Internet of Things Applications”, IJCST.

[11]. Park, N, & Kang, N, (2016), “Mutual Authentication Scheme in Secure Internet of Things Technology for Comfortable Lifestyle”, mdpi/journal/sensors.

[12]. Balasubramaniam, R, Sathya, R, Ashicka, S & Senthil Kumar, S, (2016), “an analysis of RFID. authentication schemes for internet of things (IoT) in healthcare environment using elgamal elliptic curve cryptosystem”, IJRTER.

[13]. Michael, E, Herbert, J. Mattord, J, (2012),” Principles of Information Security, Fourth Edition”, Course Technology.

[14]. K. Fan, W. Jiang, H. Li, & Y. Yang, (2018), “Lightweight RFID Protocol for Medical Privacy Protection in IoT”, IEEE, 2018.