A Multi-protocol Security Framework to Support Internet of Things

Biplob R. Ray[1], Morshed U. Chowdhury[2(✉)], and Jemal H. Abawajy[2]

[1] School of Engineering and Technology,
CQ University, Rockhampton, Australia
b.ray@cqu.edu.au
[2] School of Information Technology, Deakin University, Burwood, Australia
{muc,Jemal}@deakin.edu.au

Abstract. In this paper we are proposing a multi-protocol security framework for sensors and actuators used in Internet of Things (IoT). This is to make sure that IoT security framework is capable of accommodating a number of secure communication protocols to support diverse need of IoT systems. The proposed framework will extend scope of combining common security functionalities like mutual authentication, malware injection of all integrated secure communication protocols and will make these services universally available to them. The IoT provision requires all diverse actuators and sensor networks connected together. The aim of the proposed security framework is to secure this connected diverse networks universally with least amount of performance tradeoff.

Keywords: IoT · Framework · Security · Multi-protocol · Sensors and actuators

1 Introduction

While securing a system like IoT, security provision needs a mechanism to accommodate more than one security protocols to ensure security and business needs of the system. A framework with multi-protocol adaptation capability will be appropriate to provide security for diverse networked system like IoT [1]. Furthermore, the IoT security systems have to ensure that redundant security services are not repeatedly executed for the system by multi-protocols. As an example, an ownership protocol [2] implemented mutual authentication to an actuator and then immediately the same system executed tracking protocol [3] which require to execute mutual authentication too. In this case the duplication of mutual authentication is not required as long as earlier one is still valid. A system like IoT needs to stop execution of this redundant security services to make system scalable.

A number of work identified in literature that worked to provide security framework for IoT. Among them, the proposed framework by Ray et al. [1] have attempted to integrate multiple security protocols. However, none of the existing protocols have a working process that allows a unified framework to integrate security protocols therefore they can be used universally. This gives us a rational to develop a unified security framework which is capable to adapt multi-protocol and eliminate duplicate

R. Deng et al. (Eds.): SecureComm 2016, LNICST 198, pp. 257–270, 2017.
DOI: 10.1007/978-3-319-59608-2_14

security services for the system. In our work we have extended the framework from [1] to achieve two things:

- Multiple security protocol adaption with in the same framework
- Elimination of redundant security service execution for the system.

The rest of the paper is organized as follow: in Sect. 2 we detail the relevant existing work and the system model used in this paper. The proposed framework and techniques are detailed in Sect. 3. The working process and computational details of the required techniques for the framework are detailed in Subsects. 3.2 and 3.3. Finally in Sect. 4 we conclude our paper.

2 Background

To allow seamless connectivity of global objects, multiple administrative domains need to work together collaboratively. The IoT system must have a security mechanism that are well accepted (universal) [1, 4] by all administrative domains. This prompts the need of a security framework that offers unification for smoother integration of diverse networked systems. Moreover, it also needs to offer scalability, security and adaptability to make it usable with IoT. In Subsect. 2.1, we have presented existing research and development on security framework, secure and scalable identification techniques, universal security clearance to reduce security trade-off. The system model used in this paper is detailed in Subsect. 2.2.

2.1 Literature Review

The security risks poses by sensors and actuators systems is a serious concern [2, 4] for IoT deployment. To make security assurance acceptable by all involved entities in IoT, the sensors and actuators system need to offer stronger security with easily deployable universal security framework. However not much work has been done to address this issue. We have identify three sensors and actuators security related frameworks proposed in [4, 7, 8]. In [7], Konidala et al. have proposed a security framework for RFID-based applications in smart home environment. This framework was designed to protect consumer privacy in application level but it does not provide any guideline to protect communication between reader and objects. It has proposed the use of HTTPS to secure communication between mobile device and RFID backend server in application layer. The smart home is a micro part of the IoT system.

Dong Seong et al. [4] have proposed a framework to achieve universal authentication and authorization for RFID multi-domain System. The work in this paper [4] can be considered the pioneer work to address security requirements for sensors and actuators in the context of IoT. This paper [4] also acknowledged a need of a universal security framework to address security of sensors and actuators systems. However, the work is focused to achieve authentication and authorization only and did not consider other security properties required to be protected in the context of IoT. Lim et al. [8] have proposed a cross- layer framework to address privacy of IoT system. The paper

[8] has stated that traceability of sensors and actuators is a multi-layer problem, and called for a multi-layer solution to address the problem appropriately. Their privacy protection framework works in physical and MAC layer for protection from traceability [8]. This work used randomized bit encoding scheme to mitigate 'same-bit' problem, and proposed a more secure system model that can protect the unique identifier of actuator autuators against disclosure to eavesdroppers and unauthorized interrogators.

Most recently, Cisco has announced a flagship Cisco IoT System which has 6 layers: Network Connectivity, Fog Computing, Security, Data Analytics, Management with Automation and Application Enablement Platform [9]. The security layer of this system has four sub-layers: authentication, authorization, network enforced policy and inherent security analysis [9]. The Cisco IoT system aims to address security through network- powered technology. Using this system, devices connecting to the network will take advantage of the inherent security that the network provides (rather than trying to ensure security at the device level) [9]. It left users privacy on the hand of effective processes and policies of the organization. The security of Cisco IoT system does not ensure device level security which is one of the crucial concerns for ubiquities computing. Inclusion of the security layer in Cisco IoT system clearly justify that there is a serious need of security protection for IoT systems. However their security solution does not address requirements of IoT systems such as openness, unification, device level security protection. Most importantly, Cisco security layer considered security of the IoT ecosystem but forgot the security at individual IoT domain, device and business service level.

Ray et al. proposed a security framework in [1] to combine multi-protocol in a framework. This framework is the most relevant work in the literature which aims to support diverse system for IoT. However, it doesn't have any scope to adapt a new protocol in it. In addition it does not show how a client and master reader communicate with each other within the system.

In this paper we extend the work from [1] and address the following:

- Multiple security protocol adaption with in the same framework
- Elimination of redundant security service execution for the system.

2.2 System Model

In this section, we present an IoT system model where our proposed multiprotocol framework can be used to secure the system. Our adapted IoT system for FP is shown in Fig. 1. As illustrated in Fig. 1, the system is connecting diverse actuator systems to ensure a global information network for objects. Each system of the IoT controlled by different administrative domains with a common system structure as shown in zoomed view in Fig. 1. This common system architecture has a master reader which coordinates all other readers (client readers) of the system. The master reader is responsible to represent the entire system to another system.

As illustrated in Fig. 1, the system model's master readers are communicating over Internet cloud. The authorized client readers will be able to execute the mutual authentication and exchange information of the object according to the need of the

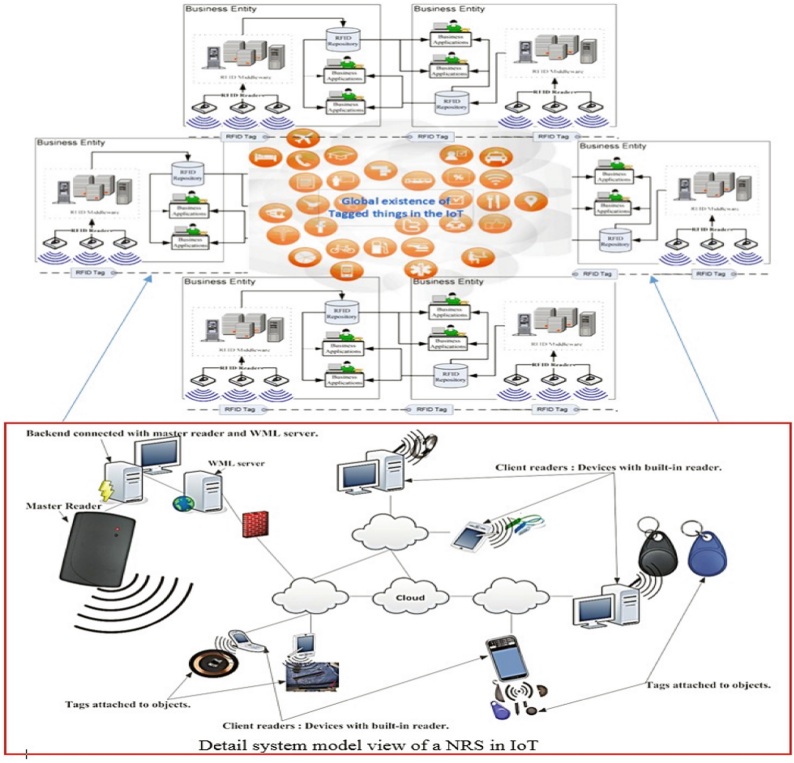

Fig. 1. Working system model for our updated FP

system. However, this information need to be transported to other system via master readers. The developed framework will have a client module in client readers to support the master module of master readers.

The client and master module of the framework coordinate each other to ensure security protection of the IoT system. It uses Security Check Handoff (SCH) to dynamically choose security services required for an object based on system's business requirement. To illustrate the workflow of the system better, we have reading process using our proposed framework in an IoT system illustrated in Fig. 2. As we can see in Fig. 2, master reader of a system can presents it to another system with in IoT network. It also contribute mainly in systems security management.

3 Proposed Framework and Techniques

In this section, we state our improved framework that will be the holistic unified security solutions for IoT. This is followed by the SCH technique to support improved framework. We also detail the working process of improved mutual authentication. We will only state detail of our improvement over contributions in [1].

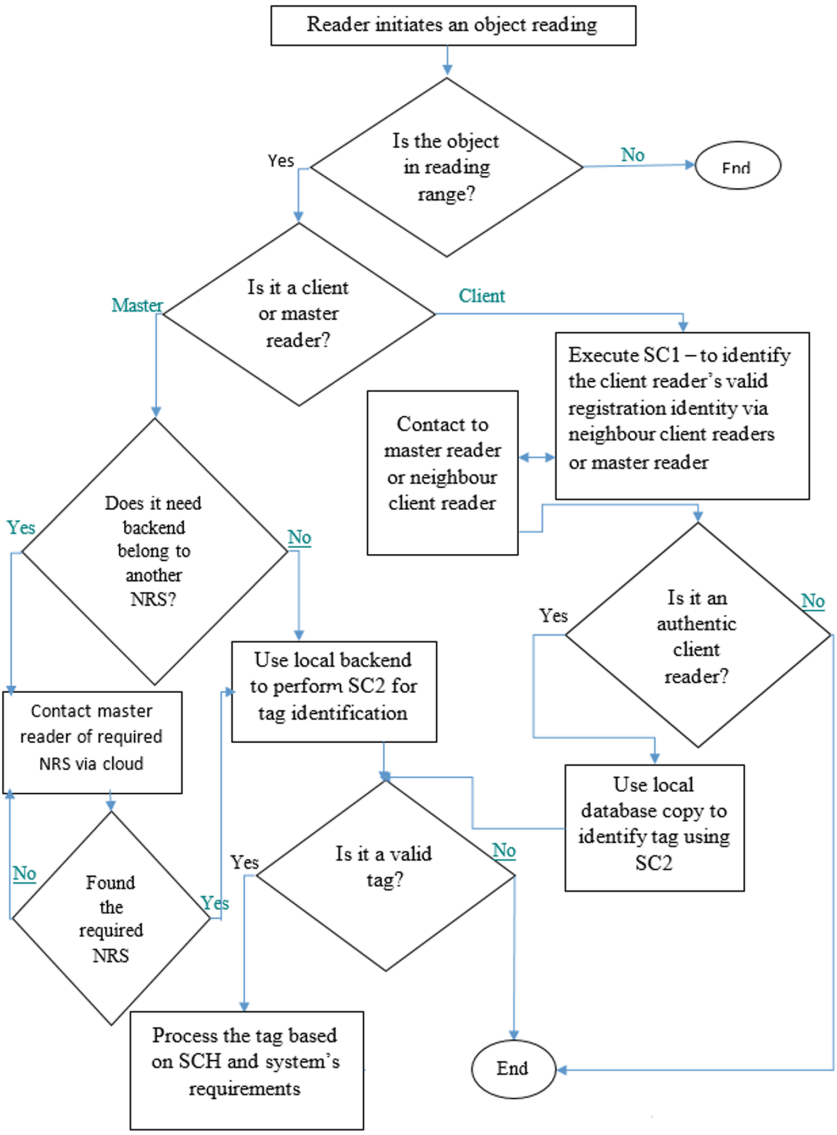

Fig. 2. A generic workflow of our system model that uses our proposed framework

3.1 The Framework

The updated framework is illustrated in Fig. 3. This framework is extending common services to all the required security protocols by adapting them in a unified framework. The framework has adapted System Components (SC) 1 to 4 from [1]. The functionality of each of these SCs will be same as detailed in contribution [1]. The execution of SC1, SC2 and SC4 ensures mutual authentication between readers and

actuators in the system domain to safeguard the system. The SC3 is a simple malware command checking layer that stop actuators to pass malicious command to the back-end. To integrate a security protocol for specific security requirement we add a new layer name integration layer. We present detail about the integration layer (SC_I^i) below.

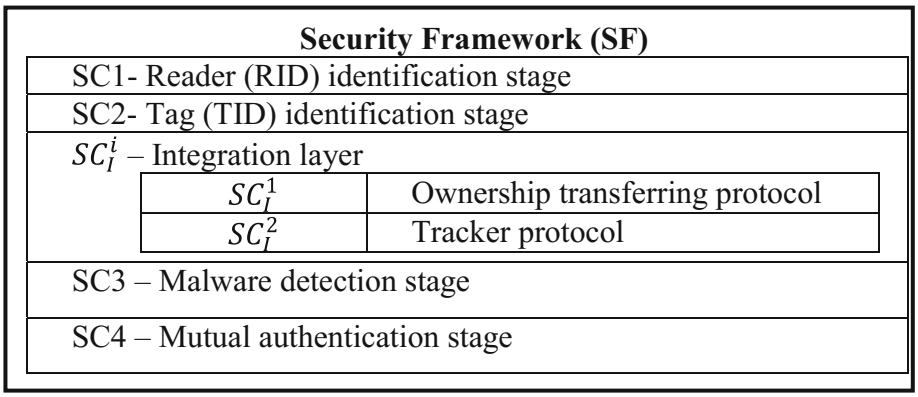

Fig. 3. The unified proposed framework

System Component $I(SC_I^i)$: is integration layer that accommodates additional security services that are specific to a business need. Our work integrates two business needs and their specific security services as shown in Fig. 3. Each sub-layer of integration layer SC_I is identified by a unique i value as superscript that makes the presenting symbol of this layer (SC_I^i). The i value is a sequential number for each integrated security protocol that uniquely identify the service of the integration sub-layers. An IoT system can add integration sub-layers based on their need. It can also dynamically choose which sub-layer to run for a particular execution using our updated SCH. However, security protocols need to satisfy two simple requirements below to be added to an integration sub-layer of SF.

- They have to support universal operation.
- They should not duplicate security services provided by SF in SC1 to SC4

The number of security protocols to be integrated in integration layer depend of system's requirements and need. This unified framework assumes that the system will have a list of authentic readers in backend along with actuators information. The integration layer manages the implementation of these integrated security protocols using SCH that is explained in the following sub-section.

3.2 SCH to Support the SF

In this section, we will present our improved SCH to work with our new integration layer. The SCH is the technique which not only provide faster security clearance but

also provide a mechanism to manage system components of SF. It coordinates and provides security services based on system's security and business need. The updated SCH has two system bits and identify bits to accommodate integration layer as illustrated in Fig. 3.

The security clearance bit type identify the security clearance status of the actuators. The integration bit type specify the integration requirement for the tag by the system.

The last SCH bits (Identity bits) only required if integration bit is "*ON*". The identity bits used to pass the identification value of the specific integration protocol requested to be executed by the system. The proposed SCH only allow one integration protocol to be executed by a specific execution at a time. We present the discussion below to comprehend different possible SCH bits combination a system might need in FP executions.

Let us first consider two system bits (security clearance bit and integration bit) so $SCH_b|_{size}(b_T) = 2\,bits$ where security clearance bit control security related system components (SC1, SC2, SC3 and SC4) and integration bit controls integration layer (SC_I^i) system components. Each bit has two states $ON(1)$ and $OFF(0)$ makes four maximum combinations when $SCH_b|_{size}(b_T) = 2\,bits$ as detailed below

- If $SCH_b = 0(OFF)0(OFF)$
  - In this situation security clearance bit is $0(OFF)$, the actuator is not subject to security clearance (need to execute SC 1 and SC 4). All actuator s will have its initial security clearance value is $0(OFF)$.
  - As integration bit is $0(OFF)$, the FP system does not require to execute any integrated protocol for the actuator.
- If $SCH_b = 1(ON)0(OFF)$
  - In this case security clearance bit is $1(ON)$, the actuator is subject to faster security clearance (need to pass SC 1 to SC 2). In most cases, after its initial identification the actuator sets its security clearance bit $1(ON)$.
  - However as the integration bit is $0(OFF)$, the FP system does not require to execute any integrated security protocols for the actuator.
- If $SCH_b = 1(ON)1(ON)$
  - In this event the security clearance bit is $1(ON)$ and also the integration bit is $1(ON)$ Therefore the actuator is subject to faster security clearance (need to pass SC 1 to SC2 only) and the FP system require to execute an integrated security protocol from its sub-layer. The selection of the integrated protocol will be based on identification bits of the SCH.
- If $SCH_b = 0(OFF)1(ON)$
  - Here the security clearance bit is $0(OFF)$, the actuator is not subject to security clearance (need to execute SC 1 and SC 4). This is ideal initial condition of the actuator.
  - However, the integration bit is $1(ON)$ therefore FP system require to execute an integrated security protocol from the listed protocols of its sublayer.

The identification of the integration layers protocols will be represented using superscripted i value of the integrated protocol. For an example, if the SCH value for an

actuator is $SCH = 1110$ then the system can conclude that the actuator's security clearance and integration bits both are in *ON* state. The system also can extract that the identification value that is 10 bits which is equivalent of decimal value 2. This means system should run integration protocol with superscripted i value 2.

The working process of the new SCH is illustrated in Fig. 4 which shows that if integration bit is *OFF* then identification bits are excluded from SCH to make it moderated size value. We store the random position value and SCH value in the backend to ensure intruders will not be able to exploit the system.

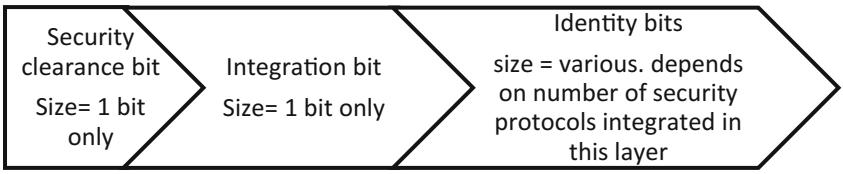

Fig. 4. Components of SCH

The SCH bit works as a bond of this framework to hold all the layers together and provide a means to dynamically execute system components based on systems status, requirement and security need.

3.3 Working Process of SF

This section detail working procedure of our proposed unified framework. The updated SCH is an integral part of the framework as it supports the framework to achieve its objectives. The framework works as a black-box so rest of the techniques used in the framework can be replace by an updated one if required. Similarly, the security protocols adapted in integration sub-layer can be also replaced by an updated one, if required. A new sub-layer can also be added to accommodate a new security protocol as required by the system (Fig. 5).

In the discussion of this section our proposed SF is supported by

- Improved SCH from Subsect. 3.2.
- Reader/actuators identification and mutual authentication techniques detailed in [1].

The SF uses 96 bits frame formats illustrated in Fig. 6 to detail its process. The frames have 5 bits header which carry information about the type of packet. There is a End of a Header (EH) field that is one bit equivalent null non-writeable space to separate header from payload. The payload portion carry actual data that is required to execute relevant operations.

The frame always has a CRC-32 value to handle error in transmission.

The five bits header information will represent type of packet based on their "*ON*" bits that is 1. The detail of each "*ON*" bits shown below:

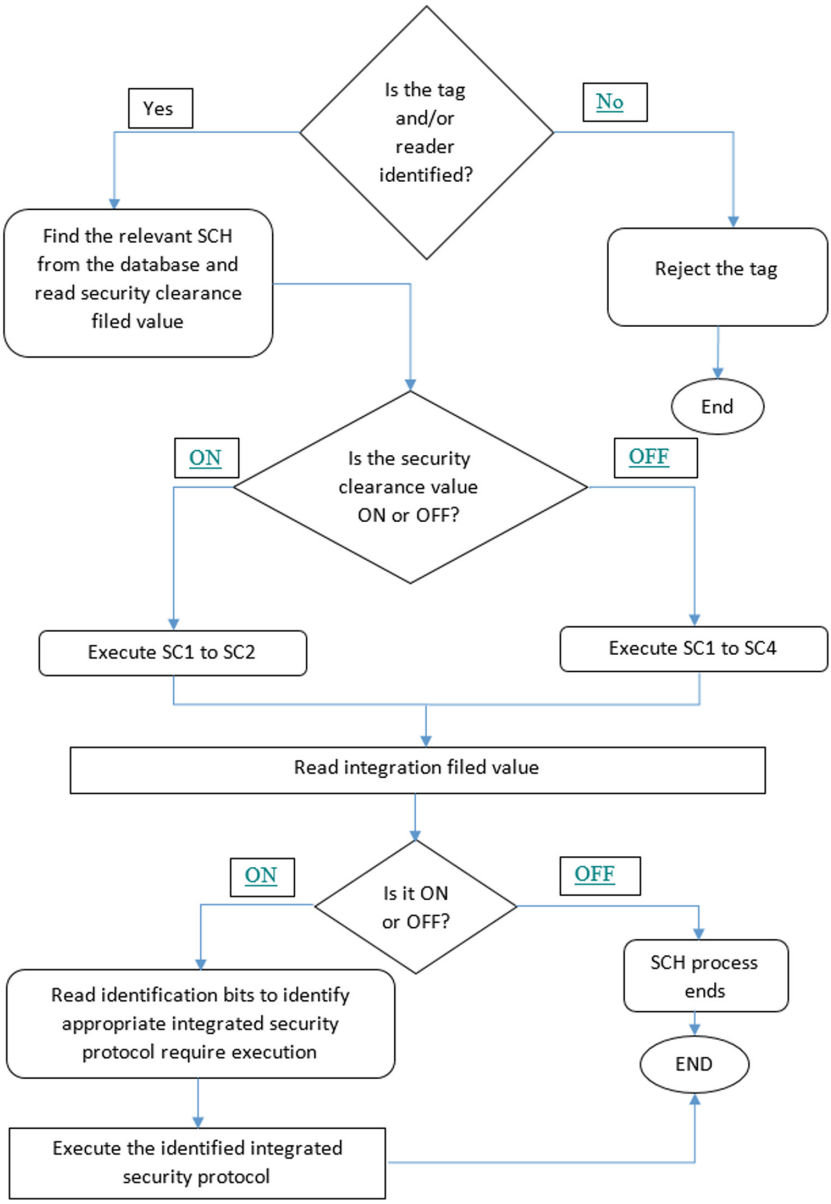

Fig. 5. Working flow of the updated SCH

Header	EH	Payload	CRC
5 bits	Null bit	90 bits	32 bits

Fig. 6. Generic FP frame format

10000 = Reader identification
01000 = Actuator identification
00100 = Malware detection
00010 = Mutual authentication
00001 = Integration layer's protocol execution

These packet type values are different than SCH status value as header bits only identify the type of packet. These values provide a degree of state full communication by keeping some state information of the security clearance. Let us detail a specific case to understand the working mechanism of our SF better. Here, we discuss updated details only that need to be consider in association to details from [1].

Let us assume that an object came to a client reader's vicinity that meet with the specification below:

- The actuator is read by the specific system for first time.
- It is in the range of a client reader.
- The system require to execute an integrated protocol for the actuator as the actuator is subject to an ownership transfer.

We choose above specification because it will allow us to detail most possible communication combinations of the proposed framework. The above detail specify that the system require to execute a SCH status where $SCH_b = 0(OFF)1(ON)[1_2]$ that means the object need to be processed by SC1 to SC4 and it also need to pass through integration layer to execute an integrated security protocol as integration bit is $1(ON)$. The last bit inside the square bracket is the integration protocol identification value. In this specific case, identification value is 1_{10} for ownership transfer protocol therefore SCH_b holds value 1_2 value to represent 1_{10}. The $SCH_b = 0(OFF)1(ON)$ value is default for all actuator s which are read for first time by a reader with an integration protocol execution request.

When the actuator come to the reader's vicinity, the reader sends a signal to the actuator for reading, using the frame format illustrates in Fig. 7.

11000	EH	Randomized reader's identification value		CRC
5 bits header	Null bit	l bits payload	90- l bits pad	32 bits

Fig. 7. First FP frame format

The frame of this first communication has a header value to specify that it is a reader and actuator identification packet. The frame has a randomized reader identification value that occupies 90 bits or less. If the l bits payload is lesser than 90 bits then a randomized padding value will be used.

The actuator then respond with a frame as illustrates in Fig. 8 below. In this frame actuators transport a 16 bits randomized actuator set value, a read count value of 10 bits and 64 bits hash value.

11000	EH	Randomized Actuator set value	Read count value	A hash value	CRC
5 bits header	Null bit	16 bits	10 bits	64 bits	32 bits
	90 bits payload				32 bits

Fig. 8. Second FP frame format

The system at this point initiate reader and actuator identification process.

As the actuator is read by a client reader of the system, it requires to send a communication to the master reader or neighbor client reader for authorization by sending randomized actuator set value and reader's identification. If the client reader is registered and authorized to process the actuator then the master reader will allow the client reader to do the rest of the process.

If the actuator is read by the client reader for second time then it execute the actuator identification process by itself without requiring authorization from anyone else. The whitelist of registered reader is used to check the reader's registration. The neighbor client readers are those who have a trusted relation with the respective client readers and registered. The communication process of this client reader's validation is illustrated in Fig. 9.

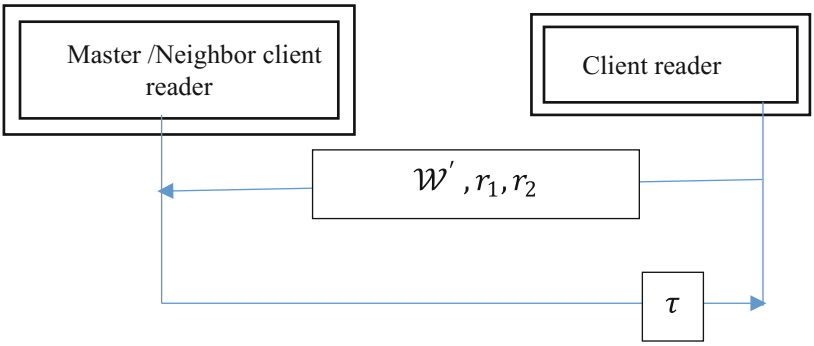

Fig. 9. Verify identity of a client reader to eliminate rogue readers

11000	EH	Token value	A hash value	unused	CRC
5 bits header	Null bit	16 bits	64 bits	10 bits	32 bits
		90 bits payload			32 bits

Fig. 10. SC4 FP frame format

Let r_2 be the randomized reader ID, $r_h = (h)(r_2)$ is the hashed value of r_2 protect RR_{DB} from disclosure, $\mathcal{W}'$ be the randomized actuator set value and r_1 is a l bit random number that is used to randomize readers ID and actuator set value. The client reader sends $\mathcal{W}'$, r_1 *and* r_h values to master reader/neighbor client reader to validate itself. The master reader then verify the readers ID using its registered client reader's white list and validity actuator set value $\mathcal{W}$.

$$h(r_2) \stackrel{?}{=} h(RR_{DB} \oplus r_1) \tag{1}$$

If Eq. (1) returns true and actuator set value $\boldsymbol{W}$ is within the valid range, the master reader sends client reader a random success token $\boldsymbol{\tau}$ which is calculated using Eq. (2).

$$\tau = \mathcal{W} \oplus RC'_T \tag{2}$$

In our updated SF, the client reader use this token value in mutual authentication stage to verify itself. If master reader cannot find reader ID in registered reader's database then it reject all communication from the reader.

After receiving the token τ, the valid client reader use actuator identification technique from [1] to identify the actuator. If the actuator ID is valid then it retrieve SCH value from the database. In our case this system is reading the actuator for first time and it requires to execute the ownership transferring process therefore the SCH value of the actuator will be $SCH_b = 011$. The detail of each bit of SCH_b value is shown below

0_2 = the actuator is not subject to security check handoff
1_2 = The actuator need to execute a integration layer protocl
$1_2 = 1_{10}$ = the actuator requres to execute the first integration layer protocol.

Based on SCH_b values, the reader execute SC3 for the actuator to identify any malicious command in the values transmitted by the actuator. If actuator's transmitted values are clean then it executes the mutual authentication process in SC4.

Otherwise it rejects all the communicated information. In addition to mutual authentication process detailed in [1], the client reader also send τ to validate its own identify. The new frame format of the communication of SC4 is illustrated in Fig. 10 which is an updated version of Fig. 11.

As we can see in Fig. 11, we are transmitting a token code and a hash value.

Because of this new token value, our updated mutual authentication process is shown in Fig. 11. As illustrated in Fig. 11, the master or a neighbour client reader send the verification token τ to the reader. The reader send this in its communication along with r_4 that constitute SC4 (mutual authentication) of the framework and protocol details in [1]. The actuator executed Eq. (4) of Fig. 11 to verify the identity of the reader before executing Eq. (5) as illustrated in Fig. 11. The token code and hash value let the actuator verify the validity of the reader. These values also let the actuator update its read count information and actuator ID, if required.

At the end of the successful execution of SC4, the actuator wait for a communication from it's current owner to execute ownership transfer protocol as specified in

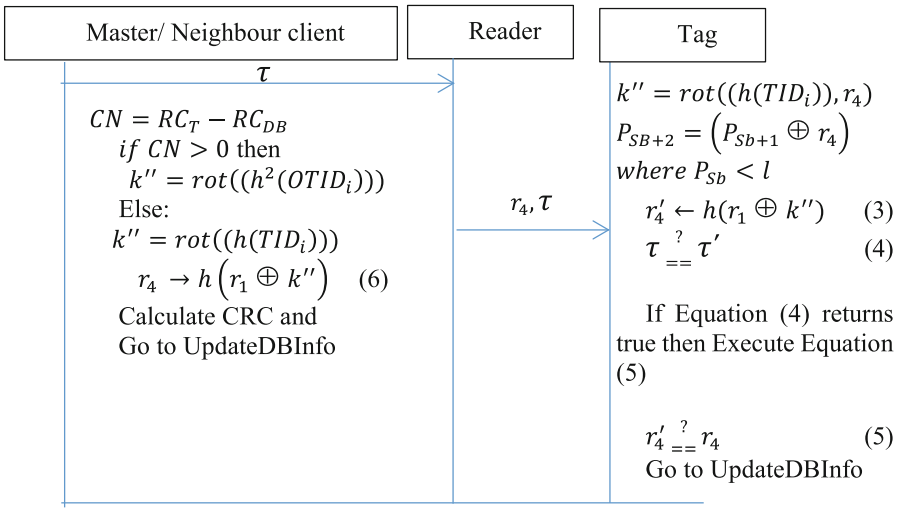

Fig. 11. Updated mutual authentication process for SC4

SCH which follow the execution detail discussed in [1]. The execution cycle remains incomplete until the ownership transfer is done. During this incomplete stage, the actuator does not communicate with any other owner except the one just executed mutual verification. The actuator gets back to normal state as soon as it finishes it's ownership transfer (Complete execution cycle) as specified in *SCH* value. The overall process flow of our unified framework is illustrated in Fig. 2.

4 Conclusion

In this paper, we proposed a security framework to ensure there is a scope for security protocols adaptation. The integrity layer of the protocol can adapt any number of security services and protocols to support the entire system. Our proposed holistic unified framework in this chapter is built on enhanced SC4 and SCH detailed in this paper. These improvements are done to ensure that our framework is ready to support unified security service requirements of IoT. The IoT has many administratively controlled domains as a result universality and unification of security and business services are crucial to increase acceptance of uses. This chapter detailed working process of our enhanced SF and updated packet formats required to achieve objectives of our study. The solutions proposed in this study are simple and can work with existing hardware. This study also considered the constraint of the actuator s computational capability to ensure high implacability of the SF.

References

1. Ray, B.R., et al.: Scalable RFID security framework and protocol supporting Internet of Things. Comput. Networks **67**, 89–103 (2014)
2. Ray, B.R., et al.: Secure mobile RFID ownership transfer protocol to cover all transfer scenarios. In: 7th International Conference on Computing and Convergence Technology (ICCCT) (2012)
3. Ray, B.R., Chowdhury, M., Abawajy, J.: Secure object tracking protocol for the Internet of Things. IEEE Internet of Things J. **PP**(99), 1
4. Dong Seong, K., Taek-Hyun, S., Jong Sou, P.: A security framework in RFID multi- domain system. In: 2007 ARES Second International Conference on Availability, Reliability and Security, pp. 1227–1234 (2007)
5. Roberti, M.: The history of RFID technology. RFID J. LLC (2005). http://www.rfidjournal.com/articles/view?1338
6. MacBean, N.: 'Electronic pickpocketing' looms as next threat in credit card fraud, police, security experts say (2014). http://www.abc.net.au/news/2014-05-30/electronic-pickpocketing-looms-as-next-credit-card-fraud-threat/5486806
7. Konidala, D.M., Kim, D., Yeun, C.Y., Lee, B.: Security framework for RFID-based applications in smart home environment. JIPS **7**(1), 111–120 (2011)
8. Lim, T.-L., Li, T., Yeo, S.-L.: A cross-layer framework for privacy enhancement in RFID systems. Pervasive Mob. Comput. **4**(6), 889–905 (2008)
9. Wang, et al.: New Cisco Internet of Things (IoT) System Provides a Foundation for the Transformation of Industries (2015). http://www.marketwatch.com/story/new-cisco-internet-of-things-iot-system-provides-a-foundation-for-the-transformation-of-industries-2015-06-29