IEEE International Conference on Smart Grid and Smart Cities (ICSGSC), Kuala Lumpur, Malaysia, August 2018.

Cyber Physical Sensors System Security: Threats, Vulnerabilities, and Solutions

Areej Essa, Tariq Al-Shoura, Ahmad Al Nabulsi, A. R. Al-Ali, and Fadi Aloul
Department of Computer Science and Engineering
American University of Sharjah
United Arab Emirates
{G00075082, B00074137, aalnabulsi, aali, faloul}@aus.edu

***Abstract*—A Cyber Physical Sensor System (CPSS) consists of a computing platform equipped with wireless access points, sensors, and actuators. In a Cyber Physical System, CPSS constantly collects data from a physical object that is under process and performs local real-time control activities based on the process algorithm. The collected data is then transmitted through the network layer to the enterprise command and control center or to the cloud computing services for further processing and analysis. This paper investigates the CPSS' most common cyber security threats and vulnerabilities and provides countermeasures. Furthermore, the paper addresses how the CPSS are attacked, what are the leading consequences of the attacks, and the possible remedies to prevent them. Detailed case studies are presented to help the readers understand the CPSS threats, vulnerabilities, and possible solutions.**

Keywords—Cyber Physical Sensor Systems, Smart Sensor, Cyber Security, Smart Cities, Smart Grid

I. INTRODUCTION

Cyber Physical Systems (CPS) today play a major role in industry 4.0 applications such as smart factory, smart energy, smart transportation, smart building, smart health and smart cities. A conceptual model of a CPS has four layers that can handle a physical object. The layers are Cyber Physical Sensor System (CPSS), networking, applications, and security. The CPSS layer consists of sensors, single chip computing platforms, controllers, and actuators that are interfaced with a physical object. The network layer provides the link between the CPSS and the application layer utilizing different internet of things (IoT) communication protocols. The application layer handles the collected data storing, processing, analyzing, visualizing and user interfacing services. In addition to that, there is a security layer that overlaps with all three layers [1-3]. Fig. 1 shows an overview of the CPS conceptual model.

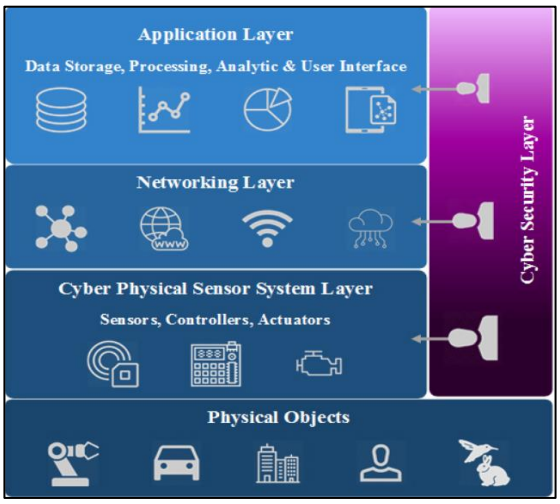

Fig. 1. Shows the CPS conceptual model

The CPSS collects the physical object status and transmits it to the application layer through the network layer. The data is then processed and analyzed in the application layer. The outcome will be transmitted back to the physical layer in the form of commands to actuate the physical object accordingly. Each of these layers is vulnerable and prone to cyber-attacks that can cause some disturbance or malfunction resulting in severe consequences [4-7]. The paper discusses the CPSS functions, security threats and vulnerabilities, and provides possible solutions to these threats.

The rest of the paper is organized as follows. The next section gives background of CPSS. Section III describes the various attacks and countermeasures in CPSS. Detailed case studies are also presented. Finally, Section IV concludes the paper.

II. CYPER-PHYSICAL SENSOR SYSTEM

A simplified CPSS module consists of sensing elements, controller with built-in Wi-Fi and/or Ethernet access points, and actuators. The sensing elements are designed to provide highly reliable results requiring less maintenance at a low cost. A self-description model of the sensing element in CPSS consists of three elements; the basic information, the measuring, and the skills, as shown in Table 1 [3]. Examples of sensing elements include temperature, gas, pressure, flow, speed and humidity sensors and the later is taken as part of the case study in this paper. The basic information is a set of attributes that help to identify the basic sensor information such as the function *id()* that retrieve the ID, *enable()* to know whether the sensor is currently enabled or disabled, *ready()* to confirm whether is it prepared to read data and *location()* to indicate where it is currently located. The measuring element of a sensor is a set of functions that provide information on the sensor's readings. For example, the function *response()* calculates the response time of the signal output and other useful information about the signal. The third, skills, is a set of functions that do additional operations on the sensor readings and permissions on the sensor functionality. For example, the function *convert()* converts the readings to other forms before transmitting them. Table 1 illustrates the basic information, measuring and skills elements of a smart sensor.

TABLE I. CYBER-PHYSICAL SENSOR SELF-DESCRIPTION MODEL [3]

Self-description model		
Basic information	***Measuring***	***Skills***
id(): int	linearity(): int	convert(): bool
enable(): bool	resolution(): int	calibrate(): bool
dataread(): bool	responsetime(): int	correction(): bool
location(): x, y, z = double	repeatabiltyaccuracy(): int	correctionfactor(): int

III. Attacks on CPSS and Remedies

Cyber-Physical Sensor Systems are vulnerable to various types of attacks. Four major types of cyber security attacks will be discussed, mainly: packet injection, man-in-the-middle, impersonation, and denial of service attacks. The possible implication of these attacks on the CPSS is discussed, and some of the remedies to protect against the attacks will be addressed.

A. *Packet Injection Attacks*

1) *Definition*

Packet injection is a computer network security term that refers to the fabrication/alteration of packets in a way that constructs the packets to appear as if they were part of the normal communication [4]. In Fig. 2, a packet injection attack is shown where an attacker alters a bit value in a packet.

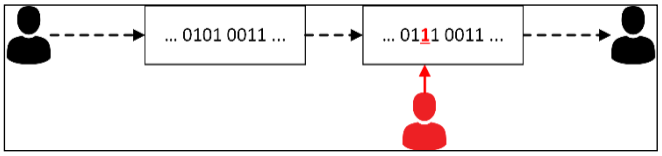

Fig. 2. Packet Injection attack concept

2) *Implications on Cyber-Physical Sensor Systems*

In CPSS, packet injection attacks could cause various threats by attacking the different sensor attributes. In this section, we will discuss examples of such threats if four of the attributes; ID, resolution, location, and humidity has been altered.

- 0Fig. 3 illustrates an attack that alters the sensor's ID. Here, the sensor adds its ID (12)10 in the binary format (1100)2 to the packet to be sent to the server. The attacker alters a bit in the packet causing a change in the binary representation of the ID to (1110)2. Changing the ID will cause the server to assume that the data has been collected from a different sensor (14)10, which could be the cause of a wrong actuation in some cases, and if the attack on the ID is persistent, this could cause the server to assume that the desired sensor (12)10 is offline or faulty.

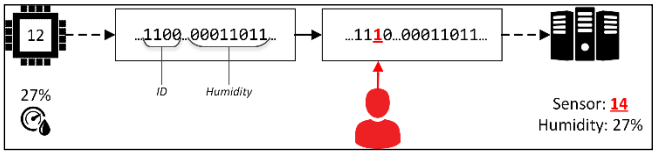

Fig. 3. Packet Injection attack on the sensor's ID

- An attack that alters the sensor's resolution is presented in Fig. 4. The sensor indicates the resolution that the ADC is currently operating on (10)10 bits in the binary format (1010)2, and adds it to the packet to be transmitted, the attacker alters a bit that represents the resolution, this will cause the second party to interpret the data collected from the sensor differently and assumes the resolution to be (14)10 bits. Thus, the second party will obtain a faulty data, this could cause a wrong actuation in some cases, or improper modeling of the system as the different resolution would lead the server to do the required computations based on a different sampling rate.

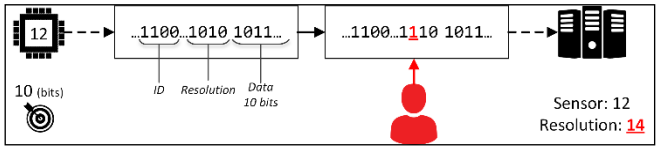

Fig. 4. Packet Injection attack on the resolution data

- In Fig. 5, the value of the location (52)10 has been shared in a Binary Coded Decimal (BCD) format (0101 0010)2 and added to the packet to be transmitted, and while the packet is being sent, an attack alters the sensor's location data occurs, leading to the improper modeling of the sensor's environment and could cause a false actuation if the actuation algorithm takes into consideration the sensor physical location.

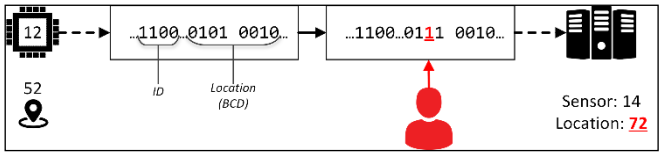

Fig. 5. Packet Injection attack on the location data

- An attack that alters the sensor's humidity, as shown in Fig. 6, will lead the second party to presume a different humidity value other than the measured one, leading the second party to make decisions based on false data and accordingly to false actuation. Fig. 6 illustrates the scenario where the sensor incorporates the binary value of the humidity (27)10 as (00011011)2 into the packet and transmit it, where the attacker manipulates a bit that represents the humidity, making the value (00001011)2, causing the server to assume that the humidity at the sensor's side is 11%.

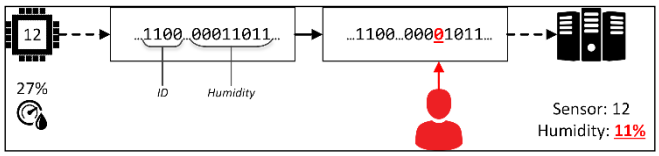

Fig. 6. Injection attack on the humidity

3) *Packet Injection Attack Remedy*

One of the remedies against injection attacks is to validate that a packet has not been the victim of an injection by introducing a *cryptographic checksum*. A cryptographic checksum is a small data block that is derived from the original packet's data. The checksum will aid in detecting any errors or manipulations that may have occurred during the packet transmission. The sender applies an algorithm to generate the checksum and appends the checksum at the end of the message to be transmitted. The receiver can ensure the message integrity by applying the same algorithm and checking the generated checksum against the received checksum. If the two do not match, then it can be deduced that an alteration to the packet has taken place during transmission and the packet may be dropped [5].

The cryptographic checksums algorithms are divided into two categories: Keyless cryptographic checksum, and Keyed cryptographic checksum. The Keyless cryptographic checksum requires no keys to generate the checksum; these algorithms include MD5 and SHA-1. Keyed cryptographic

checksum, however, uses keys to generate the checksum, one example of such is DES [6] [7].

Fig. 7 illustrates an example of algorithms using the modulus of *16* to generate a checksum. The sensor will add the ID to the humidity value and take the modulus of the summation as an input to generate the checksum. The checksum will be appended with the transmitted data. As can be seen, the value of the ID has been altered while in transmission, however, the receiver can detect the alteration by comparing the received checksum and comparing it to the expected checksum according to the received data and possibly dropping the corrupted data packet.

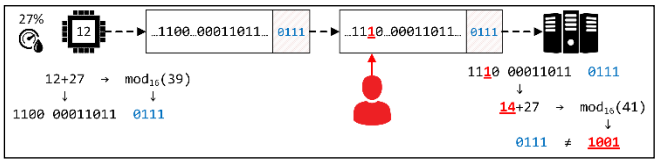

Fig. 7. Packet Injection attack remedy using checksums

B. *Man-in-the-Middle Attack*

1) *Definition*

Man-in-the-middle attack refers to the type of attacks where the attacker is positioned in a conversation between two parties. The attacker can be passively eavesdropping to the communication to obtain confidential information which is known as sniffing or by actively trying to resend packets that have been obtained through sniffing which is known as replay attack [8].

2) *Implications on Cyber-Physical Sensor Systems*

Sniffing attacks implications depend on the sensitivity of the data being transmitted. An example of such would be the could be the sensor's location. As shown in Fig. 8, different information could be transmitted without the need for confidentiality like the sensor's ID.

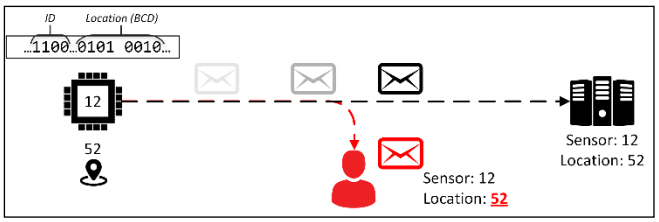

Fig. 8. Sniffing attack

In replay attacks, the attacker resends packets that have been captured previously, to the targeted distention as a legitimate packet. In Fig. 9, the attacker replays the packet that has been sniffed in the previous example; the second party acknowledges the packet as legitimate and registers the faulty location value.

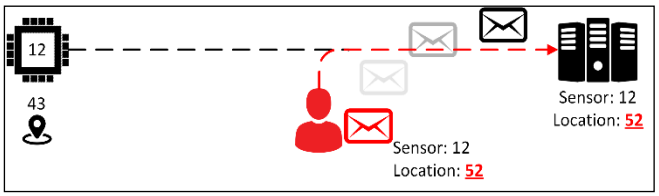

Fig. 9. Replay attack

3) *Man-in-the-Middle Attack Remedy*

One of the remedies against Sniffing attacks is the *encryption* of data. Encryption is the process of converting readable data, often known as *plain-text*, into a version that can only be understood by the targeted party, commonly known as *cipher-text*, by converting the data back into plain-text. Some of the known encryption algorithms are Triple DES, RSA, Blowfish, and the Advanced Encryption Standard (AES) [9].

Fig. 10 illustrates an example of the encryption of the location value:

1. The sensor converts the ID (12)10 to its binary representation (1100)2 and the current location (52)10 into its BCD value (0101 0010)2.
2. The sensor encrypts the location value to become (1010 0001)2 using a symmetric key known by the server.
3. The sensor adds the ID and the encrypted location value to the packet and transmits it.
4. When the packet value is sniffed by the attacker, the attacker is not able to obtain the current location of the sensor.
5. The server decodes the value (1010 0001)2 using a symmetric key known by the sensor and retrieves the actual location of the sensor (52)10.

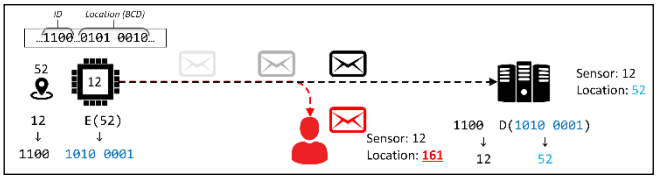

Fig. 10. Sniffing attack remedy

Even though encryption hides the data and protects it from any intrusion, encrypted packets can still be replayed and cause a problem. One of the remedies against Replay attacks is the use of counters that will indicate the next packet ID-number as illustrated in Fig. 11:

1. Both the sensor and the server know that the next packet to be exchanged should be identified with an index of 3, accordingly, the sensor incorporates the (3)10 as (0011)2 into the packet and transmits it.
2. The attacker sniffs the transmitted packet, and at the same time, the server acknowledges the current value sent as the message index (3)10 matches the expected message index by the sensor and both the sensor and the server increments the next message index to (4)10.
3. The attacker replays the same message sniffed in the previous step to the server.
4. The server rejects the replayed message as the replayed message index (3)10 does not match the expected message index (4)10.
5. When the new location is to be sent, the sensor sends incorporates the new message index (4)10 to the packet (0100)2.
6. The server checks the message index (4)10 and acknowledges the new value sent.

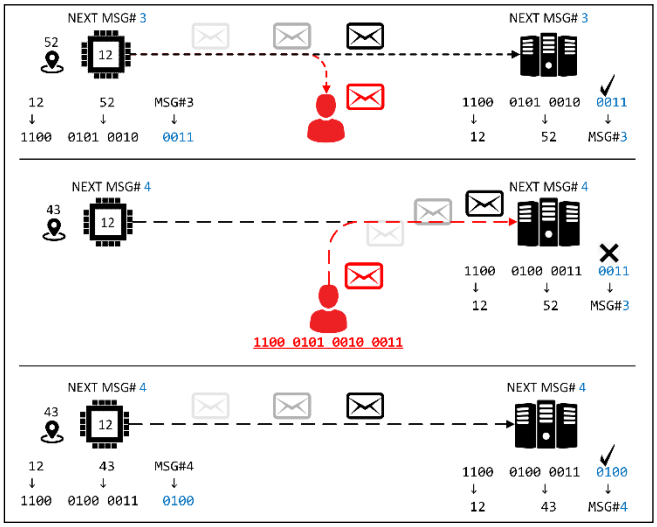

Fig. 11. Replay attack remedy

Other Replay Remedies include adding a timestamp to the packet, utilization of session identifiers, One-time passwords (OTP), and Kerberos [10].

C. Impersonation Attack

1) Definition

In this type of attack, the attacker steals the identity of a legitimate party claiming to be that they are that specific party. The consequences of such an attack have a significant impact because the attacker can be granted access to systems and confidential data. Moreover, when the attacker is given the privilege, they can execute illegal commands, cause a system to shut down if the attacker was impersonating an admin, or listen to a communication between two parties [11]. This type of an attack is an attack on the authentication factor, and reliable measures need to be taken to avoid it.

2) Implications on Cyber-Physical Sensor Systems

Impersonation attack can be executed on the Cyber-Physical Sensor Systems by either impersonating the smart sensor or impersonating the server. In Fig. 12, a demonstration is shown where an attacker impersonates the smart sensor to the server. In this scenario, the attacker was able to impersonate the smart sensor by sniffing its ID. Here, the actual humidity of the environment is 51%. The attacker will construct a packet with the same ID of the desired smart sensor and will transmit a humidity value different to from the actual humidity. The transmitted packet holds a humidity value of 27%, and this wrong value will lead the server to the improper modeling of the system causing the initiation of a wrong actuation. In this case, the attacker will be able to disturb the functionality of the system and perform malicious actions.

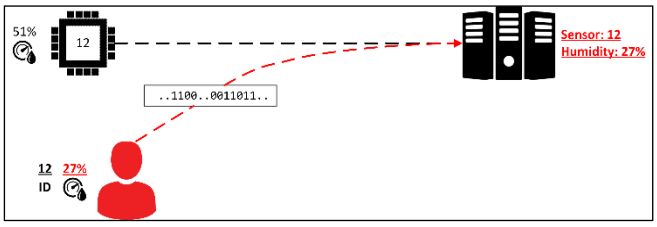

Fig. 12. Impersonation attack

3) Impersonation Attack Remedy

In the literature, there exist several methods that prevent impersonation attacks. In this paper, we will address a well-known and significant method to protect against this attack which is based on Public-Key Cryptography [12]. In this type of cryptography, each party will contain two keys; a public key and a private key. The public key and the private key are generated from each other; however, the knowledge of one key cannot extract the other. The public key is disclosed and known to the public. While the private key is only known by the owner of the key pair. These keys have a unique feature as any message encrypted using any of the keys can only be decrypted using the other key, i.e., if a party encrypts a message using the private key, the public key must be used to decrypt the message and vice versa. This feature is what allows the authentication of the sender and the confidentiality of the data to be maintained.

The concept of this encryption and decryption through the public and private keys will be used to protect against impersonating the smart sensor. The packet transmitted from the smart sensor to the server will be encrypted using the private key of the smart sensor. Since a packet that has been encrypted using the private key can only be decrypted using its known public key, then the server can decrypt it using the public key of the smart sensor. If the packet correctly decrypts, the server can authenticate the sender (smart sensor). This is also known as digitally signing the packet. In Fig. 13, we demonstrate how the server authenticates the smart sensors. The scenario is as follows:

1. Actual humidity: (51)10 = (00110011)2
2. Packet before encryption: [.. 1100 .. 00110011 ..]
3. Packet encrypted using the private key of the smart sensor (PR_a): [0101 … 10100011 …]
4. At the server, decrypt using the public key of the smart sensor (PU_a): [1100 … 00110011 …]

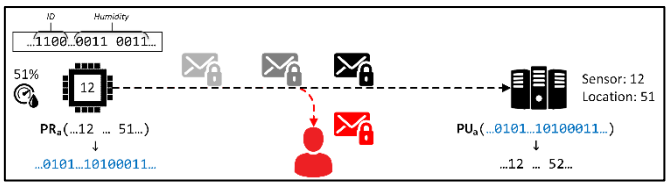

Fig. 13. Impersonation attack remedy

Hence, the server can authenticate when a packet is received from the smart sensor before executing any commands.

D. Denial of Service Attack

1) Definition

Denial of Service attack involves flooding a system by traffic aiming to make the system break down entirely, work in less capacity, or fail to serve on time. Moreover, the Distributed Denial of Service (DDoS) also achieves the attack on resource availability by involving multiple compromised systems attack multiple resources, such as the end devices and the network, at the same time.

2) Implications on Cyber-Physical Sensor System

In the context of the cyber-physical systems, DoS attack is performed on the server by flooding it with traffic that is identical to the traffic of the smart sensor. It can also occur

by flooding the smart sensors with traffic identical to the traffic of the server. Consequently, the smart sensor or the server becomes unresponsive to any new incoming packets.

In Fig. 14**,** an illustration of a DoS attack on the smart sensor is shown. Here, the attacker will impersonate the server and flood the smart sensor with legitimate looking traffic. The smart sensor buffer will overflow and consequently become unresponsive. Moreover, this will disable the smart sensor from receiving any new requests from the actual server.

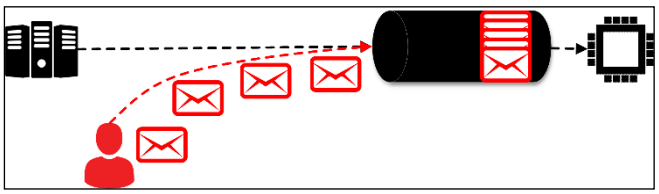

Fig. 14. Denial of Service attack

3) Denial of Service Attack Remedy

In [13], the authors propose a methodology for mitigating DoS attacks. The proposed method involves two modules: an attack detection module and a packet filtering module. The attack detection module filters the packets in order to extract its characteristics. An example of a characteristic to be used in DoS attack detection in-valves the source IP address. and the number of times the packet has been received per time. After the characteristics have been extracted, this information is used by the second module, the packet filtering module, to filter malicious packets. In the literature, several techniques have been proposed that aims to detect and mitigate DoS attacks. One of the simplest types of filtering modules involve checking the number of requests made per time, often called the threshold. If this threshold has reached, any requests from the source will be rejected. In Fig. 15, we present how the Denial of Service attack is prevented by setting a counter for the number of received packets. Every time the smart sensor receives five packets from the source (server), it will reject any other packets coming from that source for a given period of time [13].

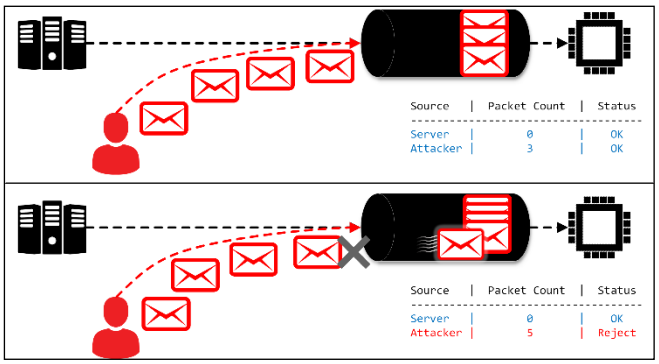

Fig. 15. Denial of Service attack remedy

IV. Conclusion

The paper presented a conceptual model for a generic cyber physical system with its layers. A brief description and function of each layer were introduced. The Cyber Physical Sensors System security threats and vulnerabilities were explored in an elaborated manner. Comprehensive case studies with examples to illustrate threats and vulnerabilities were presented along with proposed solutions. When designing, implementing and operating CPS, it is highly recommended that security of CPSS issues must be taken with high priority care to prevent systems failure.

References

[1] V. Navickas, S. Kuznetsova and V. Gruzauskas, “Cyber-Physical Systems Expression in Industry 4.0 Context,” in *Financial and credit activity: problems of theory and practice*, vol. 5, no. 23, pp. 188-197, 2017.

[2] E. Song, G. FitzPatrick and K. Lee, “Smart Sensors and Standard-Based Interoperability in Smart Grids,” in *IEEE Sensors Journal*, vol. 17, no. 23, pp. 7723-7730, July 2017.

[3] C. Berger, A. Hees, S. Braunreuther and G. Reinhart, “Characterization of Cyber-Physical Sensor Systems,” in *Procedia CIRP*, vol. 41, pp. 638-643, 2016.

[4] J.-W. Kang, I.-Y. Joo and C. Dae-Hyun, “False Data Injection Attacks on Contingency Analysis: Attack Strategies and Impact Assessment,” IEEE Access, vol. 6, pp. 8841-8851, February 2018.

[5] F. Cohen, “A cryptographic checksum for integrity protection,” in *Computers & Security*, vol. 6, no. 6, pp. 505-510, December 1987.

[6] H. Michail, G. Athanasiou, G. Theodoridis, A. Gregoriades and C. E. Goutis, “Design and Implementation of Totally-self Checking SHA-1 and SHA-256 Hash Functions’ Architectures,” in *Microprocessors and Microsystems*, vol. 45, pp. 227-220, 2016.

[7] H. Tang, Q. Sun, X. Yang and K. Long, “A Network Coding and DES Based Dynamic Encryption Scheme for Moving Target Defense,” in *IEEE Access*, vol. 6, pp. 26059-26068, May 2018.

[8] S. Prowell, R Kraus, and M. Borkin, “Seven deadliest network attacks,” Syngress, 1st edition, pp. 101-120, 2010.

[9] M. Rhee, “Asymmetric Public-Key Cryptosystems,” in *Wireless Mobile Internet Security*, 2nd edition, Wiley, May 2013.

[10] Z. Tbatou, A. Asimi, Y. Asimi and Y. Sadqi, “Kerberos V5: Vulnerabilities and perspectives,” in *Proc. of the 3rd World Conference on Complex Systems* (WCCS), Morocco, 2016.

[11] D. Bala, S. Maity and S. Jena, "Mutual Authentication for IoT Smart Environment Using Certificate-less Public Key Cryptography,” in *Proc. of the 3rd International Conference on Sensing, Signal Processing and Security* (ICSSS), India, 2017.

[12] S. Ur Rehman, K. Sowerby and C. Coghill, “Analysis of Impersonation Attacks on Systems using RF Fingerprinting and Low-End Receivers,” in *Journal of Computer and System Sciences*, vol. 80, no. 3, pp. 591-601, 2018.

[13] A. Nur and M. Tozal, “Defending Cyber-Physical Systems against DoS Attacks,” in *Proc. of the IEEE International Conference on Smart Computing* (SMARTCOMP), USA, May 2016.