Security of the sensory data in the cloud

F A Tsvetanov1 and M N Pandurski1

1 Department of Communication and computer engineering and technology, South-West University, Blagoevgrad, 2700, Bulgaria

E-mail: fttsvetanov@gmail.com

Abstract. The integration between the clouds and the sensor networks makes it possible to solve the problem of the limited computing power and the limited capacity of the sensor networks, as well as for the storage and processing of the collected data, without unnecessarily increasing the cost of the sensor networks. A significant challenge is the security of the data transmission connection between the sensors and the cloud. An algorithm has been proposed that ensures security in the transmission of data from the sensor network to the cloud via the DLS security protocol. The algorithm is implemented through the ThingSpeak cloud, and the connection between the sensor network and the cloud is implemented through the MQTT protocol, creating the flow that will gather sensor data, using Node-RED and Generate a self-signed certificate with OpenSSL. The algorithm was tested with Wireshark software, which confirmed that the data packets sent from the sensor network to ThingSpeak use the MQTT data transfer protocol and connect to communication port 8883, which is protected/encrypted via TLS / SSL protocol.

1. Introduction

A typical wireless sensor network contains many spatially distributed, self-regulating sensors that jointly monitor and measure various environmental parameters. The sensors provide a variety of useful data that can be used to monitor and control the environment in which they are located. The amount of data generated by sensor networks is enormous, heterogeneous and multidimensional in nature. Significant hardware storage and computing power are required to store and process this data [1].

Unlike traditional networks, WSN has its limitations in design and resources. Resource constraints include limited power, short communication range, low bandwidth, limited processing, and storage in each sensor node. Design limitations depend on the application, which is based on the observed environment [2]. Designing an additional ability to process the collected data leads to an increase in the cost of the sensors. Any increase in the price of a single sensor has a multiplier effect on the total cost of the sensor network, as the sensor network can contain hundreds or even thousands of sensors.

Cloud computing provides tremendous computing power and storage space. Cloud computing allows systems and users to use the platform as a PaaS service, infrastructure as an IaaS service, and software as a SAAS service at a very low cost. These services are provided by various cloud providers most often for a fee. The integration between the clouds and the sensor networks is a great opportunity and solution to the problem of the limited computing power and the limited capacity of the sensor networks, for storage and processing of the collected data, without unnecessarily increasing the cost of the sensor networks. The implementation of this integration has formulated a new paradigm called Sensor Cloud Computing [3].

One of the most important barriers to cloud acceptance is security, which is closely linked to legislative and technical components in terms of ensuring data protection. Traditional security
mechanisms such as identity, authentication and authorisation are no longer sufficient for clouds in their current form [4]. The challenges of security and privacy in the cloud require further research. Researchers from academia, industry and standards organisations continue to work and provide potential solutions to these challenges [5], [6], [7], [8].

In this paper, an algorithm is proposed to increase the security in the transmission of data from the sensor network to the cloud via the SSL security protocol. The operability of the algorithm is checked and proved.

2. Algorithm for increasing the security of transmitted sensor data in the cloud

The algorithm of figure 1 presents one of the possible variants for increasing the security of the transmitted sensor data via TSL protocol. This variant includes a ThingSpeak cloud, MQTT protocol. We create a stream that collects sensor data using Node-RED. The protection of the connection from the sensor network to ThingSpeak is realised by generating a self-signed certificate with OpenSSL. The arguments for choosing this algorithm are given below.

![Algorithm for the security of sensor data via TSL.](image)

**Figure 1.** Algorithm for the security of sensor data via TLS.

2.1. Arguments for choosing the algorithm

2.1.1. Selecting Cloud

To upload the sensor data, we select the ThingSpeak cloud [9], which is an open platform for IoT data based on public cloud technology. ThingSpeak enables real-time data collection, analysis and activation with the Open API (HTTP or MQTT). With applications and plugins, there is the ability to store data, visualise, monitor and integrate user data with various third-party platforms, including leading IoT platforms such as ioBridge, Arduino, Raspberry pi. Twilio, Twitter, ThingHTTP,
MATLAB. Sensor data is collected in each channel, which has eight fields that can contain any type of data, three location fields and one status field [10].

2.1.2. Select a protocol for connecting the network to the cloud and sending data
The MQTT protocol is selected, which is designed mainly for connecting devices with limited power over wireless networks and is used for publishing / subscribing. It is considered one of the main protocols in the implementation of IoT. MQTT is used to send short messages to and from a broker. ThingSpeak is used as an MQTT broker so that devices can send messages to ThingSpeak. Messages may contain the current office temperature collected by a sensor, etc. ThingSpeak accepts the message and stores its contents in a ThingSpeak channel. Once the data is in the channel, it can be easily visualised and analysed with MATLAB code. The protocol works with TCP / IP sockets or WebSockets. MQTT over WebSockets can be secured with SSL. The publish/subscribe architecture allows messages to be moved to client devices without the device constantly connecting to the server [11]. MQTT supports various mechanisms for data authentication and security. It is important to note that these security mechanisms configured on the MQTT broker, and the client must adhere to the existing mechanisms. There are three ways a broker can verify the identity of an MQTT client: client IDs, usernames and passwords, and client certificates. TLS (SSL) protocols can be used to protect the content of MQTT sensor data messages.

2.1.3. Choosing a security protocol and hardware realisation
The choice of this protocol determined by the capabilities of the selected hardware implementation. The result of a study by the company RF Desk security solution shows that there are two possibilities [12]:

- **First**, by selecting a computer motherboard with a chip that allows you to store cryptographic keys used for encryption. Many computers have Trusted Platform Modules hardware chip. Веднъж активиран, Trusted Platform Module предоставя пълни възможности за криптиране на диска, и поддържа защитен режим на свързане на SSL сокет за комуникация с брокера ThingSpeak MQTT.

- **Second**, the use of the external hardware security module (HSM). The module uses TCP / IP and is added to the management system, generating and securely storing cryptographic keys.

We choose the Transport Layer Security (TLS) protocol because it protects the data that is transmitted online between a web browser and a website via HTTPS and because it ensures that the data retains its original integrity and ensures confidentiality through encryption [24]. TLS security is part of the TCP / IP protocol, not MQTT. TLS operates over other transport protocols such as TCP, which means it is over the fourth layer of the OSI model. It is generally used as a transport layer, but in some cases may be part of the presentation or application layers. TLS security provides an encrypted channel using asymmetric cryptography for the personal message authentication code, which increases the reliability of the MQTT message—protecting all parts of the MQTT message, not just the payload of the message. At this stage, the TLS protocol is the most secure when transmitting data online [13].

2.2. Security Configuration with certification (authority) keys
2.2.1. Create the flow that will gather sensor data, and pass it to Thing Speak cloud via MQTT protocol, using software Node-RED.
Node-Red provides both MQTT nodes for a subscription (login) and publishing (login). We secure the connection via TLS / SSL encryption [14]. The nodes we use are DHT sensors, functional nodes (which use msg.payload to send data), MQTT output node, which connects to the server/cloud mqtt.thingsspeak.com. It publishes in the topic about the channel temperature/humidity in our ThingSpeak accounts and accordingly channels fig.2.
2.2.2. Connecting to an MQTT Broker/Server and Publish Using The Publish Node

To connect to an MQTT broker/server, enter the IP address or name of the Broker and the port we are using (default 1883). The MQTT broker offers encryption and username/password authentication. When a client posts a message to the Broker, he sends the subject of the message and QoS.

2.2.3. Security settings

- **Generate a self-signed certificate**
  
  When using our own MQTT broker, we use software such as OpenSSL (which is an open-source cryptographic tool for TLS / SSL protocols) to protect the connection. For this purpose, we create a self-signed certificate using open-SSL.

- **Cloud protection**:
  
  An arbitrary username and MQTT API key provided to connect to ThingSpeak via MQTT. The MQTT URL broker used mqtt.thingspeak.com. To configure the MQTT client and communicate with the ThingSpeak MQTT broker, we must have one of the connection options shown in table 1:

| Port  | Connection Type | Encryption |
|-------|-----------------|------------|
| 1883  | TCP             | None       |
| 8883  | TCP             | TLS/SSL    |
| 80    | WebSocket        | None       |
| 443   | WebSocket        | TLS/SSL    |

By default, the MQTT broker does not use activated security but uses port 1883 for unencrypted communication. To use the ThingSpeak TCP security configuration, we change the default protected port for MQTT to 8883. The monitoring of the sensor data in ThingSpeak MQTT is shown in figure 3.
3. **Check the efficiency of the algorithm to transmit encrypted data**

The validation of the algorithm for data encryption was performed using the Wireshark open-source software for monitoring and analysing sent/received data packets and analysis of network protocols. Wireshark is a tool for detecting (sniffer (a computer program that recognises and records a variety of restricted information, especially the secret passwords needed to gain access to files or networks)) and analysing packages. It captures network traffic on the local network and stores this data for offline analysis from Ethernet, Bluetooth, Wireless (IEEE.802.11), Token Ring, Frame Relay connections, etc. [15]. The result of the inspection is shown in figure 4.

![Figure 4](image)

**Figure 4.** The results from the Wireshark packets capture.

The verification confirms that the data packets are sent from the sensor network to ThingSpeak, using MQTT protocol for data transmission, and connected to the communication port 8883 which is protected/encrypted by TLS / SSL protocol.
4. Conclusion

The integration between the clouds and the sensor networks makes it possible to solve the problem of storing and processing large volumes of data without unnecessarily increasing the cost of sensor networks. An important integration challenge is the security of the connection when transmitting data between sensors and the cloud. The researched literature sources show that the research problem is topical and the subject of research by both the academic and the industrial community. Different solutions are depending on the specific challenge to be solved by designing network controllers, establishing standards for security, optimisation number of sensors in the network, software countermeasures and implementation of specialised hardware security devices. In this paper, we propose an algorithm that ensures security in the transmission of data from the sensor network to the cloud via the DLS protocol. The algorithm is implemented through the ThingSpeak cloud, the connection between the sensor network and the cloud is through the MQTT protocol, creating the flow that will gather sensor data, using Node-RED and Generate a self-signed certificate with OpenSSL. The algorithm was tested with Wireshark software, which confirmed that the data packets sent from the sensor network to ThingSpeak use the MQTT data transfer protocol and connect to communication port 8883, which is protected/encrypted via TLS / SSL protocol.

References

[1] Tsvetanov F and Pandurski M 2018 Some aspects for the integration of sensor networks in cloud structures Proceedings of International Conference on High Technology for Sustainable Development pp 249-253
[2] Hristov G Tsvetanov F and Georgieva I 2016 Cloudy Technologies in Industry, Knowledge – International Journal Vol 15.2 pp 891-897
[3] Mell P and Grance T 2011 NIST Definition of Cloud Computing, Recommendations of the National Institute of Standards and Technology
[4] Chadwick D Fan W Costantino G Lemos R Cerbo F Herwono I Manea M Mori P Sajjad A and Wang X 2020 A cloud-edge based data security architecture for sharing and analysing cyber threat information, Future Generation Computer Systems Volume 102 pp 710-722
[5] Mohammad S and Ebrahim AI 2019 A secure and energy-efficient platform for the integration of Wireless Sensor Networks and Mobile Cloud Computing Computer Network Volume 16
[6] Coppolino L, D’Antonio S Mazzeo G and Romano L 2019 A comprehensive survey of hardware-assisted security: From the edge to the cloud Internet of Things Volume 6 pp 1-17
[7] Intel Threat Detection Technology, 2018, Accessed June 23, 2020, https://www.intel.com
[8] Microchip, AN2559, Smart IoT Wireless Sensor Node using XMEGA® AU MCUand LoRa Technology, http://ww1.microchip.com/downloads/en/AppNotes/00002559A.pdf
[9] https://thingspeak.com
[10] Ray P 2017 A survey of IoT cloud platforms Future Computing and Informatics Journal Volume 1 Issues 1–2 pp 35-46 https://doi.org/10.1016/j.fcij.2017.02.001
[11] https://uk.mathworks.com/help/thingspeak/mqtt-basics.html
[12] https://www.rcdevs.com/products/openotp-hsm-hardware-security-modules/
[13] Lake J 2019 What is TLS, and how does it work? Available at: https://www.comparitech.com/blog/information-security/tls-encryption/, Accessed 23 June 2020
[14] https://nodered.org/
[15] Wireshark – Go Deep, Available at: https://www.wireshark.org