Multimedia Tools and Applications
https://doi.org/10.1007/s11042-021-11389-8

1211: AIOT SUPPORT AND APPLICATIONS WITH MULTIMEDIA

An approach to adding simple interface as security gateway architecture for IoT device

Nikola Pavlović[1] · Marko Šarac[1] · Saša Adamović[1] · Muzafer Saračević[2] · Khaleel Ahmad[3] · Nemanja Maček[4] · Deepak Kumar Sharma[5]

Received: 10 October 2020 / Revised: 7 July 2021 / Accepted: 30 July 2021

Abstract
It is not a secret that Internet of Things (IoT) devices often come with not so realistic processing power (i.e. processing power and storage requirements) that would provide a basis for strong security and encryption algorithms. This work proposes an approach to adding a simple interface as a security gateway architecture for IoT devices. The security interface provides mapping for the IoT device remote services as long as support for stronger cryptographic algorithms. The solution improves the security of the data that IoT devices send to remote services by performing compatible cryptographic algorithms on the data before it sends to remote services. The result of this work is the development of a security interface that provides support for any cryptographic algorithm, uses Internet Protocol (IP) mapping to prevent access to the devices behind the interface from non-authorized IP addresses. As such it provides robust protection against attacks and data manipulation. The work is tested for memory usage and the strength of the security it provides.

Keywords IoT security · Gateway security · IoT communication systems · IoT network performance

1 Introduction

A complex network of smart devices is composed of the IoT. This device needs a constant connection to the Internet to exchange data which makes the 5G network an ideal choice from the perspective of low latency and high data peak rates [18]. IoT connects to many machines and devices using wired and wireless networks, and current 5G networks can initialize 106 devices per square kilometers with up to 10 megabytes per second per square kilometer and 1ms round-trip latency. These characteristics of current 5G networks make them an ideal choice for IoT devices in terms of connectivity, but threats must be considered and addressed properly. Cyber-attacks are not new to IoT since it is deeply involved in activities and societies, thus it is becoming necessary to take cyber defense seriously.

✉ Khaleel Ahmad
khaleelahmad@manuu.edu.in

Extended author information available on the last page of the article

Published online: 12 August 2021

Applying security in IoT has been recognized as one of the biggest obstacles. This paper presents an analysis of the status and concerns regarding IoT security. Kevin Ashton introduced the Internet of Things in 1998. He stated that the IoT is the network of computers that know everything about humans and that they specialize to use the data that they collect without any assistance from a human. Devices are further linked. Hence, IoT may refer to the interconnectivity among often-used electronic devices [26]. IoT's ability to offer different services has made it the fastest-growing technology. It has made a huge impact on the environment and social life. Further development of network infrastructure with the respect of 5G and 6G networks is speeding up the usage of IoT devices even more, and therefore security issues must be addressed and analyzed carefully.

The goal of IoT is to change the way we live today through the use of smart devices around us to perform everyday tasks and tasks with minimal human interference. The words that are used with the IoT are smart cities, smart houses, smart transport and infrastructure, etc. The main contribution to this paper is:

- Develop a security interface for IoT devices
- Add IP mapping for all devices on the security interface
- Add rules to firewall to prevent access of third parties to interface
- Develop solution in Node.js and test memory usage for Advanced Encryption Standard (AES), Data Encryption Standard (DES), Triple DES
- Experiment on Arduino compatibility with a security interface

A few working groups and industry leaders have suggested IoT system standardization, but they have sadly not decided on a solution. In response to the growing demand for connected devices and services worldwide, IoT has created excessive demand for security. For IoT to do its' potential, it needs protection against vulnerabilities and potential attackers. A variety of attacks as well as several threats are rising daily both in number and complexity – either to execute, as an adversary, or to get disposed of, as a user. The IoT should guarantee the suitability and trustworthiness of the processed data to give users useful output. To ensure robustness and reliability at the service level and to be able to support security, this is a significant need for such systems. Security and privacy problems are a growing concern for customers in their shift towards the IoT. Implementing IoT in-home and workspace introduces new security issues. Customers and suppliers should consider that challenge and must be cautious with such security and privacy concerns.

The paper is organized into 6 sections. Related works relating to security for IoT devices are provided in the second section. In the third section, a proposed solution to adding a simple interface as security gateway architecture for IoT devices is developed and presented. The fourth section, presented an experiment to check for memory usage and usability. In the fifth section, the analysis is presented with what is accomplished, positive and negative sides to the security interface, and possible improvements. Concluding remarks and future works are given in the last section.

2 State of the art – related work

Nawir et al. [17] described what are the key elements that involve IoT systems, their relationships, and the rise of security issues in various environments where they are integrated. These devices are mostly used in homes, healthcare, and transportation. To support billions of IoT gadgets across the world, wireless community infrastructure is needed to be not the handiest, exceptionally scalable in terms of its capacity, however, also optimally managed with unique provider wishes of diverse IoT verticals [23]. Mobile Internet and the IoT are the two foremost drivers of destiny cellular networks and will span a vast prospect for 5G. The 5G generation is being defined because of the first community designed to be scalable, versatile, and strength-clever for the hyper-connected IoT world [7]. According to [12], 5G will deal with many elements of lifestyles in the future, which include home, work, and transportation, and can be characterized by high visitors extent density, high connection density, or excessive mobility, composing the fundamental features of IoT ecosystems [15].

Four important components of IoT are the person, smart object, technological ecosystem, and operation. In health care, IoT is used to monitor patient health. Some examples of these devices are glucose monitors, connected inhalers, ingestible sensors, asthma monitors, and others. All these devices are tracking and collecting data and sending it to the cloud to do the processing. Because progress reports of patients' medical conditions are confidential it is required that this data be encrypted. When it comes to home devices this device needs to be secure and autoimmune to any unauthorized access. The most frequent attacks and techniques are displayed in Table 1.

Attacks on IoT devices can be done on different network levels. In the Table 2 attacks are sorted and explained based on the network level they are executed at.

In this paper, the authors attempted to describe different attacks within IoT to help researchers and developers to include appropriate security measures in their development.

Mahmoud et al. [13, 14] seek to contribute to a better understanding of threats. The authors explained why IoT devices are highly beneficial to attackers. Most IoT devices work without human interaction so, it is easy for the attackers to gain physical access to them. These devices also operate using wireless networks which makes it easier for an attacker to obtain confidential information by conducting a man-in-the-middle attack. Most of the devices cannot support complex security algorithms due to their hardware limitation. In this paper, the authors provided an overview of the most important Internet of Things security problems with a focus on challenges surrounding the devices and services. The authors concluded that much work needs to be done by both end-users and vendors. It is important to define a standard that will address the shortcomings of the current IoT security mechanisms.

Table 1 Most frequent attacks on the internet and techniques

Attack	Target	Weakness	Technique
Denial of Service (DoS) Attack [3]	All IoT devices are connected to the internet.	Making IoT devices inaccessible to their intended users.	Flooding the IoT device with a request that is never completed.
Wormholes [9]	Location of the packets.	Problematic in checking the routing information.	Retransmitting of recorded packets inside IoT network.
Spoofed, alter, or replay network request [20]	Routing information. Detecting IoT devices.	High network latency	The attacker listens to network activity. Operates only when the device stops sending the signal, sending the unreliable signal.
Sybil [5]	The integrity of data security and resource utilization.	Launch a threat to the geographic routing protocol. Costly network.	The attacker operates like a normal user.

Table 2 Attacks on IoT devices per network layer

Layer	Attacks	Methods/Strategies attacks
Physical	Jamming	Creates radio interference and exhaustion on IoT devices.
	Tampering	Creates compromised nodes.
Data Link	Collision	Simultaneously transmit two nodes on the same frequency.
	Exhaustion	By repetitive collision, the nodes.
	Unfairness	Using the above link-layer attacks.
Network	Spoofed altered or replayed routing information	The compromised network will have request loops that will repel or attack network requests from other nodes.
	Selective forwarding	Selectively gather information and transmit it.
	Sinkhole	Inside attack that forces all devices to reroute their connection to the compromised node.
	Sybil [20]	A single node claims multiple identities.
	Wormholes [9]	Retransmitting of recorded packets inside IoT network.
	Acknowledgment spoofing	Spoof the link-layer acknowledgments for overhead packets.
Transport	Flooding	Generate new requests until the IoT device uses all its resources.
	De-synchronization	Interruption of an existing connection.
Application	Attacks on reliability (clock altering, selective data forwarding, and data exaggeration)	The attackers operate like normal user in the IoT system. Attackers can execute malicious activities in that IoT system.

Usha and Bobby [24] provided a comprehensive review of attacks on each layer of the network. New network protocols such as IPv6 and 5G should be implemented to boost security devices to accomplish a dynamic mashup of IoT topology. Most of the attacks happen on the perception layer, network layer, and application layer. Most of the signals transferred between IoT devices can be compromised by disturbing the waves. The relay attack will exploit the confidentiality of this layer. Relay attacks can be made by altering, replaying, or spoofing the identity information provided by the device. Another attack that can be done on this layer is the timing attack. A timing attack is performed by analyzing the required time to perform the encryption. The result of this attack is that attackers gain access to encryption keys. Attackers can gain physical access to the node and capture all information and data. This is called a node capture attack. On the network layer, the most popular attacks are the Denial of Service attack (DoS) and man in the middle attack. Due to IoT lacking standards or security policies, a high number of devices are sustainable to attacks on the Application layer. Different software and applications have different or no security algorithms. The big problem here is that different Internet of Things devices must be compatible with one another. To ensure this vendor provides a security mechanism that will ensure this device can "talk" to each other, which makes them vulnerable to already discovered attacks. The authors concluded that devices should use newer network standards. They proposed the realization of a smart framework for this device with end-to-end point security. There is required for new identification, software, wireless, and hardware technologies to overcome IoT challenges. To provide users with valuable output, IoT devices need to guarantee that processed data is secure. Most smart home systems run on the AllJoyn framework using an asymmetric Elliptic Curve Cryptography to perform the authentications during system operation. This device uses a Wi-Fi router to be the center node of the system. This means that the entire security of the smart home system is based on the security of the Wi-Fi router. Most popular smart home devices such as Sense sleep monitor, Nest Cam Indoor security camera, WeMo switch, and Amazon Echo reveal sensitive user interaction in their network packets. This means that there is a need for better encryption to protect user privacy. The author also proposed a protocol in which people can scarcely control the disclosure of their personal information. The authors concluded that by the privacy principle the users should be able to keep control of their data and opt-out of the information collecting without negative consequences. Security and privacy at a device level are critical to improving the overall security of the Internet of Things devices.

Alrawi et al. [1], presented a security evaluation of home base internet of things devices. They considered the exploitation of the hub device to be equivalent to exploiting all the connected devices. Hubs bridge low-energy devices to IP networks, and they have a pre-established trust relationship. Many hubs are susceptible to man-in-the-middle attacks because they support older versions of Transport Layer Security (TLS) or Secure Sockets Layer (SSL) protocols to maintain compatibility with all devices. Many hubs run an internal webserver with TLS on port 443 and SSL on port 22. The certificate used for TLS is self-signed. These hubs use weak ciphers like Rivest Cipher 4 (RC4) and insecure protocols such as SSLv3. The authors run audits on Insteon hub, Wink 2, Sonos Speakers, Nvidia Shield, Google Home, Samsung SmartTV, and Samsung SmartThings and all these devices had the same problem with expired self-signed certificate and usage of a weak cipher. Devices that run Universal Plug and Play (UPnP) have no authentication at all and allow anyone on the local area network to control the device. The example of these devices is MiCasaVerda VeraLite, Wink 2, Sonos, Bose SoundTouch 10, Samsung SmartTV, Logitech Harmony, and Roku. The authors proposed the following. For vendors to implement better security solutions. End users should change the default configuration, enable encryption, and disable remote access to hubs. Internet service providers can see that communication comes from a specific device and port. They could implement technical policies to block certain ports.

Noura et al. [19] proposed one round cipher algorithm for multimedia IoT devices. The solution authors proposed guards against passive attacks (modifying or deleting content packets). The authors of this paper addressed the problem of securing the distributed multimedia systems in IoT (MIoT), which consists of cameras and microphones. The algorithm authors proposed to use only one dynamic key-dependent round function. In addition to simple operations, the cipher is based on a dynamic key approach and dynamic sub-matrices sections. Several experiments are conducted, making this algorithm more robust compared to existing ones.

The motivation for this work comes from the observed problem that all IoT devices have. IoT hubs provided by the manufacturer, if exist, provide low or no all security features. Mainly these hubs are to integrate different IoT devices from the same brand. Other IoT hubs are mainly used to observe IoT devices in the smart home (smart cameras, intercoms, and others) and display the data they provide on PC. The solution is a simple interface that will work for any IoT device and network infrastructure. It connects to the home router, uses the firewall provided by the router as an additional layer to prevent any other attackers from accessing smart devices. The main feature of work is that it provides support for any cryptographic algorithm that remote servers used to provide data to IoT devices.

This means that devices such as Arduino or Raspberry Pi can be used to run the server and add a layer of security to all smart devices on the home network.

3 Proposed solution

A developed solution will be presented from theory to practical implementation and security evaluation. This solution is based on the following environment:

- Custom home server (hub) for all smart devices connected to it. Using wired connections between the smart device and home servers. Adding a layer of security on the home server – firewall, serializing data, compressing data, and encryption.
- Using a programming language so the server would be able to run on any device.
- Preventing smart devices from directly communicating to the Internet and the Internet to smart devices. All communication needs to be done through the home server.
- Creating custom port and IP address mapping so the connection to smart devices would be filtered. This can be done using a Router Firewall but with a solution, it would be easier to monitor which device is calling what service and getting a response (see Fig. 1).

The security of the smart device is the equivalent of the security of wireless networks [11]. The exploitation of the hub device is equivalent to exploiting all the connected devices. Hubs bridge smart devices to IP networks, and they have a pre-established trust relationship. Hub security is equivalent to the security of the wireless network they are connected to. Some smart devices have no support for hubs, so their security is based on the security of the network. To improve the security of these devices we proposed a solution based on the custom home server (hub) for all smart devices regardless of their support for hubs. This solution consists of a home server, a wired or wireless connection to a smart device, and a home router. All data sent from smart devices to their remote services will be intercepted and parsed by the server. This means unnecessary leaked information can be removed about the device and properly encrypt the package before sending it to the

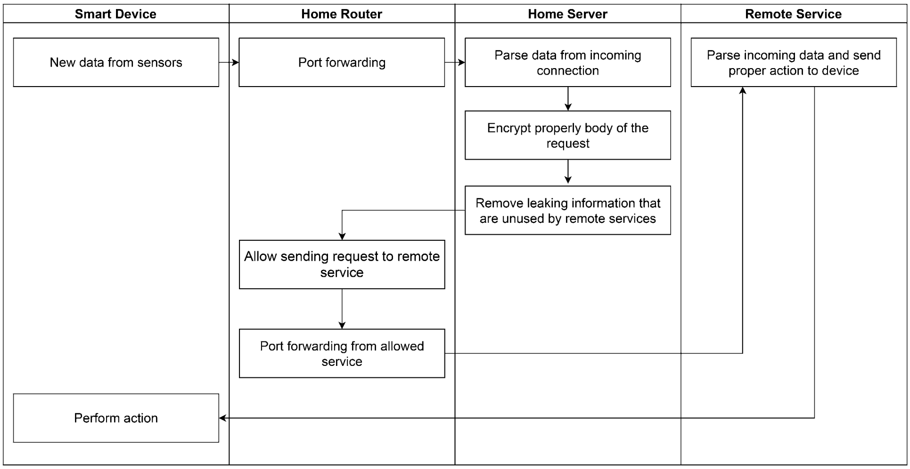

Smart Device
Home Router
Home Server
Remote Service
New data from sensors
Port forwarding
Parse data from incoming connection
Parse incoming data and send proper action to device
Encrypt properly body of the request
Remove leaking information that are unused by remote services
Allow sending request to remote service
Port forwarding from allowed service
Perform action

Fig. 1 UML Activity Diagram for the proposed solution

service. The data from the smart device needs to be parsed and prepared for the remote service in the correct format. To intercept this data, we are using home router port forwarding. Using a sniffing application such as Wireshark [10] we can detect which device is sending a request to which service, their IP address, and port. A smart device then can be mapped to send this request to the home server. If the device supports a wired connection, there is no need for port forwarding, and the solution can simply connect this device to the home server and intercept the data.

The home server does the following: Get data from the smart device; Parse data; Make sure this data is sent properly encrypted to remote service (using HTTP(s) protocol). Additional features we want on the home server:

- Map each device's IP address and port to their service without using port forwarding logic.
- Stop any connection to a smart device that is not from the allowed IP address, firewall for smart devices.
- Smart device data monitor for any suspicious requests.
- If possible, using a strong encryption algorithm [8] (only possible if remote service is supporting different encryption standards). This should be possible to activate per smart device in-home network.

The home server should be written in a programming language that can be run on most devices. In solution, Node.js [22] is used as a programming language. Node.js has good support for most devices. Some process managers for Node.js such as PM2 have good support for the container approach. This means the home server would be in a container and any attacker would have a hard time connecting to it. PM2 also supports cluster mode. Since *Node.js* is a single-threaded language cluster mode allows applications to utilize all cores of the CPU allowing applications to be scalable. This greatly increases the performance of the server depending on the number of core CPUs. Each process is created on a new cluster. If an attacker tries to exploit any process on the server, the cluster will destroy the process after a certain amount of time to make sure the server works as intended. Using the home server, the work will prevent the following attacks:

- Man in the middle [21], there will be no way for attackers to directly sniff data from smart devices. The only data they will be able to get is one from the router to the internet. If properly encrypted there will be a very low chance of doing any exploitation.
- Directly connecting to a smart device and doing any exploit on it. All connections from remote to a smart device are forwarded to the home server and then checked if the request is coming from approved sources.
- Devices in local area networks that have no authentication will have a new layer of security based on authentication on the home server. To gain access to any smart house device authentication and authorization on the home server would be needed.

VegeHub is a solution that is used for garage door sensors, mailbox, livestock, home automatization, wireless sensor data gathering, and environmental monitoring. The hub can take data from a maximum of 4 sensors. When it comes to security, VegeHub operates on TLS 1.2 encryption standard. It comes with the most used website certificates. The connection between the smart device and VegeHub is wired but between VegeHub and the home, the router is wireless.

4 Security, trust, and solution limitations

There are 4 points of interest in the proposed solution as shown in Fig. 1. The first point of interest is the Smart device. Smart devices use sensors to collect data and to process that data it needs to send to a remote server. Since this device has low processing power, data collected from external sources is poorly encrypted or not encrypted at all. To secure this data on the internet, a solution needs to intercept it. Intercepting data is done from the second point of interest. The second point of interest is the router. Every household has one or many routers. To forward data to the server router will redirect data from selected devices to servers. To do this port forwarding functionality is used that exists on routers.

To process this data from the router's server is capturing it locally. The data that comes from the device has the following request header and request body. The request header has information like request URL, the request method, status code, version of HyperText Transfer Protocol request (HTTP/1.1 or HTTP/2), encoding information, user agent information, authorization information, and content type information. Some IoT Hubs [6] to work correctly send additional information regarding the hub in the header (*Iothub-message-Id*, *IoThub-correlation-Id*, *Iothub-user-Id).*

Most of the data in request headers are not used by remote service so it can be omitted. The request body has data that is required for remote service to parse. To prevent data leaking, the server is omitting unused data from each request to remote service. All requests from servers made to the internet are using the HTTP/2 version of the protocol. To further improve the security of each request it is possible to add a layer of encryption for the request body. This means if a remote service has the functionality to use different encryption algorithms, the server can add it here. For example, the server can generate a Rivest–Shamir–Adleman (RSA) key pair [25] and add a public key on remote service or generate any symmetric key to use with AES, DES, or Triple DES [4]. New prepared requests can now be processed and sent to remote service.

The request is sent from the server to the home router which then forwards it to the internet. The only entry point for any IoT device in the home network is through router requests made to the home server. The same goes for another way around. Remote service parses data server send and sends back appropriate action for the device to do. Again, the home router is forwarding this request to the server. Any request for an IoT device is forwarded to the server. The validity of the request is checked on the server. The server is then answering the following questions:

- Do home servers expect remote service to send a request to an IoT device?
- Does the request body and header contain any suspicious data?

If the server concludes that the request is valid it will be sent to an IoT device in the local area network. After this IoT device will do the action that is requested by the remote service. The proposed solution is compatible with IoT devices that use wireless, cable, or serial connections to send data to remote services.

Memory usage is evaluated using AES, DES, and Triple DES, and here are the results. The measurements are displayed in megabytes. Heap is a memory segment dedicated to storing reference types like objects, strings, and closures. The Heap total represents the total size of the heap used by the server. The heap used is the size of the heap used when the logic of the program is executing. External refers to the memory usage of C++ objects bound to JavaScript objects managed by V8. V8 is Google's open-source high-performance JavaScript and Web Assembly engine, written in C++. It is used in Chrome and Node.js.

This memory is used on IoT Hub, the hub should have at least 256 megabytes of memory for it to work properly and this is shown in the Tables 3, 4 and 5 where we have tested different encryption standards with a proposed solution.

From the memory measurements, it is noticeable that the home server does not require more than 200 megabytes of memory to perform encryption/decryption on smart device data. The lowest memory usage is in the AES algorithm. DES and Triple DES have stable memory usage but are higher than AES.

The limitation of using Node.js is that it is single-threaded. This is due to compatibility issues with different platforms and integrations. To make it multi-threaded it needs to run under a process manager such as Process Manager 2 (PM2). The solution requires a database to save keys for different devices for encryption/decryption purposes. This means if anything happens to the database it will make the solution unusable. This can be solved by using a memory database. The memory database is in the IoT hub. This database does not have any specific protection in terms of cryptographic solutions but only from the access point, that no remote request cannot access it. To get access to the memory database one must be directly connected to the IoT hub.

When the proposed solution is expecting a response from a remote service and it does not respond it halts the process on the server. The current solution uses timeouts to wait for remote service to respond, if it does not respond until then, the process is killed, and no data is sent back to the smart device. The most important limitation of all is that most of the code running on smart devices is not open-sourced so getting data from Smart Device, parsing, and sending it encrypted to remote service is not possible without direct contact

Table 3 AES memory usage from 20 processes in megabytes

	heapTotal	heapUsed	external
1	160,66	142,12	14,21
2	160,66	142,13	14,21
3	160,66	142,18	14,21
4	160,66	142,19	14,21
5	160,66	142,22	14,21
6	160,66	142,23	14,21
7	160,66	142,25	14,21
8	160,66	142,26	14,21
9	91,61	85,59	9
10	91,61	85,61	9
11	91,61	85,64	9
12	91,61	85,65	9
13	91,61	85,67	9
14	91,61	85,68	9
15	91,61	86,84	11,64
16	91,61	86,85	11,64
17	91,61	86,86	11,64
18	91,61	86,87	11,64
19	91,61	86,6	9,37
20	91,61	86,6	9,39
Ave	47,692	43,4408	4,66

Table 4 DES memory usage from 20 processes in megabytes

	heapTotal	heapUsed	external
1	123,16	103,05	10,99
2	120,61	85,54	12,35
3	120,61	85,55	12,35
4	120,61	85,57	12,35
5	120,61	85,59	12,35
6	120,61	85,59	12,35
7	120,61	85,63	12,35
8	120,61	85,64	12,35
9	120,61	85,65	12,35
10	120,61	85,8	12,36
11	120,61	85,82	12,36
12	120,61	85,83	12,36
13	120,61	85,84	12,36
14	120,61	85,86	12,36
15	120,61	85,87	12,36
16	120,61	85,88	12,36
17	120,61	85,9	12,36
18	120,61	85,91	12,36
19	120,61	85,9	12,36
20	120,61	85,92	12,36
Ave	48,295	34,6468	4,915

with the manufacturer. Some manufacturers however provide documentation for developers and dashboards where data can be changed, improved, and provided to remote services in a different format.

5 Security evaluation

In Fig. 2. Is shown a class diagram for the proposed solution. IoT devices send data to an observer. This data can be anything. In this work, Arduino [2] is used with sensors for temperature and humidity. Temperature and humidity are measured, and this data is sent to the internet. Before sending it directly to the internet, the router is forwarding this data to the Observer (home server).

The observer is parsing the data, performing encryption, and sends it to a remote service. Remote service is parsing data, decryption in that process, and based on data it sends a notification to IoT devices to perform certain actions. The home router is once again forwarding this request to the observer which decrypts the data and sends it to the IoT device to do the requested action.

The proposed solution network structure is shown in Fig. 3. It consists of IoT devices, IoT hub, router, firewall, and remote service server. IoT device sends requests to IoT hub, the hub after performing its logic sends a request to router, passes the firewall and waits for the response from the remote server. After the response is sent, it passes the

Table 5 Triple-DES memory usage from 20 processes in megabytes

	heapTotal	heapUsed	external
1	124,41	104,72	14,05
2	124,41	104,73	14,05
3	123,11	90,71	9,75
4	123,11	90,72	9,75
5	123,11	90,73	9,75
6	123,11	90,76	9,75
7	123,11	90,77	9,75
8	123,11	90,78	9,75
9	123,11	90,8	9,75
10	123,11	91,22	9,75
11	123,11	91,23	9,75
12	123,11	91,25	9,75
13	123,11	91,26	9,76
14	123,11	91,27	9,76
15	123,11	91,29	9,76
16	123,11	91,3	9,76
17	123,11	91,31	9,76
18	123,11	91,33	9,76
19	123,11	91,34	9,76
20	123,11	91,35	9,76
Ave	49,296	36,9774	4,0736

firewall, gets the port forwarded to the IoT hub to check the request, and then forwarded to the IoT device.

The sensor is connected to Arduino. Here two different types of connections to the server. The first was using the internet and the second was using the serial port. When the serial port is used, there is no need to use the home router. The home server is connected to the internet by a home router. Here class has a serial connection or port observer for certain requests. Function observer takes params for serial connection and server if data is coming from the internet. Before emitting the data to a remote server, the home server can add here any cryptography algorithm to encrypt or decrypt the data. After encryption or decryption, the server can emit (broadcast) requests to the desired remote service. The same observer class works the other way around when requests come from remote services to IoT devices.

On Fig. 4. Wireshark is used to capture network requests that come from an IoT device to the internet. As shown in the image, the network request body is in plain text. Using the proposed solution, the security of the network request body can be much improved.

Since the solution supports serial port connection to IoT devices, USBCap [16] (Fig. 5.) software is used to capture requests coming from the device on the serial port. All requests that leave the device are properly encrypted and do not have any additional data that could be used to collect information about the device behind the IoT hub. This means that there is no need to worry about the security of data from device to home server.

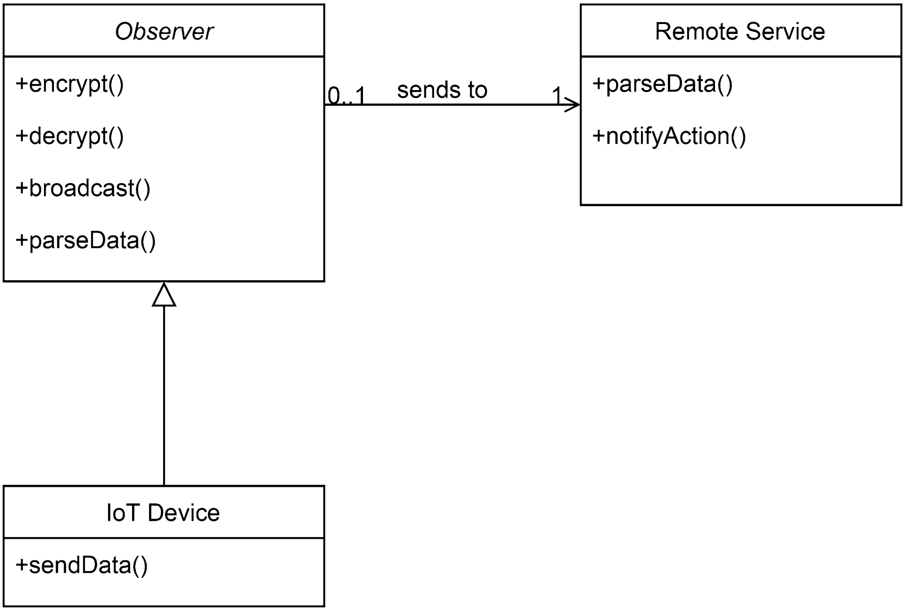


Fig. 2 Class diagram for the proposed solution

Previous research provides analysis of the specific attacks done on the IoT network or the solutions that are still not secure and sustainable to specific attacks, such as open specific ports or using round ciphers made just for IoT devices to save energy. Our solution has no open ports, is directly connected to the router, and has its validation and authentication as well as encryption of the data.

This prevents attackers from changing the packets or directly sending requests to IoT devices. No requests can be sent to an IoT device directly and it will be stopped by the IoT hub. The results of the solutions are:

- All data sent or received by the IoT device or remote server is encrypted.
- A firewall is used to prevent access to IoT devices.

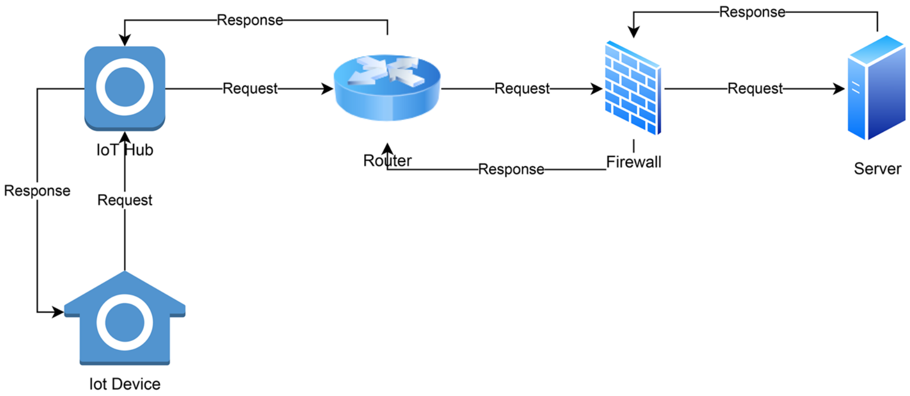
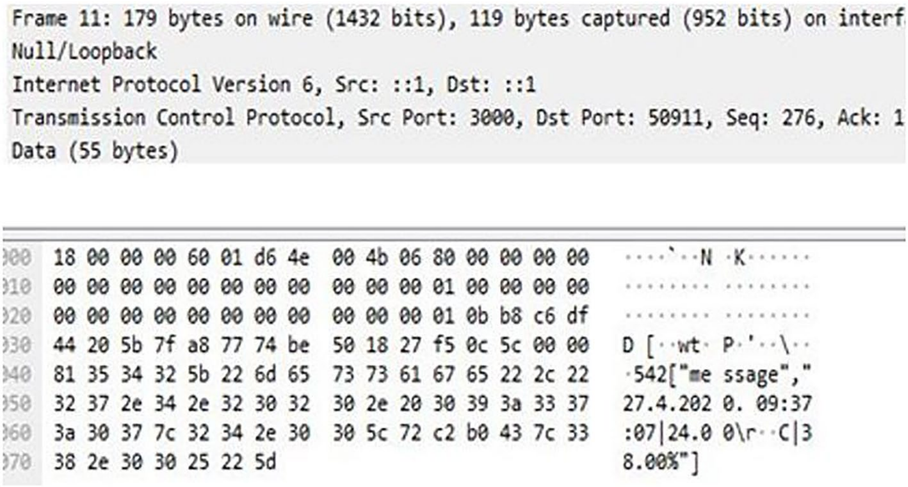


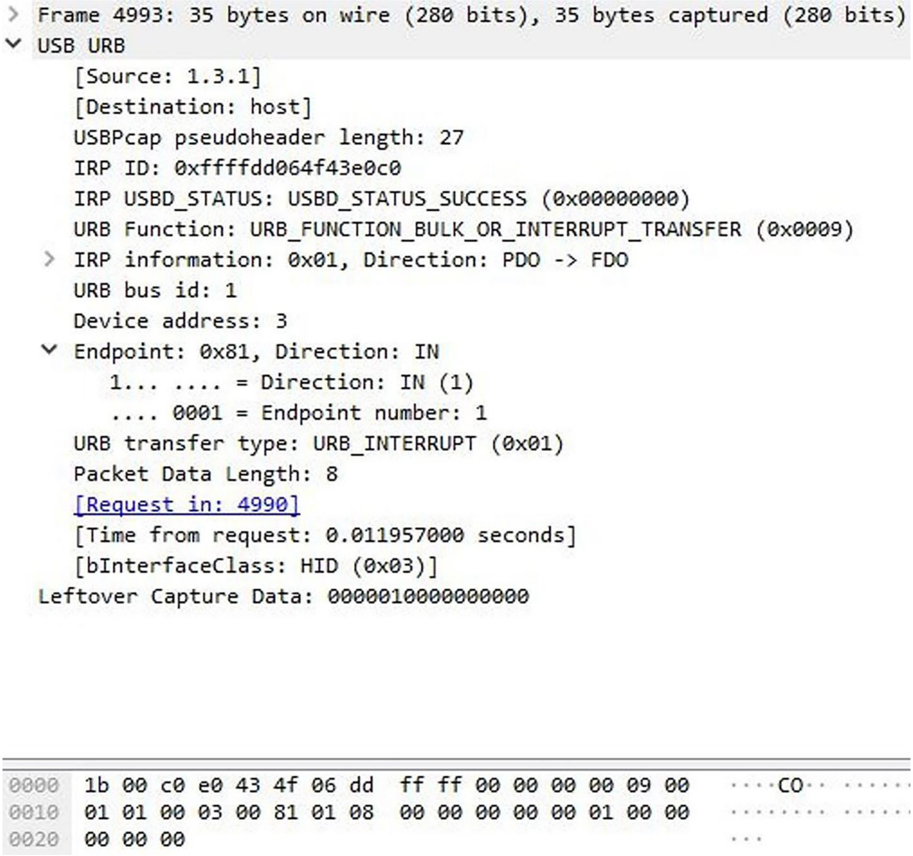
Fig. 3 The network structure for the proposed solution



```
Frame 11: 179 bytes on wire (1432 bits), 119 bytes captured (952 bits) on interf
Null/Loopback
Internet Protocol Version 6, Src: ::1, Dst: ::1
Transmission Control Protocol, Src Port: 3000, Dst Port: 50911, Seq: 276, Ack: 1
Data (55 bytes)
```

```
000  18 00 00 00 60 01 d6 4e  00 4b 06 80 00 00 00 00   ····`··N ·K······
010  00 00 00 00 00 00 00 00  00 00 00 01 00 00 00 00   ········ ········
020  00 00 00 00 00 00 00 00  00 00 00 01 0b b8 c6 df   ········ ········
030  44 20 5b 7f a8 77 74 be  50 18 27 f5 0c 5c 00 00   D [··wt· P·'··\··
040  81 35 34 32 5b 22 6d 65  73 73 61 67 65 22 2c 22   ·542["me ssage","
050  32 37 2e 34 2e 32 30 32  30 2e 20 30 39 3a 33 37   27.4.202 0. 09:37
060  3a 30 37 7c 32 34 2e 30  30 5c 72 c2 b0 43 7c 33   :07|24.0 0\r··C|3
070  38 2e 30 30 25 22 5d                                8.00%"]
```

Fig. 4 Internet Capture using Wireshark



```
> Frame 4993: 35 bytes on wire (280 bits), 35 bytes captured (280 bits)
v USB URB
     [Source: 1.3.1]
     [Destination: host]
     USBPcap pseudoheader length: 27
     IRP ID: 0xffffdd064f43e0c0
     IRP USBD_STATUS: USBD_STATUS_SUCCESS (0x00000000)
     URB Function: URB_FUNCTION_BULK_OR_INTERRUPT_TRANSFER (0x0009)
   > IRP information: 0x01, Direction: PDO -> FDO
     URB bus id: 1
     Device address: 3
   v Endpoint: 0x81, Direction: IN
        1... .... = Direction: IN (1)
        .... 0001 = Endpoint number: 1
     URB transfer type: URB_INTERRUPT (0x01)
     Packet Data Length: 8
     [Request in: 4990]
     [Time from request: 0.011957000 seconds]
     [bInterfaceClass: HID (0x03)]
  Leftover Capture Data: 0000010000000000
```

```
0000  1b 00 c0 e0 43 4f 06 dd  ff ff 00 00 00 00 09 00   ····CO·· ······
0010  01 01 00 03 00 81 01 08  00 00 00 00 00 01 00 00   ········ ······
0020  00 00 00                                           ···
```

Fig. 5 Package capture using USBCap

- No IoT device is directly connected to the internet but rather to an IoT hub.
- Stronger cryptographic solutions can be used because of the more processing power and memory that has the IoT hub.

6 Conclusion and future works

This paper provides the flexibility of using any IoT device without any concerns about the security it provides. The solution provides a simple interface that adds the best security incompatibility with remote services that IoT devices connect to. This means that solution provides security that is compatible with remote service. Cryptography algorithms are moved from IoT devices to the host server. The server provides confidentiality, integrity, and availability in a way met in productional networks. This is done by using existing cryptographic algorithms. To improve security for each IoT device in a local area network, the IoT device manufacturer needs to provide a flexible interface for the device to connect to. This means that IoT device data sent to the server can be encrypted using one of many modern encryption algorithms. An IoT device manufacturer needs to provide a list of IP addresses that an IoT device connects to. This solution can prevent any other IP address from trying to connect to the IoT device in a local area network.

As future work, the server will be optimized to utilize a wider range of IoT devices and be compatible with different encryption algorithms. Additionally, the proposed solution should be adapted to work without the need for home routers and IP mapping. The entire firewall solution should be implemented inside the server and observe requests to IoT devices in the local area network and add additional layers of authentication and validation. This way the server setup would be easier to set up in any environment without additional network knowledge.

Author contributions Methodology (N.Pavlovic, M.Sarac, S.Adamovic); Resources, Investigation (M.Saracevic, N.Macek); Conceptualization (K.Ahmad, D.K. Sharma).

Funding This research received no external funding.

Data availability and material (data transparency) Not applicable.

Declarations

Human rights This article does not contain any studies with human participants or animals performed by any of the authors.

Conflicts of interest The authors declare that they have no conflict of interest.

References

1. Alrawi O, Lever C, Antonakakis M, Monrose F (2019) SoK: Security Evaluation of Home-Based IoT Deployments. 2019 IEEE Symp Secur Priv (SP). https://doi.org/10.1109/sp.2019.00013
2. Andriansyah M, Subali M, Purwanto I, Irianto S, Pramono R (2017) e-KTP as the basis of home security system using arduino UNO. 2017 4th Int Conf Comput Appl Inf Process Technol (CAIPT). https://doi.org/10.1109/caipt.2017.8320693

3. Bechtel M, Yun H (2019) Denial-of-Service Attacks on Shared Cache in Multicore: Analysis and Prevention. 2019 IEEE Real-Time Embed Technol Appl Symp (RTAS). https://doi.org/10.1109/rtas.2019.00037
4. Bhat B, Ali A, Gupta A (2015) DES and AES performance evaluation. IEEE Int Conf Comput Commun Autom 15–16 May 2015 Greater Noida, India. https://doi.org/10.1109/ccaa.2015.7148500
5. Buford J, Yu H, Lua EK (2009) P2P Networking and Applications (Morgan Kaufmann Series in Networking (Hardcover)) (1st ed.). Morgan Kaufmann
6. Cirani S, Ferrari G, Iotti N, Picone M (2015) The IoT hub: a fog node for seamless management of heterogeneous connected smart objects. 2015 12th Ann IEEE Int Conf Sens Commun Netw Workshops (SECON Workshops). https://doi.org/10.1109/seconw.2015.7328145
7. Chávez-Santiago R, Szydełko M, Kliks A, Foukalas F, Haddad Y, Nolan KE, Kelly MY, Masonta MT, Balasingham I (2015) 5G: The Convergence of Wireless Communications. Wirel Pers Commun 83(3):1617–1642. https://doi.org/10.1007/s11277-015-2467-2
8. Faheem M, Jamel S, Hassan AAZ, Shafinaz N, Mat M (2017) A Survey on the Cryptographic Encryption Algorithms. Int J Adv Comput Sci Appl 8(11). https://doi.org/10.14569/ijacsa.2017.081141
9. Goyal M, Dutta M (2018) Intrusion Detection of Wormhole Attack in IoT: A Review. 2018 Int Conf Circuits Syst Digit Enterp Technol (ICCSDET). https://doi.org/10.1109/iccsdet.2018.8821160
10. Iqbal H, Naaz S (2019) Wireshark as a Tool for Detection of Various LAN Attacks. Int J Comput Sci Eng 7(5):833–837. https://doi.org/10.26438/ijcse/v7i5.833837
11. Kavianpour A, Anderson M (2017) An Overview of Wireless Network Security. 2017 IEEE 4th Int Conf Cyber Secur Cloud Comput (Cscloud). https://doi.org/10.1109/cscloud.2017.45
12. Liu G, Jiang D (2016) 5G: Vision and Requirements for Mobile Communication System towards Year 2020. Chin J Eng 2016:1–8. https://doi.org/10.1155/2016/5974586
13. Mahmoud R, Yousuf T, Aloul F, Zualkernan I (2015a) Internet of things (IoT) security: Current status, challenges and prospective measures. 2015 10th Int Conf Internet Technol Secur Trans (ICITST). https://doi.org/10.1109/icitst.2015.7412116
14. Mahmoud R, Yousuf T, Aloul F, Zualkernan I (2015b) Cyber Security and Internet of Things: Vulnerabilities, Threats, Intruders and Attacks. J Cyber Secur Mobil 4(1):65–88. https://doi.org/10.13052/jcsm2245-1439.414
15. Mavromoustakis CX, Mastorakis G, Batalla JM (Eds.) (2016) Internet of Things (IoT) in 5G Mobile Technologies. Model Optim Sci Technol. Published. https://doi.org/10.1007/978-3-319-30913-2
16. Mielczarek W, Moń T (2015) USB Data Capture and Analysis in Windows Using USBPcap and Wireshark. Comput Netw 431–443. https://doi.org/10.1007/978-3-319-19419-6_41
17. Nawir M, Amir A, Yaakob N, Lynn OB (2016) Internet of Things (IoT): Taxonomy of security attacks. 2016 3rd Int Conf Electron Des (ICED). Published. https://doi.org/10.1109/iced.2016.7804660
18. Neves P, Calé R, Costa M, Gaspar G, Alcaraz-Calero J, Wang Q, Nightingale J, Bernini G, Carrozzo G, Valdivieso N, García Villalba LJ, Barros M, Gravas A, Santos J, Maia R, Preto R (2017) Future mode of operations for 5G – The SELFNET approach enabled by SDN/NFV. Comput Stand Interfaces 54:229–246. https://doi.org/10.1016/j.csi.2016.12.008
19. Noura H, Chehab A, Sleem L, Noura M, Couturier R, Mansour M (2018) One round cipher algorithm for multimedia IoT devices. Multimed Tools Appl 77(14):18383–18413. https://doi.org/10.1007/s11042-018-5660-y
20. Rong C, Zhao G, Yan L, Cayirci E, Cheng H (2013) RFID Security. Comput Inf Secur Handb 345–361. https://doi.org/10.1016/b978-0-12-394397-2.00018-0
21. Sarma R, Barbhuiya F (2019) Internet of Things: Attacks and Defences. 2019 7th Int Conf Smart Comput Commun (ICSCC). https://doi.org/10.1109/icscc.2019.8843649
22. Sun H, Bonetta D, Humer C, Binder W (2018) Efficient dynamic analysis for Node. js. Proc 27th Int Conf Compiler Constr. https://doi.org/10.1145/3178372.3179527
23. Sasipriya S, Vigneshram R (2016) An overview of cognitive radio in 5G wireless communications. 2016 IEEE Int Conf Comput Intell Comput Res (ICCIC). https://doi.org/10.1109/iccic.2016.7919725
24. Usha D, Bobby M (2018) Privacy Issues In Smart Home Devices Using Internet Of Things – A Survey. Int J Adv Res 6(10):566–568. https://doi.org/10.21474/ijar01/7839
25. Zhou X, Tang X (2011) Research and implementation of RSA algorithm for encryption and decryption. Proc 2011 6th Int Forum Strat Technol. https://doi.org/10.1109/ifost.2011.6021216
26. Zunino C, Valenzano A, Obermaisser R, Petersen S (2020) Factory Communications at the Dawn of the Fourth Industrial Revolution. Comput Stand Interfaces 71:103433. https://doi.org/10.1016/j.csi.2020.103433

Authors and Affiliations

Nikola Pavlović[1] · Marko Šarac[1] · Saša Adamović[1] · Muzafer Saračević[2] · Khaleel Ahmad[3] · Nemanja Maček[4] · Deepak Kumar Sharma[5]

Nikola Pavlović
nikola.pavlovic.141@singimail.rs

Marko Šarac
msarac@singidunum.ac.rs

Saša Adamović
sadamovic@singidunum.ac.rs

Muzafer Saračević
muzafers@uninp.edu.rs

Nemanja Maček
nmacek@megatrend.edu.rs

Deepak Kumar Sharma
dk.sharma1982@yahoo.com

[1] Faculty of Informatics and Computing, Singidunum University, Belgrade, Serbia

[2] Department of Computer sciences, University of Novi Pazar, Novi Pazar, Serbia

[3] Department of Computer Science and Information Technology, Maulana Azad National Urdu University, Hyderabad, India

[4] Faculty of Computer Sciences, Megatrend University, Belgrade, Serbia

[5] Department of Information Technology, Netaji Subhas University of Technology, Delhi, India