IJITCE

ISSN 2347- 3657

International Journal of Information Technology & Computer Engineering

Email : ijitce.editor@gmail.com or editor@ijitce.com

International Journal of Information Technology & Computer Engineering

ISSN 2347–3657
Volume 11, Issue 4, Oct 2023

AN IMPLEMENTATION OF ACCESS-CONTROL PROTOCOL FOR IOT HOME SCENARIO

Dr.K.NIRANJAN REDDY [1] ,S.SAI RAM [2] ,K.SAI SHIVA [3] ,B.NIKHIL [4] ,

ABSTRACT: The internet of things comes into our daily life. It connected lots of resource-constrained devices, denoted as smart device, in an Internet-like structure. Considering the computing burden, the CoAP protocol is developed for serving the resource constrained device and maps to HTTP for integration with existing web. In this paper, an access-control protocol will be introduced. The protocol is designed for IoT(Internet of Things) home scenario. Like the most IoT we can see, the IoT home scenario contains lots smart devices which collect some private information from us. To protect those data, an access-control protocol is needed. The protocol is deployed into Contiki OS and evaluated using the powertrace and some other tools. The results shows the protocol we designed takes a little more memory usage than an OAuth based authorisation protocol but smaller power consumption and more suitable for small scale IoT environment.

INTRODUCTION

INTERNET of things (IoT) [1] has undoubtedly made great changes to the lives of people, ranging from intelligent furniture to smart city. IoT connected a large amount of devices and make them accessible remotely through Internet. While the IoT offer use great convenience, it also expose lots of private data on the internet. In home scenario, IoT devices can help users access and control their appliances remotely. The connected appliances may record informations about the user. While lots of your valuable personal privacy information stored in IoT devices like monitor or camera, IoT devices may become target of malicious users. As all the IoT devices in home connected to the Wi-Fi network, an access-control protocol is necessary for protecting users privacy

[1] Assco.prof and Head of the Department, Department Electronics and Communication Engineering,

[2,3,4] Student,Department of Electronics and Communication Engineering,

CMR INSTITUTE OF TECHNOLOGY, KANDLAKOYA VILLAGE, MEDCHAL RD, HYDERABAD, TELANGANA,501401

ISSN 2347–3657
Volume 11, Issue 4, Oct 2023

In the scenario that resource owner grants someone else to access resource, OAuth is very popular as it can authenticate the third-party without giving password. With the communication with an authentication server, the third-party keep a secret token to access users data. The scenario of OAuth used is similar to the IoT home scenario, but the power consumption and memory usage of OAuth is a great burden for IoT devices. So, a new access-control protocol is designed and implemented in our research for IoT home scenario. As my partner have done the design of the protocol, my research is deploying the protocol he designed for home scenario to real machine. In the home scenario, most devices use low-power chips which only have low-power CPU, limited RAM and ROM. As the result of that, deploying the protocol to IoT devices, which are low-power chips, is most challenging part in the whole implementation. This research is to deploying the protocol (device part) to low-power chips

OBJECTIVE:

The objective of this implementation is to develop an access-control protocol for an IoT home scenario that addresses the following goals

1. Enhanced Security: Implement a robust access-control system that ensures only authorized individuals can gain entry into the home, minimizing the risk of unauthorized access and potential security breaches.
2. Remote Monitoring and Control: Enable homeowners to remotely monitor access points and control access privileges using a centralized control application or mobile application, providing convenience and flexibility.
3. User-Friendly Interface: Design a user-friendly interface that allows homeowners to easily manage access privileges, add or remove authorized individuals, and receive real-time notifications about access events.
4. Integration with IoT Devices: Integrate the access-control protocol with other IoT devices within the home, such as smart locks, security cameras, and alarm systems, to create a comprehensive home security solution.

PROBLEM STATEMENT:

The current problem in traditional home access-control systems is their limited

ISSN 2347–3657
Volume 11, Issue 4, Oct 2023

functionality, lack of remote access, and often complex management processes. Traditional lock-and-key mechanisms or basic electronic access systems do not provide the desired level of security or convenience for modern homeowners. The lack of real-time monitoring and control capabilities restricts homeowners' ability to manage access remotely, leading to potential security vulnerabilities.

LITERATURE REVIEW

Constrained Application Protocol is a communication protocol based on REST model, which designed for low-power devices. It is a specialized Web transfer protocol for use with constrained nodes and constrained network. Because both CoAP and HTTP follows the REST model, they are very similar. [2] The server offer resources under a URL and client can access using the methods same as Http, e.g. GET, PUT, POST, and DELETE. As it designed based on REST model, it can be used as a common communication protocol which can transfer data between different devices in the IoT system. CoAP is designed to be used on microcontroller with as low as 10 KB of RAM and 100 KB of code space. As my research is deploying the protocol to a low-power chip, a resource-saved protocol is really helpful. A research about using CoAP to implement a lightweight security scheme for vehicle tracking system was done at 2013.They stated that the overhead security of CoAP is too high. [3] In their design, they use a payload embedded low cost symmetric-key based robust authentication and key management mechanism instead of handshaking and ciphersuite agreement of standard TLS and DTLS. For better performance in IoT system, they also modificate the header of CoAP to implement security special mechanism which better than the standard one. To implement the CoAP with their security mechanism, they introduce a new option AUTH in CoAP header to enable the security function. In their performance test, the security scheme they combined with CoAP only have a little effect on performance. The latency increase a little bit because of the header of CoAP become larger. It has a low overhead due to the authentication based on payload embedded symmetric key. Implementing the security scheme based on payload embedded content is an innovative ideal of modification of the CoAP. It can save resource which is really precious for devices in IoT system. In other way, it doesnt affect the performance of communication. Their research is for vehicle tracking system so that latency

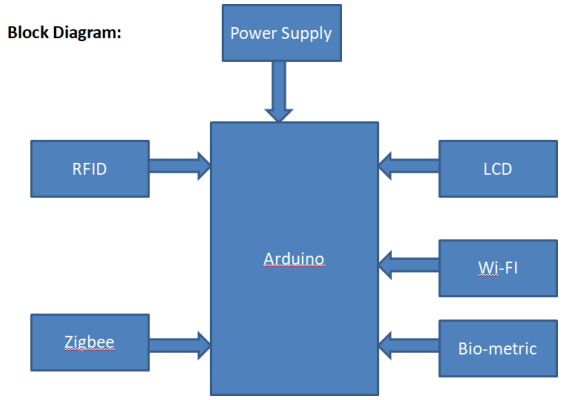
ISSN 2347–3657
Volume 11, Issue 4, Oct 2023

affect a lot. However, in home scenario, the latency is low because all devices connected to the same network without any long-distance transfer. So the disadvantage (increasing latency) of their design has no effect on our scenario. In performance aspect, this design fit our scenario exactly There is a research about implementing a low-power CoAP for contiki operating system in 2011. [4]. Their aim is to achieve high power efficiency for CoAP communication on Contiki platform. On typical IoT platform, one of components which are most power-consuming is the radio transceiver. The power consumption of listening packets is as expensive as receiving packets. [4] To reduce the waste of power, several radio duty cycling(RDC) algorithms have been designed. RDC algorithms allows node to turns off radio chip for most of time while still being able to send and receive message. They also took advantage of the Contiki REST layer abstraction to modify the CoAP : they separate the acknowledge frame from response data. With this modification, the client doesnt need to resend the data during the time the server processing the request it receive. It enabling long processing times and avoiding unnecessary retransmission. Their low-power CoAP implementation for Contiki achieve a high energy efficiency at the cost of a higher latency. The radio chip on server doesnt keep listening all the time, hence not all packets sent by client can be received and increasement of latency. In out research, in home scenario, most IoT devices uses battery as power. High energy efficiency means the devices can keep working longer. This research shows a clear way to reduce the energy cost of devices. The energy efficiency may become a concern in further research but not in current state. An OAuth-based authorisation service architecture is published in 2015 which called IoT-OAS. [5] The goal of the architecture is revlieving smart object form burden of handling large amount of authorization-related information. In the architecture, an external authentication service is set to handle authorization information. And the smart object can keep the application logic as simple as possible while outsourcing all the authentication functions. Considering the computing burden of using HTTP, they using different communication protocol among different roles in the architecture. The somart object only deal with message on COAP protocol. An hybrid gateway-based communication is used in their implementation. The application protocol extternal client used is different from that used by smart object. The gateway manage the communication in authentication

ISSN 2347–3657
Volume 11, Issue 4, Oct 2023

process. It translate the incomming message from HTTP and CoAP to the other

IMPLEMENTATION



Communication A gateway-based communication protocol is used in our implementation. There are two communication protocol we used under the same network. Considering the power limitation, we can not allow the IoT device running two server at the same time. To solve the problem, we set a HTTP-CoAP proxy on gateway. The data flows goes as Fig.4 The proxy is on the gateway. The packet owner/visitor send to device will go to the proxy first. Then the proxy translate it into a CoAP packet and send it to device. So does the packet send to owner/visitor by device

Implementation Environment Summary Gateway, owner and visitor supposes to be deployed to high-power environment: the gateway should be deployed to a router or a server which can running all the time. The owner and visitor program should be deployed to smartphone or desktop. In the home scenario, most devices use low-power chips which only have low frequencies CPU, limited RAM and ROM space. As the result of that, deploying the protocol to IoT devices is the most challenging part in the whole implementation.

Counter on Device The access control is meaningless if an authorized token can be used forever. Most authorisation protocol has a timestamp which indicates when the authorisation will expire. In OAuth 2, a lifetime is used in authorisation token to indicate when the token expires. [7] Timer is the critical function module in the timestamp based key expiration. However, as target of the implementation is low power device, the CPU on chips may not contain timer on its own. The CPU without timer can not expire an authorisation token if token expiration mechanism is timestamp-based. Even the CPU has timer, more network transmissions are required for synchronize time with time server. For better portability of the device part program, the timestamp-based token expiration can not be used. A counterbased expiration mechanism is used on device program. The detail information of the mechanism can be seen in Fig.5. Every

ISSN 2347–3657
Volume 11, Issue 4, Oct 2023

device has its own counter. The length of counter is 4 byte, so the range of the counter number is from 0 to 232. When the device booting, the gateway would send a random initial counter number to the device. The device will set its current counter number to this initial counter number. In the Fig 5, the current counter number is 13. The default counter window size is 5, which means the counter number range from current counter number minus 5 to current counter number minus 1 is valid counter number. The device also keep an array of integer which named counterWindow. It record how many times left each valid counter number can be accessed. The index in array can map to specific counter number. Values in this array are times counter number their index map to can be used. valid tokens at the same time is constrained by the size of counter window. For example, the default counter window is 5. So there is 5 tokens are valid at most. In home scenario, 5 valid token at the same time should be enough. It can also be customize basing on different need. In this mechanism, one more user at the same time only take one byte more in devices RAM. So the protocol can support large amount of visitor at the same time. The counter-based token expiration mechanism has one drawback. The token is exactly the same for the same counter number in same device. When the counter number goes to maximum, it will become zero in next time. So the counter number will finally go back to a number it used before if the counter didnt be reset. For prevent this kind of attack, we set the length of counter to 4 bytes. The counter number ranges from 0 to 232. And when device reboot, it will get a new random counter number from gateway. With these two configuration, the counter number became almost impossible to use some number it used.

CONCLUSION In this paper, we proposed implementation of the protocol designed by my partner. The implementation details and some challenges in implementation has been described. The implementation approach has been applied to significant IoT home scenario device characterized by constrained memory space and limited computational power. The response time of the server is also considered and the experimental result shows the response time of the device is feasible. In further research, we will try to compress memory footprint of Contiki OS and the Erbium CoAP server. With lower memory footprint, we can deploy our protocol to chips whose power is lower and make this implementation more feasible for IoT home scenario.

ISSN 2347–3657
Volume 11, Issue 4, Oct 2023

REFERENCES

[1] F. Liu, "A survey of the internet of things," in International Conference on E-Business Intelligence, 2010.

[2] Z. Shelby, K. Hartke, and C. Bormann, "The constrained application protocol (coap)," 2014.

[3] A. Ukil, S. Bandyopadhyay, A. Bhattacharyya, and A. Pal, "Lightweight security scheme for vehicle tracking system using coap," in International Workshop on Adaptive Security, 2013, p. 3.

[4] M. Kovatsch, S. Duquennoy, and A. Dunkels, "A low-power coap for contiki," in IEEE Eighth International Conference on Mobile Ad-Hoc and Sensor Systems, 2011, pp. 855–860. [5] S. Cirani, M. Picone, P. Gonizzi, and L. Veltri, "Iot-oas: An oauth-based authorization service architecture for secure services in iot scenarios," IEEE Sensors Journal, vol. 15, no. 2, pp. 1224–1234, 2015.

[6] "The contiki operating system," 2016, http://www.contiki- os.org.

[7] D. Hardt, "The oauth 2.0 authorization framework," 2012.

[8] A. Dunkels, J. Eriksson, N. Finne, and N. Tsiftes, "Powertrace : Network-level power profiling for low-power wireless networks lowpower wireless," Swedish Institute of Computer Science, 2011.

[9] "Contiki hardware," http://www.contiki-os.org/hardware.html.

[10] T. Instruments, "Cc2420 datasheet," Reference SWRS041B, 2007.

[11] T. Instruments, "Msp430f5438 datasheet," Reference SLAS655B, 2010.