Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 78 (2016) 646 – 651

Procedia Computer Science

International Conference on Information Security & Privacy (ICISP2015), 11-12 December 2015, Nagpur, INDIA

Secure and Cost-effective Application Layer Protocol with Authentication Interoperability for IOT

Ajit A. Chavan[a], Mininath K. Nighot[b]

[a] *Department of Computer Engg., KJCOEMR,Pune-,India.*
[b]*Assistant Professor,Department of Computer Engg.,KJCOEMR,Pune-,India.*

Abstract

Internet of Things is future trend that assure to enhance and upgrades our everyday living by utilizing intelligent objects and sensors collectively. These devices are supposed to make use of constrained application protocol (CoAP) to communicate at application layer. Communication security is mainstay of constrained environments. Confidential communication within constrained devices will be carried out by secure CoAP (CoAPs) which makes use of DTLS protocol. To cope with constrained devices we use integration of CoAP and DTLS compressed by following 6LoWPAN standards. Compressed DTLS minimizes packet size and possibly avoids fragmentation. In addition to this we used the raw public key concept over DTLS to authenticate the multivendor constrained devices. Evaluation results shows that CoAPs with Raw Public Key provides communication security and authentication portability in multivendor environment at minimal energy consumption. CoAPs with Raw Public Key improves the interoperability as well as lifetime of network.

Peer-review under responsibility of organizing committee of the ICISP2015

Keywords:CoAP, DTLS, CoAPs, Generic Header Compression, Next Header Compression, 6LoWPAN, IoT

1. Introduction

The IETF standardize IPV6 over Low-power Wireless Personal Area Network (6LoWPAN) to provide routing in low power and lossy wireless sensor networks. In IoT smart devices are interconnected through such IPV6 protocol. Due to congestion control algorithm and low power and lossy links TCP performance proves to be inefficient in wireless sensor networks. Thus preferably connectionless UDP is used in IoT. Furthermore HTTP protocol which uses TCP as underlying protocol to run, is inefficient in constrained environments. The IETF proposed connectionless and compact Constrained Application Protocol as new standard for IoT[1]. CoAP is especially

Peer-review under responsibility of organizing committee of the ICISP2015
doi:10.1016/j.procs.2016.02.112

designed to satisfy the requirement such as low overhead, simplicity and multicast communication support in resource constrained environments.

CoAP proposes to use DTLS as the underlying security protocol for data encryption, authentication, integrity protection and for automatic key management[2]. CoAP with DTLS is called as secure CoAP and abbreviated as CoAPs. DTLS requires number of message exchanges to establish secure connection between interacting nodes. Though DTLS provides wide range of cryptographic services, it was originally designed for networks in which message length was not main criteria. That's why use of DTLS as it is for constrained devices is inefficient. To comply with constrained devices and networks, 6LoWPAN header compression mechanisms are defined. DTLS supports partial interoperability for IoT devices in PSK mode in which manufactures have to preshare keys to provide secure communication. Obtaining such trust in multivendor environment is difficult. On the other hand supporting X.509 based Public Key Infrastructure (PKIX) is challenging in IoT environments.

In this paper we use the concept of raw public keys with compressed DTLS in CoAP[3]. This integration provides three benefits. First, energy saving by reducing message size. Second, avoids 6LoWPAN fragmentation at link layer MTU when size of datagram is larger. Third, reduces the burden of constrained devices from storing and transmitting X.509 certificates while doing DTLS handshake.

2. Literature Survey

In this section we discuss the approaches that aim to provide end to end security solutions for constrained environments. Communication between nodes in 6LoWPAN networks and conventional Internet secured by using compressed IPsec[4]. Encapsulating Security Payload (ESP) extension headers and Authentication Header (AH) are compressed using Next Header Compression (NHC). Above solution is extended with implementation of IPSec in tunnel mode[5]. Trusted Platform Module provides hardware support for RSA algorithm, used with DTLS in 6LoWPAN[6]. As they have used DTLS as it is, this proves to be inefficient because of redundant bits in message. Keoh*et al*[7] proposed the architecture to provide the secure network access and unicast, multicast key management with extended DTLS. Elliptic Curve Cryptography implemented on constrained devices do not consider protocol implementation, DTLS, 6LoWPAN shim layer and application code[8].

3. Background

It is difficult for resource constrained devices connect in secure and trustworthy manner, because of heterogeneity in IoT networks. In this section we provide brief introduction of technologies used in the design of compact CoAPs with raw public key.

3.1. CoAP

CoAP is application layer protocol especially designed for constrained devices. CoAP uses UDP to run as underlying protocol. CoAP provides REST interface to provide efficient communication among the devices. To protect CoAP communication, DTLS has been selected as basic security protocol. CoAP using DTLS security is termed as secured CoAP (CoAPs) like the TLS secured HTTP as HTTPs. CoAP uses Universal Resource Identifier (URI) to access the resources on destination device. CoAP protocol uses "coap" URI scheme. CoAP securely accesses the web resource on destination device as follows:

coaps://IPV6address:port/Resource_name

CoAP is simply a request-response type protocol and provides both types of communication viz. reliable and unreliable. Devices using the CoAP may act as client, server or both. The reasons that a new protocol is defined for constrained IP networks, instead of simply reusing HTTP, is to greatly reduce overhead in implementation complexity and to reduce the bandwidth requirements. Such data reduction also helps to increase reliability by reducing link layer fragmentation and reduce latency in typical low-power lossy wireless networks, such as IEEE 802.15.4.

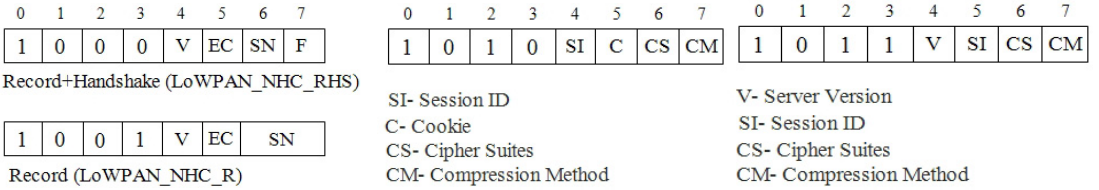
3.2. DTLS

DTLS is complete security protocol that performs key exchange, authentication and securing application data by using algorithms and negotiated keying material. DTLS contains the two layers, lower layer and upper layer[2]. Lower layer contains the record protocol and upper layer contains one of the three protocols, Handshake, Alert and ChangeCipherSpec or data. The handshake process uses ChangeCipherSpec to indicate record protocol should protect subsequent messages by using cipher suite. Alert protocol is used to communicate error message between nodes. Record header contains the fragment field and content type. Fragment field contains one of three handshake protocol, alert protocol, ChangeCipherSpec protocol or data based on value contained in content type. The record header protocol has the responsibility of protecting upper layer protocols. DTLS handshake protocol is chatty and contains number of message exchanges in asynchronous manner. The handshake messages are organized in flights and used to exchange the information like cipher suites, security keys and compression methods.

3.3. 6LoWPAN

The 6LoWPAN standard proposed header compression and fragmentation scheme of IPV6 datagrams for 6LoWPAN networks[9].This standard provides two techniques Next header compression (NHC) and IP header compression (IPHC). The IPHC encoding compresses the header length up to 7 bytes in multi hop networks. The IPV6 extension header and UDP header are compressed by using NHC. By using 6LoWPAN standard for NHC only headers up to UDP can be compressed. Absence of NH bit in NHC for UDP mentions compressed UDP, so new NHC technique must be defined.

3.4. DTLS Compression[3]



(a) (b) (c)

Fig. 1LoWPAN Encoding for different DTLS Headers[3].(a) LoWPAN_NHC encoding Record+Handshake Header and Record Header Only.(b)LoWPAN_ NHC encoding for Clienthello Message. (c)LoWPAN_NHC encoding for ServerHello message

The Record protocol always adds the 13 bytes header to each outgoing packet from the device that uses DTLS protocol. Similarly handshake protocol adds 12 bytes of header to each handshake message. 6LoWPAN NHC reduces the header lengths of record and handshake up to 5 and 3 bytes respectively[3]. This is applicable to fresh handshake only.

Fig. 1 shows various 6LoWPAN_NHC encodings for DTLS headers. Fig. 1(a) shows encodings for record and handshake headers as LoWPAN_NHC_RHS and LoWPAN_NHC_R as encoding for record only. The Version, Epoch, Sequence Number and Fragment can be compressed based on value they hold. EC value can be 0 or 1, that's why most of the times 8 bit EC is used. Two bits of SN in LoWPAN_NHC_R encoding allows 16,24,32 or 48 bit sequence number can be used. If F is 0 then handshake message is not fragmented and the fields fragment length and fragment offset are omitted. If 1 then both fields are carried inline.

Fig. 1(b) represents the encoding for ClientHello message as (LoWPAN_NHC_C). Is SI field is 0, new handshake is initiated, then only random fields needs to be transmitted and all other are omitted. The Compression method and Ciphersuite have their default values and therefore do not need to be negotiated. The random field in the ClientHello is always carried inline whereas the version field is always omitted. Fig. 1(c) represents encoding for ServerHello message as LoWPAN_NHC_SH and is similar to ClientHello message. All other handshake messages are carried inline.

4. Proposed System

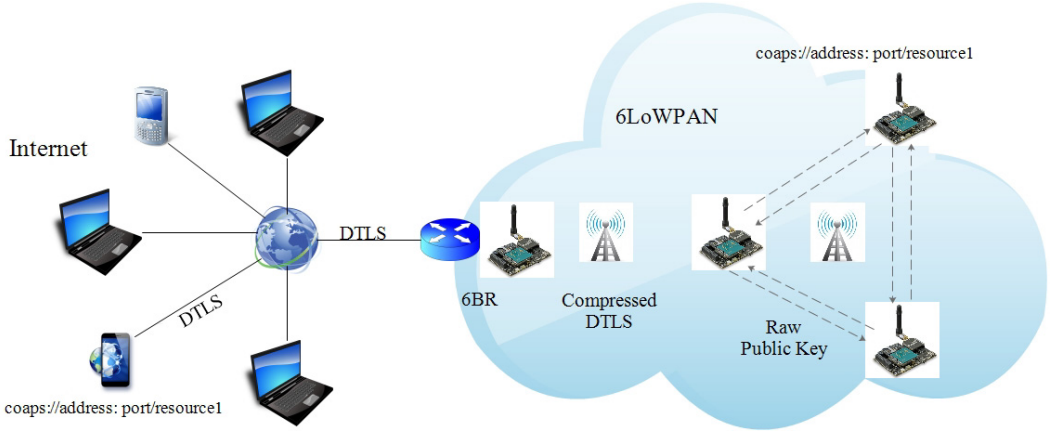


Fig. 2Set up of IoT containing constrained devices using coaps with Raw Public Key

Fig. 2 shows typical network setup for IoT, in which 6LoWPAN network containing CoAPs enabled devices are connected through 6LoWPAN Border Router (6LBR) with conventional Internet. As shown in figure header compression is applied in the 6LoWPAN network only, between constrained devices and 6LBR. To allow the devices in multivendor environment to authenticate each other, the concept of raw public key has been introduced instead of X.509 certificates[10]. The TLS client device is configured with raw public key and also is able to process the raw public key accepted from server. Client initiates handshake process by sending Client_Hello message to the server.Server Verifies the client and reply Hello_Verify_Req back to client. Client verifies server and sends C_Hello to server, also server replies with S_Hello to client. After Key exchange process if server requires client's authentication, server can send CertificateRequest message to client requesting raw public key from client. Client who has raw public key configured, returns it in the Certificate payload to the server.Also server provides raw public key in certificate payload back to client.

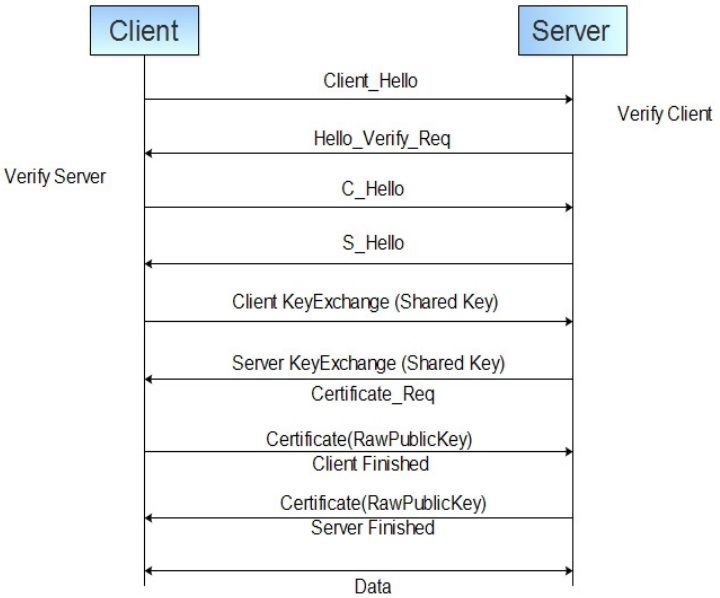


Fig.3CoAPs using Raw Public Key.

5. Evaluation

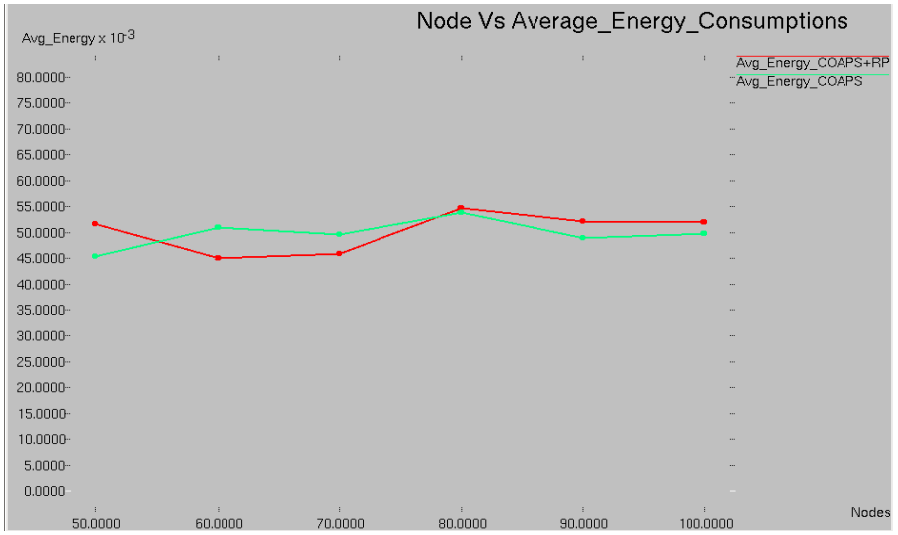


Fig.4 (a) Shows Average Energy Consumption in the ntwork.

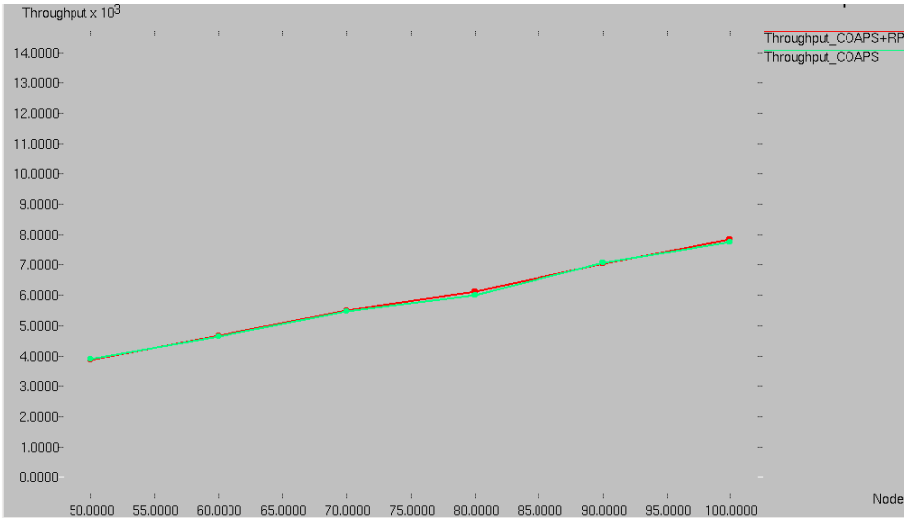


Fig. 4(b) Shows Throughput of the nodes for different protocols.

We have implemented this system in different scenario considering variable number of nodes in the network. Fig. 4(a) shows the average energy consumption in the network. We compared CoAPs and CoAPs with Raw Public Key (CoAPs+RPK), results shows that average energy consumption in both cases does not differ much more. So implementation of Raw Public Keys over CoAPs does not affect the performance. Fig 4(b) shows the comparison of throughput in both the cases does not vary much more. CoAPs+RPK does not consume maximum energy than CoAPs, allowing maximum lifetime for nodes as well as network, also enhances interoperability among the nodes.

6. Conclusion

Secure CoAP is the basic need of resource constrained devices in real IoT environment. Datagram Transport Layer Security is the standard protocol to empower secure CoAP. New 6LoWPAN header compression schemes reduces overhead of DTLS protocol. Also use of Raw Public Key by DTLS provides the authentication portability at device level without consuming too much energy. CoAPs using compressed DTLS with raw public key is efficient in energy consumption of nodes, memory requirement, network response and authentication interoperability.

References

1. Z. Shelby, K. Hartke, C. Bormann, and B. Frank. (2013, May). Constrained Application Protocol (CoAP). Internet-Draft draft-ietf-corecoap-16[Online].Available: http://datatracker.ietf.org/drafts/current/.
2. Datagram Transport Layer Security Version 1.2, RFC Standard 6347,Jan. 2012.
3. European Seventh Framework Programme FP7-288879-Collaborative Project.
4. S. Raza, S. Duquennoy, A. Chung, D. Yazar, T. Voigt, and U. Roedig,. Securing communication in 6LoWPAN with compressed IPsec. in *Proc. 7th Int. Conf. DCOSS*, Barcelona, Spain, Jun. 2011,pp. 1–8.
5. J. Granjal, E. Monteiro, and J. S. Silva. Network-layer security for the internet of things using TinyOS and BLIP. *Int. J. Commun. Syst.*, 2012,doi: 10.1002/dac.2444.
6. T. Kothmayr, C. Schmitt, W. Hu, M. Brunig, and G. Carle, "A DTLS based end-to-end security architecture for the internet of things with two-way authentication," in *Proc. IEEE 37th Conf. Local Comput. Netw.Workshops*, Oct. 2012, pp. 956–963.
7. S. Keoh, S. Kumar, and O. Garcia-Morchon. (2013, Feb.). Securing the IP-Based Internet of Things with DTLS [Online]. Available:http://www.ietf.org/1id-abstracts.html.
8. A. Liu and P. Ning. Tinyecc: A configurable library for elliptic curve cryptography in wireless sensor networks. in Proc. Int. Conf. Inf. Process.SensorNetw. (IPSN'08), 2008, pp. 245–256.
9. IPv6 Over Low-Power Wireless Personal Area Networks (6LoWPANs):Overview, Assumptions, Problem Statement, and Goals. RFC Standard 4919, Aug. 2007.
10. Using raw public keys in Transport Layer Security and Datagram Transport Layer Security. RFC Standard 7250,June 2014.