CPSC 490 Project Proposal: Secure Communication with Constrained IoT Devices

Melina Delgado
May 2019

1 Introduction

Everyday objects and non-standard devices that capable of wireless communication, devices otherwise known as the Internet of Things (IoT), have become more and more pervasive in society as their demand increases. Tiny devices, such as sensors and actuators, are being embedded into everyday objects. Sensor nodes capture events in the physical world and transfer them into the virtual world so that they can be processed and acted upon, while actuators alter the real world based on data obtained from the physical world or cyber world, periodically or spontaneously [1]. Some applications of these very small IoT devices may include motion-activated doors, light bulbs and switches, and more.

Much work has been accomplished in the last decade to integrate IoT devices with the internet. However, some IoT devices (such as sensors and actuators) are “constrained” because they are physically constrained by size, weight, available power and energy [2]. Their limited power, memory, and processing resources present a few major difficulties: existing application protocols (e.g. HTTP) are too heavyweight for constrained IoT devices, and sometimes, security considerations have yet to be implemented. Constrained networks face the same problem, where constraints may include low achievable bit rate/throughput, high packet loss, and more.

To deal with some of these difficulties, the Constrained Application Protocol (CoAP) was defined. CoAP is a specialized web transfer protocol for use with constrained nodes and constrained networks [3]. CoAP provides similar functionality for constrained networks as the one HTTP provides for conventional networks. For example, CoAP uses PUT, POST, GET, and DELETE methods to communicate, update or remove resources hosted by constrained nodes [1].

Currently, many IoT devices use Enrollment over Secure Transport (EST) to securely obtain certificates. This is basic secure enrollment on IoT devices. However, on constrained IoT, we need a more lightweight solution. This public key infrastructure (PKI) technology is too heavy for resource-constrained devices, for EST messages are too large. I plan to focus on a protocol that combats this issue: EST over Secure CoAP. The IETF Internet-Draft explains
a way to “transport EST payloads over secure CoAP (EST-coaps) to allow low-resource constrained devices to use existing EST functionality for provisioning certificates” [4].

2 Background

2.1 The Constrained Application Protocol (CoAP)

The Constrained Application Protocol (CoAP) is a specialized web transfer protocol for use with constrained nodes and constrained networks. CoAP designs a generic web protocol for constrained nodes often have 8-bit microcontrollers (with limited ROM and RAM), and for constrained networks, such as IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs), that have a typical throughput of 10s of kbit/s. CoAP easily interfaces with HTTP by providing similar request/response interactions and including Web concepts such as URIs. The protocol was designed for machine-to-machine (M2M) applications [3].

CoAP realizes a subset of Representational State Transfer (REST) architecture common with HTTP, optimized for M2M applications. The main features of CoAP include UDP binding (with optional reliability supporting unicast and multicast requests), asynchronous message exchanges, low overhead and parsing complexity, URI and Content-type support, simple proxy and caching capabilities, a stateless HTTP mapping (allows proxies to be built providing access to CoAP resources via HTTP or for HTTP to be realized alternatively over CoAP), security binding to Datagram Transport Layer Security (DTLS), fulfills M2M requirements in constrained environments [3].

The CoAP interaction model is similar to the client/server model of HTTP, except a CoAP implementation may act in both client and server roles. A CoAP request is equivalent to that of HTTP, sent by a client to request an action on a resource, then, the server sends a response with a Response Code, which may include a resource representation [3].

This message exchange is asynchronous, dealt over a datagram oriented transport such as User Datagram Protocol (UDP). Its layer of messages support optional reliability; CoAP’s four types of messages are Confirmable (carry requests), Non-confirmable (carry requests), Acknowledgement, and Reset, where responses can be carried in any of these (or piggybacked in Acknowledgements). CoAP is a single protocol with a single layer, where requests, responses, and messages are just features of the header [3].

CoAP is bound with DTLS over UDP (just as HTTP is secured using Transport Layer Security, TLS, over TCP) and has four security modes: NoSec (no protocol-level security), PreSharedKey (enabled DTLS with pre-shared keys, where there may be one key for each node the CoAP node needs to communicate with, or, one key authenticates a peer as a member of a group), RawPublicKey (enabled DTLS where the device has an identity and an asymmetric key pair without a certificate), Certificate (enabled DTLS where the device has an asymmetric key pair with an X.509 certificate) [3].
In “NoSec” mode, the system sends packets over normal UDP over IP, indicated by the “coap” scheme and the CoAP default port. The other three security modes are achieved using DTLS, indicated by the “coaps” scheme and the DTLS-secured CoAP default port. In some constrained nodes and networks, and depending on the cipher suite, all modes of DTLS may not be applicable, depending on whether sufficient resources are available [3].

The work on Constrained RESTful Environments (CoRE) and its support for certificate status checking requires further study. A standard is defined in RFC 7252, but a protocol is not proposed.

2.2 Enrollment over Secure Transport (EST)

Public key infrastructure (PKI) is generally used to authenticate the identity of users and devices by means of publicly signed key pairs in the form of digital certificates. EST is a certificate management protocol targeting PKI clients that need to acquire client certificates and associated Certification Authority (CA) certificates. It supports client-generated public/private key pairs and key pairs generated by the CA. EST does not transfer messages over HTTP/DTLS/UDP; it does so over HTTP/TLS/TCP [5].

Both the EST clients and server can mutually authorize and authenticate each other. When the client initiates a TLS-secured HTTP session, a specific EST service is requested, the client and server authenticate and verify each others’ authorization, and then the server processes the client request [5].

When the EST client requests a CA certificate, it authenticates and verifies the authorization of the EST server. It can also obtain its own certificate or renew/rekey its existing certificate by submitting an enrollment request to the EST server, which will then authenticate and authorize the EST client [5].

2.3 EST over Secure CoAP (EST-coaps)

Because low-resource devices may use CoAP for message exchange, EST-coaps defines how to transport EST payloads over secure CoAP to allow low-resource constrained devices to use EST [4].

EST-coaps uses DTLS for transport security, CoAP for message transfer and signalling, and UDP for transport, rather than TLS, HTTP, and TCP. EST-coaps supports certificate-based client Authentication. EST-coaps can transport certificates and private keys, where certificates are responses to enrollment or re-enrollment requests or trusted certificate list requests, where private keys are transported as responses to server-side key generation requests [4].

3 Problems

The Certificate security mode has not been implemented–its standard is only outlined in CoAP. It makes suggestions on how to manage certificates with CoAP, but it is not currently defined.
EST-coaps is an Internet-Draft proposing a solution to this problem. I plan to design and implement a constrained IoT device running EST-coaps as described in the Internet-Draft.

4 Deliverables

1. EST-coaps implementation
2. IoT device using the EST-coaps protocol
3. M2M communication with EST-coaps

References


