SRUP: The Secure Remote Update Protocol[?]

Speaker: Hsu, Yin-Hong

01 10, 2017

Outline

Introduction

Publish/Subscribe MQTT

SRUP

Protocol detail

References

C2 and software update for IOT

- Enable the devices to receive and react to messages pertaining to the device itself
- This enables a human-operator or autonomous agent to interact with the device remotely

Software Update in the Internet of Things

- data-driven software
  - Application’s behaviour is determined by a combination of software and data
- using this type of approach can potentially simplify many routine software updates

The challenge of remote software update

- IoT device often have their primary UI provided by a network connection
- it must be impossible for the update process to cause the device be unusable
- be able to track whether a particular device has been updated

Software Update paradigm

- two means to initiate a software update
  - pushing the software to the device
  - triggering the device to fetch the software update itself
- use the second approach according to the first one is insucure
- to monitor a known repo, waiting for an indication that updates are available

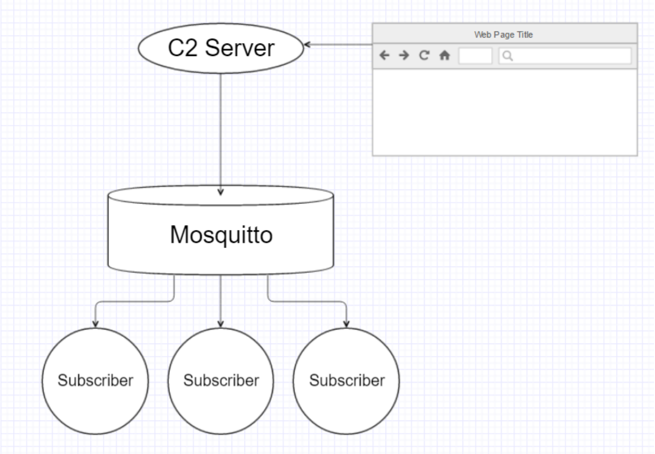
Cryptograph Security Considerations

- the URL of data can be found, can be verified by TLS
- their content checked for integrity using a hashing function (SHA2)
- to monitor a known repo, waiting for an indication that updates are available

MQTT

- a lightweight brokered publish/subscribe protocol
- message will routed via broker to subscriber while publisher issue a message
- this research use MOSQUITTO as broker
  - implements the MQTT protocol v3.1 and v3.1.1
  - open source

SRUP


C2 Server
Web Page Title
Mosquitto
Subscriber
Subscriber
Subscriber

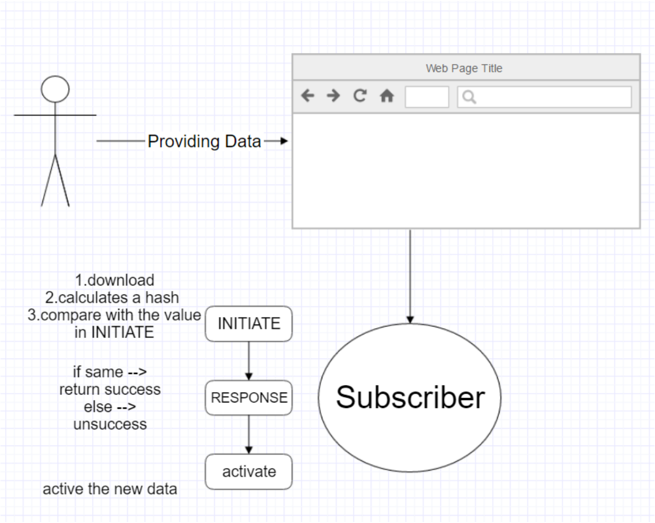
SRUP

- although SRUP can be used in support of software updates on IoT devices, it is not a protocol for software updates
- SRUP does not attempt to provide a protocol for the actual download of the data but handled over a conventional HTTPS connection to the server providing the software
- previous version of the software would not be overwritten due to the purpose of recovery

SRUP - advantage

- it makes it easy to target individual devices for specific updates
- provides a confirmable mechanism ensure that the update has been successfully and correctly received by device

SRUP - action



Web Page Title

Providing Data

1.download
2.calculates a hash
3.compare with the value
in INITIATE

INITIATE

if same -->
return success
else -->
unsuccess

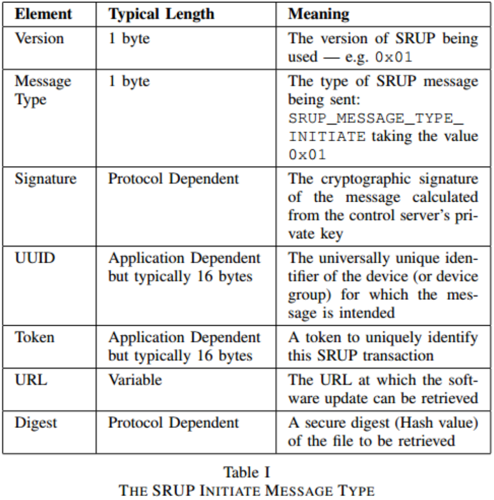
RESPONSE

active the new data

activate

Subscriber

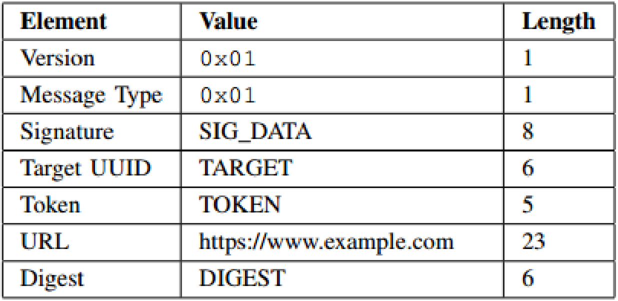
Initiate MSG



Element	Typical Length	Meaning
Version	1 byte	The version of SRUP being used — e.g. `0x01`
Message Type	1 byte	The type of SRUP message being sent: `SRUP_MESSAGE_TYPE_INITIATE` taking the value `0x01`
Signature	Protocol Dependent	The cryptographic signature of the message calculated from the control server's private key
UUID	Application Dependent but typically 16 bytes	The universally unique identifier of the device (or device group) for which the message is intended
Token	Application Dependent but typically 16 bytes	A token to uniquely identify this SRUP transaction
URL	Variable	The URL at which the software update can be retrieved
Digest	Protocol Dependent	A secure digest (Hash value) of the file to be retrieved

Table I
THE SRUP INITIATE MESSAGE TYPE

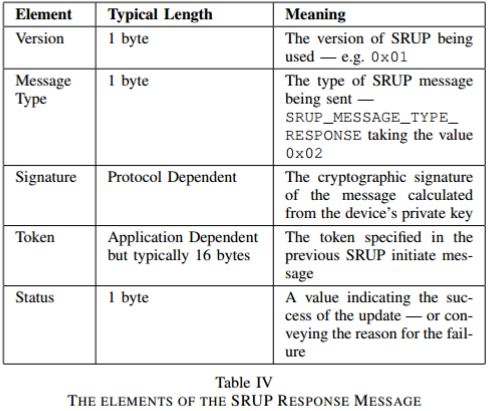
Initiate MSG - example



Element	Value	Length
Version	0x01	1
Message Type	0x01	1
Signature	SIG_DATA	8
Target UUID	TARGET	6
Token	TOKEN	5
URL	https://www.example.com	23
Digest	DIGEST	6

Table II
THE ELEMENTS OF AN EXAMPLE SRUP INITIATE MESSAGE

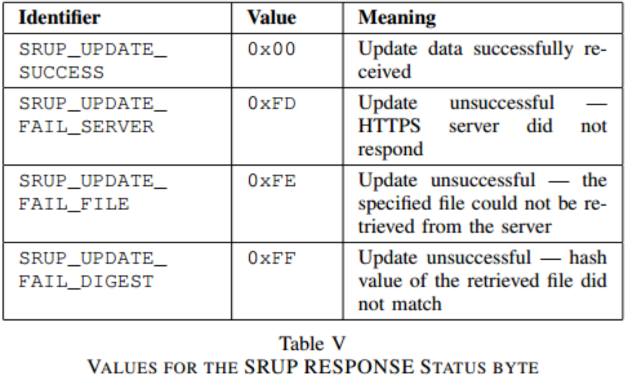
Response MSG



Element	Typical Length	Meaning
Version	1 byte	The version of SRUP being used — e.g. `0x01`
Message Type	1 byte	The type of SRUP message being sent — `SRUP_MESSAGE_TYPE_RESPONSE` taking the value `0x02`
Signature	Protocol Dependent	The cryptographic signature of the message calculated from the device's private key
Token	Application Dependent but typically 16 bytes	The token specified in the previous SRUP initiate message
Status	1 byte	A value indicating the success of the update — or conveying the reason for the failure

Table IV
THE ELEMENTS OF THE SRUP RESPONSE MESSAGE

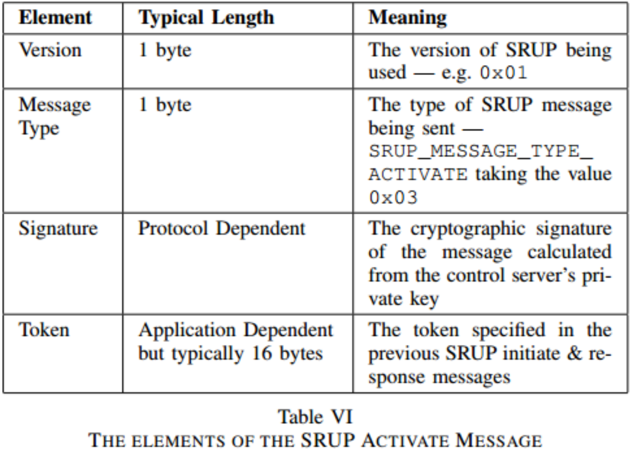
Response MSG - example



Identifier	Value	Meaning
SRUP_UPDATE_ SUCCESS	0x00	Update data successfully received
SRUP_UPDATE_ FAIL_SERVER	0xFD	Update unsuccessful — HTTPS server did not respond
SRUP_UPDATE_ FAIL_FILE	0xFE	Update unsuccessful — the specified file could not be retrieved from the server
SRUP_UPDATE_ FAIL_DIGEST	0xFF	Update unsuccessful — hash value of the retrieved file did not match

Table V
VALUES FOR THE SRUP RESPONSE STATUS BYTE

Activatew MSG - example



Element	Typical Length	Meaning
Version	1 byte	The version of SRUP being used — e.g. `0x01`
Message Type	1 byte	The type of SRUP message being sent — `SRUP_MESSAGE_TYPE_ACTIVATE` taking the value `0x03`
Signature	Protocol Dependent	The cryptographic signature of the message calculated from the control server's private key
Token	Application Dependent but typically 16 bytes	The token specified in the previous SRUP initiate & response messages

Table VI
THE ELEMENTS OF THE SRUP ACTIVATE MESSAGE

SRUP - advantage

- it makes it easy to target individual devices for specific updates
- provides a confirmable mechanism ensure that the update has been successfully and correctly received by device

References

Thanks for Your Attentions