CoRE Working Group   Namhi Kang
Internet Draft   Duksung Women's University
Intended status: Standard Track   October 15, 2013
Expires: April 15, 2014

Secure initial-key reconfiguration for resource constrained devices
draft-kang-core-secure-reconfiguration-00

Abstract

This document presents a secure method to configure a key for a node when it initially joins to network that is currently in operation. The method is suited for a scenario, where resource constrained objects are interconnected with each other and thus form a network called Internet of Things. It is assumed that communications for all nodes are based on TCP/IP protocols and some of the nodes use the constrained application protocol (CoAP). The method does not cover all operations of secure bootstrapping, but it is intended to securely support self-reconfiguration of the pre-installed temporary key of new node.

Status of this Memo

Copyright Notice

Table of Contents

1. Introduction 4
2. Terminology 5
3. System Architecture 6
4. Process Flow 8
5. Security Considerations 9
6. IANA Considerations 10
7. Acknowledgments 10
8. References 10
8.1. Normative References 10
8.2. Informative References 11

1. Introduction

A rapidly growing number and various types of devices including smart small things such as sensors and actuators are trying to connect with Internet as time goes by. This draft presents a simple but efficient approach to reconfigure a secure key for resource constrained small things that are often defined as network nodes having 8 bit processing microcontrollers with limited amounts of memory. The network is also constrained one (e.g. 6LoWPAN having high packet error rates and a typical throughput of 10s of kbit/s) [CoAP].

Pre-shared key (PSK) based secure schemes are well known and frequently used for various security services in Internet. All such schemes strictly assume that the PSK is only known to two entities involved in current security service. Consequently, the security of the schemes are compromised if the assumption is broken.

However, it is still not clear how PSK of resource constrained node can be initially configured in a secure manner in Internet of things (IoT). Typically, things used for IoT might be manufactured and installed by different subjects (simply person) [SecCons]. That is, in general situation, a system administrator may make orders to several different installers. After that, each of the installers purchases one or more different set of things from one or more different manufacturers. It is also unlikely that a single subject installs all nodes used for a large application domain (e.g. all nodes in huge building).

This draft considers a scenario, where nodes are initially configured by an installer during bootstrapping phase. If a PSK is also required to be configured in this phase, the trust between installer and system administrator is extremely important. This is not easy process. Even further, if the case is settled, there are several secure threats and vulnerabilities to be handled.

The basic idea of the method specified in this document is motivated from a lock of suitcase. Simple and default password such as '0000' or '1234' is initially setup on a lock of suitcase in selling. Owner can change the password after purchasing. In our method, similarly, initial key of a node is configured by installer during bootstrapping phase. When the node join to an existing network, the key (i.e. PSK) can be securely reconfigured.

2. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119] when they appear in ALL CAPS. These words may also appear in this document in lower case as plain English words, absent their normative meanings.

This draft uses notations and abbreviations as follows.

SBI(i)

Shorten abbreviation of a secure bootstrapping initiator i (i.e. new node required to be reconfigured)

SBR(c)

Shorten abbreviation of a secure bootstrapping respondent c (i.e. a type of controller)

SBS(s)

Shorten abbreviation of a secure bootstrapping server s (i.e. an authenticated register or authentication server)

ID_A

Denoting 32bits identifier (ID) of entity A

NID_A

Denoting network ID used for access to communication entity A (e.g. IPv4 or IPv6 address and port number).

RN_A

Denoting 128bits integer used for a secure random number generated by entity A; for example, a random number generated by SBI is referred to as RN_i.

IK_N

Denoting 128bits symmetric key pre-installed by installer or manufacturer for node N; The key is used for a partial transaction of mutual authentication and derivation of PSK (see section 4 in detail).

PSK

Shorten abbreviation of a 128bits pre-shared key derived from the IK. The PSK is a shared key between a node and authenticated register (or authentication server) in a specific service domain. A PSK can be used to derive session keys for various security protocols designed by service administrator (see [RFC4764] for example).

TS

Denoting time stamp of operation; it enables sender (TS generator) to inform timeliness and uniqueness to receiver.

SK_cs

Denoting a 128bits symmetric key shared between entity c and s.

||

Notation used to denote concatenation of data.

V

Notation used to denote a logical operator Exclusive OR.

E(M, SK)

Denoting a function to encrypt a plain text 'M' by using a symmetric key SK.

D(C, SK)

Denoting a function to decrypt a cipher text 'C' by using a symmetric key SK.

Other security related terminologies used in this document are based on [RFC4949].

3. System Architecture

Secure bootstrapping is regarded as a difficult problem in Internet of Things. This is mainly because lots of things connected to Internet are resource constrained. Especially, user interface they have is not enough for doing configurations manually by person (i.e. inadequate or even no in/out equipment such as display or keyboard).

As one of solutions, this document proposes a method which allows a node to reconfigure a symmetric key automatically upon joining to existing network.

The method of this document is based on a straightforward scenario, where resource constrained things in IoT such as sensors or actuators are generally designed and manufactured according to their own specific tasks in advance. Also, a pre-defined controller covers and communicates with his associated things according to his rolls defined in a service domain. For example, a thermostat manages and communicates several temperature sensors, humidity sensors, windows, heating controller, air conditioner, and more. This document does not assume that a system administrator trusts an installer even though he makes orders for the installer. This is because trust and responsibility of installer who buys and install devices is different from those of system administrator.

In this scenario, the following transactions MUST be done prior to the key reconfiguration.

1. System administrator makes orders including setup information required to be initially configured. These are ID and NID of controller for each of nodes, temporary key used as an IK. In particular, all nodes handled by a single installer can share the same IK similarly to the default password for all suitcases manufactured by a single company.

2. System administrator also stores the same initial information for each of nodes in authentication server (or authenticated register). Note that a controller can also perform operations of an authentication server in case of a small network.

3. Installer purchases devices and then configures the information requested by the administrator in doing installation. Some of the information for a node may be pre-configured by manufacturer.

4. When a node joins to network, it knows NID of his associated controller with which he can communicate. Also, authentication server has lists of IDs for new nodes.

5. PSK reconfiguration phase is then started.

In order to make a practical and reasonable method, the proposed method requires only a single cryptographic primitive that is AES with 128bits length of key [AES]. All cryptographic primitives cannot be installed on resource restricted devices, mainly because of limited size of flash or RAM. For this reason, CoAP also do not consider all modes of cryptographic operations in DTLS which is a regarded secure protocol for CoAP applications. In case of establishing a CoAP session using a pre-shared key mod of DTLS, implementation of cipher suite TLS_PSK_WITH_AES_128_CCM_8 specified in [RFC6655] is mandatory.

4. Process Flow

There are three message exchanges between new node SBI(i) and network (SBR(c) and SBS(s)). A controller SBR(c) MAY include functions of both SBR(c) and SBS(s) depending on the size of application domain. Mutual authentication and PSK reconfiguration procedures are shown in Figure 1.

```
-------                        ------                          ------
 SBI(i)                        SBR(c)                          SBS(s)
-------                        ------                          ------
   |                             |                                |
   |         ID_i, RN_i          |                                |
   | --------------------------->|                                |
   |                             |ID_i, ID_c, RN_i, RN_c, TS, TID |
   |                             |------------------------------->|
   |                             |                                |
   |                             |                                |
   |                             |       E(IK_i || TID,SK_cs)     |
   |                             |<-------------------------------|
   |                             |                                |
   |                             |                                |
   | ID_c, E(RN_i || RN_c, IK_i)|                                |
   |<----------------------------|                                |
   |                             |                                |
   |                             |                                |
   |       E(RN_c, PSK)          |                                |
   |---------------------------->|                                |
   |                             |                                |
   |                             |                                |
```

Figure 1 Message Exchange for PSK Reconfiguration

When a new node SBI(i) joins an existing network, he generates a random number RN_i and sends it with his identifier ID_i to SBR(s).

Upon receiving the message, SBR(c) generates a random number RN_c and a serial number used as a transaction ID (i.e. TID). Then he sends the two numbers with his ID_c, time stamp (TS) and the message received from SBI(i) to the authentication server SBS(s). TS allows SBR(s) to derive an expiration of key and verify the freshness of the arrived message. Specific period of the expiration of key (i.e. PSK) does not covered in this document.

The authentication server SBS(s) now can derive a new PSK for the node SBI(i) and replace the IK_i, which he initially stored, to the PSK, where the PSK for SBI(i) is derived as follows.

    PSK_i = E(RN_i V RN_c, IK_i)

At this moment, IK_i does not known to SBS(c). After reconfiguration of key for node SBI(i), SBS(s) encrypts the concatenation value of IK_i and ID_i with the symmetric key SK_cs which is a shared key between SBS(s) and SBR(c).

On receiving the encrypted value from SBS(s), SBR(c) can know the key IK_i thereby calculating PSK. SBR(c) encrypts the concatenation value of RN_i and RN_c with key IK_i. Then it sends the encryption value and his ID_r to SBI(i).

Finally, SBI(i) can reconfigure his PSK thereafter sending the encryption value of RN_c with the new key PSK to SBR(c) to verify himself.

5. Security Considerations

The method of this draft uses a single cryptographic primitive AES [AES]. Single cryptographic primitive implementation is rationally suited for the scenario where applications or services require a secure session (confidentiality of data) in IoT. Because small devices with low computing power and little storage are major entities. In this draft, a single primitive AES is used for secure bootstrapping (exactly PSK reconfiguration phase). Further, the PSK can be used for session key derivation and entity authentication.

As discussed in ESP-PSK [RFC4764], it goes without saying that a single cryptographic primitive may not support extensible security services such as identity protection, perfect forward secrecy and

others. However, small devices consisting of Internet of Things might not support all of security services inherently. Service developer should therefore define a scope of his service strictly and consider trade-off between capability and security.

Security analysis and evaluation of various aspects of the method remain to be done.

6. IANA Considerations

This memo includes no request to IANA

7. Acknowledgments

(TBD)

8. References

8.1. Normative References

[RFC4764] F. Bersani, H. Tschofenig, "The EAP-PSK Protocol: A Pre-Shared Key Extensible Authentication Protocol (EAP) Method", RFC 4764, January 2007.

[RFC4949] Shirey, R., "Internet Security Glossary, Version 2", RFC 4949, August 2007.

[RFC6655] McGrew, D. and D. Bailey, "AES-CCM Cipher Suites for Transport Layer Security (TLS)", RFC 6655, July 2012.

[CoAP] Shelby, Z., Hartke, K., Bormann, C., and B. Frank, "Constrained Application Protocol (CoAP)", draft-ietf-core-coap-18 (work in progress), June 2013.

[SecCons] O. Garcia-Morchon, S. Kumar, S. Keoh, R. Hummen, R. Struik, "Security Considerations in the IP-based Internet of Things", Internet draft (draft-garcia-core-security-06), September 2013.

[AES] National Institute of Standards and Technology, "Specification for the Advanced Encryption Standard (AES)", Federal Information Processing Standards (FIPS) 197, November 2001.

8.2. Informative References

[RFC2119] S. Brander, "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

Author's Addresses

Namhi Kang
Duksung Women's University
Seoul Korea
Email: kang@duksung.ac.kr
URI: http://www.duksung.ac.kr