Core Working Group
Internet Draft
Intended status: Informational
Expires:March 29, 2014

J. Zhu
M. Qi
China Mobile

Sep 29, 2013

Group Authentication
draft-zhu-core-groupauth-01

Abstract

The group communication is designed for the communication of Internet of Things. A threat is identified in [I-D.ietf-core-groupcomm] that current DTLS based approach is unicast oriented and there is no supporting on group authentication feature. Unicast oriented authentication will causing serious burden when a large number of terminal nodes will be involved inevitably. In another aspect, some terminals will own the same characteristics, such as owning same features, in the same place, working in the same time, etc. With this mechanism, all terminals can be authenticated together with little signaling and calculation at the same time. It will reduce the network burden and save time. This draft describes the security of group authentication and an group authentication implementation method for the Internet of things.

The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html

This Internet-Draft will expire on March 29, 2014.

Table of Contents

1. Introduction

With the development of Internet of Things, a large number of

terminal nodes will be involved inevitably. The unicast authentication communication from big amount terminals will merge together in the network, and causing serious burden to the server. Although IP multicast technical is introduced for group communication in [I-D.ietf-core-groupcomm], IP multicast relies on the unicast authentication at initial stage. In another aspect, some terminals will own the same characteristics, such as owning same features, in the same place, working in the same time, etc. With this mechanism, all terminals can be authenticated with little signaling and calculation at the same time. It will reduce the network burden and save time.

This draft describes the security of group authentication and an group authentication implementation method for the Internet of things.

2. Conventions used in this document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC-2119 [RFC2119].

In this document, these words will appear with that interpretation only when in ALL CAPS. Lower case uses of these words are not to be interpreted as carrying RFC-2119 significance.

2.1. Definitions

Terminals: It is such a constrained device also recognized as group node in this document which communicates with server, through direct or indirect connections, which can be seen by the server. If some constrained devices like Zigbee node only communicates with sink node and it can’t be seen by the server, such constrained device is not recognized as terminal.

Group agent: It is a device on behalf of group nodes (terminals) to make mutual authentication with network server.

Network server: It is the core network server which is responsible for authenticating the legality of terminal and provides specific services for them.

3. Problem Statement

3.1. Use cases

Nowadays the normal authentication mechanism in network is a traditional unicast authentication method between a single terminal and a single network entity. The authentication mechanism will be finished based one 1-2 round of challenge-response conversation separately. If there are many terminals, the cost for authentication will be increased.

Now for some M2M service, it may be a large amount of terminal used for an M2M service. These terminals are placed in the same location, will be used for the same purpose, and own same behavior. These terminals can be worked together as a group. In these scenarios, the existing authentication mechanism is no longer appropriate. When a large number of terminals want to access network server, huge number of authentication signaling will be generated by the unicast authentication method. What is more, it will cause network congestion and lead to DoS attack. For some terminal devices which is restricted with limited computing capability and power, the traditional unicast authentication will increase the computational burden of these terminals and drain their poor battery.

The following use cases are identified at this point:

Smart Metering: A large amount of smart power meter terminals are deployed in a block. The smart meter uploads meter report frequently through the network to smart meter server. What is more, smart meter server queries all terminals periodically to check whether the terminal is workable or not. So smart meters report at the same time, or the smart meter server need to re-configure all smart meters at the same time. In fact, there are other type of smart metering which has no agent node. This will lead to overload since each apartment needs to equip with a meter and they access network parallel at the same time and it will not be considered in this draft.

Remote Vehicle Management: IOT terminals contains GPS location reporter, remote air condition control, etc. would be installed in some special Vehicles like Taxi. It will send information such as position information, navigation, remote diagnosis, on-board communication, news and

entertainment information etc. to the network server in order to make better vehicle scheduling, vehicle monitoring and vehicle controlling. So it needs to connect to the network and make authentication at first. However, such vehicles would gather in a small place like airport, train station, etc. The frequency of connection from these terminals to server will cause overloading.

Intelligent home: various sensors equipped with communication modules are deployed in a house to monitor house conditions and make a control when necessary. These sensors collect and report house related information like the status of door open/close, indoor temperature to its owner through a network, and take actions by following the regulating instructions send by the owner.

3.2. Problem statement

In the current smart metering service use cases, a large amount of smart power meter terminals are deployed in a block. The smart meter uploads meter report frequently through the network to smart meter server. What is more, smart meter server queries all terminals periodically to check whether the terminal is workable or not. Therefore, the meter requires frequent and network communication.

In such use cases, when all the meters access network parallel at the same time, or when the server sends message to all meters, the terminals will connect to the network in a short time period (1sec ~ 1min). Assume there are 19 buildings in the block, and each building has 25 floors on average with 10 apartments in each floor. If each apartment is equipped with 1 smart power meter, then 4760 meters will be deployed in total in the block. This will cause pressure to the network.

So an agent node has been introduced to aggregate the message from these meters and then send out these meters data to the server together. After the agent is introduced, the connection between meters and servers is split into two parts: one is the connection between meters, the other is and the one between agent and server. Usually the agent is responsible for the authentication of the meters.

The server is responsible for the authentication of the agent only and gets all information about meters such as ID, data, from agent.

The current security mechanism is:

1. Each meter is authenticated with the agent. Agent will authenticate the meter one by one. After that, agent should make mutual authentication with server. Then server can confirm agent identity.

2. Meter will set up security connection with agent, and agent will also set up security connection with server. When a meter wants to send data to server. It should send the data confidential protected to agent first. Agent will decrypt the data and transfer it to server by using the security protection mechanism between agent and server.

However, this procedure has the following security problems:

1. Since all meters are authenticated by the agent and no direct authentication from server to meter. The server can get meter's ID and data only through agent. So the agent Due to the key position in the authentication, the security protection about agent is very important. Server could not authenticate meters directly. It can only rely on the agent. However, the agent would be placed in un-secure place or owned by different user rather than the server owner. If the agent is compromised or lay to server, agent can act as a middle attacker that makes fake authentication to meters and report fake ID to servers.
2. Another security problem is related with agent and server. Under this scenario, all information from meters will be transferred through agent. So agent will know all information generated by meters. However, under some scenario, agent would be owned and used by different user other than the meters' and servers' owner. So under this assumption, the agent should not get the message from meter to server. So meters should set up an secure end-to-end tunnel with server. It should request another authentication and key generation procedure in addition to authenticate with agent. This will bring complexity and overhead to the system.

4. Requirement

In order to reduce the cost and simplify a lot of overhead with the same characteristics of these groups of meter or sensor node group-based operations, it is needed to provide group authentication. For example, when smart meters perform bulk configuration information updates, it is needed to ensure that the correct identity of the user node within the group, to prevent the configuration information is wrong node receives. In addition, when smart meters report meter readings to the electricity system platform, it is also needed to be able to prove the correctness of the identity of smart meters, to prevent malicious node reporting false readings.

5. Group Authentication Solution

5.1. Introduction

Group authentication is a kind of authentication technologies that a group of users or terminals can be authenticated together at the same time. Instead of authenticating a number of terminals of a group one by one, group authentication mechanism treats these terminals in the group as a whole, and authenticates them together. Each group has a unique identifier, and an agent, which can be called as group agent, group gateway, etc.

Group authentication comprises following two phases as following:

1. The first phase is that user/terminal should be authenticated whether it belongs to a given group. This can be implemented through the proprietary authentication technology in a group, such as Zigbee or any others.

2. The second phase is that mutual authentication should be made between a given network entity, and a group agent who is responsible to delegate all terminals in the group.

   After the authentication, terminals and network entity can generate separated session keys individually if there is some demand to make individual communication between network entity and each terminal.

5.2. Detailed group scenario description

For group authentication, there is detailed network description as following. There are 5 nodes inside a given group. They are A1, A2, A3, A4, and A5 which is group agent. And the given group can be named as group A. All nodes in group A can communicate with each other. What is more, A5 is able to communicate with network entity directly. Network entity will store the group information, such as identifiers, root keys used for all nodes inside the group. Network entity is also responsible for generating group authentication vector. The scenario is shown as below.

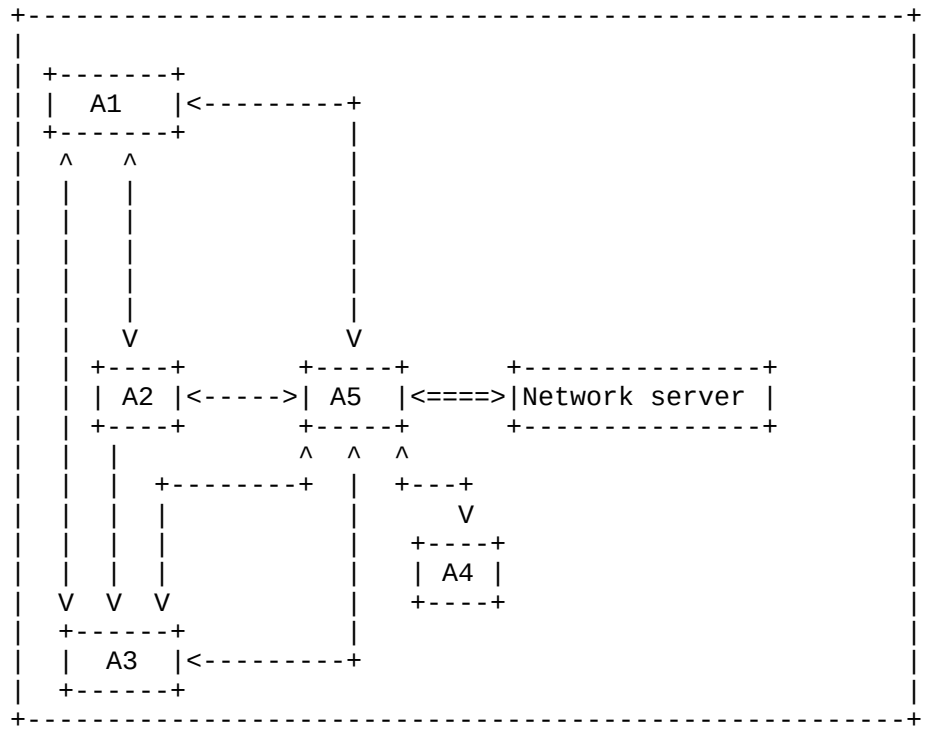

Figure 1 Group Authentication Architecture

o A5 (group agent) communicates with other nodes, i.e. A1, A2, A3, A4 by inner group protocol. All nodes should contain such models as inner group communication model, group authentication mode. Inner group communication model can be used to sending/receiving

the group authentication message. Group authentication model can be used to generate authentication vectors/response and to authenticate peers.

- Group agent will make mutual authentication with network entities. There are two kinds of network entities. Network server is responsible for mutual authentication action with group agent. And Network Server is responsible for group authentication vector generation and forwarding AV to network server. After the authentication, terminals and network entity can generate separated session keys individually if there is some demand to make individual communication between network entity and each terminals.

- Group agent who represents the whole group, communicates with network entity, and generate group session key through authentication with the network server.
- Pre-configure of the group

All the group nodes should be configured with sub key k_1, k_2, k_3, k_4, k_g, which will be used for mutual authentication in the group and separated communication.

5.3. Group scenario procedure

As mentioned above, group authentication can be divided into two phases.

In the first phase, group member, say Ai, sends authentication request to group agent at first as following.

1. Group member Ai sends message to trigger authentication at first.
2. Group agent sends authentication request to each group member.
3. Group member Ai verifies group agent at first. If success, Ai will generate session key for the communication with group agent, and sends response containing such session key back to group agent. If not success, the authentication is failed and group authentication procedure will be abort.
4. Group agent authenticates each group member Ai through the response message and record the authentication result in a mapping

table.
After the inner group authentication, all of group members are authenticated by group agent, and second phase can be performed.

5. Group agent sends message to network server to trigger the authentication outside the group.

6. Meanwhile, group agent sends authentication vector request to network server with group agent identity.

7. Network Server will generate authentication vector according to group agent identity.
8. What is more, network server should be able to recognize that is a group authentication is performed based on group agent identity. Network Server will generate session key for each group members by using pre-configured group member information and the same keying material in above step.
9. Network Server will send such authentication vector and session keys together back to network server.
10. Network server will perform mutual authentication with group agent.
11. Group agent authenticate group agent and send authentication response back to network server.
12. Network server authenticates group agent. If success, it can be considered that group agent and all group terminals is authenticated successfully.
13. Group agent will communicate with network server to choose the confidential and integrity protection algorithms.
14. After that, group agent will send keying material, selected algorithms to each group member.
15. Group member will generate session keys.

After these two phases, each terminal is authenticated with network server and generate independently session key with network server.

6. Security Considerations

TBD

7. IANA Considerations
There are no IANA considerations associated to this memo.

8. Conclusions

This memo describes the problem raised by using one-to-one authentication for huge number of Internet of Things terminals. After that, group authentication requirement is raised and a group authentication mechanism is proposed. By using the proposed group authentication mechanism, the exploited group agent can behalf of all involved group nodes to make mutual authentication with network server, but the agent can not realize the content transmitted between both of them since it is just a intermediate node to forward messages which are encrypted by specific session key between the group node and network server. The trust relationship is between group nodes and network server. After authentication, the trust relationship between group nodes and group agent are not needed.

9. Acknowledgement

Thanks very much to Bert Greevenbosch, Stefanie Gerdes, Kepeng Li for their helpful comments and significant suggestions to revise this document.

10. References
10.1. Normative References
10.2. Informative References

Authors' Addresses
Judy Zhu
China Mobile
Unit 2, 32 Xuanwumenxi Ave,
Xicheng District,
Beijing 100053, China
Email: zhuhongru@chinamobile.com

Minpeng Qi
China Mobile
Unit 2, 32 Xuanwumenxi Ave,
Xicheng District,
Beijing 100053, China
Email: qiminpeng@chinamobile.com