Applying Generic Bootstrapping Architecture for use with Constrained Devices

Jouni Korhonen
jouni.nospam@gmail.com

***Abstract*—This paper discusses the possible use and required modifications of the 3GPP[1] Generic Authentication Architecture as a security framework for constrained devices within Internet of Things deployments. The deployment architecture in this paper assumes a cellular operator running the requires backend infrastructure and also providing the wide area network access over a cellular wireless broadband. The constrained devices are not assumed to have means for accessing cellular networks.**

I. Introduction

Internet of Things (IoT) and Machine-to-Machine (M2M) communication has recently gained considerable momentum, and it is predicted the market in those areas will be counted in tens of billions of new connected devices in coming few years. Various telecommunication standardization organizations have also worked on their view of the desired architecture, use cases and actual protocol set. For example, ETSI[2] [9] has its functional architecture for M2M communications and 3GPP [5] has its own view of 3GPP architecture system improvements for Machine-Type Communications (MTC). They both share the complex system architecture and interface design approach.

At the same time, IETF[3] has worked on a protocol suite [10], [15], [7] that is targeted for IoT use cases and deployments but does not as such require huge well-defined underlying architecture foundation for developers to get started with their applications. Furthermore, the recent developments on residential home networks are looking forward re-enabling end-to-end communication along with the introduction of IPv6, and have the proliferation of networking technologies in an increasingly broad range and number of devices. These residential networks may also have nontrivial segmentation based, for example, on services different segments are supposed to provide. One promising networking service is offering network access for various low-power constrained IoT devices.

Another growing trend is the replacement of a cable using a modern, technically IPv6 ready, cellular radio technology such as LTE. For example, the throughput and reliability of LTE is at least at the same level, yet often exceeding, a typical consumer fixed digital subscriber line (DSL) access. Cellular access in home gateways for residential networks introduces a cellular operator as a wireless broadband Internet Service Provider (ISP). Also, it is expected that a cellular operator would like to have a stake on IoT deployments within residential networks beyond the home gateway, in a way or other. However, these devices cannot be expected to be cellular access capable or contain smart cards such as Universal Integrated Circuit Card (UICC) for credentials authenticating against cellular operator network using e.g. UMTS AKA algorithm [1].

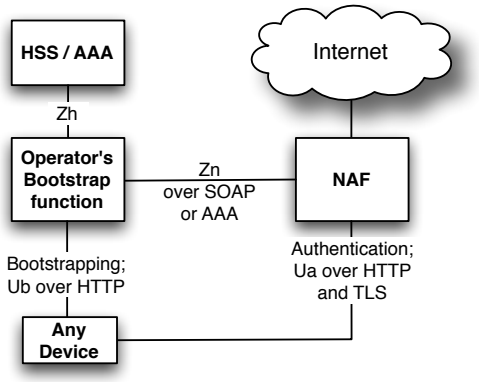

Figure 1: Overall Generic Bootstrapping Architecture

3GPP has developed a versatile Generic Bootstrapping Architecture (GBA) [3] that allows bootstrapping a shared secret between an end user/device and the network side bootstrapping function (BSF). This allows then authenticating the end user/device against a third party service provider network application server (NAF) in a way that there is no need for user enrollment phase nor secure deployment of keys. The service provider retrieves the required earlier bootstrapped key material from the bootstrapping function on need basis and the actual authentication between the end user/device and the application function uses, for example, shared key-based mutual authentication using Pre-Shared Key Ciphersuites for Transport Layer Security (PSK-TLS) [8], [2]. The interface between the service provider and operator's bootstrapping function is either based on AAA or SOAP protocols [4]. The overall GBA architecture is illustrated in Figure 1. For example, ETSI M2M functional architecture already allows the use of GBA.

One problem area with IoT deployments has been the bootstrapping of the security. This also involves the authentication to the network and also the authorization for a specific set of services. These are the issues this paper tries to address and discusses a solution proposal that also allows a cellular

[1]The 3rd Generation Partnership Project (3GPP)
[2]The European Telecommunications Standards Institute (ETSI)
[3]Internet Engineering Task Force

operator be part of the general IoT service offering and making use of its existing assets on the network side. One obvious solution would be utilizing the GBA and adjust the general operation of the GBA to fit constrained devices in *IoT spirit*. This paper proposes a solution where the GBA is applied to residential networks and for constrained devices. Furthermore, we look into how to fit the *modified GBA* into residential network deployments in a way it does not restrict the home gateways or the end user too much. We also discuss necessary mappings of GBA interfaces to CoAP protocol [15] and use of PSK-TLS over Datagram Transport Layer Security (DTLS) [13]. Lastly, we discuss issues related to the provisioning of credentials for constrained devices and possible ways to avoid GBA style bootstrapping of the constrained devices.

II. Light Weight GBA for Constrained Devices

A. Proposed solution overview

We developed our proposed *light weight GBA*-based solution based on the following goals and assumptions:

- Reuse the existing GBA as much as possible.
- Do not require the constrained end devices to have any smart card for authentication but if such exists it should be usable without any changes outside the end device.
- Do not require the use of certificates and public key cryptography in general.
- Try to avoid bootstrapping between the end device and operator's BSF; rather if devices can be pre-provisioned before distribution that should be allowed.
- The end used should be able to change the residential network service provider and/or gateway vendor without affecting the existing *light weight GBA* deployment.
- Promote the use of TLS with pre-shared keys for mutual authentication between the end device and the gateway device in the residential network.
- Allow derivation of further key material for enabling secure group communication within the deployment under the management of the same gateway device.
- The (home) gateway in the residential network share a trust relationship with the operator hosting the BSF.

Our *light weight GBA* architecture is shown in Figure 2. Assuming the constrained device (IoT device) is already bootstrapped with proper keys and related information, it can initiate a mutual authentication procedure towards the (home) gateway device (i.e. NAF in GBA). During the authentication process the gateway device contacts the operator's bootstrapping function over an existing SOAP-based interface [4] that provides the gateway with earlier bootstrapped key material and related security profile, which can then be used for mutual authentication and deriving further application keys between the end device and the gateway. We will discuss the bootstrapping and required information further in Section II-B. If the authentication succeeds the gateway knows the end device is legitimate and can access the resources within the residential network. For example, the (home) gateway device can offer resource directory and resource collection services [14] among others.

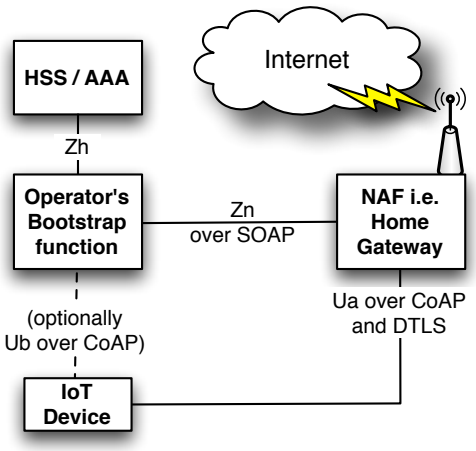

Figure 2: GBA modified for IoT within residential networks using cellular WAN access

Originally the bootstrapping between the end device and the bootstrapping function (BSF) used HTTP-Digest-AKA [12]. However, that depends on the presence of UICC (usually) and uses HTTP. In our solution, we try to avoid bootstrapping and the dependency on a physical UICC smart card when possible. There has been recent efforts to make use of other credentials than those of AKA for bootstrapping the security [6]. Unfortunately [6] requires the use of certificates for the server authentication, which is not we actually want. In our solution we propose a simplified *CoAP-Digest-AKA*, when online bootstrapping is needed.

GBA already defines the use of PSK-TLS over HTTP as one of the security profiles for the mutual authentication between the end device and the NAF. Our solution makes use of this and proposes using *PSK-TLS over DTLS over CoAP* [15]. The shared key used as a master key to generate TLS session keys is the shared key bootstrapped between the (constrained) end device and the bootstrapping function. The mutual authentication procedure according to the GBA requires the end device to name the NAF using a NAF_ID, which essentially is a concatenation of the fully qualified domain name (FQDN) of the NAF and the security protocol identifier between the end device and the NAF. The discovery of the NAF_ID is one of the open issues in our proposed solution and it will be discussed further in Section III.

B. Provisioning devices and bootstrapping the security

The original GBA assumed every end device is equipped with an UICC smart card that both shares a permanent key with the network and is able to run UMTS AKA algorithm [1] in its Universal Subscriber Identity Module (USIM) module. From the provisioning point of view, we basically have the following three choices:

1) If the end device has an UICC, then it is able to run the AKA algorithm at any time, perform the bootstrapping against the network and generate the required keys and

values for the use of subsequent authentication against the NAF. Each end device would need their own UICC.
2) Should the physical UICC be avoided, the (constrained) end devices could be equipped with a *soft-SIM*, which is functionally equivalent but implemented using software in the device and some non-volatile memory. The "easier" upgrading capability of *soft-SIM*s is both a virtue and a threat, at least from operators point of view.
3) Yet another possibility is to pre-provision required keys and identifiers to the (constrained) end devices' non-volatile memory (done by the operator providing the BSF) before distributing them and using a ridiculously long key life times. Albeit not the cleanest and the most secure solution but definitely the most light weight. This approach does not either require online bootstrapping between the end device and the operator's BSF.

The proposed *light weight GBA* would prefer using the provisioning alternatives 2) or 3). Furthermore, the (constrained) end device needs the following information and keys in order to authenticate against the gateway device (NAF) and to derive a gateway specific key (Ks_(ext_)NAF key):

- Ks key derived from AKA algorithm's CK and IK keys.
- B-TIB, a transaction identifier naming the keys.
- RAND, which is produced by the AKA algorithm.
- IMPI, which is the identity of the device (in this case).

Additionally, the end device has to know the FQDN of the gateway device (i.e. NAF) and because we do not want to depend on the presence of the UICC, the key derivation function uses keys that are stored in the device itself (i.e. the *"gba-me"* mode for deriving the keys).

III. Challenges and Future Work

The *light weight GBA* proposed in this paper has few technical challenges that need further work. Note that we are blindly bypassing the initial constrained device distribution issues, since the proposed solution is based on the assumption the (cellular) operator is doing that willingly and purposely. Also, we do not go further evaluating whether the proposed solution makes commercial sense.

The end device needs to know the FQDN of the gateway i.e. NAF device in order to be able to generate a proper NAF_ID. That cannot be statically provisioned or even configured if we ever want to allow users in the residential network to e.g., swap the gateway device, change the actual network service provider or host different virtual services in the gateway device.

One possible way to solve the discovery of the NAF FQDN would be defining an algorithm to generate one based on the information received during the host configuration step. A naive solution could be forming the NAF FQDN out of the link-local address of the NAF device and one of the dynamically learned domain names of DNS search list. There are obviously other approaches as well, but such solution could be easily piggy-backed on top of IPv6 stateless address autoconfiguration [11].

As already discussed earlier, there is a need to specify the use of CoAP with PSK-TLS and specifically define the procedures for *CoAP-Digest-AKA*. Finally, it would make sense to look for other bootstrapping algorithms/protocols that both do not depend on mechanisms or assets specific to cellular operators these days and still are considered light weight (e.g. do not depend on certificates or public key cryptography).

IV. Conclusions

This paper proposed and discussed the use of *light weight Generic Bootstrapping Architecture* for constrained devices. We briefly showed few implementation choices how to achieve a solution, where a cellular operator would have visibility and a role as a trusted party for bootstrapping the security within the residential network IoT deployments. We also identified several shortcomings and challenges that still need further work.

References

[1] 3GPP. 3G security; Security architecture, http://www.3gpp.org/ftp/Specs/html-info/33102, December 2011.
[2] 3GPP. Generic Authentication Architecture (GAA); Access to network application functions using Hypertext Transfer Protocol over Transport Layer Security (HTTPS), http://www.3gpp.org/ftp/Specs/html-info/33222.htm, Decemberl 2011.
[3] 3GPP. Generic Authentication Architecture (GAA); Generic Bootstrapping Architecture (GBA) , http://www.3gpp.org/ftp/Specs/html-info/33220.htm, December 2011.
[4] 3GPP. Generic Authentication Architecture (GAA); Zh and Zn Interfaces based on the Diameter protocol, http://www.3gpp.org/ftp/Specs/html-info/29109.htm, April 2011.
[5] 3GPP. System improvements for Machine-Type Communications (MTC), http://www.3gpp.org/ftp/Specs/html-info/23888.htm, December 2011.
[6] 3GPP. GBA extension for re-use of SIP Digest credentials, CR S3-120236, February 2012.
[7] A. Brandt, J. Buron, and G. Porcu. Home Automation Routing Requirements in Low-Power and Lossy Networks. RFC 5826, IETF, April 2010.
[8] P. Eronen and H. Tschofenig. Pre-Shared Key Ciphersuites for Transport Layer Security (TLS). RFC 4279, IETF, December 2005.
[9] ETSI. Machine-to-Machine communications (M2M); Functional architecture. TS 102 690 v1.1.1, ETSI, Octoberr 2011.
[10] J. Hui and P. Thubert. Compression Format for IPv6 Datagrams over IEEE 802.15.4-Based Networks. RFC 6282, IETF, September 2011.
[11] J. Jeong, S. Park, L. Beloeil, and S. Madanapalli. IPv6 Router Advertisement Options for DNS Configuration. RFC 6106, IETF, November 2010.
[12] A. Niemi, A. Arkko, and A. Torvinen. Hypertext Transfer Protocol (HTTP) Digest Authentication Using Authentication and Key Agreement (AKA). RFC 3310, IETF, September 2002.
[13] E. Rescorla and N. Modadugu. Datagram Transport Layer Security Version 1.2. RFC 6347, IETF, January 2012.
[14] Z. Shelby. CoRE Link Format. Internet Draft draft-ietf-core-link-format-11, IETF, January 2012. Work in progress.
[15] Z. Shelby, K. Hartke, C. Bormann, and B. Frank. Constrained Application Protocol (CoAP). Internet Draft draft-ietf-core-coap-08, IETF, November 2011. Work in progress.