ERICSSON WHITE PAPER
284 23-3284 Uen | February 2016

BOOTSTRAPPING SECURITY

THE KEY TO INTERNET OF THINGS ACCESS AUTHENTICATION AND DATA INTEGRITY

Security will be a major challenge as billions of devices join the Internet of Things, and different technologies will compete to provide appropriate solutions. Managing authentication on a large scale is a challenge already successfully met by the telecommunications industry in the form of mutual authentication with secret credentials in the SIM. This technology can now be extended to provide equally strong authentication and data integrity for the Internet of Things.

INTRODUCTION

The rapid deployment of connected devices and huge increase in data volumes are becoming a major concern for operators – and an opportunity at the same time. In the near future, billions of Internet of Things (IoT) devices will be connected to new IoT services [1]. However, analysts indicate that security issues could be a significant inhibitor to the deployment of many of these services [2].

Secure communication in IoT-type systems currently requires many levels of configuration and/or application-level proprietary algorithms, which discourages users from implementing protection and often encourages functionality to be prioritized over security. The lack of secured links exposes data to attacks and theft, and fraudsters and hackers are already beginning to show increasing interest in this area [3] [4].

This paper shows how Generic Bootstrapping Architecture (GBA) technology [6], based on the Authentication and Key Agreement (AKA) protocol used in network access authentication, can provide device authentication and support communication security at the transport layer.

AUTHENTICATION AND DATA INTEGRITY IN THE IOT

Security encompasses a multitude of aspects ranging from the protection of resources, information and identities to resisting both physical and network-based attacks. In an IoT context, one aspect of security is safeguarding the integrity and confidentiality of machine-to-machine (M2M) data and the authentication of each device placed within an M2M network. End-to-end data confidentiality can be achieved through encryption, which can be supplied by existing security protocols, such as Transport Layer Security (TLS) or Internet Protocol Security.

However, there is an issue with these protocols when choosing the keys and user certificates needed for authentication and/or encryption. As is the case with TLS, transport protocols often depend on the provisioning of credentials signed and managed by third parties. Here, the management is usually arranged in the form of public key infrastructures. This means there is a particular need to ensure that certificates are securely provisioned, updated and revoked, whether they are issued by a network operator or a third party. Alternatively, some IoT devices, proprietary hardware or firmware have hard-coded credentials embedded in the device.

All of these methods can be used to achieve a good level of security. However, the large infrastructure required to manage certificates makes them less attractive for large deployments over a long period. As an alternative to certificates, a manufacturer can produce devices with hard-coded credentials. However, these credentials would be mainly used for identification and not for data integrity. In this case, the inherent difficulties associated with future updates of the hard-coded credentials make the whole set of devices vulnerable to attacks and reverse engineering to obtain these credentials. Updating these types of hard-coded device credentials after a breach can be very cumbersome and expensive.

One way to solve these issues is to leverage the existing 3GPP network authentication framework that is an inherent part of cellular networks. Cellular networks use strong authentication and communication security, where the Universal Integrated Circuit Card (UICC) acts as the secure storage point of the secret keys on the device side. Building on this framework, GBA technology provides the means to implement AKA with GBA generating time-limited session keys during the SIM authentication [6]. The generated keys can be used for creating, for example, a TLS-based protected communication channel. Furthermore, GBA can also be used over non-cellular connectivity options like Wi-Fi. For capillary networks [5], GBA can also cover non-3GPP devices; in other words, those devices that do not have a UICC or cellular network access.

In the following sections, this paper will detail the GBA technology and demonstrate how it can provide mutual authentication and data integrity in a capillary network setting at very little cost for the mobile network operator (MNO).

AN INTRODUCTION TO GBA

GBA is a key bootstrap method standardized by the 3GPP [6]. The protocol enables the creation of service or application keys through authentication using 3GPP subscription credentials. The credentials are typically stored on a SIM card, which runs on an UICC. Alternatively, they can be provided as remotely managed credentials [7] stored and managed on an embedded UICC (eUICC) such as the GSMA-specified eSIM [8] [9] [10].

GBA consists of two main components in the network: the Bootstrapping Server Function (BSF) and the Network Application Function (NAF). The BSF authenticates the subscriber with the 3GPP subscription using the 3GPP AKA protocol. As the SIM card is in the device, the device can be regarded as being authenticated. The mutual authentication between the SIM and the network results in the generation of a bootstrapping session key (Ks) – see Figure 1 – at both the device end and the BSF end. The BSF then provides an identifier for the Ks – a Bootstrapping Transaction Identifier (B-TID) – to the device. The device uses the Ks as a root key for generating application-specific session keys for GBA-enabled services.

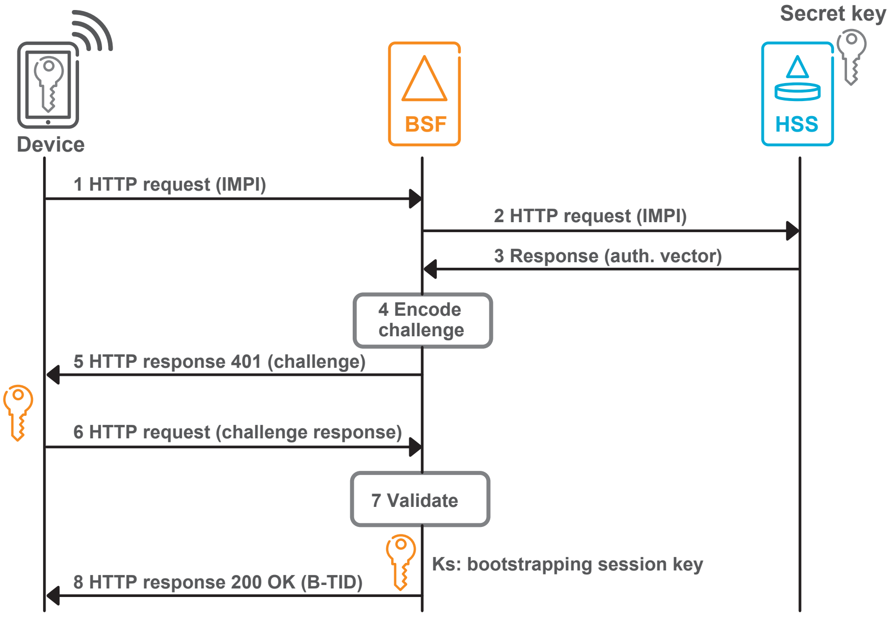

Secret key
Device
BSF
HSS
1 HTTP request (IMPI)
2 HTTP request (IMPI)
3 Response (auth. vector)
4 Encode challenge
5 HTTP response 401 (challenge)
6 HTTP request (challenge response)
7 Validate
Ks: bootstrapping session key
8 HTTP response 200 OK (B-TID)

Figure 1: GBA bootstrapping procedure.

The NAF forms the authentication function of a web service, and communicates with the BSF to get a NAF-specific shared key material (KsNAF) for the device being authenticated. There needs to be a trust relationship between the NAF and the BSF or operator, as well as a secure channel for communicating the KsNAF. When the NAF gets the KsNAF from the BSF, the device and NAF can use it for authentication and to establish a secure communication channel, as shown in Figure 2. The bootstrapping depicted in Figure 1 can either take place before the flow shown in Figure 2, or it can be run at step 3 in Figure 2.

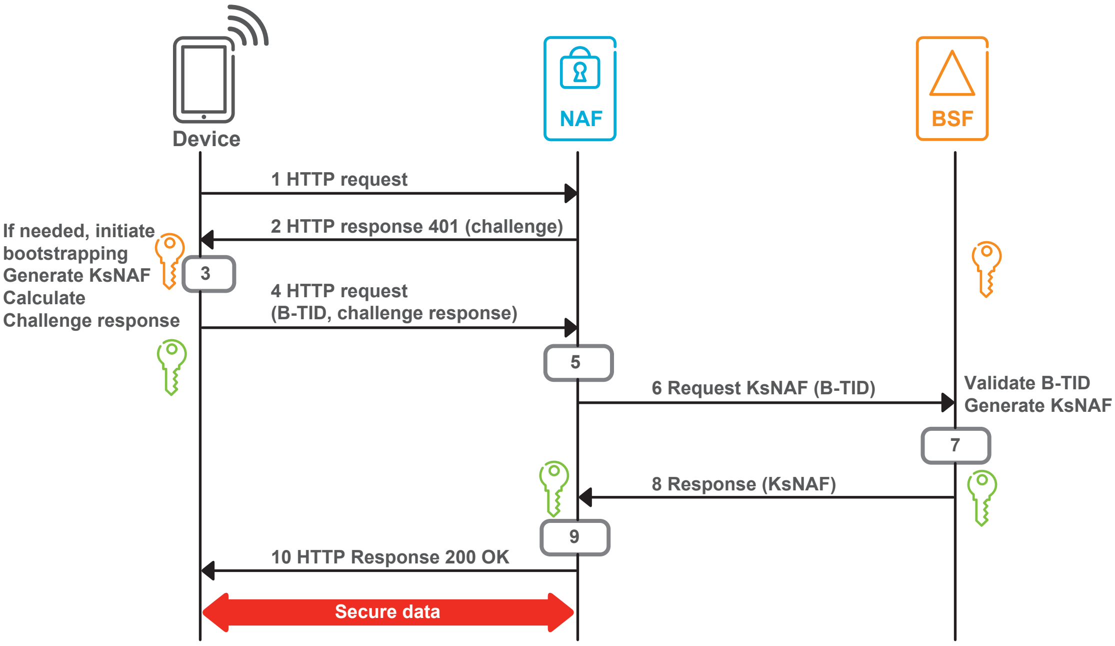


Figure 2: Bootstrapping usage procedure.

SECURE CONNECTION OF 3GPP AND NON-3GPP DEVICES

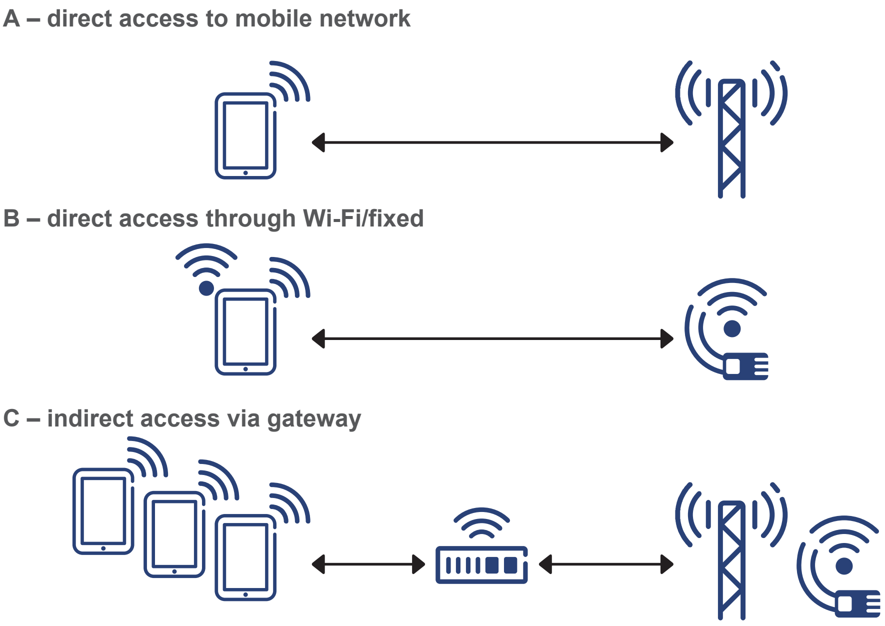
Strong transport layer communication security generally relies on cryptographic algorithms and authentication. This often means server-side authentication with certificates and client authentication through user credentials, such as passwords or hard tokens, as well as the use of a security protocol for data protection.

For autonomous devices, which often operate in public areas, the user credentials need to be protected against both physical and network-side attacks. Protection against attacks can be achieved by having secure storage and a trusted execution environment in the device. However, if the security configuration of the devices becomes too complex – for example, through the use of multiple types of user credentials, software for VPNs, specific network configurations through virtual local area networks and private Access Point Names, or by requiring per-device manual configuration – security can become costly and difficult to manage. This will become even more apparent as the number of IoT devices grows into the billions.

The security requirements for IoT devices vary depending on where and for what purpose they are used. In some cases, device authentication or integrity protection of data is enough, while other scenarios have more demanding security requirements. In those cases, strong security solutions suitable for constrained devices are required. One key element of security is a strong and secure identity for an entity. In addition, some IoT devices may not have the capacity to handle certain security-related tasks, such as certificate validation, or may consume an unacceptable amount of resources.

The solution described in this paper leverages on GBA technology, which generates time-limited keys during the GBA bootstrapping procedure. After GBA-based authentication to the service/NAF, the keys can be used for setting up any type of secure communication, such as a pre-shared key-based TLS (TLS-PSK) session, between the client and the service. The service-specific pre-shared keys are derived from the bootstrapped Ks.

GBA can be used in all three scenarios depicted in Figure 3.



A – direct access to mobile network

B – direct access through Wi-Fi/fixed

C – indirect access via gateway

Figure 3: Access methods.

Devices equipped with an (e)UICC that utilize 2G, 3G or even LTE – in other words, 3GPP connected devices – are ideal for implementing GBA. Devices with 3GPP credentials, but no 3GPP radio, are also supported by standard GBA. However, the majority of IoT devices are not expected to use 3GPP access technologies and will not have 3GPP credentials. Instead, they will be constrained devices unaware of 3GPP networks and will be connected to the internet or an enterprise intranet using one of the following modes:

- directly – for example, by using Wi-Fi or a wired connection such as Ethernet
- indirectly through a gateway (GW), often using a short-range, low-power radio solution such as 6LoWPAN or ZigBee.

In this paper, these devices are referred to as non-3GPP connected devices. Devices belonging to the first category – directly connected devices – cannot benefit from GBA-based security, as they have no 3GPP credentials available. Devices connected indirectly through a GW are often served by a GW that utilizes 3GPP connectivity. These non-3GPP devices could benefit from the 3GPP credentials of the GW, and be provided with GBA-based security by the GW. Current efforts to evolve GBA involve technologies that delegate the GBA session credentials to the non-3GPP device. This delegation allows a constrained device to accomplish end-to-end secure sessions and to benefit from strong GBA authentication.

CAPILLARY NETWORKS AND CONSTRAINED DEVICES

Capillary networks [5] are one building block in the IoT. These networks consist of one or more devices; often sensors, actuators or other kinds of constrained devices that are connected to the public network through a capillary network gateway (CGW).

In the capillary network, the devices can be connected using various wired or wireless access methods, including IEEE 802.15.4, Bluetooth Low Energy, wireless LAN and Ethernet. In a typical case, the CGW interconnects the capillary network with the mobile network using 3GPP access – for example, 2G, 3G or LTE.

The features that can be achieved through GBA, such as strong identification and authentication, and the fact that no user interaction is required, are desirable in M2M devices that usually operate autonomously. The next section looks at how GBA can accomplish authentication and secure communication for all devices in the capillary network.

AUTHENTICATION AND SECURE COMMUNICATION WITH GBA IN CAPILLARY NETWORKS

The basic configuration and connectivity for a capillary network is shown in Figure 4. The CGW aggregates the constrained devices and is configured to perform network authentication. From the enterprise side, the constrained devices can be seen as resources of the authenticated CGW. The interfaces between the CGW and the MNO and enterprise (Ua and Ub in Figure 4) are defined in the 3GPP standard TS 33.220 [6]. In the future, the Ua interface, which is based on HTTP, could be expanded to other protocols like Constrained Application Protocol (CoAP) [11]. The third interface shown in the figure is the Zn interface between the enterprise software and the MNO, which is also defined in [6].

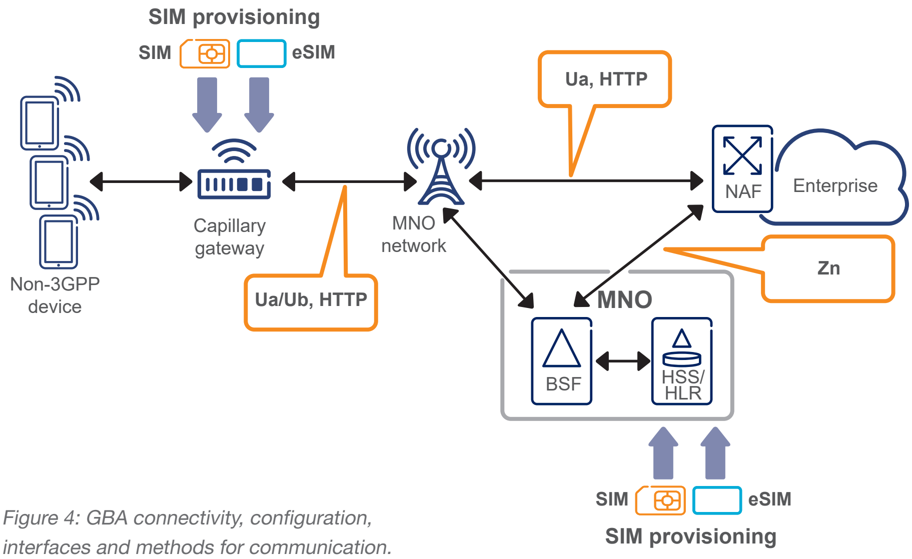


Figure 4: GBA connectivity, configuration, interfaces and methods for communication.

The procedures for GBA-based mutual authentication and data protection for capillary networks are as follows:

If not already bootstrapped, the CGW can either decide to do the bootstrapping with the BSF proactively, or it is triggered by the response from the NAF to an unauthorized request from a non-3GPP device. This is shown in Figure 5. The NAF response to an unauthorized request is an HTTP 401 message, where the realm parameter has the prefix 3GPP-bootstrapping@. This prefix indicates that the service/NAF supports GBA-based authentication.

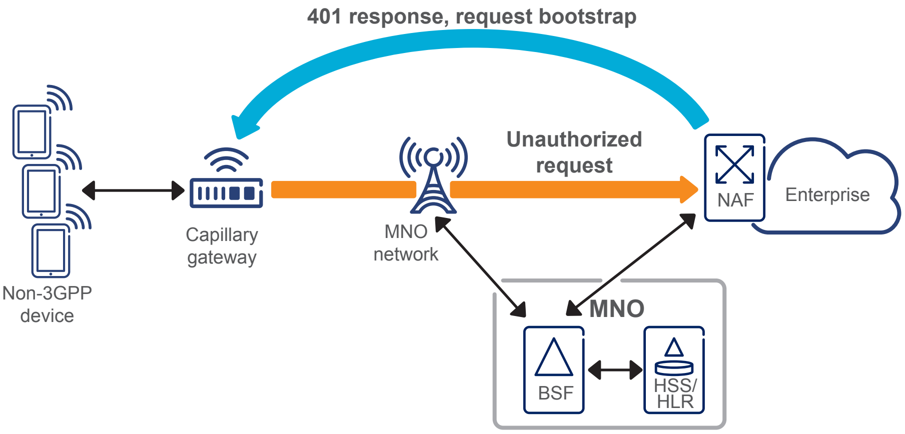


Figure 5: Connection initialization.

During the bootstrapping phase, when the CGW and the network authenticate each other, the GBA client in the CGW and the BSF both generate the Ks. This is shown in Figure 6. This key is never exposed by the GBA client on the CGW or transferred outside the BSF. The Ks is time-limited; it can be configured to expire after a predetermined time. Likewise, the KsNAF derived from the bootstrap session key will have a lifetime that does not exceed that of the Ks.

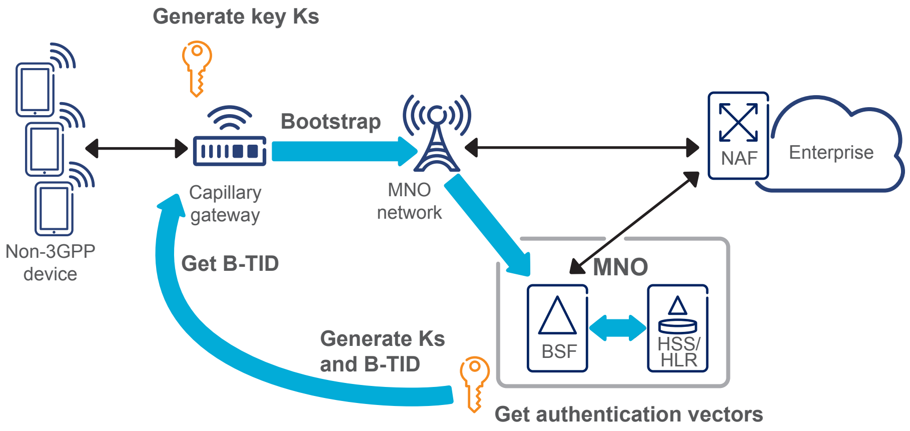


Figure 6: GBA on CGW – bootstrapping.

To initiate a GBA-secured session with a service, any application that wants to use GBA on the CGW asks the GBA client for service-specific credentials: the B-TID and the KsNAF. The latter is a key derived from the Ks.

The application uses the B-TID and the derived Ks for answering the HTTP 401 authentication challenge and authenticating itself to the service/NAF. This B-TID is used by the NAF to retrieve the corresponding KsNAF from the BSF.

The BSF identifies the bootstrapping context based on the B-TID and generates the KsNAF from the Ks. The NAF can now validate the authentication response sent by the application.

In addition to using the KsNAF for authentication, the CGW/application and NAF can use it for securing communication – for example, by establishing a TLS session based on KsNAF.

Figure 7 shows how a CGW uses the KsNAF to authenticate and possibly protect the integrity of an initial message sent to the NAF using HTTP Digest. When receiving the protected message, the NAF obtains the corresponding KsNAF from the BSF and authenticates the message. Alternatively, as shown in Figure 8, the CGW and NAF can use the KsNAF as a shared secret for TLS, and a TLS-PSK tunnel is negotiated before the actual data transfer.

In both these examples, hop-by-hop security is delivered with the CGW bearing the responsibility for GBA and using it for the security between the CGW and the NAF. The security between the constrained device and the CGW would rely on some capillary network access security and associated credentials. This "first-hop" security segment, between the CGW and the end device, is the next challenge in the security paradigm, and it is addressed in the following section of this paper.

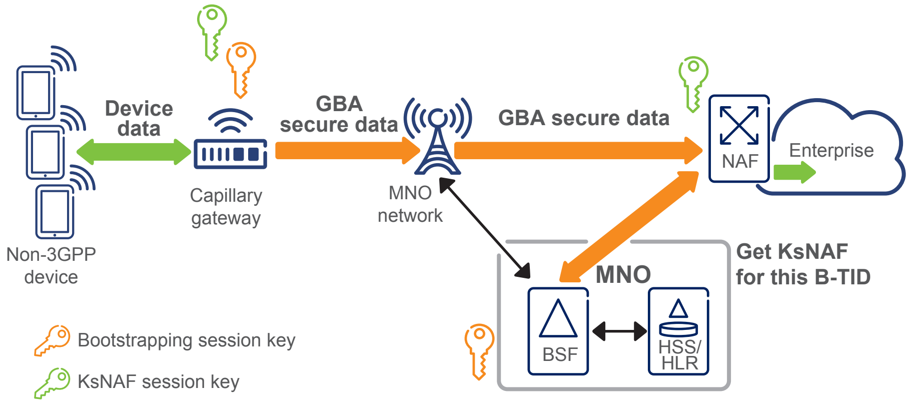


Figure 7: GBA-secured data transfer through HTTP Digest.

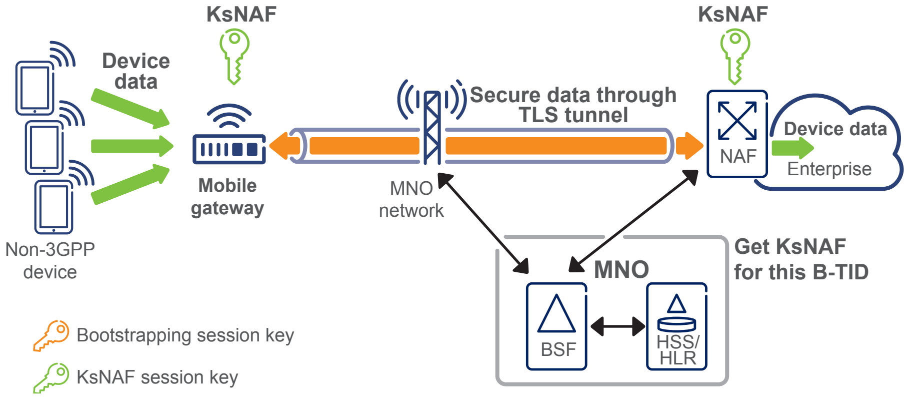


Figure 8: GBA-secured data transfer through TLS-PSK.

DELEGATED GBA IN CAPILLARY NETWORKS

As previously mentioned, one of the biggest challenges is to extend the authentication and data protection between the NAF and the CGW all the way to the end device in the capillary network. An extension to GBA, called Delegated GBA, can be used to create capillary-device-specific GBA session keys by utilizing a device ID – provisioned in the end device – in the derivation of session keys at the CGW. These session keys are an extension of the already established Ks and KsNAFs described earlier in this paper.

As depicted in Figure 9, after the end device and the CGW complete the Delegated GBA method, the end device has a KsNAF key, which is a function of the original bootstrapping Ks of the CGW, the device ID and the intended NAF fully qualified domain name. This new key is now shared between the NAF and the end device for authentication and data protection. The capability of a constrained device and CGW to accomplish the delegation of the key depends on the protocols available within the capillary network.

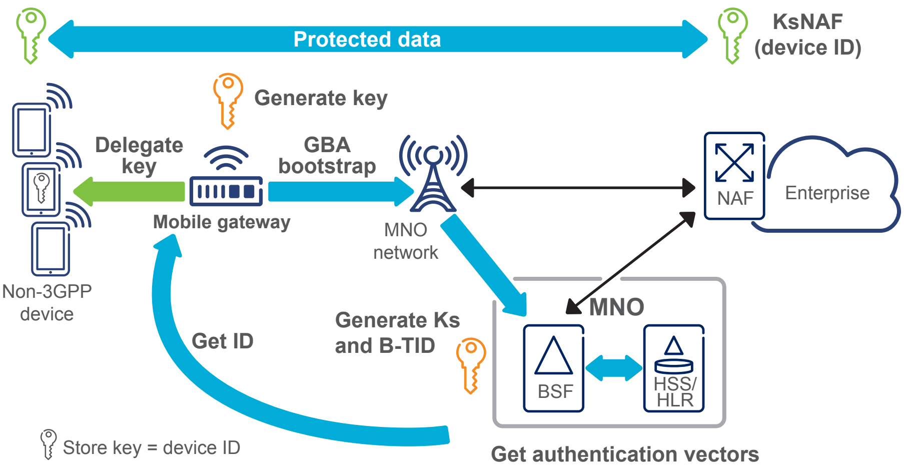


Figure 9: Delegated GBA.

CONCLUSION

With billions of devices joining the IoT, security will be a major challenge, and different technologies will compete to provide the appropriate solutions. Managing large-scale authentication was a challenge that the telecommunications industry successfully met many years ago in the form of mutual authentication with secret credentials in the SIM. This technology can now be expanded to provide equally strong authentication and data integrity in the IoT.

It is possible to create a secure tunnel between the CGW – and in some cases the capillary device with Delegated GBA – and the enterprise service, based on time-limited authentication keys, with a strong integrity- and confidentiality-protected channel for data. Its security is based on credentials on an eUICC, implying that the device in question has a 3GPP subscription. The eUICC acts as a safe element for the secure storage and use of these credentials.

Unlike alternative security approaches that require additional recurring costs and complex certificate maintenance, the GBA security framework is self-maintainable, as it reuses the credentials in the eUICC.

GBA leverages the existing identity and authentication infrastructure deployed by mobile operators. The 3GPP credentials provide a strong identity, and the authentication framework can be used with this identity. Furthermore, the solution scales effortlessly to billions of IoT devices, as there is no additional cost involved in obtaining data protection for 3GPP-enabled devices and for non-3GPP-enabled devices in combination with CGWs.

REFERENCES

[1] Beecham Research, “1 Billion Cellular M2M Connections by 2020”, February 2015, available at: http://www.beechamresearch.com/news.aspx?id=1088

[2] Gigaom, “The internet of things needs a new security model. Which one will win?”, January 2014, available at: https://gigaom.com/2014/01/22/the-internet-of-things-needs-a-new-security-model-which-one-will-win/

[3] V3.co.uk, “CES 2015: FTC warns of Internet of Things security risks”, January 2015, available at: http://www.v3.co.uk/v3-uk/news/2389013/ces-2015-ftc-warns-of-internet-of-things-security-risks

[4] The Economist, “Hacking the planet”, July 2015, available at: http://www.economist.com/news/leaders/21657811-internet-things-coming-now-time-deal-its-security-flaws-hacking

[5] Ericsson Review, “Capillary networks – a smart way to get things connected”, September 2014, available at: http://www.ericsson.com/res/thecompany/docs/publications/ericsson_review/2014/er-capillary-networks.pdf

[6] 3GPP, “3GPP Specification detail; 3GPP TS 33.220 – Generic Authentication Architecture (GAA); Generic Bootstrapping Architecture (GBA),” accessed February 2016, available at: http://www.3gpp.org/DynaReport/33220.htm

[7] 3GPP, “3GPP Specification detail; 3GPP TR 33.812 – Feasibility study on the security aspects of remote provisioning and change of subscription for Machine to Machine (M2M) equipment”, accessed February 2016, available at: http://www.3gpp.org/DynaReport/33812.htm

[8] ETSI, “ETSI TS 103 383 V12.5.0; Smart Cards; Embedded UICC; Requirements Specification (Release 12)”, August 2014, available at: https://www.etsi.org/deliver/etsi_ts/103300_103399/103383/12.05.00_60/ts_103383v120500p.pdf

[9] ETSI, Work Programme, “Smart Cards; Embedded UICC; Physical, Logical, and Electrical Characteristics; (Release 12)”, June 2013, available at: https://portal.etsi.org/webapp/WorkProgram/Report_WorkItem.asp?WKI_ID=42517

[10] GSMA, “Embedded SIM Remote Provisioning Architecture”, December 2013, available at: http://www.gsma.com/connectedliving/wp-content/uploads/2014/01/1.-GSMA-Embedded-SIM-Remote-Provisioning-Architecture-Version-1.1.pdf

[11] IETF, “The Constrained Application Protocol (CoAP)”, June 2014, available at: https://tools.ietf.org/html/rfc7252

GLOSSARY

AKA	Authentication and Key Agreement
B-TID	Bootstrapping Transaction Identifier
BSF	Bootstrapping Server Function
CGW	capillary network gateway
CoAP	Constrained Application Protocol
eUICC	embedded Universal Integrated Circuit Card
GBA	Generic Bootstrapping Architecture
GW	gateway
HLR	home location register
HSS	Home Subscriber Server
IMPI	IP multimedia private identity
IoT	Internet of Things
Ks	Bootstrapping session key
KsNAF	NAF-specific shared key material
M2M	machine-to-machine
MNO	mobile network operator
NAF	Network Application Function
TLS	Transport Layer Security
TLS-PSK	pre-shared key-based TLS
UICC	Universal Integrated Circuit Card