Virtual ITS European Congress, 9-10 November 2020

Paper number ITS-TP2149

A Security Strategy for the LinkBeyond Mobile MaaS Application

José Simão[a,b], Nuno Cruz[a,c], André Costa[d], Manuel Relvas[d]

[a] FIT - Future Internet Technologies, ISEL - Instituto Superior de Engenharia de Lisboa, IPL – Instituto Politécnico de Lisboa

[b] INESC-ID Lisboa

[c] LASIGE, Faculdade de Ciências, Universidade de Lisboa

[d] A-to-Be Mobility Technology, SA

Abstract

The ability to use/optimize current infrastructures determines the success of the seamless mobility experience. Paradigms like MaaS (Mobility as a Service) emerge based on a holistic view of people's mobility needs. A-to-Be's LinkBeyond Mobile multi-service system, is made of a mobile application integrating transport operators and mobility players and a wireless device for access control, supported by a multi-operator Back-End infrastructure. Different kinds of mobility services (e.g. bus) and utility services (e.g. parking) have different connectivity requirements. To ensure universality, the wireless device and mobile application use Bluetooth for communication. Because using native Bluetooth security features would hinder user experiences and given the different scenarios of interaction between the elements of the LinkBeyond system, this paper describes a security strategy which identifies the risks of elements and communication channels involved, proposing an end-to-end security approach, compatible with the user experience and with the capabilities of the hardware.

Keywords:

Mobility application, Multi-service, Threat modelling, End-to-End security

Introduction

Mobility paradigms are changing, not only in the way services are made available, but especially in the way how they are presented and consumed by citizens. The concept of Mobility as a Service (MaaS) is a fundamental instrument, demanding new tools to support the new business and operating models.

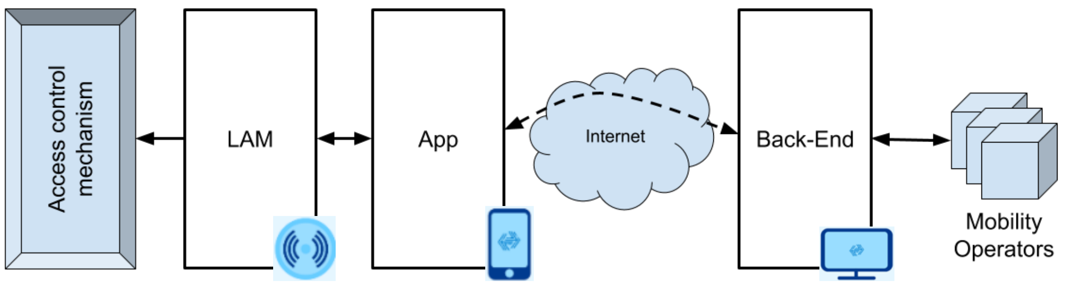
The A-to-Be's LinkBeyond Mobile solution (1) aims to position itself in the market as a way of interacting with mobility services through a mobile application, connecting operators and service providers to deliver a seamless mobility experience. The interaction with these mobility services allows to perform the discovery and consumption of services generating a transaction for payment, among other functionalities. In terms of architecture, the A-to-Be's LinkBeyond Mobile solution is made of an application on a mobile device (APP) that can communicate with a wireless device known as Local Access Mediator (LAM) module also developed by A-to-Be, which will interact with the operator's machine in order to provide the service. This communication between the application and the LAM module is performed on Bluetooth Low Energy (BLE) (2). Under different situations, the application also communicates, with a secure remote element via the Internet, the Back-End (BE), from where it obtains commands to activate different services.

Given the nature of the LinkBeyond Mobile solution, security is a critical requirement in order to avoid or detect attacks on different elements and the channels of communication between them. These attacks can arise from vulnerabilities in the communication protocols, but also from application design flaws. It is therefore necessary to define a security strategy that includes the protection of physical resources (LAM and mobile device) and virtual resources (APP and Back-End) exposed to cyberspace.

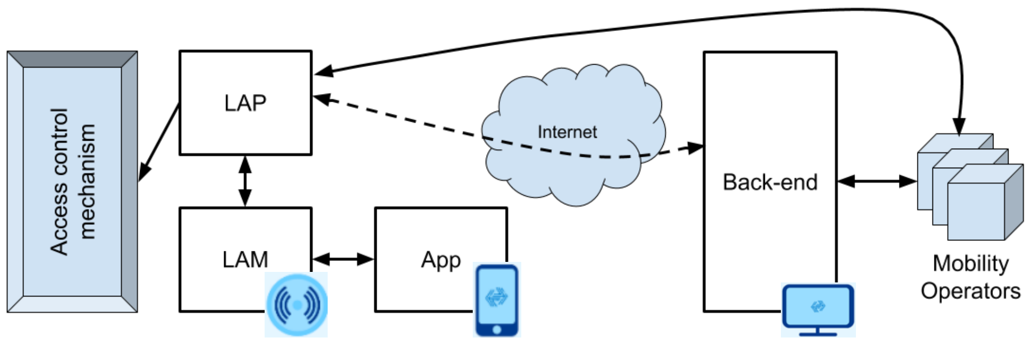
The architecture of LinkBeyond Mobile identifies two reference scenarios: a) a remote reference model, where transactions are commanded from the Back-End (e.g. vehicle vacuuming); b) a local reference model, where the transactions are commanded from the LAM (e.g. parking without mobile network coverage) in interaction with a Local Access Processor (LAP). In any of the scenarios the LAM must ensure the authenticity of the commands sent by the application, even when the application has no connection to the Back-End. Although the BLE technology provides secure pairing of devices, the use of this functionality is not anticipated, considering the usability requirements of the system.

This paper presents the relevant use cases of LinkBeyond Mobile and the main challenges addressed by its security strategy, including the connectivity and threat model regarding the different actors and communication channels.

Figure 1.a) presents the *remote scenario.* In this scenario, the LAM has no connectivity besides an eventual BLE link with the mobile application and the physical connection to the controlled service. The mobile application can have remote connection to the Back-End or could have connected to the Back-End sometime before standing near the LAM. Therefore, the application has Internet connectivity before interacting with the LAM but not necessarily when already in the radio range of BLE. When connected to the Internet, the model assumes the application has a secure connection to the Back-End. Figure 1.b) represents the *local scenario*, where the application has no back-end communication available, but LAM has indirect Back-End connectivity via a Local Access Processor (LAP) available on the LAM's local network.



(a) Remote scenario



(b) Local scenario

Figure 1: Reference models for different scenarios

Use Cases

LinkBeyond Mobile supports multiple scenarios that are portrayed on the following four use cases. Each use case poses significant challenges due to the lack of connectivity to the Back-End during the transaction, at some or even all of the elements. All the following use cases consider that the user has previously performed an online login in the Back-End and is correctly authenticated and authorized.

We use the term *transaction* to represent the interaction between the APP and LAM. It should not be confused with an interaction between the APP and Back-End to validate the acquisition and/or access to the service, although these two interactions (APP ↔ LAM and APP ↔ Back-End) can occur simultaneously or not from the user's point of view.

App Online and LAM offline

The first use case is characterized by the online status of the App. On this scenario the LAM is Offline. Examples of this use case are fuelling, car wash, or any other service where the App is online with the ability of receiving commands from the Back-End at the beginning and end of a transaction, with both commands triggered by the user. The App will present the nearby services for the user to select the desired service, receiving this information from nearby LAMs through BLE advertisements. The user is able to select and consume a service through the App connected to the Back-End. After a successful transaction the App will receive a command to be sent to the LAM for the LAM to process the service. LAM will confirm successful service activation to the App in order for the App to relay the information to the Back-End.

App Online and LAM Offline, with time constraints

This use case is like the previous one but with different time constraints and transaction process. An example of this use case is a BUS trip, for which the transaction is only to be finished at the end of the trip by the system. Additionally, this kind of service also has time constraints. For example, when entering the BUS, the communication between the App and LAM should be minimum to avoid congestions. Service termination may be inferred without user intervention, for example by detecting the lack of a BLE advertisement or GPS positioning.

APP Offline, LAM Online

In this case the App may not have connectivity to the Back-End and the LAM is connected to a local server. This is a common use case in an underground car parking. The App must autonomously generate a command with the information about the user. The LAM redirects this information to a local server (LAP) which checks if admission is possible of not. At the end of the service, the LAM/LAP sends confirmation to the Back-End.

APP and LAM Offline

In this use case both App and LAM don't have connectivity to the Back-End during the transaction. An example is entering and exiting a metro gateway. A list of services and secure commands must be previously obtained from the Back-End. When near a LAM announcing a supported service, the App will then use the correct command. The end of service is reported to the Back-End when connectivity is restored.

Summary

Table 1 presents examples of the previous use cases, including subway and bus portico activation, indoor parking, gasoline supply and car washing service. Notice that the APP and LAM are not online simultaneous.

Table 1: Use case examples

		LAM	
		online	*offline*
App	*online*		• Subway – receives from back-end command to enter • Bus – receives from back-end activation command • Bus – uses activation command • Bus – detects exit and sends information to back-end • Gasoline supply
	offline	• Parking – uses entrance command • Parking – uses exit command	• Subway – uses entrance command • Subway – used exit command

Risk Assessment

This section briefly presents the major security risks, how they can be explored, and the security strategies proposed to minimize them, including cryptographic mechanisms and application-level good practices.

During the evaluation the following assets were identified:

- The user, who is composed of his identity and status information;
- The confidentiality, integrity and authenticity of communication between the different components; The confidentiality, integrity and authenticity of the different physical and logical components;
- The system should be resilient to attacks that seek its unavailability;

Threat modeling is a process to be done in various stages of the development, and is particularly important in initial stages, during the system's architecture definition. There are widely accepted threat modeling processes. STRIDE (3) defines six threat categories, known as Spoofing (**S**), Tampering (**T**), Repudiation (**R**), Information disclosure (privacy breach or data leak) (**I**), Denial of service (DoS) (**D**) and Elevation of privilege (**E**).

The identified threats are:

- Threats to communication channels, that is, all the threats that listen, modify and reproduce the traffic that circulates in the different channels of communication. Examples of attackers will be all those who will have access to different communication channels, internet operators or individuals who capture traffic on wireless networks (Cellular, WiFi and BLE).
- Individuals with access to different components. They can be less trusted collaborators who have physical access to the servers and devices that make up the entire LinkBeyond Mobile ecosystem, as well as collaborators with access to the source code that runs on the components. As well as users interested in exploiting the system for their own benefit.

Globally the system must detect and resist to several types of attacks such as BLE network Sniffing, LAP/LAM/APP tampering or spoofing, Man-in-the-Middle and identity theft. Figure 2 identifies all the elements of LinkBeyond Mobile and their communication channels. The type of communication channel varies between elements. The `W` channel represents the Internet and the connection to the Web Services provided by the Back-End. Channel `B` is the Bluetooth Low Energy (BLE), channel `L` represents the local network and channel P is a private network. The `W`, `L`, and `P` channels will use a protocol supported on the TCP/IP stack.

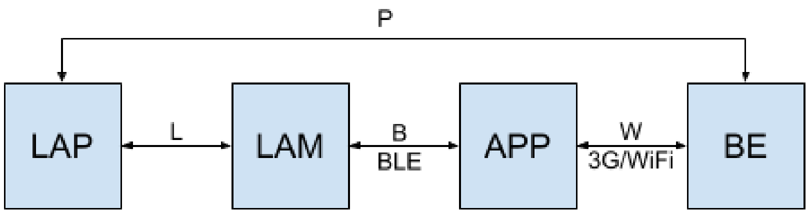


Figure 2: Summary of channels and elements

In general, communications should ensure that messages are exchanged with confidentiality and integrity, also considering the detection of message replay attacks. However, channel P is ideally private which will mitigate the risks.

The LAM is controlled by the operator. It is however publicly exposed and uses a widely used wireless protocol for which low-cost network analyzers are available. APP runs on general operating systems (Android and iOS) and must be able to communicate securely with the Back-End. It is also mandatory to store cryptographic material (obtained through the Back-End or established directly with the LAM) in a protected location managed by the application and operating system.

The latest Open Web Application Security Project (OWASP) study on mobile application vulnerabilities (4) shows that the biggest vulnerabilities come from not using the correct security mechanisms available in application development libraries. Along with misuse of the platform, unsafe data storage (e.g. incorrect password storage) and unsafe use of communications (e.g. non-certificate verification) take the top 3 places in the table of the top 10 vulnerabilities.

Table 2 presents a summary of the risks, for each of the assets previously identified. In addition, there should be mechanisms that discourage fraud whenever possible. Identified are all use cases where the application is not online, thus advising additional user validation, for example through a visual access log (e.g. matching license plates with users).

Table 2: List of risks, exploitation classification and mitigation

Risk	Name	Exploitation [difficulty]	Mitigation
R_{User}.1	Obtaining User Credentials	Social engineering or phishing [Average]	User education.
R_{APP}.1	Copy of user identity	Malware to access private storage [Average] and root access to storage area [Easy].	Provide means of time limiting cached information, such as access tokens.
R_{APP}.2	Copy of cryptographic keys stored in the application	Through access with root credentials to the device it is possible to obtain a copy of all the information that is available. [Average]	Use security enclaves available at the mobile device.
R_{APP}.3	Application Tampering	Adulterate the application using reverse engineering with tools that allow decompiling the application and access intermediate code/source thus generating a new malicious application. [Average]	User education to use the App only from Play Store of App Store.

$R_{LAM}.1$	Copy of cryptographic material	Through physical access to a LAM it may be possible to extract cryptographic keys stored in it, exploiting processor failures or accessing through debugging interfaces. [Difficult]	Use security enclaves available at the device.
$R_{LAM}.2$	LAM tampering/spoofing	The simulation of a LAM can lead to a man-in-the-middle attack to intercept the communications between the APP and the real LAM and manipulate the information exchanged or to analyse/change the communications between APP and LAM. [Average]	User education and mitigation of the risks related to the communication channels.
$R_{B}.1$	BLE Sniffing	Listening to the communication channel can be done using a BLE channel analyser. [Average]	Cipher and authenticate messages, with sequence numbers, and session keys. Disseminate previous transactions on a non-deterministic way using future user transactions.
$R_{B}.2$	Man-in-the-Middle	The tampering / generation of messages in the communication channel can be affected with the use of a BLE transmitter. [Difficult]	
$R_{B}.3$	Messages replay	Message can be repeated with an application that binds to the LAM and sends the message [Easy]; using a radio transmitter on the BLE frequency [Difficult]; or a combination of observing, tampering and repetition [Difficult].	
$R_{BE}.1$	Obtaining Cryptographic Material	(i) An external attacker gains privileged access to the server and key repository; [Difficult] (ii) A back-end administrator copies private cryptographic material. [Easy]	Apply security controls at the BE, by following current best practices, including the guidelines provided by the Open Web Security

R_{BE}.2	Obtaining user credentials	(i) an attacker who gains access to the backend database; [Difficult] (ii) an attacker attempting to log in with different passwords from a dictionary or by brute force [Easy]	Project (4)
R_{BE}.3	Back-end simulation	An attacker can simulate a Back-End and present it to the application running on a device that is also controlled by it. The attacker can gain insider information on the side of the application. [Easy] The attacker can also simulate a Back-End and present it to an application of a user to obtain information that can later be used to simulate the user. [Average]	
R_{BE}.4	Injection of commands at the Back-End	An attacker can exploit the Back-End via APP using specially crafted character sequences. These can be injected from, for example, the B-channel, by using specially crafted BLE announcements which will be sent to the Back-End. [Average]	
R_{W}.1	Network Sniffing	In this communication channel the modification would have to be done through the WiFi network (e.g. using a malicious access point) [Easy] or 3G [Difficult]. The attack can also be triggered by the managers of the networks where the communications pass. [Average]	Use Transport Layer Security (TLS), a end-to-end, well known, widely available, secure protocol
R_{W}.2	Man-in-the-Middle		
R_{W}.3	Message replay		
R_{LAP}.1	Tampering of access control list	(i) An external attacker gains remote and privileged access to the LAP and access list [Difficult] (ii) a LAP administrator manipulates the access list [Easy]	Apply security controls at the LAP, by following current best practices.

R_P.1	Man-in-the-Middle	Attacker must have physical access to the P-channel and use man-in-the-middle techniques to obtain, change, and generate spoofed messages. [Difficult]	Ensure the security of the P channel, for example by using a VPN.

Implementation of End-to-End Security

The implementation of the security strategy is limited by factors related to non-functional requirements (selected hardware, protocols, etc.) and functional requirements (form of interaction with APP and the access control mechanism). Since LinkBeyond Mobile uses BLE, the highest security mode supported is Mode 1 Level 4, which requires pairing using Secure Connections based on AES-CMAC and P-256 elliptic curves, only available in version 4.2 of Bluetooth. However, security modes are only available after pairing, and because LAM does not include any channel for key exchange, the only acceptable pairing mode is “Just Works”. In this situation we are facing Mode 1 Level 1 security, i.e. we do not have any active BLE security mechanism, so security must be ensured by other mechanisms.

The LAM analyzed in this study uses a Kinetis K64F NXP processor from the ARM Cortex-M4 family. Integration tests of the mbedtls library (5) have been done. During tests it was also determined that the transmission of a single BLE frame, with 23 bytes, is done in approximately 100 ms. To minimize the time needed to exchange commands, and improve the user experience, each protocol message should fit a single BLE frame. In order to do so, a protocol based on symmetric keys and mechanisms is used, inspired by the Kerberos protocol, where Back-End plays the role of Key Distribution Centre (KDC) and Ticket-Granting Service (TGS). Also, like Kerberos, the proposed protocol needs a symmetric shared key between the LAM (with the role of “service”) and Back-End (with the role of KDC/TGS). This key must be installed in a secure way in the LAM. The operator uses a secure channel to install a Back-End signed symmetric key, which the LAM can validate with a built-in trusted public key. Figure 3 depicts these two keys installed at the LAM.

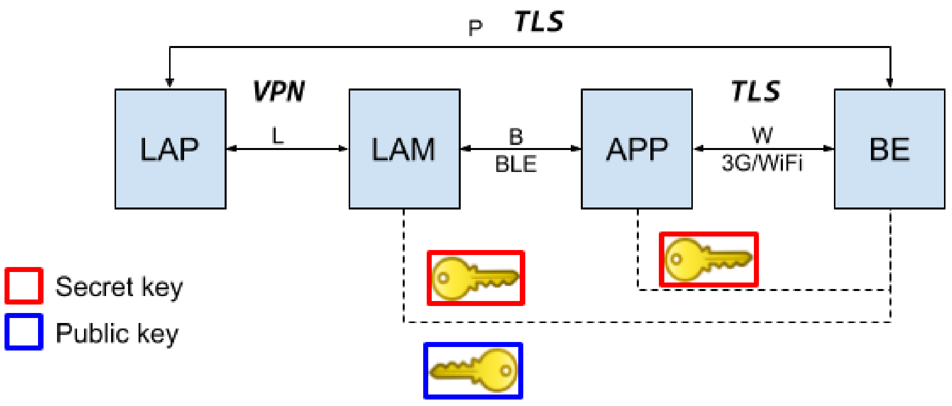


Figure 3: Summary of cryptographic elements and channels

In the previously identified use cases the command is obtained considering a) a BLE advertisement obtained from a LAM; b) according to a GPS position obtained from the APP; or c) user selection. In all cases, the LAM must verify the authenticity of the command and, consequently, the APP that is sending it. To do so, the LAM analyses the command through a process ensuring that a) the APP successfully authenticates to the Back-End b) can detect a replay of previous commands. The APP presents commands to the LAM which are obtained from the Back-End and encapsulated in a ticket. This ticket, generated by the Back-End, and delivered to the APP via secure channel (e.g. HTTPS), uses authenticated cipher, that is, it is encrypted and authenticated with the key shared between the LAM and the Back-End. Figure 3 depicts the ephemeral secret key shared between the APP and the BE.

Conclusions

A security strategy is essential for an ecosystem like LinkBeyond Mobile. The solution is currently under development, with some use cases currently being tested in real deployments. Given this approach, the security strategy is comprehensive to allow the mitigation of different risks anticipated or not. The strategy is divided into four major use cases, which depend on the APP and LAM having stable connection (direct or indirect) to the Back-End during transaction and constrains regarding the activation time of the access control mechanism. When the introduction of stronger mechanisms would decrease the user experience, we suggested risk minimization strategies using non-protocol mechanisms, such as installing video surveillance to discourage fraud or campaigns to rise the security awareness of users. A security strategy should not be understood as an immediate and quick solution to all risks. The strategy must be periodically renewed to make sure risks introduced by new use cases are covered by the proposed solutions or need to be specifically addressed.

Acknowledgments

This work was supported by A-to-Be, through the project “A-to-Be LinkBeyond: Estratégia de segurança”.

References

1. **LinkBeyond, A-to-Be.** https://www.a-to-be.com/mobility-payments/a-to-be-linkbeyond-line/. [Online]

2. **Bluetooth Special Interest Group.** *Bluetooth Core Specification Version 4.1.* 2013.

3. *A Descriptive Study of Microsoft's Threat Modeling Technique.* **Scandariato, Riccardo and Wuyts, Kim and Joosen, Wouter.** 2, s.l. : Springer-Verlag New York, Inc., 2015, Vol. 20. 0947-3602.

4. **OWASP, Open Web Application Security Project.** https://www2.owasp.org/www-project-mobile-security-testing-guide/. [Online]

5. **MbedTLS.** Linaro - https://www.trustedfirmware.org/. [Online]