Device Location Certification for the Internet of Things

Deliverable I: Initial Architecture

Author:
Miguel L. Pardal

Secondary Authors:
Rui L. Claro
Gabriel A. Maia
Henrique F. Santos
Pedro E. Carmo
Sheng Wang

SureThing is a scientific project funded by the Portuguese national funding agency for science, research and technology (FCT) with reference No. PTDC/CCI-COM/31440/2017

Published: March 2019

Last Revision: November 2019

Contents

Sumário Executivo

O projeto SureThing aborda uma necessidade de segurança na Internet das Coisas (IdC): a criação e validação de certificados de localização. O objetivo é permitir que os dispositivos limitados, necessários para a prestação de serviços na IdC, possam fazer prova da sua localização ou pedir provas a outros dispositivos. Os certificados emitidos usando o SureThing contêm dados de localização obtidos usando técnicas com medições de rede sensíveis à localidade, por exemplo, com valores locais de WiFi e Bluetooth. O enquadramento SureThing pretende ser extensível para que novas técnicas de localização, desenvolvidas no projeto ou pela comunidade, possam ser facilmente incorporadas.

Este documento apresenta uma versão inicial da arquitetura do SureThing, em que são identificados os conceitos mais importantes, com base em arquiteturas de trabalhos relacionados.

Executive Summary

The SureThing project addresses an Internet of Things (IoT) security need: creating and validating location certificates. Its goal is to allow for constrained devices, needed in the provisioning of IoT services, to obtain proof of their location or to request proof of location to other devices. The certificates issued using the SureThing framework contain location data, obtained and verified using one or more locality-sensitive measurements, for example, WiFi and Bluetooth signals. The SureThing framework aims to be extensible so that new location techniques, developed in the project or by the community, can be incorporated.

This document presents the initial version of the SureThing architecture, with the identification of the most important concepts, based on architectures from related work.

Chapter 1

Introduction

The scale and geographic dispersion of the Internet of Things (IoT) [23] will surpass the size of the current day Internet in, at least, 3 orders of magnitude. The IoT will be the largest and most widely distributed system ever, with a multitude of connected sensors and actuators. The majority of the interactions will occur between machines without human intervention. The current Internet already has some serious, unresolved security problems. Adding physical world autonomous connections brings even more concerns about attacks and their consequences to people and goods [19].

The SureThing project is addressing a timely IoT security need: creating and validating location certificates. Its goal is to allow for constrained devices, needed in the provisioning of IoT services, to obtain proof of their location or to request proof of location to other devices. The certificates issued using the SureThing framework will contain location data, obtained and verified using one or more state-of-the-art techniques. These include locality-sensitive network measurements, including WiFi and Bluetooth fingerprinting, and ambience sensing.

The existing techniques only consider the use of smartphones. We are researching ways to adapt these techniques to more limited devices [22] and to protocols better suited to the IoT, like COAP [15] and MQTT [13]. The SureThing framework aims to be extensible in order to allow for the novel techniques developed in this project or by the research community to be easily integrated as they appear.

The SureThing framework will allow developers to choose between faster location proofs and more reliable proofs, which will be digitally signed and kept in a ledger. The witness models, providing a validation role for other devices at the same location, will play an important role, particularly when only limited cryptographic mechanisms are

available. The witness models assist in validating location, and provide orchestrations involving identity providers and using anonymization techniques to assure adequate witness privacy protection.

Overview

This document is structured as follows. Chapter 2 presents work related to location systems and location proof systems. Chapter 3 presents the initial version of the architecture of the SureThing framework. Chapter 4 concludes the document.

Chapter 2

Related Work

Location information is central to many applications, and geographic information systems facilitate the development of such location-aware solutions. The location information provided by these systems can be verified by location proof systems. Privacy-preservation techniques are also important to protect the information.

2.1 Geographic Information Systems

Geographic information systems (GIS) are used in location systems as the means to locate and manage information about points of interest and routes. Web mapping software such as Google Maps, Bing Maps and OpenStreetMap, and satellite imagery software of which Google Earth is an example, are widely-known types of GIS. These systems are used, for example, to plan trips [14] or to perform package deliveries.

Web mapping systems sometimes feature user-contributed or crowdsourced information, with users being able to submit corrections or new points of interest. OpenStreetMap is notable in this regard because it is a collaborative project built entirely on crowdsourced data. Often, web mapping systems expose APIs that allow their integration in custom applications. MapBox is one example of a web mapping system whose primary focus is providing custom maps for embedding in other applications.

2.2 Location Tracking

Location systems can provide *physical* and *symbolic* information. The former locates an object in absolute terms within a coordinate system, while the latter consists on

abstract information about the position of an object. Symbolic location systems usually can only provide low-precision physical positions. Location systems can also be classified as *absolute*, where a shared reference is used for all located objects, and *relative*, where each object can have a different frame of reference. Absolute positions can typically be transformed into relative positions with ease, but the reverse can prove to be complex, requiring the use of triangulation or trilateration and the knowledge of the absolute positions of other objects [12].

Location techniques can also be broadly divided into *geolocation* and *microlocation* techniques. The former focuses on determining geographic location, while the latter can determine the location of something within a limited space, such as an university campus, building or room, with typically much higher precision than geolocation techniques. Often, microlocation can be used for indoor location, i.e. in confined environments without a clear view of the sky and where precision is usually more important. A useful example of such an environment is a building with multiple floors.

A location tracking system uses one or more technologies to collect location information about a device usually attached to an object or used by a person. This allows the system to have the set of location points of an object or person during a specific time period, therefore enabling the tracking of the location of warehouse items, vehicles or people, for instance.

2.2.1 Technologies

One use case of GPS technology is managing a fleet of vehicles. Transportation companies often have or sub-contract a *fleet management system* with vehicle location tracking to optimize the costs and use of vehicles. Providers of these systems like InoSat[1] and CarTrack[2] install a GPS tracking unit in a vehicle and this device reports location and other relevant data, like vehicle speed and temperature, to servers. *InoSat* configures and only supports a specific type of on-board devices for tracking and all involved entities accept that the installed unit is trustful. This on-board device collects relevant data, connects to the nearest Global System for Mobile Communications (GSM) cell tower and sends the information to servers of *InoSat*. The company provides a web service where the client visualizes collected information. Being a proprietary solution, a client

[1] `http://www.inosat.pt/`
[2] `https://www.cartrack.pt/`

or regulator cannot access the data directly or guarantee that data is tamper-resistant.

There are also indoor location tracking solutions using other technologies. *Locix*[3] provides a *Local Positioning System* consisting on dedicated and proprietary devices to track assets in a warehouse. These units are positioned together with the object to track and the system enables the location and tracking of inventory by using the 802.11ac Wi-Fi standard[4] and proprietary algorithms. The devices can also be integrated with fleet tracking and operations systems though an Application Programming Interface (API).

2.2.2 APIs

Location context is relevant for many mobile applications. For this reason, the most widely used mobile operating systems, Android and iOS, both provide APIs that manage the access of applications to the Global Navigation Satellite Systems (GNSS) supported by most smartphones sold in the last decade [2, 3]. These systems can be used for stand-alone geolocation of a mobile device, but are of limited use in areas without line-of-sight to a sufficient number of satellites, including indoors and underground. Their accuracy can also vary widely depending on the number and visibility of the satellites and characteristics of the receiver, from 1 meter to over 50 meters [6, 17].

Additional APIs made available by Google [11], Apple [5] and Microsoft [1] on Android, iOS and Windows, respectively, use their proprietary services to combine GNSS with additional information, such as Wi-Fi access points and GSM[5] cell tower information, to determine the location of a device with better accuracy, lower power usage, or under poor sky visibility conditions. This additional information is kept in large databases, actively collected and maintained by the mentioned entities. However, these APIs often require the device to have an active Internet connection, and that users agree to privacy policies governing the access of these companies to user location data. These solutions are not stand-alone, as part of the computation may be offloaded to their online services.

[3] `https://www.locix.com/`

[4] `https://standards.ieee.org/standard/802_11ac-2013.html`

[5] Global System for Mobile communications, the cellular communication standard, that popularized the use of mobile phones in Europe and around the world.

2.3 Location Proofing

The majority of the previously presented works use trusted devices to collect location information. However enabling the usage of untrusted devices to collect location data increases the number of devices that can be used for this purpose and reduces the cost of implementing a location tracking system by using smartphones and other low cost devices. These devices are considered untrusted because GPS signals, for instance, can be spoofed, therefore mechanisms have been created to prove the location of one untrusted device using other trusted or untrusted devices.

Saroiu and Wolman defined *location proof* as a mechanism to allow mobile devices to prove their location to applications and services [20]. The authors considered that a component of an existent wireless infrastructure such as a Wi-Fi Access Points (AP) and a cellular cell tower can issue meta-data which mobile devices can use to prove their location. A device can therefore request a location proof from the infrastructure and this proof can be sent to applications with the intent of proving the location of the mobile device. The scenario implemented takes advantage of beacon frames transmitted by a Wi-Fi AP when announcing its existence. The concept assumes the AP is a trusted witness. The authors suggest the use of APs with a GPS module where a person places the AP outside of the building to setup the GPS coordinates and then places it in the desired indoor location. However this procedure requires substantial human intervention and it is not practical as most APs do not have this module and are directly placed at the desired location.

The system uses asymmetric cryptography to guarantee authentication and encryption, where each participating node contains a public and a private key. The holder of the private key sends messages encrypted with this key and other nodes use the paired public key of the sender to decrypt the message, authenticating the sender. This is also known as a *digital signature*. Additionally other nodes can encrypt messages with the public key of one node, ensuring that these messages are only decrypted by this node, as it is the only holder of its private key.

The protocol starts when a client receives a beacon frame and then sends a proof request with the public key of the client and the sequence number of the frame signed with the private key. The sequence number prevents *replay attacks*, where requests are

repeated or delayed, and the signature prevents *integrity attacks*, where the message is tampered. After validating the request, the AP broadcasts a signed location proof containing its public key, the public key of the client, the current timestamp and the latitude and longitude geographical coordinates of the location. The AP does not check if the client received the location proof. Upon receiving the proof, the client signs it and transmits it to the application or service to use, who then decrypts the message with the public key of the client and checks the public keys contained within the content of this location proof.

The following works go beyond locating the user, by also proving that location in a verifiable way. Location proof systems focus on countering location spoofing. Among other things, these systems allow for the implementation of location-based authentication schemes.

CREPUSCOLO [8] uses strategically placed *Token Providers* and neighboring devices to provide location verification, even in areas with reduced amounts of users. Privacy is preserved through the use of periodically changing pseudonyms. In contrast with neighbor-based location proof systems APPLAUS [24] and LINK [21], CREPUSCOLO can protect against not just simple collusion attacks but also wormhole attacks, where packets are tunneled by an attacker from one physical place to another.

The SureThing system [10] allows devices to produce and validate location certificates, to make proof of their location and to reliably verify the location of other devices. Like CREPUSCOLO, SureThing uses neighboring devices as witnesses, together with the geographic location obtained by each device, e.g. using GNSS. A central component, the *verifier*, is responsible for certifying proofs. To prevent collusion, the system takes advantage of the diversity of witnesses and of the concept of *redundancy* (multiple witnesses) and *decay* (proofs from the same witness lose their value as they are used to testify user presence).

LINK [21] works similarly to SureThing, using neighboring devices reachable over short-range wireless technologies, such as Bluetooth, as witnesses. This system also associates a trust score with each device. As with SureThing, a central component is responsible for the decision process that certifies claims. In LINK, this process takes into account historical data and trust score trends to detect colluding users. Unlike SureThing, LINK does not make use of witness diversity: while each user/device is still expected to have the means to locate themselves (e.g. using GNSS), they are not

expected to collect different types of location proofs from the environment, but from LINK neighbors only (it assumes users are not alone very often).

Agadakos et al. [4] proposed the Icelus system, which can locate users and model their movement through the use of IoT devices and smart environments. The paper details concrete measures for preserving privacy when parts of the system are hosted by third parties: third-party hubs can only learn the distance between devices, and not the positions of the devices. This is ensured through the use of a custom sub-protocol for communication between the hubs and the entities that request location proofs. This sub-protocol uses *secure multi-party computation*, *additively homomorphic encryption* and *additive blinding*. The accuracy and efficiency of the system were evaluated and compared with smartphone-only approaches, with Icelus presenting lower false acceptance and false rejection rates. The researchers concluded that their approach exhibited a low impact on the performance of the tested IoT devices, with effectiveness bound to improve as the number of deployed IoT devices increases.

2.4 Privacy-preservation techniques

Privacy is a primary concern when dealing with exact and certifiable user location information, as provided by location systems and location proof systems. Beresford and Stajano [7] defined *location privacy* as a type of information privacy that consists in "the ability to prevent other parties from learning one's current or past location".

Location privacy has been an object of research since before location systems became ubiquitous with the rising popularity of smartphones. In 2002, Langheinrich presented pawS [16], a system that lets users make privacy choices as location data is captured. In 2003, Myles et al. [18] presented a system that, through the use of machine-readable privacy policies, controls the release of location data as it is requested.

Beresford and Stajano [7] proposed a framework for privacy protection based on the use of pseudonyms that change periodically. Icelus [4] uses homomorphic encryption for processing data on third-party servers, such that the respective entities can process but not learn the location of the users.

Fawaz and Chin [9] designed LP-Guardian, a location privacy protection framework for Android that works on a per-app basis. It prevents user identification and can be deployed in practice without modifying other applications, while still providing useful

location information. Instead of simply blocking all location requests or always returning false information, their solution decides whether to provide location information and if it should be anonymized on a per-application, per-location basis, requesting a decision from the user if preferences have not been previously set for the application requesting the location. LP-Guardian can also provide applications with synthetic routes so those can calculate the traveled distance, for example, without accessing the true route coordinates. Notably, a solution for navigation applications, that require precise location information for extended periods of time, is not provided. The evaluation performed by the authors of LP-Guardian, on Android 4.3, found it to be easy to deploy, to have acceptable energy consumption, and to cause a tolerable loss in application functionality. LP-Guardian requires the Android operating system to be modified, therefore it does not work with the smartphones currently in the market.

Chapter 3

Architecture

SureThing defines a model comprised of different entities that all contribute to generate a central concept: the location proof. Figure 3.1 presents the current iteration of the conceptual model.

For the trust root of SureThing, we assume that there is a Certification Authority (CA), responsible for signing a public key certificate for each user. Each user has its own key pair, with a private and a public key. The CA certifies all the needed keys among the users, being a trusted entity.

SureThing has two main entities: a Prover P and a Verifier V. We assume that each user of the system has to have a unique identifier when he is registering in the system. P is the user that wants to proof his location and will have to gather proofs from his neighbors. V is some entity that requires a location proof from a user. V is providing a service to P, depending on his location, and has to receive location proofs that indicate that P is really in the claimed place.

It is important to notice that a user of our system can, at a given point, ask for some location proof for himself, but can also, in some cases, be a witness for others. A Witness W is a user that is nearby P and that will give him a proof of his presence in the place.

There are different proof techniques that depend on location technologies, both for macrolocation and for microlocation. These techniques produce a fingerprint. The Verifier then needs a fingerprint matcher to compare the fingerprint presented by the prover with the witness accounts.

These concepts can be instantiated in different ways, for different systems, to support specific use cases.

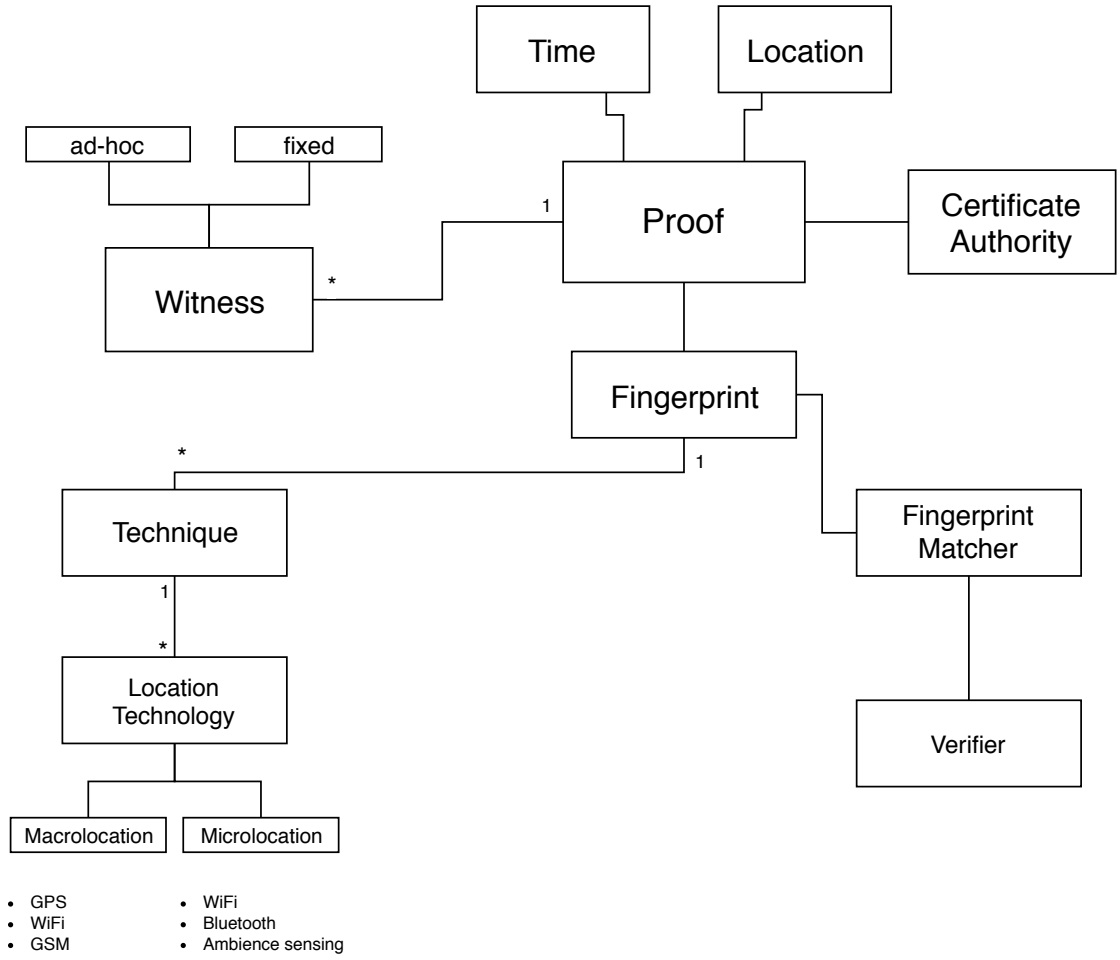

Figure 3.1: Main concepts of the SureThing framework.

Chapter 4

Conclusion

This document presented an initial version of the SureThing framework: the entities and interactions between them. The proposal is supported by related work in the topics of location technologies and location proofs.

The SureThing conceptual model will have to be further detailed and instantiated in the context of two use cases. The first, Smart Tourism, which will build an application providing tourists with awards for each visit to a predefined set of locations, making use of fast location proofs. The second, Smart Taxes, will build another application, making use of more reliable proofs, with trusted witnesses. The use of digital notaries with time-stamping and long term proof archival will also be considered.

The widespread use of SureThing location proofs will significantly improve the security decisions of policies for the IoT. This will lead to more secure and trustable services in the near future.

Bibliography

[1] Windows.Devices.Geolocation Namespace - UWP App Developer — Microsoft Docs.

[2] Canalys research release 2008/082, 2008.

[3] GPS and mobile handsets, 2010.

[4] Ioannis Agadakos, Per Hallgren, Dimitrios Damopoulos, Andrei Sabelfeld, and Georgios Portokalidis. Location-enhanced authentication using the IoT. In *Proceedings of the 32nd Annual Conference on Computer Security Applications - ACSAC 16*. ACM Press, 2016.

[5] Apple Inc. Core Location — Apple Developer Documentation.

[6] Christine Bauer. On the (in-)accuracy of GPS measures of smartphones. In *Proceedings of International Conference on Advances in Mobile Computing & Multimedia - MoMM 13*. ACM Press, 2013.

[7] A.R. Beresford and F. Stajano. Location privacy in pervasive computing. *IEEE Pervasive Computing*, 2(1):46–55, jan 2003.

[8] Eyup S. Canlar, Mauro Conti, Bruno Crispo, and Roberto Di Pietro. CREPUSCOLO: A collusion resistant privacy preserving location verification system. In *2013 International Conference on Risks and Security of Internet and Systems (CRiSIS)*. IEEE, oct 2013.

[9] Kassem Fawaz and Kang G. Shin. Location privacy protection for smartphone users. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security - CCS 14*. ACM Press, 2014.

[10] João Ferreira and Miguel L. Pardal. Witness-based location proofs for mobile devices. In *17th IEEE International Symposium on Network Computing and Applications (NCA)*, November 2018.

[11] Google LLC. Location and Context APIs.

[12] J. Hightower and G. Borriello. Location systems for ubiquitous computing. *Computer*, 34(8):57–66, 2001.

[13] Urs Hunkeler, Hong Linh Truong, and Andy Stanford-Clark. Mqtt-s—a publish/-subscribe protocol for wireless sensor networks. In *2008 3rd International Conference on Communication Systems Software and Middleware and Workshops (COMSWARE'08)*, pages 791–798. IEEE, 2008.

[14] Dongwook Kim and Sungbum Kim. The role of mobile technology in tourism: Patents, articles, news, and mobile tour app reviews. *Sustainability*, 9(11):2082, nov 2017.

[15] Matthias Kovatsch, Martin Lanter, and Zach Shelby. Californium: Scalable cloud services for the internet of things with coap. In *2014 International Conference on the Internet of Things (IOT)*, pages 1–6. IEEE, 2014.

[16] Marc Langheinrich. A privacy awareness system for ubiquitous computing environments. In *UbiComp 2002: Ubiquitous Computing*, pages 237–245. Springer Berlin Heidelberg, 2002.

[17] Xingxing Li, Maorong Ge, Xiaolei Dai, Xiaodong Ren, Mathias Fritsche, Jens Wickert, and Harald Schuh. Accuracy and reliability of multi-GNSS real-time precise positioning: GPS, GLONASS, BeiDou, and Galileo. *Journal of Geodesy*, 89(6):607–635, mar 2015.

[18] G. Myles, A. Friday, and N. Davies. Preserving privacy in environments with location-based applications. *IEEE Pervasive Computing*, 2(1):56–64, jan 2003.

[19] Rodrigo Roman, Jianying Zhou, and Javier Lopez. On the features and challenges of security and privacy in distributed internet of things. *Computer Networks*, 57(10):2266–2279, 2013.

[20] Stefan Saroiu and Alec Wolman. Enabling new mobile applications with location proofs. In ACM, editor, *Proceedings of the 10th workshop on Mobile Computing Systems and Applications*, page 9, 2009.

[21] Manoop Talasila, Reza Curtmola, and Cristian Borcea. LINK: Location verification through immediate neighbors knowledge. In *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, pages 210–223. Springer Berlin Heidelberg, 2012.

[22] Wade Trappe, Richard Howard, and Robert S Moore. Low-energy security: Limits and opportunities in the internet of things. *IEEE Security & Privacy*, 13(1):14–21, 2015.

[23] Dieter Uckelmann, Mark Harrison, and Florian Michahelles. *Architecting the internet of things*. Springer Science & Business Media, 2011.

[24] Zhichao Zhu and Guohong Cao. APPLAUS: A privacy-preserving location proof updating system for location-based services. In *2011 Proceedings IEEE INFOCOM*. IEEE, apr 2011.