To appear in:
Proceedings of The ACM SIGBED International Conference on Embedded Software (EMSOFT), Seoul, Republic of Korea, October 15–20, 2017

Work-in-Progress: Contextual Callbacks for Resource Discovery and Trust Negotiation on the Internet of Things

Marten Lohstroh
University of California, Berkeley
marten@eecs.berkeley.edu

Hokeun Kim
University of California, Berkeley
hokeunkim@eecs.berkeley.edu

Edward A. Lee
University of California, Berkeley
eal@eecs.berkeley.edu

ABSTRACT

This paper introduces contextual callbacks, which allow environments to authenticate themselves to nearby devices and advertise local services in response to the reception of radio-broadcast announcements that are emitted by mobile devices.

CCS CONCEPTS

•Computer systems organization →Embedded and cyber-physical systems; •Human-centered computing →*Mobile computing;* **•Security and privacy →Mobile and wireless security;**

ACM Reference format:
Marten Lohstroh, Hokeun Kim, and Edward A. Lee. 2017. Work-in-Progress: Contextual Callbacks for Resource Discovery and Trust Negotiation on the Internet of Things. In *Proceedings of EMSOFT'17 Companion, Seoul, Republic of Korea, October 15–20, 2017,* 2 pages.
DOI: 10.1145/3125503.3125629

1 INTRODUCTION

The Internet of Things (IoT) facilitates the composition of cyber-physical systems (CPS) through networking. Key to the functioning of "smart things" is maintaining semantic ties to the physical processes they interact with; a **context** is naturally considered to impose constraints that should lead to a particular set of allowed/disallowed behaviors. Context awareness requires computation to be "anchored" by inputs that encode physical or logical relationships in an application's environment. In IoT and CPS-related research, there is a lot of emphasis on physical sensing and actuation, but the context made up of logical relationships between entities, such as trust and ownership, is equally important.

If it is incumbent on a computational process to contextualize itself using cues from its environment, then the following questions arise: "Where do these cues come from?" and "Under which circumstances can they be trusted?" The first question raises the problem of **resource discovery**, while the second is concerned with the problem of **trust negotiation**. There are numerous ways in which tampering with physical resources can lead to very impactful security breaches (e.g., side channel attacks on wireless channels [5]) or safety hazards (e.g., remote attacks on traffic controllers [2]); therefore, it is essential to establish a trust relationship and set up a secure channel with newly discovered resources *prior to* engaging with them.

We propose an architecture that facilitates resource discovery and trust negotiation for IoT applications beyond the scope of simplified local-area home automation scenarios. Our design builds upon an existing IoT authentication and authorization framework and leverages the novel concept of **contextual callbacks**, which forms the key contribution of this paper.

2 BACKGROUND

The availability of mechanisms to establish and manage trust is important for the success of the IoT [7]. Applying context-aware computing to the IoT [6] can foster interoperability among a heterogeneous collection of Things. The idea of leveraging context awareness for trust management in IoT applications has been suggested in [1]. However, realization of this idea is still an underexplored area. Callbacks or asynchronous callbacks have been used in the past as a mechanism for dealing with discovery and initialization problems in computer systems (e.g., [8]).

3 ARCHITECTURE

Our proposed design extends an existing open-source toolkit called SST (Secure Swarm Toolkit) [3]. SST provides an authentication and authorization infrastructure for the IoT environment. In SST, devices have trust relationships with *Auth*, a *locally centralized* authentication and authorization entity, hosted on an **edge device** such as a smart router. SST brokers trust between devices via *globally distributed* trust relationships between Auths. This scheme eliminates the need for a centralized Cloud-based authentication service, which a recent mass outage of Google's OnHub routers [4] has illustrated can easily become a single point of failure.

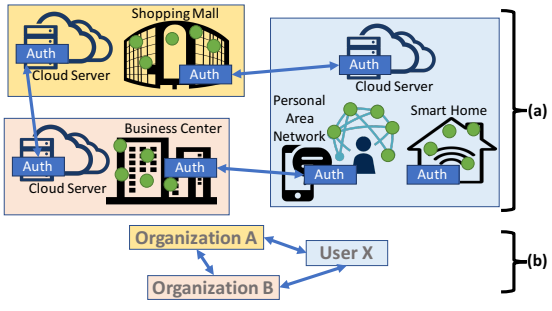

Figure 1: (a) SST: locally centralized, globally distributed. (b) Extension: logically centralized, physically distributed.

This work was supported in part by TerraSwarm, one of six centers of STARnet, a Semiconductor Research Corporation program sponsored by MARCO and DARPA.

EMSOFT'17 Companion, Seoul, Republic of Korea

978-1-4503-5186-7/17/10…$15.00
DOI: 10.1145/3125503.3125629

Auth can be seen as a proxy for an entity (e.g., an individual, a group, or an organization) that administers trust relationships with devices owned or managed by that entity. We extend the notion of Auth to a *collection* of instances that jointly represent a single entity, where an instance can be characterized as follows: 1) global representation, 2) local/edge representation, and 3) mobile representation. The global representation of Auth is deployed in the Cloud and will be used for accepting callbacks from edge devices, as Section 4 explains. The mobile representation is used for discovery, leveraging its mobile presence. The local/edge representation is used for facilitating secure communication between local devices, some of which may be mobile. Figure 1 illustrates (a) the existing SST architecture and (b) our extension of it, which yields a *logically centralized, physically distributed* architecture.

4 EXAMPLE: INDOOR LOCALIZATION

Consider the following scenario: a user walks into a retail store with a shopping list on her mobile device. Upon entering the store, her device receives a message from the retailer, advertising two services: an indoor localization service and an inventory service. If the user trusts the retailer, her mobile device can now automatically match the shopping list against the local inventory, calculate the optimal route through the store, and provide turn-by-turn navigation.

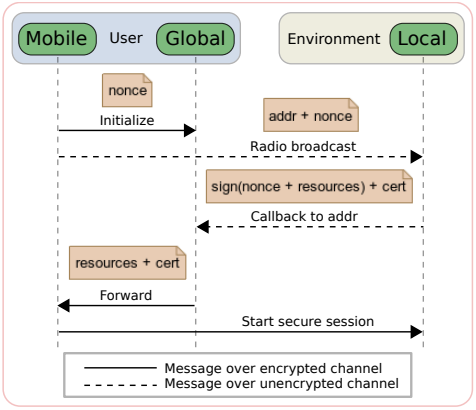

Figure 2: Resource discovery via contextual callback.

We enable this scenario through the message exchange detailed in the sequence diagram in Figure 2. We let the user's mobile instance of Auth periodically announce itself by broadcasting a message that contains a random number (nonce) and reference to the user's global Auth. The environment runs Auth on an edge device to listen for announcements. In response to an announcement, the environment sends a message—a contextual callback—to the user's global Auth. The callback message contains a digital certificate (associated with the environment's Auth) along with a signed bundle that contains the nonce and a list of available resources. The user's global Auth relays the callback to the mobile device, and the user then decides whether or not to engage with any of the presented resources based on the provided certificate. The signed nonce in the callback message ensures that the callback indeed originated from the user's earlier announcement.

5 DISCUSSION

Conventional resource discovery (e.g., zeroconf, mDNS) works on the assumption that a device is connected to a trusted network. Mobile devices use radio antennas to connect to networks, and can typically only connect to one network at a time. Often, multiple wireless networks are available in the same physical space. Unaffiliated devices cannot explore password-protected networks, even though precisely the availability of certain resources may provide a reason for requesting network access. We observe that conventional discovery, at best, can explore only a small fraction of the resources that a physical environment may have to offer.

On the other hand, edge devices can simultaneously listen for broadcast radio packets of different modalities via different antennas (e.g., Bluetooth, WiFi, Zigbee) while remaining connected to the Internet. For a proof of concept, we have confirmed that a mobile phone with stock Android can be made to periodically broadcast an arbitrary string (which could be a URL) simply by entering this string as an SSID in the list of its known networks. In order to retrieve the string, all that an edge device has to do is listen for 802.11 Management Frames that are sent out by the phone when it starts looking for known networks. This string can provide the reference needed to issue a contextual callback.

Due to the limited range of radio signals, callbacks are bound to emerge from the vicinity where the device announces itself. And with a limited time to live for each nonce, a callback can be tied to a particular time window. Letting mobile devices announce themselves via radio, instead of actively searching for resources on a network, circumvents the problem of access limitations and imposes very little overhead, which is especially advantageous for battery-powered mobile devices. Announcements can be anonymized through the use of a web service similar to URL-shortener services, such as goo.gl or bitly.com, except it will not redirect HTTP requests but instead forward callback messages. Finally, our approach is universal in the sense that contextual callbacks can be made to work with virtually any radio technology.

REFERENCES

[1] Yosra Ben Saied, Alexis Olivereau, Djamal Zeghlache, and Maryline Laurent. 2013. Trust management system design for the Internet of Things: A context-aware and multi-service approach. *Computers & Security* 39, Part B (November 2013), 351–365.

[2] Branden Ghena, William Beyer, Allen Hillaker, Jonathan Pevarnek, and J. Alex Halderman. Green Lights Forever: Analyzing the Security of Traffic Infrastructure. In *8th USENIX Workshop on Offensive Technologies (WOOT 14)* (2014).

[3] Hokeun Kim, Eunsuk Kang, Edward A. Lee, and David Broman. A Toolkit for Construction of Authorization Service Infrastructure for the Internet of Things. In *Proceedings of the Second International Conference on Internet-of-Things Design and Implementation* (2017) *(IoTDI '17)*. ACM, 147–158.

[4] Ian Morris. 2017. Google's Latest Failure Shows How Immature Its Hardware Is. *Forbes* (Feb. 2017). http://www.forbes.com/sites/ianmorris/2017/02/24/googles-latest-failure-shows-how-immature-its-hardware-is/

[5] Ali Akbar Pammu, Kwen-Siong Chong, Weng-Geng Ho, and Bah-Hwee Gwee. 2016. Interceptive side channel attack on AES-128 wireless communications for IoT applications. In *2016 IEEE Asia Pacific Conference on Circuits and Systems (APCCAS)* (2016-10). 650–653.

[6] Charith Perera, Arkady Zaslavsky, Peter Christen, and Dimitrios Georgakopoulos. 2014. Context Aware Computing for The Internet of Things: A Survey. *IEEE Communications Surveys Tutorials* 16, 1 (2014), 414–454.

[7] Zheng Yan, Peng Zhang, and Athanasios V. Vasilakos. 2014. A survey on trust management for Internet of Things. *Journal of Network and Computer Applications* 42 (2014), 120–134.

[8] Yiling Yang, Yu Huang, Xiaoxing Ma, and Jian Lu. 2016. Enabling Context-Awareness by Predicate Detection in Asynchronous Environments. *IEEE Trans. Comput.* 65, 2 (2016), 522–534.