2018 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/ 12th IEEE International Conference On Big Data Science And Engineering

Providing Context-Aware Security for IoT Environments Through Context Sharing Feature

Everton de Matos*†, Ramão Tiago Tiburski*, Leonardo Albernaz Amaral*, Fabiano Hessel*
Pontifical Catholic University of Rio Grande do Sul (PUCRS) - Porto Alegre - RS - Brazil*
(everton.matos.001, ramao.tiburski, leonardo.amaral)@acad.pucrs.br, fabiano.hessel@pucrs.br
Meridional Faculty (IMED), Polytechnic School - Passo Fundo - RS - Brazil†
everton.matos@imed.edu.br

***Abstract*—Nowadays security and privacy in Internet of Things (IoT) environments is a real issue. Traditional security mechanisms use a non-aware approach, in which static parameters are used to provide secure decisions. IoT is a dynamic environment. Thus a non-static approach for security provision becomes mandatory. Context-aware security appears as a viable choice for this kind of processing. It uses the context information of IoT environments thus providing dynamic security. When together with context sharing feature, it can add new dimensions to the IoT security. Context sharing allows the use of off-domain context information to the security provision. This paper defines an Edge-Centric Context Sharing Architecture that provides context-aware security by using shared context information. Moreover, we discuss the challenges in the context-aware security area.**

***Keywords*—Context-Aware Security; Internet of Things; Context Sharing; Edge Computing; Reasoning.**

I. Introduction

The Internet of Things (IoT) computing paradigm embeds mobile networking and information processing capability into a wide array of gadgets and everyday items [1]. As miniaturization continues and computing capacity still increases, IoT devices are becoming more powerful. There is a common sense that IoT devices generate many data. The context-aware computing helps in interpret and understand these data in a proper way, producing context information [2].

The context information is considered any high-level information, sometimes semantic, that can be used to characterize the situation of an entity (e.g., person, place, or computing device). In most cases, the context information is stored individually by the systems. Context sharing is a feature that allows the systems to share context information to heterogeneous entities "understand" different context information across application domains [2].

The IoT is a dynamic environment in which entities are in constant change. In light of this, the traditional static security mechanisms become inadequate. The context-aware security (CAS) has added new dimensions to the old fashion security by using context information to provide security decisions [3].

There is a need for platforms that provide context-aware security to integrate different IoT verticals. In this sense, the main contributions of this paper are:

- A vision of both context-aware security and context sharing technologies, and how the shared context can be used to provide security. Moreover, the state-of-the-art in context-aware security is presented.
- The definition of an Edge-Centric Context Sharing Architecture able to make secure decisions based on shared context information.
- A discussion regarding which are the next steps for fostering the development of new context-aware security platforms.

The remainder of the paper is structured as follows: Section II presents some background concepts. Section III presents the related work. Section IV presents the proposed architecture. Section V discuss the next steps in context-aware security. Finally, Section I concludes the paper.

II. Context-Aware Security and Context Sharing

The context-aware technology brings completely new experience for the application and users. In most cases, securing the applications is done the old way. Traditionally, security requirements are assumed to be relatively static since security decisions do not change with context [4]. However, the use of context information to provide security decisions is a key feature to mitigate some security problems [5]. Moreover, the use of shared context information can integrate different verticals of IoT environments.

A. Context-Aware Security

The context-aware security (CAS) is defined by Mostéfaoui and Brézillon [6] [7] as: "*a set of information collected from the user's environment and the application environment and that is relevant to the security infrastructure of both the user and the application*." For example, while detecting an intrusion during communication, the security mechanism may adapt to strong authentication method [8].

The context-unaware mechanisms can be inadequate for the Internet of Things due to its dynamic and heterogeneous environment. The context information can be used to reconfigure security mechanisms and adjust security parameters.

Fig. 1 shows an example of a possible attack in an IoT environment and how CAS deals with this issue. Fig. 1-A shows a standard IoT application scenario that performs some decision making. For example, *WHEN* the temperature is bigger than 27°C *AND* the user location is defined as "Home", *THEN* a specific window of the house is open.

2324-9013/18/31.00 ©2018 IEEE
DOI 10.1109/TrustCom/BigDataSE.2018.00257

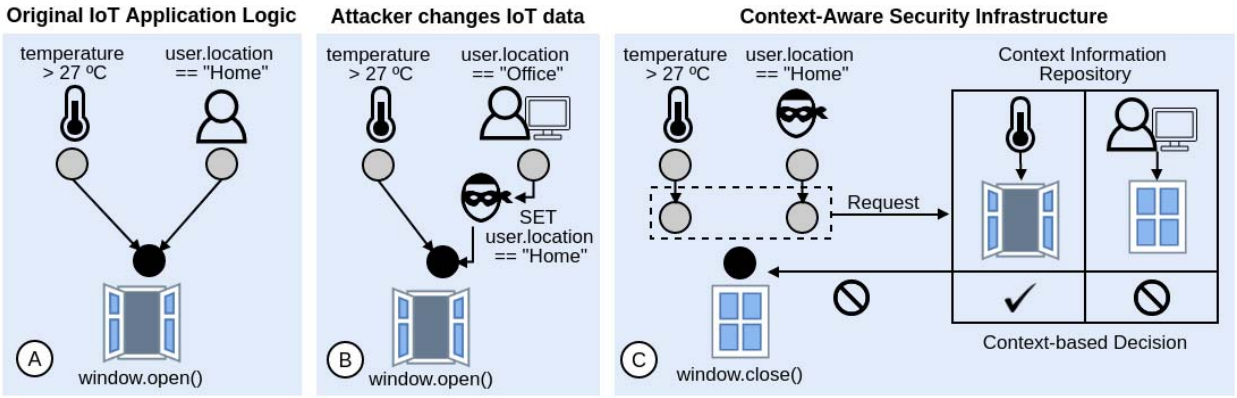

Fig. 1. Overview of Context-Aware Security in IoT environments.

In Fig. 1-B, the temperature continues the same, but the user location was changed. In this sense, an attacker may *SET* the user location to "Home" in order to open the house window (i.e., get access to a determined resource).

The CAS mechanism takes care of such issue as demonstrated in Fig. 1-C. Every time that a crucial security decision has to be made by the application, it may check the information with the Context Information Repository. It contains the last context information sensed (i.e., near real-time). Also, it has an history with the context sensed in a determined situation, for example, every day at this time the user tends to be at "Office," so it is unusual to his location be "Home." In this sense, for the example of Fig. 1-C, the window will not be opened.

To the implementation of CAS in IoT environments three main areas must be considered: (i) authentication, (ii) authorization and access control, and (iii) privacy-preserving. Next items present an overview of each area [4] [8] [5].

- **Authentication:** Traditional authentication methods require much user interaction in the form of manual log-ins, logouts, and file permissions. These manual interactions violate the vision of non-intrusive ubiquitous computing. Traditional security mechanisms are context-insensitive (i.e., they do not adapt their security policies to a changing context). Reliable authentication is an essential requirement for secure systems. Moreover, well-known technologies can be used for authentication, such as face recognition, iris scanner, and biometric technology. Besides these technologies, the use of context information strengthens the authentication process.
- **Authorization and Access Control:** Although different, these two areas are presented together since most approaches try to reach both. Many existing computer networks comply with "allow" and "deny" based access control policies. "Allow" means granting access when the user or device credential matches with pre-stored credentials and "deny" means blocking access when the user or device credential does not match with pre-stored credentials. This type of system can be considered static because it does not take into consideration other factors such as contextual information from the user or device environment while making allow and deny decisions. However, the IoT has a dynamic environment, where flexible security policies using contextual information can potentially increase the effectiveness of security decisions.
- **Privacy-Preserving:** Since information reflecting users' daily activities (e.g., travel routes, buying habits), it is considered by many users as private it would be no surprise that one of the requirements to ubiquitous applications would be privacy preservation. The context information can be used to determine when or not to keep user information private.

B. Context Sharing

The use of context sharing enables heterogeneous computational entities in pervasive computing environments to have a common set of concepts about context while interacting with one another. By reusing well-defined context of different domains, it is possible to compose large-scale context information without starting from scratch [13]. Besides of sharing context information, such platform can be used to reduce the processing effort of the entities, once they receive context information instead of reasoning for it.

Let's consider a scenario to clarify the vision of context sharing. In this scenario, the focus is on sharing the context of a home-care patient when some important events related to the health condition occurs. The context information acquired by home-care sensors triggers the event of a patient having a heart attack. This event (i.e., context information) must be shared with whom may be interested on it. An ambulance may receive this context to attend the patient.

The received context information can be used in new processing. For example, the context can be shared with the urban traffic infrastructure to drain the traffic with a "green wave" in traffic lights to decrease patient's waiting time. Moreover, the context information of the ambulance arriving at the patient's home is used to open the door to facilitate paramedics access, making a security decision (i.e., access control, authorization).

TABLE I
OVERVIEW OF CONTEXT-AWARE SECURITY SYSTEMS.

Systems	Authentication	Authorization	Access Control	Privacy-Preserving	Scope	Uses Shared Contex
[5]	No	Yes	Yes	No	User	No
[9]	Yes	Yes	Yes	Yes	WSN	Partial
[10]	Yes	No	Yes	No	Automotive	Partial
[11]	Yes	No	No	No	VANET	Partial
[12]	Yes	No	No	No	Mobile	No
Our work	**Yes**	**Yes**	**Yes**	**Yes**	**IoT environments**	**Full**

In the previous example, the context was shared within different IoT verticals, such as home-care, ambulances service, and traffic. Each vertical may use the shared context for different processing, as to provide context-aware security.

III. RELATED WORK

This section analyzes context-aware security systems based on main application areas: (i) authentication, (ii) authorization, (iii) access control, and (iv) privacy-preserving. Moreover, the scope of each work is presented and if it uses shared context information for CAS (i.e., if it performs a kind of context sharing).

Table I presents the comparison of analyzed works. For the use of shared context information issue, the analyzed works can be categorized into two groups: full heterogeneity, and partial heterogeneity. The ones of full heterogeneity comprise the systems which use context information of different sources and formats to provide security decision. The partial heterogeneity systems use context information of local or similar groups. Next paragraphs present the definitions of analyzed systems.

Trnka et al. [5] propose a solution that extends role-based access control (RBAC) with certain context awareness elements. It is based on using security levels, which are granted to user based on his context. Hosseinzadeh et al. [9] propose a Context-Aware Role Based Access Control (CARBAC) scheme. It controls the access of the users to the system following their role and the current context information.

Gansel et al. [10] solution is focused in automotive scenarios. They propose an access control model that is inherently aware of the context of the car and the applications. SVM-based Context-Aware Security Framework (SVM-CASE) [11] automatically determine the boundary between the misbehaving nodes and well-behaved nodes in VANETs.

Context-Aware Scalable Authentication (CASA) [12] makes authentication easier or harder based on some users parameters (e.g., a user's location or time since the last login) rather than making it uniformly hard for all cases.

Although all the analyzed works provide solutions related to context-aware security, they may differ in its architecture and how they provide security. Each work has its focus, one time that ones have the objective to protect the whole infrastructure, there are systems with a specific goal. By using context information to provide security, most of the systems were deployed for dynamic situations, where the location and status are a point.

The works [9], [10], and [11] use shared context information for the security decisions. However, the source of this context information is from the same or similar entities of the systems deployed many times in a near location. In this sense, they not reach full heterogeneity. To reach it, the system must use context information from heterogeneous sources deployed in different locations or networks. Although analyzed works provide context-aware security features in their architecture, they do not care about the heterogeneity of the IoT environments. A solution that considers this requirement and reaches a full heterogeneity is needed to mitigate the challenges of the area.

IV. EDGE-CENTRIC CONTEXT SHARING ARCHITECTURE

The importance of having a context sharing architecture is strongly related to the need of users and applications to share information between different places. Our proposed architecture performs context sharing feature. In light of this, it is possible to provide context-aware security across domains. To the best of our knowledge, a platform that provides context-aware security through context sharing was not deployed yet.

The proposed architecture takes benefit of fog and edge computing approaches to minimize network communications, thus reducing failure points and improving scalability. The definition of the architecture is illustrated in Fig. 2. It is composed of two main systems: Context Sharing and Context Provider. Context Sharing System is placed at the fog level and is responsible for sharing the context information with other entities, or Context Sharing Systems instances, including different fogs. Context Provider Systems are placed at the edge layer, embedded or connected directly to IoT devices, and are responsible for generating context information and making secure decisions (i.e., context-aware security). It also has a Cloud layer for storage and coordination. More context sharing details and architecture evaluation can be seen in our previous work [14].

Context should be a first-class security component in order to drive the behavior of IoT devices. This would allow smart objects to be enabled with context-aware security solutions, in order to make security decisions adaptive to the context in which transactions are performed. At the same time, context information should be managed by taking into account security and privacy considerations. In particular, current IoT devices (e.g. smartphones) can obtain context information from other entities of their surrounding environment, as well as to provide contextual data to other smart objects [3]. These

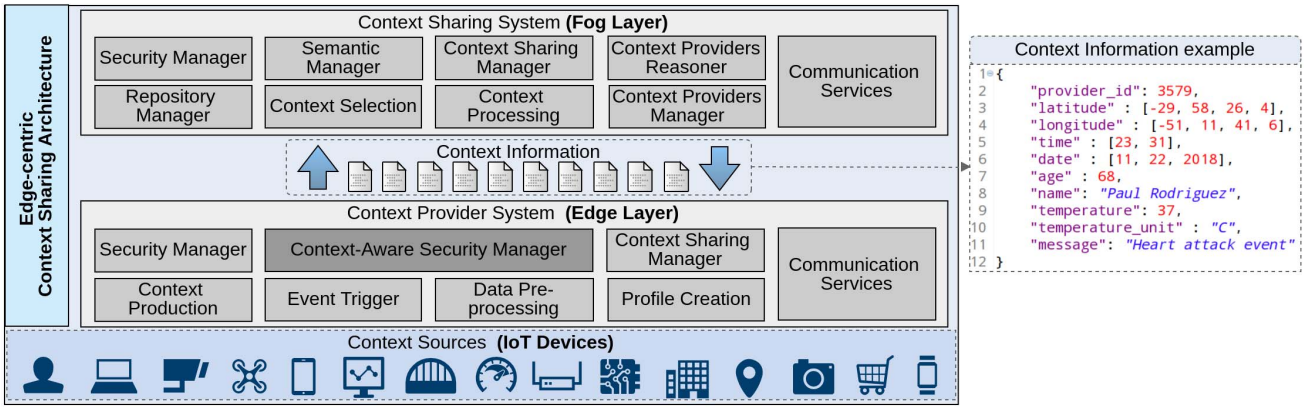

Fig. 2. Edge-Centric Context Sharing Architecture overview.

communications can be performed through lossy networks and constrained devices, which must be secured by suitable security mechanisms and protocols.

The IoT is composed of sensitive domains, such as healthcare, transportation, home-care, among others. It is highly significant to protect the privacy and confidentiality of personal data from unauthorized access while stored or transmitted. It is even more crucial and difficult to administrate the information and physical security in ubiquitous environments with numbers of participants continuously joining and leaving the space [9]. Moreover, high-level context information can be reasoned and inferred by considering privacy concerns. For example, a smartphone could be configured to giving the name of the city where the user is, but not the GPS coordinates [3].

The proposed solution for context-aware security (CAS) can provide authentication, authorization, access control, and privacy-preserving to fog and edge computing environments by using shared context information. Fig. 3 shows an overview of the proposed CAS solution, the Context-Aware Security Manager mechanism. The core operation to provide context-aware security is by using pre-defined security rules. These rules are mostly defined for a specific domain that the mechanism is deployed. The mechanism works as follows steps: (i) it receives the shared context information, (ii) matches the received context with the historical one, (iii) infers security decisions by the rules. There are some details of these steps that are defined in the next paragraphs.

- **Event Handler:** It is responsible for receiving the shared context information (i.e., new event) and analyze it alongside with ConSec Instance and Context Security Reasoner modules. The Context Analyzer module manages the context information. It can interpret the context information to extract any data (e.g., application vertical, type of data, source device) to help in the selection to which kind of rules the context information must be submitted. It also matches the context with a security operation type (e.g., authorization, access control, etc.) and the Output Action can be started (e.g., give access to an entity, change a status, acts on a device).
- **Context Security Reasoner:** This module is responsible for the reasoning process of the Context-Aware Security Manager. It is composed of Web Ontology Language, that is responsible for the classification/modeling of the context for the reasoning process. The Working Memory module fire the Lightweight Rules that infer possible security decisions. These rules scheme is defined via C Language Integrated Production System (CLIPS) rules. This module acts alongside the Security Rules of the ConSec Instance module.
- **ConSec Instance:** As reasoning process needs context information to match with the rules, it uses this module to query for it. This module is composed of two databases: (i) Context Information and (ii) Security Rules. The first one has a historical set of the context of past events. The second one is composed of pre-defined security directives (e.g., IF *contextA* AND *contextB* THEN *giveAccess*) that will be converted into rules by the Context Security Reasoner module. The Repository Manager helps in access the two databases and update it when needed.
- **Context Acquisition:** Context information has a short lifetime once the IoT is composed of devices that eventually move or change status. The primary function of this module is to get new context information when needed. It has the Context Production Interfaces to make easy the connection with the subsystems that will produce the context information. In this sense, it could perform a request, receive new context information from the Context Production (see Fig. 2), and update the Context Information database of the ConSec Instance module.

V. Next Steps in Context-Aware Security

There is a need for defining a horizontal context-aware security platform that goes beyond vertical solutions. Although the analyzed platforms provide some context-aware security feature in their architecture, none of them care about using shared context information. Some platforms were developed to a specific scenario. In this way, they lack in some features that a horizontal platform must have. It is essential to research,

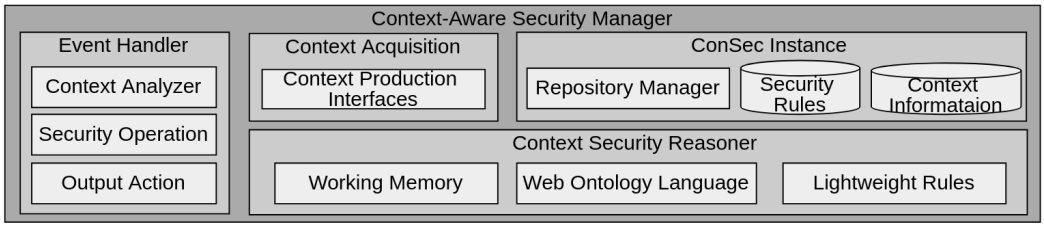

Fig. 3. Overview of the mechanism for Context-Aware Security from the Edge-centric Context Sharing Architecture.

study, and develop context-aware security towards the evolution and consolidation of the IoT. In this sense, there are some next steps that should be addressed.

1) Heterogeneity: It is a challenge in context-aware security area. Some platforms perform security decisions with a kind of shared context information. However, they usually do not care about the heterogeneous environments, which are very common in IoT. The integration of different domains is not envisioned in most platforms. Ontologies, web services, and lexical databases are examples of technology has been successfully used in the IoT to hide devices patterns [2] [15], which leads us to believe that those technology can be used in the IoT context-aware security area as well.

2) Fog and Edge Computing: Fog computing is a distributed paradigm that provides cloud-like services closer to the network edge. The adoption of a Fog/Edge approach is also related to the scalability and real-time sharing features. The use of Edge Computing concept, which is related with data processing at the IoT devices themselves may be a way of reducing the extra information exchanged. The adoption of the Edge Computing paradigm by context-aware security platforms minimizes the latency and network overhead. Also, the decentralization of Edge Computing enables the implementation of complex heterogeneous IoT environments.

3) Hybrid Reasoning: The hybrid approach of Fog and Edge Computing also enables a hybrid reasoning for the security decision. Recent works point that a hybrid and integrated reasoning approach would be beneficial for heterogeneous IoT environments [2]. Depending on the devices capabilities, the reasoning method could be different. For example, a constrained one can reason with lightweight rules, while a more capable device may reason with ontologies.

VI. Conclusions

With the IoT development, some traditional security methods started to become obsolete. Context-aware security can add new perspectives to the security and privacy area. However, context-aware security by itself is not enough to cope with the large heterogeneity of IoT. Context sharing appears as a way to provide context-aware security feature in heterogeneous environments. This paper presented an architecture that uses shared context information from different domains to provide security decisions. Moreover, we have shown the next steps in the context-aware security area and hope to promote the overall work towards the development of the field.

References

[1] L. Atzori, A. Iera, and G. Morabito, "The internet of things: A survey," *Computer Networks*, vol. 54, no. 15, pp. 2787–2805, 2010.

[2] C. Perera, A. Zaslavsky, P. Christen, and D. Georgakopoulos, "Context aware computing for the internet of things: A survey," *Communications Surveys Tutorials, IEEE*, vol. 16, no. 1, pp. 414–454, 2014.

[3] J. L. H. Ramos, J. B. Bernabe, and A. F. Skarmeta, "Managing context information for adaptive security in iot environments," in *2015 IEEE 29th International Conference on Advanced Information Networking and Applications Workshops*, March 2015, pp. 676–681.

[4] J. Al-Muhtadi, A. Ranganathan, R. Campbell, and M. D. Mickunas, "Cerberus: a context-aware security scheme for smart spaces," in *Proceedings of the First IEEE International Conference on Pervasive Computing and Communications, 2003. (PerCom 2003).*, March 2003, pp. 489–496.

[5] M. Trnka and T. Cerny, "On security level usage in context-aware role-based access control," in *Proceedings of the 31st Annual ACM Symposium on Applied Computing*, ser. SAC '16. New York, NY, USA: ACM, 2016, pp. 1192–1195.

[6] G. K. Mostefaoui and P. Brezillon, "Modeling context-based security policies with contextual graphs," in *IEEE Annual Conference on Pervasive Computing and Communications Workshops, 2004. Proceedings of the Second*, March 2004, pp. 28–32.

[7] P. Brezillon and G. K. Mostefaoui, "Context-based security policies: a new modeling approach," in *IEEE Annual Conference on Pervasive Computing and Communications Workshops, 2004. Proceedings of the Second*, March 2004, pp. 154–158.

[8] K. Habib and W. Leister, "Context-aware authentication for the internet of things," in *Eleventh International Conference on Autonomic and Autonomous Systems [Internet].[cited 2015 Dec 10]. Available from: Wolfgang Leister*, 2015, p. 6.

[9] S. Hosseinzadeh, S. Virtanen, N. Díaz-Rodríguez, and J. Lilius, "A semantic security framework and context-aware role-based access control ontology for smart spaces," in *Proceedings of the International Workshop on Semantic Big Data*, ser. SBD '16. New York, NY, USA: ACM, 2016, pp. 8:1–8:6.

[10] S. Gansel, S. Schnitzer, A. Gilbeau-Hammoud, V. Friesen, F. Dürr, K. Rothermel, C. Maihöfer, and U. Krämer, *Context-Aware Access Control in Novel Automotive HMI Systems*. Cham: Springer International Publishing, 2015, ch. 8, pp. 118–138.

[11] W. Li, A. Joshi, and T. Finin, "Svm-case: An svm-based context aware security framework for vehicular ad-hoc networks," in *2015 IEEE 82nd Vehicular Technology Conference (VTC2015-Fall)*, Sept 2015, pp. 1–5.

[12] E. Hayashi, S. Das, S. Amini, J. Hong, and I. Oakley, "Casa: Context-aware scalable authentication," in *Proceedings of the Ninth Symposium on Usable Privacy and Security*, ser. SOUPS '13. New York, NY, USA: ACM, 2013, pp. 3:1–3:10.

[13] F. Boavida, A. Kliem, T. Renner, J. Riekki, C. Jouvray, M. Jacovi, S. Ivanov, F. Guadagni, P. Gil, and A. Triviño, *People-Centric Internet of Things–Challenges, Approach, and Enabling Technologies*. Cham: Springer International Publishing, 2016, ch. 44, pp. 463–474.

[14] E. de Matos, R. T. Tiburski, L. Amaral, and F. Hessel, "Context interoperability for IoT through an edge-centric context sharing architecture," in *2018 IEEE Symposium on Computers and Communications (ISCC) (ISCC 2018)*, Natal, Brazil, Jun. 2018.

[15] Princeton University, ""About WordNet." WordNet. Princeton University." 2010. [Online]. Available: http://wordnet.princeton.edu