Securing Communication Between Service Providers and Road Side Units in a Connected Vehicle Infrastructure

Sami S. Albouq and Erik M. Fredericks
Oakland University
Rochester Hills, MI, USA
{ssalbouq,fredericks}@oakland.edu

***Abstract*—Road Side Units (RSUs) within the connected vehicle infrastructure are vulnerable to security and access control challenges. RSUs may share resources with unreliable SPs that could lead to information leakage due to an insecure messaging infrastructure. To mitigate these concerns, we introduce an automated service provision mechanism that enables a controlled messaging infrastructure using a Distributed Security Framework (DSF). Service provision is accomplished by providing RSUs with publish/subscribe brokers that enable authorized SPs to distribute their services as topics and define access rights through the DSF. The DSF acts as a secure middle layer that is hosted by fog computing nodes to ensure close proximity to RSUs, handles resource authorization (i.e., topic creation in specific brokers), and provides identity authentication of both RSUs and SPs. The DSF uses an attribute-based access control model to enable both SPs and RSUs to define and dynamically manage attribute-based access policies to cope with run-time changes of protection requirements. We validate the DSF framework in a simulated smart highway environment comprising interconnected RSUs and SPs to demonstrate our technique's effectiveness.**

I. Introduction

Road Side Units (RSUs) are fundamental components of the connected vehicle infrastructure. An RSU is a stationary device that is located at predefined positions on highways and streets to provide different types of operation (e.g., extends communication between vehicles and connects vehicles with backend services) [2]. Service Providers (SPs) can use RSUs to provision services for vehicles (e.g., weather alerts) and at the same time, other SPs can use these services to collect information for future predictions (e.g., traffic management systems) or other operations. For example, an advertisement (AD) service provider may be interested in vehicle traffic information that is provisioned by a traffic management service via RSUs to enable a large number of AD disseminations. However, these immediate communications may not be properly secured and intentionally or inadvertently reveal sensitive information to those that do not have proper authorization. In this paper, we present a technique for deploying, provisioning, and securing services over RSUs to mitigate such concerns.

It is generally difficult to regulate and secure communications in heterogeneous and distributed systems in connected vehicle (CV) environments where SPs from different domains (e.g., weather and healthcare) intend to use multiple types of RSUs to provision or consume services. To this end, most proposed solutions focus on securing RSU communications in relation to vehicles, while the interaction between RSUs and SPs have not yet received much attention [2][12]. The heterogeneity of provisioning services over RSU environments demand varying degrees of granularity for access control mechanisms and an inadequate or unreliable authorization mechanism can significantly increase the risk of unauthorized use of RSUs and services [15]. As a result, RSUs need a feasible framework that can manage and secure communication with SPs.

In this paper, we introduce the Distributed Security Framework (DSF), an independently administrated security manager that may delegate responsibility for service creation and policy specification. DSF can be deployed close the edge of the network and act as a security middle layer between deployed services and provisioned services over RSUs. DSF integrates authentication where both entities (i.e., SPs and RSUs) are verified against their claimed identities and SPs are evaluated based on a set of fine-grained access control policies in terms of identity, environment, and resource attributes, thereby ensuring that RSUs determine which authenticated SP is allowed access to their resources. Moreover, DSF provides SPs with a decentralized single-sign-on mechanism that enables persistent authorization.

DSF leverages fog computing to use virtualized computations near RSUs and attribute-based access control (ABAC) to enable policy specification. Moreover, the service provision exploits a publish/subscribe (pub/sub) model to implement a controlled messaging infrastructure [3]. The DSF and SPs use fog computing to deploy operation to nearby RSUs to enable computation, storage, and communication resources to be placed near the edge of network devices to mitigate latency challenges. The service provision adopts a pub/sub model as a distributed messaging infrastructure to create services over RSUs in the form of topics and provides pub/sub operations that are available for authorized and deployed SP services. DSF enables SPs and RSUs to use ABAC to provide dynamic access rights to their resources without prior knowledge of the requesting nodes. In this paper, we refer to deployed services as a set of computations placed in the fog computing nodes and service provision as a mechanism that provides availability

978-1-5386-1465-5/17/$31.00 ©2017 IEEE

and usability of deployed services via RSUs.

We demonstrate the use of the service deployment, provision, and DSF on a smart highway infrastructure that enables intercommunication between RSUs and SPs. Experimental results suggest that DSF can provide dynamic resource authorizations for both RSUs and SPs with minimal processing overhead. The remainder of this paper is organized as follows. Section II provides related work and background information on RSUs, fog computing, pub/sub, and ABAC. Section III describes the service deployment and DSF techniques. Section IV then describes our experimental setup and results. Lastly, Section V discusses our findings and presents future directions.

II. Background and Related Work

This section presents background material and related work on RSUs, fog computing, publish/subscribe, and ABAC.

A. Road Side Units

RSUs are stationary devices that can be installed inside a road side electronic cabinet or road side poles and are assumed to be equipped with storage, processors, and networking capabilities that enable communication to vehicles via the Dedicated Short Range Communication (DSRC) protocol. [14]. An RSU is a unit that can facilitate the communication between vehicles and SPs and other devices by transferring data over DSRC in accordance with the industry standards.

B. Fog Computing

Fog computing is an approach that extends the paradigm of cloud computing to the Internet of Things (IoT) by placing higher-power nodes between end-network devices and the cloud. The concept of fog computing was originally developed by Bonomi *et al.* as a virtualized platform that can provide services similar to the cloud and includes additional advantages such as proximity to consumers, dense geographic coverage, and mobility support [3]. Andrea *et al.* used fog computing to create the Rainbow platform that comprises multi-agent systems to provide services for smart city such as monitoring, managing and controlling devices remotely [5]. Salonikias *et al.* broadly described an access control technique that utilizes fog computing for Intelligent Transport Infrastructure (ITS) [13]. The goal is to access services provided by SaaS through RSUs, which include PDPs that connected to PIP in the cloud for access decision and policy retrievals. However, Salonikias's technique imposes privacy concerns toward requested services by vehicles and a clear access control structure.

C. Publish/Subscribe Mechanism

Pub/sub is message-oriented paradigm that leverages the producing and consuming concept to facilitate machine-to-machine communications [4]. Pub/sub decouples direct communications between nodes and allows communication via a broker that can handle common communication tasks, such as connecting, subscribing, and publishing. Most proposed techniques that used the pub/sub scheme in CVs focused on the communication between RSUs and vehicles [11] [8] [9]. Tulika *et al.* used pub/sub for information dissemination between RSUs and vehicles, where RSUs were interconnected to form a distributed hash-table-based broker [11]. Ilias *et al.* proposed techniques for message dissemination using topic, content, and the hybrid pub/sub between RSUs and vehicles [8] [9]. The design goal is to create a middleware for vehicular networks that consider location and time for notifications. However, previous techniques did not consider authorization and security issues in their designs, specifically topic ownership and operation rights on topics. Conversely, DSF acts as a framework to resolve unknown data provenance and possibly fake data dissemination.

D. Attribute-based Access Control

ABAC is a model for defining access control where access rights are granted to users through the use of policies that combine attributes [6]. Typically, policies comprise subject, object, and environment attributes that each specify a value, where the attributes are properties or capabilities that can be used for an access control decision process. The subject represents the entity requesting to perform an operation upon an object. The object a resource is an entity to be protected from unauthorized use. The environment is a dynamic factor that depends on the subject and object. The National Institute of Standards and Technology defined ABAC formally in 2014 to improve secure information sharing within organizations (i.e., healthcare) and between organizations (i.e., cloud federations) while maintaining control of that information [7]. For example, Shorouq *et al.* exploited ABAC in federated cloud to provide an identity and access management system where users are granted access to federated data when their identity attributes match the policies [1].

III. Approach

In this section, we describe our conceptual architectural model for service deployment and secure service provision in a CV infrastructure.

A. Secure Architectural Model Overview

Figure 1 presents our architectural model that comprises cloud computing, fog computing, edge of network, and CVs. Cloud computing is used for long term storage and computation, while fog computing is used for short term storage and computations beside service deployments and security management. The edge of network serves as a gateway and interface between the fog computing and CV layers to enable communication and service provisions between mobile and stationary nodes (e.g., vehicles and SPs). The advantage of our design enables SPs to place a set of computations in fog computing nodes for immediate and quick service provision through the edge of network devices. Thus, we introduce two types of fog communications, heavy and light as shown in Figure 1 (B1, B2), to avoid long-distance connection with cloud computing, allow hierarchy communication to the cloud, and enable operations near the edge of network devices.

In our model, we exploit light fog for service deployment and the pub/sub paradigm for service provision. To deploy services in fog computing, SPs need to encapsulate computations in light fog nodes that can provide operations and activities through the edge network devices (e.g, RSUs) as shown in Figure 1 (C2). Moreover, the encapsulated computations need to be configurable by heavy fog nodes to enable connections with cloud computing. RSUs use the pub/sub paradigm to serve as a decoupled messaging infrastructure and enable provisioning services via brokers. Vehicles and SPs can connect to deployed services through brokers that provision services as topics, which denote the subjects of the services. Whichever nodes are interested in a service subscribe to the associated topic to receive relevant information.

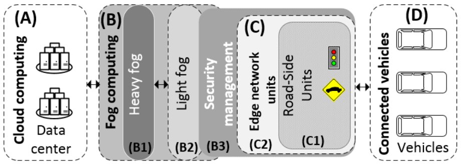


Fig. 1: An architectural model of communication in CV infrastructure

Assuming all nodes (i.e, SPs and vehicles) act appropriately is an increasingly risky assumption to make because nodes may misbehave, cause information leakage or initiate serious attacks. Therefore, we defined a new security management layer to be an independently administrated unit in which a *security manager* is responsible for controlling resources and sensitive data visibility (c.f., Figure 1 (B3)). In particular, a set of brokers is attached to a security management node (light-fog) that can handle security tasks, such as authentication, where vehicles and SP identities are verified via access control in which resources can be authorized to a specific node. Note this paper mainly focuses on the security aspect of the communication between SPs and RSUs.

B. Security Scheme

In this section, we describe the DSF approach for securing RSUs and service provisions. Nodes are required to register their identities to a Trusted Authority (TA) (e.g, governmental transportation authorities) before participating in order to guarantee authentication and enable secure communications. We assume that the TA is always trusted and can never be compromised, and can provide nodes with information used for identity verification to prevent impersonation and access control attacks. The TA makes sure attackers do not gain control of the network and protects sensitive data.

SPs must be authenticated and authorized to launch secure services through RSUs. The SPs need to send registration and connection requests to the RSU's *security manager* seeking the permission for providing service(s) (e.g, creating topics). The *security managers* expect only requests from the fog and edge of network layers and ignore any incoming requests from different layers (e.g, cloud computing) to ensure services are close to RSUs. If an SP sends a connection request directly to an RSU, the security agent (a guard agent that is responsible for identity and access control verification) evaluates the request against authentication and authorization. If the node is authenticated then the agent verifies the authorization which may depend on multiple criteria, such as the action that is being requested, the resource on which the action is being requested, and the groups to which the authenticated node belongs or the roles that the node plays. We next describe the main components and operations of this security scheme.

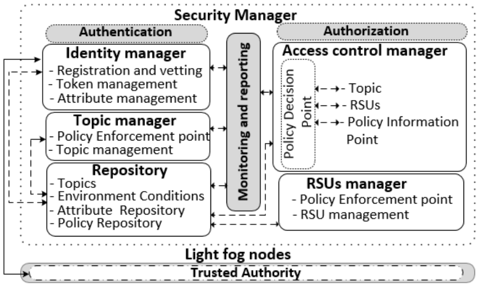


Fig. 2: Security manager components

1) Security Manager: The *security manager* is a light-fog node that is responsible for establishing secure communications between SPs and RSUs to effectively exchange trusted messages. Thus, a *security manager* can authenticate, authorize, monitor, and register nodes that are either intended to provide or use services. Figure 2 shows the main components of the *security manager* that are described next:

Identity Manger (IdM): The IdM is an entity that is responsible for verifying, negotiating, and registering SPs and RSUs. In addition, the IdM issues temporary identities for nodes that successfully satisfied the registration requirements. Figure 2 shows the main components of IdM. *Registration and vetting* is used to verify the identities and attributes provided by subject-nodes (RSU or SP) with the TA. The *attribute manager* can extract, store, update, and delete attributes of a specific subject. These attributes are properties or capabilities that can be used for the access control decision process. The *token manager* handles token generating, renewing, and revoking, where a token is a temporary secret identity that can be used for authentication. Hence, every subject needs to send a registration request that is digitally signed and encrypted to allow the *Registration and vetting* processes the request and verifies the content with the TA.

Topic and RSU managers: These managers provide authorization, configuration, and creation capabilities for topic and RSU information, respectively. Both entities have *Policy Enforcement Point* (PEP) components that serve as gatekeepers to the resources that each manager intends to access. PEP will intercept a subject node access request (i.e., create, delete, or modify) to a resource, create a decision request to the *Access control manager* to obtain an access decision (i.e. permit or deny), and act on the received decision. Note when the PEP sends the evaluation request it needs to include the subject's identity, target resource, and the operation type in order to avoid denial of access.

The *RSU manager* is a component that can interact with both RSUs or RSU owners to perform different operations such as add, edit, and remove policies. These policies are stored in the repository storage and are used for access control decisions. In addition, the *RSU manager* allows RSUs modify their attribute values automatically and dynamically change the access control.

The *Topic manager* is a component that allows SPs to deploy services with authorization rights. These services can be created in topics based over RSUs and associated with stored authorization policies. The initial step of service provision over a particular RSU involves gaining permission from the Access Control Manager (ACM) that the particular RSU accepts hosting the service.

RSU owners and SPs can define a set of access control policies that specify which topics or RSUs are authorized access based on subject's attributes. RSU owners and SPs can formally define their access control as a conjunction of attribute conditions $\{cond_1 \wedge ... \wedge cond_n \}$. Each attribute condition (*cond*) is in the form of $\langle\ name_a,\ \sigma,\ v\ \rangle$, where $name_a$ is the name of the attribute a, σ is a comparison operator such as $=$, $<$, $\leq$, $\geq$, $>$, etc., and v is the value of attribute a. RSU owners and SPs then send the resource information and the policies to the (*RSU manager* and *Topic manager*) to store them in the repository.

Access control manger: ACM is a component for processing access resource requests. The ACM comprises two main entities: *Policy information point* (PIP) and *Policy decision point* (PDP). The PIP serves to retrieve attributes or data required for policy evaluation to provide the information needed by the PDP. The PDP computes access decisions based on the available information and the applicable policies.

Monitoring and reporting: This entity continually monitors DSF operations that provide environment, policy, and information changes during the running time. In addition, it can participate in revoking tokens and granting access to specific resources. For example, when an SP provisions a service over an RSU, the SP will provide DSF with a set of policies for service authorization. These policies may get changed by the SP over time and invalidate existing authorizations.

2) Brokers over RSUs: We assume RSUs are trusted nodes, cannot be compromised, and install pub/sub brokers that provide several operations: connect, disconnect, publish, subscribe, and unsubscribe. These operations are available for nodes (e.g., SPs) that are authorized by the security agents of the brokers. The security agents are entities that intercept each request sent to the brokers for security evaluation, such as verifying the requestor identities and checking the access resource permissions with the *security manager*.

Topic Creation and Publish/Subscribe: When an SP is granted permission to create a topic over a specific RSU for a service provision, it only needs to send a topic creation request that includes the SP token and the topic name. The *security agent* will intercept and communicate with the *security manager* to check the permission of the request. The broker then creates the topic and associates it with the SP identity to ensure provenance of data.

C. Experiment Setup

In this section, we present our experimental setup that was used to demonstrate the effectiveness of the service deployments and provisions along with the DSF. For this experiment, we simulated a smart-highway interconnected infrastructure model, where RSUs were independently deployed in predefined locations to act as nodes covering segments on the highway and enable interactions with authorized deployed services for service provisions. In our simulation, we selected the highway to be a length of $10km$ that was divided into segments equal to the number of RSUs. We chose the length of the highway to be long enough to simulate real-life scenarios, where each RSU's region is represented by a zip code area used for service deployments.

Furthermore, we deployed local nodes that represented both the DSF and SPs and off-site virtual nodes to act as RSUs. This setup enabled us to simulate the communication of the fog computing environment. In particular, we deployed 8 virtual machines (VMs) that represented RSUs that enabled different interactions with SPs over provisioned services. Each VM included our modified version of the Mosquitto broker to provide a secure pub/sub messaging infrastructure for service provision [10]. The SPs and RSUs were assigned 16 attributes that defined their identities and operations (e.g., is-active, type-of-service, and zip code); The chosen number of assigned attributes enabled enough random policy generations for resource protections based on empirical evidence. Furthermore, each attribute was assigned a value selected from predefined lists to increase the chance of attribute similarities and enable resource authorizations.

IV. Experimental Results

We next describe our experiment set up and present our experimental results from applying the DSF to a simulated smart highway environment. Each simulation runs with a fixed number of RSUs and a different combination of predefined parameters for SPs, attributes, topics, and policies. We initially performed 190 treatments to achieve statistical significance. The total number treatments were divided into 3 sets based on the main operations of the proposed technique (i.e., 160 pub/sub and 30 authentications) to perform independent measurement. In the first treatment set, the authentication phase, we conducted 3 experiments where each was repeated 10 times to calculate the performance average of each set and evaluate messaging delay. Every experiment was initialized with 50, 100, and 150 SPs that each had 16 attributes to find the increment of the computation delay.

The other two treatment sets included the pub/sub simulations. These treatment sets were performed independently (e.g., publish) and was divided into 2 groups based on the policy attribute number that was selected for each topic from the SP attributes. Moreover, every group was run with a different number of attributes, 5 and 10, to study the effect of increasing the number of attributes in relation to the number of requests. Thus, we defined a fixed number of generated policies equal (i.e., 30) to enable different policy selections for each topic. Every treatment set was repeated 4 times with a different number of requests (25, 50, 75, and 100) to find the increment average delay and was randomly assigned to

150 SPs to ensure requests were sent from different nodes. Finally, each SP could select topics from a predefined matrix with 109 entries that enable different topic creations.

A. *Experimental Results*

In this section, we discuss the results from applying DSF to the service deployment in the simulation.

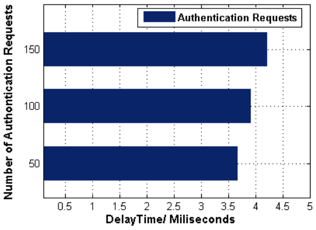
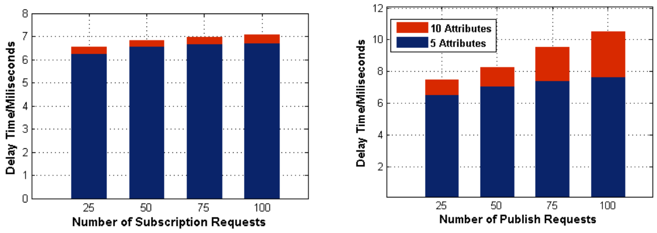


Fig. 3: Performance of authentication

Figure 3 presents the result of the performance measurements of the authentication phase. The figure shows that when the number of requests is 50, the average performance delay is nearly 3.65 *ms*. However, the average performance increased slightly to become almost 4 *ms* when the number of requests is doubled to be 100. This indicates that the average performance increased 0.25 *ms* due the increased of the number of requests, resulting in a long wait for request completions. Similarly, when the number of authentication request is tripled, the average performance becomes 4.4 *ms* indicating that an increment of nearly 0.8 *ms* for the extra 100 requests.



(a) Subscribe

(b) Publish

Fig. 4: Performance of publish and subscribe operations

Figure 4a and 4b show the results of the subscribe and publish operations. The subscribe average performance results of the 25 - 100 requests with 5 attributes policy selection indicate a minimal increment in term of the performance delay. Similarly, when the number of attributes was increased to 10 for the policy selection, the increase in delay minimal. In contrast, the publish operation results show a notable difference when the number of attributes and request increased. For example, when the number of requests is 100 and the number of attributes are 5, the average performance is about 7.8 *ms*, while when the number of attributes is 10 the average performance is 10.2 *ms*. This result is due to the processing time to acknowledge the publication and the authentication process through DSF.

V. Conclusion and Future Work

In this paper, we presented an inial concept for secure service provision over RSUs. The service provision was carried out by providing each RSU with a copy of a pub/sub broker that enabled a controlled messaging infrastructure through the DSF framework, where DSF is a secure middle layer that is hosted in fog computing nodes to ensure close proximity to RSUs. Furthermore, DSF exploits the ABAC model to provide dynamic access rights, where policies are dynamically defined via attributes. DSF enables authorization, authentication, and monitoring of both RSUs and SPs. Finally, we demonstrated the validity of DSF with a simulation of a smart highway infrastructure network.

Future work includes extending DSF to accept requests (i.e., a topic subscription) from vehicles with privacy considerations, specifically hidden identity. Moreover, we intend to formalize DSF operations.

VI. Acknowledgment

This work has been supported in part by NSF grant CNS-1657061, Islamic University in Medina, and Oakland University. Any opinion, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of supported by the NSF agent, Islamic University in Medina, and Oakland University or other research sponsors.

References

[1] S. Alansari, F. Paci, and V. Sassone. A distributed access control system for cloud federations. In *2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS)*, pages 2131–2136, June 2017.

[2] Sami. Saad. Albouq and Erik. M. Fredericks. Lightweight detection and isolation of black hole attacks in connected vehicles. In *2017 IEEE 37th International Conference on Distributed Computing Systems Workshops (ICDCSW)*, pages 97–104, June 2017.

[3] Flavio Bonomi, Rodolfo Milito, Jiang Zhu, and Sateesh Addepalli. Fog computing and its role in the internet of things. In *Proceedings of the first edition of the MCC workshop on Mobile cloud computing*, pages 13–16. ACM, 2012.

[4] Patrick Th Eugster, Pascal A Felber, Rachid Guerraoui, and Anne-Marie Kermarrec. The many faces of publish/subscribe. *ACM computing surveys*, 35(2):114–131, 2003.

[5] Andrea Giordano, Giandomenico Spezzano, and Andrea Vinci. *Smart Agents and Fog Computing for Smart City Applications*, pages 137–146. Springer International Publishing, Cham, 2016.

[6] V. C. Hu, D. R. Kuhn, and D. F. Ferraiolo. Attribute-based access control. *Computer*, 48(2):85–88, Feb 2015.

[7] Vincent C Hu, David Ferraiolo, Rick Kuhn, Arthur R Friedman, Alan J Lang, Margaret M Cogdell, Adam Schnitzer, Kenneth Sandlin, Robert Miller, Karen Scarfone, et al. Guide to attribute based access control (abac) definition and considerations (draft). *NIST special publication*, 800(162), 2013.

[8] Ilias Leontiadis. Publish/subscribe notification middleware for vehicular networks. In *Proceedings of the 4th on Middleware Doctoral Symposium*, MDS '07, pages 12:1–12:6, New York, NY, USA, 2007. ACM.

[9] Ilias Leontiadis, Paolo Costa, and Cecilia Mascolo. A hybrid approach for content-based publish/subscribe in vehicular networks. *Pervasive and Mobile Computing*, 5(6):697 – 713, 2009. PerCom 2009.

[10] R Light. Mosquitto-an open source mqtt v3. 1 broker. *URL: http://mosquitto. org*, 2013.

[11] Tulika Pandey, Deepak Garg, and Manoj Madhav Gore. Publish/subscribe based information dissemination over vanet utilizing dht. *Frontiers of Computer Science*, 6(6):713–724, Dec 2012.

[12] M. H. Park, G. P. Gwon, S. W. Seo, and H. Y. Jeong. Rsu-based distributed key management (rdkm) for secure vehicular multicast communications. *IEEE Journal on Selected Areas in Communications*, 29(3):644–658, March 2011.

[13] Stavros Salonikias, Ioannis Mavridis, and Dimitris Gritzalis. *Access Control Issues in Utilizing Fog Computing for Transport Infrastructure*, pages 15–26. Springer International Publishing, Cham, 2016.

[14] Qing Xu, Tony Mak, Jeff Ko, and Raja Sengupta. Vehicle-to-vehicle safety messaging in dsrc. In *Proceedings of the 1st ACM International Workshop on Vehicular Ad Hoc Networks*, VANET '04, pages 19–28, New York, NY, USA, 2004. ACM.

[15] Yang Zhang, Jing Zhao, and Guohong Cao. Service scheduling of vehicle-roadside data access. *Mobile Networks and Applications*, 15(1):83–96, 2010.