Using Semantic Technologies to Mine Vehicular Context for Security

Sandeep Nair Narayanan, Sudip Mittal & Anupam Joshi
{sand7, smittal1, joshi}@umbc.edu
University of Maryland, Baltimore County, Baltimore, MD 21250, USA

***Abstract*—The number of sensors, actuators and electronic control units present in cars have increased in the last few years. The Internet-of-Things (IoT) model has transformed modern vehicles into a co-engineered interacting network of physical and computational components. Vehicles have become a complex cyber-physical system where context detection has become a challenge. In this paper, we present a rule based approach for context detection in vehicles. We also discuss various attack surfaces and vulnerabilities in vehicular IoT. We propose a system which collects data from the CAN bus and uses it to generate SWRL rules. We then reason over these rules to mine vehicular context. We also showcase a few use-cases as examples where our system can detect if a vehicle is in an unsafe/anomalous state.**

Keywords*—*Internet of Things, Semantic Web, Context Mining, Cyber-Physical Systems, Vehicular Security.

I. Introduction

Cars have become an ubiquitous mode of transportation and 'connected cars' using leading technologies like, advanced wireless communications, on-board computer processing, advanced vehicle-sensors, GPS navigation, smart infrastructure, and others are transforming the way we travel[1]. To realize this dream of connected ubiquitous cars ongoing research needs to develop a networked environment supporting very high speed transactions among vehicles (V2V), and between vehicles and infrastructure components (V2I) or hand held devices (V2D) to enable numerous safety and mobility applications.

However, this network environment needs to be able to determine the 'context of an individual car' in real time so as to ensure passenger safety and system synchronization. In the presence of a malicious activity, there will be a deviation from the normal context of the vehicle. For example, if we sense that "it is dark", "vehicle is moving" and "headlights are off", then the overall context is unsafe. In this paper we describe a semantic web based approach to mine such 'vehicular context' in real time, which can be used to create connected cars for the future.

In research, context had been defined by multiple researchers. Among them a popular definition is "*Context is any information that can be used to characterize the situation of an entity. An entity is a person, place or object that is considered relevant to the interaction between a user and an application, including the user and applications.*" [1] In the vehicular domain the entities include hardware devices like sensors, control units, actuators, wireless devices etc. and the different information which can characterize the situation include the sensor measurements, actuator states etc. Some of the sensory measurements can include speed, acceleration, location of the vehicle, distance proximity of the vehicle from another object, etc. and the states of actuators can include state of headlights is "on", state of wipers is "off", etc.

So as to determine vehicular context, we consider vehicles as an Internet-of-Things (IoT) model with various inbuilt Electronic Control Units (ECU), Sensors and Actuators. Some general purpose control units present in a modern vehicle are Anti-lock Brake System (ABS), Adaptive Cruise control, Active Suspension, Active Vibration Control, Entertainment System, Lane Keeping Assist, Electronic Power Steering, Adaptive Front lighting, etc. These ECUs get input from different sensors and perform various mechanical actions using actuators. Apart from doing their own functions, ECUs communicate between each other so as to efficiently perform their functions. To ease inter-controller communication, a common internal communication bus was introduced. Bosch proposed the controller area network (CAN bus) and the updated specification, CAN 2.0, was published in the year 1991. Since then various updates have been made to the CAN bus model with more recent ones being introduced in the year 1993.

In modern vehicles there are different sensors, actuators and control units communicating between each other over the common broadcast based CAN bus. As a result of lack of authentication mechanisms [16], if attackers can introduce any well formatted CAN messages on the CAN bus, it will be considered valid by default. Previous research demonstrated various attacks possible on vehicles. Compromising the system using malicious modification of ECU is described by Hoppe et al. [15]. The demonstrated attacks include manipulation of car windows of a running car [14] and controlling the vehicles warning lights. They were able to manipulate even Airbag control systems using similar techniques. But these techniques required the modification of ECU, which is considered a bit hard for a normal attacker. The next category of attacks is done by the introduction of an external device on the vehicle. Charlie Miller and Chris Valasek demonstrated how to control a vehicle by attaching an external device on the OBD-II port of a vehicle [18]. But it became popular when the same researchers were able to control a Jeep Cherokee [19] remotely with factory settings. In the demonstration, they were able to disable the vehicle by exploiting a bug in one of its systems. It appears that this is not a one off incident, since reports claim that [24] many popular car makers like Volkswagen, Skoda, Volvo, etc. are vulnerable to a crypto attack on keyless entry which allows unauthorized access to these vehicles.

In the past few years the average number of sensors per

[1] http://ops.fhwa.dot.gov/travelinfo/infostructure/aboutinfo.htm

car have increased from 24 in 2002 to 70 in 2013[2]. With the introduction of smart cars and self driving cars this number is bound to increase further. Semantic contextual inference can offer more precision and maintainability over other methods in such situations. Using various semantic web technologies like Ontologies and RDF, we can represent each of these sensors separately and use a reasoner to identify any inconsistencies. Same context could be determined from different sets of sensors. Such redundancies could be utilized to verify the context of the vehicle or otherwise to detect anomalies. For example, we can detect that the vehicle is moving from the speed sensors or by the combination of different other sensors like engine rpm and gear position of the vehicle. If both of them leads to the inference that the vehicle is moving, then it is a normal scenario. But if one infers the vehicle is moving and the other set contradicts, it represents a potential anomaly.

In our system, we collect streaming data from a vehicle's CAN bus and use this data against a set of SWRL rules[3]. These rules are extracted from historical system data to mine vehicular context. We also discuss a few use-cases where we derive vehicular context using these SWRL rules and use them to detect anomalous states. The rest of the paper is organized as follows. Section II provides a brief review of related literature followed by Section III which describes the attack surface of a vehicle. Section IV describes the proposed architecture while Section V shows how extracted context can be used for detecting anomalies. Then Section VI concludes the paper by discussing future research.

II. Literature Review

Vehicular system uses CAN bus for communication between different systems. Since CAN bus is built for speed and efficiency, it lacks various authentication features. But the introduction of modern communication features introduced different vulnerabilities. Wolf et al. [26] looked at the possibility of securing the system by adding cryptographic techniques on ECU's. A stream of research focus on developing better secure protocols for communication between components connected on the common bus. LCAP [13] is a light weight modification of the CAN protocol which introduced authentication primitives with relatively lesser communication overhead and computational complexity of using strong cryptographic primitives. Their technique is based on a magic number which is a one-way hash function attached to the extended identifier field. Another proposed protocol with authentication is CANAuth [25]. In their paper they identified the different constraints on the CAN bus system including real-time constraints, poor bi-directional communication, restrictions on message length etc. But one of the attractions of their protocol is its backward compatibility. They achieved this property by sending the authentication data out of band. Yet another proposed protocol is LiBrA-CAN [11] which focused on authentication primitives for a group of devices than focusing on single devices using key splitting and MAC mixing. They also proposed variations of their technique with master oriented and distributed schemes.

Most of these techniques requires modifications at hardware level to implement, which makes these techniques feasible only for new vehicles. But, there is an immediate requirement to secure vehicles already on the road. The importance of detection mechanism over prevention mechanisms is discussed by Koscher et al. [6] in their work. Unlike prevention techniques which requires extensive hardware level modifications, 'detect and mitigate' could be applicable to new and old vehicles alike. Ruta et al. [23] extracted the data flowing on the CAN bus using the OBD-II port and combined this with external information like weather information and location information using data fusion algorithms. They were able to infer road and traffic conditions, driving behaviors etc. and then give suggestions to the driver using their technique. OBD_SecAlert [20] is another attempt to detect malicious activities by analyzing data flowing over the CAN bus. It used Hidden Markov Models to identify malicious behaviors. It has now become very popular to attach a device on the OBD-II port and detect driving behaviors, vehicle health monitoring etc., since different car insurance companies like Geico and Progressive releasing gadgets and apps to monitor them. They are even promising discounts on insurance rates based on these analyzed data.

One of the issues with using statistical and machine learning techniques alone in vehicular settings is the lack of precision. Hence in this paper we try to use context awareness to identify potential malicious activities. We try to detect malicious activities by inferring context from the data on the common communication bus. Research in the field of context awareness existed from early 90's along with ubiquitous computing and pervasive computing systems. Now similar techniques have been extended for Internet of Things also. One of the earlier work in this area is Context Toolkit [8]. It has a context widget, interpreter and aggregator abstractions for the development and deployment of applications. Several extensions to it like Enactor [9] and Intelligeibilty Toolkit [17] are also available as further enhancements to it. Another middleware based solution based on ontology for context-aware computing is Service Oriented Context-Aware Middle-ware (SOCAM) [12]. It uses OWL for representation, reasoning and classification. Yet another service based middleware which uses ontology is MoCA [22]. It is used to handle context in a highly heterogeneous mobile computing environment. Devaraju et al. [7] did an in-depth evaluation of context sensing solutions and proposed another middleware solution, Context-aware Service Platform (CaSP), for mobile environments. Another work which focus on collection and aggregation of sensed information from multiple sources is Sensor Information Management [2] (SIM). It is mainly developed for smart home environments.

Most of these solutions were designed to be distributed so that it is highly scalable. But as far as vehicles are concerned, we have only devices in hundreds and not in millions. But we are concerned about the velocity at which the different events happen and the speed at which the context inference should happen. To addresses this, we try to aggregate the local contexts using an additional layer which is explained in detail in the section IV.

[2]http://cvrr.ucsd.edu/ece156/AutomotiveSensors-Review-IEEESensors2008.pdf

[3]https://www.w3.org/Submission/SWRL/

III. Attack Surfaces in Vehicular IoT

CAN bus is a broadcast based common bus which provides fast and efficient interconnection channel between different systems in a vehicle like Anti-Lock Braking System, Adaptive Cruise Control, Automatic Lane-Assist and Adaptive Lighting Control. Due to lack of strong authentication primitives attached with CAN bus, any message on the CAN bus, in valid format, is a valid message. The attack surface describes all of the different points where an attacker could get into the system and where they could get data out[4] . The different actors in a vehicular set-up are driver, passenger, mechanic and external entities. We categorize the attack surface as *Internal Hardware Attack Surface, On-Vehicle Device Attack Surface, External Hardware Attack Surface* and *Wireless Attack Surface*.

- **Internal Hardware Attack Surface:** As described above, due to lack of authentication primitives any malignant internal hardware can be a potential attack surface. Normally the mechanic or any other external entities who can get physical access to the vehicle can exploit this attack surface. They can do it by reprogramming the different Electronic Control Units (ECUs) which are responsible for aggregating and controlling different associated systems present in a vehicle. For example, a different ECU might exist for different systems like, Anti-lock Braking System, Adaptive Cruise Control, Adaptive Lighting Controls etc. Another potential way to exploit this attack surface is by replacing the original ECU with a malignant one.
- **On-Vehicle Device Attack Surface:** There are different devices which are accessible to driver and passenger actors. Driver has different controls like steering, brakes, steering mounted controls (audio systems, cruise controls, paddle shift gears), etc. The passengers also have access to systems like audio system, climate controls systems, etc. Most of these devices have only hardware controls. But nowadays they are replaced or added with software controls like touch screen displays, mobile phone apps etc. to control them. Some of the ways in which various actors can manipulate the vehicle is by attaching a USB drive to the audio system, or by introducing a malicious input into the onscreen displays or remote control units associated with different systems.
- **External Hardware Attack Surface:** The trend of attaching external devices on to a vehicular system is becoming popular. Different insurance companies in the US like Progressive or wireless providers like Verizon have devices which could be attached to the OBD-II port of the vehicle, which is directly connected to the CAN bus. The insurance companies use this to analyze driver behavior so that they can give discounts, while some other devices help users to locate the vehicle in a parking lot, remote start the vehicle or track a vehicle in case of a theft. Most of these devices are driven by software which need not be devoid of bugs and could be exploited by malicious hackers. There are some devices available in the market which just need to be placed on top of the CAN bus to tap the information from it. For example, CANCrocodile from WagenControl is a contact less CAN bus reader which allows to get data from CAN bus without interfering with the physical CAN bus.
- **Wireless Attack Surface:** With the advent of wireless communication revolution, different communication technologies like Blue-tooth, Wi-Fi, Mobile networks, Satellite radio, GPS network, RF sensors, etc. are also made available in the vehicle. For example, Bluetooth is used to connect to the audio systems or OBD-II attached devices. Multiple Android and IOS apps are available now which can communicate with these devices. For connected vehicles, even Internet is made available. Mobile applications are made available by many manufacturers for various smart cars which can be controlled using mobile devices. Another potential access point is through the RF devices used for central locking systems. All of these systems present a new potential attack surface, the attackers can try to exploit.

[4]https://www.owasp.org/index.php/Attack_Surface_Analysis_Cheat_Sheet

IV. Proposed Architecture

A typical vehicle has various sensors and actuators communicating over a common communication channel. Due to the primary focus on efficiency and simplicity, when the protocols were developed, it lacked required authentication features. As a result, any message available on the CAN bus is considered to be a valid instruction and the receiving control units and actuators have no way to distinguish it from a malicious instruction. In this paper, we try to aggregate and extract context from these different data exchanges to distinguish normal data flow from a malicious data flow.

In a vehicle the communication happens over the common CAN bus. We can tap the data flowing on the bus using the OBD-II (On-Board Diagnostic) port which is mandatory on all the vehicles in the US from late 1990's. The data flowing on the bus can use multiple protocols like ISO 15765-4 (CAN), ISO 11898 (raw CAN), SAE J1939, etc. Apart from this heterogeneity, the amount of data available on the common communication medium also prompted us propose a multi-tiered mechanism to extract context. The different layers are depicted in Figure 1. The Local Context Detection (LCD) layer will convert the crude and heterogeneous data into a higher plane for further context aggregation. The Cross Component Context Inferencing Engine (C3IE) aggregates the different local contexts from LCD to infer overall state of the system. We also propose the use of a Rule Mining Engine to extract knowledge from historical data to support the inferencing process. Each of the components are described in detail below.

A. Local Context Detection

The data from the OBD-II port is a stream of bits which is first converted into valid CAN messages. The Local Context Detection (LCD) layer's first job is to extract such valid messages from the stream of bytes. The CAN messages can contain raw sensor measurements like temperature, speed, acceleration, throttle pedal position, etc. In order to make the inference process faster and meaningful, the LCD layer again process these sensor messages into a higher context plane. For

example, the speed sensor emits the raw speed measurements at regular intervals. Instead of introducing the raw values as it is into the knowledge base, the LCD layer will process them and extracts more useful context like "high speed", "sudden acceleration", "normal speed", etc. and associates them with their respective entities in the Ontology.

Another important reason for having the LCD layer is to ensure portability of our system to different vehicles. Since the communication channels are not completely standardized and different manufacturers still use proprietary protocols, the LCD layer allows for the reuse of same rules and inference procedures. The main functions of this layer include conversion of raw bits into CAN messages and aggregation of similar CAN messages to generate more meaningful local context. Once the local context is inferred, the LCD layer pushes it into the C3IE's knowledge base.

B. Cross Component Context Inference Engine

The LCD layer extracts the local contexts from different sensors and actuators from the common communication bus and deliver them to the Cross Component Context Inference Engine (C3IE). The four different components in C3IE are IoT Ontology which captures the logical semantics between different components, Domain Instances which are the specific instantiations of entities in the Ontology, Domain specific SWRL rules which captures the relationships between different components and a Reasoner which infers the current state of the whole system. The different components are explained in detail in the following subsections.

1) IoT Ontology: One of the benefits of developing an Ontology is knowledge reuse. Hence instead of developing a new ontology, we re-use the IoT-Lite [5] meta-Ontology, which is a light weight version of SSN [6] (Semantic Sensor Network) ontology. In order to match our specific use-case, we extend it by adding some properties. IoT-lite ontology is developed so as to allow cheaper processing time while querying it. It classifies the IoT devices as Sensing devices, Tag devices and Actuator devices. Each of the devices are associated with an attribute which is a measurable quantity and one or more devices are associated with it. We extend this ontology by associating object properties like "senseAttribute" and "affectsAttribute" to the devices such that we can keep track of the devices which are related to different attributes of the system. Also we add a "hasStateValue" data property to the attributes of the system.

2) Domain Instances & Domain Specific SWRL rules: This is the part which customizes the ontology with domain specific knowledge. In the domain of vehicles, the different sensors may include "vehiclespeedsensor", "environmentlightsensor", "distacesensor" (which may be acoustic based, radar based or image based) and actuators like brake, accelerator, headlights and step-up motors. Now we need to add domain specific rules for the reasoner to fuse local context information from different sensors, actuators, etc. and generate a global context. We use Semantic Web Rule Language (SWRL) to represent such rules in the Ontology. Some of the sample SWRL rules are specified in section V.

[5]https://www.w3.org/Submission/2015/SUBM-iot-lite-20151126/

[6]https://www.w3.org/2005/Incubator/ssn/ssnx/ssn

3) Reasoner: In our Ontology, we define a special attribute called "currentsystemstate", which can have two values "normal" and "anomalous". It should be noted that our proposed technique is not only supposed to detect activities of malicious attackers, it also detects any inconsistent, dangerous or unexpected behavior of the system. For example, a sudden large change in the steering wheel angle, while the vehicle is moving at fast speed ranges is a dangerous behavior, caused by a malicious attacker or not. Sometimes it can be caused by a faulty component, but that doesn't stop the vehicle to be in a dangerous situation. Hence our proposed technique detects not only an attack state, but all dangerous or unsafe states. In the vehicular domain, we cannot use a static reasoner since the facts keep on changing. Hence we need to use reasoners like C-SPARQL [3, 4], IMaRS [10](Incremental Materialization for RDF Streams), TrOWL [21], C-SPARQL on S4 [5] or Streaming SPQRQL [5] which can work on the stream of data rather than on static set of facts.

C. Historic Data Aggregation & Rule mining

The presence of domain knowledge stored as SWRL rules is an important component of our system. The better information which could be captured with them, the better performance the overall system will have. Hence, we introduce the rule mining component, which uses statistical and machine learning techniques which generate these rules automatically. Once generated, the rules are ran against the already present rules and are tested for consistency before being added into the SWRL rules.

V. Rule Based Context Extraction Use-case

Context-aware computing has been successfully deployed for multiple application domains. Here we try to apply it on the vehicular sensors in order to detect anomalous activities. As a part of analysis, we have collected live data from cars using CAN Bus shield and analyzed the possibility of converting them in to context in a higher plane. To create a meaningful test scenario, we consider a subset of sensors and actuators deployed in a real vehicle. Then we show the different instantiations in our Ontology and some of the SWRL rules developed. The different sensors in the subset include "vehiclespeedsensor" which reports speed of the vehicle at regular intervals, "environmentlightsensor" which report intensity of ambient light around the vehicle, "distancesensor" which report the distance of the vehicle from an obstacle and "steeringanglesensor" which report current angle of the steering wheel from straight position. The actuators include "brake", "accelerometer" and "headlight". In the Ontology each sensor is associated with an attribute of the system, in this case a car. For example, the distancesensor will senseAttribute distance.

LCD layer will pick up raw CAN messages and interpret them to corresponding sensor values. Now the vehiclespeedsensor will be sending current speed of the vehicle in quick successions. But LCD layer will generate a higher level context for speed like, speed is "low", "high", "average" etc. Another example which depicts the functionality of the LCD layer is that different vehicles have different sensors for the distance sensor. Some may have radar sensors while some others may have acoustic sensors. But LCD layer will convert these

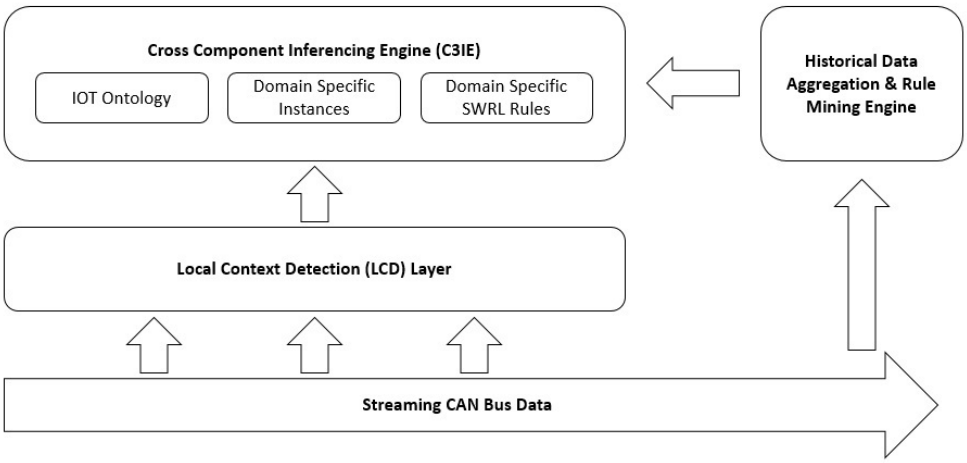

Fig. 1: Overall System Architecture.

different values to a common distance value. LCD layer will continuously monitor the data flowing on the common bus and generate higher level contexts. When it detects a change in value of any of the local contexts for sensors, it will remove the corresponding local context of the sensor from the knowledge base and inserts new local context information. To insert new context, it will check the Ontology to infer that the corresponding sensor will affect a specific attribute of the system and it will update it accordingly. This will allow the reasoner to work on the attributes of the system directly. We will now describe SWRL rules which can be used to detect some abnormal behaviors.

```
hasComponent (vehicle, distancesensor)^
    hasComponent (vehicle, vehiclespeedsensor)
    ^ hasStateValue (distanceProximity, "low")
    ^ hasStateValue (vehiclespeed, "high") ==>
    hasStateValue (currentvehiclestate,
    "Anomalous Collision")}
```

In the above SWRL rule, it will first check if the vehicle has sensor components distance sensor and vehiclespeed sensor. This is added so that the rules will be generic for different vehicles. Some of them may or may not have a specific sensor. If they are available, the attributes associated with them will be properly updated. Hence in that case if the distance proximity has value “low” and vehiclespeed has value “high”, it implies an anomalous situation.

```
hasComponent (vehicle,
    environmentlightsensor) ^ hasComponent
    (vehicle, vehiclespeedsensor) ^
    hasComponent (vehicle, headlight) ^
    hasStateValue (environmentlightsensor,
    "low") ^ hasStateValue (headlight, "off")
    ^ hasStateValue (vehiclespeed, "high") ==>
    hasStateValue (currentvehiclestate,
    "Anomalous No Light")}
```

In this example, in the presence of environmentlightsensor, vehiclespeedsensor and headlight actuator, if the ambient light is low and vehicle speed is high and headlight actuator state is off, then it represents an anomalous situation.

```
hasComponent (vehicle,
    steeringwheelpositionsensor) ^
    hasComponent (vehicle, vehiclespeedsensor)
    ^ hasStateValue (vehiclemovementdirection,
    "high") ^ hasStateValue (vehiclespeed,
    "high") ==> hasStateValue
    (currentvehiclestate, "Anomalous Sudden
    Direction Change")
```

This is another example in which a sudden change in the angular position of the steering when speed is high in the presence of steeringwheelpositionsensor and vehiclespeedsensor inferring anomaly.

VI. Challenges & Future work

In this paper we have described a rule based approach to mine vehicular context where the data is collected from a car’s CAN bus. We discussed the basic CAN bus architecture and different attacks possible on vehicles. The different strategies include reprogramming ECU, introducing foreign components into the system and exploiting bugs which may be present in the system. Lack of authentication is one of the main vulnerability in the vehicular bus architecture. Since introduction of new protocols can aid only future vehicles and not existing ones, we chose to propose a detect and mitigate technique. We modeled the vehicular system into IoT architecture and proposed a system which uses context to detect malicious activities. We extended the IoT-lite Ontology for use with vehicular systems and populated it with required instances. We also introduced an additional layer which converts the sensor data into relevant local contexts to aid faster processing. We then used SWRL rules to model different anomalous scenarios, put them in Protege and validated that our proposed model can detect anomalous scenarios.

Our current work assumed that we have SWRL rules to detect anomalous scenarios. But writing all rules manually is quite cumbersome, inefficient and error prone. Our future work will focus on analyzing historical data and mine SWRL rules automatically from them. Once the rules are generated, they

could be either manually verified or we can use the reasoner to check for consistency. Another advantage of using rule mining is that it can help the system to adapt to different driving conditions and different drivers automatically. Yet another challenge we faced is the real time inference from the high volume and velocity data. Even though we have an additional LCD layer to ease the process, we plan to analyze and test the performance against various existing streaming reasoning systems like CSPARQL, Streaming SPARQL, TrOWL etc.

REFERENCES

[1] Gregory D Abowd et al. "Towards a better understanding of context and context-awareness". In: *International Symposium on Handheld and Ubiquitous Computing*. Springer. 1999, pp. 304–307.

[2] Seung-Ho Baek et al. "Sensor information management mechanism for context-aware service in ubiquitous home". In: *IEEE Transactions on Consumer Electronics* 53.4 (2007), pp. 1393–1400.

[3] Davide Francesco Barbieri et al. "C-SPARQL: a continuous query language for RDF data streams". In: *International Journal of Semantic Computing* 4.01 (2010), pp. 3–25.

[4] Davide Francesco Barbieri et al. "Querying rdf streams with c-sparql". In: *ACM SIGMOD Record* 39.1 (2010), pp. 20–26.

[5] Andre Bolles, Marco Grawunder, and Jonas Jacobi. "Streaming SPARQL-extending SPARQL to process data streams". In: *European Semantic Web Conference*. Springer. 2008, pp. 448–462.

[6] Stephen Checkoway et al. "Comprehensive Experimental Analyses of Automotive Attack Surfaces." In: *USENIX Security Symposium*. San Francisco. 2011.

[7] Anusuriya Devaraju, Simon Hoh, and Michael Hartley. "A context gathering framework for context-aware mobile solutions". In: *Proceedings of the 4th international conference on mobile technology, applications, and systems and the 1st international symposium on Computer human interaction in mobile technology*. ACM. 2007, pp. 39–46.

[8] Anind K Dey, Gregory D Abowd, and Daniel Salber. "A conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications". In: *Human-computer interaction* 16.2 (2001), pp. 97–166.

[9] Anind K Dey and Alan Newberger. "Support for context-aware intelligibility and control". In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM. 2009, pp. 859–868.

[10] Bernardo Cuenca Grau, Christian Halaschek-Wiener, and Yevgeny Kazakov. "History matters: Incremental ontology reasoning using modules". In: *The Semantic Web*. Springer, 2007, pp. 183–196.

[11] Bogdan Groza et al. "LiBrA-CAN: Lightweight Broadcast Authentication for Controller Area Networks". In: ().

[12] Tao Gu, Hung Keng Pung, and Da Qing Zhang. "A service-oriented middleware for building context-aware services". In: *Journal of Network and computer applications* 28.1 (2005), pp. 1–18.

[13] Ahmed Hazem and Hossam AH Fahmy. "LCAP-A Lightweight CAN Authentication Protocol for securing in-vehicle networks". In: *10th escar Embedded Security in Cars Conference, Berlin, Germany*. Vol. 6. 2012.

[14] Tobias Hoppe and Jana Dittman. "Sniffing/Replay Attacks on CAN Buses: A simulated attack on the electric window lift classified using an adapted CERT taxonomy". In: *Proceedings of the 2nd workshop on embedded systems security (WESS)*. 2007, pp. 1–6.

[15] Tobias Hoppe, Stefan Kiltz, and Jana Dittmann. "Security threats to automotive CAN networksPractical examples and selected short-term countermeasures". In: *Reliability Engineering & System Safety* 96.1 (2011), pp. 11–25.

[16] Karl Koscher et al. "Experimental security analysis of a modern automobile". In: *2010 IEEE Symposium on Security and Privacy*. IEEE. 2010, pp. 447–462.

[17] Brian Y Lim and Anind K Dey. "Toolkit to support intelligibility in context-aware applications". In: *Proceedings of the 12th ACM international conference on Ubiquitous computing*. ACM. 2010, pp. 13–22.

[18] Charlie Miller and Chris Valasek. "Adventures in automotive networks and control units". In: *DEF CON* 21 (2013), pp. 260–264.

[19] Charlie Miller and Chris Valasek. "Remote exploitation of an unaltered passenger vehicle". In: *Black Hat USA* (2015).

[20] S. N. Narayanan, S. Mittal, and A. Joshi. "OBD_SecureAlert: An Anomaly Detection System for Vehicles". In: *2016 IEEE International Conference on Smart Computing (SMARTCOMP)*. May 2016, pp. 1–6. DOI: 10.1109/SMARTCOMP.2016.7501710.

[21] Jeff Z Pan et al. "Exploiting tractable fuzzy and crisp reasoning in ontology applications". In: *IEEE Computational Intelligence Magazine* 7.2 (2012), pp. 45–53.

[22] R Couto Antunes da Rocha and Markus Endler. "Middleware: Context management in heterogeneous, evolving ubiquitous environments". In: *IEEE Distributed Systems Online* 7.4 (2006), pp. 1–1.

[23] Michele Ruta et al. "A mobile knowledge-based system for on-board diagnostics and car driving assistance". In: *UBICOMM 2010, The Fourth International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies*. Citeseer. 2010, pp. 91–96.

[24] Darlene Storm. *Hack to steal cars with keyless ignition: Volkswagen spent 2 years hiding flaw*. Available at http : / / www . computerworld . com / article / 2971826 / cybercrime - hacking/hack - to - steal - cars - with - keyless - ignition - volkswagen - spent - 2 - years - hiding - flaw.html.

[25] Anthony Van Herrewege, Dave Singelee, and Ingrid Verbauwhede. "CANAuth-a simple, backward compatible broadcast authentication protocol for CAN bus". In: *ECRYPT Workshop on Lightweight Cryptography*. Vol. 2011. 2011.

[26] Marko Wolf, André Weimerskirch, and Thomas Wollinger. "State of the art: Embedding security in vehicles". In: *EURASIP Journal on Embedded Systems* 2007.1 (2007), pp. 1–16.