# ANT-Centric IoT Security Reference Architecture—Security-by-Design for Satellite-Enabled Smart Cities

Kwok-Yan Lam, *Senior Member, IEEE*, Sananda Mitra, Florian Gondesen, and Xun Yi

***Abstract*—Internet of Vehicles (IoV), a special form of Internet of Things (IoT), is an important enabler of intelligent transportation system, which is one of the most strategic applications in smart city initiatives. In order to achieve its intended functionalities, IoV requires anytime anywhere connectivity, which cannot be satisfied by traditional networking technologies. Space–air–ground-integrated network (SAGIN) is widely believed to be an ideal infrastructure for connecting IoV. In this article, we present an approach for understanding the security issues of complex IoT systems, and propose a security reference architecture for assessing security risks and addressing the security requirements. Specifically, we propose an activity-network-things (ANT)-centric security reference architecture, which is based on the three architectural perspectives in studying IoT systems, namely, device, Internet, and semantic. We discuss the limitations of existing IoT system architecture models, which are mainly evolved from the enterprise system architecture with some adaptation to the inherent features of IoT systems. Our approach can help manage the security risks by focusing on the critical activities performed in different microperimeters within an IoT system. The proposed architecture includes an organized process to understand the security requirements and select specific parameters for tailored security controls that are commensurate with organization-specific and application-specific security impacts of IoT. Our architecture is flexible enough to cater for any IoT application, and hence, can be easily applied to the case of SAGIN-enabled IoV.**

***Index Terms*—Cybersecurity, Internet of Things (IoT), Internet of Vehicles (IoV), security reference architecture, smart city, space–air–ground-integrated network (SAGIN).**

## I. Introduction

INTERNET of Vehicles (IoV) is characterized by complex computing and sensing capabilities, and is highly dynamic and mobile. Due to the very nature of vehicular systems, IoV needs anytime anywhere connectivity and has life-critical security requirements. IoV, consisting of smart cars with different levels of vehicular autonomy, can be modeled as an Internet of Things (IoT) system with integrated sensing and control components for automated navigation and enhanced safety.

Being a highly complex mobile system, IoV is wirelessly connected to roadside infrastructures, which, in turn, are connected to the backend control center through broadband cables, for exchanging vehicular and traffic control information. However, when IoV needs to operate out of the urban or suburban area, satellite communications are an alternative for maintaining safety and traffic management functions. The emergence of high-throughput satellite communications allows mobile IoT devices to be operational in remote or even isolated geographic locations. For instance, apart from IoV, offshore marine monitoring systems acquire large scale and real-time marine or nautical data with the help of satellite communications in the absence of any other reliable network infrastructure. Recent advancements in efficient trading and management of spectrum for satellite-based systems hold the promise to allow the connectivity of a large number of bandwidth-hungry IoT objects in remote and isolated areas [1]–[5]. Dynamic trading may also utilize scalable and incentive-driven blockchain platforms [6] to pave the way for trustworthy sharing of spectrum resources, especially benefiting satellite enabled smart mobility applications such as IoV.

The anytime anywhere connectivity requirements of IoV in order to achieve enhanced road safety and optimized traffic management cannot be satisfied by traditional networking technologies. The space–air–ground-integrated network (SAGIN), an emerging network architecture approach resulted from the integration of satellite systems, aerial systems, and terrestrial communication systems [7], [8], is widely believed to be an ideal infrastructure for meeting the connectivity needs of IoV. However, as an open and heterogeneous network, SAGIN applications are vulnerable to cyber attacks. Sensitive and mission-critical applications such as IoV need to be rigorously protected before they can realize the benefits of SAGIN. Fig. 1 shows a network-centric view of a SAGIN-enabled IoV system model, which illustrates the openness and heterogeneity of the underlying network infrastructure.

Nevertheless, through constant communications with the roadside infrastructure, IoV also helps in efficient traffic

Manuscript received September 16, 2020; revised January 11, 2021 and March 11, 2021; accepted April 2, 2021. Date of publication April 16, 2021; date of current version April 7, 2022. This work was supported by the Cyber Security Agency of Singapore (CSA), under its Repertoire of Initiatives Leveraging on Research Institutes and Think-Tanks to Contribute to the International Community "Toward a Secure and Trusted IoT Ecosystem." *(Corresponding author: Sananda Mitra.)*

Kwok-Yan Lam, Sananda Mitra, and Florian Gondesen are with the School of Computer Science and Engineering, Nanyang Technological University, Singapore (e-mail: kwokyan.lam@ntu.edu.sg; sananda.mitra@ntu.edu.sg; fgondesen@ntu.edu.sg).

Xun Yi is with the School of Computer Science and Software Engineering, RMIT University, Melbourne, VIC 3000, Australia (e-mail: xun.yi@rmit.edu.au).

Digital Object Identifier 10.1109/JIOT.2021.3073734

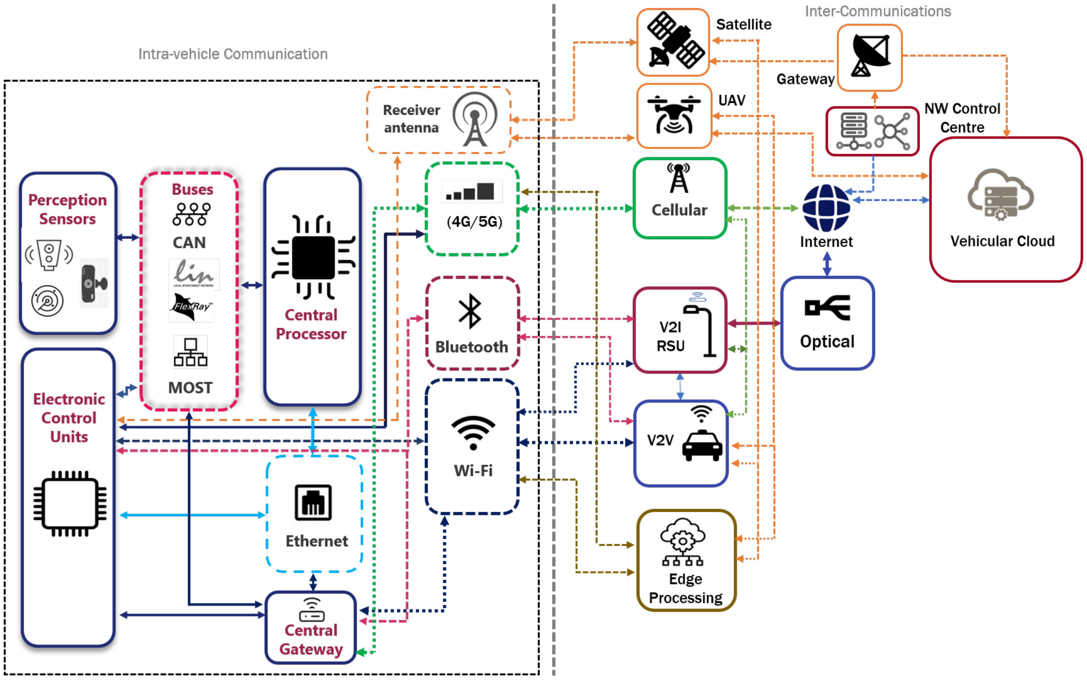

Fig. 1. Network-centric view.

management and enables intelligent transportation systems, and can potentially be used as an infrastructure platform for supporting other smart city applications. Empowered by the onboard sensors, intelligent control mechanism,s and pervasive cloud, fog, and edge computing, modern cars implement a range of autonomy features. As such, each car may be modeled as a complex IoT system connected via a grid of infrastructured and infrastructureless communication technologies. A wide range of IoV applications constitute the intelligent transportation system, which is a part of the crucial services provided in a smart city paradigm [9], [10]. Some of the typical safety critical applications are listed as follows.

1) *Real-Time Autonomous Driving Assistance:* To help distinguishing modern cars with varying degrees of autonomy, the society of automotive engineers (SAE International) proposed six levels of autonomy [11]. Such cars with varied autonomy levels typically integrate with diverse smart city applications (e.g., intelligent transportation system, smart grid, etc.) [12], [13] as IoT components. The autonomous driving performance is mostly dependent on the accuracy of individual vehicle perception, decision, and actuation [14]. A fleet of such smart vehicles can coordinate their movement utilizing an ubiquitous communication framework such as SAGIN. To complement safe driving, a number of applications that integrate with dashboard displays and navigation maps are gaining traction recently [15]. The applications that use smart end-user devices (smartphones, tablets, etc.) plugged in to the smart cars for providing distraction-free navigation, communication, and entertainment services via dashboard or in car entertainment units are being launched by various industry giants [16], [17]. Cloud-based diagnostics and timely over-the-air software updates are also some of the important services that provide autonomous driving assistance. Therefore, an ubiquitous, high throughput, and reliable IoV infrastructure is a must.
2) *Collision Avoidance System:* The objective of this service is to generate emergency warning messages if there is a possibility of collision between two vehicles. To optimize collision avoidance, information, such as traffic flow, number of vehicles, road condition, etc., can either be relayed in peer-to-peer communication or transmitted to a cloud-based server via heterogeneous communication [18]–[20]. Different levels of warning signals are then disseminated to the target vehicles based on factors, such as position, vicinity, etc. Depending on the level of autonomy and make of the vehicle, a collision avoidance system can also take control of the braking/steering actuation directly in the event of an imminent crash [21]. Communication infrastructures such as SAGIN have the ability to support large-scale and real-time transmission collision warning or collision mitigation systems seamlessly. As collision avoidance applications strictly require low latency and high reliability, these systems are dependent on an efficient communication substructure and the edge and fog framework.
3) *Traffic Management:* Traffic management applications provide guidance for efficient driving and navigation based on the data collected from different onboard or onroad sensors [22], [23]. Traffic management services act on information acquired from different onboard or onroad sensors. To optimize the traffic flow in nonurban areas, information, such as road conditions, number of vehicles, etc., can be transmitted from satellites, unmanned ariel vehicles (UAVs) [24], and hence,

SAGIN can be a useful option. Navigation guidelines can be communicated and coordinated from a cloud-based server considering factors, such as vehicle density, travel time, fuel status, etc. An efficient edge/fog computing framework can aid the real-time service requirements of this application.

In addition to the above, a range of other applications, such as smart parking, vehicle insurance, infotainment streaming, etc., exist to cater user needs, which do not necessarily have intersections with the safety-critical applications. Modern cars with their varied levels of autonomy are at the core of intelligent transportation systems and are targeted by numerous threat actors not only because they are a rich repository of information but also because they can be used to launch attacks of severe proportions. Misunderstood or overlooked security requirements can be fatal in these kind of smart city applications.

Albeit their significant benefits in terms of ubiquitous and flexible network coverage, SAGIN-enabled IoV applications are exposed to diverse cyberattacks due to their open and heterogeneous nature. Thus, sensitive and safety-critical applications such as IoV need to be rigorously protected before they can realize the benefits of anytime anywhere connectivity of SAGIN. For example, given the open network configuration of vehicular systems, one cannot make any assumption on the environment, and hence, more sophisticated security design such as the "zero trust principle" should be adopted in the design of secure IoV systems.

Security protection of IoV is a critical and yet challenging requirement. *First*, targeted cyberattacks on IoV could jeopardize personal safety leading to loss of life; and *second*, the ubiquitous connectivity of IoV over open and heterogeneous SAGIN networks makes the threat landscape rapidly evolving [25], [26]. The distributed and porous nature of the IoV fabrics pose significant challenges in security and resilience. The openness and scale also make it tough for security practitioners to establish and maintain a secure boundary as traditionally done in enterprise IT applications. The research works till date [27], [28] indicate that, in general, IoT systems have important characteristics that make them very different from typical IT systems, for example, as follows.

1) IoT devices lie at the intersection of sensor end-points and computing nodes, and thus, they interact with the physical world in a different manner compared to traditional IT devices.
2) Due to operational requirements in the field, many IoT devices must comply with stringent measures of performance, reliability, and resilience, quite unlike those in IT systems.
3) A large number of IoT devices are deployed in uncharacteristic locations. Thus, they need to be handled differently from traditional IT devices in terms of management, monitoring, and servicing such as updates of the software.

Compared to typical IoT systems, the security issues of IoV are even more challenging due to the following features.

1) *Dynamic and Mobile Connectivity Fabric*—Unlike the network structure applicable in the case of static IoT devices, vehicles are highly mobile. Thus, real-time heterogeneous V2X connectivity is a distinct characteristic of IoV.
2) *Need for Low Latency and High Reliability*—IoV systems are heavily dependent on an efficient communication substructure. The delay in V2X transmission or in communication of critical information from edge/fog/backend to any vehicle may compromise safety of the entire system. It is also crucial for the system designers to consider allowable latency bounds of safety-related applications in IoV systems [29] while choosing specific parameters of the cryptographic primitives and security protocols.
3) *Scale of the Network*—IoV consists of tens of thousands of smart cars distributed over a wide geographical area. Achieving scalability in terms of management, monitoring, updating, etc., may be harder than in typical IoT systems.

To analyze the security requirements and design for appropriate security controls of complex IoT systems such as IoV, a comprehensive reference architecture can be particularly helpful. In this article, we propose an IoT security reference architecture design for assessing security risks and addressing security requirements of IoT systems in a structured and granular manner. Specifically, we propose an activity-network-things (ANT)-centric security reference architecture, which is based on the three architectural perspectives in studying IoT systems, namely, device, Internet, and semantic [30], [31]. In our security analysis, we adopted a systematic approach to analyze the security requirements in depth from each of the three different perspectives. Our security reference architecture by design can establish a structured relationship between the security requirements and the necessary controls. The proposed security architecture is flexible enough to incorporate all the essential underlying components of SAGIN-enabled IoV networks and can also be generalized as a security reference architecture for other smart city IoT applications.

This article is organized as follows. In Section II, we discuss the cybersecurity challenges of implementing IoT systems in general and IoV systems in particular. In this section, we also discuss existing reference architecture designs and why a modular approach is useful to study and assess cybersecurity risks of complex IoT systems. In Section III, we propose a generic IoT security architecture for guiding the security design, which can be extended for IoV systems in the open and heterogeneous network environment. In this section, we also show that the security reference architecture design applicable to IoV systems captures the security characteristics of a SAGIN framework as well. In Section IV, we discuss the characteristics of some security mechanisms for addressing the identification and authentication requirements of IoT systems. The discussion of this article is concluded in Section V.

## II. Cybersecurity Challenges and Related Works

Major security challenges of an IoT system [32]–[34] are distinct from generic IT systems. Some of the critical technical challenges, which force us to rethink the whole cybersecurity paradigm, include the following.

1) The complexity of an IoT system in terms of the number of devices connected and wide-ranging communication and processing requirements.
2) The increased exposure to physical attacks due to deployment in the field.
3) A large volume of complex interactions comprising data of varying sensitivity between various system components.
4) The limited amounts of power, storage, memory, or processing capability of low-end devices may not support complex cryptographic operations, as is required to enforce standard security controls.

The security of IoV is a challenging issue due to its open and dynamic nature. Various factors plague the security landscape of IoV. Some of the typical issues arise due to the increased exposure of the system components in a perimeterless framework; reliance on heterogeneous communication systems that impose practical constraints on performance, reliability, and security; and interaction with a plethora smart application in smart city setup (explained in Section I). IoV inherits the challenges of IoT, and on top of that adopting an integrated network structure such as SAGIN will add to the complexities with respect to heterogeneity and scalability. In addition to that a connected car itself is exposed to serious cyberattacks through its various interfaces with the open and heterogeneous public network [5], [35]. Security challenges of a smart car are manifold. The vital ones include protection from data breach in any state, ensuring integrity of system components, such as perception sensors, electronic control units (ECUs), advanced driving-assistance systems (ADASs), etc., secure peer-to-peer/client-server communication, secure transmission and activation of (over-the-air) firmware updates, and secure diagnostics.

Trust management is also equally challenging in IoV due to the performance overhead that key management, identity management, authentication, or remote attestation induce for resource-constrained processors onboard a smart car, in roadside units (RSUs) or low-end edge devices [27], [36]. Lightweight authentication and attestation protocols (e.g., physically unclonable function (PUF) based [37], [38]) may be considered for these low-resource processors. The challenges of decentralized trust management and access control may also be managed using an underlying blockchain fabric [6]. The physical security of IoV endpoints is also critical due to the open environment in large-scale deployments, especially in the case of static endpoints, such as RSUs, edge devices, electric charging stations, etc. Cloud-based remote management and monitoring, such as [39] and [40] may be used to register, track, and control endpoints on top of standard physical security control measures to ensure end-to-end security in a perimeterless environment.

### A. IoT Security Reference Architectures

To address the complex cybersecurity issues of IoV in a holistic and risk-based manner, a sophisticated architecture design based on the security-by-design principle is important. A security reference architecture must support a system state constituting of secure assets, secure communications, and secure processing for an IoT application. To have a granular and optimized design approach for a security reference architecture, in-depth understanding of underlying technological aspects is a must. The visual representations of the association of security risks and requirements with the different architectural elements integrated together in a framework can provide a better and more efficient picture of the security considerations.

Introduced by Zheng *et al.* [9], the layered model is the most predominant style for representing IoT architectures. Architectural models with three to seven layers have been proposed till date to support modeling of IoT systems and for providing interoperability [4], [28], [41], [42]. As an extension of IoT, it is a practical necessity for IoV to have a generic and flexible reference architecture establishing the relation between the different building blocks. A generic architecture can help choosing the user-vehicle technological elements optimally for underlying applications. Contreras-Castillo *et al.* [43] identified gaps in layered architectures [44]–[46] not having security as a design consideration in safety-critical systems such as IoV. They proposed a seven layered architectural model where six layers are dedicated for ranges of operations such as driver notification management for processing of information in the cloud and an extra security layer transversal to the other six. Though the authors propose a model with cross-layered consideration for security measures, the lack of granularity in identifying the security requirements does not make the reference model easily applicable.

The application of appropriate security control measures relies on precise perception of the security requirements of each element participating in the acquisition, transmission, and intelligent realization of data. The reference models depicted in literature do not tie together the effect of various critical activities in determining the security objectives of an IoT system. Let us take as an example a study on security practices of smart cars released by the European Union Agency for Network and Information Security (ENISA) [47]. The study identifies sensitive assets, defines a high-level reference model of a smart car, maps threats and potential risks to the sensitive assets, identifies countermeasures, and refers standards for high-level security domains. This document presents a perspective associating critical assets to attack scenarios, cascading impacts, and countermeasures. But the authors themselves note that the model is a representative high-level view and does not incorporate the intricate complexities in an automotive scenarios. The security requirement identification in this study is based just on the threat taxonomy. An attack scenario they take as an example to illustrate their approach is "large-scale deployment of a rogue firmware after hacking OEM back-end servers" and one of the countermeasures they list for this threat states "allow and encourage the use of strong authentication mechanisms." Though the model associate assets, which would be affected by this attack, it neither clearly performs security requirement analysis of the specific elements affected nor does it pinpoint the nodes among which should the security controls be applied to mitigate such a threat. To address such gaps in security design, this article proposes an ANT-centric security reference architecture.

## III. ANT-Centric Security Reference Architecture

In this work, we propose a security reference architecture framework, which can provide an organized and efficient way to identify the security requirements and associate the best practices and security controls for mitigating the risks. To manage the complexity of an IoT system, the design of the security reference architecture adopts the principle of "security in a zero trust environment." This requires the identification of critical areas in the system that handle the most sensitive assets, as well as how the data flows across the system in relation to those critical areas. Once all attack surfaces are identified and assessed, appropriate security controls must be placed as close to the critical areas as possible, creating microperimeters around them.

In order to allow for better understanding of the system environment, system features, and typical security requirements from the angle of the end-to-end security needs of *critical activities*, our design is based on the juxtaposition of three different views or perspectives of an IoT system.

1) The *activity-centric* view helps to understand the context of system components and helps in identification of critical nodes vis-à-vis activities where sensitive data are processed, stored, or handled in an IoT system.
2) The *network-centric* view helps to understand the connectivity among all components and thus, the flow of sensitive data through the system. While IoT systems generally use the Internet, local networks with IoT-specific nonstandard communication protocols make the network-centric perspective of the reference architecture imperative for comprehending cybersecurity requirements.
3) The *things-centric* view helps to understand the nature of the security issues of heterogeneous data acquisition with the help of diverse devices not having any clear security perimeter.

The proposed security architecture can provide means for system owners to determine the sensitivity of the data and the criticality of the activities that process these data. The sensitivity of IoT data may be inherent to the nature of the data (e.g., video data of a key installation), or due to the environment within which such data are acquired (e.g., the temperature reading of a server processing classified data). Additionally, processing at edge nodes may turn out to be more sensitive due to the nature of the processing (e.g., face recognition models in camera video feed) rather than the raw data itself. The security requirements are primarily identified based on the activity-centric model while the network-centric and things-centric models allow further refinement of the risk assessment by providing different system perspectives.

### A. Activity-Centric View

In the proposed reference architecture, the notion of critical activities is the key to identify appropriate security control measures for each type of node or subsystem used in an IoT system. We identified eight basic activities that require certain security controls at the nodes performing them to ensure secure operation of the whole IoT system. These activities are related to changing the quality of data (SA, PP, CE, and DA) or important to dissemination of data (NT, ST, CC, and IO). The quality of data is related to their value to adversaries and the importance to the proper functioning of the IoT system.

1) *Sensing and actuating (SA)* is the interaction with the physical world, measuring or changing physical parameters, thus translating between data and physical properties. While actuation also requires safety measures, which are out of scope, the main impact on security is the edge to the physical world, mainly requiring physical protection and authentication for proper functioning. Adversaries may manipulate the physical world, which may often not easily be detected on the sensor level and therefore, requires downstream data sanity checks. Autonomous cars consist of many systems performing this activity, e.g., environmental sensors, window lifter, etc.
2) *Preprocessing (PP)* reduces raw sensor data to the required information, usually for higher efficiency or privacy requirements. This reduction of data may impede data sanity checks at later stages. Adversaries compromising a PP node may be able to access data that are not available at later stages.
3) *Processing (DA)* enriches data from multiple sources to decisive information, rendering performing nodes highly critical for operation and interesting to adversaries. Examples for such nodes in a smart car are ECU and ADAS.
4) *Crypto endpoint (CE)* is the activity of applying or verifying integrity codes or encrypting or decrypting data. At this stage, data are available in both forms (plaintext and ciphertext).
5) *Network transport (NT)* is transmitting data over (open) networks. The main issue is availability, which is addressed by SAGIN.
6) *Storage (ST)* is data at rest available for later access. Cars/vehicular clouds may keep records of all sensor data to reconstruct accidents.
7) *Command and control (CC)* refers to the control or configuration of subordinate nodes, allowing adversaries capable of compromising such a node to easily compromise the underlying nodes.
8) *Interfacing with external systems (IO)* is exchanging data with entities that are not under the control of the IoT systems. This includes users of the system. As interfaces allow other entities to access, incoming attacks can be expected. Inputs cannot be trusted and need to be sanitized. Providing data requires authentication and access control. In an autonomous vehicle (AV), not only the V2V and V2I communication but also the human interface components such as the steering wheel can be modeled as interfaces.

The list of critical activities may be extended based on the requirements of an IoT system. In the case of SAGIN-enabled

TABLE I
CRITICAL ACTIVITIES, SPECIFIC RISKS, AND EXAMPLES OF CRITICAL NODES THAT SUFFER FROM THOSE RISKS IN A SAGIN-ENABLED IOV FRAMEWORK

| **Critical Activities** | **ID** | **Specific Risks (examples; not exhaustive)** | **Critical Nodes** |
|---|---|---|---|
| Sensing and actuating | SA | Manipulation of sensor data participating in the cruise control. Can reveal vehicle location and contain personally identifiable information (PII). Compromise of actuation nodes entails safety risks. | On-board Sensors, GPS receiver |
| Preprocessing | PP | Can reveal information about the decision-making process for navigation, vehicle position etc. | Central Processor, On-board Camera |
| Encrypting or decrypting sensitive data, generating or verifying auth. tags | CE | Can reveal information about the nature of navigation, position, driving patterns, car conditions, software packages, updates. | GPS module, Central Processor, Central Gateway, ECUs, RSUs, Cloud |
| Network transport | NT | Exposure may lead to compromise of entire intra-vehicular communication structure. Can reveal information like location, position in a fleet etc. | Central Gateway |
| Processing | DA | Manipulate driving and actuation process, compromise passenger privacy | Central Processor, ECUs, Cloud |
| Storing | ST | Compromise may lead to exposure of aggregated information like adaptive driving patterns, car conditions, passenger details containing PII | Central Processor, Cloud |
| Controlling, configuring and patching underlying nodes | CC | Integrity of all underlying operations can be compromised | Central Processor, Cloud |
| Data interfacing with users or external networks/systems | IO | Retrieve confidential data or provide bogus input data to the interfacing systems | Central Gateway, Cloud |

IoT systems such as IoV, the proposed security reference architecture considers eight critical activities, as listed in Table I.

## B. Network-Centric View

There are numerous network protocol suites that may be used in IoV, with varying security features or even security flaws. The required level of security controls must be maintained end-to-end while traversing all respective networks possibly having a complicated topology. This might include switching between different protocols and even multiple stages of encryption and decryption, complicating the security scenario. Nodes enabled with higher computing resources at the edge can store and process raw data to reduce the traffic load as well as the response time supporting real-time data processing or delay-sensitive operations of IoV. Multiple operations executed at the edge can thus cause serious concerns in terms of data security and privacy. Raw and processed data handled by the edge nodes are more vulnerable to attacks due to the perimeterless structure and the edge nodes being closer to the field in terms of deployment [48]. The network-centric view in Fig. 1 is part of the proposed architecture that provides insights into the limited range or wide area networks in an IoV environment. This view also helps to group the communication framework into public, private, and protected segments for structured RiskAssess, thus helping to identify nodes that perform critical activities. The ANT-centric security architecture aims to focus on the protection of critical activities and adopt end-to-end security with the notion of microperimeters so as to avoid security assumptions on the underlying physical network. The aforementioned network architecture diagram illustrates the fact that the ANT security architecture is designed to cater for heterogeneous network characteristics.

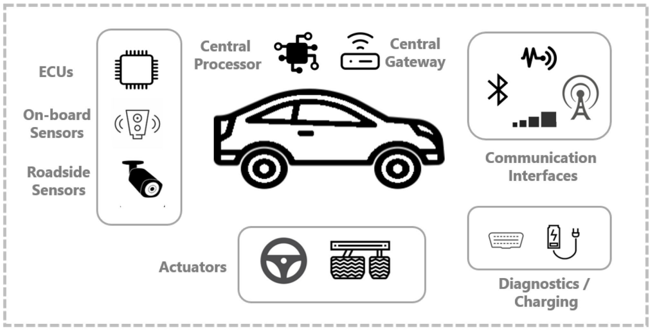

Fig. 2. Things-centric view.

Research studies [26], [49]–[51] show that the threats posed to a connected autonomous or semi-AV not only compromise passenger or road safety but may also be footholds for further attacks targeting the IoV infrastructure. The area of cybersecurity in AVs and IoV technology has not been explored rigorously in spite of the fact that numerous attacks have been identified. Therefore, it is important for practitioners to analyze the severity of potential cyberthreats at different segments for intra-AV and inter-AV networks. Table II shows a summary of security issues that can be identified by system designers or integrators consulting the intricate system details brought forward by things- and network-centric views. Once the security issues are recognized, the critical activities/nodes of the system can be easily be identified.

## C. Things-Centric View

IoT devices or things perform a range of tasks, such as sensing, actuation, storage, preprocessing, etc., with the help of ubiquitous connectivity. Fig. 2 shows how the things-centric view can provide insights into the variety of sensors or

TABLE II
SECURITY ISSUES IN SAGIN-ENABLED IoV

| | **Threats** | **Target** | | **Attack Vectors** | **Foothold** | **Access** | **Compromised** | | |
|---|---|---|---|---|---|---|---|---|---|
| | | **AV** | **IoV** | | | | **C** | **I** | **A** |
| S | Spoofing | ✓ | | Blinding or forging sensor input, forging the data needed to calculate navigation ranges | Perception sensors, GPS receivers | Remote, Proximity | | ✓ | ✓ |
| T | Tampering | ✓ | ✓ | Packet injection, on-board storage manipulation, map database poisoning | OBD II, Communication interfaces, V2X platforms | Remote, Proximity | | ✓ | |
| R | Repudiation | | ✓ | Manipulation of notifications, warning messages (amounting to erasure of accountability) | V2X applications, IoV platforms | Remote | | ✓ | |
| I | Information Disclosure | ✓ | ✓ | Side-channel analysis, packet sniffing, interception of radio waves | ECUs, Communication interfaces, On-board sensors | Remote, Proximity | ✓ | | |
| D | Denial of Service | ✓ | | Sybil, sinkhole, wormhole, jamming | Communication interfaces, V2X communication nodes | Remote | | | ✓ |
| E | Elevation of Privilege | ✓ | ✓ | Keyloggers, brute force, packet sniffing | Telematics and Infotainment systems, Communication interfaces, V2X comm. nodes | Remote, Proximity | ✓ | ✓ | ✓ |

TABLE III
MAPPING OF CRITICAL ACTIVITIES WITH SECURITY CONTROLS IN SAGIN-ENABLED IoV

| **Control Measures** | **Identifier** | Critical Activity Identifier | | | | | | | | Security Attributes | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | SA | PP | CE | NT | DA | ST | CC | IO | C | I | A |
| Entity Authentication | EA | ✓ | | ✓ | | | ✓ | ✓ | | | ✓ | |
| Key Management | KM | | | ✓ | | | | | | ✓ | ✓ | |
| Symmetric Key Encryption | SKE | | | ✓ | | | ✓ | | | ✓ | | |
| System Integrity / Hardware Security | SI | | ✓ | ✓ | | ✓ | ✓ | | | | ✓ | |
| Physical Security | PS | ✓ | | | | ✓ | ✓ | | | | ✓ | |
| Access Control | AC | ✓ | | | ✓ | ✓ | ✓ | | | ✓ | | |
| User Authentication | UA | | | | | | ✓ | | ✓ | | ✓ | |
| Intrusion Detection and Prevention | IDP | | | | | | | | | ✓ | ✓ | |
| Redundancy | RDN | ✓ | | | ✓ | | ✓ | | | | | ✓ |
| Data Integrity / Message Authentication | DI | | | ✓ | | | | ✓ | ✓ | | ✓ | |
| Data Sanity Checks | SAN | | ✓ | | | ✓ | | | ✓ | | ✓ | |
| Configuration Management | CM | | | | | | | ✓ | | ✓ | ✓ | ✓ |
| Life Cycle Management | LCM | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |

devices in an IoV environment. The things-centric view could, thus, help to identify whether capabilities of the end devices or physical things in terms of computation, power, etc., are sufficient for execution of adequate cryptographic operations.

### *D. Security Controls*

Our reference architecture design also provides a list of security controls commensurate with the risk exposure of critical nodes identified through the aforesaid ANT-centric RiskAssess. The list of controls is in line with the adoption of the zero trust principle, which means that the selection of necessary checks and controls is based on the premise that there is no implicit trust in any part of the network. The proposed security reference architecture maps 13 standard security control measures to their appropriate critical activities for uncomplicated and effective decision making in the case of SAGIN-enabled IoV systems, as shown in Table III.

Considering the impact of physical attacks on large-scale IoV in our model, we recommended physical security that maps to critical activities "sensing and actuating" (SA) at the endpoints (static or onboard the vehicles), "processing" (DA) at the central processor of the vehicle or the edge/cloud, and "storage" (ST) onboard a vehicle, in an RSU, or in the edge/cloud. As per Table III, we attribute this type of critical activities primarily to *integrity*. The tamper-resistant design of the critical nodes identified through our architecture will ensure physical security in such cases, where the practitioner may choose an appropriate security level as explained in the next section, to ascertain the commensurate security control recommendations. In addition, to ensure the integrity of underlying nodes, one may need cloud-based remote management, which maps to "configuration management" (CM) in our model.

### *E. Security Levels*

Our design associates each control measure to its three levels of security (high, medium, and low), as applicable. We designed our security reference architecture to be applicable to any generic IoT domain. Therefore, it contains recommendations that use the currently highest feasible security measures

(H) to the lowest security requirements that can be considered secure to date (L) providing flexibility to the system designers or integrators. The security levels are aligned with the impact levels of the FIPS 199 standard [52]. Standard SAEJ3061 [53] based on ISO26262 [54] integrates automotive cybersecurity and functional safety. Once completed, this standard should be consulted to ascertain the appropriate level of control measures in automotive use cases. A security practitioner must choose levels of security for the control measures based on the considerations of reputation, personal/organizational interest, financial/data loss, and public safety associated with the respective critical activity. It is worth noting that the issue of limited computing resource during complex cryptographic operations (e.g., authenticated encryption) may be addressed by choosing lightweight alternatives. Lightweight cryptographic primitives are explored in recent works such as [55]. Once standardized, options for better lightweight ciphers suitable for resource restricted IoT environments can be obtained. Whereas, the issue of limited spectrum/bandwidth in face of cryptographic mechanisms can be alleviated by selecting stream ciphers or small block sizes to limit the overhead caused by ciphertext expansion. In the case of IoV, personal safety should be the top priority, and the security control levels associated with driving decision making should be high. Table V presents the example of *entity authentication* and *message authentication* and the choices of security levels for them. Sensing and actuation activity is important for personal safety, and one of the security controls aligned to this activity is “entity authentication.” Thus, the key sensor module/node, e.g., camera acquiring safety critical data such as road conditions, must be protected with high level of entity authentication. Ensuring the integrity and consistency of data during all stages of navigation is crucial for preventing manipulation by an adversary or being changed by an error. Entities in a SAGIN-enabled IoV framework that exchange navigation-specific messages over untrusted networks can act as crypto endpoints to ensure “message authentication/data integrity.” As crypto endpoints are scattered throughout an IoV framework, the security architects and the system integrators must choose appropriate levels of security controls based on the impact level of the risks identified.

### *F. Architecture Application*

Our architecture focuses on the notion of critical activities, which are associated with a list of security control measures. For each control measure, the architecture provides details based on the required security level for the critical activity. To facilitate analysis, we present the logical model for applying the proposed security reference architecture, which helps to explain the systematic multiphase methodology. The model provides guidelines to specify security requirements and appropriate recommendations that can be utilized for a wide range of IoT applications due to the generic nature of the proposed reference architecture. It also helps in developing an overall vision and strategy for assessing security compliance in targeted IoT applications.

TABLE IV
NOMENCLATURE

| ID | Object | Remark |
|---|---|---|
| PVEnt | Physical/Virtual Entity | Components or elements in the system |
| CAct | Critical Activity | Sensitive activities in the system |
| SCon | Security Control | Controls to be applied in the system |
| SLev | Security Levels | High, Medium, Low levels of security |
| TLoc | Location of PVEnt | Location of the entity as a Thing |
| TCap | Capability of PVEnt | Capability of entity as a Thing |
| NTyp | Type of Network | Where PVEnt belongs in the system |
| NPro | Network protocol | Followed by PVEnt in network NTyp |
| ADat | Type of Data | Handled/stored/processed by PVEnt |
| APro | Type of Processing | Accomplished by PVEnt on ADat |
| AMdl | Adversarial model | Adversary actor type and position |

The nomenclature that we followed is shown in Table IV. The process iterates over all entities of the IoT system. Entities may be components, nodes, subsystems, or devices, either physical or virtual. The things-centric view (thingCV) maps the entity (PVEnt) to a set tuples of location and capability {(TLoc, TCap)}, network-centric view (networkCV) maps the entity (PVEnt) to a set tuples of network type and protocol {(NTyp, NPro)}, and activity-centric view (activityCV) maps the entity (PVEnt) to a set tuples of data and processing {(ADat, APro)} as follows:

$$\begin{aligned}\text{thingCV}(\text{PVEnt}) &\longrightarrow \{(\text{TLoc}, \text{TCap})\} \;\subset S_T\\ \text{networkCV}(\text{PVEnt}) &\longrightarrow \{(\text{NTyp}, \text{NPro})\} \;\subset S_N\\ \text{activityCV}(\text{PVEnt}) &\longrightarrow \{(\text{ADat}, \text{APro})\} \;\subset S_A.\end{aligned}$$

Thus, our overall ANT-centric architecture (antCV) maps any entity (PVEnt) to a set of sextuples {(TLoc, TCap, NTyp, NPro, ADat, APro)}, where the same PVEnt may map to different sextuples as per the combination of its thing, network, and activity centric point-of-view in the system

$$\begin{aligned}\text{antCV} : \{\text{Domain of PVEnt}\} &\longrightarrow S_T \times S_N \times S_A\\ \text{antCV}(\text{PvEnt}) &\longrightarrow \{(\text{TLoc}, \text{TCap}, \text{NTyp}, \text{NPro}, \text{ADat}, \text{APro})\}.\end{aligned}$$

This definition of the entity (PVEnt) is passed to the RiskAssess process, along with the formal model for the adversary (AMdl) consisting of the threat actor (AdvAct) and their position (AdvPos). RiskAssess identifies the pairs {(target, impact)} relevant to the vulnerabilities of the entity as well as the motive of the adversary. RiskAssess of the PVEnt sextuples corresponding to the adversarial models will result in multiple such {(target, impact)} pairs in the system

$$\begin{aligned}&\text{PVEnt} = \{(\text{TLoc}, \text{TCap}, \text{NTyp}, \text{NPro}, \text{ADat}, \text{APro})\}\\ &\text{AMdl} = \{(\text{AdvAct}, \text{AdvPos})\}\\ &\text{RiskAssess}(\text{PVEnt}, \text{AMdl}) \longrightarrow \{(\text{target}, \text{impact})_i\}_{i=1,\ldots,n}.\end{aligned}$$

Each target derived from RiskAssess (PVEnt, AMdl) will map with a subset of critical activities (CAct) in the system, as shown in Table I. Each CAct will map to a subset of security controls (SCon), as shown in Table III. Each SCon in this structure will be set to a security level (SLev) according to the impact of the target derived from RiskAssess (PVEnt, AMdl), as follows:

$$(\text{target}, \text{impact}) \longmapsto \big\{(\text{CAct}_j, \text{impact})\big\}_{j=1,\ldots,m} \quad \forall \text{ target}$$

$$\begin{aligned}(\mathrm{CAct}_j, \mathrm{impact}) &\longmapsto \{(\mathrm{SCon}_k, \mathrm{impact})\}_{k=1,\ldots,p_j} \quad \forall\, i\\ (\mathrm{SCon}_k, \mathrm{impact}) &\longmapsto \{(\mathrm{SCon}_k, \mathrm{SLev}_k)\}_{k=1,\ldots,p_j} \quad \forall\, i.\end{aligned}$$

Thus, we obtain a comprehensive set of tuples ($\mathrm{SCon}_j$, $\mathrm{SLev}_j$) of security controls (SCon) and corresponding security level recommendations (SLev) for the entity (PVEnt)

$$\begin{aligned}\mathrm{PVEnt} &\longmapsto \{(\mathrm{TLoc}, \mathrm{TCap}, \mathrm{NTyp}, \mathrm{NPro}, \mathrm{ADat}, \mathrm{APro})\}\\ &\longmapsto \{(\mathrm{target}, \mathrm{impact})_i\}_{i=1,\ldots,n}\\ &\longmapsto \big\{\{(\mathrm{SCon}_k, \mathrm{SLev}_k)\}_j\big\}_i\\ &\qquad \forall\; k = 1, \ldots, p_j \text{ per SCon}\\ &\qquad \forall\; j = 1, \ldots, m \text{ per CAct}\\ &\qquad \forall\; i = 1, \ldots, n \text{ per target.}\end{aligned}$$

Finally, each security control (SCon) corresponding to the PVEnt should be assigned the maximum security level (SLev) for each critical activity (CAct) to ensure that the entity is protected by the maximum common denominator (out of H, M, and L) of security control (SCon) for each target

$$\begin{aligned}\mathrm{PVEnt} \longmapsto &\bigcup_{i,j}\{(\mathrm{SCon}_k, \max(\mathrm{SLev}_k))\}\\ &\text{combining all CAct for } j = 1, \ldots, m\\ &\text{combining all target for } i = 1, \ldots, n.\end{aligned}$$

To instantiate, we again make reference to the ENISA reference model example stated in Section II-A. We apply our architecture on top of their model, and consider that the things-centric and network-centric views and critical activity classification are on par with our understanding of the SAGIN-enabled IoV framework. For the attack scenario "large-scale deployment of a rogue firmware after hacking OEM back-end servers," the activity-centric view classifies the back-end server as the command and control (CC) node. From Table III, one can identify that data integrity/message authentication (DI) security control maps with the CC critical activity and is an appropriate control measure for checking the authenticity of the firmware messages in this case. From Table V, we can find the high-medium and low levels of standard message authentication mechanisms available. But to choose among the levels a practitioner must understand the factors such as computation capability of the verifying node (things centric) or the characteristics of the underlying communication protocols supporting the exchange of the message authentication packets (network centric). Thus, the three views complement each other to help professionals to have a holistic idea of for deploying IoT systems, which are secure-by-design.

### G. Comparison With Similar Models

In the previous sections, we have shown how the proposed architecture binds together the system challenges, security requirements, and different levels of control measures in a heterogeneous IoT framework. The existing models for IoV do not provide the coverage of different security perspectives or the granularity of security controls. Neither do they provide a structured methodology for integrating security controls based on system features. Table VI presents the comparison of our proposed architecture with the most relevant reference models for IoV [43], [47] across their salient features.

TABLE V
SECURITY CONTROL LEVELS FOR ENTITY AUTHENTICATION AND MESSAGE AUTHENTICATION IN SAGIN-ENABLED IoV

| **Entity Authentication Recommendations (EA)** | |
|---|---|
| H | Replay-resistant Challenge Response Authentication using Asymmetric keys ACRYPT_H, HASH_L<br>or Symmetric keys HASH_L |
| M | Replay-resistant Challenge Response Authentication using Asymmetric keys ACRYPT_M, HASH_L<br>or Symmetric keys HASH_L |
| L | Replay-resistant Challenge Response Authentication using Asymmetric keys ACRYPT_L, HASH_L<br>or Symmetric keys HASH_L |

| **Message Authentication Recommendations (DI)** | |
|---|---|
| H | Digital Signatures ACRYPT_H<br>or HMAC HASH_L (256 bit key) |
| M | Digital Signatures ACRYPT_M<br>or HMAC HASH_L (192 bit key) |
| L | Digital Signatures ACRYPT_L<br>or HMAC HASH_L (128 bit key) |

| **HASH – cryptographic hash functions** | |
|---|---|
| **H** | SHA3-512 or SHA-512 |
| **M** | SHA3-384 or SHA-384 |
| **L** | SHA3-256 or SHA-256 |

| **ACRYPT – asymmetric cryptography** | |
|---|---|
| **H** | Elliptic curve P-521 or RSA (15360 bit key length) |
| **M** | Elliptic curve P-384 or Curve448 or RSA (7680 bit key length) |
| **L** | Elliptic curve P-256 or Curve25519 or RSA (3072 bit key length) |

## IV. DISCUSSION

Due to the complexity and interoperable nature of the practical system environment, the threats and security requirements vary throughout the IoV framework. Thus, the security recommendations must be based on thorough understanding of the system specifications, security requirements, and impacts of attacks. Let us take the example of integrity protection of navigation-related messages and trust management of inter-AV navigation entities to illustrate the considerations. To counter such risks, strong and efficient authentication mechanisms are required in a SAGIN-enabled IoV framework. To choose suitable authentication mechanisms, the system integrators must consider factors, such as the computational overhead for cryptographic operations (things centric), the bandwidth of communication protocols (network centric), and the critical nodes substantiating the identity or verifying the data (activity centric).

TABLE VI
COMPARISON OF THE PROPOSED ANT-CENTRIC SECURITY REFERENCE ARCHITECTURE WITH SIMILAR MODELS [43], [47] PROPOSED FOR IoV

| **Parameters** | **Contreras-Castillo et al. [43]** | **ENISA [47]** | **Proposed Architecture** |
|---|---|---|---|
| Security as Design Consideration | Basic | Ad-hoc | Granular (ANT-centric views) |
| Derivation of Security Requirements | Not included | Not straightforward | By design (Activity-centric) |
| Selection of Control Measures | Not included | Based on attack scenarios | Specific (mapped to Critical Activities) |
| Determining Security Control Levels | Not included | Not included | Precise (overlaying ANT perspectives) |
| Generalization to IoT Applications | Specific to IoV | Focused on Smart Cars | Flexible (generic for any IoT system) |
| Plug and Play Technology Update | Hard (substantial interpretation due to high level design) | Hard (substantial interpretation due to high level design) | Seamless based on modular ANT views |

As a security reference architecture, the proposed design does not include recommendations for specific security protocols. Instead, it specifies the general features and security parameters required for such protocols. Almost invariably, existing IoT applications generally recommended the use of traditional standard protocols such as the transport layer security (TLS). However, depending on the business objective and nature of data processed/handled in the IoT application, security designers may prefer mechanisms that are optimized for a particular use case in terms of security, computational effort, communication overhead, etc. For completeness, and as a continuation of the analysis in Section III-D and III-E, we briefly discuss some features of three distinct authentication mechanisms especially relevant to an IoT or a SAGIN-enabled IoV framework.

### A. Public Key Based

Public-key message authentication is generally accomplished through the *digital signature* paradigm. Elliptic curves provide an efficient algebraic structure for high security guarantees with lower parameter sizes. Thus, NIST approved elliptic curves with the standardized digital signature algorithm (DSA) [56] are popular in practical applications, including IoV. The core message authentication routine using ECDSA is pretty straightforward, as follows.

1) *Setup* fixes the curve as a tuple $T(p, a, b, G, n, h)$, where $p, a,$ and $b$ define the desired elliptic curve group

$$E(F_p) : \left\{(x, y) \mid y^2 = x^3 + ax + b \bmod p\right\} \cup \{\mathcal{O}\} \quad (1)$$

and $G, n,$ and $h$ denote the generator, its order, and cofactor.
2) *KeyGen* generates the public and secret key pair $\{pk, sk\}$ for the sender. The public key $pk$ is shared with the recipient for verification.
3) *Signature* on the navigation message $M$ is computed by the sender using the secret key $sk$ as follows, with the signature published/sent as a commitment of $M$:

$$\text{signature} \longleftarrow \text{ECDSA}(sk, M). \quad (2)$$

4) *Verification* of the signature on the navigation message $M$ is done by the recipient, using the public key $pk$ of the sender, to return either Accept or Reject

$$\text{Accept/Reject} \longleftarrow \text{ECDSA}(pk, M, \text{signature}). \quad (3)$$

The signature verification will result in Reject if: 1) the public key does not correspond to the private key or 2) if the signature does not pertain to the specific message $M$. Either way, a spoofing attempt on the navigation message $M$ or the corresponding signature will be thwarted. The signature mechanism ensures message authentication as well as identity authentication for the sender in case public-key infrastructure (PKI) is used. In the case of GPS communication, the control segment or operational control system is responsible for signing navigation messages on behalf of the satellites.

### B. Implicit Hardware Based

By definition, a PUF is a circuit that maps input bitstrings (challenges) to output bitstrings (responses) based on physical properties that depend on manufacturing variations, such that the responses are unique for each instance. As a PUF response is a measurement of physical parameters, small deviations between multiple responses to the same challenge are expected. Because of the unclonbability property, it is also impossible to create a model to predict the challenge response behavior. The size of the challenge response space renders it infeasible to exhaustively acquire all challenge response pairs (CRPs).

These properties would allow a simple lightweight authentication protocol where a backend needs to enroll a sufficient number of CRPs for each PUF instance into a DB. For authenticating a PUF-enabled device, one of the enrolled challenges $c_i$ is sent to the device, which evaluates it and returns the response $r'_i$. If the Hamming distance $d_H$ between the received response $r'_i$ and the stored response $r_i$ is below a threshold $t$, the device is authenticated. To prevent replay attacks, each CRP must be only used once. The protocol is depicted in Fig. 3.

Several more complex protocols have been proposed [57], of which, some allow using PUF constructions that have a small challenge response space such as SRAM PUF, which utilizes the startup behavior of SRAM cells. The protocols based on a fuzzy extractor [58] use error correction codes to exactly reproduce a cryptographic key from a PUF response, using the PUF as a key storage. However, using error correction codes creates significant overheads on the device side [37], even when using the concept of the reverse fuzzy extractor [59], which shifts decoding work from the device to the more powerful backend. A variant of the reverse fuzzy extractor is depicted in Fig. 4.

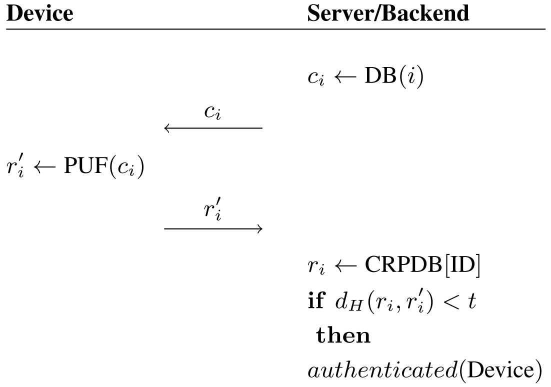
Device
Server/Backend
$c_i \leftarrow \mathrm{DB}(i)$
$c_i$
$r'_i \leftarrow \mathrm{PUF}(c_i)$
$r'_i$
$r_i \leftarrow \mathrm{CRPDB}[\mathrm{ID}]$
**if** $d_H(r_i, r'_i) < t$
**then**
$authenticated(\mathrm{Device})$

Fig. 3. Simple PUF authentication protocol. For each device, a sufficient number of CRPs needs to be enrolled to a database (DB). Each CRP can only be used once.

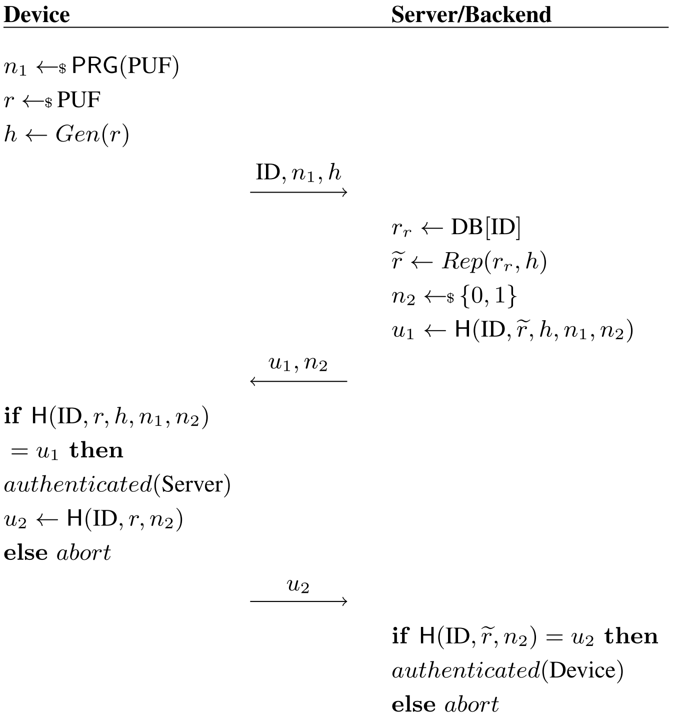
Device
Server/Backend
$n_1 \leftarrow_\$ \mathsf{PRG}(\mathsf{PUF})$
$r \leftarrow_\$ \mathsf{PUF}$
$h \leftarrow Gen(r)$
$\mathrm{ID}, n_1, h$
$r_r \leftarrow \mathrm{DB}[\mathrm{ID}]$
$\widetilde{r} \leftarrow Rep(r_r, h)$
$n_2 \leftarrow_\$ \{0,1\}$
$u_1 \leftarrow \mathsf{H}(\mathrm{ID}, \widetilde{r}, h, n_1, n_2)$
$u_1, n_2$
**if** $\mathsf{H}(\mathrm{ID}, r, h, n_1, n_2)$
$= u_1$ **then**
$authenticated(\mathrm{Server})$
$u_2 \leftarrow \mathsf{H}(\mathrm{ID}, r, n_2)$
**else** $abort$
$u_2$
**if** $\mathsf{H}(\mathrm{ID}, \widetilde{r}, n_2) = u_2$ **then**
$authenticated(\mathrm{Device})$
**else** $abort$

Fig. 4. Mutual authentication protocol [37] based on the modified reverse fuzzy extractor authentication protocol by Maes [60]. In the proposed protocol, the TRNG is replaced by a PRG seeded by a sample from the PUF. Each device's PUF response $r$ needs to be enrolled in the DB with the identifier (ID) of the device.

### C. Symmetric Cryptography With Delayed Key Disclosure Based

TESLA [61], is a symmetric key message authentication algorithm that depends on a *commitment-verification* paradigm based on shared randomness. The shared randomness is stored as the seed $r$, which is hashed iteratively (using a collision-resistant cryptographic hash function) to generate a verifiable sequential keychain $\{k_1, k_2, \ldots, k_N\}$, as follows:

$$r \xrightarrow{H} k_1 \xrightarrow{H} k_2 \xrightarrow{H} \cdots \xrightarrow{H} k_{N-1} \xrightarrow{H} k_N. \quad (4)$$

The keys in this keychain are released in the reverse order, from $k_N$ (the base key) to $k_1$, so that any intermediate key $k_i$ may be verified against the base key $k_N$ or a previously released intermediate key $k_{i+j}$ (for $1 < i + j < N$) as follows:

$$H^{N-i}(k_i) \overset{?}{=} H(k_N) \quad \text{or} \quad H^j(k_i) \overset{?}{=} H(k_{i+j}). \quad (5)$$

Note that the complexity for such a check is $\mathcal{O}(N)$, dependent on $N$, the size of the keychain, and may be significantly faster [practically $\mathcal{O}(1)$] if every intermediate key is available. Once the keychain is set up, the authentication of message and verification of the intermediate keys between the parties execute as a *commitment-verification* routine.

1) *Commit* to a navigation message $M$ using an *undisclosed* intermediate key $k_i$ using a cryptographically secure unforgeable message authentication code (MAC)

$$\text{commit} \longleftarrow \text{MAC}(k_i, M). \quad (6)$$

2) *Verify* the commitment after the disclosure of $k_i$ as follows.
   a) Verify key $k_i$ as $H^j(k_i) \overset{?}{=} H(k_{i+j})$, using a previously released intermediate key $k_j$ (or base key $k_N$).
   b) Verify authentication tag generated by the MAC as

$$\text{commit} \overset{?}{=} \text{MAC}(k_i, M). \quad (7)$$

The security of the scheme borrows from: 1) preimage resistance property of the hash function $H$ to protect premature disclosure of $k_i$; 2) collision resistance property of the hash function $H$ to protect against commitment malleability; and 3) unforgeability of the MAC to protect against malleability of the navigation message $M$. TESLA requires the shared seed to be authenticated at the beginning. The authentication is typically done with the help of classical asymmetric schemes, such as signatures and a PKI. This does not impact the performance much, because this only has to be done for each new receiver.

For each authentication scheme, the system integrators can refer to the list of recommended control measures as stated in Section III-E. It will help them choose the cryptographic primitives (PRG, Hash, MAC, Signature, etc.) commensurate with the level of acceptable risks evaluated through the ANT approach. The proposed security reference architecture is completely flexible. The practitioners can add or replace recommendations suiting their own requirements and based on the latest standards and best practices.

### D. Scalability of Authentication Schemes

The various communication patterns in IoV require appropriate cryptographic schemes, yet scalability remains an issue. In this section, we discuss how choosing different cryptographic schemes can affect the scalability. Vehicles receive sensitive data from other random vehicles in the vicinity as well as from the infrastructure, which prohibits the usage of preshared symmetric keys. The public-key cryptography typically incurs significant computational overheads, which may also affect real-time requirements. As such, processing of messages introduces additional delay in communication, which may not be negligible for time-critical information. For broadcast communication, TESLA can be an alternative that scales like public-key cryptography in terms of key material required while having a computational effort scaling as symmetric key

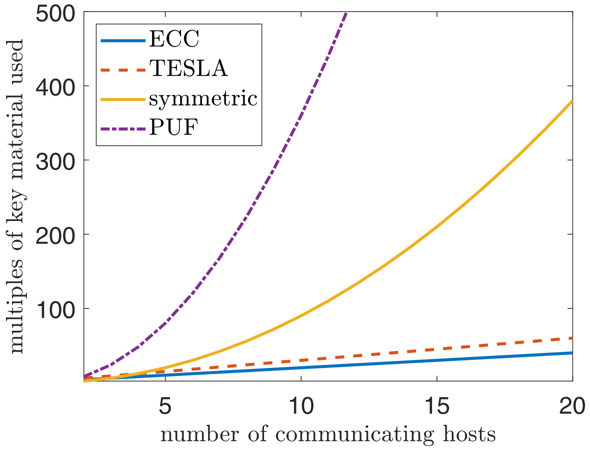

Fig. 5. Growth of key material with respect to the number of communicating entities. Key material increases linearly for ECC or Tesla and quadratically for symmetric key schemes, assuming peer-to-peer communication. Obtaining symmetric keys from a PUF has overhead due to error correction and low entropy, and this number also needs to be multiplied by the number of challenge–response pairs enrolled.

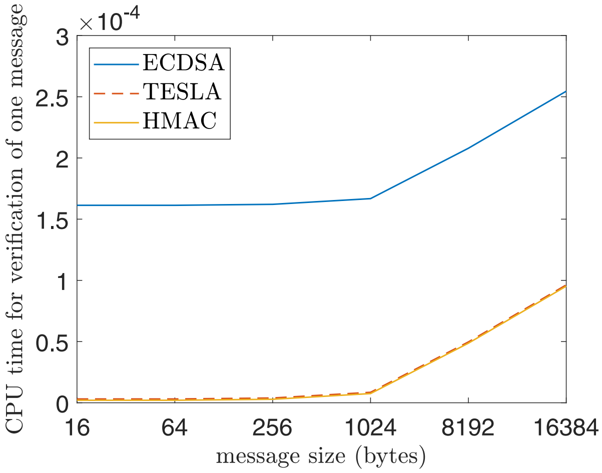

Fig. 6. CPU time (in seconds) needed to verify a message using SHA3-256 as the hash function. ECDSA consists of one hash and one ECC operation. HMAC uses one hash containing the message and one hash of the digest with padding and key. Tesla adds one additional hash operation over the key if the previous key chain is available.

schemes. PUF scales like symmetric keys, though amount of key storage required on the backend side is multiplied by the challenges used. PUF may be a viable way to tie a vehicle by hardware to its identity registered with the respective authority. Fig. 5 shows the scalability of the discussed authentication primitives with respect to the total key material used, assuming each entity is in communication with all other entities. Fig. 6 shows the CPU time used by a Raspberry Pi 4 to verify messages with ECDSA, HMAC, and TESLA using HMAC. Message authentication also entails an overhead in data transmission. This overhead mainly depends on the message size (as shown in Fig. 7) but also on the protocol and key sizes used. It is important to note that all these factors contribute to the latency of the schemes, and the security designer must take into account the allowable latency bound for the target application [29] while choosing the appropriate lightweight authentication protocol.

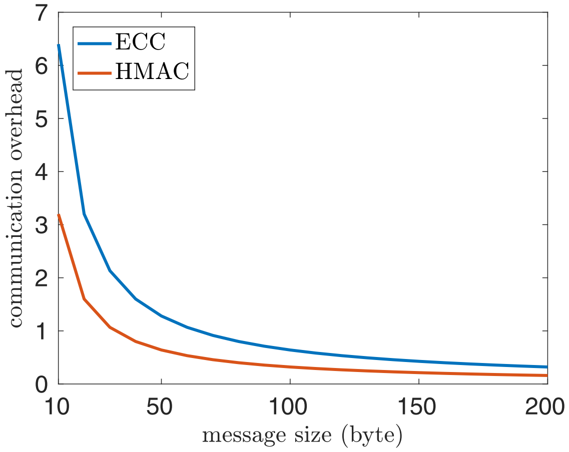

Fig. 7. Communication overhead factor (digest or signature size / message size) for the authentication schemes, neglecting protocol overhead. ESCDSA P-256 has a higher overhead compared to HMAC with a 256 bit digest.

## V. Conclusion

IoV, a special form of IoT, is an important enabler of intelligent transportation system, which is one of the most strategic applications in smart city initiatives worldwide. IoV networks, which may include semiautonomous or fully AVs, can be modeled as IoT systems with complex sensing and control capabilities. IoV is helpful not only in enhancing navigation and safety but also for enabling efficient traffic management. In order to achieve its intended functionalities, IoV requires anytime anywhere connectivity that cannot be satisfied by traditional networking technologies, and SAGIN is widely believed to be an ideal infrastructure for connecting IoV, and for supporting smart cities in general. However, security protection of IoV is a critical and yet challenging requirement.

This article presented an approach for understanding the security issues of complex IoT systems, and proposed an IoT security architecture for assessing security risks and addressing security requirements of IoV systems. We proposed an ANT-centric security reference architecture, which is based on the three architectural perspectives in studying IoT systems, namely, device, Internet, and semantic. In our security analysis, we adopted a systematic approach to analyze the security requirements and architectural design of IoT applications from each of the three different perspectives. This framework may be particularly useful to IoT system architects and system integrators. Based on the proposed generic architecture, the practitioners can flexibly design their own system architecture and select control measures commensurate with the notion of data sensitivity, activity criticality, and risk acceptance of their organizations.

In this proposal, we illustrated how the ANT-centric approach for the IoT security reference architecture helps to understand intended system features, connectivity of components (e.g., IoV and SAGIN), and the state, flow, and criticality of data throughout the system. The proposed architecture provides a structured and granular methodology to identify risks, realize the impact of those risks on crucial pressure points (critical nodes), and assign the appropriate level of security control to achieve desired end-to-end protection. The proposed security architecture captures all underlying components of IoV and SAGIN networks, and can be easily generalized as a security reference architecture for other smart city applications.

## Acknowledgment

This work also benefited from discussions at the Global Forum on Cyber Expertise (GFCE) Annual Meetings, and The Netherlands Organization for Applied Scientific Research (TNO) Singapore office, as part of its IoT Security Landscape Study initiative, commissioned by the Dutch government and CSA. The authors also received further valuable feedbacks from the Singapore International Cyber Week (SICW)'s International IoT Security Roundtable and the U.K.-based IoT Security Foundation (IoTSF) Annual Conference where the IoT Security Reference Architecture was discussed.

## References

[1] F. Li, K.-Y. Lam, M. Jia, K. Zhao, X. Li, and L. Wang, "Spectrum optimization for satellite communication systems with heterogeneous user preferences," *IEEE Syst. J.*, vol. 14, no. 2, pp. 2187–2191, Jun. 2020.

[2] F. Li, K.-Y. Lam, H.-H. Chen, and N. Zhao, "Spectral efficiency enhancement in satellite mobile communications: A game-theoretical approach," *IEEE Wireless Commun.*, vol. 27, no. 1, pp. 200–205, Feb. 2019.

[3] F. Li, K.-Y. Lam, X. Li, X. Liu, L. Wang, and V. C. M. Leung, "Dynamic spectrum access networks with heterogeneous users: How to price the spectrum?" *IEEE Trans. Veh. Technol.*, vol. 67, no. 6, pp. 5203–5216, Jun. 2018.

[4] F. Li, K.-Y. Lam, L. Meng, H. Luo, and L. Wang, "Trading-based dynamic spectrum access and allocation in cognitive Internet of Things," *IEEE Access*, vol. 7, pp. 125952–125959, 2019.

[5] F. Li, K.-Y. Lam, X. Li, Z. Sheng, J. Hua, and L. Wang, "Advances and emerging challenges in cognitive Internet-of-Things," *IEEE Trans. Ind. Informat.*, vol. 16, no. 8, pp. 5489–5496, Aug. 2020.

[6] M. B. Mollah *et al.*, "Blockchain for the Internet of Vehicles towards intelligent transportation systems: A survey," *IEEE Internet Things J.*, vol. 8, no. 6, pp. 4157–4185, Mar. 2021.

[7] J. Liu, Y. Shi, Z. M. Fadlullah, and N. Kato, "Space-air-ground integrated network: A survey," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 4, pp. 2714–2741, 4th Quart., 2018.

[8] N. Cheng *et al.*, "Air-ground integrated mobile edge networks: Architecture, challenges, and opportunities," *IEEE Commun. Mag.*, vol. 56, no. 8, pp. 26–32, Aug. 2018.

[9] L. Zheng *et al.*, "Technologies, applications, and governance in the Internet of Things," in *Internet of things—Global Technological and Societal Trends From Smart Environments and Spaces to Green ICT*. Aalborg, Denmark: River Publ., 2011.

[10] L.-M. Ang, K. P. Seng, G. K. Ijemaru, and A. M. Zungeru, "Deployment of IoV for smart cities: Applications, architecture, and challenges," *IEEE Access*, vol. 7, pp. 6473–6492, 2018.

[11] SAE International. (2018). *J3016B: Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles*. Accessed: Aug. 19, 2020. [Online]. Available: http://www.sae.org/standards/content/j3016_201806/

[12] J. Guerrero-Ibáñez, S. Zeadally, and J. Contreras-Castillo, "Sensor technologies for intelligent transportation systems," *Sensors*, vol. 18, no. 4, p. 1212, 2018.

[13] Y. Wang, Z. Su, Q. Xu, T. Yang, and N. Zhang, "A novel charging scheme for electric vehicles with smart communities in vehicular networks," *IEEE Trans. Veh. Technol.*, vol. 68, no. 9, pp. 8487–8501, Sep. 2019.

[14] P. H. L. Rettore, B. P. Santos, A. B. Campolina, L. A. Villas, and A. A. F. Loureiro, "Towards intra-vehicular sensor data fusion," in *Proc. IEEE 19th Int. Conf. Intell. Transp. Syst. (ITSC)*, 2016, pp. 126–131.

[15] F. Yang, S. Wang, J. Li, Z. Liu, and Q. Sun, "An overview of Internet of Vehicles," *China Commun.*, vol. 11, no. 10, pp. 1–15, Oct. 2014.

[16] Google. (2015). *Android Auto*. Accessed: Aug. 19, 2020. [Online]. Available: https://www.android.com/auto/

[17] Apple, Inc. (2014). *Apple CarPlay*. Accessed: Aug. 19, 2020. [Online]. Available: https://www.apple.com/sg/ios/carplay/

[18] F. Lyu *et al.*, "Characterizing urban vehicle-to-vehicle communications for reliable safety applications," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 6, pp. 2586–2602, Jun. 2020.

[19] M. R. Hafner, D. Cunningham, L. Caminiti, and D. Del Vecchio, "Cooperative collision avoidance at intersections: Algorithms and experiments," *IEEE Trans. Intell. Transp. Syst.*, vol. 14, no. 3, pp. 1162–1175, Sep. 2013.

[20] T. Taleb, A. Benslimane, and K. B. Letaief, "Toward an effective risk-conscious and collaborative vehicular collision avoidance system," *IEEE Trans. Veh. Technol.*, vol. 59, no. 3, pp. 1474–1486, Mar. 2010.

[21] W. Wu, Z. Yang, and K. Li, "Internet of Vehicles and applications," in *Internet of Things: Principles and Paradigms*. Cambridge, MA, USA: Morgan Kaufmann Publ., 2016, pp. 299–317.

[22] M. A. Salman, S. Ozdemir, and F. V. Celebi, "Fuzzy traffic control with vehicle-to-everything communication," *Sensors*, vol. 18, no. 2, p. 368, 2018.

[23] Y. He, D. Sun, M. Zhao, and S. Cheng, "Cooperative driving and lane changing modeling for connected vehicles in the vicinity of traffic signals: A cyber-physical perspective," *IEEE Access*, vol. 6, pp. 13891–13897, 2018.

[24] M. Eslami and K. Faez, "Automatic traffic monitoring using satellite images," in *Proc. 2nd Int. Conf. Comput. Eng. Technol.*, vol. 6, 2010, pp. V6-130–V6-135.

[25] W. Zhuang, Q. Ye, F. Lyu, N. Cheng, and J. Ren, "SDN/NFV-empowered future IoV with enhanced communication, computing, and caching," *Proc. IEEE*, vol. 108, no. 2, pp. 274–291, Feb. 2020.

[26] M. Gerla, E.-K. Lee, G. Pau, and U. Lee, "Internet of Vehicles: From intelligent grid to autonomous cars and vehicular clouds," in *Proc. IEEE World Forum Internet Things (WF-IoT)*, 2014, pp. 241–246.

[27] A. Burg, A. Chattopadhyay, and K.-Y. Lam, "Wireless communication and security issues for cyber–physical systems and the Internet-of-Things," *Proc. IEEE*, vol. 106, no. 1, pp. 38–60, Jan. 2018.

[28] J. Lin, W. Yu, N. Zhang, X. Yang, H. Zhang, and W. Zhao, "A survey on Internet of Things: Architecture, enabling technologies, security and privacy, and applications," *IEEE Internet Things J.*, vol. 4, no. 5, pp. 1125–1142, Oct. 2017.

[29] Y. Qian, K. Lu, and N. Moayeri, "A secure VANET MAC protocol for DSRC applications," in *Proc. IEEE GLOBECOM. Global Telecommun. Conf.*, 2008, pp. 1–5.

[30] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of Things (IoT): A vision, architectural elements, and future directions," *Future Gener. Comput. Syst.*, vol. 29, no. 7, pp. 1645–1660, 2013.

[31] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of Things: A survey on enabling technologies, protocols, and applications," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 4, pp. 2347–2376, 4th Quart., 2015.

[32] K. Boeckl *et al.*, "Considerations for managing Internet of Things (IoT) cybersecurity and privacy risks," U.S. Dept. Commerce, Nat. Inst. Stand. Technol., Gaithersburg, MD, USA, Rep. 8228, 2019.

[33] V. Gazis *et al.*, "Short paper: Iot: Challenges, projects, architectures," in *Proc. 18th Int. Conf. Intell. Next Gener. Netw.*, 2015, pp. 145–147.

[34] M. Fagan, K. Megas, K. Scarfone, and M. Smith, "Core cybersecurity feature baseline for securable IoT devices: A starting point for IoT device manufacturers," U.S. Dept. Commerce, Nat. Inst. Stand. Technol., Gaithersburg, MD, USA, Rep. 8259, 2019.

[35] K. B. Kelarestaghi, M. Foruhandeh, K. Heaslip, and R. Gerdes, "Intelligent transportation system security: Impact-oriented risk assessment of in-vehicle networks," *IEEE Intell. Transp. Syst. Mag.*, early access, Jan. 23, 2019, doi: 10.1109/MITS.2018.2889714.

[36] A. Chattopadhyay, K.-Y. Lam, and Y. Tavva, "Autonomous vehicle: Security by design," *IEEE Trans. Intell. Transp. Syst.*, early access, Jun. 30, 2020, doi: 10.1109/TITS.2020.3000797.

[37] F. Gondesen, S. Mitra, and K.-Y. Lam, "Feasibility of PUF-based authentication on ATtiny devices with off-the-shelf SRAM," in *Proc. 6th ACM Workshop Cyber-Phys. Syst. Security*, 2020, pp. 2–10.

[38] W. Feng, Y. Qin, S. Zhao, and D. Feng, "AAoT: Lightweight attestation and authentication of low-resource things in IoT and CPS," *Comput. Netw.*, vol. 134, pp. 167–182, Apr. 2018.

[39] Google. Cloud IoT Core. Accessed: Jan. 8, 2021. [Online]. Available: https://cloud.google.com/iot-core/

[40] IBM. *Watson IoT Platform*. Accessed: Jan. 8, 2021. https://www.ibm.com/cloud/watson-iot-platform

[41] L. Atzori, A. Iera, and G. Morabito, "The Internet of Things: A survey," *Comput. Netw.*, vol. 54, no. 15, pp. 2787–2805, 2010.

[42] M. Ammar, G. Russello, and B. Crispo, "Internet of Things: A survey on the security of IoT frameworks," *J. Inf. Security Appl.*, vol. 38, pp. 8–27, Feb. 2018.

[43] J. Contreras-Castillo, S. Zeadally, and J. A. Guerrero-Ibañez, "Internet of Vehicles: Architecture, protocols, and security," *IEEE Internet Things J.*, vol. 5, no. 5, pp. 3701–3709, Oct. 2018.

[44] N. Liu, *Internet of Vehicles: Your Next Connection*, vol. 11. Shenzhen, China: Huawei WinWin, 2011, pp. 23–28.

[45] K. Golestan, R. Soua, F. Karray, and M. S. Kamel, "Situation awareness within the context of connected cars: A comprehensive review and recent trends," *Inf. Fusion*, vol. 29, pp. 68–83, May 2016.

[46] J. Wan, D. Zhang, S. Zhao, L. T. Yang, and J. Lloret, "Context-aware vehicular cyber-physical systems with cloud support: Architecture, challenges, and solutions," *IEEE Commun. Mag.*, vol. 52, no. 8, pp. 106–113, Aug. 2014.

[47] European Union Agency for Cybersecurity. (2019). *Good Practices for Security of Smart Cars*. Accessed: Sep. 15, 2020. [Online]. Available: https://www.enisa.europa.eu/publications/smart-cars

[48] R. Roman, J. Lopez, and M. Mambo, "Mobile edge computing, fog et al.: A survey and analysis of security threats and challenges," *Future Gener. Comput. Syst.*, vol. 78, pp. 680–698, Jan. 2018.

[49] J. Petit and S. E. Shladover, "Potential cyberattacks on automated vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 2, pp. 546–556, Apr. 2015.

[50] D. Dominic, S. Chhawri, R. M. Eustice, D. Ma, and A. Weimerskirch, "Risk assessment for cooperative automated driving," in *Proc. 2nd ACM Workshop Cyber-Phys. Syst. Security Privacy*, 2016, pp. 47–58.

[51] M. T. Garip, M. E. Gursoy, P. Reiher, and M. Gerla, "Congestion attacks to autonomous cars using vehicular botnets," in *Proc. NDSS Workshop Security Emerg. Netw. Technol. (SENT)*, San Diego, CA, USA, 2015.

[52] S. M. Radack, "Standards for security categorization of federal information and information systems," U.S. Dept. Commerce, Nat. Inst. Stand. Technol., Gaithersburg, MD, USA, Rep. 199, 2004.

[53] SAE International. (2016). *J3016: Cybersecurity Guidebook for Cyber-Physical Vehicle Systems*. Accessed: Aug. 19, 2020. [Online]. Available: https://www.sae.org/standards/content/j3061

[54] International Organization for Standardization. (2018). *ISO 26262-1:2018: Road vehicles—Functional safety* Accessed: Sep. 14, 2020. [Online]. Available: https://www.iso.org/standard/68383.html

[55] K. McKay, L. Bassham, M. Sönmez Turan, and N. Mouha, "Report on lightweight cryptography," U.S. Dept. Commerce, Nat. Inst. Stand. Technol., Gaithersburg, MD, USA, Rep. 8114, 2016.

[56] C. F. Kerry and C. R. Director, "Federal information processing standards publication digital signature standard (DSS)," U.S. Dept. Commerce, Nat. Inst. Stand. Technol., Gaithersburg, MD, USA, Rep. 186-4, 2013.

[57] J. Delvaux, R. Peeters, D. Gu, and I. Verbauwhede, "A survey on lightweight entity authentication with strong PUFs," *ACM Comput. Surveys*, vol. 48, no. 2, p. 26, 2015.

[58] Y. Dodis, L. Reyzin, and A. Smith, "Fuzzy extractors: How to generate strong keys from biometrics and other noisy data," in *Proc. Int. Conf. Theory Appl. Cryptogr. Techn.*. 2004, pp. 523–540.

[59] A. Van Herrewege *et al.*, "Reverse fuzzy extractors: Enabling lightweight mutual authentication for PUF-enabled RFIDs," in *Proc. Int. Conf. Financ. Cryptogr. Data Security*, 2012, pp. 374–389.

[60] R. Maes, "Physically unclonable functions: Constructions, properties and applications," Ph.D. dissertation, Dept. Electr. Eng., Katholieke Universiteit Leuven, Leuven, Belgium, 2012.

[61] A. Perrig, D. Song, R. Canetti, J. Tygar, and B. Briscoe. (2005). *Timed Efficient Stream Loss-Tolerant Authentication (TESLA): Multicast Source Authentication Transform Introduction*. Accessed: Sep. 4, 2020. [Online]. Available: https://tools.ietf.org/html/rfc4082

**Kwok-Yan Lam** (Senior Member, IEEE) received the B.Sc. degree (First Class Hons.) in computer science from the University of London, London, U.K., in 1987, and the Ph.D. degree from the University of Cambridge, Cambridge, U.K., in 1990.

He is currently a Professor with the School of Computer Science and Engineering, Nanyang Technological University (NTU), Singapore. From August 2020, he is also on part-time secondment to the INTERPOL as a Consultant at Cyber and New Technology Innovation. Prior to joining NTU, he was a Professor with the Tsinghua University, Beijing, China, from 2002 to 2010, and has been a Faculty Member of the National University of Singapore, Singapore, and the University of London since 1990. He was a Visiting Scientist with the Isaac Newton Institute, University of Cambridge and a Visiting Professor with the European Institute for Systems Security, Karlsruhe, Germany. His research interests include distributed systems, IoT security infrastructure, distributed protocols for blockchain, biometric cryptography, homeland security, and cybersecurity.

Prof. Lam received the Singapore Foundation Award from the Japanese Chamber of Commerce and Industry, in 1998, for recognition of his R&D achievement in information security in Singapore.

**Sananda Mitra** received the B.Tech. degree in information technology and the M.Tech. degree in computer science and engineering from Maulana Abul Kalam Azad University of Technology, Kolkata, India, in 2009 and 2011, respectively.

She is currently a Senior Research Engineer with the Smart Platform Infrastructure Research on Integrative Technology (SPIRIT) Laboratory, School of Computer Science and Engineering, Nanyang Technological University (NTU), Singapore. Prior to joining NTU, she was an Assistant Professor with Techno International New Town, Kolkata, India, from 2011 to 2018. Her research interests include IoT, network security, wireless networks, and blockchain technology.

**Florian Gondesen** received the Diplom-Ingenieur degree in computer science and engineering from Hamburg University of Technology, Hamburg, Germany, in 2013.

He is currently a Senior Research Engineer with the Smart Platform Infrastructure Research on Integrative Technology (SPIRIT) Laboratory, School of Computer Science and Engineering, Nanyang Technological University (NTU), Singapore. Prior to joining NTU, he was a Research Associate with the Hamburg University of Technology from 2013 to 2018. His research interests include IoT security, biometrics, and brain–computer interfaces.

**Xun Yi** received the Ph.D. degree from Xidian University, Xi'an, China, in 1995.

He is currently a Professor of Computer Science and Software Engineering with RMIT University, Melbourne, VIC, Australia. He has published more than 200 articles with over 20 IEEE Transaction papers. His research areas include privacy preserving techniques, IoT and cloud security, blockchain, mobile security, secure electronic commerce, and applied cryptography.

Prof. Yi is a member of ARC College of Expert from 2017 to 2019 and an Associate Editor for IEEE Transactions on Dependable and Secure Computing from 2014 to 2018.