Secure attack detection framework for hierarchical 6G-enabled internet of vehicles

Hichem Sedjelmaci, Nesrine Kaaniche, Aymen Boudguiga, Nirwan Ansari

Secure Attack Detection Framework for Hierarchical 6G-enabled Internet of Vehicles

Hichem Sedjelmaci, [1] *Member, IEEE,* Nesrine Kaaniche[2], Aymen Boudguiga[3], Nirwan Ansari[4], *Fellow, IEEE*
[1] Ericsson, R&D Security, Massy Palaiseau, France
[2] SAMOVAR, Télécom SudParis, Institut Polytechnique de Paris, France
[3] Université Paris-Saclay, CEA-List, Palaiseau, France
[4] Advanced Networking Lab.,New Jersey Institute of Technology, Newark, NJ, USA

The Sixth Generation Heterogeneous Network (6G HetNet) is a global interconnected system that serves a myriad variety of applications and services across multiple domains such as satellite, air, ground, and underwater networks. It provides a platform for the development of novel Internet of Things (IoT) applications and services, particularly for the Internet of Vehicles (IoV), which encompasses all devices involved in intra-vehicle and inter-vehicle communications. However, this evolution towards a unified and huge cellular infrastructure creates new security challenges that require an intelligent attack detection framework to safeguard the network against cyber-security threats.

This paper proposes a hierarchical attack detection framework for 6G-enabled IoV. This framework relies on the processing capacities of edge nodes to satisfy the main 6G Key Performance Indicators (KPIs), such as trustworthiness, latency, connectivity, data rate and energy consumption. Federated Learning (FL) and non-cooperative gaming are used to train attack models and improve the detection process over time. The cooperative detection process based on FL is executed by security entities, IoV devices, edge servers and Security Information and Event Management (SIEM) to improve the detection accuracy over time. To harden the security of the proposed attack detection framework, a robust Stackelberg security game is developed to identify malicious IoV devices and edge servers, and select suitable IoV devices and edge servers to participate in the training and attack detection processes. The identification and selection process mainly relies on computing a reputation score based on the activities of these IoV devices and edge servers. As compared to current security monitoring and detection solutions, our framework balances detection accuracy and reduced network overhead, specifically as the system scales up, i.e., when the malicious traffic is high. In addition, it mitigates threats from both external and internal adversaries.

***Index Terms*—6G, Internet of Vehicles, Federated Learning, Stackelberg Game, Intrusion and attacks detection.**

I. Introduction

Telecommunication operators envision the 6G HetNet architecture to extend the 5G network by adding satellite and underwater networks, as presented in Fig. 1. As such the 6G network will encompass the following heterogeneous domains: (a) The underwater network will interconnect underwater devices such as wireless sensor nodes, underwater drones and submarines [1]–[3]. (b) The ground network will interconnect static and mobile nodes such as industry 5.0 sensors, actuators and processes, smart meters, and vehicles. These distributed IoT devices will use TeraHertz (THz) and millimeter Wave (mmWave) communications [2]. (c) The air network will interconnect flying devices, such as airplanes, Unmanned Aerial Vehicles (UAVs) and drones [4], [5]. To prevent disconnections and hence reduce the end-end network delay, the flying devices will establish wireless communication links with the devices deployed in the underwater and ground areas. (d) The satellite network will offer a variety of safety and infotainment services such as weather forecast, system navigation and television broadcasting. The main purpose of incorporating the satellite network in the 6G architecture is to enhance data broadcasting and forwarding at a large scale, while guaranteeing a high and reliable quality of service in a disaster-recovery communications network [6].

Some recent works have already discussed the main building blocks of the 6G architecture [3], [7], [8]. Letaief *et al.* [8] introduced a roadmap for 6G deployment and a set of AI-enabled use cases for 6G, targeting mainly QoS improvement. Gui *et al.* [3] described four communication services that will be supported by the 6G architecture: Massive Low Latency Machine Type communication (MLLMT), Mobile Broad Bandwidth and Low Latency communication (MBBLL), Massive Broad Bandwidth Machine Type communication (MBBMT) and 6G-Lite. 6G-Lite is dedicated for Cooperative Intelligent Transportation Systems (C-ITS) and Internet of Vehicles (IoV). Mao *et al.* [7] highlighted the main characteristics of 6G-enabled IoT applications. Note that in this work, we will be considering the use-case of 6G-enabled Internet of Vehicles (IoV) as a particular scenario of IoT. [7] investigated and analysed the use of Machine Learning (ML) algorithms for adapting radio resources from high-frequency bands to short-range ones.

As it will interconnect a huge number of heterogeneous devices and transport an ever-growing amount of data, the 6G architecture will raise new cyber-security challenges. For example, new breaches originating from the malicious use of virtualization functions targeting network functions employed for QoS improvement [9]. Indeed, 6G threats will target the edge intelligence and intelligent network management [10] (e.g., threats related to critical infrastructures and SDN/NFV) via the adoption of old attacks such as Denial of Services (DoS), man in the middle, or deception attacks [11]. Fortunately, integrating AI models by design in 6G will not

Corresponding author: Hichem Sedjelmaci (email: hichem.sedjelmaci@ericsson.com).

only enhance network management capabilities but will also serve to provide self-monitoring and self-healing from new threats [12]. In fact, by pushing AI elements the closest possible to data-owners or devices to reduce latency, 6G will indirectly provide a way to integrate efficiently intrusion detection systems using deep or collaborative learning.

AI-based intrusion detection systems in practice often rely on hybrid detection techniques [13], [14]. These systems combine detection rules (i.e., signatures) defined by security experts and machine learning techniques such as deep learning, reinforcement learning and Federated Learning (FL) [14]. As compared to conventional attack detection techniques, AI-based intrusion detection systems can potentially identify unknown attacks with high accuracy, and reduce the number of false positives and false negatives. In this paper, we propose a novel attack detection framework specifically designed for hierarchical 6G networks, with a focus on the IoV use-case. Our framework relies on two levels of FL involving a set of distributed IoV devices and edge servers, and a centralized security system (e.g.,Security Information and Event Management (SIEM)). The goal of our framework is to detect both internal and external attacks on the IoV network, while satisfying the main 6G Key Performance Indicators (KPIs), which include latency, connectivity degree, energy consumption, and data rate. Furthermore, to improve the detection rate, both IoV devices and edge servers run a security game based on the Stackelberg approach. This non-cooperative game involves a leader (i.e., a security agent) and a follower (i.e., an attacker), and only allows trusted IoV devices and trusted edge servers to participate in network monitoring. According to experimental results, our framework outperforms state-of-the-art solutions for intrusion detection solutions in 6G wireless networks in terms of attack detection accuracy and network overhead.

The remainder of this paper is organized are follows. Section II reviews the related works and introduces security and functional requirements and Section III details the adversary model. Section IV presents the proposed intrusion detection framework. Section V discusses the implementation results of the IoV use case before concluding in Section VI.

II. Cyber Security in 6G Wireless Networks

In this section, we present state-of-the-art security solutions to protect the 6G network from attackers. The advantage and weakness of each solution is highlighted.

A. Application and data levels

Li *et al.* [2] conceived a blockchain solution to secure the AI applications of the 6G network from tampering threats using a reputation metric. This solution relies on four components: distributed edge servers, a blockchain module, a trusted entity, and users' equipment. These components cooperate together to prevent external attacks. Mousa *et al.* [15] proposed an authentication scheme based on robust public key encryption to secure the 6G wireless network from external attacks, such as man in the middle and spoofing attacks. The user equipment, base station, and authentication server run the authentication protocol and only the authenticated equipment is allowed to join the network. In the security analysis, they proved that their scheme is robust against network attacks, while being lightweight. Li *et al.* [16] developed an efficient edge caching system for 6G architecture to thwart attacks targeting data confidentiality. This caching system relies on physical layer security mechanisms and a probabilistic caching model. Although the aforementioned security mechanisms [2], [15], [16] are robust against external threats, they fail to detect internal attacks, such as infected virtual networking functions, and compromised fog and cloud servers.

B. Network level

Stergiou *et al.* [17] presented a novel approach to protect 6G wireless networks from attacks targeting IoT, cloud computing, and fog devices. They have proposed a data management mechanism that employs a cache decision system to reduce end-to-end latency and ensure energy efficiency. This system is monitored by both centralized cloud and distributed edge servers, to detect and prevent known attacks through different edge points. On the other hand, Liu *et al.* [18] applied FL algorithms in 6G communications to address privacy concerns. They introduced three defense modules: an aggregation algorithm to collaboratively aggregate updates, a reputation framework to select relevant and honest entities, and a detection system. However, their solution has not been evaluated through simulations. Mao *et al.* [7] developed an intrusion detection system that uses Kalman filters to detect attackers that target the energy consumption of IoT devices. They assessed the trade-off between energy consumption and quality of service, and demonstrated that their simulated solution ensures low computation overhead and high network throughput. Nevertheless, their detection system has not been evaluated against attacks that occur in constrained IoT networks, namely, targeting connectivity and network latency. Finally, Zhou *et al.* [19] addressed task offloading in the context of 6G networks. They have proposed a multi-layer solution that provides efficient data computation and transmission, and secure computation services at the user's side. Their approach includes a quantitative security analysis model to assess the trust level of the task offloading model implemented by user equipment.

C. Key Performance Indicators (KPIs)

Table I presents a comparison among the current cyber security solutions applied in the 6G architecture in terms of attack detection performances and their impact on the main 6G KPIs. In the following, we highlight the main considered 6G KPIs and examples of attacks impacting those KPIs.

- *Energy:* Various network attacks can target energy-constrained devices in a 6G network. The resource exhaustion attack is an example of these attacks, which consists of forcing the victim device to perform computationally expensive tasks that leads to its energy depletion.
- *Connectivity degree:* The connectivity degree is considered as a main KPI of MBBMT services because the distributed IoT devices in MBBMT require a massive connectivity in order to share and process a huge amount

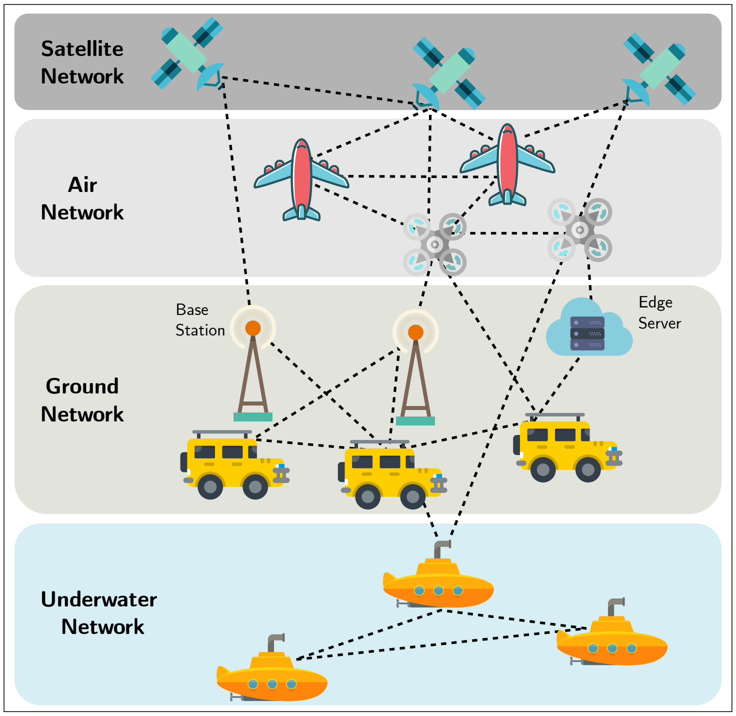


Fig. 1: Architecture of 6G Enabled Networks

TABLE I: Comparison among different cybersecurity solutions for 6G-enabled IOT applications

Solution	Approach	Internal attacks	External attacks	Accuracy of detection	Energy consumption	Connectivity	Latency	Data rate
[2]	Blockchain	Non detectable	Detectable	Medium	High	Medium	High	N.A.
[16]	Cryptographic-based solution	Non detectable	Detectable	Medium	High	Medium	Low	N.A.
[15]	Caching system	Non detectable	Detectable	Medium	N.A.	N.A.	High	N.A.
[17]	Caching system	Detectable	Detectable	Medium	Medium	Medium	Low	High
[18]	Federated Learning (FL)	Detectable	Detectable	N.A.	N.A.	High	Low	High
[7]	Kelman Filtering	N.A.	Detectable	High	Medium	N.A.	Low	High
[19]	Rules-based attacks detection	N.A.	Detectable	N.A.	Low	High	Low	High
Proposed solution	Hierarchical FL	Detectable	Detectable	High	Medium	Medium	Low	High

NOTE: N.A. is the abbreviation for Not Applicable

of data [7]. Mao *et al.* [7] noticed that positioning accuracy could be one of the parameters used to enhance the quality of connectivity in the 6G network. However, positioning accuracy can be the target of GPS spoofing attack [20].

- *Network Latency:* Intelligent transportation systems such as vehicular ad-hoc network and drones network are ones of the 6G-Lite scenarios, where the latency is a main KPI. Remote cyber-attacks can target network latency by infecting mobile devices (such as automotive and drone nodes). Once infected, they start running unnecessary computations and sharing extra information which increases the end-to end network latency.
- *Data rate:* Augmented reality and virtual reality are main MBBLL applications of 6G-enabled IoT network. To achieve a high quality of service and high quality of experience, the data rate and latency should be taken into account [7]. However, attacks such as Black Hole and Selective forwarding attacks drop relevant network packets, thus leading to a rapid decrease of data rate, and subsequently causing an increase on the network latency.

Several other KPIs are important in the context of the Internet of Vehicles (IoV). In addition to energy, connectivity, latency, and data rate, the Quality of Experience (QoE) is another indictor driving the implementation of mission-critical applications in 6G-enabled IoV, as explained in [5]. Furthermore, given different personalized needs of vehicles for computing-intensive applications, how to utilize heterogeneous computing resources in 6G networks to ensure personalized QoE for vehicles has become a challenge. In the context of 6G edge computing offloading, QoE depends on the computation overhead. The proposed security architecture utilizes QoE to address the security concerns associated with offloading and computation tasks. Furthermore, the trustworthiness of the edge nodes and IoV devices is taken into consideration during the offloading and computation process. The QoE metric is inherently integrated into the task offloading procedure, in conjunction with the trust metric that reflects the level of trust for both the IoV device and the edge server.

III. Security Model

In this work, we consider a Dolev and Yao attacker model [21]. That is, the attacker is able to Read, Drop and Send valid messages. A Read action refers to receiving or intercepting messages. Meanwhile, a Send action refers to forging and replaying messages. A Drop action refers to filtering. In practice, we consider two types of adversaries defined as follows:

- *External adversaries* — malicious entities that seek to hijack main security properties such as data confidentiality or integrity, as well as network service availability. They may also include other form of attacks, such as the one aiming to deteriorate the featured 6G KPIs, by increasing latency, exhausting network resources, generating overhead, or increasing the energy footprint (e.g., altering the signal strength intensities). Examples of external attacks include DoS, man in the middle, black hole, eavesdropping and poisoning. Some of these attacks can be targeting 6G virtualisation functions, network configuration modules and AI algorithms for QoS management.
- *Internal adversaries* — malicious IoV devices and malicious edge servers. Legitimate devices can become malicious after the installation of a malware or a fake update. Once infected, malicious devices will target network configuration and ML configuration [10]. In addition, they can start misbehaving by dropping neighbours packets and injecting malicious messages.

Both external and internal adversaries are capable of executing known and unknown attacks. In a known attack, the adversary exhibits a known malicious action defined by an attack signature. They refer to the security threats that have already been identified within the security community. In this context, known attacks may include traditional attack vectors, such as malware, phishing, denial-of-service (DoS) attacks, and ransomware. However, in an unknown attack, also known as a zero-day exploit attack, the adversary launches an attack that has not been executed before (i.e., new malicious action) or at least has not reported yet. These attacks are often more difficult to defend against because there is no existing knowledge for them. To protect against unknown attacks, the security model should include measures for identifying and responding to security incidents. By addressing both known and unknown attacks in the security model, we establish a proactive approach to securing the system against a range of potential threats.

IV. Detection Framework for Hierarchical IoV Networks in the 6G era

In this section, we propose a cyber-detection framework for the 6G network. It prevents the occurrence of internal and external threats, e.g., attacks targeting wireless communications and private data. The proposed framework leverages a collaborative monitoring and detection performed by security entities deployed in a hierarchical way within the network. In the following subsections, we first introduce the considered architecture by presenting the involved entities and detailing the attack detection process (Section IV-A). Then, we present a new trust model based on a non-cooperative game (relying on the Stackelberg security game) for the detection of non-trusted security entities (Section IV-B).

A. Architecture

Our solution applies federated learning between all the involved security entities to train attack detection models to detect malicious behaviors at different levels of the network architecture.

1) Entities

Our solution relies on three types of security entities for securing the network from attackers: IoV devices, edge devices and a Security Information and Event Management (SIEM), as illustrated in Fig. 2. These entities collaborate together during the monitoring and detection process as follows:

- ***IoV devices*** are denoted by $\mathcal{D}$. Two kinds of IoV devices are defined, Cluster Member ($\mathcal{CM}$) and Cluster Head ($\mathcal{CH}$) devices. Each $\mathcal{CH}$ manages a set of $\mathcal{CM}$s, and is elected by different members, according to its performances indicators and its Maliciousness Degree (MD). For more details on how to compute the MD for the IoV network, we refer readers to Reference [22]. The MD and aforementioned 6G's KPIs are the main selectors for the $\mathcal{CH}$'s election. For example, the device that exhibits a low MD (as compared to its neighboring devices), while considering the 6G's KPIs will be elected as $\mathcal{CH}$. The latter runs the hierarchical FL-security framework (as explained in subsection IV.A.2) to detect the malicious $\mathcal{CM}$, located within its neighborhood. When a malicious $\mathcal{CM}$ is detected, an Alert message is sent from the monitoring $\mathcal{CH}$ to SIEM (through the edge server) for further investigation. This Alert message contains the identity of the malicious $\mathcal{CM}$, and the values of its KPIs. Note that the $\mathcal{CH}$ could also act as a malicious device. Consequently, to overcome this security issue, $\mathcal{CM}$ devices monitor the behavior of their $\mathcal{CH}$ by running a rule-based attack detection technique. In our security architecture as illustrated in Fig. 2, the $\mathcal{CH}$ responsible for managing its set of cluster members is ideally located within the range of an edge server. However, in cases when the $\mathcal{CH}$ is not within the range of the edge server, a cluster head election is launched to select a new $\mathcal{CH}$ close to the edge (i.e., a device that has a high connectivity degree). Furthermore, the device with a lower malicious degree (as compared to its neighboring devices), while taking into account the 6G's KPIs, particularly the connectivity degree, will be elected as the new $\mathcal{CH}$.
- ***Edge Server*** is denoted by $\mathcal{E}$. It is a powerful device that monitors the behavior of $\mathcal{CH}$ located within its range. It also analyzes the Alert messages received from the $\mathcal{CH}$ device to confirm or refute the malicious behaviors of suspected $\mathcal{CM}$ devices. The edge device runs an attack detection technique using the FL algorithm (as explained in subsection IV-A.2). The edge server sends a Report message to SIEM for further investigation in order to provide final decisions (i.e., confirm or reject the malicious behavior of a suspected $\mathcal{CM}$ and $\mathcal{CH}$). This Report message contains the identities of suspected $\mathcal{CH}$s and $\mathcal{CM}$s, and their related KPIs and MDs.
- ***SIEM*** is denoted by $\mathcal{S}$. It is a trusted cloud server that monitors the behaviors of distributed edge servers to detect malicious ones. In addition, it analyzes the Alert and Report messages to verify the malicious behaviors of suspected $\mathcal{CM}$s and $\mathcal{CH}$s. First, $\mathcal{S}$ performs the ag-

gregation process, which consists of aggregating all the relevant information (MDs and KPIs) related to suspected $\mathcal{E}$s, $\mathcal{CH}$s and $\mathcal{CM}$s. Then, the updated values of MDs and KPIs are used as inputs for the FL algorithm to accurately detect malicious $\mathcal{E}$s, $\mathcal{CH}$s and $\mathcal{CM}$s.

2) Hierarchical FL-security framework Workflow

FL introduced by Google in 2016 [23], is a collaborative machine learning approach that enables multiple clients to collectively train a shared model without revealing their sensitive local data to a central aggregation server. In FL, each client trains its own machine learning model locally by using its own data. Afterwards, the resulting model weights are transmitted to an aggregation server. The server then combines all the clients' models, typically by averaging their weights. Finally, the updated global model is distributed back to all the clients. This iterative training process is repeated multiple times. FL is intriguing because it safeguards the confidentiality of clients' sensitive data by keeping them undisclosed. Moreover, it allows for the expansion of the training dataset since the dataset for training the global model can be perceived as an aggregation of all the clients' datasets. FL has found effective applications in various domains, such as facilitating collaborative training of disease detection models among hospitals without exposing patient health data or enabling security companies to collaborate on training threat detection models while maintaining the privacy of their attack repositories and logs.

We present in this section our FL-based framework for intrusion detection in a hierarchical 6G-IoV network (Fig 2).

The FL algorithm based on an unsupervised approach is divided into three phases: clustering, training and detection. During the clustering process, a trusted and reliable $\mathcal{CH}$ device is elected based on the monitored MDs and KPIs. It corresponds to the $\mathcal{CM}$ that presents a low MD with interesting KPIs (i.e., low energy consumption and network latency; high connectivity degree and data rate). The elected $\mathcal{CH}$s share their training models and the list of malicious $\mathcal{CM}$s with their neighboring edge server. For the training process, we assume that, at time t, all the $\mathcal{CH}$s and their associated $\mathcal{CM}$s possess the same global training model ψ_t, which is also shared with the edge servers. We denote by:

- ψ_t^i the global model at $\mathcal{E}$, where $i = \{1, \ldots, m\}$ and m is the total number of involved edge servers,
- ψ_t^j the global model at $\mathcal{CH}$, where $j = \{1, \ldots, m'\}$ and m' is the number of $\mathcal{CH}$s located within the range of each edge server.

First, each $\mathcal{CH}$ gathers data from its associated $\mathcal{CM}$s. Then, it trains a local model ψ_{t+1}^j. Next, $\mathcal{CH}$ uploads its local training model ψ_{t+1}^j's updates to the edge server $\mathcal{E}$. $\mathcal{E}$ aggregates the local training models of its $\mathcal{CH}$s to obtain the updated global training model ψ_{t+1}^i. Afterwards, $\mathcal{S}$ receives the ψ_{t+1}^i's updates from $\mathcal{E}$s, and aggregates them to obtain a global model. The latter is also trained with $\mathcal{S}$'s local data to obtain the final global model ψ_{t+1}. At the end of this iteration, ψ_{t+1} is shared with all the framework's entities.

During the detection phase, the updated training model ψ_{t+1} (i.e., anomaly detection model) is used by $\mathcal{CH}$s and $\mathcal{E}$ to detect malicious behaviors and monitor their respective targets. Alert and Report messages are sent to $\mathcal{S}$ for further investigation and verification.

The proposed framework consists of a two-layer system that gathers heterogeneous entities with different trust levels. The IoV layer involves several groups of $\mathcal{CM}$s, where each group is managed by a selected $\mathcal{CH}$, as explained in Section IV. The exchanged data flows between $\mathcal{CM}$s and $\mathcal{CH}$ include sensitive information such as logs information, alerts, etc..., that may expose the vulnerabilities of the vehicles' devices. In order to avoid this issue, our framework proposes a combination of a symmetric encryption scheme and a signature mechanism to ensure the confidentiality and integrity of the exchanged data between devices. We rely on the Advanced Encryption Standard (AES) as a symmetric encryption scheme, which is a well-established block-cipher. The communicating parties within the same cluster share the same encryption key k. Before sharing data with $\mathcal{CH}$, each cluster member $\mathcal{CM}_l$ encrypts them as $C_l = Enc_{AES}(D_l, k)$ where $l = \{1, \ldots, m"\}$, $m"$ is the number of $\mathcal{CM}$s whithin a particular cluster, and D_l is the collected data by $\mathcal{CM}_l$. Then, the enciphered data C_l will be signed by $\mathcal{CM}_l$, using an asymmetric encryption scheme such as Elliptic Curve Digital Signature Algorithm (ECDSA) [24] and its private key denoted by $sk_{\mathcal{CM}_l}$. The resulting signature is $\sigma_{\mathcal{CM}_l}$. A lightweight alternative to the signature is computing a Hashed Message Authentication Code (HMAC) on C_l. Computing HMAC requires sharing a symmetric key between $\mathcal{CM}_l$ and $\mathcal{CH}$. The edge level includes different edge servers $\mathcal{E}$s connected to the SIEM. $\mathcal{E}$s share models' updates and relay Alert and Report messages that include sensitive information that can enable external adversaries to conduct various attacks [25], [26] such as membership inference attacks, reconstruction and inversion attacks, *etc.* To mitigate these issues, our framework proposes the same combination of encryption and signatures schemes, as in the IoV level, applied at each edge server $\mathcal{E}$. This combination guarantees the confidentiality and integrity of exchanged data.

B. Stackelberg game for the detection of non-trusted devices

The proposed framework relies on a Stackelberg security game [27] to identify malicious $\mathcal{CH}$s and $\mathcal{E}$s. It is a non-cooperative game between the leader player (a security agent) and the follower player (an opponent agent, i.e., attacker) [27]. The principle of the Stackelberg security game entails the leader player aiming to detect the maximum number of opponent agents, and the follower player setting to launch attacks without being detected by the security agents.

1) Security game definition

As detailed in Section IV, the $\mathcal{CH}$ is elected based on its MD and KPIs. However, the MD value can be altered and cannot be considered as reliable. Thereby, to elect the most trusted $\mathcal{CH}$s and edge servers $\mathcal{E}$, two security games are set-up: (i) $\mathcal{E}$ and $\mathcal{CH}$ play respectively the roles of leader and follower players, where the goal of the leader is to detect the non-trusted $\mathcal{CH}$, and (ii) the SIEM $\mathcal{S}$ and edge server $\mathcal{E}$ play respectively the roles of leader and follower players, where the goal of the leader is to detect the non-trusted server.

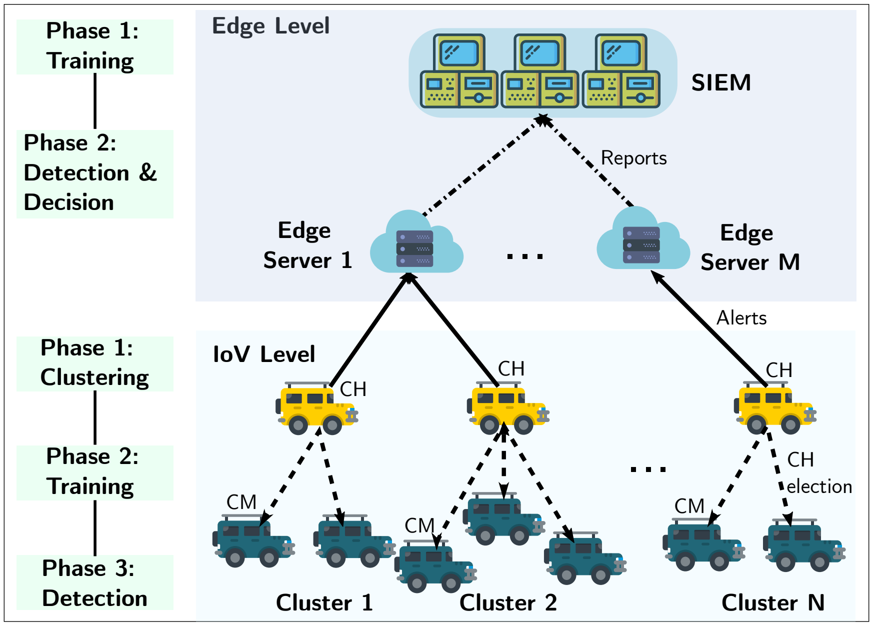


Fig. 2: FL-based defense architecture

The Stackelberg security games are denoted by: $\{\theta^{\mathcal{CH}}_{(i,t)}, \theta^{\mathcal{E}}_{(i,t)}, \theta'^{\mathcal{E}}_{(i,t)}, \theta'^{\mathcal{S}}_{(i,t)}\}$, where $i \in \{1, \ldots, I\}$ and I is the maximum number of iterations considered in the game; $\theta^{\mathcal{CH}}_{(i,t)}, \theta^{\mathcal{E}}_{(i,t)}, \theta'^{\mathcal{E}}_{(i,t)}, \theta'^{\mathcal{S}}_{(i,t)}$ correspond to the follower $\mathcal{CH}$, the leader edge $\mathcal{E}$, the follower edge $\mathcal{E}$, and the leader SIEM $\mathcal{S}$, respectively.

In the following, we denote by $\phi^1_i(t), \phi^2_i(t), \phi^3_i(t)$ and $\phi^4_i(t)$, the number of malicious entities and i is the ith iteration of the game: $\phi^1_i(t)$ is the number of malicious $\mathcal{CH}$s that are located in the neighborhood of leader edge server $\mathcal{E}$, $\phi^2_i(t)$ corresponds to the number of malicious $\mathcal{CH}$s detected by leader edge server, $\phi^3_i(t)$ is the number of malicious edge servers that are located in the neighborhood of leader SIEM $\mathcal{S}$, $\phi^4_i(t)$ is the number of malicious edge servers detected by the leader SIEM.

In addition, we refer to the pure strategies of the different leader and follower players by $S^1_i(t)$, $S^2_i(t)$, $S^3_i(t)$ and $S^4_i(t)$, where i is the ith iteration of the game. These strategies are represented by $S^1_i(t) = s^1_i(t) \in \phi^1_i(t)$, $S^2_i(t) = s^2_i(t) \in \phi^2_i(t)$, $S^3_i(t) = s^3_i(t) \in \phi^3_i(t)$ and $S^4_i(t) = s^4_i(t) \in \phi^4_i(t)$, where $s^1_i(t), s^2_i(t)$, $s^3_i(t)$ and $s^4_i(t)$ correspond respectively to a malicious behavior executed by a $\mathcal{CH}$ node, a malicious $\mathcal{CH}$ detected by $\mathcal{E}$, a malicious behavior executed by an edge server, and a malicious edge server detected by $\mathcal{S}$.

To evaluate the success of different security games, we consider p^1_i as the probability of follower $\mathcal{CH}$ adopting the strategy $s^1_i(t)$, p^2_i as the probability of the leader edge adopting the strategy $s^2_i(t)$, p^3_i as the probability of the follower $\mathcal{E}$ adopting the strategy $s^3_i(t)$, and p^4_i as the probability of the leader $\mathcal{S}$ adopting the strategy $s^4_i(t)$. Note that $\sum_{i=1}^{I} p^1_i = 1$, $\sum_{i=1}^{I} p^2_i = 1$, $\sum_{i=1}^{I} p^3_i = 1$, and $\sum_{i=1}^{I} p^4_i = 1$.

2) *Utility*

The utility functions of the leader and follower players depend on their rewards and required costs. Leaders' rewards correspond to the detection accuracy of both malicious $\mathcal{CH}$s and $\mathcal{E}$s, and the leaders' costs are the amounts of computation and communication overheads required by leader players to reach a high detection accuracy. Followers' rewards are the attacks success rates of both malicious $\mathcal{CH}$s and $\mathcal{E}$s, while followers' costs are the required computation overhead required by both malicious $\mathcal{CH}$s and $\mathcal{E}$s to initiate attacks without being detected by leaders. The utility functions of leader and follower players are defined in Eqs. (1), (4), and (7), respectively.

$$\begin{aligned} & U^{\mathcal{E}}_{(i,t)}(S^2_i(t), S^1_i(t)) \\ &= R^{\mathcal{E}}_{(i,t)}(S^2_i(t), S^1_i(t)) - C^{\mathcal{E}}_{(i,t)}(S^2_i(t), S^1_i(t)), \end{aligned} \tag{1}$$

where $R^{\mathcal{E}}_(i,t)$ and $C^{\mathcal{E}}_(i,t)$ correspond to the reward and the generated cost of the leader edge server, respectively, which are computed according to Eq. 2.

$$\begin{aligned} & R^{\mathcal{E}}_{(i,t)}(S^2_i(t), S^1_i(t)) = D^{\mathcal{E}}_{(i,t)}(S^2_i(t), S^1_i(t)) \\ & -[F^{\mathcal{E}}_{(i,t)}(S^2_i(t), S^1_i(t)) + F'^{\mathcal{E}}_{(i,t)}(S^2_i(t), S^1_i(t))], \end{aligned} \tag{2}$$

$D^{\mathcal{E}}_{(i,t)}$ is the detection rate of malicious $\mathcal{CH}$, and $F^{\mathcal{E}}_{(i,t)}$ and $F'^{\mathcal{E}}_{(i,t)}$ correspond respectively to the false positive and false negative rates against a legitimate $\mathcal{CH}$, calculated by the edge server. Note that $D^{\mathcal{E}}_{(i,t)}$, $F^{\mathcal{E}}_{(i,t)}$ and $F'^{\mathcal{E}}_{(i,t)} \in [0, 1]$.

$$\begin{aligned} & C^{\mathcal{E}}_{(i,t)}(S^2_i(t), S^1_i(t)) \\ &= \alpha_1.O^{\mathcal{E}}_{(i,t)}(S^2_i(t), S^1_i(t)) - \alpha_2.O'^{\mathcal{E}}_{(i,t)}(S^2_i(t), S^1_i(t)), \end{aligned} \tag{3}$$

$O^{\mathcal{E}}$ and $O'^{\mathcal{E}}$ are respectively the computation and communication overheads generated by the edge server to detect accurately the malicious behaviors of a monitored $\mathcal{CH}$. Note that $C^{\mathcal{E}}_{(i,t)} \in [0, 1]$ and the weights α^1 and α^2 are in $[0, 1]$.

$$U^{\mathcal{S}}_{(i,t)}(S^4_i(t), S^3_i(t))$$

$$= R^{\mathcal{S}}_{(i,t)}(S_i^4(t), S_i^3(t)) - C^{\mathcal{S}}_{(i,t)}(S_i^4(t), S_i^3(t)), \tag{4}$$

Similar to $\mathcal{E}$, we, hereafter, express both the reward and the cost of the SIEM $\mathcal{S}$, as follows:

$$R^{\mathcal{S}}_{(i,t)}(S_i^4(t), S_i^3(t)) = D^{\mathcal{S}}_{(i,t)}(S_i^4(t), S_i^3(t))$$
$$-[F^{\mathcal{S}}_{(i,t)}(S_i^4(t), S_i^3(t)) + F'^{\mathcal{S}}_{(i,t)}(S_i^4(t), S_i^3(t))], \tag{5}$$

$$C^{\mathcal{S}}_{(i,t)}(S_i^4(t), S_i^3(t))$$
$$= \alpha_3.O^{\mathcal{S}}_{(i,t)}(S_i^4(t), S_i^3(t)) - \alpha_4.O'^{\mathcal{S}}_{(i,t)}(S_i^4(t), S_i^3(t)), \tag{6}$$

$D^{\mathcal{S}}_{(i,t)}$, $F^{\mathcal{S}}_{(i,t)}$ and $F'^{\mathcal{S}}_{(i,t)} \in [0,1]$ correspond respectively to the detection rate of a malicious edge server, false positive rate and false negative rate. $C^{\mathcal{S}}_{(i,t)} \in [0,1]$ is the required computation and communication overheads to accurately detect the malicious behaviors of edge server, where $\alpha_3, \alpha_4 \in [0,1]$. In the following, we denote by $\mathcal{F}$ and $\mathcal{L}$ the follower and the leader players, respectively.

$$U^{\mathcal{F}}_{(i,t)}(S_i^{\mathcal{F}}(t), S_i^{\mathcal{L}}(t))$$
$$= R^{\mathcal{F}}_{(i,t)}(S_i^{\mathcal{F}}(t), S_i^{\mathcal{L}}(t)) - C^{\mathcal{F}}_{(i,t)}(S_i^{\mathcal{F}}(t), S_i^{\mathcal{L}}(t)), \tag{7}$$

Here, $S_i^{\mathcal{F}}(t)$ could be either $S_i^1(t)$ or $S_i^3(t)$, while $S_i^{\mathcal{L}}(t)$ could be either $S_i^2(t)$ or $S_i^4(t)$.

$$R^{\mathcal{F}} = \mathcal{CH}_{(i,t)} = -R^{\mathcal{E}}_{(i,t)}(S_i^2(t), S_i^1(t)), \tag{8}$$

$$R^{\mathcal{F}} = \mathcal{E}_{(i,t)} = -R^{\mathcal{S}}_{(i,t)}(S_i^4(t), S_i^3(t)), \tag{9}$$

$$C^{\mathcal{F}}_{(i,t)}(S_i^{\mathcal{F}}(t), S_i^{\mathcal{L}}(t)) = \beta.O^{\mathcal{F}}_{(i,t)}(S_i^{\mathcal{F}}(t), S_i^{\mathcal{L}}(t)), \tag{10}$$

where $O^{\mathcal{F}}_{(i,t)}$ is the computed overhead generated by a malicious follower player (i.e., either a malicious $\mathcal{CH}$ or a malicious $\mathcal{E}$) to execute malicious behaviors without being detected by the leader player (i.e., edge server $\mathcal{E}$ or SIEM $\mathcal{S}$). Note that $\beta \in [0,1]$ is the weight parameter.

3) *Stackelberg security equilibrium solution*

In order to reach an equilibrium, we first recall that the strategies of leader players, $\theta^{\mathcal{E}}_{(i,t)}$ and $\theta^{\mathcal{S}}_{(i,t)}$, depend on the strategies of follower players, $\theta^{\mathcal{CH}}_{(i,t)}$ and $\theta^{\mathcal{E}}_{(i,t)}$, at each iteration i and subsequent iterations I, and vice-versa. Thus, as shown in [28], the Stackelberg equilibrium is determined, recursively. The optimal utility functions of the players at the equilibrium point are defined as $\min$ and $\max$ functions below.

$$U^{*\mathcal{E}}_{(i,t)}(S_i^{*2}(t), S_i^{*1}(t))$$
$$= \max_{p_i^2} \min_{p_i^1} Q^1.U^{\mathcal{E}}_{(i,t)}(S_i^2(t), S_i^1(t)), \tag{11}$$

$$U^{*\mathcal{S}}_{(i,t)}(S_i^{*4}(t), S_i^{*3}(t))$$
$$= \max_{p_i^4} \min_{p_i^3} Q^2.U^{\mathcal{S}}_{(i,t)}(S_i^4(t), S_i^3(t)), \tag{12}$$

$$U^{*\mathcal{F}}_{(i,t)}(S_i^{*\mathcal{F}}(t), S_i^{*\mathcal{L}}(t))$$
$$= \max_{p_i^{\mathcal{F}}} \min_{p_i^{\mathcal{L}}} Q^{\mathcal{F}}.U^{\mathcal{F}}_{(i,t)}(S_i^{\mathcal{F}}(t), S_i^{\mathcal{L}}(t)), \tag{13}$$

where $Q^1 = p_i^2.p_i^1$ and $Q^2 = p_i^4.p_i^3$. In Eq. 13, when $S_i^{\mathcal{F}}(t)$ is $S_i^1(t)$, $S_i^{\mathcal{L}}(t)$ will be $S_i^2(t)$. Hence, $p_i^{\mathcal{F}} = p_i^1$, $p_i^{\mathcal{L}} = p_i^2$ and $Q^{\mathcal{F}} = p_i^1.p_i^2 = Q^1$. In addition, when $S_i^{\mathcal{F}}(t)$ is $S_i^3(t)$, $S_i^{\mathcal{L}}(t)$ will be $S_i^4(t)$. Hence, $p_i^{\mathcal{F}} = p_i^3$, $p_i^{\mathcal{L}} = p_i^4$ and $Q^{\mathcal{F}} = p_i^3.p_i^4 = Q^2$.

The total utility functions of the leader and follower players are computed as shown in Eqs. (14)-(16).

$$U^{\mathcal{E}}_{Total}(S_i^2(t), S_i^1(t)) =$$
$$Q^1.[\sum_{i=1}^{I} U^{*\mathcal{E}}_{(i,t+1)}(S_i^{*2}(t+1), S_i^{*1}(t+1)) + \sum_{i=1}^{I} U^{\mathcal{E}}_{(i,t)}(S_i^2(t), S_i^1(t))], \tag{14}$$

$$U^{\mathcal{S}}{}_{Total}(S_i^4(t), S_i^3(t)) =$$
$$Q^2.[\sum_{i=1}^{I} U^{*\mathcal{S}}_{(i,t+1)}(S_i^{*4}(t+1), S_i^{*3}(t+1)) + \sum_{i=1}^{I} U^{\mathcal{S}}_{(i,t)}(S_i^4(t), S_i^3(t))], \tag{15}$$

$$U^{\mathcal{F}}_{Total}(S_i^{\mathcal{F}}(t), S_i^{\mathcal{L}}(t)) =$$
$$Q^{\mathcal{F}}.[\sum_{i=1}^{I} U^{*\mathcal{F}}_{(i,t+1)}(S_i^{*\mathcal{F}}(t+1), S_i^{*\mathcal{L}}(t+1)) + \sum_{i=1}^{I} U^{\mathcal{F}}_{(i,t)}(S_i^{\mathcal{F}}(t), S_i^{\mathcal{L}}(t))], \tag{16}$$

As shown in Eqs. (17) and (18), the leader players, $\theta^{\mathcal{E}}_{(i,t)}$ and $\theta^{\mathcal{S}}_{(i,t)}$, aim to maximize their total utility functions, while considering the best responses from follower players (as defined in Eq. (16)).

$$\forall S'^1_i(t), \forall p'^1_i, \max_{p_i^2} \min_{p_i'^1} U^{\mathcal{E}}_{Total}(S_i^2(t), S'^1_i(t)) \tag{17}$$

such that:

$$Q^1.[\sum_{i=1}^{I} U^{*\mathcal{E}}_{(i,t+1)}(S_i^{*2}(t+1), S_i^{*1}(t+1)) + \sum_{i=1}^{I} U^{\mathcal{E}}{}_{(i,t)}(S_i^2(t), S_i^1(t))]$$
$$\leq Q'^1.[\sum_{i=1}^{I} U^{*\mathcal{E}}_{(i,t+1)}(S_i^{*2}(t+1), S_i^{*1}(t+1)) + \sum_{i=1}^{I} U^{\mathcal{E}}_{(i,t)}(S_i^2(t), S_i'^1(t))]$$

where $Q'^1 = p_i^2.p'^1_i$ and $p'^1_i \geq p^1{}_i$.

$$\forall S'^3_i(t), \forall p'^3_i, \max_{p_i^4} \min_{p'^3_i} U^{\mathcal{S}}_{Total}(S_i^4(t), S'^3_i(t)) \tag{18}$$

such that:

$$Q^2.[\sum_{i=1}^{I} U^{*\mathcal{S}}_{(i,t+1)}(S_i^{*4}(t+1), S_i^{*3}(t+1)) + \sum_{i=1}^{I} U^{\mathcal{S}}_{(i,t)}(S_i^4(t), S_i^3(t))]$$
$$\leq Q'^2.[\sum_{i=1}^{I} U^{*\mathcal{S}}_{(i,t+1)}(S_i^{*4}(t+1), S_i^{*3}(t+1)) + \sum_{i=1}^{I} U^{\mathcal{S}}_{(i,t)}(S_i^4(t), S_i'^3(t))]$$

where $Q'^2 = p_i^4.p'^3_i$ and $p'^3_i \geq p_i^3$.

In this Stackelberg security game, the leader players, $\theta^{\mathcal{E}}_{(i,t)}$ and $\theta^{\mathcal{S}}_{(i,t)}$, aim to determine the optimal strategies of follower players, $\theta^{\mathcal{CH}}_{(i,t)}$ and $\theta'^{\mathcal{E}}_{(i,t)}$, defined as $S_i^{*1}(t)$ and $S_i^{*3}(t)$ by solving Eqs. (17) and (18).

Non-cooperative games are considered between the follower and leader players, since the leader players aim to maximize their utility functions, while minimizing the utility functions of their opponents (i.e., malicious $\mathcal{CH}$ and malicious edge server $\mathcal{E}$), and vice-versa. The goal of the leader players, $\theta^{\mathcal{E}}_{(i,t)}$ and $\theta^{\mathcal{S}}_{(i,t)}$, is to predict the future strategies of malicious $\mathcal{CH}$ and malicious edge server $\mathcal{E}$, defined as $S_i^{*1}(t)$ and $S_i^{*3}(t)$. Therefore, as shown in Algorithm 1, the $\mathcal{CH}$ and edge server are categorized as non-trusted devices when their utility functions, $U^{\mathcal{E}}_{Total}(S_i^2(t), S_i^1(t))$ and $U^{\mathcal{S}}_{Total}(S_i^4(t), S_i^3(t))$, reach $\max_{p_i^2} \min_{p'^1_i} U^{\mathcal{E}}_{Total}(S_i^2(t), S'^1_i(t))$ and $\max_{p_i^4} \min_{p'^3_i} U^{\mathcal{S}}_{Total}(S_i^4(t), S'^3_i(t))$, respectively. In this case, the malicious $\mathcal{CH}$ and malicious edge server $\mathcal{E}$ will be deprecated from the network, and only the trusted devices ($\mathcal{CH}$s and edge servers) will afterwards participate in the monitoring and attack detection process.

```
i ← 1;
while $i \leq I$ do
    Leader $\theta^{\mathcal{E}}$ computes $U^{\mathcal{E}}_{(i,t)}(S_i^2(t), S_i^1(t))$;
    Leader $\theta^{\mathcal{S}}$ computes $U^{\mathcal{S}}_{(i,t)}(S_i^4(t), S_i^3(t))$;
    if $U^{\mathcal{E}}_{(i,t)}(S_i^2(t), S_i^1(t)) = U^{\mathcal{E}}_{(i,t)}(S*_i^2(t), S*_i^1(t))$ then
        $\theta^{\mathcal{E}}$ Monitors $U^{\mathcal{CH}}_{(i,t)}(S_i^2(t), S_i^1(t))$
    else
        if $p'^1_i \geq p_i^1$ then
            $\theta^{\mathcal{E}}$ computes $U^{\mathcal{E}}_{Total}(S_i^2(t), S_i^1(t))$
        else
            if $U^{\mathcal{E}}_{Total}(S_i^2(t), S_i^1(t)) = \max_{p_i^2} \min_{p'^1_i} U^{\mathcal{E}}_{Total}(S_i^2(t), S'^1_i(t)))$ then
                $\mathcal{CH}$ device that executes the strategy $S_i^1(t)$ is categorized as a malicious device and will be ejected from the network.
            end
        end
    end
    if $U^{\mathcal{S}}_{(i,t)}(S_i^4(t), S_i^3(t)) = U^{\mathcal{S}}_{(i,t)}(S*_i^4(t), S*_i^3(t))$ then
        $\theta^{\mathcal{S}}$ Monitors $U^{\mathcal{CH}}_{(i,t)}(S_i^4(t), S_i^3(t))$
    else
        if $p'^3_i \geq p_i^3$ then
            $\theta^{\mathcal{S}}$ computes $U^{\mathcal{S}}_{Total}(S_i^4(t), S_i^3(t))$
        else
            if $U^{\mathcal{S}}_{Total}(S_i^4(t), S_i^3(t)) = \max_{p_i^4} \min_{p'^3_i} U^{\mathcal{S}}_{Total}(S_i^4(t), S'^3_i(t)))$ then
                Edge server that executes the strategies $S_i^3(t)$ is categorized as malicious server and will be ejected from the network.
            end
        end
    end
    $i \leftarrow i + 1$
end
```

Algorithm 1: Algorithm for detection of non-trusted devices ($\mathcal{CH}$ and edge server $\mathcal{E}$)

Fig. 3 illustrates the interaction between the hierarchical security framework and the security game to accurately detect malicious IoV devices and malicious edge servers.

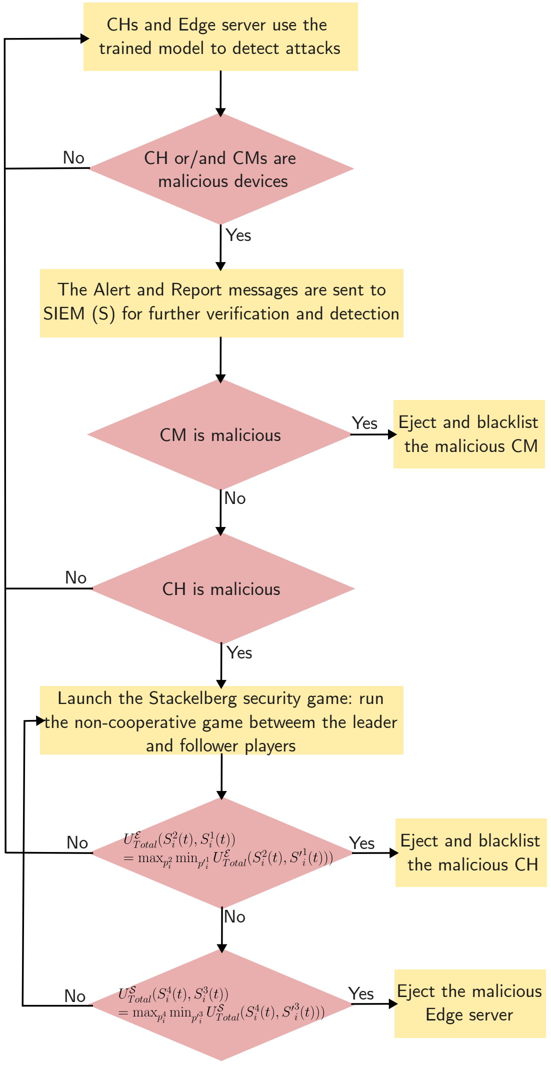

CHs and Edge server use the trained model to detect attacks
No
CH or/and CMs are malicious devices
Yes
The Alert and Report messages are sent to SIEM (S) for further verification and detection
CM is malicious
Yes
Eject and blacklist the malicious CM
No
No
CH is malicious
Yes
Launch the Stackelberg security game: run the non-cooperative game betweem the leader and follower players
No
$U^{\mathcal{E}}_{Total}(S_i^2(t), S_i^1(t)) = \max_{p_i^2} \min_{p'^1_i} U^{\mathcal{E}}_{Total}(S_i^2(t), S'^1_i(t)))$
Yes
Eject and blacklist the malicious CH
No
No
$U^{\mathcal{S}}_{Total}(S_i^4(t), S_i^3(t)) = \max_{p_i^4} \min_{p'^3_i} U^{\mathcal{S}}_{Total}(S_i^4(t), S'^3_i(t)))$
Yes
Eject the malicious Edge server

Fig. 3: Flowchart of the attack detection process

V. Case Study

The 6G-enabled IoV network involves a set of underwater, underground and air sub-networks as described in Fig 1. The IoV network is characterized by the high mobility of IoV devices. Main KPIs of IoV are the latency, the data rate and the connectivity degree [3].

A. Simulation Results

For our experiments, we simulate a set of distributed devices that play the role of IoV nodes (e.g., ground vehicles, drones and underwater vehicles). The total number of distributed devices is equal to 40 nodes. At the beginning of our simulation, we randomly set the values of the main KPIs (e.g., data rate, latency, energy consumption, and connectivity degree).

The FL algorithm trains a feed-forward neural network with 47 inputs and 2 hidden layers, each with 20 and 10 neurons, respectively. The considered loss function is the cross-entropy. The number of iterations for FL training varies from 5 to 50 iterations. The learning rate is equal to 0.01. The batches sizes of $\mathcal{CH}$, edge server $\mathcal{E}$ and the SIEM $\mathcal{S}$ are equal respectively to

5, 30 and 60. We used the network attack data set introduced in [29], which includes nine types of network attacks and one normal traffic behavior. This data set contains 175341 training and 82332 testing records. It corresponds to the real IoT attacks that may be performed against the radio access network (for example, of a 6G network).

Relevant features and data points used for training the model and inferencing are extracted from the network attack data set [29]. Basically, the main features used during the training and attack detection process include the total source bytes, destination bytes, the number of source packets, the number of destination packets, protocol types, the number of packets dropped and forwarded, the number of normal and abnormal records, and the number of unique source/destination IP addresses.

1) Stackelberg security game

As illustrated in Fig. 4, we analyze the security performance of the proposed detection framework with and without activating the Stackelberg security game. Specifically, we compute the robustness metric, which is defined as the attack detection rate minus the false positive rate. The duration of the game varies from 2 seconds to 10 seconds, during which the leader and follower players interact between each other to increase their utility functions and decrease the utility function of their opponents. The leader players aims to determine the optimal strategies for each follower player and execute their specific strategies to increase their utility functions over time. At the end of each game duration, we compute the mean value of the robustness metric by summing up the robustness values of the leader players and dividing it by the total number of leader players. It is apparent from Fig. 4 that by activating the proposed Stackelberg game to monitor the suspected devices and hence detect the non-trusted devices, the mean robustness metric increases, specifically when the game duration reaches 10 seconds. This is in contrast the case where the Stackelberg security game is not activated, which results in a high false positive rate against the legitimate follower players ($\mathcal{CH}$s and $\mathcal{E}$s).

2) Detection Accuracy

We define the detection accuracy as the metric, computed by dividing the number of trusted IoV clusters by the total number of IoV clusters. The trusted IoV clusters are the clusters where $\mathcal{CH}$s have the lowest MDs (as compared to their neighbors $\mathcal{CM}$s) and satisfy the network requirement KPIs. The number of iterations varies from 5 to 50 iterations as illustrated in Fig. 5. At each iteration, the FL algorithm and Stackelberg game are executed at the IoV and edger levels, and cover the clustering, training and detection process as explained in Section IV.

Fig. 5 shows that the detection accuracy increases with respect to the number of iterations during the training. Indeed, the FL based cooperative detection reduces the false positives and false negatives. Meanwhile, the Stackelberg security game detects accurately the suspected $\mathcal{CH}$ or edge server that exhibits malicious behaviors.

3) Detection and false positive rates

As compared to current centralized attack detection frameworks for 6G such as [16], [17], the proposed detection system shows a high detection rate and low false positive rate when analyzing attack detection rates. This is especially true when the amount of malicious traffic reaches 45% (of all data traffic), as shown in Figs. 6(a) and 6(b). The proposed solution achieves a high level of security by monitoring the suspected behaviors of $\mathcal{CH}$s and edge servers using the Stackelberg game. Equilibrium states are reached where the follower $\mathcal{CH}$ launches an attack, and the leader edge detects this attack; or when the follower edge server runs an attack and the leader SIEM detects this malicious behavior. In addition, our cooperative attack detection is improved by the inputs of the different devices at different hierarchical levels.

4) Computation overheads

As shown in Fig. 6(c), we vary the amount of malicious traffic from 10% to 45% and analyze the computation overhead generated by our proposed solution and the centralized detection frameworks from the state-of-the-art [16], [17]. As illustrated in Fig. 6(c), the centralized detection framework requires a high computation overhead to achieve a high accuracy detection, specifically when the amount of malicious traffic is close to 45%. This is mainly due to huge amount of data collected at the centralized node (e.g., SIEM) to carry out the training and attack detection process. However, in our proposed solution, a low computation overhead is required to

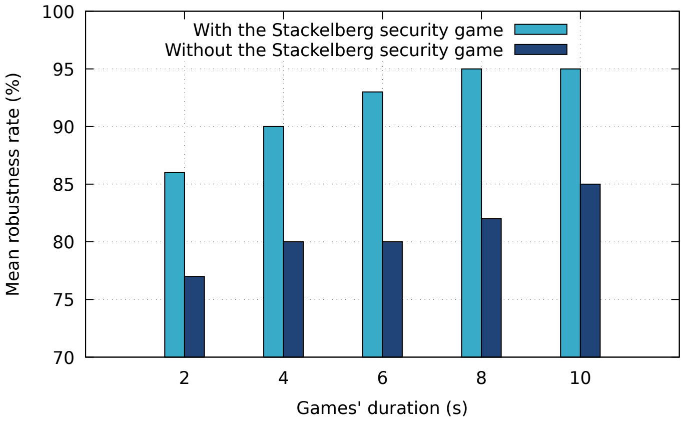


Fig. 4: Robustness of the proposed framework

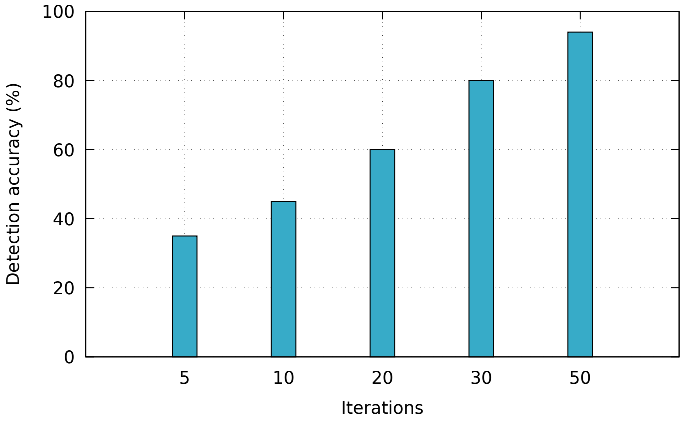
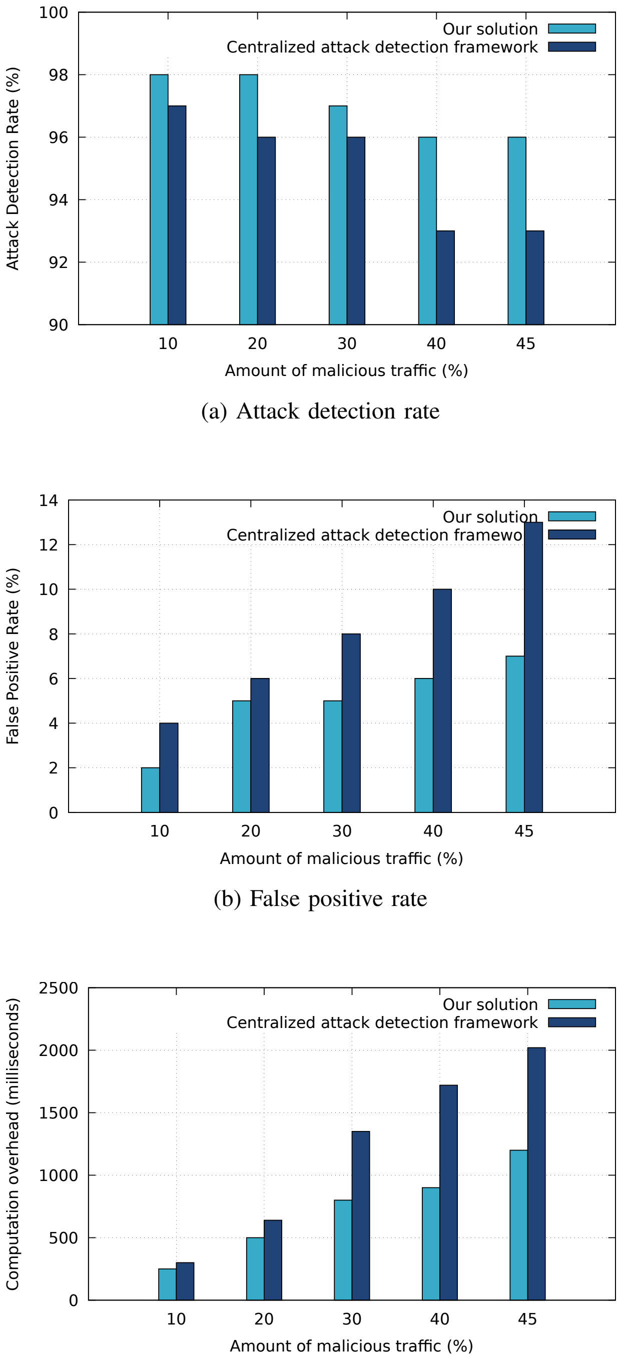


Fig. 5: Detection accuracy of the proposed framework



(a) Attack detection rate

(b) False positive rate

(c) Computation overhead

Fig. 6: Hierarchical vs Centralized detection framework

prevent the occurrence of internal and external adversaries. Indeed, to accelerate the attack detection process at the IoV and edge levels, two detection layers (at IoV devices, edge devices and SIEM) are combined together.

VI. Conclusion

Provisioning applications and services for the emerging 6G network has drawn much attention from IT and telecommunications operators. However, ensuring efficient security mechanisms in the context of 6G architectures remains largely unexplored. In this research work, we have proposed a new collaborative cyber security framework, based on a multi-level FL algorithm and Stackelberg security games, to secure 6G-enabled IoV networks from attacks that target the main KPIs of 6G architectures. Specifically, we have proposed a concrete construction of a hierarchical attack detection framework that leverages the processing capabilities of IoV nodes, edge servers and SIEM. The proposed attack detection framework presents a crucial step in the research and development of cyber security and AI systems in emerging 6G-enabled IoV networks, and is expected to benefit the academic and industrial communities.

Our future work will focus on evaluating alternative ML models and integrating the proposed solution into dynamic environments to demonstrate the versatility of the proposed methodology in ever-evolving settings. In addition, we will explore the use of fully homomorphic encryption as discussed by [30]–[33] to make the intrusion detection more privacy-preserving. However, using homomorphic encryption will require the adaptation of the used models and may result in a loss of accuracy, and we will carefully evaluate this trade-off in our future research.

Acknowledgment

This work is an extended and enhanced version of the conference paper that has been presented at IEEE ICC 2022 in Seoul [34].

References

[1] W. Liu, S. Zhang, and N. Ansari, “Joint laser charging and dbs placement for drone-assisted edge computing,” *IEEE Transactions on Vehicular Technology*, vol. 71, no. 1, pp. 780–789, 2021.

[2] W. Li, Z. Su, R. Li, K. Zhang, and Y. Wang, “Blockchain-based data security for artificial intelligence applications in 6G networks,” *IEEE Network*, vol. 34, no. 6, pp. 31–37, 2020.

[3] G. Gui, M. Liu, F. Tang, N. Kato, and F. Adachi, “6G: Opening new horizons for integration of comfort, security, and intelligence,” *IEEE Wireless Communications*, vol. 27, no. 5, pp. 126–132, 2020.

[4] W. Liu, S. Zhang, and N. Ansari, “Joint laser charging and dbs placement for drone-assisted edge computing,” *IEEE Transactions on Vehicular Technology*, vol. 71, no. 1, pp. 780–789, 2022.

[5] Y. Hui, N. Cheng, Z. Su, Y. Huang, P. Zhao, T. H. Luan, and C. Li, “Secure and personalized edge computing services in 6g heterogeneous vehicular networks,” *IEEE Internet of Things Journal*, vol. 9, no. 8, pp. 5920 – 5931, 2021.

[6] M. Giordani and M. Zorzi, “Satellite communication at millimeter waves: A key enabler of the 6G era,” in *IEEE International Conference on Computing, Networking and Communications (ICNC)*, Big Island, HI, USA, 2020, pp. 383–388.

[7] B. Mao, Y. Kawamoto, and N. Kato, “AI-based joint optimization of qos and security for 6G energy harvesting internet of things,” *IEEE Internet of Things Journal*, vol. 7, no. 8, pp. 7032–7042, 2020.

[8] K. B. Letaief, W. Chen, Y. Shi, J. Zhang, and Y.-J. A. Zhang, “The roadmap to 6G: AI empowered wireless networks,” *IEEE communications magazine*, vol. 57, no. 8, pp. 84–90, 2019.

[9] M. Wang, T. Zhu, T. Zhang, J. Zhang, S. Yu, and W. Zhou, “Security and privacy in 6G networks: New areas and new challenges,” *Digital Communications and Networks*, vol. 6, no. 3, pp. 281–291, 2020.

[10] P. Porambage, G. Gür, D. P. Moya Osorio, M. Livanage, and M. Ylianttila, “6G security challenges and potential solutions,” in *Joint European Conference on Networks and Communications 6G Summit (EuCNC/6G Summit)*, 2021, pp. 622–627.

[11] C. Benzaid and T. Taleb, “Zsm security: Threat surface and best practices,” *IEEE Network*, vol. 34, no. 3, pp. 124–133, 2020.

[12] P. Porambage, G. Gür, D. P. M. Osorio, M. Liyanage, A. Gurtov, and M. Ylianttila, “The roadmap to 6G security and privacy,” *IEEE Open Journal of the Communications Society*, vol. 2, pp. 1094 – 1122, 2021.

[13] S. Kaur and M. Singh, “Hybrid intrusion detection and signature generation using deep recurrent neural networks,” *Neural Computing and Applications*, vol. 32, no. 12, pp. 7859–7877, 2020.

[14] N. Kaaniche, A. Boudguiga, and G. Gonzalez-Granadillo, “Efficient hybrid model for intrusion detection systems,” Lisbon, Portugal, 2022, pp. 694–700.

[15] A. Al Mousa, M. Al Qomri, S. Al Hajri, and R. Zagrouba, “Utilizing the esim for public key cryptography: a network security solution for 6G,” in *2nd IEEE International Conference on Computer and Information Sciences (ICCIS)*, Sakaka, Saudi Arabia, 2020, pp. 1–6.

[16] S. Li, W. Sun, H. Zhang, and Y. Zhang, “Physical layer security for edge caching in 6G networks,” in *IEEE Global Communications Conference*, Taipei, Taiwan, 2020, pp. 1–6.

[17] C. L. Stergiou, K. E. Psannis, and B. B. Gupta, “Iot-based big data secure management in the fog over a 6G wireless network,” *IEEE Internet of Things Journal*, vol. 8, no. 7, pp. 5164–5171, 2020.

[18] Y. Liu, X. Yuan, Z. Xiong, J. Kang, X. Wang, and D. Niyato, “Federated learning for 6G communications: Challenges, methods, and future directions,” *China Communications*, vol. 17, no. 9, pp. 105–118, 2020.

[19] H. Zhou, H.-F. Li, and R. Yuan, “Task offloading strategy of 6G heterogeneous edge-cloud computing model considering mass customization mode collaborative manufacturing environment,” *Mathematical Problems in Engineering Journal*, pp. 1–8, 2020.

[20] H. Wen, P. Y.-R. Huang, J. Dyer, A. Archinal, and J. Fagan, “Countermeasures for GPS signal spoofing,” in *Proceedings of the 18th international technical meeting of the satellite division of the institute of navigation (ION GNSS 2005)*, 2005, pp. 1285–1290.

[21] D. Dolev and A. Yao, “On the security of public key protocols,” *IEEE Transactions on Information Theory*, vol. 29, no. 2, pp. 198–208, 1983.

[22] H. Sedjelmaci and S. M. Senouci, “An accurate and efficient collaborative intrusion detection framework to secure vehicular networks,” *Computers Electrical Engineering*, vol. 43, pp. 33–47, 2015.

[23] P. Kairouz, H. B. McMahan, B. Avent, A. Bellet, M. Bennis, A. N. Bhagoji, K. Bonawitz, Z. Charles, G. Cormode, R. Cummings *et al.*, “Advances and open problems in federated learning,” *Foundations and Trends® in Machine Learning*, vol. 14, no. 1–2, pp. 1–210, 2021.

[24] A. Abidi, B. Bouallegue, and F. Kahri, “Implementation of elliptic curve digital signature algorithm (ecdsa),” in *Global Summit on Computer Information Technology (GSCIT)*, Sousse, Tunisia, 2014.

[25] M. Nasr, R. Shokri, and A. Houmansadr, “Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning,” in *IEEE symposium on security and privacy (SP)*, vol. 1, 2019, pp. 739–753.

[26] R. Shokri, “Auditing data privacy for machine learning,” *USENIX Association*, 2022.

[27] L. An, A. Chakrabortty, and A. Duel-Hallen, “A stackelberg security investment game for voltage stability of power systems,” in *59th IEEE Conference on Decision and Control (CDC)*, 2020, pp. 3359–3364.

[28] N. Abuzainab and W. Saad, “Dynamic connectivity game for adversarial internet of battlefield things systems,” *IEEE Internet of Things Journal*, vol. 5, no. 1, pp. 378–390, 2017.

[29] N. Moustafa and J. Slay, “Unsw-nb15: a comprehensive data set for network intrusion detection systems (unsw-nb15 network data set),” in *IEEE military communications and information systems conference (MilCIS)*, Canberra, ACT, Australia, 2015, pp. 1–6.

[30] L. Sgaglione, L. Coppolino, S. D’Antonio, G. Mazzeo, L. Romano, D. Cotroneo, and A. Scognamiglio, “Privacy preserving intrusion detection via homomorphic encryption,” in *2019 IEEE 28th International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE)*, Napoli, Italy, 2019, pp. 321–326.

[31] L. Coppolino, S. D’Antonio, V. Formicola, G. Mazzeo, and L. Romano, “Vise: Combining intel sgx and homomorphic encryption for cloud industrial control systems,” *IEEE Transactions on Computers*, vol. 70, no. 5, pp. 711–724, 2021.

[32] L. Sgaglione and G. Mazzeo, “A gdpr-compliant approach to real-time processing of sensitive data,” in *Intelligent Interactive Multimedia Systems and Services*, G. De Pietro, L. Gallo, R. J. Howlett, L. C. Jain, and L. Vlacic, Eds. Cham: Springer International Publishing, 2019, pp. 43–52.

[33] A. Boudguiga, O. Stan, H. Sedjelmaci, and S. Carpov, “Homomorphic encryption at work for private analysis of security logs,” in *Proceedings of the 6th International Conference on Information Systems Security and Privacy, ICISSP*, Valletta, Malta, 2020, pp. 515–523.

[34] H. Sedjelmaci, N. Kheir, A. Boudguiga, and N. Kaaniche, “Cooperative and smart attacks detection systems in 6G-enabled internet of things,” in *IEEE International Conference on Communications*, Seoul, South Korea, 2022.

Hichem Sedejlmaci, PhD Joined Ericsson in 2021. He leads the technical R and D activities dealing the interactions between AI and Security and drives standardization studies. Before to join Ericsson, he was a Senior Research Engineer in Cyber Security and AI, and Projects Manager at Orange Labs. He served as a Guest Editor of premium journals, such ADHOC, IEEE Network and IEEE JSAC Journals. He holds more than 20 international patents on the topics of cyber security and AI.

Nesrine Kaaniche, PhD is an Associate Professor in Cybersecurity at Télécom SudParis, Institut Polytechnique de Paris. Her major research interests include privacy enhancing technologies and applied cryptography for distributed systems and decentralised architectures.

Aymen Boudguiga, PhD is a cybersecurity researcher working at CEA, France. His current research concerns the application of homomorphic encryption to machine learning algorithms.

Nirwan Ansari (S’78–M’83–SM’94–F’09), Distinguished Professor of Electrical and Computer Engineering at the New Jersey Institute of Technology (NJIT), holds a Ph.D. from Purdue University, an MSEE from the University of Michigan, and a BSEE (summa cum laude with a perfect GPA) from NJIT. He is also a Fellow of National Academy of Inventors. He has published three books and (co-)authored over 700 technical publications, with more than half of them published in widely cited journals/magazines.