SP-CIDS: Secure and Private Collaborative IDS for VANETs

Gunasekaran Raja, *Senior Member, IEEE*, Sudha Anbalagan, Geetha Vijayaraghavan, Sudhakar Theerthagiri, Saran Vaitangarukav Suryanarayan, *Member, IEEE*, and Xin-Wen Wu, *Senior Member, IEEE*

***Abstract*— Vehicular Ad hoc NETworks (VANETs) serve as the backbone of Intelligent Transportation Systems (ITS), providing passengers with safety and comfort. However, VANETs are vulnerable to major threats that affect data privacy and network services either from an individual or distributed attacker. In this paper, a Secure and Private-Collaborative Intrusion Detection System (SP-CIDS) is proposed to detect network attacks and to mitigate security concerns. In SP-CIDS, a Distributed Machine Learning (DML) model based on the Alternating Direction Method of Multipliers (ADMM) is used, which leverages the potential of vehicle-to-vehicle collaboration in the learning process to improve the storage efficiency, accuracy, and scalability of the IDS. However, there are significant data privacy concerns possible in such collaboration, where a CIDS can act as a malicious system that has access to the intermediate stages of the learning process. Additionally, the SP-CIDS system uses Differential Privacy (DP) technique to address the aforementioned data privacy risk associated with the DML-based CIDS. The SP-CIDS system is evaluated with logistic regression, naïve bayes, and ensemble classifiers. Simulation results substantiate that a private ensemble classifier secures the training data with DP and also achieves 96.94% accuracy.**

***Index Terms*— ADMM, CIDS, differential privacy, distributed machine learning, ITS, privacy-preserving.**

I. Introduction

VEHICULAR Ad hoc NETworks (VANETs) are dynamic networks formed by the moving vehicles to exchange information among them in a self-organized manner. VANETs serves as a reliable platform for urban mobile networks and improves services such as road-safety assistance, traffic control, and infotainment capabilities for Intelligent Transport Systems (ITS). However, the network faces several challenges such as Quality of Service (QoS) provisioning, security, and privacy [1], [2]. In particular, privacy is a key security concern that needs to be addressed in all current and future interconnected networks [3]. As per the Worldwide Infrastructure Security Report (2019), Distributed Denial of Services (DDoS) attacks are the primary threat faced by enterprises and data centers around the world [4]. Growing digitalization and increasing data breaches drive the market-demand for Intrusion Detection System/Intrusion Prevention System (IDS/IPS) which is predicted to exceed a market value of 8 million USD, by the year 2025 [5].

These IDS/IPS systems find major use cases in VANET systems. The vehicles in VANET consists of three major components: On-Board Unit (OBU), Application Unit (AU), and communication module. The communication in VANET is facilitated in three primary modes namely Vehicle-to-Vehicle (V2V), Vehicle-to-Infrastructure (V2I) and Infrastructure-to-Infrastructure (I2I). VANET also consists of Road Side Units (RSUs), which are placed along the roadways, near traffic signals, or toll areas. OBUs and RSUs facilitate communication in VANET through Dedicated Short-Range Communication (DSRC) to perform functions such as data redistribution, congestion control, and messaging services [6]. It also enables connectivity with other infrastructures such as cloud, fog, Traffic Management Center (TMC), etc.

Like any other wireless network, VANETs are also susceptible to numerous threats such as spoofing, DoS, message delay attack, and so on [2], [7]. To actively counter these threats, network operators use firewalls and antivirus systems as their primary defense mechanism. Nonetheless, rogue users often slip through these systems and access the network, which can be identified using reactive technologies like malware analysis, IDS, etc [8], [9].

An IDS system monitors a host or a network and analyses the audit logs to look for malicious behavior or security policy violations [10]. Based on the objective of its protection, IDS is further classified into Host-based IDS (HIDS) or Network-based IDS (NIDS) [11]. The conventional centralized NIDS architecture has many design shortcomings, including Single Point of Failure (SPoF), and low scalability. For a distributed environment such as VANET, Collaborative IDS (CIDS) mitigates these drawbacks, by empowering vehicles to share their knowledge and computational resources.

Manuscript received July 7, 2020; revised October 10, 2020; accepted October 30, 2020. Date of publication November 24, 2020; date of current version July 12, 2021. This work was supported by the Ministry of Electronics and Information Technology (MeitY), Government of India, under Grant YFRF-DIC/MUM/GA/10(37)D on 24. 01. 2019. The Associate Editor for this article was A. Jolfaei. *(Corresponding author: Gunasekaran Raja.)*

Gunasekaran Raja, Geetha Vijayaraghavan, and Saran Vaitangarukav Suryanarayan are with the NGNLab, Department of Computer Technology, Anna University, Chennai 600025, India (e-mail: dr.r.gunasekaran@ieee.org; geethu15@gmail.com; vssaran1998@gmail.com).

Sudha Anbalagan is with the School of Computing, College of Engineering and Technology, SRM Institute of Science and Technology, Chennai 603203, India (e-mail: sudhaa@srmist.edu.in).

Sudhakar Theerthagiri is with the Department of Computer Science and Engineering, Amrita School of Engineering, Amrita Vishwa Vidhyapeetham, Chennai 601103, India (e-mail: tsudhakar105@gmail.com).

Xin-Wen Wu is with the Department of Mathematical and Computer Sciences, Indiana University of Pennsylvania, Indiana, PA 15705 USA (e-mail: xwu@iup.edu).

Digital Object Identifier 10.1109/TITS.2020.3036071

A CIDS is organized as a collection of HIDS or NIDS permitting information sharing and detecting collaborative or distributed attacks within VANET. A typical CIDS unit consists of local monitoring, global monitoring, correlation and aggregation, and data dissemination components [12]. The CIDS systems are often based on Distributed Machine Learning (DML), which is used to detect both known and unknown attacks [13]. DML algorithms are a class of multi-node ML algorithms designed to improve computational efficiency, accuracy, and scalability for large datasets. A critical security requirement for a DML-based CIDS system is to facilitate the sharing of the ML model without compromising data privacy. In particular, DML is vulnerable to model inversion attacks, in which adversaries can infer sensitive training data from learning outcomes [14].

To overcome the above mentioned challenges, Secure and Private CIDS (SP-CIDS) is proposed using DML based on Alternating Direction Method of Multipliers (ADMM), where all the vehicles collaborate to create a global classifier. In SP-CIDS, the data privacy risk associated with the DML is alleviated using Differential Privacy (DP) paradigm.

Following are the contributions of this article:

- An efficient SP-CIDS using ADMM based DML is proposed which serves to enhance the storage and computation efficiency of the IDS.
- The privacy of the training data used in DML is preserved through the DP paradigm addressing the data privacy concerns.
- The accuracy of the CIDS is enhanced by creating a global model through the collaborating neighboring vehicles by using a private ensemble-based classifier in SP-CIDS.

The paper is organized as follows: Section II provides general background techniques used in CIDS and related works. In section III, the architecture of the SP-CIDS system is explained in detail. Section IV provides an overview of distributed learning and its privacy risk. Section V discusses on how DP assists in private DML. Section VI presents the implementation and evaluation results of the SP-CIDS system. Finally, in section VII, concluding remarks of the work is provided.

II. Related Work

This section provides a literature study on various CIDS architectures, ML, and data-mining techniques that are used in the security of cyber-physical systems. The section also discusses some of the privacy preservation techniques from various literature.

DDoS attacks are a major challenge for many businesses and network security teams [4]. However, the traditional stand-alone IDS is insufficient to prevent such attacks. Motivated by the need to prevent or detect DDoS attacks, CIDS has emerged as an effective network security method. A centralized CIDS includes multiple monitoring devices that track the behavior of the host or network it controls, and share the collected data with a central analytical unit. An example of centralized CIDS is the co-operative intrusion detection framework, which uses a centralized co-operative module to receive alert data from their constituent monitors [15]. The IDS framework evaluates the received alerts and speculates the adversary's next possible step. However, this system is susceptible to SPoF and performance bottlenecks. The challenges faced by such a centralized design are mitigated in Hierarchical Intrusion Detection (HIDE) system [16]. HIDE consists of intrusion detection agents ordered hierarchically. It performs anomaly detection using statistical pre-processing and classification by neural networks. Nevertheless, as the IDS alerts are aggregated and refined at each level, information often gets lost, reducing the system's accuracy. Such information loss is reduced in Large Scale Intrusion Detection (LarSID) [17]. LarSID is a Peer to Peer (P2P) based CIDS system, designed as a publish/subscribe system. In this system, every peer has a monitoring unit and analysis unit that detects suspicious IP addresses and publishes it across the network.

Various literature studies indicate that data mining and ML techniques are used for IDS, which utilizes a trained model to classify suspicious activities as attacks or non-attacks. In addition, techniques such as dimensionality reduction in the pre-processing stage reduce the dimension of the dataset without sacrificing the information contained by it. In [18], a combination of HIDS and NIDS using deep learning techniques is proposed for Advanced Metering Infrastructure (AMI). IDS also uses a Convolution Neural Network (CNN), a subset of deep neural networks commonly used in image processing. The IDS proposed in [19] uses CNN to detect DoS attacks.

In [20], the authors proposed an IDS based on a clustering algorithm to cluster information packets into normal and anomalous. Besides ML, other techniques like Ant Colony Optimization (ACO) and game-theoretic approach are also used in IDS. Although ACO achieves a less false alarm rate, it is inefficient in handling multi-class anomalies in the network [21]. The game-theoretic approach is used in CIDS to maintain the trust among peers and detects the adversaries like a dishonest insider, DoS attacker, and free-riders [22].

Further, the challenge of the trade-off between the ML model's privacy and utility is addressed using private methods of learning, as in [23] and [24]. It is therefore important to mitigate challenges such as ensuring the scalability, performance, and storage efficiency of the CIDS system. The proposed SP-CIDS system protects the training data as well as improves the accuracy, scalability, and storage efficiency of the IDS system, by addressing all the aforementioned challenges. The DML of SP-CIDS uses ADMM for solving distributed convex optimization, which has a better convergence rate of O(1/t). ADMM [25] solves an optimization problem in the form of decomposition-coordination procedure. In ADMM, the solutions to the local subproblems are reconciled to solve a global problem. Although ADMM possesses several benefits like scalability due to parallelized data processing, absence of gradient fading [25], [26], it has some privacy issues. This is because any access to the final distributed learning model and the classifier function in each iteration could lead the adversary to make statistical inferences of training data [14]. In such situations, privacy protection methods like pure anonymization

TABLE I
LIST OF NOTATIONS USED

Symbol	Description
$\mathcal{N}$	Set of vehicles in VANET
N	Number of vehicles in set $\mathcal{N}$
l	A vehicle belongs to $\mathcal{N}$
$\hat{D}$	Total training dataset
D_l	Training dataset of vehicle l
n_l	Size of the dataset of vehicle l
f_l	Classifier of vehicle l
P_l	Set of neighboring vehicles of vehicle l
$\mathscr{L}$	Loss function of a classifier
λ_{lw}	Dual variable
ε	Privacy parameter

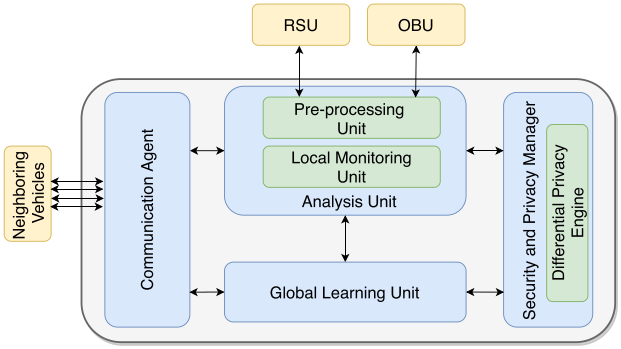


Fig. 1. Architecture of SP-CIDS.

or data sanitization are not adequate. Therefore, the DP paradigm is incorporated in SP-CIDS, which adds a systematic noise to the shared parameters to preserve the privacy of the training data.

III. SP-CIDS System Model

The SP-CIDS system in each vehicle of VANET creates an ML model to detect the attacks in the network. The DML-based CIDS improves the storage efficiency and accuracy of IDS by creating a global model through the collaborating neighboring vehicles. Table I displays the list of notations used in Section III and IV.

The architecture of SP-CIDS is composed of four main components namely Analysis Unit (AU), Communication Agent, Global Learning Unit (GLU) and Security and Privacy Manager as shown in Fig. 1. The role of each of these components is discussed as follows:

A. Analysis Unit

The AU collects network audit data from OBU, neighboring vehicles, and RSUs, which are correlated and aggregated to detect malicious events such as drop and duplicate of packets in the network. It consists of the pre-processing unit and the Local Monitoring Unit (LMU).

1) Pre-Processing Unit: The pre-processing module receives real-time VANET data, which describes the activities of the network. This module uses filtering techniques to filter noise, undesirable and incomplete data. The training dataset often has missing values due to OBU failure or unstable connectivity. To replace such missing values, some of the methods like substituting mean value method, value ignoring method, or interpolation method is used. The data is then sampled using several sampling techniques, and the final data is further normalized for efficient processing by the ML algorithm.

2) Local Monitoring Unit: The LMU receives input from the pre-processing engine and performs analysis using the global model created by the GLU. The typical operation of an LMU can be illustrated as follows. Let us consider a vehicle l in a VANET of $\mathcal{N}$ vehicles. The classifier of the vehicle l, $f_l : X \rightarrow Y$, uses the training dataset D_l, to classify any input point $x_1 \in X$ to an output $y_1 \in \{-1, 1\}$, where -1 represents normal network behaviour and 1 indicates an intrusion by malicious node. If the activity is classified as an intrusion, then the LMU triggers an alarm. If the user updates their classifier model, then re-training is performed by initializing the GLU.

B. Global Learning Unit

Upon initialization of GLU, a secure and private DML algorithm is executed to create a global classifier that is collaboratively trained over the vehicles in the VANET. For efficient decentralization of ML algorithms, GLU uses the ADMM algorithm, as the convergence is better compared to other decentralization algorithms [25]. ADMM solves the convex optimization problem in ML by breaking it into smaller pieces, each of which is easier for each vehicle to handle independently. Thus, each vehicle uses its own labeled training data to create a local classifier, that is synchronized iteratively to achieve a global classifier, which is equivalent to the classifier trained on the total dataset of all the vehicles. This training data may consist of historical intrusions, which were detected by vehicles or data created by trusted parties such as the department of transportation. The participation of RSU enables the vehicles to connect remote vehicles from a different RSU or cloud infrastructure.

C. Communication Agent

The communication agent in the vehicle acts as a communication hub between the local IDS system and other vehicles or RSUs in the VANET. All the data sharing and alert dissemination happen via this agent. It connects other vehicles in each ADMM iteration to share the immediately modified parameters using the DSRC protocol.

D. Security and Privacy Manager

The security and privacy manager component in the vehicle uses the DP paradigm, which systematically adds noise to the classifier to preserve the privacy of the training data. Therefore, this module also controls the level of noise added to the DML algorithm's collaborative parameters.

Thus, the operations performed in the various modules of the SP-CIDS, and their collaborations are illustrated in Algorithm 1.

Algorithm 1 SP-CIDS Algorithm

Input: Network audit data
Output: Intrusion alerts to all the vehicles

1: Pre-process (audit data) with respect to VANET system
2: **if** LMU needs update **then**
3: Send Vehicles' ($Preprocessed\_data$) into GLU
4: Update classifier in GLU
5: Send updated classifier to LMU
6: **end if**
7: **if** LMU detects intrusions **then**
8: trigger and send alert message to all vehicles
9: **end if**

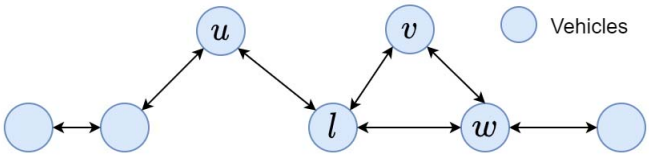


Fig. 2. Communication graph of VANET.

IV. Distributed Learning for SP-CIDS

Let us consider the network topology of VANET as a graph G ($\mathcal{N}$, $\mathbb{E}$), where $\mathcal{N}$ represents the set of vehicles in the VANET: $\mathcal{N}$ = {1,2,3,...N} and $\mathbb{E}$ is the set of edges connecting the vehicles. Vehicles in the VANET environment are dynamic in nature. A vehicle can communicate only to its one-hop neighboring vehicles within its transmission range. For instance, a vehicle l in the VANET ($l \in \mathcal{N}$) can only communicate with its neighboring vehicles $u, v, w \in P_l$, where $P_l \subset \mathcal{N}$, as illustrated in Fig. 2.

The vehicle l in the communication graph consists of labeled training dataset represented as $D_l = \{(x_{il}, y_{il}) \subset X \times Y \; i = 0, 1, \ldots, n_l\}$, where n_l is the dataset size of the vehicle l, and the data instances $x_{il} \in X \subseteq \mathbb{R}^d$ and $y_{il} \in Y = \{-1, 1\}$ is the input instance and its class label, respectively. The entire network has the total training dataset $\hat{D} = \bigcup_{l \in \mathcal{N}} D_l$. The training dataset D_l of the vehicle l is a set of data points that describes the activities happening in applications and the communication between OBUs. The output vector in D_l is -1 if it is a normal activity and 1 if it is an intrusion.

The objective of the GLU is to obtain an efficient classifier based on the combined training dataset of the real-time VANET environment. The Centralized Empirical Risk Minimization (C-ERM) is decentralized using the ADMM mechanism to suit the distributed nature of VANET. The following subsection illustrates the centralized optimization of the objective function in ML and further extends this notion to decentralized optimization in DML using ADMM.

A. Centralized-ERM Optimization

In C-ERM based ML optimization, the problem is to find a classifier, $f \; X \rightarrow Y$, using the training dataset $\hat{D}$, to classify any input point x_i, to an output $y_i \in \{-1, 1\}$.

Let us consider the objective function of centralized ML algorithm be $V_1(f|\hat{D})$. Formally, it is represented as follows:

$$\min_f V_1\left(f|\hat{D}\right) := \frac{K_1}{n_l}\sum_{l=1}^{N}\sum_{i=1}^{n_l}\mathscr{L}\left(y_{il}, f^T x_{il}\right) + kR(f) \tag{1}$$

A loss function is a mathematical representation of the classification quality and is denoted as $\mathscr{L}\left(y_{il}, f^T x_{il}\right)$. In Eq. (1), $R(f)$ is added to prevent over-fitting problems during the training phase of the learning, and this process is known as regularization. $K_1 \leqslant n_l$ is a parameter for the regularization and k controls the effect of the regularizer function. Thus, $\hat{D}$ is provided as input to the classifier, and by optimizing the C-ERM equation, a global classifier $f : X \rightarrow Y$ is obtained.

B. Distributed-ERM Optimization

The optimization problem in Eq. (1) is decentralized using ADMM. Each vehicle l models its own classifier f_l, such that it also satisfies the global consistency constraint $f_1 = f_2 = \ldots = f_N$. Then the C-ERM problem in Eq. (1) is reformulated as follows:

$$\min_{\{f_l\}_{l=1}^N} V_2 = \frac{K_1}{n_l}\sum_{l=1}^{N}\sum_{i=1}^{n_l}\mathscr{L}\left(y_{il}, f^T x_{il}\right) + N\sum_{l=1}^{N} kR(f) \tag{2}$$

According to Lemma 1 in [27], if there exists a feasible solution for Eq. (2) with the network being connected, then the equations Eq. (1) and (2) are equivalent. Eq. (2) is solved using ADMM, if each vehicle $l \in \mathcal{N}$ optimizes the D-ERM equation as follows:

$$V_l(f_l|D_l) := \frac{K_1}{n_l}\sum_{i=1}^{n_l}\mathscr{L}\left(y_{il}, f^T x_{il}\right) + rR(f_l) \tag{3}$$

Here $r = Nk$, for an individual vehicle. ADMM based DML achieves a satisfactory speed of convergence even in unconditioned objective functions. Also during implementation, ADMM is easy to parallelize and resilient to noise as well as computational errors [24]. ADMM is used to solve regularized problems, in which the function optimization and regularization happens locally and further co-ordinated globally. The augmented lagrange function of ADMM helps in solving a constrained optimization problem, which is formulated as follows for solving the D-ERM problem [27]:

$$\begin{aligned} L_l^D\left(f_l, a_{lw}, \lambda_{lw}^k\right) &:= V_l + \sum_{i \in P_l}\left(\lambda_{li}^a\right)^T (f_l - a_{li}) + \sum_{i \in P_l}\left(\lambda_{li}^b\right)^T (a_{li} - f_l) \\ &\quad + \frac{\eta}{2}\sum_{i \in P_l}\left(\| f_l - a_{li} \|^2 + \| a_{li} - f_l \|^2\right) \end{aligned} \tag{4}$$

Auxiliary variables $\{a_{lw}\}$ is introduced to separate the classifier f_l of the vehicle from its neighboring vehicles $w \in P_l$. To solve Eq. (4), a distributive ADMM procedure is used as

per Eq. (5), (6), (7) and (8):

$$f_l(t+1) := arg \min_{f_l} L_v^D \left(f_l, a_{lw}(t), \lambda_{lw}^k(t)\right) \quad (5)$$

$$a_{lw}(t+1) := arg \min_{f_l} L_v^D \left(f_l(t+1), a_{lw}, \lambda_{lw}^k(t)\right) \quad (6)$$

$$\lambda_{lw}^a(t+1) := \lambda_{lw}^a(t) + \eta(f_l(t+1) - a_{lw}(t+1)) \quad (7)$$

$$\lambda_{lw}^b(t+1) := \lambda_{lw}^b(t) + \eta(a_{lw}(t+1) - f_l(t+1)) \quad (8)$$

In the above Eq. (5), (6), (7) and (8), l denotes the vehicle of interest, $l \in \mathcal{N}$ and w is a neighboring vehicle of l and $w \in P_l$.

Eq. (6) has closed form solution, as the cost is linear-quadratic of $a_{lw}(t+1)$ [28]. Thus, the calculation of $a_{lw}(t+1)$ is eliminated and the procedure is simplified using Lemma 3 in [28]. This outcome can be achieved by initializing the dual variables $\lambda_{lw}^k = 0$ and then combining the variables λ_{lw}^a,λ_{lw}^b into a single variable designated as $\lambda_l(t) = \sum_{w \in P_l} \lambda_{lw}^k$, where $l \in N$, $w \in P_l$ and $k = a, b$.

The simplified Eq. (4) is given in Eq. (9):

$$L_l^D(t) := \frac{K_1}{n_l} \sum_{i=1}^{n_l} \mathscr{L} \left(y_{il} f^T x_{il}\right) + rR(f_l) + 2\lambda_l(t)^T(f_l) + \eta \sum_{i \in P_l} \| f_l - \frac{1}{2}(f_l(t) + f_i(t)) \|^2 \quad (9)$$

The iterative equations are reformulated as:

$$f_l(t+1) := arg \min_{f_l} L_v^D (f_l, f_l(t), \lambda_l(t)) \quad (10)$$

$$\lambda_l(t+1) := \lambda_l(t) + \frac{\eta}{2}(f_l(t+1) - f_w(t+1)) \quad (11)$$

The distributed learning algorithm for SP-CIDS as stated in Algorithm 2, is obtained by combining the iterative procedure of ADMM into a centralized ML algorithm. For any iteration $t+1$, each vehicle updates its classifier $f_l(t)$ using Eq. (10). Next, the vehicle l broadcasts its updated classifier $f_l(t+1)$ to all of its neighbours $w \in P_l$ using the communication agent. After each vehicle updates its dual variable $\lambda_l(t+1)$ using Eq. (11), the ADMM iteration $t+1$ is considered as complete. In each iteration, the vehicle $l \in \mathcal{N}$ updates its own classifier $f_l(t)$ and dual variable $\lambda_l(t)$. The only information exchanged between the vehicles is $f_l(t)$, which is safe because direct sharing of training data is avoided. The algorithm runs for Th number of iterations, which is considered as 50 for the implementation.

Algorithm 2 Distributed Learning Over VANET

Input: Network audit data from all the vehicles ($\hat{D}$)
Output: Updated classifier (f_l)

1: For each vehicle $l \in \mathcal{N}$, the classifier $\{f_l\}_{l=1}^N$ is randomly initialized.
2: The dual variable, λ_l is initialized to 0 for all the vehicles in the VANET.
3: **while** $(t+1) < Th$ **do**
4: **for each** $l \in \mathcal{N}$ **do**
5: Compute the classifier $f_l(t+1)$ with pre-processed data using Eq. (10)
6: Broadcast $f_l(t+1)$ to all the neighboring vehicles $w \in P_l$ through communication agent
7: compute the dual variable $\lambda_l(t+1)$ using Eq. (11)
8: **end for**
9: **end while**

V. Differential Privacy-Based Distributed Machine Learning for SP-CIDS

In the ADMM-based distributed learning algorithm, although the data is not shared explicitly, the collaboration has potential privacy risks such as immunity to data conditioning and robustness to noise and errors which is alleviated using the notion of DP. The following adversary model is considered for the SP-CIDS system.

A. Adversary Model

The adversary model describes the capabilities of the adversary against security, privacy, and trust of the VANET system. The adversary is capable of:

- Being a malicious insider of the system, i.e., a legal vehicle or roadside unit of the VANET system that can perform suspicious attacks.
- Accessing the distributed learning algorithm's output at each iteration as well as the final output.
- Observing the classifier (f_l) broadcast during each iteration of an distributed learning algorithm to gather information about the confidential data point, $(x_s, y_s) \in D_l$.

Consider that a dataset D_l stored at a vehicle $l \in \mathcal{N}$ contains a data point (x_s, y_s), which is confidential. In SP-CIDS, linear classifier f_l is considered, which is a function of labeled training data points. Let $A_1(.) : \mathbb{R}^d \to \mathbb{R}$ represent the distributed learning algorithm with the output $f_l = A_1(D_l)$, where D_l is the dataset used by the vehicle l. Consider another dataset D_l', which is a neighboring dataset of D_l, such that it differs from D_l by a data instance. Let these data points be $(x_s, y_s) \in D_l$ and $(x_s', y_s') \in D_l'$, such that $(x_s, y_s) \neq (x_s', y_s')$, i.e., the hamming distance between the datasets D_l and D_l' is always 1. Consider a potential adversary of this system, who knows all data in the dataset D_l except the data point (x_s, y_s). The malicious node can extract information about the sensitive data (x_s, y_s) stored in the vehicle by observing the output at each iteration of the non-private learning algorithm.

To protect the privacy of the training dataset, modifications can be introduced. However, such modifications in the dataset should only affect the algorithm's output distribution marginally [23]. In SP-CIDS, this is accomplished through DP by inducing randomness to the intermediate classifiers shared across the ADMM algorithm, which is outlined in the following subsections.

B. Differential Privacy

The SP-CIDS uses $\varepsilon-$DP mechanism in conjunction with DML techniques to preserve the privacy of the training data. A randomized mechanism S gives $\varepsilon - DP$ for every set of output X, and for any neighbor dataset of L and L',

if S satisfies the following condition : $P[S(L) \in X] \leq exp(\varepsilon) \times P[S(L') \in X]$, where ε is the privacy parameter [29]. In this technique, the adversaries cannot statistically infer the training data by observing the intermediate states shared during the decentralized learning. An ML technique is said to be private if the output distribution of the algorithm is identical or similar to any of the two neighboring datasets. The datasets L and L' are said to neighbors if and only if they are symmetrical and have the same attribute structure, differing from each other by one data instance. [23].

Thus, with the aid of $\varepsilon-$DP, the SP-CIDS protects a secret data instance, even though the adversary has access to all other non-secret data instances except the secret information.

C. Differential Privacy Based Machine Learning Algorithm for SP-CIDS

The proposed SP-CIDS system uses the laplacian noise mechanism to facilitate DP based privacy protection. This noise is drawn from a laplacian distribution with parameter λ and is added to a function f to provide ε-DP as shown in Eq. (12).

$$f(x) + Lap(\Delta f/\varepsilon) \tag{12}$$

where $\lambda = (\Delta f/\varepsilon)$, x is the input to the function f, and Δf represents the L1-sensitivity, which is the change in the output value of a function f, when only one instance of input changes. Thus, an initial positive result in DP depends on the sensitivity of the function to be learned [23]. The differential privacy based DML is illustrated in Algorithm 3 which gets the privacy parameter ε as input and produces the final private global classifier $f_l(t+1)$ at the end of Th iterations. In step 5, a laplacian noise perturbation is performed on the objective function. In steps 7-9, before broadcasting the $f_l(t+1)$, the output is perturbed using laplacian noise and the dual variable $\lambda_l(t+1)$ is computed. Thus, by observing the intermediate and final output of Algorithm 3, no adversary can obtain any useful knowledge due to the perturbation of the intermediate classifiers using DP, thereby preserving the privacy of each vehicle's training data.

D. Evaluation of Different Classifiers in SP-CIDS

SP-CIDS is implemented and evaluated using three ML classification algorithms: Logistic Regression (LR), Naive Bayes (NB) and ensemble classifier. The loss functions of LR and NB are shown in Eq. (13) and (14).

$$\mathscr{L}_{lr}\left(y_{il}, f^T x_{il}\right) = log\left(1 + exp\left(y_{il} f^T x_{il}\right)\right) \tag{13}$$

$$\mathscr{L}_{nb}\left(y_{il}, f^T x_{il}\right) = -log\, P(x_{il}, y_{il}) \tag{14}$$

To enhance the accuracy of the SP-CIDS system, it is also evaluated with an ensemble classifier, which uses a combination of outcomes from multiple classifiers. Let the decision of j^{th} classifier is denoted as $d_{j,C} \in (-1, 1)$, $j = \{1, \ldots N\}$, $C = \{1, 2, \ldots, k\}$, where N is the number of classifiers, C is the set of class labels and k is the number of classes. The output of the ensemble method is calculated as follows:

$$\sum_{j=1}^{N} d_{j,C} = max_{j=1}^{k} \sum_{j=1}^{N} d_{j,C} \tag{15}$$

In the proposed SP-CIDS mechanism, the decision $d_{j,C} = 1$ denotes the occurrence of an intrusion whereas, the decision $d_{j,C} = -1$ denotes its absence. These results of individual classifiers are aggregated and the decision with the majority of votes is the final decision of the ensemble classifier.

Algorithm 3 Differential Privacy Based DML

Input: Network audit data from all the vehicles ($\hat{D}$), privacy parameter (ε)
Output: Updated private global classifier (f_l)

1: For each vehicle $l \in \mathcal{N}$, the classifier $\{f_l\}_{l=1}^{N}$ is randomly initialized.
2: The dual variable, λ_l is initialized to 0 for all the vehicles in the VANET.
3: **while** $t + 1 < Th$ **do**
4: **for each** $l \in \mathcal{N}$ **do**
5: Choose a random laplacian noise and perturbate the objective function $f_l(t)$ of ADMM algorithm, with privacy parameter ε, as per Eq. (12).
6: Compute the classifier $f_l(t+1)$ with pre-processed data using Eq. (10).
7: Choose a random laplacian noise and perturbate the output function $f_l(t+1)$ of ADMM algorithm, with a privacy parameter ε, as per Eq. (12).
8: Broadcast $f_l(t+1)$ to all the neighboring vehicles $w \in P_l$ through communication agent.
9: compute the dual variable $\lambda_l(t+1)$ using Eq. (11)
10: **end for**
11: **end while**

TABLE II
SIMULATION PARAMETERS

Parameter	Value
Simulated Area	4 km ×4 km
Lane	Bi-directional
Vehicle Density	60-100 vehicles/km
Transmission Range	300 m
Communication Protocol (V2V/V2I)	DSRC
Vehicle Transmission Power	10 dBm
Bandwidth	10 MHz
Packet Size	400 bytes
Data Rate	8 Mbps

VI. IMPLEMENTATION AND RESULTS

The SP-CIDS system is simulated using NS-2, with VANET parameters as shown in Table II. The classifier and DP mechanisms are implemented in python and tested with the NSL-KDD dataset. NSL-KDD dataset is a modified and enhanced version of the KDDCup99 dataset. Filtering techniques are used to remove redundant records to enhance the dataset, which prevents the ML algorithm from producing biased results. The NSL-KDD dataset contains 41 attributes

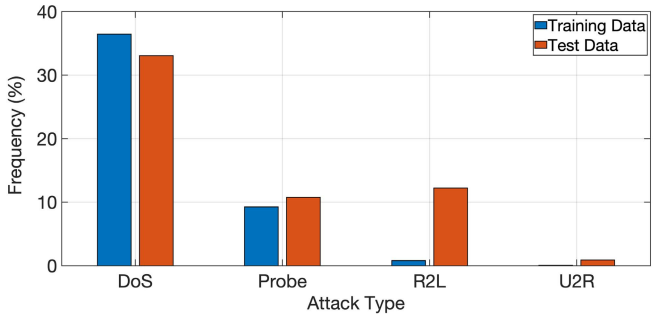


Fig. 3. Attack distribution in the NSL-KDD dataset.

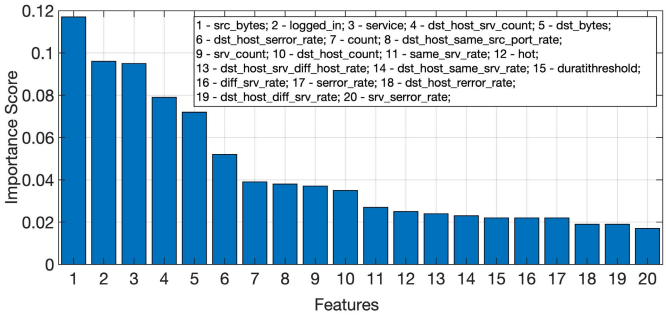


Fig. 4. Importance of features in the NSL-KDD Dataset.

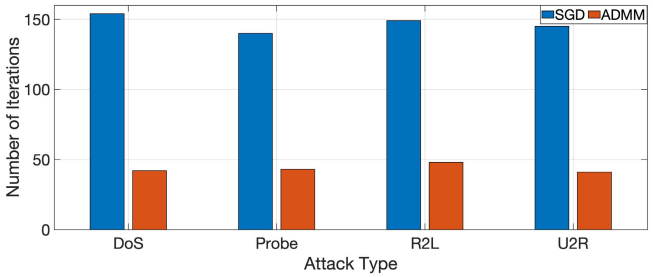


Fig. 5. Performance metrics for the distributed machine learning.

and labeled data, indicating each instance of the dataset, as an attack or non-attack [30]. Fig. 3 displays the distribution of various attacks such as DoS, Probe, Remote-to-Local (R2L), User-to-Root (U2R) used in test and training data.

Feature selection for the pre-processing is done using the Recursive Feature Elimination (RFE) technique which selects the features based on the importance score measure. It selects the important features by recursively pruning the least important features from the given dataset. Fig. 4 shows the importance measure of the features. Data normalization in the pre-processing stage enables faster computations during the ML process and increases the performance of the learner. Furthermore, as illustrated in Fig. 5, DML through ADMM converges much faster in lesser iterations when compared to traditional ML techniques such as Stochastic Gradient Descent (SGD) [31]. This enhances the efficiency of the ADMM based SP-CIDS algorithm to quickly detect the various types of attacks.

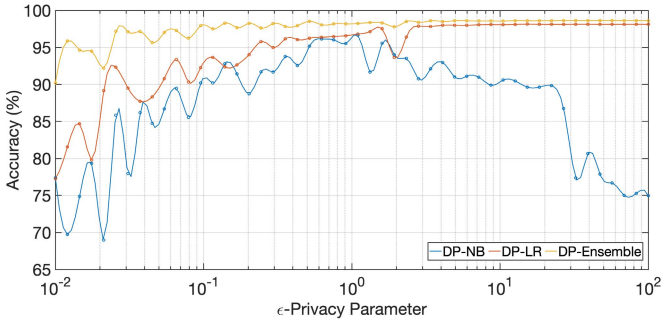


Fig. 6. Impact of privacy parameter over accuracy.

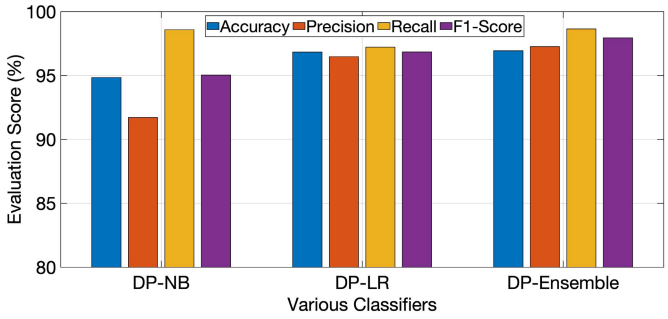


Fig. 7. Performance Metrics for the privately trained classifiers.

A. Privacy Parameter (ε) Identification

In DP, the privacy parameter (ε) controls the amount of noise added to the intermediate classifier shared across the private DML, and is directly proportional to the classifier's accuracy, as inferred in Fig. 6. With increasing accuracy, the adversary may infer more information about the training data through statistical inference mechanisms. Thus, there is a trade-off between privacy and utility of the training data, which is controlled through the ε parameter. As the epsilon increases, the accuracy of the learning increases, with a traded off privacy. Hence, as shown in Fig. 6, the privacy parameter is evaluated from 10^{-2} to 10^{2}. Further, from Fig. 6, it is inferred that the accuracy of DP-NB, DP-LR and DP-Ensemble is approximately in the range of 90-98%, in the ε range between 0.1 and 1. Among these classifiers, the DP-Ensemble performs better in terms of privacy and utility of the training dataset.

B. Model Evaluation of SP-CIDS

Fig. 7 displays the various performance metrics such as accuracy, precision, recall, and F1-score of DP-NB, DP-LR, and DP-Ensemble classifiers, privately trained using the NSL-KDD dataset. It is inferred that the DP-Ensemble learner has a higher accuracy of 96.94%, and also performs much better than the DP-LR and DP-NB in terms of precision, recall, and F1-score. As a result, the usage of the DP-Ensemble classifier with $\varepsilon = 1$ in SP-CIDS ensures better accuracy and training data privacy.

VII. Conclusion

The proposed SP-CIDS detects the attacks using DML and preserves training data privacy using the DP paradigm.

The DP notion achieves effective collaboration without any privacy leakage and motivates vehicle participation in collaboratively detecting an intrusion. SP-CIDS also uses ADMM to decentralize the machine learning algorithm to leverage the distributed nature of VANET and enhance the intrusion detection accuracy of the IDS. Further, simulation results indicate that the DP-Ensemble classifier present in the SP-CIDS provides better accuracy compared to other classifiers, in detecting attacks without compromising the data privacy.

ACKNOWLEDGMENT

This Publication is an outcome of the Research and Development work undertaken in the project under the Visvesvaraya Ph.D. Scheme of Ministry of Electronics and Information Technology, Government of India, being implemented by Digital India Corporation (formerly Media Lab Asia).

REFERENCES

[1] A. Ali *et al.*, "Quality of service provisioning for heterogeneous services in cognitive radio-enabled Internet of things," *IEEE Trans. Netw. Sci. Eng.*, vol. 7, no. 1, pp. 328–342, Jan. 2020.

[2] S. B. M. Baskaran, G. Raja, A. K. Bashir, and M. Murata, "QoS-aware frequency-based 4G+ relative authentication model for next generation LTE and its dependent public safety networks," *IEEE Access*, vol. 5, pp. 21977–21991, 2017.

[3] R. Arul, G. Raja, A. K. Bashir, J. Chaudry, and A. Ali, "A console GRID leveraged authentication and key agreement mechanism for LTE/SAE," *IEEE Trans. Ind. Informat.*, vol. 14, no. 6, pp. 2677–2689, Jun. 2018.

[4] P. Alcoy, S. Bjarnason, P. Bowen, C. Chui, K. Kasavchenko, and G. Sockrider, "NETSCOUT Arbor's 13th annual worldwide infrastructure security report," Netscout Systems, Inc., Burlington, MA, USA, Tech. Rep., 2018.

[5] Headquarters IndustryARC, Telangana, India. *Intrusion Detection System/Intrusion Prevention System (IDS/IPS) Market–Industry Analysis, Market Size, Share, Trends, Application Analysis, Growth and Forecast 2019–2025*. Accessed: 2019. [Online]. Available: https://www.industryarc.com/Research/

[6] G. Raja, A. Ganapathisubramaniyan, S. Anbalagan, S. B. M. Baskaran, K. Raja, and A. K. Bashir, "Intelligent reward-based data offloading in next-generation vehicular networks," *IEEE Internet Things J.*, vol. 7, no. 5, pp. 3747–3758, May 2020.

[7] A.-S. K. Pathan, *Security of Self-Organizing Networks: MANET, WSN, WMN, VANET*. Boca Raton, FL, USA: CRC Press, 2016.

[8] R. Mitchell and I.-R. Chen, "A survey of intrusion detection techniques for cyber-physical systems," *ACM Comput. Surveys*, vol. 46, no. 4, pp. 1–29, Apr. 2014.

[9] G. Raja, S. Anbalagan, G. Vijayaraghavan, P. Dhanasekaran, Y. D. Al-Otaibi, and A. K. Bashir, "Energy-efficient End-to-End security for software defined vehicular networks," *IEEE Trans. Ind. Informat.*, early access, Jul. 28, 2020, doi: 10.1109/TII.2020.3012166.

[10] B. I. Barry and H. A. Chan, "Intrusion detection systems," in *Handbook of Information and Communication Security*. Berlin, Germany: Springer, 2010, pp. 193–205.

[11] R. Hofstede *et al.*, "Flow monitoring explained: From packet capture to data analysis with NetFlow and IPFIX," *IEEE Commun. Surveys Tuts.*, vol. 16, no. 4, pp. 2037–2064, 4th Quart., 2014.

[12] E. Vasilomanolakis, S. Karuppayah, M. Mühlhäuser, and M. Fischer, "Taxonomy and survey of collaborative intrusion detection," *ACM Comput. Surv.*, vol. 47, no. 4, pp. 1–33, Jul. 2015.

[13] H. Huang, H. Xu, Y. Cai, R. S. Khalid, and H. Yu, "Distributed machine learning on smart-gateway network toward real-time smart-grid energy management with behavior cognition," *ACM Trans. Des. Autom. Electron. Syst.*, vol. 23, no. 5, pp. 1–26, Oct. 2018.

[14] M. Fredrikson, S. Jha, and T. Ristenpart, "Model inversion attacks that exploit confidence information and basic countermeasures," in *Proc. 22nd ACM SIGSAC Conf. Comput. Commun. Secur. (CCS)*, 2015, pp. 1322–1333.

[15] F. Cuppens and A. Miege, "Alert correlation in a cooperative intrusion detection framework," in *Proc. IEEE Symp. Secur. Privacy*, May 2002, pp. 202–215.

[16] Z. Zhang, J. Li, C. Manikopoulos, J. Jorgenson, and J. Ucles, "HIDE: A hierarchical network intrusion detection system using statistical preprocessing and neural network classification," in *Proc. IEEE Workshop Inf. Assurance Secur.*, Jun. 2001, pp. 85–90.

[17] C. V. Zhou, S. Karunasekera, and C. Leckie, "Evaluation of a decentralized architecture for large scale collaborative intrusion detection," in *Proc. 10th IFIP/IEEE Int. Symp. Integr. Netw. Manage.*, May 2007, pp. 80–89.

[18] Z. El Mrabet, M. Ezzari, H. Elghazi, and B. A. El Majd, "Deep learning-based intrusion detection system for advanced metering infrastructure," in *Proc. 2nd Int. Conf. Netw., Inf. Syst. Secur. (NISS)*, 2019, pp. 1–7.

[19] P. Liu, "An intrusion detection system based on convolutional neural network," in *Proc. 11th Int. Conf. Comput. Autom. Eng. (ICCAE)*, 2019, pp. 62–67.

[20] O. Kachirski and R. Guha, "Effective intrusion detection using multiple sensors in wireless ad hoc networks," in *Proc. 36th Annu. Hawaii Int. Conf. Syst. Sci.*, Jan. 2003, p. 8.

[21] G. Fernandes, L. F. Carvalho, J. J. P. C. Rodrigues, and M. L. Proença, "Network anomaly detection using IP flows with principal component analysis and ant colony optimization," *J. Netw. Comput. Appl.*, vol. 64, pp. 1–11, Apr. 2016.

[22] Q. Zhu, C. Fung, R. Boutaba, and T. Basar, "GUIDEX: A game-theoretic incentive-based mechanism for intrusion detection networks," *IEEE J. Sel. Areas Commun.*, vol. 30, no. 11, pp. 2220–2230, Dec. 2012.

[23] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," in *Proc. Theory Cryptogr. Conf.* Berlin, Germany: Springer, 2006, pp. 265–284.

[24] Z. Huang, R. Hu, Y. Guo, E. Chan-Tin, and Y. Gong, "DP-ADMM: ADMM-based distributed learning with differential privacy," *IEEE Trans. Inf. Forensics Security*, vol. 15, pp. 1002–1012, 2020.

[25] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Found. Trends Mach. Learn.*, vol. 3, no. 1, pp. 1–122, 2011.

[26] T. Zhang and Q. Zhu, "Dynamic differential privacy for ADMM-based distributed classification learning," *IEEE Trans. Inf. Forensics Security*, vol. 12, no. 1, pp. 172–187, Jan. 2017.

[27] P. A. Forero, A. Cano, and G. B. Giannakis, "Consensus-based distributed support vector machines," *J. Mach. Learn. Res.*, vol. 11, pp. 1663–1707, Jan. 2010.

[28] T. Zhu, G. Li, W. Zhou, and P. S. Yu, "Differentially private data publishing and analysis: A survey," *IEEE Trans. Knowl. Data Eng.*, vol. 29, no. 8, pp. 1619–1638, Aug. 2017.

[29] K. Chaudhuri and C. Monteleoni, "Privacy-preserving logistic regression," in *Proc. Adv. Neural Inf. Process. Syst.*, pp. 289–296, 2009.

[30] M. Ring, S. Wunderlich, D. Scheuring, D. Landes, and A. Hotho, "A survey of network-based intrusion detection data sets," *Comput. Secur.*, vol. 86, pp. 147–167, Sep. 2019.

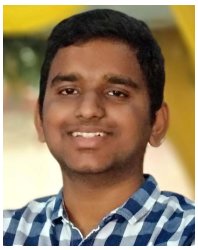
[31] L. Bottou, "Large-scale machine learning with stochastic gradient descent," in *Proc. COMPSTAT*. Berlin, Germany: Physica-Verlag, 2010, pp. 177–186.

Gunasekaran Raja (Senior Member, IEEE) received the Ph.D. degree from the Faculty of Information and Communication Engineering, Anna University, Chennai, India, in 2010. He is currently working as a Professor and the Head of the Department of Computer Technology, Anna University, where he is also the Principal Investigator with the NGNLab. He was a Post-Doctoral Fellow with the University of California at Davis, USA. His current research interests include the Internet of Vehicles, UAV communications, the Internet of Things, 5G networks, machine learning, and wireless security. He is a Senior Member of ACM and a Lifetime Member of ISTE and CSI. He was a recipient of the Young Engineer Award from the Institution of Engineers India in 2009, the FastTrack Grant for Young Scientist from the Department of Science and Technology in 2011, the Professional Achievement Award from the IEEE Madras Section in 2017, and the Visvesvaraya Young Faculty Research Fellowship from the MeitY, Government of India, in 2019.

Sudha Anbalagan received the Ph.D. degree from the Department of Information Technology, Anna University, Chennai, India. She is currently working as an Assistant Professor with the School of Computing, College of Engineering and Technology, SRM Institute of Science and Technology, Chennai. She has also worked with Tata Consultancy Services Limited, as an IT Analyst. She was also a Visiting Research Fellow with the Department of Computer Science, University of California at Davis, USA, in 2015. Her research interests include vehicular networks, UAVs, the Internet of Things, software-defined networking, network security, and machine learning. She was a recipient of the Anna Centenary Research Fellowship, during her Ph.D. Programme.

Geetha Vijayaraghavan received the bachelor's degree in computer science and engineering from Anna University, Chennai, India, in 2006, where she is currently pursuing the master's degree in computer science and engineering. She worked with FIS Global Solutions, Chennai, as a Test Lead. Her areas of interests include machine learning, crypto systems, artificial intelligence, and soft computing.

Sudhakar Theerthagiri received the M.E. degree in computer science and engineering from Annamalai University, India, and the Ph.D. degree from the Faculty of Information and Communication Engineering, Anna University, Chennai. He is currently working as an Assistant Professor with the Department of Computer Science and Engineering, Amrita School of Engineering, Amrita Vishwa Vidyapeetham, Chennai, India. His research interests include cryptography, network security, design of security protocols, and algorithms.



Saran Vaitangarukav Suryanarayan (Member, IEEE) is currently pursuing the B.E. degree in computer science and engineering from Anna University, Chennai. He was an Intern at Fourkites India Pvt., Ltd., and at Freshworks Technologies Pvt., Ltd. His areas of interests include artificial intelligence, UAV communication, algorithms, and big data analysis. He was a recipient of the Meritorious Student Awards, Kamakoti Scholarship, MIT Silver Jubilee Scholarship, and R. K. Murthy Memorial Scholarship from Anna University in the year 2019.

Xin-Wen Wu (Senior Member, IEEE) received the Ph.D. degree from the Chinese Academy of Sciences. He held research positions at the University of California at San Diego (UCSD), USA, as a Post-Doctoral Researcher, and at The University of Melbourne, Australia, as a Research Fellow. He was with the faculty of School of Information Technology and Mathematical Science, University of Ballarat, Australia, and with the faculty of School of Information and Communication Technology, Griffith University, Gold Coast, Australia. He is currently with the faculty of Department of Mathematical and Computer Sciences, Indiana University of Pennsylvania, USA. He also held several visiting positions at the University of Waterloo, Canada; Nanyang Technological University, Singapore; and the University of Duisburg-Essen, Germany. He has been working in the areas of data and cyber security, networking, and coding techniques. His research focuses on security and reliability for emerging networked and distributed systems. He has coauthored over 90 peer-reviewed articles and book chapters and three books and monographs published by prestigious publishers, including Cambridge University Press and Springer.