Journal of the Korea Academia-Industrial cooperation Society
Vol. 25, No. 1 pp. 1-11, 2024

https://doi.org/10.5762/KAIS.2024.25.1.1
ISSN 1975-4701 / eISSN 2288-4688

V2V 통신 환경에서 DoS 공격 탐지 모델 개발을 위한 데이터 생성 및 검증

이현로[1], 이민종[2], 하재철[2*]
[1]호서대학교 정보보호학과 [2]호서대학교 컴퓨터공학부

Data Generation and Verification for Development of DoS Attack Detection Model in V2V Communication Environment

Hyeonro Lee[1], Minjong Lee[2], Jaecheol Ha[2*]
[1]Department of Information Security, Graduate School, Hoseo University
[2]Division of Computer Engineering, Hoseo University

요 약 최근 자율주행차 운행 환경에서는 도로 상황 인식이나 운전 결정을 내리는데 딥러닝 기술을 사용하고 있다. 또한, 딥러닝 기술만을 사용한 자율주행은 한계점이 존재하기 때문에 차량용 애드혹 네트워크(Vehicular Ad-hoc Network, VANET) 통신을 활용하고 있다. 그러나 VANET 통신은 DoS(Denial of Service) 공격과 같은 사이버 공격에 노출될 수 있는 취약점을 내포하고 있어 이를 방어하기 위한 연구가 진행 중에 있다. 본 논문에서는 VANET의 V2V 통신 환경하에서 DoS 공격을 탐지할 수 있는 머신러닝 모델을 개발하기 위한 데이터 셋을 생성한다. 데이터 셋은 OMNeT++, SUMO, Veins, INET과 같은 시뮬레이션 툴을 사용하여 V2V 통신 환경의 속성과 공격 특성을 반영하여 생성하였다. 또한, 생성된 공격 데이터 셋이 다양한 머신러닝 모델에서 공격이 탐지될 수 있는지 그 유효성을 검증하였다. 평가 결과, 생성된 데이터 셋은 학습한 대부분의 머신 러닝 모델에서 약 97%이상의 정확도로 DoS 공격을 탐지할 수 있어, 침입 탐지 모델 학습에 유용하게 사용할 수 있음을 확인하였다.

Abstract In recent years, autonomous vehicles have been using deep learning to recognize road conditions and make driving decisions. In addition, autonomous driving that uses only deep learning technology has limitations, so it utilizes vehicular ad-hoc network (VANET) communications. However, VANET communications contains vulnerabilities that can be exposed to cyber-attacks such as denial of service (DoS), and research is underway to defend against them. In this paper, we generate a dataset to develop a machine learning model that can detect DoS attacks in the V2V communications environment of VANETs. The dataset is generated using simulation tools, such as OMNeT++, SUMO, Veins, and INET, to reflect the attributes of V2V communications and characteristics of the attacks. In addition, the attack dataset generated is validated to see if attacks can be detected by various machine learning models. The evaluation results show that the generated dataset can detect DoS attacks with an accuracy of about 97% or higher from most of the trained machine learning models, which is useful for training intrusion detection models.

Keywords : Self-Driving Car, VANET, Denial of Serveice, Dataset, Machine-learning

본 논문은 2021년도 교육부의 재원으로 한국연구재단의 지원을 받아 수행된 지자체-대학 협력기반 지역혁신 사업의 결과입니다.
(No. 2021RIS-004)
*Corresponding Author : Jacheol Ha(Hoseo Univ.)
email: jcha@hoseo.edu
Received November 2, 2023 Revised December 4, 2023
Accepted January 5, 2024 Published January 31, 2024

1. 서론

최근 첨단 ICT 기술을 적용하여 사람의 조작 없이 차량 스스로 운행하도록 하는 자율주행 자동차가 개발되고 있다. 자율주행 자동차는 교통사고 예방은 물론 운전의 수월성, 차량 내 업무 가능, 교통흐름 파악과 같은 장점으로 인해 관련 시장이 매년 크게 증가하고 있다. 그 동안 자율주행 자동차는 자동차에 설치된 레이더를 포함한 각종 센서와 카메라를 통해 물체를 감지하고 자체 장비와 인공지능 기술을 활용하여 목적지까지 주행하는 방식으로 연구되어 왔다. 그러나 센서의 인식 범위, 영상 처리의 오인식 문제 등 새로운 문제점들이 발견되어 이를 보완하고자 차량과 도로 시설물 사이의 상호 정보 교환 필요성이 대두되었다.

이에 따라 사물 인터넷 기술을 적용한 차세대 지능형 교통체계(Cooperative-Intelligent Transport Systems, C-ITS)가 제안되었다. C-ITS는 차량과 차량, 혹은 차량과 인프라 간에 양방향으로 데이터를 교환함으로써 도시에서 발생하는 교통 문제들에 신속하게 대응하고 예방하는 것을 목표로 하는 시스템이다[1].

차량용 애드혹 네트워크(Vehicular Ad-hoc Network, VANET)는 C-ITS의 목표인 데이터 교환을 실현하기 위한 기술 중 하나로서 차량 간 무선 통신을 통해 정보를 실시간으로 전달하고 교통 상황을 공유한다. 그러나 VANET은 사이버 공격에 노출될 수 있는 취약한 지점을 내포하고 있으며 이에 대한 잠재적인 위협들이 지속적으로 증가하고 있다[2]. 예를 들어 사이버 공격자는 교통 데이터를 변조하거나 중단하고 차량 간 통신을 방해할 수 있으며, 차량의 위치 및 운전자 정보와 같은 민감한 개인 정보를 탈취하거나 차량의 무단 접근을 시도할 수 있다. 특히, 자율주행 자동차에 대해 가용성을 침해하는 서비스 거부 공격(DoS, Denial of Service)이 발생한다면 큰 교통 혼잡이나 교통 사고가 발생할 수 있다.

따라서 V2V 통신 환경에서의 DoS 공격에 대응할 수 있는 연구가 필요하며, 현재까지 제안된 대응 방법으로는 침입 탐지 시스템, 암호화 및 인증, 접근 제어, SDN(Software-Defined Networking) 등과 같은 기술이 사용되고 있다.

기존 연구로 K. Verma 등은 해시 기능을 가지는 Bloom-filter 사용하여 VANET에서 해당 IP가 포함된 메시지가 사용되지 못하게 막는 IP-Chock 알고리즘을 제시한 바 있다[3]. 또한, A. Kumar 등은 VANET의 라우팅 환경에서 발생할 수 있는 Blackhole 공격을 탐지하기 위해 라우팅 프로토콜의 헤더 정보에 서명을 추가하여 패킷의 무결성을 보장하고 패킷의 전송 경로를 기반으로 블랙홀 노드를 탐지하여 다른 노드에게 경고 메시지를 전송하는 알고리즘을 제시하였다[4]. Y. Zeng 등은 VANET에서 발생할 수 있는 다양한 공격 유형을 고려하여 1D Convolutional Neural Network(1D CNN)와 Long Short-Term Memory(LSTM)을 결합한 딥러닝 모델을 통해 네트워크 트래픽을 분석하는 침입 탐지 시스템을 제시하였다[5].

본 논문에서는 VANET의 V2V(Vehicle-to-Vehicle) 통신 환경에서 발생할 수 있는 사이버 공격 중 차량의 가용성을 침해하는 DoS 공격의 유형을 분석하고, 머신러닝 기반의 침입탐지 모델을 개발하는데 필요한 데이터 셋을 생성하고자 한다. 공격용 데이터 셋은 OMNeT++, SUMO, Veins, INET과 같은 시뮬레이션 툴을 활용하였으며 시나리오 장소는 서울특별시의 여의도로 설정하여 현실성 있는 환경을 구축하였다.

V2V 통신에서 데이터 셋을 생성하기 위한 DoS 공격 시나리오는 Data Flooding, RREQ Flooding, Blackhole 공격으로 설정하였다. 생성된 데이터 셋은 머신러닝 기반의 탐지 모델 개발에 잘 적용할 수 있는지 여부를 검증하기 위해 다양한 머신러닝 모델을 학습하는데 사용되었다. 평가 결과, 생성된 데이터 셋은 대부분의 머신 러닝 모델에서 약 97%이상의 정확도로 DoS 공격을 탐지할 수 있음을 확인하였다.

2. 배경 지식

2.1 VANET 통신

VANET은 차량에 탑재된 차량 단말기(Onboard Unit, OBU)와 노변 기지국(Road Side Unit, RSU)을 통해 차량 간 통신을 제공한다. VANET의 목적은 모바일 사용자가 도로에서 이동 중에 끊김없는 연결성을 제공하고 차량 간 통신을 통해 지능형 교통 시스템(ITS)을 구축하는 것이다. VANET은 차량의 규모, 속도, 지리적 위치의 중요성에 따라 다양하며 연결성이 불안정한 특징을 가지고 있어 라우팅, 데이터 전달 및 공유, 보안과 같은 다양한 연구가 진행 중이다.

VANET의 구조는 Fig. 1과 같은데 범위가 제한된 네 대의 차량은 RSU로부터 정보를 얻을 수 있지만 RSU 통신 범위 밖에 있는 다른 차량들은 RSU와 통신할 수 없어 V2V 연결을 통해 통신해야 한다.

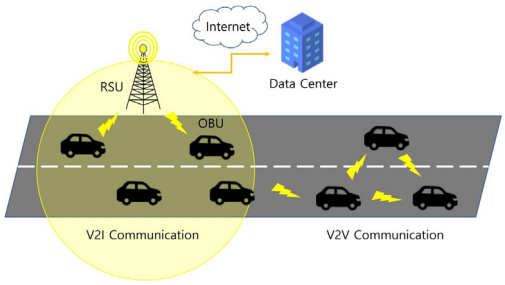

Fig. 1. VANET architecture

각 차량은 차량 간 통신을 통해 동적 애드혹 네트워킹 환경에서 주변 차량과 통신하므로 충돌 경고, 도로 상태 경고, 병합 지원 및 감속 경고와 같은 애플리케이션을 사용할 수 있다. 또한, VANET은 이메일 액세스, 웹 브라우징 오디오 및 비디오 스트리밍과 같은 인터넷 연결에도 사용할 수 있다. 안정적인 인터넷 연결을 위해 높은 대역폭이 강조되는 일부 연결 관련 애플리케이션 GPS 및 내비게이션 시스템도 VANET 통신의 이점을 누릴 수 있으며 교통 보고서와 통합되어 최종 사용자에게 가장 빠른 경로를 제공할 수 있다.

2.2 AODV 라우팅 프로토콜

AODV(Ad-hoc On-Demand Distance Vector)는 무선 환경에서 동적으로 변하는 네트워크 토폴로지에 효과적인 라우팅을 제공하기 위한 프로토콜로서 라우팅 정보를 필요할 때만 요청하는 반응형 라우팅 프로토콜이다[6]. AODV는 고속으로 주행하는 고속도로와 같은 상황에서 타 라우팅 프로토콜보다 다소 성능이 떨어지기는 하나, 본 연구의 실험과 같이 데이터가 정기적으로 교환되는 도심 지역에서는 처치량과 성능 측면에서 DSR(Dynamic Source Routing), DSDV(Destination Sequenced Distance Vector), OLSR(Optimized Link State Routing)과 같은 타 라우팅 프로토콜보다 높은 성능을 보이는 것이 실험을 통해 검증되었다[7-9]. 이에 따라 본 논문에서는 VANET의 라우팅 프로토콜을 AODV로 채택하고 해당 프로토콜의 취약점을 분석하여 공격 시나리오를 설정하였다.

AODV의 경로 탐색 프로세스는 송신 노드가 목적지 노드로 패킷을 보내야 하는 경우 시작한다. 송신 노드는 데이터를 전송하려 할 때 라우팅 테이블에 경로 정보가 없으면 RREQ 메시지를 브로드캐스트(Broadcast)한다. 이웃 노드들은 RREQ 메시지를 수신하면 송신자의 IP 주소를 자신의 라우팅 테이블에 저장하고 목적지의 IP가 아닐 경우 Hop Count 1을 증가시켜 다시 브로드캐스트한다. 수 많은 경로를 통해 목적지 노드가 RREQ 메시지를 수신하면 목적지 IP를 확인하고 같을 경우 경로 정보를 포함한 RREP 메시지를 홉 수가 가장 적은 경로로 송신 노드를 향해 전송하여 경로가 설정된다.

2.3 V2V에서의 DoS 공격

V2V 통신 환경에서 발생할 수 있는 서비스 거부공격은 사용되는 프로토콜에 따라 다양하게 나타난다. 본 논문에서는 V2V 통신 환경에서 발생할 수 있는 서비스 거부 공격 중 RREQ Flooding 공격, Blackhole 공격 그리고 Data Flooding 공격에 대해 설명한다.

VANET에서 범용적으로 연구되는 라우팅 프로토콜 AODV는 전체 네트워크에 RREQ 패킷이 넘쳐나 많은 네트워크 리소스가 소모되는 것을 방지하기 위해 한 노드가 초당 일정 수 이상의 RREQ 패킷을 전송할 수 없도록 설계되었다. 송신 노드가 처음 RREQ 패킷을 브로드캐스트한 경우 송신 노드는 RREP 메시지 수신을 위해 왕복 시간 동안 대기하며 왕복 시간 내에 RREP를 받지 못하면 새로운 RREQ 패킷을 전송한다. 이 때 RREQ 패킷에 대한 왕복 대기 시간을 계산할 때 송신 노드는 이진 지수 백오프를 사용하여 설정한다.

하지만 공격자는 이러한 규칙을 위배하여 네트워크 리소스를 소진할 수 있는 RREQ Flooding 공격을 수행할 수 있다[10]. 악의적인 노드가 네트워크의 IP 주소 범위를 알고 있다면 네트워크에 없는 많은 IP 주소나 범위 밖의 IP 주소를 선택하여 이진 백오프 방식으로 인해 점차 늘어나는 왕복 대기 시간을 고려하지 않고 대량의 RREQ 패킷을 연속적으로 전송할 수 있다. 이로 인해 전체 네트워크가 RREQ 패킷으로 가득 차게 되며 통신 대역폭이 고갈되거나 정상 노드들의 라우팅 테이블 저장 공간이 소진되어 새로운 RREQ 패킷을 수신할 수 없게 된다.

AODV에서 발생할 수 있는 서비스 거부 공격의 한 종류인 Blackhole 공격은 라우팅된 네트워크 내의 악의적인 노드가 네트워크 트래픽의 일부 또는 전체를 자신으로 전송되도록 시도하여 데이터 패킷이 목적지에 도달하지 못하고 손실되도록 하는 공격이다[11]. V2V 환경에서 Blackhole 공격은 두 가지 시나리오를 가정한다. 첫 번째는 출발지와 목적지 경로 내부에 악의적인 노드가 있는 경우이고 두 번째는 악의적인 노드가 외부에 있는 경우이다.

Fig. 2는 AODV 경로 설정에서 Blackhole 공격 동작 과정을 보여준다. 악의적인 노드는 네트워크에 브로드캐스트된 RREQ 패킷을 수신한 후 Destination Sequence Number (DSN)가 크고 Hop Count가 작은 RREP를 생성하여 송신 노드에게 다시 보낸다. AODV에서 경로의 우선 순위는 DSN과 Hop Count를 기준으로 평가되기 때문에 송신 노드가 악의적인 노드가 생성한 가짜 RREP를 수신하면 악의적인 노드를 경유하는 경로가 목적지 노드까지의 가장 최단 경로로 잘못 인식하게 된다. 이에 따라 송신 노드가 데이터를 보내게 되면 악의적인 노드가 데이터를 포워딩하지 않고 데이터 흐름을 차단할 수 있다[12].

V2V 환경의 전송 계층에서 발생할 수 있는 Data Flooding 공격은 공격 노드가 네트워크의 모든 노드에 대한 경로를 설정한 뒤 해당 경로를 따라 모든 노드에 대량의 쓸모없는 데이터 패킷을 전송하여 네트워크의 대역폭을 과부하 시킨다[13]. Data Flooding 공격은 주로 여러 노드를 감염시키고 대규모 트래픽을 발생시키는 DDoS(Distributed Denial of Service)형태로 나타난다.

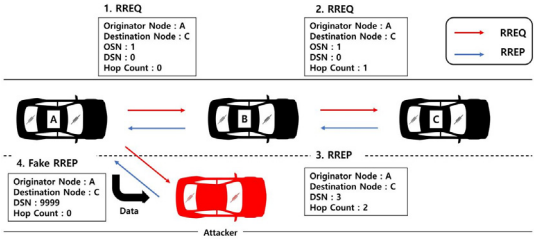

Fig. 2. Overall process of blackhole attack

3. V2V DoS 공격 시뮬레이션

VANET 환경을 시뮬레이션할 때 시뮬레이션 툴을 선택하는 것은 매우 중요한 작업이다. 시뮬레이션 툴을 선택하는 데에는 호환성, 사용자의 편의성, 확장성, 가시성 등 중요 매개변수를 고려해야 한다. 본 연구에서는 Objective Modular Network Testbed in C++(OMNeT++), Simulation of Urban MObility(SUMO), Vehicles in Network Simulation(Veins), INET을 사용하여 시뮬레이션하였으며 각 시뮬레이션 툴의 버전은 Table 1과 같다.

Table 1. Simulation tool version

	Tool	Version
Interface	OMNeT	5.2
Model library	INET	4.2.5
Mobility framework	Veins	5.7
Traffic Generator	SUMO	1.18.0
Operating System	Linux	Ubuntu 22.04

또한, 현실적인 V2V 공격 시나리오를 생성하기 위해 서울 여의도의 교통 상황을 시뮬레이션하였으며 원시 데이터를 얻기까지의 전체 시뮬레이션 개요는 Fig. 3에 도사하였다.

OMNET++는 네트워크 및 분산 시스템 시뮬레이션을 위한 오픈 소스 시뮬레이션 환경으로 네트워크 프로토콜, 통신 시스템, 분산 시스템, IoT, VANET 등 다양한 응용 분야에서의 보안 연구, 성능 분석, 프로토콜 개발에 사용된다[14]. OMNET++는 C++ 언어를 사용하는 모듈 기반 시뮬레이션으로 확장성이 뛰어나며 네트워크 노드, 통신 패킷, 데이터 흐름, 시뮬레이션 결과 등과 같은

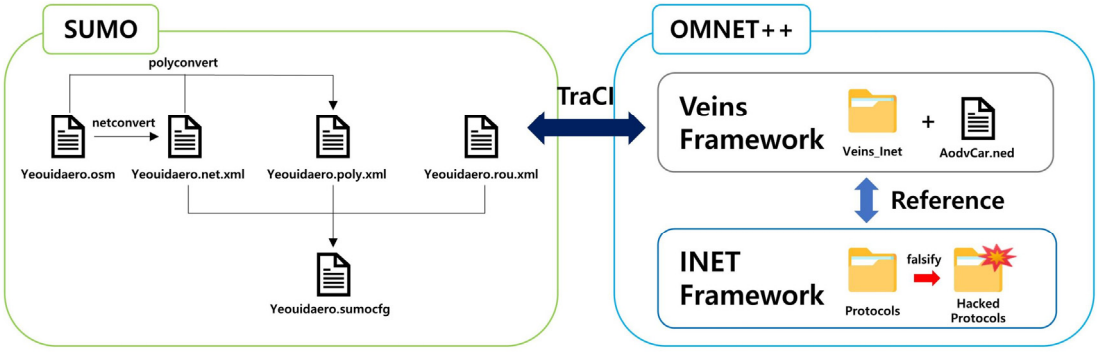

Fig. 3. Overview for DoS attack simulation

요소를 실시간으로 시각화할 수 있는 그래픽 인터페이스를 제공한다. SUMO는 도시 및 교통 시스템을 모델링하고 시뮬레이션하는데 사용되는 오픈 소스 프레임워크로 도시 내의 도로, 신호등, 교차로, 차선, 등을 모델링할 수 있으며 차량의 수, 차량의 속도, 차량의 목적지 등을 정의할 수 있다[15]. Veins는 차량 네트워크 시뮬레이션을 위한 오픈 소스 프레임워크로 트래픽 제어 인터페이스(Traffic Control Interface, TraCI)를 통해 OMNeT++과 SUMO간의 동적 상호 작용을 가능하게 한다[16]. INET은 OMNeT++에서 사용되는 오픈소스 모델 라이브러리로 TCP, UDP, IP 등과 같은 표준 프로토콜과 이더넷, IEEE 802.11과 같은 유무선 인터페이스를 지원한다[17].

3.1 교통 트래픽 생성

교통 트래픽을 발생시키기 위해 SUMO를 사용하며 5단계를 거쳐 Veins 프레임워크와 상호 작용할 수 있는 최종 SUMO 파일을 생성한다. 첫 번째는 시뮬레이션 영역 추출 단계로 지도 데이터베이스 Open Street Map을 통해 여의도의 지리 정보를 포함하고 있는 OSM 파일을 다운로드 한다. 다운로드 받은 최초 OSM 파일은 모든 도로 정보를 포함하고 있기 때문에 두 번째 단계에서는 쿼리문을 통해 인도, 자전거 도로와 같이 차량이 지나다닐 수 없는 도로를 제거한다. Fig. 4는 차로 추출 단계 전과 후의 차이를 보여준다.

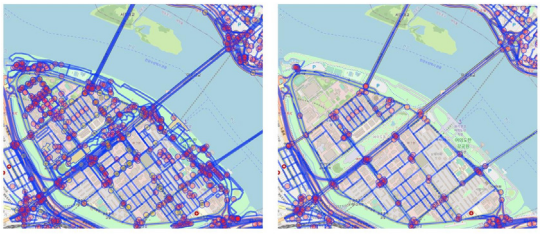

Fig. 4. Result after roadway extraction

세 번째 단계는 쿼리문을 통해 차로 정보만 포함하고 있는 OSM 파일의 형식을 SUMO 서브 툴인 net-convert를 통해 도로, 교차로, 신호등, 차선, 속도 제한 등 도로 네트워크의 모든 정보를 상세하게 나타내는 SUMO 네트워크 형식으로 변환한다. 네 번째 단계에는 SUMO 서브 툴인 poly-convert를 통해 도로 네트워크 정보 외에도 통신에 영향을 끼칠 수 있는 건물, 공원, 강 등 시나리오 내에 존재하는 장애물 정보를 추출한다. 마지막 단계는 차량의 수, 최대 속도, 가속도, 색상, 출발지, 목적지 등과 같이 시나리오에서 차량의 트래픽을 발생시킬 수 있는 정보를 정의하는 Route 파일을 생성한다.

본 연구의 시나리오에서는 출발지-목적지 경로를 두 그룹으로 설정하여 시뮬레이션한다. 첫 번째 그룹은 마포대교-여의대로-여의서로 순으로 이동하며 정상적인 V2V 통신을 하는 애드혹 네트워크 그룹이다. 두 번째 그룹은 여의대방로-여의동로-여의서로 순으로 이동하며 하나의 공격 노드가 V2V 통신을 방해하는 DoS 공격을 수행하는 애드혹 네트워크 그룹이다. 두 애드혹 네트워크 그룹에서 정의한 Route 파일의 정보는 Table 2와 같다.

Table 2. SUMO routing file

Params \ Group	Group 1	Group 2
Vehicle ID	N(Normal)	N(Normal) A(Attacker)
Class	Car	Car
Count	N : 20 A : 0	N : 19 A : 1
Acceleration	2.6	N : 2.6 A : 3
Deceleration	4.5	4.5
Max Speed	14	14
Color	Yellow	N : Yellow A : Red
Length	4	4

3.2 공격 동작을 위한 프로토콜 변조

V2V 통신 시나리오에서 차량이 사용하는 프로토콜은 Table 3과 같으며 INET 라이브러리를 사용하여 구성하였다. 공격 시나리오에서 정상 노드의 경우 INET에서 정의된 표준 프로토콜을 그대로 사용하지만 공격 노드에서 사용되는 라우팅 프로토콜의 경우 악의적인 행동을 수행하도록 표준 프로토콜에서 사용되는 함수를 수정해야 한다. 하지만 INET의 경우 AODV에 있는 함수는 함수 재정의를 허용하도록 설계되지 않았기 때문에 AODV 복사본을 만들고 필요한 함수에 악의적인 동작을 수행하도록 추가한다.

Table 3. V2V communication environment for simulation

Parameter	Value
Channel	Wireless
Physical Model	IEEE 802.11p
Internet Protocol	IPv4
Transport protocol	UDP
Routing protocol	AODV

AODV 라우팅 프로토콜은 한 노드가 초당 일정 수 이상의 RREQ 패킷을 전송할 수 없도록 설계되었기 때문에 RREQ Flooding 공격을 수행하기 위해서 공격자가 설정한 Flooding 주기마다 네트워크 IP 주소 범위 밖의 IP로 설정한 RREQ 패킷을 생성하여 전송하도록 새로운 FloodingRreq 함수를 정의한다. FloodingRreq 함수는 호출될 때 목적지 IP 주소를 1.2.3.4로 설정한 뒤 표준 프로토콜에 정의되어 있는 createRREQ 함수를 호출하여 목적지 주소를 사용한 가짜 RREQ 패킷을 생성하고 가짜 RREQ 패킷을 전송하기 위해 표준 프로토콜에 정의되어 있는 sendRREQ 함수를 호출하여 브로드캐스트 주소로 전송한다. 또한 공격자가 설정한 Flooding 주기마다 FloodingRreq 함수를 호출하도록 스케줄링하여 가짜 RREQ 패킷을 주기적으로 전송하도록 한다.

Blackhole 공격은 경로 설정 단계에서 송신 노드를 속이기 위한 단계와 경로 설정 후 송신 노드가 목적지로 전송하는 패킷을 삭제하는 단계로 나누어 수행된다. 경로 설정 단계에서 송신자가 전송한 RREQ 패킷을 수신한 공격 노드는 송신 노드에게 목적지까지의 최단 경로가 있다고 속이기 위한 가짜 RREP 패킷을 생성하는 FakeCreateRrep 함수를 정의한다. AODV에서 경로 설정은 DSN이 높고 Hop Count가 적은 기준으로 높은 우선 순위를 가지기 때문에 FakeCreateRrep 함수에서 DSN을 9999로 설정하고 Hop Count를 0으로 설정한 뒤 표준 프로토콜에 정의되어 있는 sendRREP 함수를 호출하여 송신 노드에게 가짜 RREP 패킷을 전송한다.

가짜 RREP 패킷을 받은 송신 노드는 공격 노드를 경유하는 경로를 목적지 노드까지의 최단 경로로 인식하게 된다. 경로 설정을 마친 후 공격 노드는 송신 노드에서 목적지 노드로 가는 패킷을 삭제하기 위해 데이터 그램 패킷이 전달될 때 호출되는 ensureRouteForDatagram 함수를 재정의한다. 표준 프로토콜에 정의된 ensureRouteForDatagram 함수는 라우팅 테이블에 목적지 노드까지의 경로가 존재하지 않거나 비활성 상태이면 데이터 그램 패킷을 지연시키고 경로를 찾는 startRouteDiscovery 함수를 호출하여 경로 검색을 시작하거나 데이터 그램 패킷을 큐에 넣어 두고 나중에 처리하도록 동작하지만, 공격 노드가 재정의한 함수는 해당 데이터 그램 패킷을 폐기한다.

Data Flooding은 네트워크의 모든 노드에게 대량의 패킷을 전송하여 네트워크 리소스를 고갈시키는 단순한 동작을 수행하기 때문에 표준 프로토콜을 변조할 필요 없이 UDP 애플리케이션을 사용하여 공격을 수행할 수 있다[18]. UDP 애플리케이션의 유형은 두 가지로 첫 번째는 사용자가 설정한 메시지 전송 간격마다 지정된 IP 주소로 UDP 패킷을 전송하는 역할을 하는 UDPBasicApp이고 두 번째는 UDP 패킷을 수신하고 분석하는 역할을 하는 UDPSinkApp이다. 정상 시나리오에서는 UDPBasicApp을 사용하여 Group 1에 있는 네 대의 노드가 100~1024 Byte 사이의 메시지 크기를 가진 패킷을 0.5~5초마다 전송 및 수신하도록 설정하고 공격 시나리오에서는 공격 노드의 개수, 전송 속도에 따라 네 가지의 상황으로 설정하였다.

첫 번째는 한 개의 공격 노드가 50초 동안 Group 2에 있는 모든 정상 노드에게 메시지 크기가 1400 Byte인 패킷을 100ms마다 전송한다. 두 번째는 한 개의 공격 노드가 30초 동안 Group2에 있는 모든 정상 노드에게 메시지 크기가 1400 Byte인 패킷을 10ms 마다 전송한다. 세 번째는 다섯 개의 노드가 50초 동안 Group 2에 있는 모든 정상 노드에게 메시지 크기가 1400 Byte인 패킷을 100ms마다 전송한다. 네 번째는 다섯 개의 노드가 30초 동안 Group 2에 있는 모든 정상 노드에게 메시지 크기가 1400Byte인 패킷을 10ms마다 전송한다.

정상 노드의 AODV 라우팅 프로토콜은 악의적인 행동을 수행하도록 변조된 AODV 라우팅 프로토콜로 대체되어 공격 노드가 RREQ Flooding 공격과 Blackhole 공격을 수행할 수 있도록 설계하였다. Fig. 5는 정상 차량 모듈의 구성과 RREQ Flooding 공격과 Blackhole 공격을 수행하도록 변조된 AODV 라우팅 프로토콜을 사용하는 차량 모듈 구성을 보여준다.

3.3 시뮬레이션 실행 결과

시뮬레이션을 실행하기 위해 Veins에서 제공되는 VeinsInetManager 모듈을 통해 SUMO 파일에서 실행되는 도로 교통 트래픽을 OMNeT++에서 발생시킬 수 있게 한다. 교통 트래픽은 단순히 차량을 생성, 이동, 소멸하는 동작만 수행하기 때문에 통신 환경을 정의해 놓은 INET 라이브러리와 결합하여 V2V 통신을 수행하는 교통 트래픽 시나리오를 시뮬레이션한다. V2V 통신 환경에서 DoS 공격을 시뮬레이션하기 위해 Fig. 6과 같이 시나리오 파일을 생성하고 시뮬레이션 이름, 시간, 네트워크 구성, 모듈 매개변수 등을 정의할 수 있는 omnetpp.ini파일을 빌드하여 시뮬레이션을 실행시킨다.

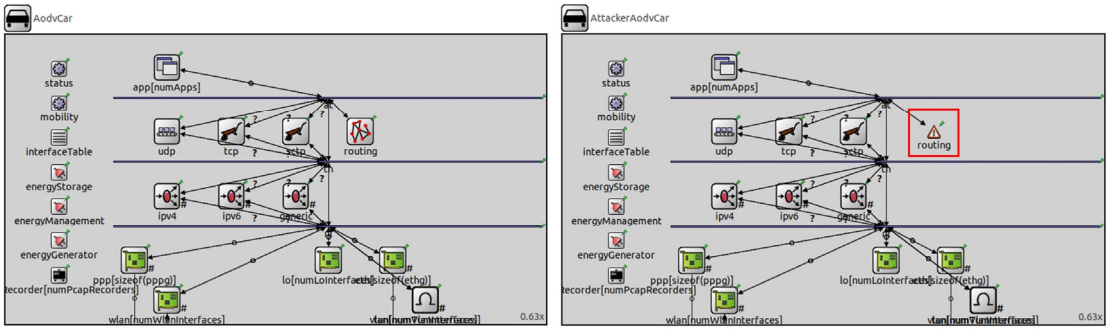

Fig. 5. (left) Normal car module configuration and (right) attacker car module configuration

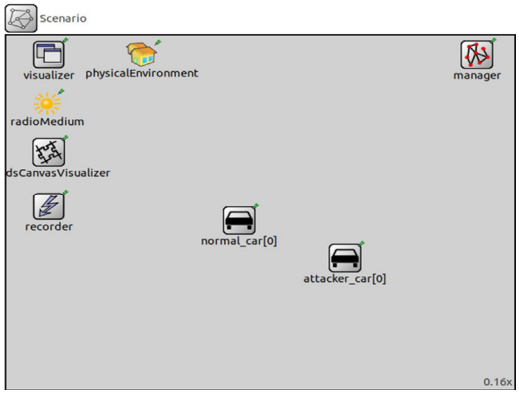

Fig. 6. Scenario design for attack simulation

공격 시나리오 시뮬레이션에서 공격 노드가 Blackhole 공격을 수행하는 결과는 Fig. 7과 같으며 3번노드가 사전에 설정된 경로를 통해 24번 노드에게 데이터를 전송할 때 공격 노드 18번이 패킷이 수신하고다음 라우팅 경로인 22번에게 패킷을 포워딩하지 않고 폐기하는 것을 확인할 수 있다.

시뮬레이션 실행 결과는 기본적으로 로그 파일과 벡터 파일로 제공되며 사용자 설정에 따라 INET에서 제공하는 pcapRecorder를 통해 네트워크 트래픽을 pcap 파일로 저장할 수 있다. 본 연구에서는 데이터 셋을 생성하기 위해 시뮬레이션 도중 모듈 간에 발생하는 이벤트 번호, 시간, 패킷 이름, 송신지 주소, 목적지 주소와 같은 정보를 포함한 로그 파일과 각 노드 별로 네트워크 트래픽을 기록한 pcap 파일을 사용하였다.

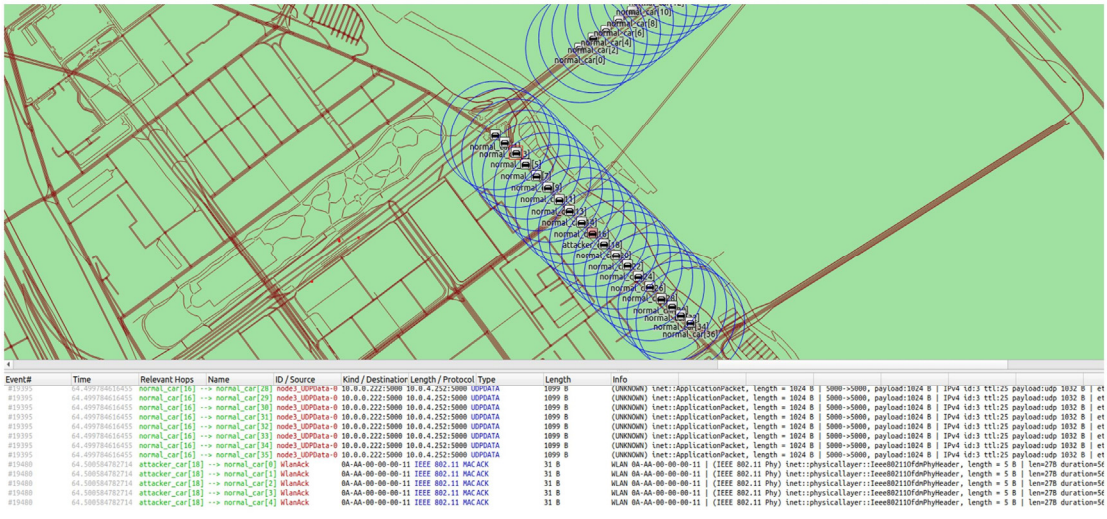

Fig. 7. BlackHole attack simulation

4. 데이터 전처리 및 데이터 셋 평가

4.1 데이터 전처리

네트워크 트래픽을 기록한 pcap 파일은 각 노드 별로 수집되었기 때문에 데이터 셋을 만들기 위해 첫 번째 단계로 데이터 병합 과정을 거친다. 두 번째 단계로 병합된 데이터는 머신러닝 모델이 학습할 수 있도록 각 패킷에서 dpkt 라이브러리를 통해 타임스탬프와 헤더 정보들을 추출하여 csv 파일로 저장한다. 무선 통신에서는 데이터를 브로드캐스트 방식으로 전송하는 특징으로 인하여 각 노드에서 모니터링 모드로 수집된 트래픽 데이터는 통신 범위가 겹치는 부분에서 똑같은 트래픽이 기록될 수 있다. 중복되는 데이터는 학습 능력에 방해될 수 있기 때문에 세 번째 단계에서는 중복 제거를 수행한다. 네 번째 단계는 중복 제거 된 데이터를 로그 파일에 포함된 시뮬레이션 시간, 공격 이벤트 번호, 공격 ip 주소와 비교하여 라벨링을 수행한다. 다섯 번째 단계에서는 각 헤더 정보를 가지고 있는 특징 외에 DoS 공격의 특성을 가지는 특징을 추가하기 위해 각 노드가 패킷을 보내는 시간 간격, 각 노드가 RREQ를 보내는 시간 간격, 특정 IP를 목적지로 하는 패킷의 시간 간격을 계산하고 모델 학습에 영향을 끼칠 수 있는 송수신자 MAC 주소, 송수신자 IP 주소, 송신자 포트 주소를 제거하였다. 마지막으로 이상치의 영향을 완화하는데 도움을 주도록 StandardScaler를 사용하여 데이터 스케일을 조정하였다. 각 공격 유형별로 전처리 과정을 거쳐 생성된 데이터 개수는 Table 4와 같다.

Table 4. Generated dataset using simulation

Class	Count	Ratio(%)
Normal	47070	76.7
RREQ flooding	8460	13.8
blackhole	456	0.7
Data flooding	5407	8.8
TOTAL	61393	100

4.2 데이터 셋 평가

전처리 과정을 거친 데이터 셋의 유효성을 평가하기 위해 Support Vector Machine(SVM), Random Forest (RF), K-Nearest Neighbors(KNN), Multi-Layer Perceptron (MLP)와 같은 지도 학습 기반의 머신 러닝 기법을 사용하여 트래픽 데이터의 성능과 정확도를 평가한다. 모델의 성능 평가 지표로 Accuracy, Precision, Recall, F1-score를 사용하였으며 수식은 다음과 같다.

$$Accuracy = \frac{TP+TN}{TP+FP+TN+FN} \quad (1)$$

$$\mathrm{P}recision = \frac{TP}{TP+FP} \quad (2)$$

$$Recall = \frac{TP}{TP+FN} \quad (3)$$

$$F1-Score = 2 \times \frac{\mathrm{P}recision \times Recall}{\mathrm{P}recision + Recall} \quad (4)$$

전체 데이터 셋 중 정상 데이터의 30%, 각 공격 데이터의 30%를 테스트 데이터 셋로 사용하기 위해 학습 데이터에서 제외하였다. 또한 머신 러닝 학습 과정에서 최적의 하이퍼 파라미터를 찾기 위한 방법으로 Grid Search 기법을 사용하였다. Grid Search 기법은 머신러닝 모델의 하이퍼 파라미터 최적화를 위한 기법으로, 가능한 모든 하이퍼 파라미터 조합을 사전에 정의한 그리드에서 시스템적으로 탐색하여 가장 우수한 모델 성능을 찾는 방법이다. Grid Search 기법을 통해 찾은 각 모델의 최적 하이퍼 파라미터는 Table 5와 같다.

Table 5. Hyperparameters for model optimization

Model	Hyper-Parameter
SVM	C=100, gamma=10
RF	n_estimators=100, criterion=gini
K-NN	n_neighbors=2, weights=distance, metric=euclidean
MLP	hidden_layer=(128, 64, 32), max_iter=100, learning_rate=adaptive

데이터 셋에 대해 머신러닝 모델을 학습시키고 평가한 결과, Table 6과 같이 RREQ Flooding의 경우 모든 머신러닝 모델에서 약 99%, Data Flooding과 Blackhole 공격은 약 97~98%의 높은 F1-Score를 보였다. 이는 본 논문에서 생성한 데이터 셋이 공격 탐지 모델의 입력 데이터로 사용하기에 유효함을 보여주는 결과이다. 이전 연구[5]와 비교해 볼 때, 본 연구에서는 데이터 셋을 얻기 위한 과정을 더 상세히 기술하였으며, 공격 데이터의 개수를 확장함으로써 데이터 불균형 문제를 완화하여 머신 러닝 모델만으로도 더 높은 정확도를 보여주었다.

추가로 고속의 통신 환경에서 머신러닝 기반의 침입 탐지 모델이 실제 환경에서 탐지할 수 있는지 확인하기

Table 6. Dataset evaluation results according to machine learing model

Model	Data	Recall	Precision	F1-score	Accuracy
Support Vector Machine	Data Flooding	0.97350585	1	0.98657509	0.99206496
	RREQ Flooding	0.99763593	1	0.99881657	0.99953358
	Blackhole	0.99009901	0.98039216	0.98522167	0.99940066
Random Forest	Data Flooding	0.987061	0.99875312	0.99287264	0.99575567
	RREQ Flooding	0.99763593	1	0.99881657	0.99953358
	Blackhole	0.99009901	0.96618357	0.97799511	0.99910099
K-Nearest Neighbor	Data Flooding	0.96734442	0.98125	0.9742476	0.98468352
	RREQ Flooding	0.99802994	1	0.999014	0.99961132
	Blackhole	0.98514851	0.95673077	0.97073171	0.99880132
Multi Layer Perceptron	Data Flooding	0.97720271	1	0.98846993	0.99317217
	RREQ Flooding	0.99763593	1	0.99881657	0.99953358
	Blackhole	0.9950495	1	0.99751861	0.99990007

위해 학습된 모델의 추론 시간을 측정해 보았다. 논문에서는 1000개의 패킷을 추론하는 시간을 측정한 결과, Random Forest가 0.013초로 가장 빨랐으며 K-Nearest Neighbor가 0.05초로 추론 시간이 가장 느린 것을 확인하였다. 이러한 DoS 공격의 추론 속도는 구현 시스템마다 다를 수 있으나 실현 가능성이 충분하다고 할 수 있다.

5. 결론

VANET의 V2V 통신 환경에서 가용성이 보장되지 못하면 교통 체증과 인명 피해를 일으키는 심각한 교통 사고가 발생할 수 있다. 따라서 VANET에서 발생할 수 있는 위협을 정확하게 탐지하고 가용성을 보장할 수 있는 대응 방안을 고려해야 한다. 이에 따라 최근 머신 러닝 기반의 침입 탐지 시스템에 대한 연구가 진행되고 있지만 가장 중요한 VANET 공격 데이터 셋은 데이터 수집의 제약 사항들로 인하여 접근이나 공유가 어렵다는 문제가 있다.

따라서 본 논문에서는 V2V 통신 환경에서의 DoS 공격 유형을 탐구하고 서울특별시 여의도를 배경으로 하는 시나리오를 OMNeT++, SUMO, Veins, INET을 사용하여 시뮬레이션하였다. 시뮬레이션에서는 악의적인 동작을 수행하는 노드를 구현하기 위해 표준 프로토콜을 변조하여 사용하였으며 각 공격을 실행시켜 실제 악의적인 행동을 수행하는지 확인하였다. 또한 시뮬레이션에서 추출된 원시 데이터는 전처리 과정을 거쳐 V2V 통신 환경에서의 DoS 공격을 탐지하기 위한 데이터 셋으로 구축되었다. 생성된 데이터 셋에 대해 머신러닝 모델을 학습시킨 결과, DoS 공격을 탐지하는 데 높은 정확도를 보였다. 데이터셋은 향후 우수한 공격 탐지 모델을 개발하는데 유용하게 활용되어 자율주행차에 대한 사이버 공격으로부터 안정성이 향상될 것으로 기대된다.

향후 연구로 서울 여의도 외에 국내 도로 교통 시스템을 참고하여 복잡하고 다양한 현실 세계의 교통 및 도로 상황을 반영한 공격 시나리오를 추가하여 연구를 진행할 계획이다. 또한 VANET의 라우팅 프로토콜에 대한 많은 연구가 진행되고 있기 때문에 AODV 라우팅 프로토콜뿐만 아니라 최신 우수하다고 평가되고 있는 VADD, MOVE, 등과 같은 위치 기반의 라우팅 프로토콜의 취약점을 분석하여 다양한 라우팅 환경에서의 공격 시나리오를 추가할 예정이다.

References

[1] A. Festag, “Cooperative intelligent transport systems standards in Europe,” IEEE communications magazine, 52(12), pp. 166-172, 2014. DOI: https://doi.org/10.1109/MCOM.2014.6979970

[2] H. Hasrouny, A. Samhat, C. Bassil, A. Laouiti, “VANet security challenges and solutions: A survey,” Vehicular Communications, 7, pp. 7-20, 2017. DOI: https://doi.org/10.1016/j.vehcom.2017.01.002

[3] K. Verma, H. Hasbullah, “IP-CHOCK (filter)-Based detection scheme for Denial of Service (DoS) attacks

in VANET," In 2014 International Conference on Computer and Information Sciences (ICCOINS) pp. 1-6, 2014.
DOI: https://doi.org/10.1109/ICCOINS.2014.6868377

[4] A. Kumar, V. Varadarajan, A. Kumar, P. Dadheech, S. Choudhary, V. A. Kumar, K. C. Veluvolu, "Black hole attack detection in vehicular ad-hoc network using secure AODV routing algorithm," Microprocessors and Microsystems, 80, 103352, 2021.
DOI: https://doi.org/10.1016/j.micpro.2020.103352

[5] Y. Zeng, M. Qiu, D. Zhu, Z. Xue, J. Xiong, M. Li,. "DeepVCM: A deep learning based intrusion detection method in VANET," In 2019 IEEE 5th intl conference on big data security on cloud (BigDataSecurity), IEEE intl conference on high performance and smart computing,(HPSC) and IEEE, intl conference on intelligent data and security (IDS) pp. 288-293, 2019.
DOI: https://doi.org/10.1109/BigDataSecurity-HPSC-IDS.2019.00060

[6] C. Perkins, E. Belding-Royer, S. Das. "Ad hoc on-demand distance vector (AODV) routing (No. rfc3561)," 2003.
DOI: https://doi.org/10.17487/rfc3561

[7] T. Marinov, B. Petkova, "Comparative Analysis of AODV and MTP Routing Protocols in VANET," In 2023 58th International Scientific Conference on Information, Communication and Energy Systems and Technologies (ICEST), pp. 257-260, 2023.
DOI: https://doi.org/10.1109/ICEST58410.2023.10187345

[8] M. A. Abdeen, A. Beg, S. M. Mostafa, A. AbdulGhaffar, T. R. Sheltami, A. Yasar, "Performance Evaluation of VANET Routing Protocols in Madinah City," Electronics, 11(5), 777, 2022.
DOI: https://doi.org/10.3390/electronics11050777

[9] M. Sohail, Z. Latif, S. Javed, S. Biswas, S. Ajmal, U. Iqbal, M.. Raza, "Routing protocols in Vehicular Adhoc Networks (VANETs): A comprehensive survey," Internet of Things, 100837, 2023.
DOI: https://doi.org/10.1016/j.iot.2023.100837

[10] P. Yi, Z. Dai, Y. Zhong, S. Zhang. "Resisting flooding attacks in ad hoc networks," In International Conference on Information Technology: Coding and Computing (ITCC'05)-Volume II, Vol. 2, pp. 657-662, 2005.
DOI: https://doi.org/10.1109/ITCC.2005.248

[11] A. Fiade, A. Triadi, A. Sulhi., S. Masruroh, V. Handayani, H. Suseno, "Performance analysis of black hole attack and flooding attack AODV routing protocol on VANET (vehicular ad-hoc network)," In 2020 8th International conference on cyber and IT service management (CITSM), pp. 1-5, 2020.
DOI: https://doi.org/10.1109/CITSM50537.2020.9268789

[12] V. Bibhu, K. Roshan, K. Singh, D. Singh, "Performance analysis of black hole attack in VANET," International Journal of Computer Network and Information Security (IJCNIS), 4(11), pp. 47-54, 2012.
DOI: https://doi.org/10.5815/ijcnis.2012.11.06

[13] M. Sattar, R. Rehman, "Interest flooding attack mitigation in named data networking based vanets," In 2019 International Conference on Frontiers of Information Technology (FIT), pp. 245-2454, 2019.
DOI: https://doi.org/10.1109/fit47737.2019.00053

[14] A. Varga, "OMNeT++. In Modeling and tools for network simulation," Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 35-59, 2010
DOI: https://doi.org/10.1007/978-3-642-12331-3_3

[15] D. Krajzewicz, "Traffic simulation with SUMO–simulation of urban mobility," Fundamentals of traffic simulation, pp. 269-293, 2010.
DOI: https://doi.org/10.1007/978-1-4419-6142-6_7

[16] C. Sommer, D. Eckhoff, A. Brummer, S. Buse, F. Hagenauer, S. Joerer, M. Segata, "Veins: The open source vehicular network simulation framework," Recent Advances in Network Simulation: The OMNeT++ Environment and its Ecosystem, pp. 215-252, 2019.
DOI: https://doi.org/10.1007/978-3-030-12842-5_6

[17] L. Mészáros, A. Varga, M. Kirsche, "Inet framework," Recent Advances in Network Simulation: The OMNeT++ Environment and its Ecosystem, pp. 55-106, 2019.
DOI: http://dx.doi.org/10.1007/978-3-030-12842-5_2

[18] F. Alhaidari, A. Alrehan, "A simulation work for generating a novel dataset to detect distributed denial of service attacks on Vehicular Ad hoc NETwork systems," International Journal of Distributed Sensor Networks, p. 17(3), 15501477211000287, 2021.
DOI: https://doi.org/10.1177/15501477211000287

이 현 로(Hyeonro Lee) [준회원]

- 2023년 2월 : 호서대학교 컴퓨터공학부 (학사)
- 2023년 3월 ~ 현재 : 호서대학교 정보보호학과 석사과정

〈관심분야〉
자동차 보안, 인공지능 보안, 정보보호

이 민 종(Minjong Lee) [준회원]

• 2018년 3월 ~ 현재 : 호서대학교 컴퓨터공학부 학부과정

〈관심분야〉
자동차 보안, 네트워크 보안, 부채널 공격

하 재 철(Jaecheol Ha) [종신회원]

• 1989년 2월 : 경북대학교 전자공학과 (학사)
• 1993년 8월 : 경북대학교 전자공학과 (석사)
• 1998년 2월 : 경북대학교 전자공학과 (박사)
• 1998년 3월 ~ 2007년 2월 : 나사렛대학교 정보통신학과 교수
• 2007년 3월 ~ 현재 : 호서대학교 컴퓨터공학부 교수
• 2009년 1월 ~ 현재 : 한국산학기술학회 이사
• 2023년 1월 ~ 현재 : 국제차세대융합기술학회 부회장
• 2024년 1월 ~ 현재 : 한국정보보호학회 회장

〈관심분야〉
암호학, 네트워크 보안, 부채널 공격, 머신러닝