이 일 구 교수 지도
석사학위 청구논문

종단간 암호화를 위한
악성 암호 트래픽의 고속 탐지 기법

2023

성신여자대학교 대학원
미래융합기술공학과
김 소 연

종단간 암호화를 위한
악성 암호 트래픽의 고속 탐지 기법

이 일 구 교수 지도

이 논문을 석사학위논문으로 제출함

2022년 11월

성신여자대학교 대학원

미래융합기술공학과

김 소 연

인 준 서

김소연의 석사학위 논문으로 인준함

2022년 11월

심사위원장 임 연 섭 (서명 또는 인)

심 사 위 원 김 경 진 (서명 또는 인)

심 사 위 원 이 일 구 (서명 또는 인)

성신여자대학교 대학원

논 문 개 요

네트워크 기술의 발전과 보편화로 초고속 인터넷을 이용해 언제 어디서나 원하는 대상과 통신할 수 있게 되었지만, 정보의 전송 과정에서 감시, 감청, 개인정보 유출, 악성코드 유포가 과거 어느 때보다 쉬워졌다. 최근 이러한 보안 위협에 대응하기 위한 대표적인 정보보호체계로 종단간 암호화(End-to-End Encryption, E2EE) 기술이 상용 네트워크 서비스에 기본적으로 사용되고 있다. 종단간 암호화 기술을 이용하면 암호 키를 알고 있는 메시지 송수신 단말은 메시지를 복호할 수 있게 되지만, 메시지를 검사하고 전달하는 정보보호체계에서 암호화된 패킷의 악성 여부를 검사하기 어려워진다. 종래에는 신뢰할 수 있는 정보보호체계를 이용해 심층 패킷 검사(Deep Packet Inspection, DPI)를 하는 연구가 진행되었으나 심층 패킷 검사를 수행하기 위해서는 종단간 암호화가 유지되기 어렵고 검사 시간이 길어진다는 한계가 있다. 또한, 인공지능을 활용한 악성 트래픽 탐지 및 분류 연구가 수행되고 있지만 페이로드 부분을 학습 요소로 활용하기 어려워서 헤더 정보만 이용하거나 통계적 특징을 추출해서 학습하기 때문에 탐지 성능을 향상하는 데에 한계가 있다.

본 연구에서는 종단간 암호화 상태를 유지하며 빠르게 알려진 악성 패턴을 검출하기 위해 차세대 네트워크 패킷 프레임 구조와 고속 패킷 검사(Fast Packet Inspection, FPI) 기법을 제안한다. 암호 트래픽의 페이로드를 학습 요소로 활용하여 악성 트래픽 학습 속도를 개선하고, 정확도를 향상시키는 Advanced FPI(Advanced Fast Packet Inspection) 프로토콜을 제안하였다. 시뮬레이션 결과에 의하면 제안하는 FPI는 종단간 암호화와 무결성을 유지하면서 심층 패킷 검사 대비 페이로드 길이가 640바이트인 환경에서 검사 커버

리지가 20%인 경우 약 14.4배, 100%인 경우 약 5.3배 빠른 속도로 패킷 검사를 수행할 수 있다. Advanced FPI는 심층 패킷 검사 대비 선형 회귀 모델에서 최소 7.63초, 의사 결정 트리 모델에서 최대 13.74초 학습 시간을 개선하였다. 정확도 성능 측면에서는 종래 방식 대비 최소 1.89%에서 최대 35.08% 악성 트래픽 분류 정확도 성능을 향상하였다.

목 차

표 차 례

그 림 차 례

Ⅰ. 서 론

최근 각종 스마트 기기가 상용화 되면서 개인 정보와 금융 정보를 수집, 처리, 가공, 전송하는 애플리케이션과 서비스가 지속적으로 확산되어 보편화 되고 있다[1]. 특히, 2019년 DMC 보고서에 따르면 스마트 기기 한 대 당 SNS(Social Networking Service)와 채팅 애플리케이션을 최소 1개 이상 이용하고 있을 정도로 활용도가 높은 것으로 조사되었다[2]. 이러한 추세는 COVID 19로 인해 비대면 활동이 늘어날 수밖에 없는 환경에서 지속해서 성장하여 앞으로 스마트 워크, 원격 회의, 원격 수업 등 네트워크를 통한 정보 교류가 폭발적으로 증가할 것으로 예측된다[3]. 하지만 네트워크를 통한 데이터 전송 과정에서 감청, 개인 정보 유출, 프라이버시 침해, 악성 코드와 같은 보안 이슈가 끊임없이 발생하고 있다[4]. 이러한 보안 위협에 대응하기 위해 종단간 암호화 기능이 웹 · 네트워크 애플리케이션의 필수 요소가 되고 있다[5,6].

종단간 암호화란 통신 시 사용하는 암호화키를 서버에 저장하지 않고, 개인 단말기에 저장하여 서버와 공격자의 감청을 불가능하게 하여 통신 보안을 보장하는 방법이다[4]. 암호화된 종단간 링크 환경은 서버로부터 송수신 메시지의 기밀성을 보장하는 큰 이점이 있으나 중간에 데이터가 변조되는 경우 사용자가 확인하기 어렵다는 문제가 존재한다[7]. 그리고 중간 경유지의 패킷 검증 단계에서 트래픽의 악성 유무를 탐지하기 어렵기 때문에 공격자의 악성 행위를 숨길 수 있는 공격 수단이 되기도 한다[8]. 그러나 종단간 암호화 네트워크 환경에서 암호화된 패킷의 악성 행위를 효과적으로 검사하는 방안에 관한 연구는 아직까지 미진한 실정이다.

네트워크 패킷의 악성 여부를 검사하는 방법 중 하나인 심층 패킷 분석

기법은 패킷의 헤더뿐만 아니라 데이터를 담고 있는 페이로드 부분까지 확인하는 기술을 말한다[9]. DPI는 대규모의 동적 네트워크 환경에서 다양한 애플리케이션을 정확하게 식별할 수 있으나 검사 과정 중 프라이버시 침해 가능성이 있고 불필요한 검증으로 인한 처리 속도 효율 저하를 발생시킬 수 있다[10]. 또한, DPI는 전송 과정 중 복호화 과정이 없는 종단간 암호화 환경에서 활용이 어렵다[11]. 이를 개선하기 위한 방법으로 제안된 암호화된 트래픽에 대한 BlindBox DPI 기법은 DPI 미들 박스(middle box)에서 암호화된 트래픽까지 검사가 가능하도록 DPI 필터링을 확장시켰지만, HTTP 애플리케이션 계층에서의 공격 규칙만 지원한다는 한계가 있다[12].

또한, 인공지능을 활용하여 암호화된 악성 트래픽 탐지 및 분류 연구가 수행되고 있다[13]. 초기에는 암호화된 트래픽에 남아있는 평문 정보를 활용하여 핑거프린트(fingerprint)를 구성하고 핑거프린트 매칭을 통해 분류하는 연구가 진행되었으나, 트래픽이 난독화되고 암호화 기술이 발전하면서 적용하기 어려워졌다[14]. 이후 헤더 정보만 이용하거나 통계적 특징을 추출해서 암호화된 트래픽을 기계학습 알고리즘으로 분류하는 연구가 진행되었으나, 해당 방식은 중요한 특징(feature)을 추출할 때 전문가에 의존해야 하며, 일반화가 어렵다는 문제가 있다. 그리고 페이로드 부분을 학습 요소로 활용하기 어려워서 성능을 향상하는 데에 한계가 있다. DPI와 같이 원시 트래픽에서 복잡한 패턴을 학습하는 방식은 암호화된 페이로드에 적용하기 어렵고, 방대한 데이터를 처리할 때 비효율성이 커진다.

전체 트래픽의 90% 이상이 암호화된 트래픽이며 더 이상 종래 네트워크 장비로는 악성 트래픽을 탐지하기 어렵고, 종단간 암호화 방식이 향후 네트워크 표준에 도입되면 탐지 과정이 더욱 어려워질 것으로 예상된다. 이러한 종래 기술의 문제점을 개선하기 위해 본 논문은 차세대 종단간 암호화 표준에 도입될 새로운 패킷 프레임 구조와 프로토콜을 제안한다. 종단간 암호화 상태에서

빠른 패킷 검증을 통해 데이터의 무결성을 보장하고, 알려진 악성 코드를 검출하는 고속 패킷 검사(Fast Packet Inspection, FPI) 방법을 제안하였다. 그리고 암호 트래픽의 페이로드를 학습 요소로 활용하여 악성 트래픽 학습 속도를 개선하고, 정확도를 향상시키는 Advanced FPI 프로토콜을 제안하였다.

본 연구의 주요 기여점은 다음과 같다.

1) 첫째, 차세대 종단간 암호화 표준에 도입될 새로운 패킷 프레임 구조인 FPI와 프로토콜 Advanced FPI를 제안한다.

2) 둘째, 제안하는 패킷 프레임 구조와 프로토콜은 종단간 암호화 환경에서 데이터 무결성과 알려진 악성 코드 검출을 지원하며, 암호화된 트래픽의 페이로드를 학습 요소로 활용할 수 있다.

3) 셋째, 데이터 일부가 변조되면 해시값이 달라져 탐지를 우회할 수 있는 종래 해시 기반 악성 트래픽 탐지 방법의 한계점을 보완한다.

4) 넷째, 종래 방식 대비 탐지 속도와 학습 속도, 탐지 분류 정확도를 개선한다.

본 논문은 다음과 같이 구성된다. Ⅱ장에서는 종단간 암호화, 심층 패킷 검사, 퍼지 해시에 대한 선행연구를 분석하며, Ⅲ장에서는 종단간 암호화 환경에서 빠른 패킷 검증을 지원하는 패킷 프레임 구조와 전송 메커니즘 FPI 그리고 암호화된 악성 트래픽 학습 속도와 정확도 향상을 위한 Advanced FPI 프로토콜을 제안한다. Ⅳ장 실험 환경에서 제안하는 방법과 종래 방법의 실험 조건과 실험 과정에 관해 설명한다. 마지막으로 Ⅴ장에서 FPI와 Advanced FPI의 성능을 평가하고, 실험 결과를 분석한다. 마지막으로 Ⅵ장 결론에서는 논문의 전반적인 내용을 정리한 뒤 연구의 의의와 향후 연구 방향에 대해 기술한다.

Ⅱ. 관련 연구

본 장에서는 연구의 배경이 되는 종단간 암호화 방식의 개념, 한계점과 연구 동향을 파악한다. 악성 암호 트래픽 탐지와 관련된 종래 방법 및 연구를 분석하고, 제안하는 방법론의 기반이 되는 퍼지 해시의 개념을 파악하고, 선행 연구를 분석한다.

2.1 종단간 암호화(End-to-End Encryption, E2EE)

종단간 암호화는 중간에 거쳐 가는 릴레이 서버나 중계기에서 복호화되지 않고 수신지에 도착한 후 송수신지가 공유하는 키에 의해 복호화된다[15]. 대표적인 기술로는 이메일 종단간 암호화 프로토콜인 PGP(Pretty Good Privacy)와 메신저 암호화 프로토콜인 OTR(Off-the-Record)이 사용되고 있으며, 송수신지 간의 안전한 키 교환을 위해 공개키 방식과 비밀키 방식이 결합되어 사용된다[16]. 2014년 카카오톡(kakaotalk) 감청 사건 발생 이후, 메신저 프로그램 보안 강화의 관심과 필요성이 대두되었고, 기밀성과 프라이버시 강화를 위해 종단간 암호화 기능이 필수적으로 내장되고 있다[17]. 그 대표적인 예로는 카카오톡의 1대1 비밀 채팅방과 텔레그램(telegram)의 1대1 채팅방이 있다. 또한 최근 비대면 온라인 서비스 수요의 증가와 함께 글로벌 기업으로 성장한 화상 회의 서비스 기업인 줌(zoom)은 보안 취약점을 보완하기 위해 종단간 암호화 도입을 가속화 중이다[18].

모바일 서비스 환경에서 뿐만이 아니라 사물 인터넷 환경에서도 개인정

보와 민감 정보의 수집을 필요로 하므로, 향후 사물 인터넷 장치에도 종단간 암호화가 필수적으로 사용될 것으로 예측된다.

하지만 암호화된 종단간 환경은 중간에 데이터가 변조되는 경우 사용자가 확인하기 어렵다는 문제가 있으며, 중간 경유지의 정보보호 체계에서 트래픽의 악성 유무를 탐지하기 어렵기 때문에 공격자의 악성 행위를 숨길 수 있는 수단이 된다. 종단간 암호화 환경에서 암호 악성 트래픽을 효과적으로 검사하는 연구는 미진한 실정이다. TABLE 1은 종단간 암호화 환경과 관련된 종래 연구를 정리한 표로, 주로 무결성 검증에 대한 연구가 진행되었다.

TABLE I. Analysis of previous studies on End-to-End Encryption

Previous studies	Ref.	Method	Limitation
Integrity verification	Sam Kumar et al. (2019) [19]	• 사물 인터넷을 위한 다대다 (Many-to-Many) 종단간 암호화 프로토콜 제안 • 트리 기반 브로드캐스트 암호화 기능을 제공하여 다대다 통신을 보장 • 분산된 환경에서 키 인증 및 위임 과정의 무결성을 보장하기 위해	• 종단간 암호화 통신 과정에서 데이터 무결성을 검증하는 부분은 고려하지 않음

	서명 기능을 고려	
Sunyoung Park et al. (2016) [20]	• 중간에 발생하는 위·변조 및 악성 코드를 탐지하기 위해 종래 종단간 암호화 환경에 전자 서명과 검증을 추가	• 검증 단계 이전에 수신 측에서 데이터를 복호화하는 과정을 거치므로, 복호화 과정 중 위협에 노출될 가능성 존재
Jongseok Choi et al. (2018) [21]	• 경량 IoT 통신 프로토콜에 종단간 암호화 기능을 도입하기 위해 새로운 IoT 프레임워크 제안 • 통신하기 전 키 교환 과정에서 서로 통신하는 노드의 속성을 사용하는 암호화 방식인 ABE(Attribute-Based Encryption) 사용	• ABE 암호화 방식을 추가적으로 고려하기 때문에 오버헤드가 생기며, 실시간성을 보장하기 어려움 • 통신 과정에서 발생하는 공격은 고려하였으나, 종단에서 생성된 악성 트래픽은 고려하지 않아서 중계기에서 탐지하기 어려움
	• 도청, 스푸핑 공격 등 4개의 공격에	

대한 보안성 평가하였으며, 타임스탬프의 무결성 보장 가능

사물 인터넷을 위한 종단간 암호화 프로토콜을 제안한 연구[19]에서는 종래 종단간 암호화 환경은 일대일 통신에 초점이 맞추어져 있어서 대규모 산업 IoT 시스템에 적합하지 않기 때문에 트리 기반 브로드캐스트 암호화 기능을 추가하여 사물인터넷 환경에서 다대다 통신이 가능하게 하였다. 이때 분산 환경에서 키 인증과 위임 과정의 무결성을 보장하기 위해 서명 기능을 고려하였다. 리소스 제약이 있는 임베디드 장치에서 실행 가능한 종단간 암호화 프로토콜을 제안하였다는 기여점이 있지만, 종단간 암호화 통신 과정에서 데이터 무결성을 검증하는 부분은 고려하지 않았다는 한계점이 존재한다.

종단간 암호화 환경에서 무결성을 보장하기 위해 [20]은 중간에 발생하는 위·변조 및 악성 코드를 탐지 하고자 종래의 종단간 암호화 환경에 전자 서명과 검증을 추가하였다. 송신자가 데이터 및 파일을 전송하면 해당 파일은 암호화되고, 암호화된 파일은 해시가 적용된 이후 전자 서명 된다. 암호화된 데이터와 전자 서명된 값은 중계 서버를 거쳐서 수신자에게 전달되어, 수신된 데이터를 복호화 후 해시값을 추출하여 전송받은 값과 비교를 수행한다. 두 개의 값이 동일하지 않다면 통신 중간에 메시지가 변조된 것으로 판단하여 통신을 중단한다. 이는 검증 단계 이전에 수신 측에서 데이터를 복호화하는 과정을 거치므로, 복호화 과정 중 위협에 노출될 가능성이 존재한다.

경량 IoT 통신 프로토콜 중 하나인 CoAP(Constrained Application

Protocol)에 종단간 보안 기능을 도입한 연구 [21]에서는 통신하기 전 키 교환 과정에서 서로 통신하는 노드의 속성을 사용하여 암호화 하는 방식인 ABE (Attribute-Based Encryption)를 사용하였다. 악성 브로커(malicious broker), 도청(eavesdropping), 재생 공격(replay attack), 스푸핑 공격(spoofing attack) 총 4가지 공격에 대한 보안성 평가를 진행하였으며, 재생 공격의 경우 공격자가 과거의 타임스탬프 값을 유효한 값으로 변경하더라도 암호화할 때 사용된 인증서와 동일한 인증서를 가진 개체(entity)에 의해서만 변경 가능하기 때문에 암호화된 타임스탬프 값을 수정하기 어려워서 타임스탬프의 무결성을 보장할 수 있다. 하지만, 제안하는 방식은 ABE 암호화 방식을 추가적으로 고려하기 때문에 오버헤드가 생기며, 실시간성을 보장하기 어렵다. 또한, 통신 과정에서 발생하는 공격은 고려하였으나 종단에서 생성된 악성 트래픽은 고려하지 않아서 중계기에서 악성 유무를 탐지하기 어렵다는 한계점이 존재한다.

또한, 인공지능을 활용하여 암호화된 악성 트래픽 탐지 및 분류 연구가 수행되고 있다[13]. 헤더 정보만 이용하거나 통계적 특징을 추출해서 암호화된 트래픽을 기계학습 알고리즘으로 분류하는 연구가 진행되었으나, 해당 방식은 중요한 특징을 추출할 때 전문가에 의존해야 하며, 일반화가 어렵다는 문제가 있다. 그리고 페이로드 부분을 학습 요소로 활용하기 어려워서 탐지 성능을 향상하는 데에 한계가 있다. DPI와 같이 원시 트래픽에서 복잡한 패턴을 학습하는 방식은 암호화된 페이로드에 적용하기 어렵고, 방대한 데이터를 처리할 때 비효율성이 커진다.

2.2 심층 패킷 분석(Deep Packet Inspection, DPI)

DPI는 라우터와 스위치들이 네트워크에서 전송되는 패킷의 헤더뿐만 아니라 데이터의 콘텐츠가 들어 있는 페이로드까지 분석하는 기술이다[22]. OSI(Open System Interconnection) 7계층 전체가 분석 대상에 해당되며 정확한 트래픽 분석이 가능하여 바이러스, 멀웨어 차단, 트래픽 관리, 맞춤 광고 제공 등 다양한 목적으로 사용될 수 있다[23]. 페이로드를 조사하는 가장 기본적인 과정은 페이로드에서 특정 패턴을 신속하게 탐지하는 것으로, 페이로드에 패턴이 존재하는 경우 그 패턴을 검출한다[24]. 그러나 탐지를 위한 패턴을 생성하기 위해 학습 시간을 필요로 하며, 패턴을 비교하기 이전에 페이로드를 복호하여 전체 데이터를 분석하는 과정에서 데이터 처리 속도가 지연되거나 보안 취약점이 발생한다[12]. 이러한 성능 열화와 보안 취약점은 감청, 데이터 유출, 위·변조, DoS(Denial of Service), 자원 소모 공격 등의 보안 피해로 이어지고 있다. 실제로 우리나라의 경우 국정원에서 감청 목적으로 통신 제한 조치 허가서를 발부받아 DPI 방식을 이용하여 감청한 사례가 있다[25].

TABLE 2는 DPI와 관련된 종래 연구를 정리한 표로, DPI는 패킷 페이로드 필터링 과정에서 많은 메모리와 CPU 자원을 소비하기 때문에 이 과정을 개선하기 위한 연구가 활발히 진행되고 있다[26]. 또한, 프라이버시를 보장하기 위해서 전체 트래픽에 DPI를 적용하는 것이 아니라, 트래픽 분류가 어려운 트래픽을 대상으로만 부분적으로 DPI를 활용하는 연구가 진행되고 있다.

TABLE Ⅱ. Analysis of previous studies on Deep Packet Inspection(DPI)

Ref.	Method	Limitation
Mohammad Al-hisnawi et al. (2017) [27]	• Bloom Filter(BF), Quotient Filter(QF), Cuckoo Filter(CF)의 대안으로 QF와 CF의 병합 프로세스인 Quotient-based Cuckoo(QCF)를 DPI에 적용 • 해당 방식은 2개의 해시 함수를 사용하여 시간을 단축시켜 오버헤드를 최소화	• 종단간 암호화 적용 시 패킷 처리 과정에서 복호화 및 암호화 과정이 추가되어 지연 시간이 발생
Kitae Kim et al. (2017) [28]	• Feature selection 기법으로 트래픽을 식별할 수 있는 특징을 추출 한 후에 분류를 진행하고, 도출된 특징에 해당하지 않는 패킷에 대해 DPI를 적용	• 미리 정의한 특징에 대한 분류에만 의존하기 때문에 알 수 없는 패킷에 대한 검증 처리가 어려움
Doround, H. et al.	• 포트 기반 방식과 기계 학습을 결합하여 DPI 트래픽 분류 속도를	• 부분적으로 DPI 방식을 사용하더라도 종래 DPI의 한계점은 그대로 가지고

(2018) [29]	높이고자 함 • 유사한 성능을 제공하는 nDPIng보다 45% 빠른 트래픽 분류 속도 제공 • 분류가 불가능한 패킷인 경우에만 페이로드에서 특징을 추출함으로써 종래 DPI 방법보다 프라이버시 보호 가능	있음
Jong-Won Kim et al. (2019) [30]	• DPI 기술을 활용하여 SDN(Software Define Network) 기반 통합형 보안 스위치 제안	• 소프트웨어 및 하드웨어의 실현 가능성 및 설계 검증이 미흡

DPI의 오버헤드를 최소화하기 위해 [27]는 Bloom Filter(BF), Quotient Filter(QF), Cuckoo Filter(CF)의 대안으로 QF와 CF의 병합 프로세스인 Quotient-based Cuckoo(QCF)를 DPI에 적용하였다. 이 방식은 2개의 해시 함수를 사용하여 CF에 비해 최대 77%, BF와 QF에 비해 최대 98%까지 시간을 단축시켜 계산 오버헤드를 최소화하였다. 하지만, 종단간 암호화 적용 시 패킷 처리 과정에서 복호화 및 암호화 과정이 추가되어 지연시간이 발생하는 한계점이 있다.

프라이버시를 보장하기 위해 특정 조건에서만 DPI 기법을 활용하기도 한다[28]. 이 연구에서는 통계 기반의 Feature selection 기법을 이용하여 트래픽을 식별할 수 있는 특징을 추출한 후 머신러닝 알고리즘을 적용해

빠르고 정확한 분류를 하는 방법을 제안함과 동시에 도출된 특징에 해당하지 않는 패킷에 대해 DPI를 적용하는 방법을 제안하였다. 그러나 이 방법은 미리 정의된 특징에 대한 분류에만 의존하기 때문에 알 수 없는 패킷에 대한 검증 처리가 어렵다는 한계가 존재한다. 또 다른 연구 [29]에서는 일차적으로 포트 기반 방식과 기계 학습 기반 방식을 결합하여 트래픽을 분류하고, 분류가 불가능한 패킷인 경우에만 페이로드에서 특징을 추출함으로써 DPI 트래픽 분류 속도를 높이고자 했다. 유사한 성능을 제공하는 종래 nDPIng 방식 대비 45% 빠른 트래픽 분류 속도를 보장했다. 부분적으로 DPI를 활용하는 방식은 기존의 DPI 방식보다 프라이버시를 보장할 수 있지만, 종래 DPI의 한계점을 그대로 가지고 있다는 문제가 있다.

고속의 네트워크 프로세스를 바탕으로 DPI 기술을 활용하여 방화벽, IPS/IDS(Intrusion Prevention System/Intrusion Detection System) 기능, NAC(Network Admission Control) 기능을 통합한 개념의 SDN(Software Define Network) 기반 통합형 보안 스위치를 제안한 선행 연구가 존재하나 소프트웨어 및 하드웨어의 실현 가능성 및 설계 검증이 미흡하다는 한계가 있다[30].

2.3 퍼지 해시(Fuzzy hash)

퍼지 해시는 컨텍스트 트리거 조각별 해싱(Context Triggered Piecewise Hashing)으로, 변경된 문서에서 부분적으로 파일 매칭을 수행하고자 2006년에 처음 도입된 유사성 분석 도구이다[31,32]. SHA256이나 MD5와 같은 해시 알고리즘은 단방향 암호화 기법으로 복호화가 불가능하며, 같은 입력값에 대해서 고정된 길이의 해시값을 출력한다. 입력의 작은 부분만 변경되어도 출력에 큰 영향을 끼쳐 전혀 다른 해시값을 출력하는데 이러한 눈사태 효과가 해시 알고리즘의 가장 큰 특징이다. 해시 함수는 데이터의 무결성을 보장하는 용도로 주로 활용되고 있다. 하지만, 입력값에 위·변조가 발생하는 경우 다른 해시 결과가 출력되어서 유사도를 판단하기가 어려워진다. 이를 해결하기 위해 퍼지 해시 개념이 도입되었으며, 초기 퍼지 해시는 입력값을 일정 조각으로 나누어 각 조각에 대한 해시를 계산한 이후 결합하여 최종 출력값을 도출한다. 블록 단위를 고정하는 방식은 입력값에 데이터가 추가되거나 삭제되면 블록 단위별 해시값이 전부 바뀌는 결과를 초래한다. 이러한 문제를 해결하기 위해 롤링 해시 개념을 추가한 컨텍스트 트리거 조각별 해싱이 제안되었다. 슬라이딩 윈도우가 입력으로 전달되면 그 값을 바탕으로 생성된 롤링 해시값이 블록 크기와 일치할 때 입력으로 들어온 값에 대해서 해시를 계산하게 된다. 즉, 입력값에 대해서 길이가 다른 조각이 생성되고 조각별로 해시를 계산하여 최종값을 도출하는 방식이다[31]. 따라서 입력값에 새로운 데이터가 추가되면 일부 조각에 대한 해시만 영향을 받아서 유사도를 측정할 수 있게 된다. 퍼지 해시 알고리즘으로 TLSH, ssdeep, sdhash 등과 같은 다양한 알고리즘이 존재하며, 그 중 ssdeep은 바이러스 토탈에서 악성 프로그램 탐지 및 분석을 위한 용도로 사용되고 있다[32].

TABLE Ⅲ은 퍼지 해시와 관련된 선행연구를 정리한 표이다. 주로 퍼지 해시는 트래픽 보다는 파일의 이상을 감지하기 위해 주로 연구되어 왔다[32, 33]. 또한, 퍼지 해시는 사이즈 제한으로 크기가 작으면 파일의 이상 징후를 감지하기 어렵기 때문에 딥러닝을 도입하여 유사도 탐지 성능을 개선하는 연구도 활발히 진행되었다[32, 33].

TABLE Ⅲ. Analysis of previous studies on Fuzzy hash

Ref.	Method	Limitation
Jesse Kornblum (2006) [31]	• 컨텍스트 트리거 조각별 해싱(Context Triggered Piecewise Hashing) 기법을 처음으로 제안	• 전체 파일 크기에 비해 상대적으로 작은 경우 파일 이상을 감지하기 어려움
Frieder Uhling (2022) [32]	• 딥러닝 근사 매칭을 통해 15% 미만의 변칙 크기에도 TLSH 및 ssdeep 알고리즘이 파일의 상관 관계를 탐지하도록 정확도 향상	• js, pdf, xlsx 등 다양한 파일 유형에 대해서 제안하는 방법론을 적용하여 평가했으나, 트래픽을 대상으로 진행하지는 않음
Thomas Gobel (2022) [33]	• 다양한 퍼지 해시 알고리즘에 대해서 자동으로 테스트하는 프레임워크 제안 • 퍼지 해시 알고리즘별 장단점 파악 가능	• html, pdf, text, doc 등 다양한 파일 형식에 대해서 퍼지 해시 알고리즘을 비교 분석 했으나, 트래픽에 대해서는 어떤 알고리즘이 효과적인지 분석된 바가 없음

S. Hiruta (2015) [34]	• 악성 프로그램 변종을 탐지하기 위해서 정적 분석 과정에서 퍼지 해시를 적용 • 퍼지 해시를 멀웨어 바이너리에 적용하고, 멀웨어 바이너리 유사성을 기반으로 멀웨어 분류	• 일부 악성 프로그램을 대상으로 진행하고, 분류가 불가능한 악성 프로그램이 존재

Ⅲ. 악성 암호 트래픽의 고속 탐지 기법

본 장에서는 종단간 암호화 환경에서 효율적으로 트래픽을 검사할 수 있는 차세대 네트워크 프레임 구조로 FPI 패킷 프레임과 Advanced FPI 프로토콜을 제안한다.

3.1 FPI(Fast Packet Inspection)

본 절에서는 종단간 암호화 환경에서 효율적으로 패킷을 검사할 수 있는 FPI 기법과 차세대 네트워크 프레임 구조를 제안한다.

1) 구조 및 동작 원리

Fig. 1은 제안하는 종단간 암호화 FPI 시스템 구조를 보여준다. FPI는 송신(sender) 노드, 중계 서버/정보 보호 체계(Relay Server including Information Security System, RS/ISS) 노드, 수신(receiver) 노드로 구성된다. 종단(end) 노드는 하나의 장치에서 데이터 전송과 수신이 가능한 트랜시버(transceiver)이며, Fig. 1은 시스템 구조를 단순화하여 제안하는 방식의 구조와 메커니즘을 명확히 설명하고자 송신 노드와 수신 노드로 나누어 표현하였다. 여기서 송신 노드는 데이터를 전송하는 노드를 의미하고, 수신 노드는 데이터를 수신하는 노드를 의미하며 통신 장치 간에 서로 송신 노드와 수신 노드가 전환되며 통신하는 상황을 가정한다. RS/ISS는 알려진 악성코드에 대한 시그니처를 해시 기반으로 생성하고, 악성 해시 리스트와 비교한 후 일치하는 경우에 차단하여 위협을 방지하는 블랙리스

트 기반으로 동작한다.

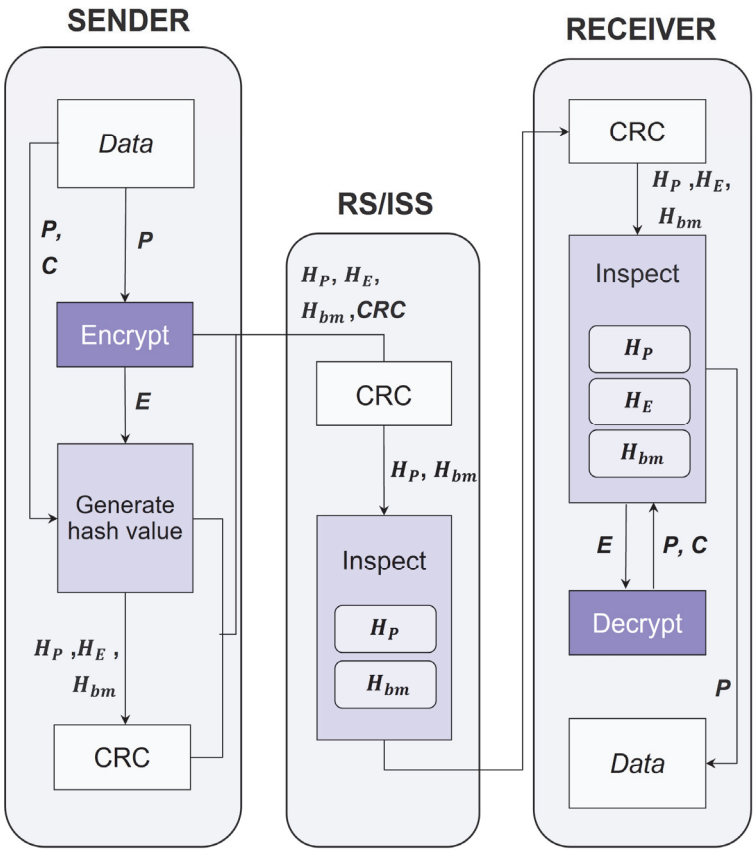

FIGURE 1 System architecture

송신 노드는 패킷을 생성하여 전송하고, RS/ISS 노드는 CRC 유효성을 확인한 뒤, 악성 해시 리스트 비교한 후 릴레이하며, 수신 노드는 데이터 수신 과정을 통해 유효성과 비교 검증을 수행한다. 이때 CRC는 채널 잡음이나 충돌로 인한 오류를 감지하기 위해 디지털 네트워크에서 일반적으로 사용되는 오류 감지 코드이다. 본 논문에서는 수신된 해시 맵과 알려진 악성 해시리스트를 비교하는 데에 처리되는 지연 시간을 줄이기 위해 CRC를 사용한다. CRC를 사용하여 해시 맵의 오류를 확인하지 않으면, 채널

노이즈로 인해 발생하는 패킷에 대해서 모두 해시 맵을 비교해야 하므로 지연 시간이 증가하게 된다. 반면, CRC를 사용하는 경우 수신노드가 잘못된 해시 맵을 빠르게 폐기하기 때문에 지연 시간을 줄일 수 있다. 마지막으로 수신 노드는 데이터 수신 과정을 통해 패킷의 유효성을 확인하고 비교 검사를 하도록 설계된다.

송신측은 데이터 페이로드의 평문(H_P)을 해시한 값, 암호화된 데이터 페이로드(H_E)를 해시한 값, 페이로드 중 특정 컴포넌트(C)를 해시 비트맵 형태로 추가한 H_{bm}을 생성하는 단계와, 이를 이용하여 CRC 코드를 산출하는 단계로 구성된다. RS/ISS는 CRC를 이용한 무결성 검사 단계와 수신한 해시값과 악성 해시 리스트를 비교하여 악성 유무를 판단하는 검증단계로 구성된다. 수신측은 중계서버/정보보호체계(RS/ISS)와 동일하게 무결성 검사 단계와 검증단계를 포함하고, 복호화를 수행하여 평문에 대한 검증을 수행하는 단계로 구성된다.

H_{bm}, H_P H_E, CRC 정보 필드를 이용해 패킷의 신뢰성과 보안성을 검사하여 사전에 등록된 악성 패킷을 차단할 수 있다. 이와 같이 FPI는 직접 패킷의 페이로드를 확인하지 않고, 헤더에 삽입된 비가역적인 해시 정보로 패킷을 분석하기 때문에 종래 기술 대비 사용자의 프라이버시를 보호할 수 있으며, 안정적인 전송 속도와 보안성을 보장할 수 있다.

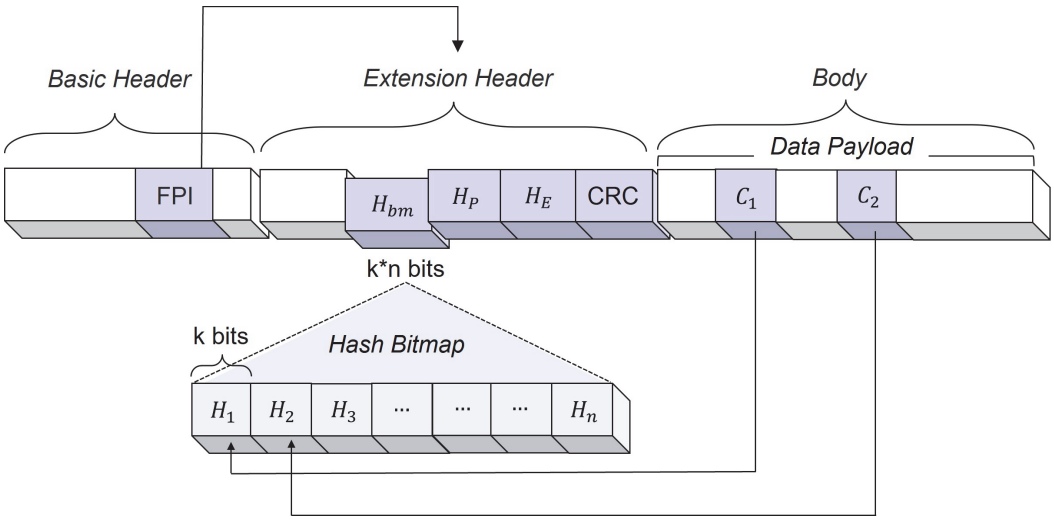

FIGURE 2 Fast Packet Inspection(FPI) frame structure

FPI의 프레임구조는 Fig. 2와 같이 구성된다. FPI는 기존 패킷과 호환 가능한 확장된 패킷 구조로 기존의 패킷 프레임 헤더를 그대로 사용하며, 확장된 새로운 헤더에 주요 컴포넌트의 해시 값과 오류검사코드를 삽입하여 악성행위를 탐지할 수 있다. CRC 코드를 동시에 삽입함으로써 채널 노이즈에 의한 확장 헤더의 요소 오류를 검출할 수 있다. FPI는 기존의 프레임 헤더인 basic header에 새로운 프레임 포맷인지를 확인하기 위한 FPI 필드가 추가된다. basic header는 기존의 장치와 호환성을 유지하기 위한 헤더이고, FPI 필드는 reserved 필드를 활용한다. FPI 필드를 통해 새로운 extension header의 유무를 확인 가능 하며, extension header에는 H_{bm}, H_P H_E, CRC 4가지 필드가 추가된다. 각 필드에 대한 자세한 설명은 다음과 같다.

H_P 필드에는 평문 데이터 페이로드를 해싱한 값 H_P가 추가되고, H_E 필드에는 암호화된 데이터 페이로드를 해싱한 값 H_E가 추가된다. H_P, H_E 필드를 통해 데이터 페이로드의 무결성을 검증할 수 있다.

H_{bm} 필드에는 페이로드에서 선정된 컴포넌트 C_1~C_n의 해시 값이 비트맵으로 추가된다. 본 논문에서는 네트워크 계층의 IP 프로토콜을 위한 identification, fragmentation offset과 애플리케이션 계층의 HTTP 프로토콜을 위한 URL과 referer를 주요 컴포넌트로 선정하였으나, source address 등 악성 트래픽 패턴으로 활용될 수 있는 정보들을 추가로 정의하여 활용할 수 있다.

IP 프로토콜에서 identification 필드는 데이터그램이 단편화되어 전송된 후 재조립 시 이용되며 fragment offset은 단편화되기 전 데이터 시작점으로부터의 차이를 나타내고, 전체 데이터그램에서 단편 일부분에 포함된 데이터의 시작 위치를 나타낸다. 첫 번째 조각이 너무 작아서 전체 전송 헤더를 포함할 수 없는 경우, 방화벽이나 수신기를 혼동시키는 DoS공격인 경우가 존재하며 이는 identification, fragment offset을 이용하여 탐지가 가능하다[35].

HTTP 프로토콜에서 referer 헤더 필드는 클라이언트가 서버로 요청을 보낼 때 해당 요청이 어떤 source URL에서 참조된 것인지를 알려주는 기능을 제공한다[36]. HTTP referer 필드는 악성 웹사이트를 은폐하는 과정에서 사용되기 때문에[36], referer 헤더 필드의 도메인이 악성 사이트 블랙리스트에 존재하는지를 확인하여, 악성 사이트를 탐지할 수 있다.

CRC 필드에는 H_P와 H_E 그리고 H_{bm}의 유효성 체크를 통해 무결성 검증을 하는 CRC 값이 삽입된다. CRC는 주로 통신에서 메시지 전송 중 노이즈에 의한 데이터의 에러 발생 여부를 탐지하기 위한 기법으로, 전송 데이터 뒤에 32bit로 생성된 순환 중복 검사 비트를 붙여 전송한다. 송수신지에 의해 합의된 디바이더를 이용하여 송신지는 연산을 통해 CRC 비트를 생성하고 데이터에 붙여 전송을 수행한다. 데이터가 수신되면 수신 측은 디바이더를 통해 동일한 연산을 수행하고, 결과값에 의해 오류 발생 여부

를 판단할 수 있다.

2) 전송 메커니즘

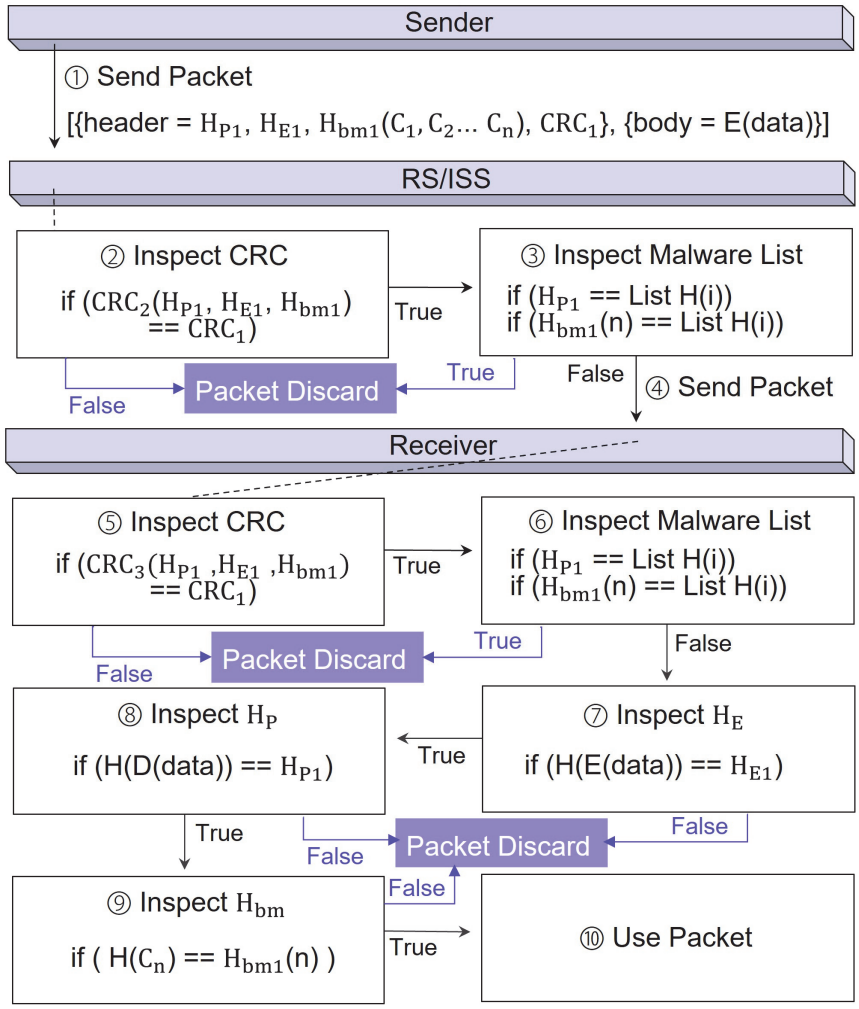

FIGURE 3 FPI flow chart

Fig. 3은 FPI 플로우 차트를 보여준다. FPI의 패킷 전송 단계는 다음과 같다.

송신 노드는 데이터 페이로드 중 미리 정해진 주요 검사 컴포넌트의 해시 값을 구해 해시맵(H_{bm})을 만들고 데이터 페이로드를 암호화하기 전과 후의 해시값(H_P, H_E)을 생성한다. 그리고 CRC를 생성한 다음 헤더의 해당 필드에 삽입하여 패킷을 생성한다. 이후 패킷을 RS/ISS 노드로 전송한다.

RS/ISS 노드는 송신 노드로부터 받은 패킷의 H_P와 H_E, 그리고 H_{bm}을 이용하여 CRC 값을 다시 생성하고, 송신 노드로부터 받은 패킷의 CRC 값과 비교한다. 다시 생성된 CRC 값과 송신 노드로부터 받은 CRC 값이 일치하지 않으면 즉시 폐기(discard)한다. CRC의 유효성이 확인되면, 악성 해시리스트와 비교하며, 악성 해시리스트와 동일한 경우 차단하고, 동일하지 않은 경우 수신 노드로 전송한다.

수신 노드는 RS/ISS 노드로부터 받은 패킷의 H_P와 H_E, 그리고 H_{bm}을 이용하여 CRC 값을 생성하고, RS/ISS 노드로부터 받은 CRC 값과 비교한다. 다시 생성된 CRC 값과 RS/ISS 노드로부터 받은 CRC 값이 일치하지 않으면 폐기하고, 일치하는 경우에는 악성 해시 리스트와의 비교를 수행한다. 동일한 경우 폐기하고, 동일하지 않은 경우 암호화된 데이터 페이로드의 해시 값과 RS/ISS 노드로부터 받은 H_E의 일치 여부를 확인한다. 일치하는 경우 데이터 페이로드를 복원하여 해시 값을 구하고, RS/ISS 노드로부터 받은 H_P의 일치 여부를 확인한다. 이후 복호화된 페이로드에서 주요 컴포넌트 C 값과 전달받은 H_{bm}의 요소의 동일한지를 확인한 뒤 동일한 경우에는 데이터 페이로드를 이용할 수 있다.

3.2 Advanced FPI

본 절에서는 종단간 암호화 환경에서 효율적으로 패킷을 검사하는 FPI 기법의 확장 연구로 Advanced FPI 프로토콜을 제안한다. Advanced FPI는 제어 프레임(control frame)이며, 퍼지 해시를 도입하여 데이터의 일부가 변조되면 해시값이 달라져 탐지를 우회할 수 있는 종래 해시 기반 악성 트래픽 탐지 방법의 한계점을 보완한다. 또한, 프라이버시를 보장하면서 암호화된 페이로드를 인공지능 학습 요소로 활용할 수 있다.

1) 구조 및 동작 원리

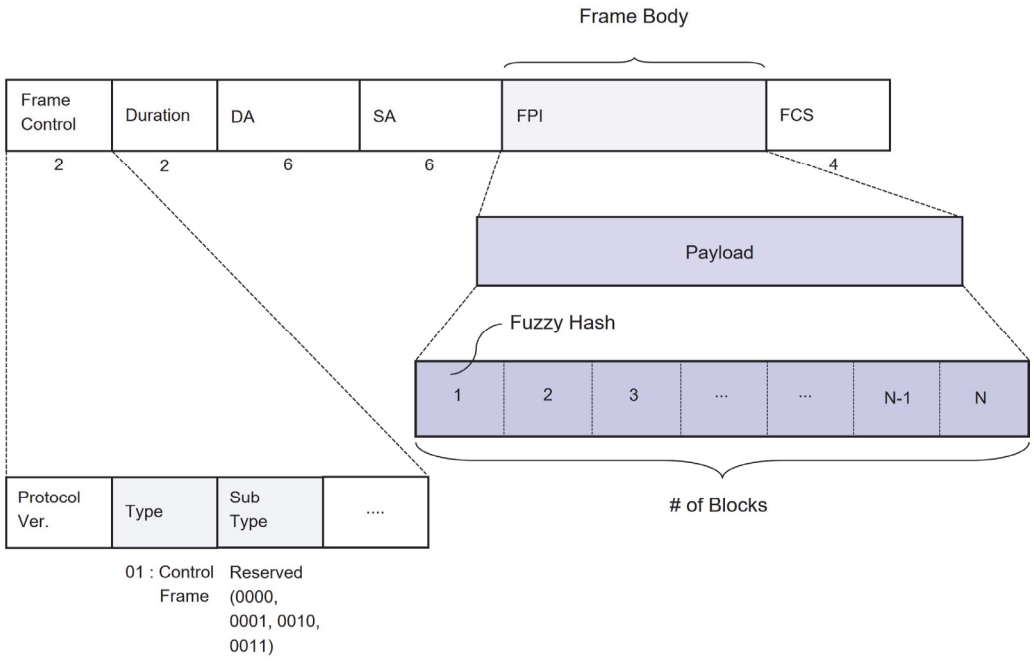

FIGURE 4 Advanced FPI control frame format

Fig. 4는 Advanced FPI의 프레임 구조를 나타낸다. Advanced FPI는

암호화된 데이터 프레임의 전송을 돕는 데 사용되는 제어 프레임으로 frame control, duration, DA(Destination Address), SA(Source Address) 필드를 포함하는 MAC 헤더와 데이터 프레임의 페이로드를 퍼지 해시한 값이 담겨있는 FPI 프레임 바디 그리고 FCS(Frame Check Sequence) 총 세 부분으로 구성된다. MAC 헤더의 경우 Frame control 필드의 type 필드는 01이며, sub type 필드는 0000, 0001, 0010, 0011과 같이 reserved를 FPI 필드로 활용할 수 있다. FPI 프레임 바디는 Advanced FPI 제어 프레임 다음에 전송하는 데이터 프레임의 페이로드 부분을 퍼지 해시한 값이 추가된다. 페이로드를 일련의 블록으로 분할하고 각 블록에 대해서 해시값을 산출하기 때문에 유사도를 측정하여 변조된 악성 트래픽을 탐지할 수 있다. FCS는 에러 검출을 위한 필드로, 수신측에서 계산한 FCS와 전송되어 도착한 프레임의 FCS를 비교하여 에러 여부를 확인한다.

2) 전송 메커니즘

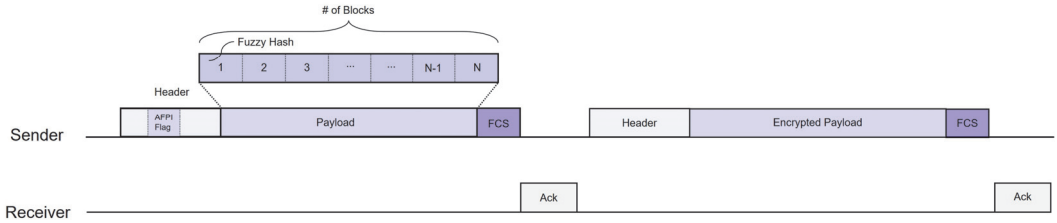

FIGURE 5 Advanced FPI protocol procedure

Fig. 5는 Advanced FPI 프로토콜 절차를 나타내며, 동작 순서는 다음과 같다. 송신단은 데이터 프레임을 전송하기 전에 FPI 제어 프레임을 전송한다. 이때 퍼지 해시된 페이로드 값, 즉 FPI 프레임 바디가 알려진 악

성 트래픽과 일정 수준 유사하다면 제어 프레임과 전송하려고 했던 데이터 프레임을 폐기한다. 그리고 송신단은 수신단으로부터 전송된 제어 프레임에 대한 응답확인을 받으면, 데이터 프레임을 암호화하여 수신단에 전송한다. 암호화된 데이터 프레임은 중계서버/정보보호체계에서 복호화되지 않기 때문에 프라이버시를 보장할 수 있고, 데이터 일부가 누락되거나 변조되더라도 악성 트래픽에 대한 유사도를 측정할 수 있어서 탐지 논리를 우회하기 어려워진다.

Ⅳ. 실험 환경

4.1 FPI 모델링

1) 실험 환경

제안하는 FPI와 종래의 방식인 DPI의 패킷 검사 속도를 비교 평가하기 위해 패킷 검증 시뮬레이션을 수행하였다. python을 통해 종단간 암호화 네트워크 환경에서 FPI와 DPI를 모델링하여 패킷 전송 시뮬레이션을 통해 패킷 검증 속도를 비교하였다.

본 실험의 가정 조건은 다음과 같다. 제안 모델 FPI와 비교 모델 DPI는 모두 사전에 키 교환이 이루어진 상태로 가정하여 이후 패킷 전송 및 검증 단계를 구현하였다. FPI의 경우 H_{bm}, H_{E}, H_{P}, CRC 값을 extension 헤더에 추가하는 방식으로 동작하고, 비교 모델 DPI의 경우에는 RS/ISS에서 암호화된 페이로드 데이터를 복호화하여 페이로드를 분석한 뒤 다시 암호화 과정을 거쳐 수신 단에 전송하는 방식으로 동작한다. 또한, DPI의 경우 두 노드 간 네트워크 통신 과정에서 거치는 모든 라우터에서 작동하지 않고, 수신 단의 최근접 라우터에서만 검증 작업을 진행하는 것으로 가정하였다. 이때, 제안 모델과 비교 모델의 데이터 페이로드 암·복호화 과정은 128비트의 CBC(Cipher Block Chainng)모드 AES(Advanced Encryption Standard) 암호화 알고리즘으로 가정하였다.

본 실험의 위협 조건은 다음과 같다. 중간자 공격, 도청 및 개인 데이터 유출이 발생할 수 있다. E2EE 암호화를 적용하지 않으면 RS/ISS 과정에서 패킷을 복호화하기 때문에 프라이버시가 침해될 가능성이 있다. 또한,

정보를 전송하는 과정에서 데이터 유출 및 도청 위험이 있고, 공격자는 패킷을 변조하여 정상 패킷으로 위장할 수 있다.

2) 비교모델 및 실험 과정

FPI와 DPI 실험 노드는 송신 노드, RS/ISS 노드, 수신 노드로 구성된다.

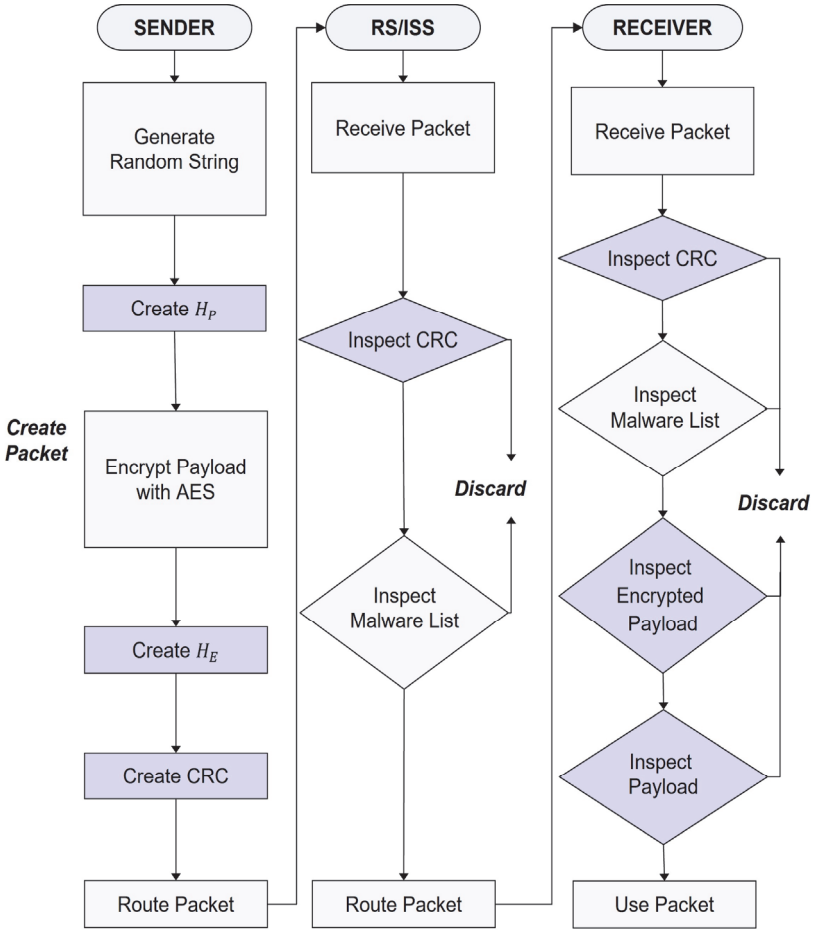

FIGURE 6 Packet transmission process for FPI

FPI의 패킷 전송 및 검증 단계는 Fig. 6과 같다.

송신 노드에서 정해진 길이의 랜덤한 데이터 페이로드를 생성한 다음 페이로드의 데이터 중 특정 컴포넌트 값을 선정하여 SHA(Secure Hash Algorithm) 256으로 해싱하여 H_{bm}에 삽입한다. 특정 컴포넌트는 페이로드를 10바이트씩 분할하여 사용하였다. 예를 들어 첫 번째 컴포넌트는 페이로드의 1번째 바이트부터 10번째 바이트, 두 번째 컴포넌트는 11번째 바이트부터 20번째 바이트로 가정한다. 생성한 데이터 페이로드는 AES 암호화 하여 데이터 페이로드 필드에 추가한다. H_P는 데이터 페이로드를 SHA 256으로 해싱하고, H_E는 해당 데이터 페이로드를 암호화한 값을 SHA 256으로 해싱 하여 해당 필드에 삽입한다. H_P와 H_E 그리고 H_{bm}으로 부터 CRC 값을 생성하여 CRC 필드에 삽입한 다음 패킷을 RS/ISS 노드로 전송한다. RS/ISS 노드는 CRC 유효성을 확인하고 악성 해시 리스트와 비교, 검증하여 패킷을 필터링 하도록 한다. 이때, 악성 해시 리스트와 값이 같거나 CRC값이 유효하지 않는 경우에는 필터에서 폐기한다. 악성 해시 리스트는 정해진 길이의 데이터를 10개씩 생성하여, 악성 해시로 지정해 비교하였다. 수신 노드는 CRC 유효성을 확인하고 악성 해시 리스트와 비교, 검증한 후, 페이로드 내 RS/ISS 노드로부터 받은 H_E와 특정 컴포넌트의 해시값에 대한 H_{bm} 그리고 AES 암호화 된 데이터 페이로드 해시값의 일치 여부를 확인한다. 만약 3가지 필드의 해시값이 일치하면 데이터 페이로드를 이용하도록 구현하였다.

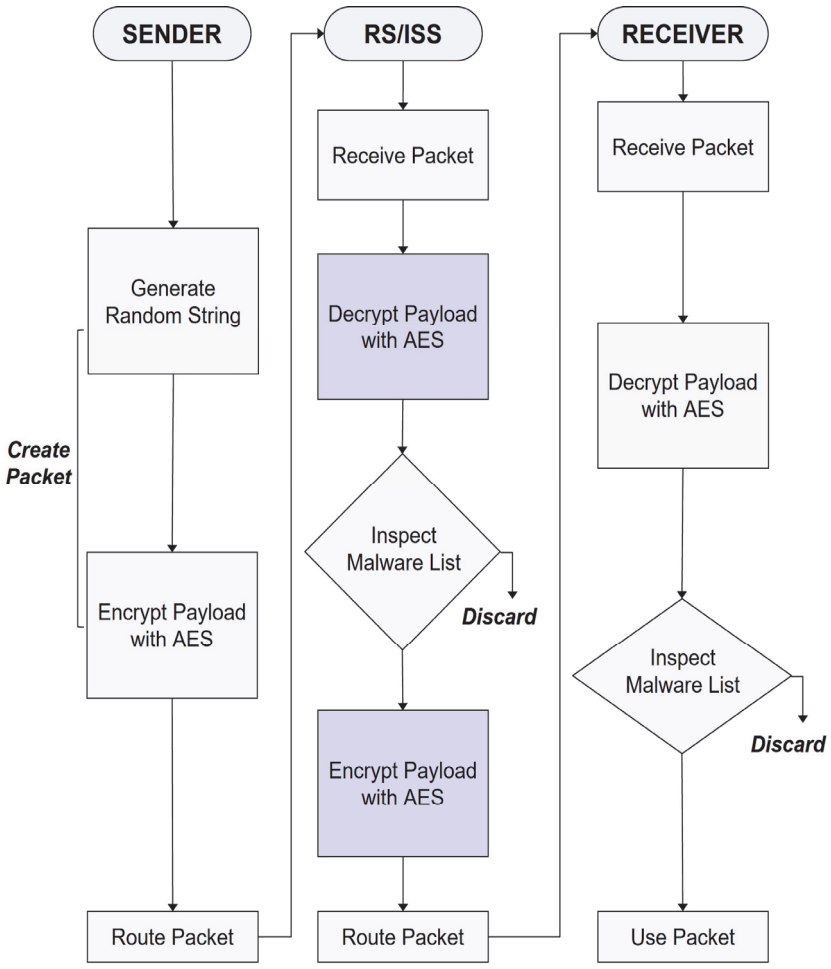

FIGURE 7 Packet transmission process for DPI

DPI의 패킷 전송 및 검증 단계는 Fig. 7과 같다.

송신 노드에서 정해진 길이의 랜덤한 데이터 페이로드를 생성한 뒤, AES 암호화한 값을 데이터 페이로드 필드에 지정하고 패킷을 RS/ISS 노드로 전송한다. RS/ISS 노드에서는 수신한 패킷을 AES 복호화하여 악성 리스트와 패킷의 데이터 페이로드를 비교한다. 일치할 경우 폐기하며, 일치하지 않는 경우 다시 암호화 과정을 거쳐 패킷을 수신 노드로 전송한다.

수신 노드 역시 수신한 패킷을 복호화 하여 악성리스트와 RS/ISS 노드로부터 받은 패킷의 데이터 페이로드를 비교한다. 동일한 경우에 폐기하고, 동일하지 않은 경우에 데이터 페이로드를 이용하도록 구현하였다. FPI와 DPI의 전송 속도는 송신 노드가 패킷을 생성하고 전송하여 패킷 검증을 거친 후, 수신 노드가 페이로드를 사용하기까지의 패킷 전송 및 검증 단계에 소요되는 시간을 비교하여 평가하였다.

4.2 Advanced FPI 모델링

1) 실험 환경

제안하는 Advanced FPI와 종래 암호 악성 트래픽을 분류하는 방식의 정확도(accuracy)와 학습 시간(training time)을 비교 평가하기 위해 다음과 같은 환경에서 실험을 진행하였다. Intel(R) core(TM) i7-10750H CPU 프로세서와 32GB 램인 windows 11 환경에서 anaconda 3, jupyter notebook, python 3.9 프로그래밍 도구를 사용하였다.

악성 암호 트래픽 분류하기 위해 UNSW-NB15 데이터셋[37]을 활용하였다. UNSW-NB15 데이터셋은 사이버 보안 센터(ACCS)의 연구소에서 IXIA PerfectStorm 도구로 작성되었으며, 일반 트래픽과 총 9개의 공격 트래픽으로 구성된다. 공격 유형은 fuzzers, analysis, backdoors, exploits, reconnaissance, shellcode, DoS, worms, generic이며, 총 49개의 특징을 포함하고 있다.

패킷의 페이로드를 학습 요소로 사용하기 위해서 UNSW-NB15 데이터셋의 PCAP 파일에서 원시 패킷 데이터를 추출하여 라벨링을 한 데이터셋

[38]으로 실험을 진행하였다. 해당 데이터셋의 구성은 TABLE Ⅳ와 같다. 패킷의 헤더는 ttl, total_len, protocol, t_delta 값이며, 페이로드는 1500바이트가 1바이트씩 분할되어 총 1500개의 특징으로 이루어져 있다.

TABLE Ⅳ. Reprocessed UNSW-NB15 dataset [38]

No.	Feature Name	Type	Description
1~1500	payload_byte_#	int64	• 페이로드를 1바이트씩 분할하여 총 1500개의 특징으로 구성 • 16진수의 문자열을 0~255 정수로 변환 • 최대 길이는 1500 바이트로, 1500 바이트 미만의 경우 zero padding
1501	ttl	int64	• 송신단에서 수신단까지의 Time to Live 값
1502	total_len	int64	• 전체 패킷의 길이
1503	protocol	object	• 프로토콜 종류
1504	t_delta	int64	• 패킷간 시간 차이
1505	label	object	• 정상트래픽과 9 종류의 악성 트래픽으로 총 10개의 label

2) 비교 모델 및 실험 과정

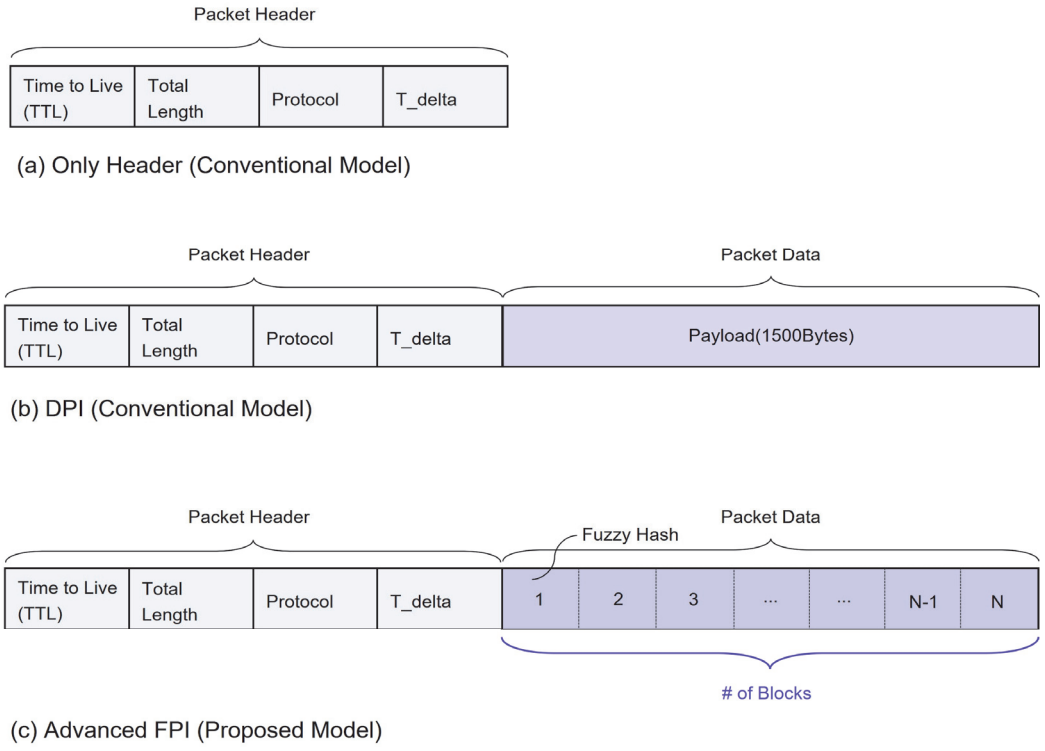

FIGURE 8 Feature vector and data utilized for training the ML model

비교 모델은 종래 암호 악성 트래픽을 분류하는 방식으로, 헤더정보의 학습 결과를 바탕으로 분류하는 방식과 DPI 방식을 선정하였다. Fig. 8은 머신러닝 모델을 훈련하기 위해 사용된 데이터와 특징 벡터를 나타낸다.

Fig. 8의 (a)처럼 헤더 정보의 학습 결과를 바탕으로 악성 트래픽을 분류하는 방식의 경우, 암호화된 페이로드는 학습 요소로 사용하기 어려우므로 ttl, total_len, protocol, t_delta 총 4가지 정보만 학습하도록 데이터 전처리를 진행하였다. Fig. 8의 (b)인 DPI는 암호화가 되어있지 않다는 전제 하에 페이로드를 모두 학습하도록 했으며, 사용한 특징은 총 1504개

로 TABLE 5와 같다. Fig. 8의 (c)에 나와 있듯이 Advanced FPI는 퍼지 해시한 데이터 페이로드를 학습 요소로 활용하였는데, 전처리 작업은 Fig. 9와 같이 진행하였다. 먼저, 1바이트씩 분할된 페이로드를 하나로 병합하고, 제로 패딩 부분을 제거한다. 이후 퍼지 해시를 진행하는데, 퍼지 해시 도구로 ssdeep을 사용하였다. 세 번째로 퍼지 해시된 페이로드를 1바이트씩 분할하여 정수 인코딩(integer encoding)을 진행한다. 퍼지 해시된 페이로드의 길이는 다양하기 때문에 제일 긴 페이로드 길이를 기준으로 제로 패딩을 추가했으며, 이후 4가지 헤더 요소를 추가하여 학습을 진행한다.

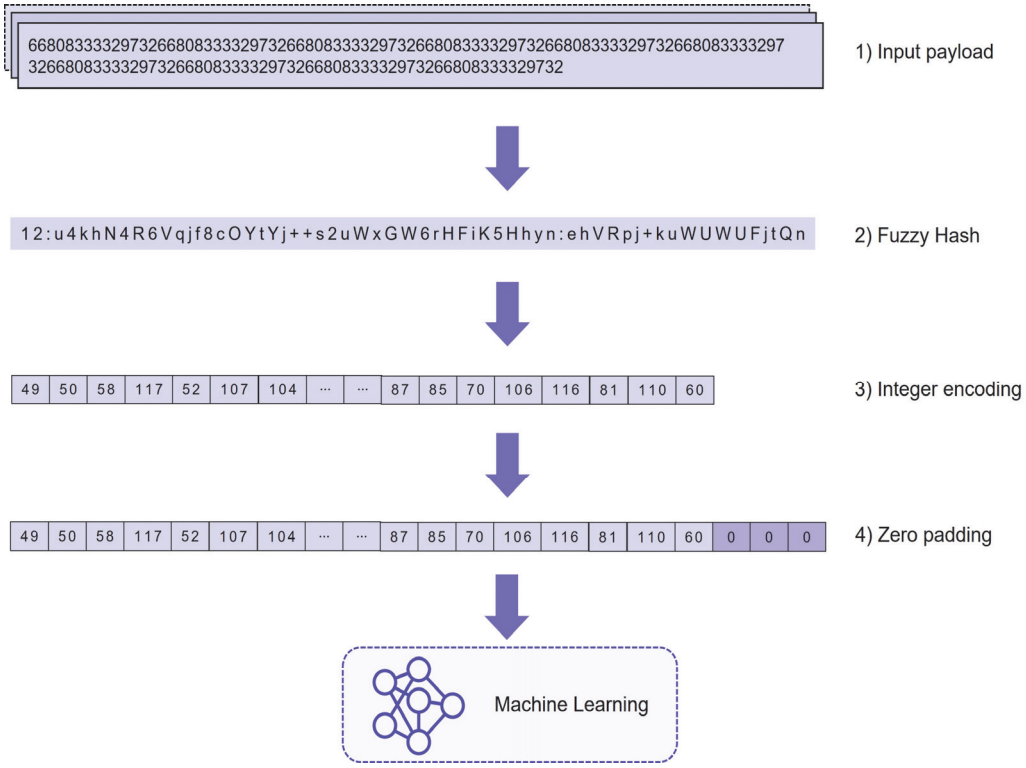

FIGURE 9 Payload data preprocessing for Advanced FPI

실험은 종래 연구[38]에서 사용한 모델 중 기계학습 모델인 선형 회귀(Linear Regression) 모델과 랜덤 포레스트(Random Forest) 그리고 의사 결정 트리(Decision Tree) 모델을 대상으로 진행하였다. 학습된 모델을 비교 분석하기 위한 평가 지표는 정확도(accuracy), 정밀도(precision), F1 점수(F1 score), 재현율(recall)로, 백분율로 환산하여 사용하였다. 추가적으로 에너지 효율을 확인하기 위해 학습 시간(training time)을 측정하였다.

Ⅴ. 성능 평가

4.1 FPI 실험 결과 및 분석

1) 패킷 전송 속도 비교 분석

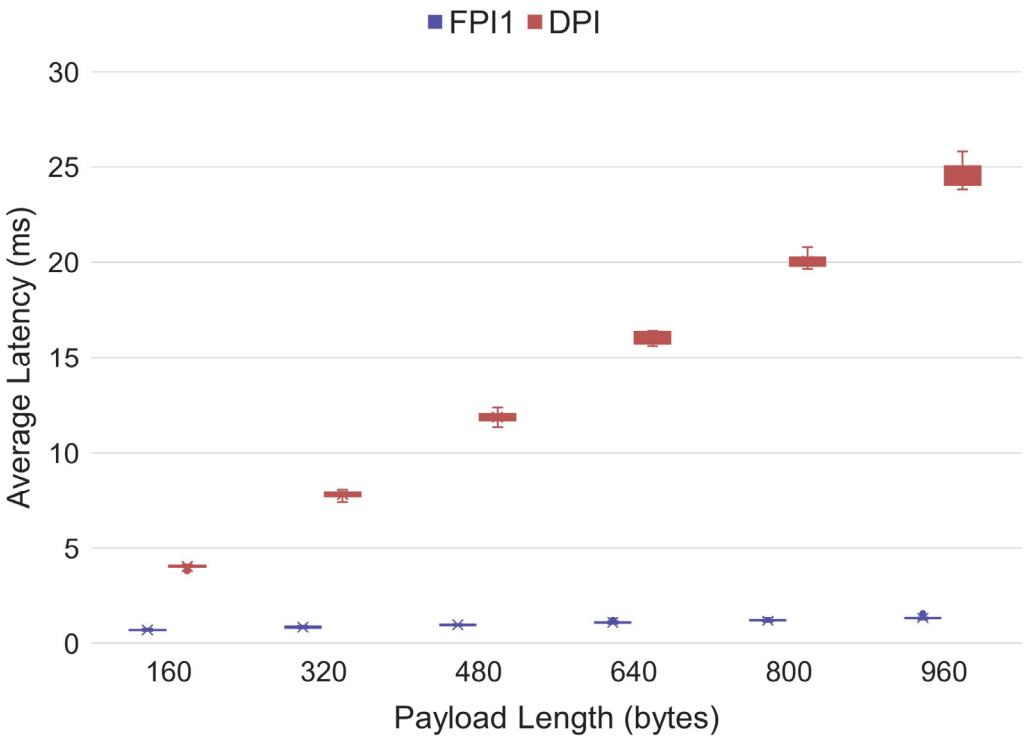

FIGURE 10 Comparison of packet transmission times of FPI and DPI

Fig. 10은 제안 모델인 FPI와 비교 모델인 DPI의 페이로드 크기에 따른 패킷 전송 시간을 비교한 그래프이다. 그래프의 X축은 송신 노드가 전송한 패킷 페이로드의 길이이며, Y축은 전송 시간의 평균을 의미한다. 한 라운드는 송신 노드에서 페이로드를 160바이트부터 960바이트까지 160바이

트 단위로 전송하여 수신 노드가 페이로드를 평문으로 복호화하여 최종 검증을 마칠 때까지를 의미하며, 페이로드의 크기별로 10,000라운드씩 진행하여 전송 시간의 평균값을 산출하였다.

실험 결과, DPI 대비 FPI의 전송 속도가 전반적으로 빠르게 측정되었다. 페이로드의 크기가 160bytes 일 때, FPI는 약 0.661ms(milli-second), DPI는 약 4ms로 DPI의 패킷 전송 및 검증 과정에서 FPI의 6배에 달하는 지연이 있음을 확인하였고, 페이로드의 크기가 960bytes일 때, FPI는 약 1,271ms, DPI는 약 24.052ms로 DPI의 패킷 전송 및 검증 과정에서 FPI의 18.9배에 해당하는 지연이 있음을 확인하였다.

따라서 FPI는 DPI보다 패킷 전송 속도가 빠르며, 페이로드의 크기가 커질수록 전송 속도의 차이가 선형적으로 증가하는 것을 확인할 수 있다. 시뮬레이션 결과를 통해 FPI는 패킷을 암호화된 상태에서 검증하여 보안성을 유지하는 동시에 RS/ISS에서 암·복호화 과정을 거쳐 데이터 페이로드를 직접 비교하는 DPI 방식보다 안정적이고 속도 측면에서 효율적인 메커니즘임을 보여준다.

2) FPI 컴포넌트 개수에 따른 패킷 전송 속도 비교 분석

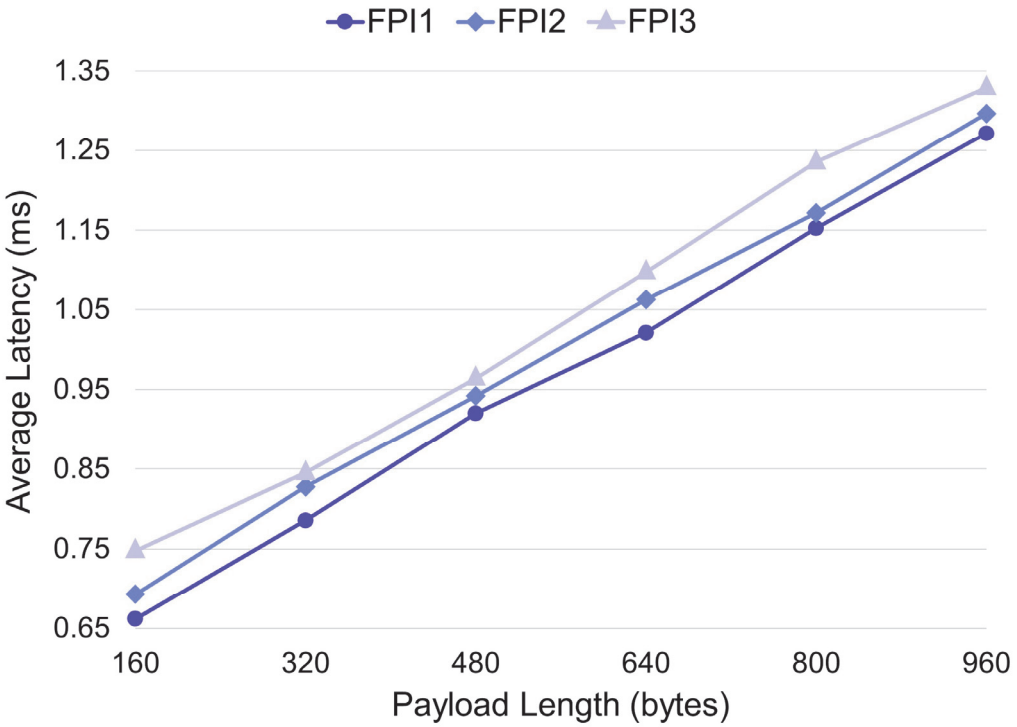

FIGURE 11 Packet transmission time comparison of FPI according to the number of components

Fig. 11은 제안 모델 FPI의 컴포넌트 개수에 따른 전송 시간을 비교한 그래프이다. 그래프의 X축은 송신 노드가 보낸 패킷 페이로드의 길이이며, Y축은 페이로드 크기에 따른 전송 시간의 평균을 의미한다. 한 라운드는 송신 노드에서 페이로드를 160바이트부터 960바이트까지 160바이트 단위로 전송하여 수신 노드가 페이로드를 평문으로 복호화하여 최종 검증을 마칠 때까지를 의미하며, 페이로드의 크기별로 10,000라운드씩 진행하여 전송 시간의 평균값을 산출하였다. FPI 1은 H_P와 H_E가 추가된 패킷, FPI 2는 H_P, H_E와 H_{bm} 요소가 1개 추가된 패킷, FPI 3은 H_P, H_E와 H_{bm}요소

가 2개 추가된 패킷을 의미한다. H_P는 페이로드, H_E는 암호화된 페이로드를 SHA256으로 해시한 값으로 크기는 32 바이트이다. H_{bm}의 컴포넌트는 페이로드를 10바이트로 분할한 조각으로 가정하였다. 따라서 FPI 1은 64 바이트, FPI 2는 96바이트 FPI 3에는 128바이트를 extension header에 추가하여 시뮬레이션을 진행하였다.

실험 결과, H_{bm} 요소의 개수가 추가됨에 따라 비례적으로 전송 시간이 지연되는 것을 확인할 수 있다. 하지만 그 차이는 최대 0.064ms, 최소 0.018ms로 매우 근소하게 측정되었다. 또한, Fig 11과 비교하였을 때, 800bytes 페이로드를 기준으로 extension header에 290개의 H_{bm} 요소를 추가하여도 DPI에서 같은 길이의 패킷을 검증한 것보다 짧은 시간이 소요되는 것을 확인하였다. 시뮬레이션 결과, H_{bm} 요소를 추가하는 것은 보안성을 향상시키는 것으로 볼 수 있기 때문에 FPI는 전송 지연 대비 높은 보안성을 제공할 수 있는 메커니즘임을 확인했다. 향후 extension header를 확장하여 다양한 요소를 빠른 시간 안에 검사할 수 있을 것으로 기대한다.

TABLE V. Comparison of DPI and FPI

Category	DPI	FPI		
Inspection coverage	100%	10%	20%	100%
Transmission speed average (ms)	15.74	1.06	1.09	2.88
E2EE retention	X	O		
Complexity of establishing environment	Complex [39]	Simple		

TABLE V는 4가지 평가 기준을 바탕으로 FPI와 DPI를 비교하여 정리한 표이다. 검사 커버리지는 전체 페이로드 중 검사에 사용되는 컴포넌트의 비중을 말한다. 페이로드 길이가 640바이트인 경우를 기준으로 검사 커버리지와 전송 속도를 평가하였다. FPI의 컴포넌트가 1개일 경우 검사 커버리지는 10%이며, 컴포넌트가 2개일 경우 20%, 전체 페이로드를 검사할 경우에는 100%에 해당한다. 검사 커버리지에 따라 전송속도를 비교하면, 검사 커버리지가 10%인 경우 DPI 대비 약 15.4배의 차이를 보였으며, 20%인 경우 약 14.4배, 100%인 경우 약 5.3배의 차이를 보였다. 검사 커버리지가 커질수록 전송 속도는 지연되지만 전체 페이로드를 검사하는 경우에도 DPI보다 빠른 속도임을 알 수 있다. 이를 통해 FPI의 메커니즘이 DPI 메커니즘보다 전송 속도 측면에서 더 효율적임을 확인하였다.

FPI는 암호화된 페이로드를 복호하지 않고 비가역적인 해시 값과 해시 맵 (H_P, H_E, H_{bm})을 extended header에 정보 필드를 실어 전송함으로써 종단간 암호화를 유지한 채 악성행위를 탐지하고 차단한다. 하지만 DPI의 경우, 페이로드를 복호화 후 악성행위를 탐지하기 때문에 종단간 암호화가 유지되지 않는다는 문제점이 발생한다. DPI는 페이로드를 평문으로 복호화하기 때문에 보안 취약점이 존재하는 RS/ISS에서 DPI가 실행된다면, 페이로드의 데이터가 노출될 수 있다.

그리고 환경 구축 복잡도 측면에서 비교하면, DPI는 패킷 페이로드 기반 분석 방법으로 실제 데이터를 중심으로 검증하기 때문에 정보 추출 과정이 어렵고, 페이로드를 저장하는 과정에서 드는 유지 및 관리 비용이 많이 들며 많은 프로세싱 자원을 필요로 하므로 제한적으로 사용된다[39]. 또한, 어떠한 하드웨어 플랫폼 및 매칭 알고리즘을 선택하느냐에 따라 DPI 엔진의 처리량과 그 비용의 차이가 크며, 패턴 매칭 유닛 구조에 따라 확장성과 유연성 등의 요소가 고려된다[40]. 반면, FPI는 기존의 네트

워크 장비를 프로토콜 및 기능에 따라 변경할 필요 없이 본래의 패킷 프레임 구조에서 헤더 부분만 확장하여 사용하고, 연산 과정 역시 해시값으로 간소화되었기 때문에 장비 교체 및 데이터 유지, 관리에 대한 비용이 크지 않다. 따라서 FPI가 전반적인 환경 구축에 대해 비용 효율적임을 확인할 수 있다.

결론적으로 종단간 암호화에서 주로 사용되는 AES를 암호화 알고리즘으로 사용한 시뮬레이터에서 FPI가 DPI보다 전반적으로 빠른 전송 속도를 보였다. 이는 DPI가 RS/ISS에서 복호화, 검증, 재 암호화, 전송 과정을 거치는 것 보다 FPI가 더 효율적으로 작동함을 의미한다. 또한 FPI의 보안성을 의미하는 H_{bm} 요소를 충분히 추가할 수 있어 전송 속도와 보안성 측면에서 FPI는 DPI보다 효과적인 패킷 검증 기술임을 확인할 수 있다.

4.2 Advanced FPI 실험 결과 및 분석

1) 학습 시간 비교 분석

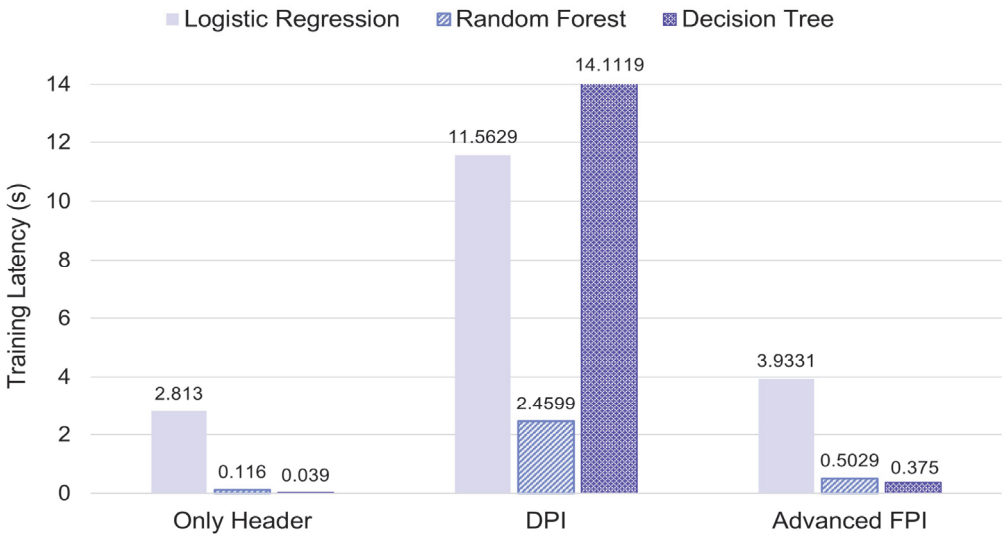

FIGURE 12 Comparison of training latency of conventional models and proposed model

Fig. 12는 머신러닝 모델에 따른 종래 방식 2가지와 제안하는 Advanced FPI의 학습 시간을 보여준다. 파이썬의 time 모듈의 time 함수를 사용하여 측정하였다. 실험 결과 머신러닝 모델별로 학습 시간에 차이가 있지만, 전체적으로 DPI가 가장 시간이 오래 걸렸으며 헤더 정보만 학습하는 종래 방식이 제일 적은 시간이 소요되었다. 제안하는 방식은 의사 결정 트리 모델에서 DPI 방식 보다 학습 시간을 최대 13.74초 단축하였고, 랜덤 포레스트 모델에서는 1.957초, 선형 회귀 모델에서는 7.63초 개선했다. only header 방식보다 최소 0.336초에서 최대 1.1181초 학습시간이 증가했으나, 이는 페이로드 부분을 학습시켰기 때문에 시간이 더 소요된 것으로 파

악된다. 페이로드 부분을 학습하는 종래 DPI 방식보다 학습시간을 단축시켰으며, 프라이버시를 보장하면서도 암호화된 페이로드를 학습요소로 활용할 수 있기 때문에 종래 방식 대비 효율적이다.

2) 정확도 성능 비교 분석

Fig. 13은 머신러닝 모델에 따른 2가지 종래방식과 제안하는 방식의 트래픽 분류 정확도 성능을 보여준다. 실험 결과에 따르면 제안하는 Advanced FPI가 전체적으로 분류 정확도 성능이 우수하다. 종래 DPI 방식보다 최소 1.89%에서 최대 20.01% 정확도 성능을 개선했으며, only header 방식 대비 최소 24.87%에서 최대 35.08% 성능이 향상되었다. 모델 성능 평가에 따른 자세한 결과는 TABLE VI에서 확인할 수 있다. 특히 의사 결정 트리 모델에서 성능이 가장 우수한 것으로 파악하였다. 결론적으로 퍼지 해시를 적용한 페이로드를 학습 요소로 활용하더라도 분류 성능 정확도가 높아지는 것을 볼 수 있다.

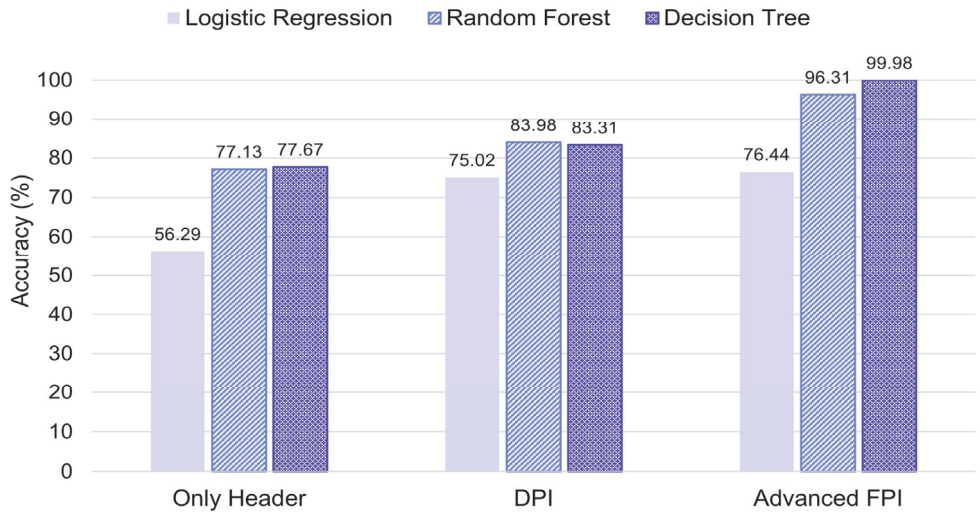

FIGURE 13 Comparison of accuracy of conventional models and proposed model

TABLE VI. Detail result for performance comparison

Model	Performance metric	Only Header	DPI	Advanced FPI
Logistic Regression	Accuracy	56.29%	75.02%	76.44%
	Precision	25.05%	56.49%	41.91%
	Recall	25.21%	52.59%	42.32%
	F1 Score	22.35%	53.36%	41.95%
Random Forest	Accuracy	77.13%	83.98%	96.31%
	Precision	52.23%	66.28%	94.02%
	Recall	50.33%	64.17%	90.81%
	F1 Score	50.73%	64.96%	92.32%
Decision Tree	Accuracy	77.67%	83.31%	99.98%
	Precision	54.75%	65.56%	99.91%
	Recall	51.82%	64.62%	99.92%
	F1 Score	52.28%	63.90%	99.91%

Ⅵ. 결론 및 향후 연구

네트워크를 통한 데이터 전송 과정에서 감청, 개인 정보 유출, 프라이버시 침해, 악성 코드와 같은 보안 이슈가 끊임없이 발생하고 있다. 최근에는 이러한 보안 위협에 대응하기 위해 종단간 암호화 기능이 웹 · 네트워크 애플리케이션의 필수 요소가 되고 있다. 본 논문에서는 종단간 암호화 상태에서 데이터 무결성과 알려진 악성코드 검출을 지원하는 패킷 프레임 구조 및 전송 메커니즘인 FPI를 제안하였다. FPI는 DPI의 주요 약점인 지연 문제를 해결하는데 중점을 둔다. 제안한 FPI 방법과 종래의 대표적인 DPI 방법을 검사 속도와 검증 커버리지를 비교 평가하기 위해 모델링하여 패킷 검증 시뮬레이션을 수행하였다. RS/ISS에서 발생하는 암호화, 검증, 복호화, 재 암호화, 전송 과정으로 인해 FPI가 DPI와 비교하여 종 단간 암호화를 유지하면서 빠른 검사 속도를 보였다. 그리고 네트워크 환경과 요구조건에 따라 검사 대상 컴포넌트의 범위를 조정하여 민첩한(agile) 환경 적응이 가능하다.

결과적으로 본 논문에서 제안하는 FPI는 보안성, 속도 성능, 유연한 확장성 요구 조건을 동시에 만족되어야 하는 네트워크 환경에 적합한 패킷 검사 방식이라고 할 수 있으며, 향후 차세대 네트워크 보안 기술로 활용될 수 있을 것이다. 기본적으로 FPI는 DPI에 비해 정확도가 떨어지지만, 해시맵을 많이 사용할수록 탐지 정확도가 높아져 처리 지연시간과 정보 유출 위험이 높아진다. 또한, FPI는 악성 노드가 블랙리스트에 기반한 동일한 해시 탐지 기술을 사용하여 동일한 해시 탐지 논리를 쉽게 우회할 수 있다는 한계가 있다.

FPI의 확장 연구로 Advanced FPI 프로토콜을 제안하였다. Advanced

FPI는 제어 프레임(control frame)으로, 퍼지 해시를 도입하여 데이터의 일부가 변조되면 해시값이 달라져 탐지를 우회할 수 있는 종래 해시 기반 악성 트래픽 탐지 방법의 한계점을 보완할 수 있다. 또한, 트래픽 검증을 위해 별도의 복호화 과정이 필요하지 않기 때문에 프라이버시를 보장하면서 페이로드를 인공지능 학습 요소로 활용할 수 있다. 실험 결과 DPI 방식 대비 학습 시간을 최소 1.957초에서 최대 13.74초까지 단축시키면서도 분류 정확도 성능을 최소 1.89%에서 최대 20.01%까지 향상시켰다.

향후 작업에서는 정보유출을 고려하여 적정 수준의 해시 블록 사이즈를 선택하고, FPI 해시맵과 Advanced FPI의 퍼지해시 블록사이즈에 따른 오버헤드 비교 분석을 통해 adaptive 체계를 구현하고 평가할 계획이다.

참 고 문 헌

[1] Wang, D., Cheng, H., He, D., & Wang, P., “On the Challenges in Designing Identity-Based Privacy-Preserving Authentication Schemes for Mobile Devices”, IEEE Systems Journal, vol. 12, no. 1, pp. 916-925, 2018.

[2] DMC MEDIA, “2019 Mobile Messenager App Usage Behavior”, DMC REPORT, Apr 2019.

[3] Tuul Triyason, Anuchart Tassanaviboon, and Prasert Kanthamanon, “Hybrid Classroom: Designing for the New Normal after COVID-19 Pandemic”, 11th International Conference on Advances in Information Technology (IAIT2020), Association for Computing Machinery, no. 30, pp. 1-8, 2020.

[4] Endeley, R.E., "End-to-End Encryption in Messaging Services and National Security—Case of WhatsApp Messenger", Journal of Information Security, 9, pp. 95-99, Jan 2018.

[5] P. R□sler, C. Mainka and J. Schwenk, "More is Less: On the End-to-End Security of Group Chats in Signal, WhatsApp, and Threema", IEEE European Symposium on Security and Privacy (EuroS&P), pp. 415-429, Apr 2018.

[6] K. Cohn-Gordon, C. Cremers, B. Dowling, L. Garratt and D. Stebila, "A Formal Security Analysis of the Signal Messaging Protocol", IEEE European Symposium on Security and Privacy (EuroS&P), pp. 451-466, Apr 2017.

[7] Karbasi, Amir Hassani, and Siyamak Shahpasand, "A post-quantum end-to-end encryption over smart contract-based blockchain for defeating man-in-the-middle and interception attacks", Peer-to-Peer

Networking and Applications, pp. 1-19, 2020.

[8] W. Wang, M. Zhu, J. Wang, X. Zeng and Z. Yang, "End-to-end encrypted traffic classification with one- dimensional convolution neural networks", IEEE International Conference on Intelligence and Security Informatics(ISI), pp. 43-48, 2017.

[9] Pedro Amaral, Jo□o Dinis, Paulo Pinto, Luis Bernardo, Jo□o Tavares, Henrique S. Mamede, "Machine learning in software defined networks: Data collection and traffic classification", IEEE 24th International Conference on Network Protocols (ICNP), pp. 1-5, Nov 2016.

[10] Hao Ren, Hongwei Li, Dongxiao Liu, Guowen Xu, Nan Cheng, Xuemin Sherman Shen, "Privacy- preserving Efficient Verifiable Deep Packet Inspection for Cloud-assisted Middlebox", IEEE Transactions on Cloud Computing, Nov 2020.

[11] J. Garcia, T. Korhonen, R. Andersson and F. V□stlund, "Towards Video Flow Classification at a Million Encrypted Flows Per Second", IEEE 32nd International Conference on Advanced Information Networking and Applications (AINA), pp. 358-365, 2018.

[12] Sherry, Justine and Lan, Chang and Popa, Raluca Ada and Ratnasamy, Sylvia, "Blindbox: Deep packet inspection over encrypted traffic", Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication, pp. 213-226, 2015.

[13] Xinjie Lin, Gang Xiong, Gaopeng Gou, Zhen Li, Junzheng Shi, and Jing Yu. "ET-BERT: A Contextualized Datagram Representation with Pre-training Transformers for Encrypted Traffic Classification", In□Proceedings of the ACM Web Conference 2022 (WWW '22), 2022.

[14] Thijs van Ede, Riccardo Bortolameotti, "Flow-Print Semi-supervised Mobile-App Fingerprinting on Encrypted Network Traffic", 2020.

[15] M. Nabeel, "The Many Faces of End-to-End Encryption and Their

Security Analysis", IEEE International Conference on Edge Computing (EDGE), pp. 252-259, 2017.

[16] Shirvanian, Maliheh, Nitesh Saxena, and Jesvin James George, "On the pitfalls of end-to-end encrypted communications: A study of remote key-fingerprint verification", Proceedings of the 33rd Annual Computer Security Applications Conference, pp. 499-511, Dec 2017.

[17] Espinoza, Antonio M., et al. "Alice and bob, who the FOCI are they?: Analysis of end-to-end encryption in the LINE messaging application", 7th USENIX Workshop on Free and Open Communications on the Internet (FOCI 17), 2017.

[18] Blum, Josh, et al. "E2E Encryption for Zoom Meetings.", Zoom Video Communications. Inc, 2020.

[19] Sam Kumar, Yuncong Hu, Michael P Andersen, Raluca Ada Popa, and David E. Culler, "JEDI: Many-to-Many End-to-End Encryption and Key Delegation for IoT", USENIX Security Symposium, pp. 1519-1536, Aug 2019.

[20] Sunyoung Park, Sangmin Park, Uiseong Park, Hyungon Kim, "Messenger Program with End-to-end Encryption and Digital Signature", The Jouranl of Korean Institute of Communications and Information Sciences, pp. 305-306, Nov 2016.

[21] Jongseok Choi, Youngjin In, Changjun Park, Seonhee Seok, Hwajeong Seo, and Howon Kim. 2018. "Secure IoT framework and 2D architecture for End-To-End security", J. Supercomput. 74, 8 pp. 3521-3535, 2018.

[22] C. Xu, S. Chen, J. Su, S. M. Yiu and L. C. K. Hui, "A Survey on Regular Expression Matching for Deep Packet Inspection: Applications, Algorithms, and Hardware Platforms", IEEE Communications Surveys & Tutorials, vol. 18, no. 4, pp. 2991-3029, 2016.

[23] V. S. Elagin, B. S. Goldshtein, A. V. Onufrienko, A. A. Zarubin and A.

A. Savelieva, "The efficiency of the DPI system for identifying traffic and providing the quality of OTT services", Systems of Signals Generating and Processing in the Field of on Board Communications. IEEE, pp. 1-5, 2018.

[24] Fang Yu, R. H. Katz and T. V. Lakshman, "Gigabit rate packet pattern-matching using TCAM", Proceedings of the 12th IEEE International Conference on Network Protocols, pp. 174-183, 2004.

[25] Jin Woo Jung, "Communication equipment related information leakage risk analysis and preparation", National Assembly Intellignece Committee Research Service Report, 2019.

[26] Mohammad Al-hisnawi, Mahmood Ahmadi, "Deep Packet Inspection Using Quotient Filter", IEEE Communications Letters, vol. 20, no. 11, pp. 2217-2220, Nov 2016.

[27] Mohammad Al-hisnawi, Mahmood Ahmadi, "QCF for deep packet inspection", IET Networks, vol. 7, no. 5, pp. 346-352, Mar 2018.

[28] Kitae Kim , Choongseon Hong, "Machine Learning and Feature Based Traffic Classification in Software Defined Network Environment", The Korean Institute of Information Scientists and Engineers, 44(1), pp. 1253-1255, 2017.

[29] H. Doroud et al., "Speeding-Up DPI Traffic Classification with Chaining," 2018 IEEE Global Communications Conference (GLOBECOM), pp. 1-6, 2018.

[30] Jong-Won Kim, Jeong-In Choi, "Software-Defined Networking(SDN) Based Integrated Security Switch Design", The Institute of Electronics and Information Engineers, pp. 552-553, 2019.

[31] Jesse Kornblum, Identifying almost identical files using context triggered piecewise hashing, Digital Investigation, Volume 3, Supplement, pp.91-97, 2006.

[32] Uhlig, Frieder, et al. "Transformer-Boosted Anomaly Detection with Fuzzy Hashes." arXiv preprint arXiv:2208.11367, 2022.

[33] Thomas G□bel, Frieder Uhlig, Harald Baier, Frank Breitinger, “A framework for automated evaluation of similarity hashing”, Forensic Science International: Digital Investigation, 2022.

[34] Shohei, H., Yukiko, Y., Hajime, S., & Hiroki, T. “Evaluation on Malware Classification by Combining Traffic Analysis and Fuzzy Hashing of Malware Binary”, 2015.

[35] John, Wolfgang & Olovsson, Tomas, “Detection of malicious traffic on back-bone links via packet header analysis”, Campus-Wide Information Systems, vol. 25, no. 5, pp. 342-358, 2008.

[36] M. Mansoori, Y. Hirose, I. Welch and K. R. Choo, "Empirical Analysis of Impact of HTTP Referer on Malicious Website Behaviour and Delivery", IEEE 30th International Conference on Advanced Information Networking and Applications (AINA), pp. 941-948, 2016.

[37] N. Moustafa and J. Slay, “UNSW-NB15: a comprehensive dataset for network intrusion detection systems (UNSW-NB15 network dataset)”, Proc. IEEE Mil. Commun. Inf. Syst. Conf. (MilCIS), pp. 1-6, Nov. 2015.

[38] Farrukh, Yasir Ali; Khan, Irfan; Wali, Syed; Bierbrauer, David; Pavlik, John; Bastian, Nathaniel (2022): Payload-Byte: A Tool for Extracting and Labeling Packet Capture Files of Modern Network Intrusion Detection Datasets. TechRxiv. Preprint. https://doi.org/10.36227/techrxiv.20714221.v2

[39] Chao Gong and K. Sarac, "IP traceback based on packet marking and logging", IEEE International Conference on Communications, Vol. 2 ,pp. 1043-1047, 2005.

[40] Po-Ching Lin, Ying-Dar Lin, Tsern-Huei Lee and YuanCheng Lai, "Using String Matching for Deep Packet Inspection", IEEE Computers, vol.

41, no. 4, pp. 23-28, April 2008.

[41] So-Yeon Kim, Sun-Woo Yun, Eun-Young Lee, So-Hyeon Bae, and Il-Gu Lee, "Fast Packet Inspection for End-To-End Encryption", Electronics9, no. 11, 2020.

ABSTRACT

Fast Detection Techniques of Malicious Encrypted Traffic for End-to-End Encryption

So-Yeon Kim

Department of Future Convergence Technology Engineering
Graduate School of Sungshin University

With the recent development and popularization of various network technologies, communicating with people at any time, and from any location, using high-speed internet, has become easily accessible. At the same time, eavesdropping, data interception, personal data leakage, and distribution of malware during the information transfer process have become easier than ever. Recently, to respond to such threats, end-to-end encryption(E2EE) technology has been widely implemented in commercial network services as a popular information security system. However, with the use of E2EE technology, it is difficult to check whether an encrypted packet is malicious in an information security system. A number of studies have been previously conducted on deep packet inspection(DPI) through trustable information security systems. However, the E2EE is not maintained when conducting a DPI, which requires a long inspection time.

Thus, in this study, a fast packet inspection (FPI) and its frame structure for quickly detecting known malware patterns while maintaining E2EE are proposed. In this study, an advanced fast packet inspection (Advanced FPI) protocol that utilizes the payload of encrypted traffic as a feature to improve the training speed of malicious traffic and improve accuracy is proposed.

Based on the simulation results, the proposed FPI allows for inspecting packets approximately 14.4 and 5.3 times faster, respectively, when the inspection coverage is 20% and 100%, as compared with a DPI method under a simulation environment in which the payload length is set to 640 bytes. Advanced FPI improves training time by at least 7.63s in the linear regression model and up to 13.73s in the decision tree model compared to deep packet inspection. In terms of accuracy performance, it improves the accuracy performance of malicious traffic classification by up to 35.08% from a minimum of 1.89% compared to the conventional method.

ACKNOWLEDGEMENTS

본 논문은 MDPI electronics에 게재된 ‘Fast Packet Inspection for End-to-End Encryption[41]’ 논문을 바탕으로 확장하여 후속 연구를 수행한 연구 결과입니다. 본 논문을 지도해주신 이일구 교수님과 FPI 연구에 도움을 준 공저자 윤선우, 이은영, 배소현 학생에게 감사드립니다.

또한, Advanced FPI 연구에 기여해 준 이연지, 이진민 학생에게 감사드립니다.