https://doi.org/10.7840/kics.2022.47.9.1370

TLS기반 패킷 검사 우회 표준 기술 분석 및 미래 전술네트워크 운용 가능성 연구

이 웅 희•, 허 준 범°

A Study on TLS-Based Packet Inspection and Its Circumvention Technologies in Future Tactical Network

Woonghee Lee•, Junbeom Hur°

요 약

본 논문에서는 TLS기반 패킷 검사 기법을 소개하고 관련 우회 기술의 동향을 소개한다. 먼저 패킷 검사 및 차단기법인 IP 차단, DNS 블로킹, SNI 필터링을 소개하고 그에 대응하는 DNS 암호화 표준 기술인 DNS-over-TLS와 DNS-over-HTTPS을 소개한다. 더하여 DoT와 DoH의 장단점과 세부적인 특징들을 분석하고 관련 프라이버시 이슈를 설명한다. 또한 새로 표준화된 DoQ를 소개하면서 DoT, DoH, DoQ의 세 표준 기술을 비교 분석한다. 더하여 SNI 암호화 기술의 표준화 진행 상황을 소개하고 그 한계점에 대해 논한다. 마지막으로 관련 표준 기술의 미래전술네트워크 운용 가능성을 살펴본다.

Key Words : Packet Inspection, Circumvention, SNI, TLS, DNS encryption, Future tactical network

ABSTRACT

This paper introduces TLS-based packet inspection techniques and introduces trends in related circumvetion technologies. First, this paper describes IP blocking, DNS blocking, and SNI filtering, which are representative packet inspection techniques and DNS-over-TLS and DNS-over-HTTPS, which are DNS encryption standard technologies to prevent those techniques. In addition, the advantages and disadvantages of DoT and its detailed characteristics are analyzed and related privacy issues are explained. We also analyze the three technologies, DoT, DoH and DoQ by comparing their characteristics with previous researches. We introduce the standardization progress of SNI encryption technology and discuss about its limitation. Additionally we look at the possibility of future tactical network operations of related standard technologies.

Ⅰ. 서 론

클라이언트와 서버 어플리케이션과의 암호화 통신을 위한 프로토콜 TLS(Transport Layer Security, 이하 TLS)가 발표된 이후 다양한 분야에 암호화 통신 기법으로 TLS가 도입됐고 여러 취약점과 한계점으로 발견되면서 TLS 또한 계속 갱신되어 1.3버전까지 발표된 상태이다[1].

※ 본 연구는 방위사업청과 국방과학연구소가 지원하는 미래 전투체계 네트워크기술 특화연구센터 사업의 일환으로 수행되었습니다 (UD190033ED).
• First Author : Department of Computer Science and Engineering, Korea University, binugoon@korea.ac.kr, 학생회원
° Corresponding Author : Department of Computer Science and Engineering, Korea University, jbhur@korea.ac.kr, 종신회원
논문번호 : 202205-088-0-SE, Received April 7, 2022; Revised June 2, 2022; Accepted June 29, 2022

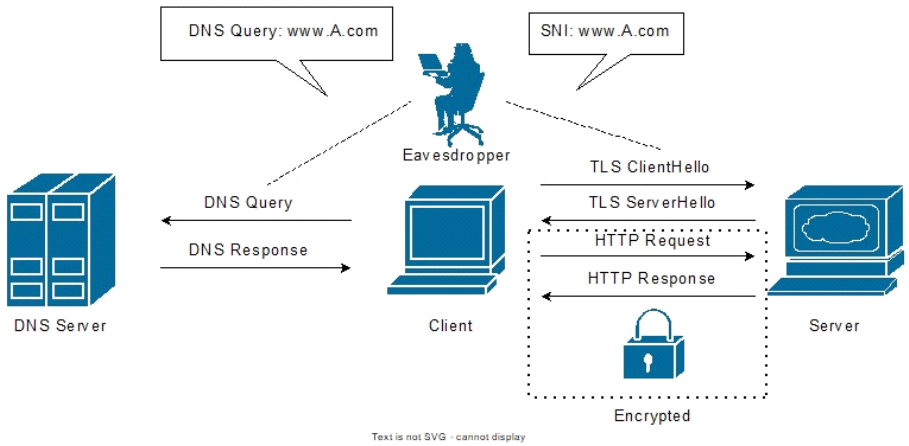


그림 1. 일반적인 DNS통신과 HTTPS 통신
Fig. 1. General DNS and HTTTPS Communication

TLS가 보급되기 전엔 주로 두 가지 방법으로 인터넷 환경상에서 불법적인 웹사이트를 검사할 수 있었다. 첫째로, HTTP 통신에서 보이는 평문을 통해 보는 방법이 있었다. HTTP 통신에서는 클라이언트가 URL을 통해 하나의 웹사이트에서 어느 경로에 있는 페이지를 요청해서 서버에게 응답을 받는다. 이 과정에서 드러나는 URL은 네트워크상에 있는 모든 관찰자가 볼 수 있다. 다른 방법으로는 차단하고자 하는 사이트의 IP주소를 차단하는 방법도 있다.

TLS가 빠르게 보급되면서 불법유해매체물에 대한 패킷 검사는 TLS의 한계점을 이용하기 시작했다. 첫째로 Domain Name System(이하 DNS) 통신을 활용하는 방법이다[2]. 현재 DNS 통신의 대부분은 포트번호 53번으로 UDP 기반의 통신을 하는데 평문으로 모든 내용을 주고받는다. 클라이언트는 웹사이트의 IP주소를 알아내기 위해 도메인 이름을 도메인 서버에게 질의하는데, 이 과정에서 드러나는 도메인 이름의 평문을 검사하는 것이다. 이를 DNS 스푸핑이라고 하는데, 접속을 차단하고자 하는 사이트로 접속하려고 하는 클라이언트의 질의를 가로채 다른 응답을 보낸다. 다른 방법으로는 HTTPS 통신에서 드러나는 SNI(ServerName Indication, 이하 SNI)를 활용하는 것이다. HTTPS는 앞서 언급한 HTTP의 평문을 TLS를 통해 암호화하여 통신하는 프로토콜이다. 이 과정에서 HTTP패킷 검사는 어느 정도 무력화되었지만 TLS 확장표준인 SNI를 이용하여 부분적으로 HTTPS를 통한 패킷 검사가 가능해졌다. 모든 TLS 암호화 통신은 암호화 과정에서 필요한 정보를 교환하는 handshake 과정을 선행한다. 이 과정에서 클라이언트는 ClientHello 메시지를 보내고 서버는 ServerHello 메시지로 응답한다. 서버는 클라이언트를 인식하고 클라이언트의 인증서를 인증하며 사용할 암호화 알고리즘을 협상하여 앞으로의 통신 과정에서 사용할 키를 합의한다. 동시에 클라이언트는 ClientHello 메시지를 받는 서버의 IP주소뿐만 아니라 서버 이름을 메시지에 포함하여 전송하는데, SNI을 통해 이를 실현한다[3]. 하지만 SNI는 서버 이름이 평문으로 담기기 때문에 클라이언트가 어느 웹사이트에 접속하고자 하는지 같은 네트워크에 존재하는 관찰자들은 알 수 있다. 이를 통해 관찰자는 웹사이트가 차단하고자 하는 웹사이트의 목록인 블랙리스트에 있는지 쉽게 확인이 가능하고 이 방법을 활용해 웹사이트로의 접속을 차단하는 방식을 SNI 필터링 혹은 블록킹이라고 한다.

본 논문에서는 일반적인 패킷 검사 및 차단 기법인 IP차단, DNS 스푸핑, SNI필터링을 살펴본다. 또한 이러한 기법을 우회할 수 있는 DNS통신에 TLS 프로토콜을 결합하는 여러 종류의 표준 DNS 암호화 프로토콜과 각각의 특징을 살펴보고 각각의 특징에 대해 비교해본다. 더하여 TLS handshake에서 드러나는 SNI를 암호화하고자 표준화가 진행되는 ESNI확장표준과 ESNI에서 갱신한 ECH의 동향을 살펴본다.

Ⅱ. TLS환경에서의 패킷 검사 및 차단 기법

2.1 IP 차단 기법

IP차단 기법은 특정 IP주소에 접속을 차단하는 것으로, 차단당한 IP주소에서 호스팅하는 모든 웹서버에 접속이 불가능하다. 더불어 오늘날 대부분의 클라우드

서비스는 하나의 IP주소에 여러 웹서버를 호스팅하기에 정상적인 웹사이트들이 차단될 수 있다는 단점이 있다. 일반 사용자의 인터넷 사용을 제한하는 특정 국가들에서 많이 사용하며, 한 연구에서는 중국의 검열의 84.5%가 IP차단 기법을 사용한다고 언급했다[9]. 중국이 검열하고 있는 주요 웹사이트가 다른 나라의 IP 대역을 사용하므로 IP차단 기술을 많이 사용한다.

2.2 DNS 스푸핑

DNS 스푸핑은 DNS서버의 응답을 변조하는 것으로 DNS서버보다 먼저 질의를 받은 감청자나 혹은 DNS서버 자체가 정확한 응답을 하지 않고 대체 주소 같은 응답을 주는 것이다. 우리나라의 불법유해정보사이트 차단에도 사용하는 방식이며 한국은 DNS서버를 운영하는 ISP 업체들의 협력을 받아 운영한다[48]. 차단하고자 하는 도메인 이름의 질의를 받으면 DNS응답의 IP주소를 다른 IP주소로 반환하거나 연결을 차단한다.

2.3 SNI 필터링

SNI필터링은 TLS 통신의 확장표준인 SNI가 차단하고자하는 특정 서버이름을 포함하였을 때 차단하는 방법이다. SNI는 하나의 IP주소로 여러 도메인을 호스팅하기에 도메인을 기준으로 발급하는 인증서를 판단하기 위해 서버의 도메인 정보를 담고있는 확장표준이다. 실제 TLS ClientHello 메시지의 SNI필드를 보면 클라이언트가 접속하고자 하는 도메인 정보를 알 수 있으며, SNI 필터링은 TCP RST패킷 혹은 FIN 패킷을 보내 접속을 종료시킨다[9].

Ⅲ. TLS 기반 DNS 암호화 통신 표준 기술

DNS암호화 통신은 여러 TLS 기반 프로토콜을 사용한다. TLS handshake를 선행하여 DNS통신의 내용을 암호화하여 DNS통신의 관찰자는 클라이언트가 접속하고자 하는 도메인 이름을 알 수 없게 된다.

3.1 DNS-over-TLS와 DNS-over-HTTPS

DNS-over-TLS와 DNS-over-HTTPS는 대표적인 DNS암호화 표준이다[12]. 주요 DNS서버와 브라우저에서 두 표준을 지원하며, 일반적으로 DNS-over-TLS를 지원하면 DNS-over-HTTPS도 지원한다.

3.1.1 DNS-over-TLS

DNS-over-TLS(이하 DoT)는 DNS통신 상황에서 감청을 방지하고 질의문을 부당하게 변경하는 것을 막기 위해 TLS를 활용해 평문을 암호화한다.[4] TCP/UDP 기반의 DNS 통신을 일반적인 TLS통신처럼 handshake 후 생성된 세션 키를 통해 암호화해 통신한다. 따라서 기존 네트워크상에 모든 관찰자가 볼 수 있었던 클라이언트의 질의를 평문으로 볼 수 없게 된다. 일반적으로 사용하던 TLS 암호화 통신을 DNS서버와 클라이언트에 적용하는 표준으로 기존 HTTPS 통신에 사용하는 포트 번호 443을 사용하는 대신 별도의 853번 포트를 사용한다. 따라서 관찰자의 입장에서 어떤 내용의 질의인 지는 알 수가 없지만, 853번 포트를 통한 질의를 했다는 사실은 알 수 있다. 현재 2022년 기준으로 DoT를 지원하는 주요 DNS 서버로는 Google, Cloudflare, Quad9, Adguard 등이 있다[5].

3.1.2 DNS-over-HTTPS

DNS-over-HTTPS(이하 DoH)[6]는 DoT와 마찬가지로 DNS암호화 통신을 하지만 기존 HTTPS통신을 사용한다는 점이 차이점이다. 기존 DNS 질의와 응답을 암호화해 HTTPS 요청과 응답에 각각 담아서 통신한다. GET 메소드를 사용할 때는 DNS 패킷을 암호화하여 URI에 담거나 JSON 포맷을 이용하는 경우도 있다[49]. POST 메소드를 사용할 때는 HTTP 메시지 안에 암호화한 DNS쿼리를 담는다. 포트번호는 기존 HTTPS통신에 쓰이는 443 포트번호를 사용하며, 따라서 기존 HTTPS 트래픽과 혼재되어 DoT 트래픽보다 알아보기 어렵다는 장점이 있다. 현재 2022년 기준으로 DoH를 지원하는 주요 DNS서버로는 Google, Cloudflare, Quad9, Adguard등이 있으며 Windows 11부터는 크롬과 파이어폭스 등, 기존 브라우저 설정으로 DoH를 지원한 것과는 다르게 운영체제상에서 DoH 설정을 지원한다[7].

3.1.3 DoT와 DoH의 Privacy Profile과 네트워크 관리와 프라이버시의 이슈

DoT와 DoH는 Privacy Profile이라는 개인 설정이 있다[14]. Privacy Profile은 Strict와 Opportunistic으로 분류되고, 전적으로 서버가 아닌 클라이언트에서 설정하는 개인 설정 Profile이다. 만약 클라이언트와 서버 간의 DNS암호화통신이 DNS서버 인증서 인증 실패 등, 특정 조건을 충족하지 못하여 실패하게 된다면 Opportunistic Privacy Profile은 원래 포트번호 53번을 사용하는 일반적인 DNS통신으로 평문을 전송한다. 반면에 Strict Privacy Profile은 DNS통신 자체가 실패하고 다시 DNS통신을 시도하지 않는다. 표 1을 보면 두 Privacy Profile의 차이점을 명확히 알 수 있

표 1. 사용 Privacy Profile에 따른 DNS 개인 정보 보호의 정도로 N은 No protected, D는 Detection Possible, P는 Protected를 의미한다
Table 1. Expected Degree of privacy protection according to the privacy profile

Usage Profile	Passive Attacker	Active Attacker
No Privacy	N	N
Opportunistic	N (D)	N (D)
Strict Privacy	P	P

다. 표의 Passive Attacker는 감청만을 하며, Active Attacker는 패킷 변조, 주입등을 할 수 있는 공격자를 의미한다. Authenticated Connection은 인증받은 DNS서버와의 연결을 의미하는데, SPKI pin 집합, DNS서버의 도메인 이름, DNSSEC 체인 등을 이용한다. 현재 DoH를 지원하는 브라우저는 대표적으로 크롬과 파이어폭스가 있다. 일반적으로 브라우저들은 Strict Privacy Profile 설정을 따로 지원하지 않고 기본적으로 사용자의 Privacy Profile은 Opportunistic Privacy Profile으로 설정되어있다. 하지만 사용자가 DoH를 브라우저에서 설정해도 DNS암호화 통신이 특정 조건을 충족하지 못하고 실패한다면 브라우저가 DNS통신을 평문으로 한다는 것을 사용자에게 알려주지 않는다[15]. 따라서 사용자는 자신의 Privacy Profile 또한 인지해야 DoT와 DoH가 주는 보안성의 혜택을 누릴 수가 있다.

DoT와 DoH는 사용하는 포트번호에 의하여 큰 차이점을 보인다. DoT는 네트워크 관리자의 입장에서 853번이라는 특수한 번호의 포트로 트래픽이 들어오기 때문에 443번 포트번호를 쓰는 DoH에 비해 트래픽 관리가 상대적으로 쉽다. 반면에 DoH는 443번 포트로 다른 HTTPS트래픽과 같이 DNS 질의와 응답이 발생하기 때문에 다른 트래픽과 구별하기 힘들어 클라이언트와의 DNS질의와 응답이 더 눈에 안 띄어 사용자의 프라이버시를 더 보장해줄 수 있다는 장점이 있다. 따라서 각각 장단점이 확실한 만큼 어떤 DNS암호화 프로토콜을 선호하는가는 계속해서 이슈가 되었다[8].

또 다른 이슈로는 DNS암호화를 지원하는 DNS서버는 사용자가 접속하는 웹사이트를 알 수 있다는 것이다. 클라이언트는 전적으로 DNS서버를 신뢰하는데 구글, 클라우드플레어 같은 특정 기업의 DNS서버에만 중심적으로 사용되는 경향이 문제가 되고 있다. 개인의 프라이버시를 보장하기 위한 기술이 오히려 특정 거대 기업에게 개인의 접속기록을 감청할 수 있는 기회를 줄 수도 있다는 우려가 있다. 따라서 그에 대한 대안으로 DNS암호화 통신을 하는 DNS서버를 계속해서 바꾸는 탈중앙화에 대한 방법을 제안하는 연구도 존재했다[20,21].

그에 따라 Oblivious DoH(이하 ODoH)라는 기술의 표준화가 진행중에 있다[22]. ODoH는 클라이언트와 쿼리를 받는 DNS 서버 사이에 프록시 서버를 두어 쿼리를 하는 종단 IP주소를 DNS서버가 알 수 없도록 한다. 더하여 ODoH는 TLS와 HTTPS에서 제공하는 암호화와는 별개로 프록시 연결만의 암호화 과정에 HPKE[23]라는 새로운 공개키 암호화 알고리즘을 사용한다. 우선 클라이언트는 DNS서버에게서 공개키를 얻은 뒤 그 암호화한 쿼리를 프록시 서버에 전송한다. 프록시 서버는 쿼리를 DNS 서버에 전송하고 DNS서버는 쿼리를 해독한 후 다시 암호화한 응답을 프록시 서버로 보낸다. 이 때 응답은 클라이언트의 암호화된 쿼리의 대칭키를 만드는데 필요한 키 자료를 담고 있어서 생성된 대칭키로 암호화한다. 따라서 클라이언트의 IP주소는 DNS서버에게 공개되지 않으므로 개인의 프라이버시 측면으로서의 이득을 볼 수 있지만, 중간에 프록시 서버를 지나가며 생기는 성능적인 손해가 우려되는 주요 문제이다. 실제로 ODoH의 도입에 가장 적극적인 Cloudflare는 북미에서 진행한 자체적인 실험으로 DoH보다는 느리지만 ODoH와 같이 클라이언트의 IP주소를 감춰주는 Tor위에서 사용하는 DoHoT (DNS-over-HTTPS over Tor)보다는 훨씬 나은 네트워크 응답시간을 가졌다고 보고했다[24]. 따라서 IP주소를 감추기 위해 Tor라는 네트워크 응답시간의 막대한 손해를 주는 방법 보다는 실제 환경에서 ODoH는 실용적인 대안이 될 것이라 기대되고 있다.

3.2 다른 DNS암호화 통신기술

3.2.1 DNS-over-QUIC

DoT, DoH를 제외하고 DNS-over-QUIC이라는 DNS암호화통신도 존재한다[10]. QUIC은 기존 TCP를 대체하고자 제시된 프로토콜이다. DNS-over-QUIC은 DNS통신을 QUIC프로토콜의 내장된 TLS프로토콜을 사용하여 암호화를 한다. DoQ는 DoT, DoH에 이어 세 번째, DNS암호화 통신 표준으로 2022년 5월 RFC 9250으로 표준화가 완료됐다[22]. DoH와 비교하여 DoQ를 지지하는 진영의 주장은 전송 프로토콜로 HTTP로 인해 발생할 수 있는 쿠키나 HTTP 헤더, ETag를 사용한 추적과 관련된 문제에서 자유롭다는

것이다[11]. 하지만 별도의 소프트웨어가 필요하고 DoT와 DoH에 비해 보급되어있지 않다[12]. DoQ는 점점 DNS 암호화 프로토콜에서 점유율을 늘려가고 있으며 실제로 Unbound[50]와 Knot[51]에서 도입을 준비중이고, DoQ에 최적화된 환경이 아님에도 핸드셰이크 시간은 실제 DNS 트래픽 중에서 DoH와 DoT보다 빠르다[25]. 지금은 DNS 서버로는 AdGuard[27]와 nextDNS[28]에서 지원하고 있고 포트번호는 784번을 사용하는 중이다.

3.2.2 DNSCrypt

이외에도 DNSCrypt라는 DNS암호화에 대한 다른 방안이 존재한다[13]. DNSCrypt는 여러 DNS 암호화 표준 기술이 등장하기 전부터 DNS 암호화통신의 필요성이 대두되며, 등장했던 대안이다. 따라서 표준화는 진행되지 않았으며 특정 소프트웨어에 의존하는 방법을 취한다. 다른 DNS암호화 통신과 다르게 TLS를 사용하지 않고 별도의 프록시 서버 소프트웨어를 통해 별도의 X25519-XSalsa20Poly1305이라는 암호화 알고리즘을 사용한다. 포트번호는 HTTPS 트래픽과 같이 포트번호 443을 사용한다. 현재 지원하는 DNS 서버로는 OpenDNS, Yandex 등에서 존재한다.

3.3 DNS 암호화 표준 기술의 비교

본 항에서는 DNS 암호화 기술 중 표준 기술로 인정받은 DoT, DoH와 DoQ를 비교한다. 크게 편의성, 대중성, 보안성에서 비교 분석하고자 한다.

3.3.1 사용자로서의 편의성

사용자로서의 편의성에서는 표준 기술을 이용하기 위한 클라이언트가 필요한 조건과 기존 DNS 통신과의 지연시간 차이로 평가하고자 한다. 필요한 조건으로는 별도의 소프트웨어나 지원이 필요한 지가 기준이다. 먼저 DoT를 사용자가 이용하기 위해서는 현재는 별도의 클라이언트 측의 리졸버를 조작할 수 있는 소프트웨어나 운영체제의 지원이 있어야 한다. 공식적으로 지원하는 OS는 안드로이드, 리눅스에서 지원한다. 클라이언트 측의 리졸버인 스텁 리졸버(stub resolver)의 역할을 할 수 있는 데스크탑용 소프트웨어로는 Stubby, Unbound, Knot resolver 등이 있고 모바일용으로는 Quad 9의 Connect라는 어플리케이션이 있다[26]. DoH는 별도의 소프트웨어나 OS의 지원으로도 사용할 수 있지만, Ⅲ-1-나에서도 언급했듯이 브라우저상에서 간단한 설정으로 일반 DNS 통신에서 DoH로 바꿀 수 있다. DoQ는 DoT와 마찬가지로 별도의 소프트웨어가 필요하며 현재 사용가능한 응용프로그램은 Adguard[27]뿐이다.

페이지 로드시간은 DNS쿼리와 응답 후 IP주소를 통해 요청된 페이지가 사용자에게 전달되기까지 걸리는 시간을 말한다. 공통된 환경에서는 암호화 과정을 거치는 DNS암호화 통신은 일반 DNS 통신보다 느리지만, [28] 또한, DoT와 DoH는 지연시간인 페이지 로드 시간에서 유의미한 시간 차이를 보이지 않았고 실제 일반 DNS통신과도 속도의 차이는 프라이버시 측면의 이득을 생각하면 감수할 정도라고 평가된다[30]. DoQ는 앞서 언급했듯이 핸드셰이크 시간에서는 DoH와 DoT를 앞서고[25], DoH와 DoT는 TCP 핸드셰이크 과정을 가지는 것과 달리 기본적으로 TLS를 적용하고 TCP를 대신하는 QUIC은 한 번의 핸드셰이크로 캐시된 정보를 이용하면 0-RTT 연결을 재개할 수 있다. 따라서 다음 DoQ통신은 더 빠른 속도를 기대할 수 있다[29]. 따라서 사용자의 편의성 측면에서는 상대적으로 필요한 조건이 낮은 DoH가 DoT와 DoQ보다 앞서는 것으로 확인되고, 지연시간에서는 DoQ가 비슷한 속도를 가지는 DoH와 DoT보다 편의성 측면에서 높다고 평가할 수 있다.

3.3.2 DNS 암호화 표준기술의 대중성

DNS암호화 표준기술의 대중성이란 얼마나 많은 DNS 서버 측에서 본 표준 기술을 지원하는가를 의미한다. 대중성 측면에서는 DoH와 DoT가 압도적으로 DoQ보다 높다고 할 수 있다. DoQ는 표준화가 확정된 지 얼마 지나지 않았고, DoT는 2016년, DoH는 2018년에 표준화가 되었기 때문이다. 단 지원하는 DNS서버의 숫자로만 보자면 DoT를 지원하는 DNS 서버의 숫자가 DoH를 지원하는 서버의 숫자보다 압도적으로 많다. Rapid 7의 Project Sonar[31]에서는 853 포트번호를 통해 얻을 수 있는 인증서의 데이터셋이 있는데 이를 통해 DoT를 지원하는 DNS서버의 수를 유추할 수 있다. 2021년 기준으로는 4000개 이상의 DoT 통신이 가능한 DNS서버가 존재한다[32]. 반면에 DoH 리졸버는 앞서 언급했듯이 GET, POST 같은 전달 방법과 형식으로는 wire format이나 JSON 형태로도 통일된 방식이 없고, 일반 TLS통신과 같은 443 포트번호를 사용하기에 정확한 리졸버의 숫자를 파악하기 힘들다. 또한 목적지 IP주소로 DNS 쿼리를 전송하는 DoT와 달리 DoH는 DNS서버의 URL에 쿼리를 보내기 때문에 사용하기 위해서는 URL주소를 정확히 알아야한다. 따라서 모든 IP주소로 DoT 쿼리를 보내 DoT를 지원하는 DNS서버의 숫자를 파악할

수 있는 DoT에 반해, 일반 사용자들이 접근가능한 오픈 리졸버의 숫자로 파악할 수 밖에 없는데 100여개로 추정된다[33]. DoQ는 Adguard, nextDNS를 제외하고는 알려진 오픈 리졸버는 없으며 개인 리졸버들이 있고 앞으로 지원하는 리졸버의 숫자가 늘어날 것이라는 전망이다[25]. 물론 브라우저 상에서 쉽게 지원해주는 DoH를 일반 사용자들이 가장 많이 사용하지만[41], 본 항에서는 지원하는 DNS 서버의 숫자로만 판단한다. 따라서 대중성 측면에서는 DoT, DoH, DoQ 순으로 정리할 수 있다.

3.3 DNS 암호화 표준기술의 보안성

DNS암호화 표준기술의 보안성은 DoT, DoH, DoQ 모두 TLS를 기본적으로 지원하므로 다른 측면으로 평가하기로 하였다. DNS 암호화 표준기술이 패킷 검사기술로부터 얼마나 저항성이 있는지를 평가하기로 한다. DNS암호화 기술인 DoT와 DoH는 표준화 초기에도 트래픽 분석을 통한 패킷 식별에 대한 우려가 있었다. 따라서 DNS 암호화 표준을 위한 패딩이 제안되었고, 그 내용은 128B와 468B의 DNS통신에 맞는 적절한 블록 크기의 패딩이었다[36]. 패딩이 없는 DoT 트래픽을 기계학습된 분류기를 통해 평균 95%의 정확도로 분류하는 연구가 있었고, 패딩을 통해 정확도가 줄어드는 것을 확인하였다.[34] DoH 또한 패딩이 없을 경우 거의 비슷한 정확도로 기계학습된 분류기로 분류가능한 것을 확인되었으며 심지어 패딩이 되어있음에도 90퍼센트가 넘는 높은 정확도로 분류가능한 것이 드러났다[35],[37]. 각 연구에서는 블록 크기의 패딩이 가지는 한계와 시간 패딩의 필요성을 말했다. 이 연구들은 실제 암호화한 DNS 통신도 정밀한 트래픽 분석으로 인해 식별될 수 있다는 결과를 보여주었으며, DoQ 또한 여기서부터 자유로울 수 없다[38]. 또한 서버 이름을 유출하는 SNI 필드와 IP주소를 검사하여 도메인 이름의 정보가 유출된다면 DNS 암호화 통신은 실효성을 잃을 수 있다. 실제로 SNI 필터링, IP 블록킹, DNS 스푸핑 같은 다양한 패킷 검사 기법을 사용하는 중국에서 검열하는 웹사이트의 37%만이 DNS 암호화 통신을 통해 접속할 수 있었다[39],[40]. 따라서 패킷 검사 측면에서 볼 때, 사용자가 요청하는 서버의 정보를 유출하지 않기 위해서는 다른 우회 기술과의 결합이 필요하다.

Ⅳ. TLS SNI 암호화 표준 동향

SNI를 암호화하는 표준은 SNI필터링을 방지하며 DNS암호화와 함께 사용하면 같은 네트워크 상에 있는 관찰자들은 더이상 클라이언트가 접속하려고 하는 웹서버의 도메인 혹은 서버이름을 알 수 없다. SNI 암호화 표준화 과정의 순서대로 동향을 살펴보기로 한다.

4.1 Encrypted SNI

Encrypted SNI(이하 ESNI)는 SNI를 암호화하고자 제시된 표준이었다[16]. ESNI는 draft가 계속 갱신되어 Encrypted ClientHello로 발전해 모질라, 클라우드플레어, 애플의 주도하에 IETF TLS working group에서 표준화가 진행중에 있다. 따라서 초창기 ESNI를 지원했던 브라우저는 파이어폭스가 유일했고, 지원하는 DNS서버는 Cloudflare 뿐이었다. ESNI는 클라이언트가 ESNI를 지원하는 DNS서버에 질의를 하는 것으로 시작한다. 여기서 DNS서버는 접속하고자 하는 웹사이트의 IP주소가 담긴 DNS응답과 함께 클라이언트가 접속하려고 하는 웹사이트의 공개키를 전달한다. 접속하려는 웹서버의 IP주소와 공개키를 전달받은 클라이언트는 웹서버의 SNI를 전달받은 공개키로 암호화한 후 ClientHello 메시지를 보낸다. 서버는 암호화된 SNI필드를 가지고 있던 개인키로 복호화한 후 확인한 SNI필드로 인증서 정보를 ServerHello를 보낸다.

ESNI는 이러한 과정을 거치기에 제약조건이 많다. 첫째로 웹사이트의 웹서버가 미리 약속된 DNS서버에게 자신이 생성한 공개키를 전달해야한다. 웹사이트가 DNS서버와 같은 기업의 CDN서버를 사용한다면 과정이 수월해진다. 둘째로 전달받은 공개키를 단기간 사용하여 중간자공격의 취약하지 않게 계속 갱신해야 한다. 셋째로 클라이언트는 ESNI를 지원하는 DNS서버, 웹사이트, 브라우저를 사용해야 지원받을 수 있다. 넷째로 DNS암호화 통신과 같이 활용해야 중간 감청자에게 자유로울 수 있다. 마지막으로는 호출되는 모든 웹서버가 모두 같은 CDN 서버에 연결되어야 한다. 별도의 CDN 서버가 있는 경우 그 서버의 SNI가 드러나기에 암호화된 SNI를 유추할 수 있기 때문이다[17].

4.2 Encrypted ClientHello

Encrypted ClientHello(이하 ECH)는 ESNI표준이 갱신되어 표준화가 진행되고 있다[16]. 2021년 1월부터 파이어폭스 브라우저에서 ESNI의 지원을 중지하고 실험적으로 ECH를 지원하고 2021년 10월부터 클라우드플레어 DNS서버에서 지원하고 있다. ECH와 ESNI의 가장 큰 차이점은 ECH는 ClientHello 메시지 전체를 암호화한다는 것이다. 따라서 SNI필드 말고도 ClientHello에 담겨있는 다른 TLS 확장표준까지 모두

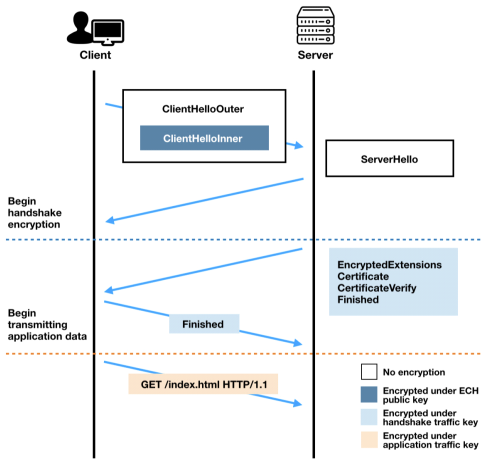


그림 2. ECH확장표준을 적용한 TLS1.3 Handshake 개요
Fig. 2. TLS1.3 Handshake with ECH

암호화된다. 특정 검열 우회도구가 사용하는 TLS 확장표준과 암호화 알고리즘 등이 담긴 TLS handshake를 Fingerprinting하여 심층패킷검사(Deep Packet Inspection)를 통해 알아낼 수 있다는 연구가 있었고[18], ECH는 ClientHello를 암호화하여 이를 해결하였다.

ECH는 ESNI와 같이 접속하고자 하는 웹서버의 공개키와 IP주소를 DNS서버로부터 받는다. 하지만 평문으로 handshake를 하는 ClientHelloOuter와 암호문으로 handshake를 하는 ClientHelloInner로 나뉘어 있다. ECH를 지원하면 ClientHelloInner를 사용하고 아니면 ClientHelloOuter를 사용하는 방식이다. 또한 실제 웹서버의 SNI로 ClientHelloOuter를 구성하는 게 아니라 CDN서버의 서버이름으로 SNI필드를 구성한다. 이런 방식으로 ClientHelloOuter를 구성하기에 그림 2에 나와 있듯이 중간자 공격자는 인증서의 도메인 이름을 CDN서버 혹은 다른 대면 서버의 이름으로 만들기 때문에 중간자 공격이 불가능해진다. CDN서버는 요청받은 ECH 메시지가 정상이면 웹서버에 전달한다. 따라서 감청자는 정확한 웹서버의 서버이름을 알 수 없고, TLS ClientHello 메시지의 대부분을 감청할 수 없다.[19] 하지만 ECH도 ESNI와 마찬가지로 브라우저, DNS서버의 지원이 필요하며, CDN서버 혹은 그에 대응하는 대면서버가 존재해야 사용할 수 있다.

V. TLS SNI 암호화 표준 동향

ECH의 가장 큰 장점이자 한계점은 SNI를 암호화한다는 것이다. SNI는 프라이버시 측면에서 보자면 클라이언트가 접속하고자 하는 서버의 이름을 드러내는 TLS 확장표준의 결점이라면, 보안을 관리하는 네트워크 관리자 측면에서는 단일 소켓에 여러 웹서버를 관리할 수 있고[42], 인증서 관리 측면에서도 Subjective Alternative Name 필드[43]를 통한 효율적인 관리가 불가능해지기 때문이다. 따라서 더 이상 SNI필드를 활용할 수 없는 CDN 서비스의 자정작용과 같은 별도의 노력이 필요하다.

또한 사용하는 DNS 서버, 브라우저, 접속하는 웹사이트의 CDN서버까지 모두가 지원해야 ECH의 혜택을 볼 수 있기 때문에 개인이 직접 구현하기 힘든 상황이고 프라이버시를 중요시 생각하는 거대 기업들의 지원에 한해서 구현할 수 있다는 것 또한 한계점이다. 다행히 브라우저는 메이저 브라우저인 파이어폭스[44], DNS서버와 CDN 서비스로는 클라우드플레어[45]가 계속해서 지원을 약속했다. 더하여 주요 브라우저 중 하나인 크롬에서는 이제야 논의를 시작한 상황[46]이고 다른 주요 CDN업체와 보안 라이브러리 관계자들도 상황을 예의주시하고 있다.

마지막으로 비교적 쉽게 패킷검사로 알 수 있었던 DNS쿼리의 내용과 TLS 핸드셰이크의 SNI필드가 DNS암호화와 ECH로 인해 알 수 없다면 새로운 검사 가능한 후보가 대두될 수도 있다는 것이다. 그 후보 중 하나는 암호화하지 않은 OCSP로[9],[47] 요청받은 서버의 인증서가 유효한가를 물을 때 인증서의 일련번호가 드러나면서 CT Log에 조회를 하여 알아볼 수 있다는 것이다.

VI. 패킷 검사 우회 표준기술의 미래전술네트워크 운용 방안

미래 무기체계에서는 현재보다 통신 단말들의 이동속도가 빨라지고, 더 많은 양의 데이터 송수신량을 요구한다. 또한 우주, 공중, 지상망의 원활한 통신이 필요하기 때문에 각 단말의 IP주소 할당, 변경 등이 신속히 이루어져야 한다. 그에 따라 제시된 개념이 지능형 네트워크 기술이다[52]. 지능형 네트워크 기술은 보다 유기적인 네트워크 관리를 위한 소프트웨어정의 네트워크(SDN) 기술, 네트워크 지능 기술 등을 총칭한다. 지능형 네트워크 하에서는 운용부대, 단말의 종류, IP주소 등 다양한 정보를 인공지능 등의 기술을 이용해 신속히 분석해 효율적으로 관리하게 된다[56].

미래전술네트워크에서의 DNS서버도 상용네트워크에서의 환경에서보다 더 효율적인 네트워크 관리가 필요하다. 예를 들어 작전 초기에 우주, 지상, 공중망

의 여러 단말 설정을 알아내고 전술망을 구성하기 위해 각 단말의 상위 부대의 서버의 도메인 이름을 DNS 서버에 조회할 수 있다. 만약 DNS통신에 중간 감청자가 있다면 패킷 검사를 통한 트래픽 분석의 취약점을 드러낼 수도 있다[54]. 이를 통해 도메인 이름을 알아내어 조회한 부대의 정보를 유추할 수 있다. 따라서 전술망의 DNS서버도 DNS암호화를 통해 이런 문제를 미연에 방지할 수 있을 것이라 기대된다. 또한 그동안 사람이 직접 수동적으로 DNS서버를 관리할 수 밖에 없었지만, 이미 SDN시뮬레이터를 통해 활발히 연구[55]되고 있는 SDN기술을 사용하는 미래전술네트워크에서는 지능적인 트래픽 관리를 할 수 있으리라 예상한다. SDN기술이 적용된 네트워크 환경에서는 상대적으로 기계학습한 분류기를 통한 트래픽 분석 공격이 가능한 DNS암호화만을 사용한 환경보다 복잡한 신경망 학습을 요구하는 등 제약조건이 많다[53]. 따라서 오히려 상용네트워크에 비해 트래픽 분석 공격에 취약하지 않으리라 기대된다.

또한 전술네트워크의 특성상 통합 관리되면서 하나의 DNS서버가 무력화되어도 작동할 수 있는 분산 DNS서버를 사용할 것이며 이는 단말이 접속하고자 하는 서버이름의 공개키를 DNS서버가 미리 저장해야 하는 ECH를 적용하기 적합한 환경이라 할 수 있다. 상용네트워크에서는 사용자가 직접 ECH를 지원하는 DNS서버에게 쿼리를 보내야하지만, 군의 네트워크에서는 정해진 DNS 서버에게만 쿼리를 보낼 수 있기에 ECH지원이 용이하기 때문이다.

적용가능한 DNS암호화 방법 중 DoH, DoT, DoQ를 비교하였을 때, 전술네트워크에 적합한 기술은 DoT라 할 수 있다. DoH는 포트번호 443을 사용해 일반 TLS통신과 구분할 수 없으므로 네트워크 트래픽 관리가 우선인 전술네트워크 환경에서는 부적합하다고 판단된다. 하지만 DoT는 별도의 포트번호 853번을 사용하여 네트워크 관리자가 DNS서버를 조회하는 트래픽을 관리할 수 있다. DoQ는 TLS통신에 익숙한 환경에서 새로운 QUIC 프로토콜에 대한 이해와 그에 따른 DoT에 비해 구현의 어려움이 우려된다.

Ⅶ. 결 론

본 논문에서는 대표적인 패킷 검사 기법을 살펴보고, 이를 우회할 수 있는 표준 기술을 살펴보았다. DNS암호화 기술들과 ECH 표준을 통해 네트워크 상에 모든 관찰자가 클라이언트가 접속하는 웹사이트의 도메인을 알 수 없게 된다. 또한 DNS암호화 통신 시 고려하여할 개인 설정, 주요 DNS암호화 표준 기술인 DoT, DoH, DoQ를 비교분석하여 소개했으며 ESNI에서 ECH까지의 SNI 암호화 기술을 설명하고 한계점을 설명하여 앞으로 표준화의 방향을 소개하였다. 마지막으로 미래전술네트워크 상에 DNS 암호화 기법과 ECH의 운용이 가능할 것임을 확인했다. 덧붙여 향후 미래전술네트워크에 적용할 수 있는 분산 DNS 테스트베드가 마련된다면 본 논문에서 제시한 DNS암호화 기법과 ECH의 실제 운용 실험을 통해 미래전술네트워크에서 수용할 수 있는 네트워크 비용인지 실증할 수 있는 후속 연구가 필요할 것이라 생각한다.

References

[1] E. Rescorla, *The Transport Layer Security (TLS) Protocol Version 1.3*, IETF RFC 8446, Aug. 2018. (https://doi.org/10.17487/rfc8446)

[2] D. Eastlake 3rd, *Domain Name System (DNS) IANA Considerations*, IETF RFC 2929, Sep. 2000. (https://doi.org/10.17487/rfc5395)

[3] D. Eastlake 3rd, *Transport Layer Security (TLS) Extensions: Extension Definitions*, IETF RFC 6066, Jan. 2011. (https://doi.org/10.17487/rfc6066)

[4] Z. Hu, et al., *Specification for DNS over Transport Layer Security (TLS)*, IETF RFC 7858, May 2016. (https://doi.org/10.17487/rfc7858)

[5] DNS Privacy Project, *Public Resolvers List*(2022), Retrieved Mar. 13, 2022, from https://dnsprivacy.org/public_resolvers/

[6] P. Hoffman and P. McManus, *DNS Queries over HTTPS (DoH)*, IETF RFC 8484, Oct. 2018. (https://doi.org/10.17487/rfc8484)

[7] Mauro Huc, *How to enable DNS over HTTPS (DoH) on Windows 11*(2021), Retrieved Mar. 13, 2022, from https://pureinfotech.com/enable-dns-over-https-windows-11/

[8] P. S. Kim, “안전한 웹사이트 접속을 위한 IETF 표준 기술 동향 분석(Analysis of IETF Standard Technology Trends for Secure Website Access),” *KICS Inf. and Commun. Mag.*, vol. 36, no. 6, pp. 32-40, May 2019.

[9] Z. Chai, A. Ghafari, and A. Houmansadr, “On the importance of Encrypted-SNI(ESNI) to

censorship circumvention," in *Proc. 9th USENIX Wkshp. FOCI 19*, Santa Clara, CA, USA, Aug. 2019.

[10] C. Huitema, et al., *Specification of DNS over Dedicated QUIC Connections*, IETF draft-ietf-dprive-dnsoquic-10, Feb. 2022. (https://doi.org/10.17487/rfc9250)

[11] Nata Kiseleva, AdGuard DNS-over-QUIC (2021), Retrieved Mar. 13, 2022, from https://adguard.com/en/blog/dns-over-quic.html

[12] C. Lu, et al., "An end-to-end, large-scale measurement of dns-over-encryption: How far have we come?," in *Proc. Internet Meas. Conf.*, pp. 22-35, Oct. 2019. (https://dl.acm.org/doi/10.1145/3355369.3355580)

[13] *DNSCrypt*, Retrieved Mar. 13, 2022, from https://www.dnscrypt.org/

[14] S. Dickinson, et al., *Usage Profiles for DNS over TLS and DNS over DTLS*, IETF RFC 8310, Mar. 2018. (https://doi.org/10.17487/rfc8310)

[15] Q. Huang, et al., "A comprehensive study of DNS-over-HTTPS downgrade attack," in *Proc. 10th USENIX Wkshp. FOCI 20*, Aug. 2020.

[16] E. Rescorla, et al., *TLS Encrypted Client Hello*, IETF draft-ietf-tls-esni-14, Feb. 2022.

[17] W. Lee and J. Hur, "A study of ESNI support status and limitations in TLS ecosystems," in *Proc. KCC 2021*, pp. 1167-1169, Jeju Island, Korea, Jun. 2021.

[18] S. Frolov and E. Wustrow, "The use of TLS in Censorship Circumvention," in *NDSS*, Feb. 2019. (https://doi.org/10.14722/ndss.2019.23511)

[19] C. Wood and C. Patton, Handshake, *Encryption: Endgame (an ECH update)*(2021), Retrieved Mar. 13, 2022, from https://blog.cloudflare.com/handshake-encryption-endgame-an-ech-update/

[20] A. Hounsel, et al., "Encryption without centralization: Distributing DNS queries across recursive resolvers," *ANRW '21*, pp. 62-68, Jul. 2021. (https://dl.acm.org/doi/10.1145/3472305.3472318)

[21] N. P. Hoang, et al., "Assessing the privacy benefits of domain name encryption," in *Proc. 15th ACM Asia Conf. Comput. and Commun. Secur.*, pp. 290-304, Oct. 2020. (https://dl.acm.org/doi/10.1145/3320269.3384728)

[22] C. Huitema, et al., *DNS over Dedicated QUIC Connections*, IETF RFC 9250, May 2022. (https://doi.org/10.17487/rfc9250)

[23] R. Barnes, et al., *Hybrid Public Key Encryption*, RFC 9180, Feb. 2022. (https://doi.org/10.17487/rfc9180)

[24] T. Verma, et al., *Improving DNS Privacy with Oblivious DoH in 1.1.1.1*(2020), Retrieved May 20, 2022, from https://blog.cloudflare.com/oblivious-dns/

[25] M. Kosek, et al., "One to rule them All? A first look at DNS over QUIC," in *Int. Conf. Passive and Active Netw. Meas.*, pp. 527-551, Mar. 2022. (https://dl.acm.org/doi/10.1007/978-3-030-98785-5_24)

[26] DNS Privacy Project, *DNS privacy clients*, https://dnsprivacy.org/dns_privacy_clients/, Retrieved May 20, 2022.

[27] Adguard Blog, *AdGuard DNS-over-QUIC*, Retrieved May 20, 2022, from https://adguard.com/en/blog/dns-over-quic.html

[28] A. Hounsel, et al., "Comparing the effects of DNS, DoT, and DoH on web performance," in *Proc. The Web Conf. 2020*, pp. 562-572, Apr. 2020. (https://dl.acm.org/doi/10.1145/3366423.3380139)

[29] Y. Cui, et al., "Innovating transport with QUIC: Design approaches and research challenges," *IEEE Internet Computing*, vol. 21, no. 2, pp. 72-76, Mar. 2017. (https://doi.org/10.1109/mic.2017.44)

[30] T. Böttger, et al., "An empirical study of the cost of DNS-over-HTTPS," in *Proc. Internet Meas. Conf.*, pp. 15-21, Oct. 2019. (https://dl.acm.org/doi/10.1145/3355369.3355575)

[31] Rapid7 Project Sonar, "*An introduction to project sonar*," https://www.rapid7.com/research/project-sonar/, Retrieved May 20, 2022.

[32] A. S. Jahromi and A. R. Abdou, "Comparative analysis of DoT and HTTPS certificate ecosystems," *NDSS Wkshp. MADWeb*, 2021. (https://doi.org/10.14722/madweb.2021.23027)

[33] curl - DNS over HTTPS, *DNS over HTTPS Publicly available servers*, Retrieved May 25, 2022, from https://github.com/curl/curl/wiki/DNS-over-HTTPS#publicly-available-servers.

[34] R. Houser, Z. Li, C. Cotton, and H. Wang, "An investigation on information leakage of DNS over TLS," in *Proc. 15th Int. Conf. Emerging Netw. Experiments and Technol.*, pp. 123-127, Dec. 2019. (https://doi.org/10.1145/3359989.3365429)

[35] S. Siby, M. Juarez, C. Diaz, N. Vallina-Rodriguez, and C. Troncoso, "Encrypted DNS → privacy? A traffic analysis perspective," in *27th Annu. Netw. and Distrib. Syst. Secur. Symp.*, Feb. 2020. (https://doi.org/10.14722/ndss.2020.24301)

[36] A. Mayrhofer, *Padding Policies for Extension Mechanisms for DNS (EDNS(0))*, RFC 8467, Oct. 2018. (https://doi.org/10.17487/rfc8467)

[37] J. Bushart, et al., "Padding Ain't Enough: Assessing the privacy guarantees of encrypted {DNS}," *10th USENIX Wkshp. FOCI 20*, 2020.

[38] G. Hu and K. Fukuda, "An analysis of privacy leakage in DoQ traffic," in *Proc. CoNEXT Student Wkshp.*, pp. 7-8, Dec. 2021. (https://doi.org/10.1145/3488658.3493782)

[39] L. Jin, et al., "Understanding the impact of encrypted DNS on internet censorship," in *Proc. Web Conf. 2021*, pp. 484-495, Apr. 2021. (https://doi.org/10.1145/3442381.3450084)

[40] S. Basso, "Measuring DoT/DoH blocking using OONI Probe: A preliminary study," in *NDSS DNS Privacy Wkshp.*, Feb. 2021.

[41] S. García, et al., "Large scale measurement on the adoption of encrypted DNS," *arXiv preprint arXiv:2107.04436*, 2021.

[42] V. Matthew, *How SNI Became a Battleground of Security v.s. Privacy*, Retreived May 25, 2022, from https://itnext.io/sni-from-a-to-z-72deffe942e1

[43] D. Cooper, et al., *Internet X.509 public key infrastructure certificate and certificate revocation list (CRL) profile*, RFC5280, May 2008. (https://doi.org/10.17487/rfc5280)

[44] K. Jacobs, *Encrypted Client Hello: the future of ESNI in Firefox*(2021), Retrieved May 24 2022, from https://blog.mozilla.org/security/2021/01/07/encrypted-client-hello-the-future-of-esni-in-firefox/.

[45] C. Patton, *Good-bye ESNI, hello ECH!, The Cloudflare Blog*, Retrieved May 24, 2022, from https://blog.cloudflare.com/encrypted-client-hello/

[46] Chrome Platform Status, *Feature: TLS Encrypted Client Hello*(2022), Retrieved May 30, 2022, from https://chromestatus.com/feature/6196703843581952,

[47] *Despite DoH and ESNI, with OCSP, web activity is insecure and not private*, Retrieved May 30, 2022, from http://blog.seanmcelroy.com/2019/01/05/ocsp-webactivity-is-not-private, 2019.

[48] J. Y. Jung, et al., "SNI field blocking and internet censorship," *IJSI*, vol. 10, no. 2, pp. 1-12, 2022. (https://doi.org/10.4018/ijsi.289601)

[49] *Make API requests to 1.1.1.1 over DoH, Cloudflare Docs*, Retrieved May 30, 2022, from https://developers.cloudflare.com/1.1.1.1/encryption/dns-over-https/make-api-requests/

[50] *NLnetlabs Unbound, dnsoverquic, configure*, Retrieved May 30, 2022, from https://github.com/NLnetLabs/unbound/commit/15eaa890f586f8044a698948247b23bca99a5124,

[51] CZ Domain Registry, *knot-dns/kdig-quic4* (2022), Retrieved May 30, 2022, from https://gitlab.nic.cz/knot/knot-dns/-/commits/kdig-quic4.

[52] J. Lee, et al., "Future communication · electromagnetic wave-intelligent network," *ICT Standardization Strategy Map*, 2019.

(https://doi.org/10.22156/CS4SMB.2021.11.06.014)

[53] J. Bhatia, et al., "SDN-based real-time urban traffic analysis in VANET environment," *Comput. Commun.*, vol. 149, pp. 162-175, 2020. (https://doi.org/10.1016/j.comcom.2019.10.011)

[54] B.-S. Kim, H.-C. An, and B.-H. No, "전술네트워크 및 전장관리체계 사이버 공격 및 방어 기술(Tactical Network and Battlefield Management System Cyber Attack and Defense Technology)," *KICS Inf. and Commun. Mag.*, vol. 32, no. 10, pp. 995-996, Aug. 2020.

[55] S. Lee, Y. Jeong, and B. Roh, "Design and implementation of simulation system for multi-domain tactical SDN using OPNET modeler," *J. KICS*, vol. 45, no. 4, pp. 739-747, Apr. 2020. (10.7840/kics.2020.45.4.739)

[56] J. Lee, K. Shin, C. Han, K.-H. Kang, and W. Hong, "Study on development of tactical networks using artificial intelligence technique," *J. KICS*, vol. 45, no. 1, pp. 191-200, Jan. 2020. (10.7840/kics.2020.45.1.191)

이 웅 희 (Woonghee Lee)

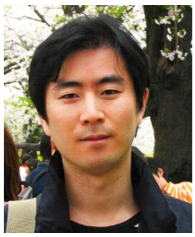
2021년 2월 : 고려대학교 컴퓨터학과 졸업

2021년 3월~현재 : 고려대학교 일반대학원 컴퓨터학과 석박통합과정

<관심분야> 네트워크 보안, 사이버범죄

[ORCID:0000-0002-9984-6879]

허 준 범 (Junbeom Hur)



2001년 2월 : 고려대학교 컴퓨터학과 학사

2005년 2월 : 한국과학기술원 전산학 석사

2009년 8월 : 한국과학기술원 전산학 박사

2009년 9월~2011년 8월 : University of Illinois at Urbana-Champaign 박사후 연구원

2011년~2015년 : 중앙대학교 컴퓨터공학부 조교수

2015년~2016년 : 고려대학교 컴퓨터학과 조교수

2016년~2021년 : 고려대학교 컴퓨터학과 부교수

2021년~현재 : 고려대학교 컴퓨터학과 교수

2021년~2022년 : ETH Zurich 방문교수

<관심분야> 응용 암호, 네트워크 보안, 클라우드 보안, 시스템 취약점

[ORCID:0000-0002-4823-4194]