# Extended TFO 프로토콜: Customized 쿠키를 통한 첫 번째 연결에서의 0-RTT 데이터 전송 기법

박홍근, 정경헌, 권태경

서울대학교

tonypark7277@snu.ac.kr, ghjeong@mmlab.snu.ac.kr, tkkwon@snu.ac.kr

# Extended TFO Protocol: A New Methodology for Sending 0-RTT Data at First Connection by Using Customized Cookie

Hong Geun Park, Gyeong Heon Jeong, Ted "Taekyoung" Kwon

Seoul National Univ.

요 약

네트워크 지연율의 증가가 온라인 서비스의 이익 감소로 연결되는 현재 사회에서는 네트워크 지연율을 낮추는 것이 매우 중요한 과제이다. 이러한 목적으로 설계된 TFO (TCP Fast Open)는 3-way handshake로 인해 강제되는 TCP의 1-RTT를 줄이기 위해 서버가 발급하는 쿠키를 사용한다. 서버에게 쿠키를 발급받은 첫 번째 연결 이후에는 SYN 패킷에 데이터를 담는 0-RTT 데이터 전송이 가능하다. 하지만 TFO를 사용하더라도 쿠키를 발급받기 전인 첫 번째 연결에서는 0-RTT 데이터를 전송할 수 없다. 따라서 본 논문에서는 Linux kernel의 TFO 프로토콜을 수정함으로써 사용자와 서버가 공통된 해시 키를 가지고 있을 경우, 사용자가 제작한 customized TFO 쿠키를 통해 첫 번째 연결부터 0-RTT 데이터 전송이 가능한 확장된 TFO 프로토콜을 소개한다.

## Ⅰ. 서 론

웹 페이지가 전부 로드되기까지 걸리는 시간인 Page Load Time (PLT)이 증가할수록 사용자가 해당 웹 페이지에 머무르는 시간이 짧아지며, 해당 웹 페이지를 다시 방문하지 않을 확률이 높아진다[1, 2]. 또한, Google의 검색 엔진에서는 사용자에게 더 짧은 PLT를 가진 웹 페이지를 우선으로 제공하는 알고리즘이 사용되고 있다[3]. 이처럼 현대 사회에서 낮은 네트워크 지연 시간의 중요성이 점점 커지고 있다. 따라서 네트워크 지연 시간을 줄이기 위해 통신의 여러 단에서 소요되는 지연율을 최소화하는 통신 프로토콜들이 개발되고 있다. 예를 들어, TLS (Transport Layer Security)의 TLS 1.3부터는 첫 번째 연결 수립 이후 세션을 재사용하여 다음 연결부터 0-RTT (Round Trip Time) 데이터를 전송하는 방식으로 기존의 1-RTT를 줄이고 있다[4].

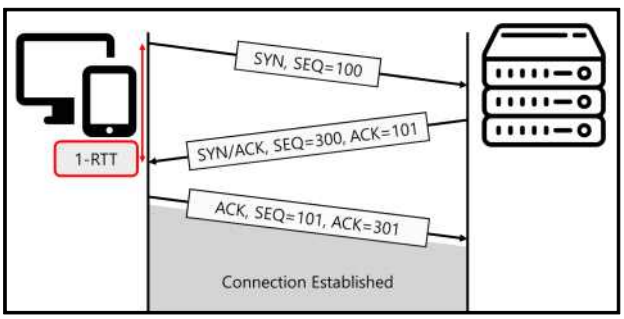


그림 1 TCP에서 연결을 맺기 위해 사용되는 3-Way Handshake.

전송 계층에서 송신자와 수신자 사이의 신뢰할 수 있는 연결을 제공하는 TCP (Transmission Control Protocol)에서는 그림 1에 묘사된 3-way handshake를 통해 연결이 수립되므로 데이터를 보내기까지 1-RTT가 강제된다. 이러한 TCP의 불가피한 1-RTT를 줄이기 위해 서버가 발급한 쿠키를 사용하는 TFO (TCP Fast Open)가 설계되었다[5]. TFO는 발급된 쿠키를 TCP header의 옵션을 통해 전달함으로써 TCP 연결 과정의 1-RTT를 줄일 수 있게 해준다. TFO에서 사용되는 쿠키는 발급을 요청한 사용자만 사용할 수 있도록 사용자의 IP 주소와 서버의 IP 주소를 해시 (Hash)하여 발급된다. TFO를 통해 연결을 수립하는 과정은 그림 2를 통해 확인할 수 있다. 그림 2의 ①에서 ③까지는 첫 번째 연결에서의, ④에서 ⑥까지는 두 번째 연결에서의 TFO 메시지 플로우를 나타낸다. ②를 통해 서버에게 쿠키를 받은 사용자는 다음 연결의 ④에서 사용하기 위해 해당 쿠키를 저장해 놓는다. 그림 2를 통해 알 수 있듯이, TFO는 서버가 발급한 쿠키를 통해 두 번째 연결의 SYN 패킷에 0-RTT 데이터를 담아 전송할 수 있다. 즉, 기존의 TFO에서는 서버에게 발급받은 쿠키가 없는 첫 번째 연결 시도에 0-RTT 데이터를 전송할 수 없다.

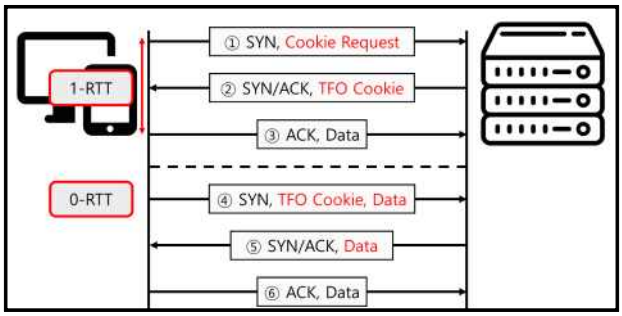


그림 2 사용자와 서버가 TFO를 이용하여 연결을 수립하는 과정.

본 논문에서는 사용자와 서버가 첫 번째 연결 수립 전에 보유하고 있는 공통된 해시 키를 사용하여 사용자가 직접 customized 쿠키를 제작함으

로써 첫 번째 연결부터 0-RTT 데이터를 전송할 수 있는 확장된 TFO 프로토콜인 Extended TFO (ETFO)를 소개하고자 한다. 앞으로는 기존 TFO의 쿠키와 ETFO의 쿠키를 구별하기 위해 기존의 TFO 쿠키를 쿠키로, Extended TFO에서 사용되는 customized 쿠키를 ETFO 쿠키라 서술하겠다. 또한 ETFO 쿠키를 만들기 위해 필요한 해시 키인 ETFO 해시 키는 사용자와 서버 간에 공유되어 있다고 가정한다.

## Ⅱ. 본론

본 논문에서는 사용자가 ETFO 쿠키를 제작하고, 서버가 ETFO 쿠키를 식별하고 검증할 수 있도록 Linux kernel에 구현된 TFO 프로토콜을 확장한다. 현재 Linux kernel에서는 서버가 사용자가 보낸 쿠키를 검증할 때, 그림 3과 같이 쿠키의 검증 결과에 따라 총 3가지의 메커니즘을 사용한다[6]. (이후 'No Key Match'를 NKM로, 'Backup Key'를 BK로, 'Primary Key'를 PK로 서술하겠다.) 세 가지의 메커니즘 중에서 BK 메커니즘은 서버의 해시 키를 보호하고자 사용된다. 서버가 하나의 해시 키를 계속 사용한다면 공격자가 서버의 해시 키를 계산할 수 있으므로, 서버는 주기적으로 해시 키를 바꾸어 주어야 한다[5]. 해시 키가 바뀌기 전에 쿠키를 받은 사용자가 해시 키가 바뀐 후의 연결에서도 해당 쿠키를 사용할 수 있도록, 서버는 그 직전에 사용했던 해시 키를 일정 기간 백업 키로 저장한다. 마지막으로 서버는 백업 키로 발급된 쿠키로 TFO 연결을 시도한 사용자에게 변경된 해시 키로 만든 쿠키를 전송한다.

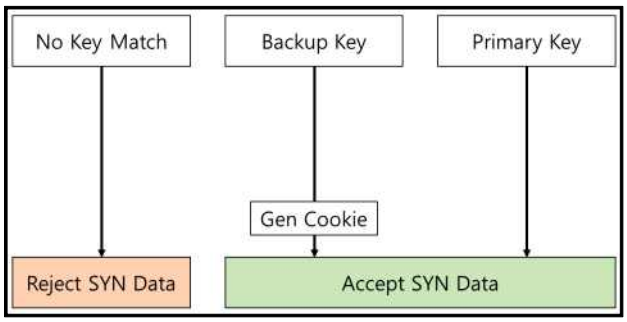


그림 3 서버에서 쿠키를 받았을 때의 세 가지 메커니즘.

본 논문에서 제시하는 기법은 이러한 백업 키에 대한 쿠키를 처리하는 메커니즘을 확장한다. 우선 ETFO를 사용하는 사용자는 자신의 IP 주소를 ETFO 해시 키로 ETFO 쿠키를 제작하여 서버에 전송한다. 서버는 사용자의 ETFO 쿠키를 식별할 수 없으므로 그림 3의 NKM 메커니즘을 사용하게 된다. 기존의 TFO는 NKM에 진입하게 된다면 무조건 0-RTT 데이터를 거절하고 일반적인 TCP 3-way handshake를 수행한다. 하지만 본 논문을 통해 제시된 ETFO에서는 NKM 메커니즘에 진입한 이후에도 BK 메커니즘을 수행할 수 있도록 분기점을 새롭게 추가함으로써 서버가 ETFO 쿠키를 검증할 수 있게 한다. 사용자의 ETFO 쿠키를 받아 NKM 메커니즘에 진입한 서버는 해당 쿠키가 ETFO 쿠키인지 다음의 방법으로 확인하게 된다. 사용자가 ETFO 쿠키를 제작할 때 사용한 정보들인 사용자의 IP 주소, ETFO 해시 키들은 모두 서버가 알 수 있기에, 서버는 해당 정보들로 ETFO 쿠키를 직접 생성 후 비교함으로써 검증한다. 만약 검증에 실패한다면, 해당 쿠키는 유효한 쿠키가 아닌 경우이므로 NKM 메커니즘을 계속 진행하여 0-RTT 데이터를 거절하고 일반적인 TCP 3-way handshake로 되돌아간다. 만약 검증에 성공한다면 ETFO로 제작된 쿠키이므로 서버는 0-RTT 데이터를 받아들임과 동시에, 추후 사용자가 TFO 연결에 사용할 쿠키를 발급해야 한다. 이를 위해 ETFO 서버는 그림 4와 같이 BK 메커니즘을 실행하며 쿠키를 생성하게 된다. 앞선 과정을 통해 생성된 쿠키를 받은 사용자는 해당 쿠키를 저장하여 추후의 TFO 연결에 사용하여 0-RTT 데이터를 전송할 수 있다.

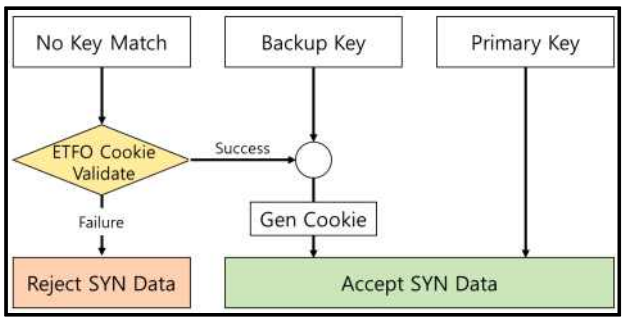


그림 4 NKM에서 BK로 넘어가는 분기점이 추가된 ETFO 프로토콜.

## Ⅲ. 결론

본 논문에서는 Linux kernel에서의 TFO 프로토콜을 확장하여 첫 번째 연결부터 0-RTT 데이터 전송이 가능한 Extended TFO 기법을 소개한다. 기존의 TFO는 첫 번째 연결을 통해 서버로부터 쿠키를 발급받기에, 첫 번째 연결에서는 0-RTT 데이터를 보낼 수 없다. 따라서 사용자는 ETFO 해시 키를 사용하여 ETFO 쿠키를 만들고, 서버는 NKM에서 BK로 넘어가는 분기점을 통해 쿠키를 검증함으로써 첫 번째 연결에서 0-RTT 데이터 전송이 가능한 Extended TFO 프로토콜을 제시한다. 또한 후속 연구로써 사용자와 서버가 공통으로 가지고 있다고 가정한 ETFO 해시 키를 첫 번째 연결 시도 전에 서로 공유할 방법을 연구하고자 한다.

## ACKNOWLEDGMENT

이 논문은 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임(No. RS-2023-00220985).

## 참 고 문 헌

[1] PR Newswire. 'customers are won or lost in one second,' finds new aberdeen report, 2008.

[2] Big Commerce. Big commerce https://www.bigcommerce.com/blog/. 4-tips-improve-ecommerce-bounce-rate-right-now-sellmore-video/.

[3] Google Inc. Using site speed in web search ranking, 2010

[4] E. Rescorla, Mozilla. 2018. The Transport Layer Security (TLS) Protocol Version 1.3. RFC 8446. Internet Engineering Task Force (IETF). https://datatracker.ietf.org/doc/html/rfc8446

[5] Cheng, et al. 2014. TCP Fast Open. RFC 7413. Internet Engineering Task Force (IETF). https://datatracker.ietf.org/doc/html/rfc7413

[6] Linux Kernel source code (v5.15.78) [Source code]. https://elixir.bootlin.com/linux/v5.15.78/source/net/ipv4/tcp_fastopen.c#216