(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2021-0103318
(43) 공개일자 2021년08월23일

(51) 국제특허분류(Int. Cl.)
H04L 29/06 (2006.01) *H04L 9/32* (2006.01)
(52) CPC특허분류
H04L 63/306 (2013.01)
H04L 63/18 (2013.01)
(21) 출원번호 10-2020-0017919
(22) 출원일자 2020년02월13일
심사청구일자 2020년02월13일

(71) 출원인
고려대학교 산학협력단
서울특별시 성북구 안암로 145, 고려대학교 (안암동5가)
(72) 발명자
이원준
서울특별시 강남구 선릉로 221, 104동 2005호 (도곡동, 도곡렉슬아파트)
노건희
경기도 성남시 분당구 서현로 170, B동 629호 (서현동, 풍림아이원플러스)
(뒷면에 계속)
(74) 대리인
특허법인엠에이피에스

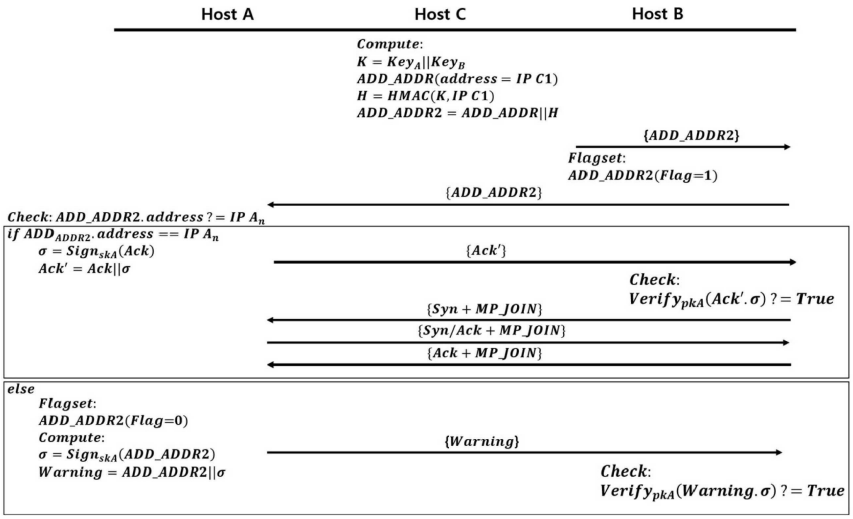
전체 청구항 수 : 총 11 항

(54) 발명의 명칭 **MPTCP의 서브플로우 보안 연결 방법 및 이를 위한 클라우드 서버, 호스트**

(57) 요 약

본 발명은 MPTCP의 서브플로우 보안 연결 방법 및 이를 이용한 클라우드 서버, 호스트에 관한 것으로서, a) 상기 클라우드 서버의 네트워크 주소에 호스트의 네트워크 주소 간에 핸드세이크(Handshake)를 수행하여 초기 연결된 상태에서, 상기 클라우드 서버에서 잉여 주소 광고2(Address Advertisement, ADD_ADDR2) 옵션을 수신하면, 상기 호스트에 확인 요청 메시지를 전송하는 단계; 및 b) 상기 호스트로부터 확인응답 메시지(ACK) 가 수신되면 서브플로우 연결을 수행하는 단계를 포함하되, 상기 확인응답 메시지는 상기 호스트가 보유한 네트워크 주소와 상기 확인 요청 메시지의 네트워크 주소가 일치하는 경우에 생성되는 방법일 수 있다.

대 표 도



(52) CPC특허분류
H04L 69/14 (2013.01)
H04L 69/163 (2013.01)
H04L 9/3242 (2013.01)
H04L 2209/72 (2013.01)
(72) 발명자
박후린
서울특별시 구로구 경인로67길 23, 201동 1202호 (신도림동, 신도림2차푸르지오)
노희준
경기도 성남시 분당구 서판교로 29, 918동 1802호 (판교동, 판교원마을한림풀에버아파트)

이 발명을 지원한 국가연구개발사업

과제고유번호	1711098528
과제번호	2019R1A2C2088812
부처명	과학기술정보통신부
과제관리(전문)기관명	한국연구재단
연구사업명	(이공)중견연구자지원(3억초과~5억이하)
연구과제명	Ambient Wi-Fi 백스캐터 네트워킹 기술
기 여 율	1/1
과제수행기관명	고려대학교
연구기간	2019.09.01 ~ 2020.02.29

명 세 서

청구범위

청구항 1

클라우드 서버와 호스트 간의 서버플로우 연결을 제공하기 위한 다중경로 전송 제어 프로토콜(MPTCP)의 서브플로우 보안 연결 시스템에 의해 수행되는 MPTCP의 서브플로우 보안 연결 방법에 있어서,

a) 상기 클라우드 서버의 네트워크 주소에 호스트의 네트워크 주소 간에 핸드세이크(Handshake)를 수행하여 초기 연결된 상태에서, 상기 클라우드 서버에서 잉여 주소 광고2(Address Advertisement, ADD_ADDR2) 옵션을 수신하면, 상기 호스트에 확인 요청 메시지를 전송하는 단계; 및

b) 상기 호스트로부터 확인응답 메시지(ACK) 가 수신되면 서브플로우 연결을 수행하는 단계를 포함하되,

상기 확인응답 메시지는 상기 호스트가 보유한 네트워크 주소와 상기 확인 요청 메시지의 네트워크 주소가 일치하는 경우에 생성되는 것인, MPTCP의 서브플로우 보안 연결 방법.

청구항 2

제 1 항에 있어서,

상기 b) 단계는,

상기 호스트로부터 전자 서명된 경고 메시지를 수신한 경우에, 상기 ADD_ADDR2 옵션을 무시하는 단계를 더 포함하되,

상기 경고 메시지는 상기 호스트가 보유한 네트워크 주소와 상기 확인 요청 메시지의 네트워크 주소가 일치하지 않는 경우에 생성되는 것인, MPTCP의 서브플로우 보안 연결 방법.

청구항 3

제 1 항에 있어서,

상기 a) 단계에서 상기 ADD_ADDR2 옵션의 플래그 값을 '1'로 설정하여 상기 확인 요청 메시지를 상기 호스트에 전송하면,

상기 호스트는 자신의 비밀 키를 이용하여 서명값을 생성한 후에 상기 서명값을 확인응답 메시지에 추가하여 상기 클라우드 서버에 전송하는 것인, MPTCP의 서브플로우 보안 연결 방법.

청구항 4

제 3 항에 있어서,

상기 b) 단계는,

상기 클라우드 서버가 상기 호스트의 공개키를 이용해 상기 확인응답 메시지를 검증하고, 상기 검증이 정상적으로 완료되면 MP_JOIN 옵션을 사용하여 서브플로우 연결을 수행하는 것인, MPTCP의 서브플로우 보안 연결 방법.

청구항 5

제 2 항에 있어서,

상기 클라우드 서버는 상기 ADD_ADDR2 옵션의 플래그 값을 '1'로 설정하여 상기 확인 요청 메시지를 상기 호스트에 전송하면, 상기 호스트는 상기 확인 요청 메시지의 플래그 값을 '0'으로 설정한 후 자신의 비밀 키를 이용하여 서명값을 생성하여 상기 서명값이 추가된 경고 메시지를 상기 클라우드 서버에 전송하고,

상기 클라우드 서버는 상기 호스트의 공개키를 이용하여 상기 경고 메시지를 검증한 후 상기 ADD_ADDR2 옵션을 무시하는 것인, MPTCP의 서브플로우 보안 연결 방법.

청구항 6

MPTCP의 서브플로우 보안 연결을 위한 클라우드 서버에 있어서,

MPTCP의 서브플로우 보안 연결 방법을 수행하기 위한 프로그램이 기록된 메모리; 및

상기 프로그램을 실행하기 위한 프로세서를 포함하며,

상기 프로세서는, 상기 프로그램의 실행에 의해,

호스트의 네트워크 주소와 자신의 네트워크 주소에 간에 핸드세이크(Handshake)를 수행하여 초기 연결된 상태에서, 잉여 주소 광고2(Address Advertisement, ADD_ADDR2) 옵션을 수신하면, 상기 호스트에 확인 요청 메시지를 전송하고, 상기 호스트로부터 확인응답 메시지(ACK) 가 수신되면 서브플로우 연결을 수행하되,

상기 확인응답 메시지는 상기 호스트가 보유한 네트워크 주소와 상기 확인 요청 메시지의 네트워크 주소가 일치하는 경우에 생성되는 것인, MPTCP의 서브플로우 보안 연결을 위한 클라우드 서버.

청구항 7

제 6 항에 있어서,

상기 프로세서는,

상기 ADD_ADDR2 옵션의 플래그 값을 '1'로 설정하여 상기 확인 요청 메시지를 상기 호스트에 전송하고,

상기 호스트로부터 비밀 키를 이용하여 생성된 서명값을 추가된 확인응답 메시지가 수신되면, 상기 호스트의 공개키를 이용해 상기 확인응답 메시지를 검증하고, 상기 검증이 정상적으로 완료되면 MP_JOIN 옵션을 사용하여 서브플로우 연결을 수행하는 것인, MPTCP의 서브플로우 보안 연결을 위한 클라우드 서버.

청구항 8

제 6 항에 있어서,

상기 프로세서는 상기 ADD_ADDR2 옵션의 플래그 값을 '1'로 설정하여 상기 확인 요청 메시지를 상기 호스트에 전송하고,

상기 호스트로부터 상기 확인 요청 메시지의 플래그 값을 '0'으로 재설정되고, 호스트의 비밀 키를 이용하여 생성된 서명값 추가된 경고 메시지를 수신하면, 상기 호스트의 공개키를 이용하여 상기 경고 메시지를 검증한 후 상기 ADD_ADDR2 옵션을 무시하는 것인, MPTCP의 서브플로우 보안 연결을 위한 클라우드 서버.

청구항 9

MPTCP의 서브플로우 보안 연결을 위한 호스트에 있어서,

MPTCP의 서브플로우 보안 연결 방법을 수행하기 위한 프로그램이 기록된 메모리; 및

상기 프로그램을 실행하기 위한 프로세서를 포함하며,

상기 프로세서는, 상기 프로그램의 실행에 의해,

클라우드 서버의 네트워크 주소와 자신의 네트워크 주소에 간에 핸드세이크(Handshake)를 수행하여 초기 연결된 상태에서, 잉여 주소 광고2(Address Advertisement, ADD_ADDR2) 옵션을 수행하기 위한 확인 요청 메시지가 수신되면, 상기 호스트가 보유한 네트워크 주소와 상기 확인 요청 메시지의 네트워크 주소가 일치하는 경우에 확인응답 메시지(ACK)를 생성하여 상기 클라우드 서버에 전송함으로써 서브플로우 연결을 수행하는 것인, MPTCP의 서브플로우 보안 연결을 위한 호스트.

청구항 10

제 9 항에 있어서,

상기 프로세서는,

상기 클라우드 서버로부터 상기 ADD_ADDR2 옵션의 플래그 값을 '1'로 설정한 확인 요청 메시지가 수신되면, 자신의 비밀 키를 이용하여 생성된 서명값을 추가하여 확인응답 메시지를 상기 클라우드 서버로 전송하는 것인, MPTCP의 서브플로우 보안 연결을 위한 호스트.

청구항 11

제 9 항에 있어서,

상기 프로세서는 상기 ADD_ADDR2 옵션의 플래그 값을 '1'로 설정한 확인 요청 메시지가 수신되면,

상기 호스트가 보유한 네트워크 주소와 상기 확인 요청 메시지의 네트워크 주소가 일치하지 않는 경우에 상기 확인 요청 메시지의 플래그 값을 '0'으로 재설정하고, 자신의 비밀 키를 이용하여 생성된 서명값 추가된 경고 메시지를 상기 클라우드 서버로 전송하는 것인, MPTCP의 서브플로우 보안 연결을 위한 호스트.

발명의 설명

기 술 분 야

[0001] 본 발명은 전송 경로를 추가하기 위한 옵션에 대해 전송 여부를 호스트와 클라우드 서버간에 확인함으로써 해당 옵션을 위조, 변조하여 인가되지 않은 제3자가 송수신단 사이의 경로에 위치하는 것을 방지할 수 있는 MPTCP의 서브플로우 보안 연결하기 위한 기술에 관한 것이다.

배 경 기 술

[0002] 사용자 단말과 네트워킹 기술의 발전에 따라 다중 네트워크 인터페이스가 탑재된 종단 기기가 폭넓게 활용되고 있다. 이러한 종단 기기는 여러 네트워크 링크 자원이 있지만, 단일 연결을 맺는 TCP에서는 동시에 활용할 수 없다.

[0003] 다중 네트워크 인터페이스를 동시에 활용하기 위해 대표적인 전송 계층 프로토콜인 TCP를 확장한 Multipath TCP(MPTCP)가 2013년 1월 IETF 실험 표준 RFC 6824로 출판되었다. MPTCP는 서브플로우(subflow)라는 개념을 도입하여 여러 경로를 통해 데이터를 전송하는 것이 가능하도록 설계되었다.

[0004] MPTCP는 여러 네트워크 링크 자원을 동시에 활용할 수 있도록 지원함으로써 보다 넓은 대역폭과 로드 밸런싱을 제공하므로 궁극적으로 사용자 경험을 향상시킬 수 있다. MPTCP에서 각각의 서브플로우는 단일 TCP 연결이며, 이러한 서브플로우의 집합이 MPTCP 연결이 될 수 있다. MPTCP는 애플사의 시리(Siri)를 지원하며, 국내에서는 KT의 네트워크를 이용하는 갤럭시 S6와 S6 엣지에 적용된다.

[0005] 도 1은 일반적인 MPTCP의 호스트 A와 호스트 B간의 연결 상태를 설명하는 개념도이다.

[0006] 도 1을 참고하면, 호스트 A는 IP A1과 IP A2를 가지고 있고, IP A1은 셀룰러 네트워크 인터페이스이고, IP A2는 WiFi 네트워크 인터페이스이다. 각각의 네트워크 인터페이스에 호스트 B와 3-웨이 핸드세이크(3-way handshake)를 수행하여 TCP 연결을 맺는다. 각각의 TCP 연결은 서브플로우이다.

[0007] 셀룰러 네트워크 인터페이스는 서브플로우 1이고, WiFi 네트워크 인터페이스는 서브플로우 2이다. 두 개의 서브플로우는 호스트 A와 호스트 B의 하나의 MPTCP 연결을 구성한다.

[0008] 이러한 다중경로 전송 경로를 제어하고, 서비스 품질(Quality of Service, QoS)를 제공하기 위하여 MPTCP는 여러 가지 옵션을 도입하였는데 그 중 하나의 옵션이 ADD_ADDR 옵션이다. MPTCP를 사용하는 종단 호스트가 추가적인 서브플로우를 맺기 위해 ADD_ADDR 옵션을 사용한다.

[0009] 모바일 환경, 네트워크 주소 변환(Network Address Translation, NAT)과 같은 상황에서 호스트의 IP 주소는 수시로 변경된다. 이때, 변경된 IP 주소를 상대 호스트에게 알려주기 위해서 ADD_ADDR 옵션을 사용한다. ADD_ADDR 옵션에 변경된 IP 주소를 삽입하여 전송하면, 이를 수신한 호스트는 해당 주소에 서브플로우 연결 요청 메시지를 전송하여 연결을 시도한다.

[0010] 도 2는 종래 기술에 따른 ADD_ADDR 옵션의 사용 과정을 설명하는 예시도이다._

[0011] MPTCP의 경로 시작은 핸드세이크 패킷인 SYN, SYN/ACK, ACK 패킷에 MP_CAPABLE 옵션을 포함하는 것을 제외하면 일반 TCP와 동일하게 진행된다.

[0012] 호스트 A와 호스트 B는 호스트 A의 주소(IP A1)와 호스트 B의 주소(IP B1)에 3-웨이 핸드세이크를 수행하여 초기 연결이 완료된 상태이다. 호스트 A는 IP A2가 사용할 경로를 추가하기 위해 패킷들에 MP_JOIN 옵션을 포함하여 호스트 B와 통신한다. 이때, 호스트 A가 자신의 또 다른 주소인 IP A2를 호스트 B에 알리기 위해 ADD_ADDR에 자신의 주소인 IP A2를 삽입하여 호스트 B에게 전송한다. 이를 수신한 호스트 B는 요청 받은 IP A2에 연결을 시

도한다.

[0013] 종래의 MPTCP에서의 ADD_ADDR 공격을 방어하기 위해, ADD_ADDR 옵션에 다른 호스트들을 인증하기 위한 해시 기반 메시지 인증 코드(HMAC, Hash-based Message Authentication)를 도입한 ADD_ADDR2 옵션을 사용할 수 있다. 그러나, ADD_ADDR2를 이용한 보안 방식은 HMAC 생성을 위한 키를 안전하게 교환해야 하는 문제점이 있다.

[0014] 이때, 안전한 HMAC 키 교환을 위해서 MPTCP에 암호화 방식을 결합함으로써 ADD_ADDR 옵션을 위조, 변조하여 허가되지 않은 제3자가 호스트들 사이에 전송되는 메시지들을 도청, 위조, 누락하는 ADD_ADDR 공격을 방어하고 있다. 이러한 방법들은 송수신단에서 메시지의 암호화, 복호화를 위한 추가적인 정보를 교환해야 하기 때문에 매우 큰 공간적, 시간적 오버헤드(Overhead)를 야기하는 문제점이 있다.

선행기술문헌

특허문헌

[0015] (특허문헌 0001) 대한민국 공개특허 제10-2050133호(발명의 명칭: 사용자 요청에 따른 MPTCP 통신 보안 활성화 방법 및 사용자 단말)

발명의 내용

해결하려는 과제

[0016] 본 발명은 전술한 문제점을 해결하기 위하여, 본 발명의 일 실시예에 따라 클라우드 서버가 수신한 ADD_ADDR 옵션을 신뢰하는 호스트에게 역방향 확인(Backward Confirmation)을 요청하여 호스트로부터 ADD_ADDR 옵션 내의 IP 주소와 자신의 IP 주소가 일치하는 경우에 서브플로우 연결을 제공하도록 함으로써 MPTCP에 존재하던 ADD_ADDR 공격과 새롭게 정의한 ADD_ADDR2 공격을 더욱 효율적으로 방어하도록 하는 것에 목적이 있다.

[0017] 다만, 본 실시예가 이루고자 하는 기술적 과제는 상기된 바와 같은 기술적 과제로 한정되지 않으며, 또 다른 기술적 과제들이 존재할 수 있다.

과제의 해결 수단

[0018] 상기한 기술적 과제를 달성하기 위한 기술적 수단으로서 본 발명의 일 실시예에 따른 클라우드 서버와 호스트 간의 서버플로우 연결을 제공하기 위한 다중경로 전송 제어 프로토콜(MPTCP)의 서브플로우 보안 연결 시스템에 의해 수행되는 MPTCP의 서브플로우 보안 연결 방법은, a) 상기 클라우드 서버의 네트워크 주소에 호스트의 네트워크 주소 간에 핸드세이크(Handshake)를 수행하여 초기 연결된 상태에서, 상기 클라우드 서버에서 잉여 주소 광고2(Address Advertisement, ADD_ADDR2) 옵션을 수신하면, 상기 호스트에 확인 요청 메시지를 전송하는 단계; 및 b) 상기 호스트로부터 확인응답 메시지(ACK) 가 수신되면 서브플로우 연결을 수행하는 단계를 포함하되, 상기 확인응답 메시지는 상기 호스트가 보유한 네트워크 주소와 상기 확인 요청 메시지의 네트워크 주소가 일치하는 경우에 생성되는 방법일 수 있다.

[0019] 또한, 본 발명의 다른 일 실시예에 따른 MPTCP의 서브플로우 보안 연결을 위한 클라우드 서버는, MPTCP의 서브플로우 보안 연결 방법을 수행하기 위한 프로그램이 기록된 메모리; 및 상기 프로그램을 실행하기 위한 프로세서를 포함하며, 상기 프로세서는, 상기 프로그램의 실행에 의해, 호스트의 네트워크 주소와 자신의 네트워크 주소에 간에 핸드세이크(Handshake)를 수행하여 초기 연결된 상태에서, 잉여 주소 광고2(Address Advertisement, ADD_ADDR2) 옵션을 수신하면, 상기 호스트에 확인 요청 메시지를 전송하고, 상기 호스트로부터 확인응답 메시지(ACK) 가 수신되면 서브플로우 연결을 수행하되, 상기 확인응답 메시지는 상기 호스트가 보유한 네트워크 주소와 상기 확인 요청 메시지의 네트워크 주소가 일치하는 경우에 생성되는 것이다.

[0020] 본 발명의 다른 일 실시예에 따른 MPTCP의 서브플로우 보안 연결을 위한 호스트는, MPTCP의 서브플로우 보안 연결 방법을 수행하기 위한 프로그램이 기록된 메모리; 및 상기 프로그램을 실행하기 위한 프로세서를 포함하며,

[0021] 상기 프로세서는, 상기 프로그램의 실행에 의해, 클라우드 서버의 네트워크 주소와 자신의 네트워크 주소에 간에 핸드세이크(Handshake)를 수행하여 초기 연결된 상태에서, 잉여 주소 광고2(Address Advertisement, ADD_ADDR2) 옵션을 수행하기 위한 확인 요청 메시지가 수신되면, 상기 호스트가 보유한 네트워크 주소와 상기 확인 요청 메시지의 네트워크 주소가 일치하는 경우에 확인응답 메시지(ACK)를 생성하여 상기 클라우드 서버에

전송함으로써 서브플로우 연결을 수행하는 것이다.

발명의 효과

[0022] 전술한 본 발명의 과제 해결 수단에 의하면, 허가되지 않은 제 3자의 호스트가 자신을 경유하는 서브플로우를 생성하는 것을 방지함으로써 도청을 방지하고, 도청을 기반으로 하는 네트워크 공격을 추가적으로 방지할 수 있다. 따라서, 본 발명은 다중 네트워크 인터페이스를 활용하는 모바일 기기, 데이터베이스 환경에서 보다 무결성을 요구하는 데이터를 향상된 수준의 보안성과 성능으로 처리할 수 있으며, 플랫폼의 종류와 상관없이 폭넓은 보안성을 제공할 수 있다.

[0023] 또한, 본 발명은 HMAC 키 교환을 위해 MPTCP에 암호화하는 방식에 의존하지 않기 때문에 ADD_ADDR2 공격에 안전할 뿐만 아니라 경량화된 서브플로우 연결을 제공할 수 있다.

도면의 간단한 설명

[0024] 도 1은 일반적인 MPTCP의 호스트 A와 호스트 B간의 연결 상태를 설명하는 개념도이다.

도 2는 종래 기술에 따른 ADD_ADDR 옵션의 사용 과정을 설명하는 예시도이다.

도 3은 본 발명의 일 실시예에 따른 MPTCP의 서브플로우 보안 연결 시스템의 구성을 설명하는 도면이다.

도 4는 ADD_ADDR 공격의 수행 상태를 설명하는 예시도이다.

도 5은 ADD_ADDR2 공격의 수행 상태를 설명하는 예시도이다.

도 6은 본 발명의 일 실시예에 따른MPTCP의 서브플로우 보안 연결 방법을 설명하는 도면이다.

발명을 실시하기 위한 구체적인 내용

[0025] 아래에서는 첨부한 도면을 참조하여 본 발명이 속하는 기술 분야에서 통상의 지식을 가진 자가 용이하게 실시할 수 있도록 본 발명의 실시예를 상세히 설명한다. 그러나 본 발명은 여러 가지 상이한 형태로 구현될 수 있으며 여기에서 설명하는 실시예에 한정되지 않는다. 그리고 도면에서 본 발명을 명확하게 설명하기 위해서 설명과 관계없는 부분은 생략하였으며, 명세서 전체를 통하여 유사한 부분에 대해서는 유사한 도면 부호를 붙였다.

[0026] 명세서 전체에서, 어떤 부분이 다른 부분과 "연결"되어 있다고 할 때, 이는 "직접적으로 연결"되어 있는 경우뿐 아니라, 그 중간에 다른 소자를 사이에 두고 "전기적으로 연결"되어 있는 경우도 포함한다. 또한 어떤 부분이 어떤 구성요소를 "포함"한다고 할 때, 이는 특별히 반대되는 기재가 없는 한 다른 구성요소를 제외하는 것이 아니라 다른 구성요소를 더 포함할 수 있는 것을 의미하며, 하나 또는 그 이상의 다른 특징이나 숫자, 단계, 동작, 구성요소, 부분품 또는 이들을 조합한 것들의 존재 또는 부가 가능성을 미리 배제하지 않는 것으로 이해되어야 한다.

[0027] 본 명세서에서 '단말'은 휴대성 및 이동성이 보장된 무선 통신 장치일 수 있으며, 예를 들어 스마트 폰, 태블릿 PC 또는 노트북 등과 같은 모든 종류의 핸드헬드(Handheld) 기반의 무선 통신 장치일 수 있다. 또한, '단말'은 네트워크를 통해 다른 단말 또는 서버 등에 접속할 수 있는 PC 등의 유선 통신 장치인 것도 가능하다. 또한, 네트워크는 단말들 및 서버들과 같은 각각의 노드 상호 간에 정보 교환이 가능한 연결 구조를 의미하는 것으로, 근거리 통신망(LAN: Local Area Network), 광역 통신망(WAN: Wide Area Network), 인터넷 (WWW: World Wide Web), 유무선 데이터 통신망, 전화망, 유무선 텔레비전 통신망 등을 포함한다.

[0028] 무선 데이터 통신망의 일례에는 3G, 4G, 5G, 3GPP(3rd Generation Partnership Project), LTE(Long Term Evolution), WIMAX(World Interoperability for Microwave Access), 와이파이(Wi-Fi), 블루투스 통신, 적외선 통신, 초음파 통신, 가시광 통신(VLC: Visible Light Communication), 라이파이(LiFi) 등이 포함되나 이에 한정되지는 않는다.

[0029] 이하의 실시예는 본 발명의 이해를 돕기 위한 상세한 설명이며, 본 발명의 권리 범위를 제한하는 것이 아니다. 따라서 본 발명과 동일한 기능을 수행하는 동일 범위의 발명 역시 본 발명의 권리 범위에 속할 것이다.

[0030] 이하 첨부된 도면을 참고하여 본 발명의 일 실시예를 상세히 설명하기로 한다.

[0031] 도 3은 본 발명의 일 실시예에 따른 MPTCP의 서브플로우 보안 연결 시스템의 구성을 설명하는 도면이다.

[0032] MPTCP의 서브플로우 보안 연결 시스템은 사용자 단말(100)과 클라우드 서버(200)를 포함한다.

[0033] 예시적으로, 클라우드 서버(200)는 통신모듈, 메모리, 프로세서 및 데이터베이스를 포함한다. 통신모듈은 사용자 단말(100)과 데이터 통신을 처리한다. 메모리에는 사용자 단말(100)에 대해 네트워크 인터페이스를 제공하기 위한 서브플로우 보안 연결 방법을 수행하기 위한 프로그램이 저장되어 있다. 프로세서는 메모리에 저장된 프로그램을 실행하되, 이하에서 설명할 사용자 단말(100)을 통해 처리되는 서브플로우 보안 연결 방법을 수행하기 위한 프로그램의 각 동작에 대응하는 처리를 수행한다.

[0034] 구체적으로, 사용자 단말(100)은 스마트폰과 같이 다양한 기능이 탑재 되거나 또는 다양한 프로그램(예를 들어, 모바일 앱 등)을 자유롭게 탑재 및 설치할 수 있는 지능형 단말일 수 있다.

[0035] 도 3에 도시된 바와 같이, MPTCP의 서브플로우 보안 연결 시스템은 잉여 주소 광고(Address Advertisement, ADD_ADDR) 옵션을 이용한 공격을 방어하기 위한 것으로써, 기존의 MPTCP에 존재하던 ADD_ADDR 공격과 새롭게 정의한 ADD_ADDR2 공격을 더욱 효율적으로 방어하는 서브플로우 연결 방법을 제공한다.

[0036] 공격자(adversary) 호스트(300)는 네트워크 상의 모든 패킷을 도청할 수 있다고 가정하고, 도청한 내용을 바탕으로 패킷을 생성하여 위변조할 수 있는 모델이라고 가정할 수 있다. 공격자 호스트(300)는 호스트(100)와 클라우드 서버(200) 사이에 위치하여 위조된 ADD_ADDR 옵션을 클라우드 서버(200)에 전송한다. 클라우드 서버(200)는 수신한 ADD_ADDR 옵션을 신뢰하는 호스트(100), 즉 이미 네트워크 경로가 생성되어 연결되어 있는 호스트(100)에게 플래그 값을 설정한 ADD_ADDR옵션(Flagset: ADD_ADDR2(Flag=1))을 전송한다. 플래그 값을 설정한 ADD_ADDR 옵션을 수신한 호스트(100)는 ADD_ADDR 옵션 내의 IP 주소와 자신의 IP 주소를 대조하여 이에 대한 응답 메시지를 클라우드 서버(200)에 전송한다.

[0037] 일반적인 ADD_ADDR 옵션은 수신단에서 메시지 인증을 수행할 방법이 없기 때문에 인가받지 않은 제 3자의 호스트(또는 공격자 호스트)에서 ADD_ADDR 옵션을 위조 혹은 변조하여 공격자 호스트를 경유하도록 하는 서브플로우를 생성하도록 유도할 수 있다.

[0038] 공격자 호스트에서는 ADD_ADDR 옵션에 자신의 IP 주소를 삽입하여 전송하면, 이를 수신한 호스트는 공격자 호스트의 IP 주소에 서브플로우 연결 요청 메시지를 전송한다. 이러한 방법을 이용하여, 공격자 호스트(300)는 통신하고 있는 클라우드 서버(200)와 호스트(100) 사이에 위치하여 전송되는 메시지들을 도청, 위조, 누락할 수 있다.

[0039] 도 4는 ADD_ADDR 공격의 수행 상태를 설명하는 예시도이다.

[0040] 호스트 A의 네트워크 주소인 IP A1과 호스트 B의 네트워크 주소인 IP B1에 경로가 있다고 가정한다.

[0041] 공격자인 호스트 C는 자신의 IP 주소인 IP C1으로 가짜 ADD_ADDR 옵션을 생성하여 호스트 B에게 전송한다. 호스트 B는 ADD_ADDR 옵션을 확인하여 호스트 C의 주소인 IP C1으로 MPTCP서브플로우 연결 요청 메시지인 SYN+MP_JOIN를 전송한다. 서브플로우 연결 요청 메시지를 수신한 호스트 C는 해당 메시지의 발신 주소를 자기 자신의 IP 주소인 IP C1으로 바꾸어 호스트 A에게 전송한다. 위와 같은 과정을 계속 반복하여 최종적으로 호스트 C를 경유하는 하나의 서브플로우를 만들 수 있다.

[0042] 이러한 ADD_ADDR 공격을 방지하기 위해서 ADD_ADDR 옵션에 해시 기반 메시지 인증 코드(HMAC, Hash-based Message Authentication)를 도입하여 RFC 6824에서 ADD_ADDR2라고 정의하고 있다.

[0043] 도 5은 ADD_ADDR2 공격의 수행 상태를 설명하는 예시도이다.

[0044] 공격자 호스트인 호스트 C는 하나 이상의 서브플로우를 점유하고, 패킷에 대해서 수정이나 누락시킬 수 있으며, 적합한 HMAC 생성을 위한 호스트 A의 키(Key_A)와 호스트 B의 키(Key_B)를 호스트 A와 호스트 B 간의 3-웨이 핸드세이크를 도청함으로써 알고 있다고 가정한다.

[0045] RFC 6824에 따르면, 호스트 A가 ADD_ADDR2 옵션을 새로운 IP 주소와 함께 호스트 B에게 전송한다면, 호스트 A는 HMAC 키와 ADD_ADDR2 옵션의 일부를 넣어 HMAC 값을 생성한다.

[0046] HMAC 키는 K= Key_A ‖ Key_B 로 생성된다. 이때의 Key_A 와 Key_B 는 초기 3-웨이 핸드세이크 과정에서 교환되고, ‖는 결합(concatenation)을 의미한다.

[0047] ADD_ADDR 옵션은 주소 ID, IP 주소 및 포트 필드로 구성되어 있으나, 호스트 C는 ADD_ADDR 옵션 필드 중에서 IP 주소 필드를 자신의 IP 주소인 IP C1으로 채우고 나머지 필드는 임의로 선택한다. 다음으로, 호스트 C는 획득한

Key_A 와 Key_B를 바탕으로 HMAC 키를 생성한다.

[0048] 이렇게 생성한 K와 IP C1에 대한 HMAC 값을 H=HMAC(K, IP C1)과 같이 생성하여 ADD_ADDR 메시지와 결합하여 ADD_ADDR2 메시지를 생성한다.

[0049] HMAC 값(H)이 생성된 후에 호스트 C는 가짜 ADD_ADDR2 메시지를 호스트 B에게 전송한다. 이를 수신한 호스트 B는 다른 HMAC 값인 H' 을 생성하고, H'와 수신한 H가 일치하는지를 확인한다.

[0050] 호스트 B는 H'와 H가 일치하면 ADD_ADDR2 옵션은 적합한 것으로 판단하여 ADD_ADDR2 공격은 성공하게 된다. 즉, HMAC값이 추가된 ADD_ADDR2 옵션으로는 ADD_ADDR2공격을 방어할 수 없다.

[0051] 이는 HMAC 값을 으로 MPTCP서브플로우 연결 요청 메시지에 추가하더라도, HMAC 생성을 위한 키가 안전하게 교환되지 않는다면, 공격자인 호스트 C는 얼마든지 자신의 메시지에 대해서 적합한 HMAC키를 생성하여 인증 과정을 우회할 수 있기 때문이다. 이러한 방식으로ADD_ADDR 공격을 수행하는 것을 ADD_ADDR2 공격이라고 정의한다.

[0052] ADD_ADDR2 공격은 도4에 도시된 ADD_ADDR 공격과 동일하지만, ADD_ADDR 공격과 ADD_ADDR2 공격은 공격자가 적합한 HMAC 값을 생성할 수 있는 능력이 있는지가 큰 차이점이라 할 수 있다. ADD_ADDR2 공격을 수행할 수 있는 공격자는 HMAC 생성을 위한 키를 입수하여 적합한 HMAC 값을 생성하여 HMAC로 보호받고 있는 ADD_ADDR 옵션을 위조할 수 있다.

[0053] ADD_ADDR2 메시지를 위한 함수 및 보안과 관련된 함수를 정의하면 다음과 같다.

[0054] 정의 1에서는 ADD_ADDR2 메시지를 위한 함수에서는 메시지 생성을 위한 함수(Gen())을 정의하고, 메시지 생성 함수는 ID, IP 주소, 포트 번호 및 플래그를 입력으로 받아 ADD_ADDR2 옵션을 생성한다. 이렇게 생성된 메시지는 Send()와 Rev() 함수를 통해서 송수신할 수 있도록 한다.

[0055] 정의 1은 ADD_ADDR2 메시지를 위한 함수로서 다음과 같이 ADD_ADDR2 옵션 교환을 위한 함수를 정의한다.

[0056] $Gen(A_{ID}, A_{IP}, P_t, H)$ 함수는 호스트 간의 연결 ID를 나타내는 A_{ID}, 연결을 원하는 주소를 나타내는 A_{IP}, 포트 번호를 나타내는 P_t, HMAC 값을 나타내는 H를 입력받아 ADD_ADDR2 옵션을 생성한다.

$Send_A^B(m)$

[0057] 함수는 호스트 A가 메시지 m을 호스트 B에게 전송하기 위한 함수이다.

[0058] $Rev_A(m)$ 함수는 호스트 A가 메시지 m을 수신하기 위한 함수이다.

[0059] $Gen(A_{ID}, A_{IP}, P_t, H)$에 의해서 생성된 ADD_ADDR2 메시지의 집합은 $Q=\{m|m= Gen(A_{ID}, A_{IP}, P_t, H)\}$으로 정의된다.

[0060] 정의 2에서는 보안과 관련된 함수를 정의한다. 플래그를 설정하는 함수(Flagset()), IP 주소를 확인하는 함수(Check()), 공개키와 비밀키를 생성하는 함수(Kgen()), 메시지에 서명하는 함수(Sign()), 그리고 서명을 검증하는 함수(Verify())를 정의한다.

[0061] 정의 2 (보안 관련 함수)에서는 보안 인자(n)가 주어질 경우에 메시지에 대한 서명과 검증을 위한 함수를 아래와 같이 정의한다.

[0062] $Flagset_A(m, F, f)$ 함수는 호스트 A가 메시지 m의 플래그 값 F를 f로 설정한다.

[0063] $Check_A(m, F, f)\in\{0, 1\}$ 함수는 호스트 A가 메시지 m의 플래그 값 F와 f를 비교하여 같은 경우 1을 그렇지 않은 경우 0을 반환한다.

[0064] (pk, sk)=Kgen(n) 함수는 n비트 길이의 공개키와 비밀키 쌍(pk, sk)을 생성한다.

[0065] $\sigma=Sing_{sk}(m)$ 함수는 비밀키(sk)를 이용하여 메시지 m에 서명한다.

[0066] $Verify_{pk}(m, \sigma)$ 함수는 메시지 m을 공개키(pk)를 이용하여 서명값(σ)과 비교하여 동일하면 1을 반환하고, 그렇지 않다면 0을 반환한다.

[0067] 도 6은 본 발명의 일 실시예에 따른MPTCP의 서브플로우 보안 연결 방법을 설명하는 도면이다.

[0068] 호스트 C는 호스트 A의 키Key_B와 호스트 B의 키Key_B를 초기 연결을 위한 3-웨이 핸드세이크 과정에서 도청하여

알고 있다고 가정한다.

[0069] 상기에서 정의한 함수들을 이용하여 MPTCP의 서브플로우 보안 연결 방법을 설명하면, 적합한 HMAC을 생성할 수 있는 호스트 C는 자신의 주소가 담긴 ADD_ADDR2를 도청한 호스트 키(Key_A, key_B)를 바탕으로 자신의 주소인 IP C1에 대해 키(K)를 이용하여 HMAC값을 생성한다.

[0070] 호스트 C는 HMAC 값을 ADD_ADDR 옵션에 추가하여 ADD_ADDR2를 생성하여 호스트 B에게 전송한다. ADD_ADDR2 옵션을 수신한 호스트 B는 이를 검증하지 않고, 신뢰하는 호스트인 호스트 A에게 ADD_ADDR2의 플래그 값을 1로 설정함으로써 확인 요청 메시지를 보내게 된다. 여기서, 호스트 B는 클라우드 서버(200)이고, 호스트 A는 사용자 단말이 될 수 있다.

[0071] 확인 요청을 받은 호스트 A는 자신이 보유하고 있는 IP 주소인 IP A_n과 수신한 ADD_ADDR2 옵션의 IP 주소를 비교하여 일치한다면, 호스트 A의 비밀 키(skA)로 Ack에 대해 서명값을 생성한 후에 추가하여 ACK'를 생성한다.

[0072] 호스트 A는 전자 서명된 ACK ' 메시지를 호스트 B에게 전송하고, 호스트 B는 호스트 A의 공개키(pkA)로 Ack' 에 대해 검증 후DP 위조되지 않았다면, Syn+MP_JOIN을 호스트 A에게 전송함으로써 서브플로우 연결을 시도한다.

[0073] 만일, 호스트 A가 수신한 확인 요청 메시지의 주소와 자신의 주소인 IP A_n이 일치 하지 않는다면, 호스트 B에게 전자 서명된 경고 메시지를 전송한다. 이때, 호스트 A는 확인 요청 메시지의 플래그 값을 0으로 다시 설정 후에 자신의 비밀키(skA)로 서명하여, 이를 덧붙여 경고 메시지를 생성한 후 호스트 B에게 전송한다. 이를 수신한 호스트 B는 호스트 A의 공개키(pkA)로 메시지를 검증하여 호스트 C로부터 수신한 ADD_ADDR2 옵션을 무시한다.

[0074] 한편, 호스트 C는 다음과 같은 후속 공격도 가능하다. 호스트 C는 상기한 서브플로우 보안 연결 과정을 도청하고 있으므로, ACK' 혹은 경고 메시지를 누락시키고 자신의 주소를 삽입한 SYN + MP_JOIN 메시지를 호스트 B에게 전송할 수 있다. 이러한 후속 공격을 방어하기 위해서, 전자서명을 적용한다. 호스트 A는 자신의 비밀 키로 ACK와 경고 메시지에 대한 서명을 생성한 후 서명값이 추가된 ACK와 경고 메시지를 호스 트B로 전송한다. 이를 통해 호스트 B는 자신이 수신한 메시지가 적합한 호스트인 호스트 A로부터 온 것임을 확신할 수 있다. 즉, 호스트 C는 호스트 A의 비밀 키를 올바르게 추측하지 않는 이상 호스트 A를 가장하여 가짜 서브플로우를 생성할 수 없게 된다.

[0075] 본 발명은 보안성 증명을 위해서 정의1과 정의2에서 정의된 함수들을 이용하여 정의3에서 공격자 모델을 정형적으로 정의하고, 도 5에 도시된 공격자 모델에서 공격자가 ADD_ADDR2공격을 수행하여도 공격이 성공할 확률이 매우 적어 보안성을 우수함을 알 수 있다.

[0076] 정의 3에서는 공격자 모델과 공격 방법을 정의한다. 정의된 공격자 모델은 ADD_ADDR2 공격을 수행하기 위해 MPTCP 오라클에 ADD_ADDR2 공격을 위한 네트워크 상에서 교환되는 모든 정보를 요청할 수 있다.

[0077] 오라클은 (A_{ID}, A_{IP}, P_t, H, pk)=Oracl()로 정의된다. 즉, 오라클은 ADD_ADDR2 옵션을 만들기 위한 호스트의 ID, IP, 포트번호, HMAC 값 및 공개키(pk)를 반환한다

[0078] 정의 3 (공격자 모델)에서는 MPTCP 오라클로부터 ADD_ADDR2 공격을 위한 정보를 받아 공격을 수행한다. MPTCP 오라클은 다음과 같이 정의된다.

[0079] (A_{ID}, A_{IP}, P_t, H, pk)=Oracl() 함수는 ADD_ADDR2 옵션 구성을 위한 정보를 반환한다.

[0080] 위에서 정의한 함수를 바탕으로 공격 모델은 $AD\_Atk_{A,\Pi}$ (n)으로 정의한다.

[0081] 본 발명은 추가적인 서브플로우 연결 방법(Π) 을 따르는 ADD_ADDR2 공격을 정의하면, ADD_ADDR2 공격, $AD\_Atk_{A,\Pi}$ (n)은 다음과 같이 정의된다.

[0082] 1. Kgen(n)을 이용하여 공개키와 비밀키 쌍 (pk, sk)를 만든다. 이 때의 비밀키(sk)는 공격자 A에게 공개되지 않는다.

[0083] 2. 공격자 A는 Oracle()을 이용하여 MPTCP 오라클에 ADD_ADDR2 공격을 위한 정보를 얻는다.

[0084] 3. MPTCP오라클을 통해 얻은 정보 (A_{ID}, A_{IP}, P_t, H, pk)를 이용하여 메시지 m을 생성한다.

[0085] 4. 공격자 A가 메시지의 플래그값 F가 f로 설정해야 할 경우, 공격자 ASMS $Flagset_A$(m1, F, f)를 수행한다.

[0086] 5. 메시지 m의 서명 값(σ)이 생성되어야 한다면, 공격자 A는 응답받은 정보로부터 비밀키(sk')를 생성하고, $\sigma=Sign_{sk'}(m)$을 수행한다.

[0087] 6. 공격자 A는 결과를 확인하기 위해 $Verify_{pk}(m,\sigma)$ 를 수행한다.

[0088] 공격자 A는 다항 시간 내에서만 연산을 할 수 있는 능력을 가지고, 서브플로우에 대한 모든 정보는 오라클에 질의하여 얻을 수 있다. 공격자는 서브플로우에 대한 정보와 HMAC 생성을 위한 키를 오라클에 질의하여 메시지를 구성할 수 있다.

[0089] 신뢰할 수 있는 호스트는 공격자 A가 생성한 메시지에 대해서 Check 함수를 사용하고, 그 결과에 대해서 서명을 한다. 하지만, 공격자는 서명에 대한 키는 오라클에 질의하여 얻을 수 없으므로, 결국 공격에 성공하기 위해서는 결과를 서명한 메시지를 추측할 수 밖에 없다. 이때, 신뢰할 수 있는 호스트인 서명자의 키를 추측한다면, 키의 길이가 n일 때 2^n번 키의 비트에 대해서 추측해야 하며, 성공 확률은 $1/2^n$ 이 된다. 따라서, 공격자는 ADD_ADDR2 공격에 대해 성공 확률이 매우 낮아지게 되므로, 본 발명의 일 실시예에 따른 MPTCP의 서브플로우 보안 연결 방법은 보안성을 가진다고 할 수 있다.

[0090] 즉, 클라우드 서버(200)는 수신한 ADD_ADDR2 옵션을 신뢰하는 호스트에게 역방향 확인(Backward Confirmation)을 요청하는데, 현재 연결되어 있는 서브플로우를 통해 호스트에게 ADD_ADDR2 옵션을 전송하였는지를 확인하게 된다. 이러한 MPTCP의 서브플로우 보안 연결 방법은 하나 이상의 네트워크 상 교환되는 모든 정보를 알 수 있는 강력한 공격 모델에서 보안성을 보인다 할 수 있으므로, ADD_ADDR2 공격에 대해 보안이 보장되고 경량화된 서브플로우 연결 방식을 제공할 수 있다.

[0091] 이상에서 설명한 본 발명의 실시예에 따른 MPTCP의 서브플로우 보안 연결 방법은, 컴퓨터에 의해 실행되는 프로그램 모듈과 같은 컴퓨터에 의해 실행 가능한 명령어를 포함하는 기록 매체의 형태로도 구현될 수 있다. 이러한 기록 매체는 컴퓨터 판독 가능 매체를 포함하며, 컴퓨터 판독 가능 매체는 컴퓨터에 의해 액세스될 수 있는 임의의 가용 매체일 수 있고, 휘발성 및 비휘발성 매체, 분리형 및 비분리형 매체를 모두 포함한다. 또한, 컴퓨터 판독가능 매체는 컴퓨터 저장 매체를 포함하며, 컴퓨터 저장 매체는 컴퓨터 판독가능 명령어, 데이터 구조, 프로그램 모듈 또는 기타 데이터와 같은 정보의 저장을 위한 임의의 방법 또는 기술로 구현된 휘발성 및 비휘발성, 분리형 및 비분리형 매체를 모두 포함한다.

[0092] 전술한 본 발명의 설명은 예시를 위한 것이며, 본 발명이 속하는 기술분야의 통상의 지식을 가진 자는 본 발명의 기술적 사상이나 필수적인 특징을 변경하지 않고서 다른 구체적인 형태로 쉽게 변형이 가능하다는 것을 이해할 수 있을 것이다. 그러므로 이상에서 기술한 실시예들은 모든 면에서 예시적인 것이며 한정적이 아닌 것으로 이해해야만 한다. 예를 들어, 단일형으로 설명되어 있는 각 구성 요소는 분산되어 실시될 수도 있으며, 마찬가지로 분산된 것으로 설명되어 있는 구성 요소들도 결합된 형태로 실시될 수 있다.

[0093] 본 발명의 범위는 상기 상세한 설명보다는 후술하는 특허청구범위에 의하여 나타내어지며, 특허청구범위의 의미 및 범위 그리고 그 균등 개념으로부터 도출되는 모든 변경 또는 변형된 형태가 본 발명의 범위에 포함되는 것으로 해석되어야 한다.

부호의 설명

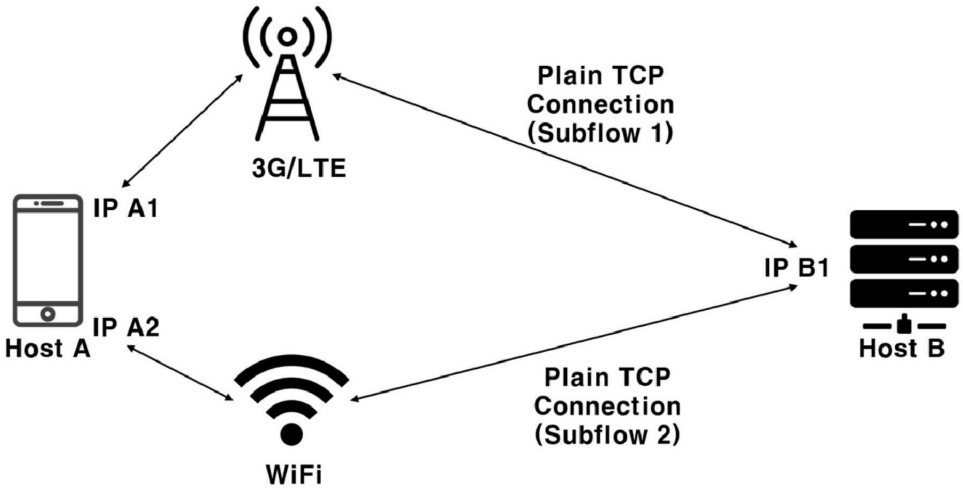
[0094] 100: 호스트

200: 클라우드 서버

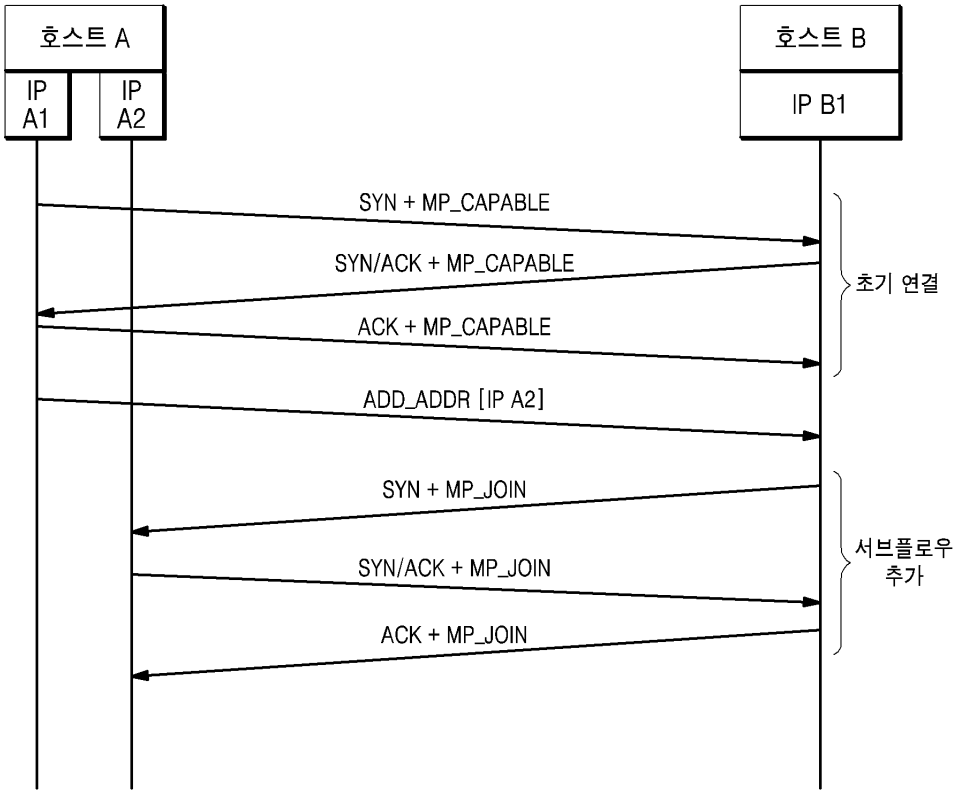
300: 공격자 호스트

도면

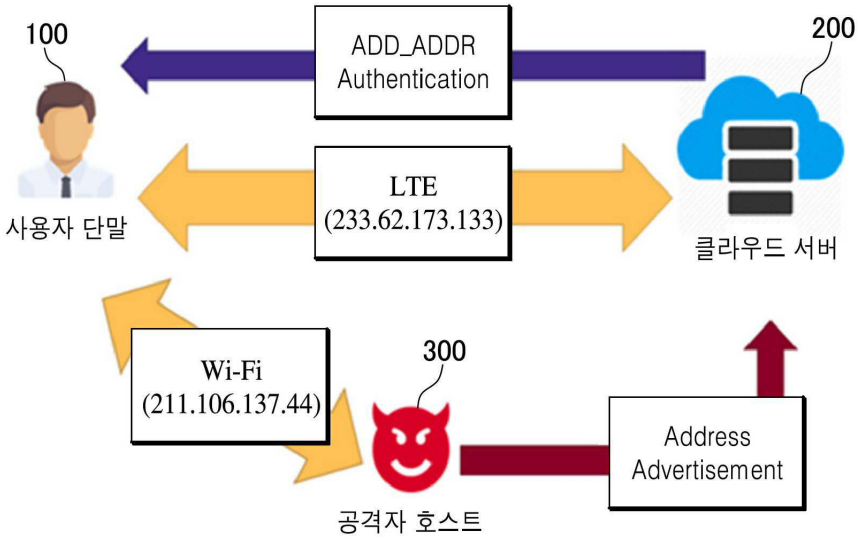
도면1


Plain TCP
Connection
(Subflow 1)
3G/LTE
IP A1
IP B1
IP A2
Host A
Host B
Plain TCP
Connection
(Subflow 2)
WiFi

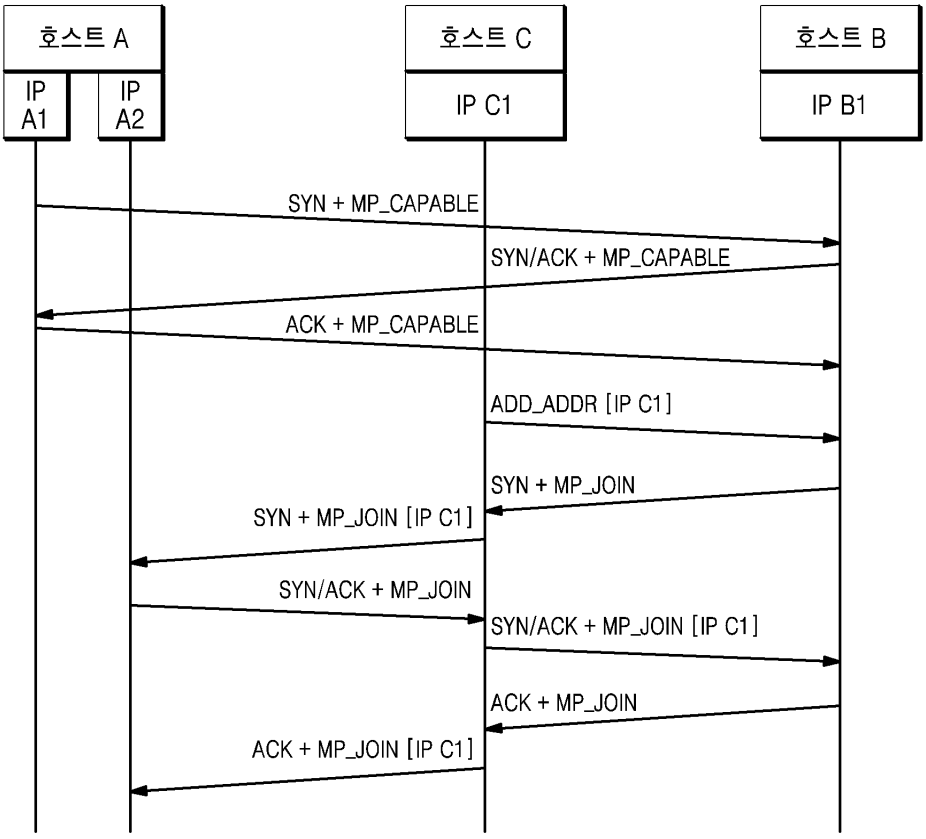
도면2


호스트 A
IP A1
IP A2
호스트 B
IP B1
SYN + MP_CAPABLE
SYN/ACK + MP_CAPABLE
ACK + MP_CAPABLE
초기 연결
ADD_ADDR [IP A2]
SYN + MP_JOIN
SYN/ACK + MP_JOIN
ACK + MP_JOIN
서브플로우
추가

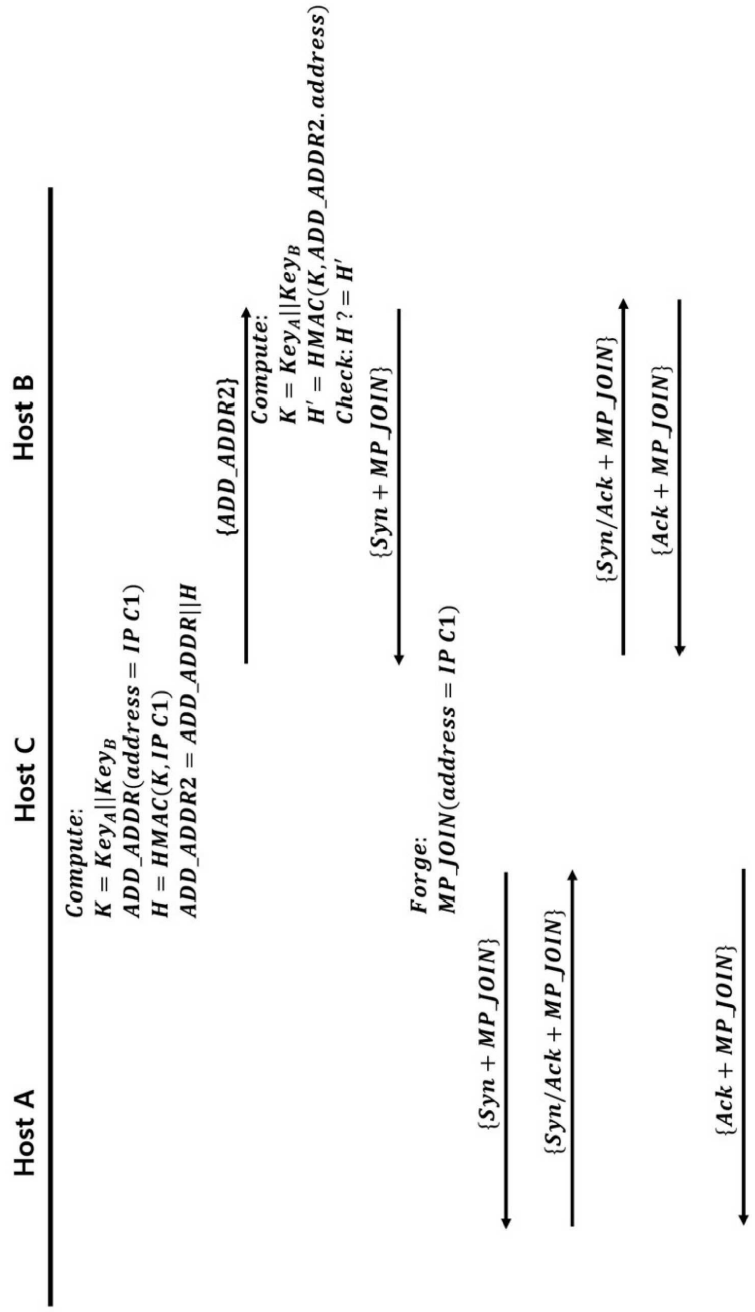
도면3


100
ADD_ADDR
Authentication
200
LTE
(233.62.173.133)
사용자 단말
클라우드 서버
Wi-Fi
(211.106.137.44)
300
Address
Advertisement
공격자 호스트

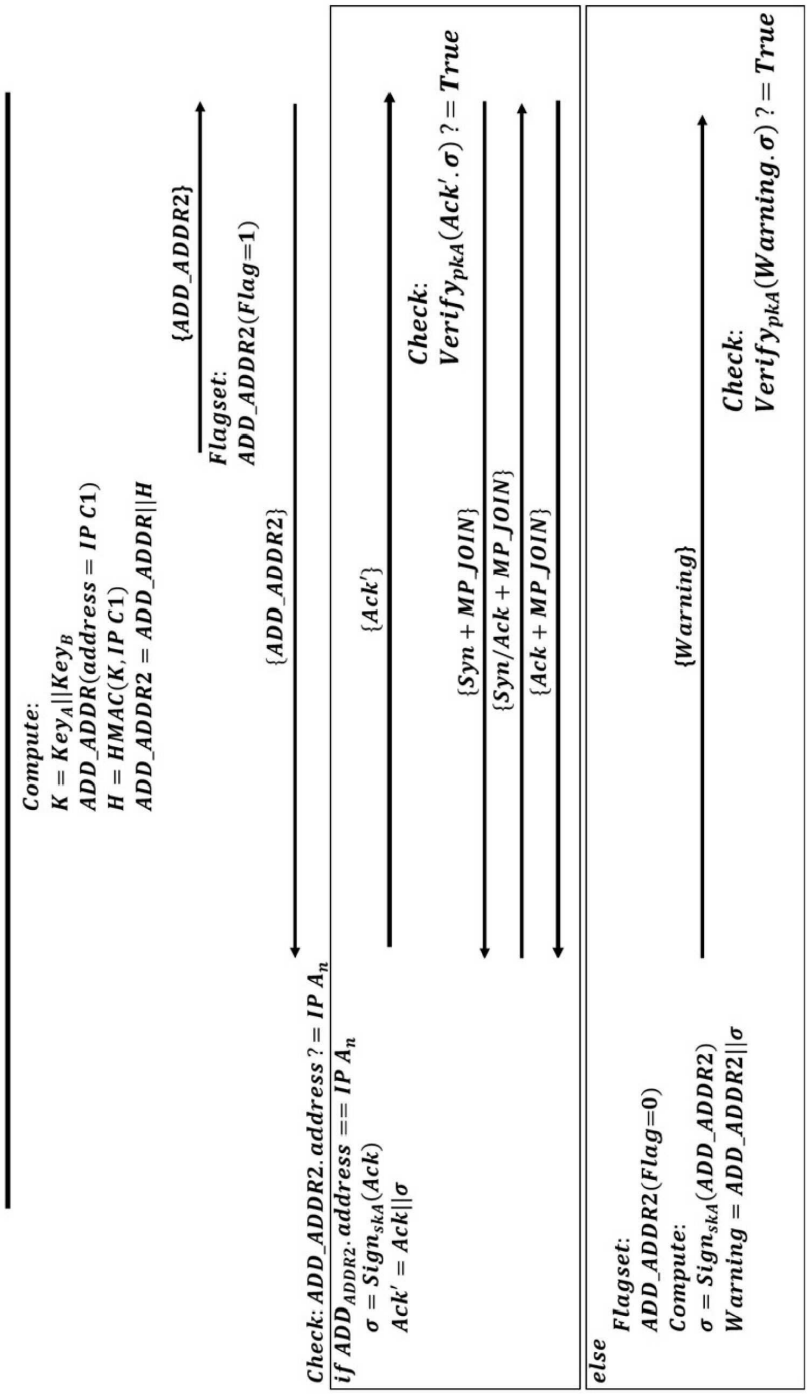
도면4


호스트 A
IP A1
IP A2
호스트 C
IP C1
호스트 B
IP B1
SYN + MP_CAPABLE
SYN/ACK + MP_CAPABLE
ACK + MP_CAPABLE
ADD_ADDR [IP C1]
SYN + MP_JOIN
SYN + MP_JOIN [IP C1]
SYN/ACK + MP_JOIN
SYN/ACK + MP_JOIN [IP C1]
ACK + MP_JOIN
ACK + MP_JOIN [IP C1]

도면5


Host A
Host C
Host B
Compute:
$K = Key_A||Key_B$
$ADD\_ADDR(address = IP\ C1)$
$H = HMAC(K, IP\ C1)$
$ADD\_ADDR2 = ADD\_ADDR||H$
{ADD_ADDR2}
Compute:
$K = Key_A||Key_B$
$H' = HMAC(K, ADD\_ADDR2.address)$
Check: $H ? = H'$
{Syn + MP_JOIN}
Forge:
$MP\_JOIN(address = IP\ C1)$
{Syn + MP_JOIN}
{Syn/Ack + MP_JOIN}
{Syn/Ack + MP_JOIN}
{Ack + MP_JOIN}
{Ack + MP_JOIN}

도면6


Host A
Host C
Host B
Compute:
$K = Key_A||Key_B$
$ADD\_ADDR(address = IP\ C1)$
$H = HMAC(K, IP\ C1)$
$ADD\_ADDR2 = ADD\_ADDR||H$
{ADD_ADDR2}
Flagset:
ADD_ADDR2(Flag=1)
{ADD_ADDR2}
Check: $ADD\_ADDR2.address\ ?= IP\ A_n$
if $ADD_{ADDR2}.address == IP\ A_n$
$\sigma = Sign_{skA}(Ack)$
$Ack' = Ack||\sigma$
{Ack'}
Check:
$Verify_{pkA}(Ack'.\sigma)\ ?= True$
{Syn + MP_JOIN}
{Syn/Ack + MP_JOIN}
{Ack + MP_JOIN}
else
Flagset:
ADD_ADDR2(Flag=0)
Compute:
$\sigma = Sign_{skA}(ADD\_ADDR2)$
$Warning = ADD\_ADDR2||\sigma$
{Warning}
Check:
$Verify_{pkA}(Warning.\sigma)\ ?= True$