(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2014-0095523
(43) 공개일자 2014년08월01일

(51) 국제특허분류(Int. Cl.)
H04W 12/04 (2009.01) *H04W 12/06* (2009.01)
(21) 출원번호 10-2014-7014546
(22) 출원일자(국제) 2011년10월31일
심사청구일자 2014년05월29일
(85) 번역문제출일자 2014년05월29일
(86) 국제출원번호 PCT/FI2011/050953
(87) 국제공개번호 WO 2013/064716
국제공개일자 2013년05월10일

(71) 출원인
노키아 코포레이션
핀란드핀-02150 에스푸 카일알라덴티에 4
(72) 발명자
홀트만스 실케
핀란드 에프아이-01800 클라우칼라 하르카푸론티에 15
레이티넨 페카 요하네스
핀란드 에프아이-00640 헬신키 피쿠마틴티에 11 비
(74) 대리인
제일특허법인

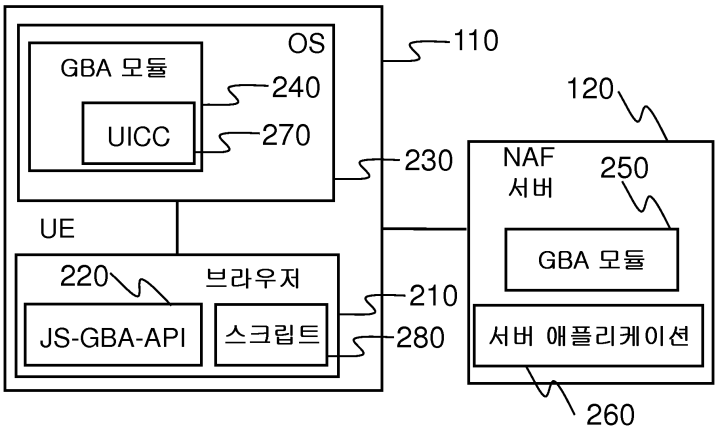
전체 청구항 수 : 총 20 항

(54) 발명의 명칭 **외부 코드에 대한 보안 메커니즘**

(57) 요 약

외부 코드에 대한 보안 메커니즘을 제공하는 방법이 제공되고, 방법은 서버 특정 부트스트래핑 키(Ks_NAF)에 대한 요청을 포함하는 외부 코드를 수신하는 단계를 포함한다. 방법은 서버 식별자(NAF-Id) 및 보안 토큰을 결정하는 단계를 더 포함한다. 더욱이, 방법은 서버 식별자(NAF-Id)에 기초하여 서버 특정 부트스트래핑 키(Ks_NAF)를 생성하는 단계와, 서버 특정 부트스트래핑 키(Ks_NAF) 및 보안 토큰을 이용하여 외부 코드 특정 부트스트래핑 키(Ks_js_NAF)를 생성하는 단계를 포함한다. 방법은 또한 외부 코드의 보안 메커니즘에 대해 외부 코드 특정 부트스트래핑 키(Ks_js_NAF)를 사용하는 단계를 포함한다.

대 표 도 - 도2



특허청구의 범위

청구항 1

외부 코드에 대한 보안 메커니즘을 제공하기 위한 방법으로서,

서버 특정 부트스트래핑 키(Ks_NAF)에 대한 요청을 포함하는 상기 외부 코드를 수신하는 단계와,

서버 식별자(NAF-Id)를 결정하고, 상기 서버 식별자(NAF-Id)에 기초하여 상기 서버 특정 부트스트래핑 키(Ks_NAF)를 생성하는 단계와,

보안 토큰을 결정하는 단계와,

상기 서버 특정 부트스트래핑 키(Ks_NAF) 및 상기 보안 토큰을 이용하여 외부 코드 특정 부트스트래핑 키(Ks_js_NAF)를 생성하는 단계와,

상기 외부 코드의 상기 보안 메커니즘에 대하여 상기 외부 코드 특정 부트스트래핑 키(Ks_js_NAF)를 이용하는 단계를 포함하는

외부 코드에 대한 보안 메커니즘 제공 방법.

청구항 2

제 1 항에 있어서,

제 1 랜덤 챌린지(RAND1) 및 제 2 랜덤 챌린지(RAND2)를 이용하여 상기 보안 토큰을 결정하는 단계를 더 포함하는

외부 코드에 대한 보안 메커니즘 제공 방법.

청구항 3

제 2 항에 있어서,

상기 제 2 랜덤 챌린지(RAND2) 및 상기 외부 코드 특정 부트스트래핑 키(Ks_js_NAF)를 상기 외부 코드 특정 부트스트래핑 키(Ks_js_NAF)의 검증을 위해 애플리케이션 서버로 전송하는 단계를 더 포함하는

외부 코드에 대한 보안 메커니즘 제공 방법.

청구항 4

제 3 항에 있어서,

상기 전송하는 단계는

상기 제 2 랜덤 챌린지(RAND2) 및 상기 외부 코드 특정 부트스트래핑 키(Ks_js_NAF)를 포함하는 응답 외부 코드를 전송하는 단계를 더 포함하는

외부 코드에 대한 보안 메커니즘 제공 방법.

청구항 5

제 1 항에 있어서,

장치의 브라우저 애플리케이션에 의해 애플리케이션 서버로부터 상기 외부 코드를 수신하는 단계와,

상기 브라우저 애플리케이션의 애플리케이션 프로그래밍 인터페이스(JS-GBA-API)에 의해 상기 서버 식별자(NAF-Id) 및 상기 보안 토큰을 결정하는 단계와,

상기 애플리케이션 프로그래밍 인터페이스(JS-GBA-API)에 의해 운영 체제의 부트스트래핑 모듈로부터 상기 서버 특정 부트스트래핑 키(Ks_NAF)를 요청하는 단계와,

상기 애플리케이션 프로그래밍 인터페이스(JS-GBA-API)에 의해 상기 부트스트래핑 모듈로부터 상기 서버 특정 부트스트래핑 키(Ks_NAF)를 수신하는 단계와,

상기 애플리케이션 프로그래밍 인터페이스(JS-GBA-API)에 의해 상기 외부 코드 특정 부트스트래핑 키(Ks_js_NAF)를 생성하는 단계를 더 포함하는

외부 코드에 대한 보안 메커니즘 제공 방법.

청구항 6

제 1 항 내지 제 5 항 중 어느 한 항에 있어서,

장치의 브라우저 애플리케이션과 애플리케이션 서버 사이에 전송 계층 보안(TLS) 터널을 설정하는 단계와,

도메인 이름(FQDN) 및 보안 프로토콜 식별자를 포함하는 상기 서버 식별자(NAF-Id)를 결정하는 단계를 더 포함하는

외부 코드에 대한 보안 메커니즘 제공 방법.

청구항 7

제 6 항에 있어서,

상기 보안 프로토콜 식별자는 전송 계층 보안(TLS)의 암호 슈트(ciphersuite)를 이용하여 형성되는

외부 코드에 대한 보안 메커니즘 제공 방법.

청구항 8

제 1 항 내지 제 7 항 중 어느 한 항에 있어서,

키 도출 함수를 가진 상기 외부 코드 특정 부트스트래핑 키(Ks_js_NAF)를 생성하는 단계를 더 포함하는

외부 코드에 대한 보안 메커니즘 제공 방법.

청구항 9

제 1 항 내지 제 8 항 중 어느 한 항에 있어서,

상기 외부 코드는 자바 스크립트 코드를 포함하는

외부 코드에 대한 보안 메커니즘 제공 방법.

청구항 10

제 1 항에 있어서,

전송 계층 보안(TLS) 마스터 키를 사용하여 상기 보안 토큰을 결정하는 단계를 더 포함하는

외부 코드에 대한 보안 메커니즘 제공 방법.

청구항 11

장치로서,

적어도 하나의 프로세서와,

컴퓨터 프로그램 코드를 포함하는 적어도 하나의 메모리를 포함하되,

상기 적어도 하나의 메모리 및 상기 컴퓨터 프로그램 코드는 상기 적어도 하나의 프로세서와 함께 상기 장치로 하여금 적어도,

서버 특정 부트스트래핑 키(Ks_NAF)에 대한 요청을 포함하는 외부 코드를 수신하고,

서버 식별자(NAF-Id)를 결정하고, 상기 서버 식별자(NAF-Id)에 기초하여 상기 서버 특정 부트스트래핑 키(Ks_NAF)를 생성하고,

보안 토큰을 결정하고,

상기 서버 특정 부트스트래핑 키(Ks_NAF) 및 상기 보안 토큰을 이용하여 외부 코드 특정 부트스트래핑 키(Ks_js_NAF)를 생성하고,

상기 외부 코드의 보안 메커니즘에 대해 상기 외부 코드 특정 부트스트래핑 키(Ks_js_NAF)를 사용하게 하도록 구성되는

장치.

청구항 12

제 11 항에 있어서,

상기 적어도 하나의 메모리 및 상기 컴퓨터 프로그램 코드는 상기 적어도 하나의 프로세서와 함께 상기 장치로 하여금 적어도,

제 1 랜덤 챌린지(RAND1) 및 제 2 랜덤 챌린지(RAND2)를 이용하여 상기 보안 토큰을 결정하게 하도록 더 구성되는

장치.

청구항 13

제 11 항 또는 제 12 항에 있어서,

상기 적어도 하나의 메모리 및 상기 컴퓨터 프로그램 코드는 상기 적어도 하나의 프로세서와 함께 상기 장치로 하여금 적어도,

상기 장치의 브라우저 애플리케이션에 의해 애플리케이션 서버로부터 상기 외부 코드를 수신하고,

상기 브라우저 애플리케이션의 애플리케이션 프로그래밍 인터페이스(JS-GBA-API)에 의해 상기 서버 식별자(NAF-Id)를 결정하고,

상기 애플리케이션 프로그래밍 인터페이스(JS-GBA-API)에 의해 운영 체제의 부트스트래핑 모듈로부터 상기 서버 특정 부트스트래핑 키(Ks_NAF)를 요청하고,

상기 애플리케이션 프로그래밍 인터페이스(JS-GBA-API)에 의해 상기 부트스트래핑 모듈로부터 상기 서버 특정 부트스트래핑 키(Ks_NAF)를 수신하고,

상기 애플리케이션 프로그래밍 인터페이스(JS-GBA-API)에 의해 상기 외부 코드 특정 부트스트래핑 키

(Ks_js_NAF)를 생성하게 하도록 더 구성되는

장치.

청구항 14

제 11 항 내지 제 13 항 중 어느 한 항에 있어서,

상기 적어도 하나의 메모리 및 상기 컴퓨터 프로그램 코드는 상기 적어도 하나의 프로세서와 함께 상기 장치로 하여금 적어도,

전송 계층 보안(TLS) 마스터 키를 이용하여 상기 보안 토큰을 결정하게 하도록 더 구성되는

장치.

청구항 15

컴퓨터 실행가능 프로그램 코드를 포함하는 컴퓨터 판독 가능한 매체 상에 내장된 컴퓨터 프로그램으로서,

상기 컴퓨터 실행가능 프로그램 코드는 장치의 적어도 하나의 프로세서에 의해 실행될 때 상기 장치로 하여금,

서버 특정 부트스트래핑 키(Ks_NAF)에 대한 요청을 포함하는 외부 코드를 수신하고,

서버 식별자(NAF-Id)를 결정하고, 상기 서버 식별자(NAF-Id)에 기초하여 서버 특정 부트스트래핑 키(Ks_NAF)를 생성하고,

보안 토큰을 결정하고,

상기 서버 특정 부트스트래핑 키(Ks_NAF) 및 상기 보안 토큰을 이용하여 외부 코드 특정 부트스트래핑 키(Ks_js_NAF)를 생성하고,

상기 외부 코드의 보안 메커니즘에 대해 외부 코드 특정 부트스트래핑 키(Ks_js_NAF)를 이용하게 하는

컴퓨터 프로그램.

청구항 16

외부 코드에 대한 보안 메커니즘을 제공하기 위한 방법으로서,

서버 특정 부트스트래핑 키(Ks_NAF)에 대한 요청을 포함하는 외부 코드를 송신하는 단계와,

보안 토큰을 결정하는 단계와,

서버 식별자(NAF-Id)를 이용하여 상기 서버 특정 부트스트래핑 키(Ks_NAF)를 생성하는 단계와,

상기 서버 특정 부트스트래핑 키(Ks_NAF) 및 상기 보안 토큰을 이용하여 외부 코드 특정 부트스트래핑 키(Ks_js_NAF)를 생성하는 단계와,

상기 외부 코드의 상기 보안 메커니즘에 대해 상기 외부 코드 특정 부트스트래핑 키(Ks_js_NAF)를 이용하는 단계를 포함하는

외부 코드에 대한 보안 메커니즘 제공 방법.

청구항 17

제 16 항에 있어서,

부트스트래핑 서버 기능(BSF)으로부터 상기 서버 특정 부트스트래핑 키(Ks_NAF)를 요청하는 단계와,

도메인 이름(FQDN) 및 보안 프로토콜 식별자를 포함하는 상기 서버 식별자(NAF-Id)를 결정하는 단계를 더 포함하는

외부 코드에 대한 보안 메커니즘 제공 방법.

청구항 18

제 16 항 또는 제 17 항에 있어서,

외부 코드 특정 부트스트래핑 키(Ks_js_NAF)를 수신하는 단계와,

생성된 외부 코드 특정 부트스트래핑 키(Ks_js_NAF)를 상기 외부 코드의 상기 보안 메커니즘에 대해 수신된 외부 코드 특정 부트스트래핑 키(Ks_js_NAF)와 비교함으로써 상기 외부 코드 특정 부트스트래핑 키(Ks_js_NAF)를 검증하는 단계를 더 포함하는

외부 코드에 대한 보안 메커니즘 제공 방법.

청구항 19

애플리케이션 서버로서,

적어도 하나의 프로세서와,

컴퓨터 프로그램 코드를 포함하는 적어도 하나의 메모리를 포함하고,

적어도 하나의 메모리 및 컴퓨터 프로그램 코드는 적어도 하나의 프로세서와 함께 애플리케이션 서버로 하여금 적어도,

서버 특정 부트스트래핑 키(Ks_NAF)에 대한 요청을 포함하는 외부 코드를 송신하고,

서버 식별자(NAF-Id)를 이용하여 서버 특정 부트스트래핑 키(Ks_NAF)를 생성하고,

보안 토큰을 결정하고,

상기 서버 특정 부트스트래핑 키(Ks_NAF) 및 상기 보안 토큰을 이용하여 외부 코드 특정 부트스트래핑 키(Ks_js_NAF)를 생성하고,

상기 외부 코드의 상기 보안 메커니즘에 대해 상기 외부 코드 특정 부트스트래핑 키(Ks_js_NAF)를 사용하게 하도록 구성되는

애플리케이션 서버.

청구항 20

컴퓨터 실행가능 프로그램 코드를 포함하는 컴퓨터 판독 가능한 매체 상에 내장된 컴퓨터 프로그램으로서,

상기 컴퓨터 실행가능 프로그램 코드는 애플리케이션 서버의 적어도 하나의 프로세서에 의해 실행될 때 상기 애플리케이션 서버로 하여금 적어도,

서버 특정 부트스트래핑 키(Ks_NAF)에 대한 요청을 포함하는 외부 코드를 송신하고,

서버 식별자(NAF-Id)를 이용하여 서버 특정 부트스트래핑 키(Ks_NAF)를 생성하고,

보안 토큰을 결정하고,

상기 서버 특정 부트스트래핑 키(Ks_NAF) 및 상기 보안 토큰을 이용하여 외부 코드 특정 부트스트래핑 키(Ks_js_NAF)를 생성하고,

상기 외부 코드의 상기 보안 메커니즘에 대해 상기 외부 코드 특정 부트스트래핑 키(Ks_js_NAF)를 사용하게 하는

컴퓨터 프로그램.

명 세 서

기 술 분 야

[0001] 본 발명은 일반적으로 애플리케이션 웹 서버에 의해 제공되는 외부 코드에 대한 보안 메커니즘에 관한 것이다. 본 발명은 특히 전적인 것은 아니지만 서버, 폰 브라우저 및 운영 체제가 자바 스크립트 코드와 같은 외부 코드를 통한 브라우저로부터의 셀룰러 기반 인증 정보(credential)의 안전한 사용을 가능하게 할 수 있는 방법에 관한 것이다.

배 경 기 술

[0002] 오늘날, 다양한 텍스트 및 비텍스트 콘텐츠 타입을 제공하는 많은 전세계 웹 페이지(예를 들어 HTML 문서)가 이용 가능하다. 동시에, 통신, 특히 무선 통신 분야는 현재 급진적인 팽창을 겪고 있다. 이러한 기술적 팽창은 누구나 퍼스널 컴퓨터(PC)와 PC 기반의 브라우저를 사용해서 할 수 있는 바와 같이 개인 휴대 정보 단말기(PDA), 랩탑, 셀룰로폰, 태블릿 및 다른 전자 장치와 같은 작은 핸드헬드 전자 디바이스가 웹 서버 또는 데이터베이스와 같은 정보 소스에 연결하도록 해준다. 콘텐츠를 웹으로부터 핸드헬드 장치로 표시하는 수개의 작은 디바이스 클라이언트 브라우저가 이용 가능하다.

[0003] 자바 스크립트 또는 유사한 스크립트 언어로 작성된 HTML 문서와 같은 웹 콘텐츠 내의 스크립트 명령이 사용된다. PC 기반의 브라우저에서 실행된 스크립트 명령은 PC 기반의 브라우저에 이용 가능한 정보 콘텐츠의 일부 또는 모두를 생성할 수 있다.

[0004] 새로운 멀티미디어 가능 이동 단말기(멀티미디어 폰)는 애플리케이션 개발자에 개방형 개발 플랫폼을 제공하여, 독립적인 애플리케이션 개발자가 멀티미디어 환경을 위한 새로운 서비스 및 애플리케이션을 설계하도록 해준다. 그 후, 사용자는 새로운 애플리케이션/서비스를 이의 이동 단말기에 다운로드하여 이동 단말기에서 이용할 수 있다.

발명의 내용

해결하려는 과제

[0005] 따라서, 애플리케이션 웹 서버와 이동 단말기의 보안 관리 모듈의 상호 작용은 전체 보안을 위해 중요하다. 외부 소스로부터 검색되는 자바 스크립트와 같은 외부 코드를 포함하는 웹 콘텐츠에 대해 이동 단말기의 보안 관리 모듈을 이용하기 위한 개선된 솔루션이 필요하다.

과제의 해결 수단

[0006] 본 발명의 제 1 예시적인 양태에 따르면, 외부 코드에 대한 보안 메커니즘을 제공하는 방법이 제공되고, 방법은

[0007] 서버 특정 부트스트래핑 키(Ks_NAF)에 대한 요청을 포함하는 외부 코드를 수신하는 단계와,

[0008] 서버 식별자(NAF-Id)를 결정하고, 서버 식별자(NAF-Id)에 기초하여 서버 특정 부트스트래핑 키(Ks_NAF)를 생성하는 단계와,

[0009] 보안 토큰을 결정하는 단계와,

[0010] 서버 특정 부트스트래핑 키(Ks_NAF) 및 보안 토큰을 이용하여 외부 코드 특정 부트스트래핑 키(Ks_js_NAF)를 생성하는 단계와,

[0011] 외부 코드의 보안 메커니즘에 대해 외부 코드 특정 부트스트래핑 키(Ks_js_NAF)를 사용하는 단계를 포함한다.

[0012] 실시예에서, 방법은 제 1 랜덤 챌린지(random challenge)(RAND1) 및 제 2 랜덤 챌린지(RAND2)를 사용하여 보안 토큰을 결정하는 단계를 더 포함한다. 방법은 제 2 랜덤 챌린지(RAND2) 및 외부 코드 특정 부트스트래핑 키(Ks_js_NAF)를 외부 코드 특정 부트스트래핑 키(Ks_js_NAF)의 검증을 위해 애플리케이션 서버로 전송하는 단계

를 더 포함할 수 있다. 제 2 랜덤 챌린지(RAND2) 및 외부 코드 특정 부트스트래핑 키(Ks_js_NAF)를 포함하는 응답 외부 코드가 전송될 수 있다.

[0013] 실시예에서, 방법은

[0014] 장치의 브라우저 애플리케이션에 의해 애플리케이션 서버로부터 외부 코드를 수신하는 단계와,

[0015] 브라우저 애플리케이션의 애플리케이션 프로그래밍 인터페이스(JS-GBA-API)에 의해 서버 식별자(NAF-Id) 및 제 2 랜덤 챌린지(RAND2)를 결정하는 단계와,

[0016] 애플리케이션 프로그래밍 인터페이스(JS-GBA-API)에 의해 운영 체제의 부트스트래핑 모듈로부터 서버 특정 부트스트래핑 키(Ks_NAF)를 요청하는 단계와,

[0017] 애플리케이션 프로그래밍 인터페이스(JS-GBA-API)에 의해 부트스트래핑 모듈으로부터 서버 특정 부트스트래핑 키(Ks_NAF)를 수신하는 단계와,

[0018] 애플리케이션 프로그래밍 인터페이스(JS-GBA-API)에 의해 외부 코드 특정 부트스트래핑 키 (Ks_js_NAF)를 생성하는 단계를 더 포함한다.

[0019] 전송 계층 보안(TLS) 터널은 장치의 브라우저 애플리케이션과 애플리케이션 서버 사이에 설정될 수 있다. 서버 식별자(NAF-Id)는 도메인 이름(FQDN) 및 보안 프로토콜 식별자를 포함함으로써 결정될 수 있다. 보안 프로토콜 식별자는 전송 계층 보안(TLS)의 암호 슈트(ciphersuite)를 사용하여 형성될 수 있다.

[0020] 실시예에서, 방법은 키 도출 함수를 가진 외부 코드 특정 부트스트래핑 키(Ks_js_NAF)를 생성하는 단계를 더 포함한다. 외부 코드는 자바 스크립트 코드를 포함할 수 있다.

[0021] 실시예에서, 방법은 전송 계층 보안(TLS) 마스터 키를 사용하여 보안 토큰을 결정하는 단계를 더 포함한다.

[0022] 본 발명의 제 2 예시적인 양태에 따르면, 장치가 제공되고, 장치는

[0023] 적어도 하나의 프로세서와,

[0024] 컴퓨터 프로그램 코드를 포함하는 적어도 하나의 메모리를 포함하고,

[0025] 적어도 하나의 메모리 및 컴퓨터 프로그램 코드는 적어도 하나의 프로세서를 사용하여 장치가 적어도

[0026] 서버 특정 부트스트래핑 키(Ks_NAF)에 대한 요청을 포함하는 외부 코드를 수신하고,

[0027] 서버 식별자(NAF-Id)를 결정하고, 서버 식별자(NAF-Id)에 기초하여 서버 특정 부트스트래핑 키(Ks_NAF)를 생성하고,

[0028] 보안 토큰을 결정하고,

[0029] 서버 특정 부트스트래핑 키(Ks_NAF) 및 보안 토큰을 이용하여 외부 코드 특정 부트스트래핑 키(Ks_js_NAF)를 생성하고,

[0030] 외부 코드의 보안 메커니즘에 대해 외부 코드 특정 부트스트래핑 키(Ks_js_NAF)를 사용하게 하도록 구성된다.

[0031] 보안 토큰은 제 1 랜덤 챌린지(RAND1) 및 제 2 랜덤 챌린지(RAND2)를 사용하여 결정될 수 있다.

[0032] 실시예에서, 적어도 하나의 메모리 및 컴퓨터 프로그램 코드는 적어도 하나의 프로세서를 사용하여 장치가 적어도

[0033] 장치의 브라우저 애플리케이션에 의해 애플리케이션 서버로부터 외부 코드를 수신하고,

[0034] 브라우저 애플리케이션의 애플리케이션 프로그래밍 인터페이스(JS-GBA-API)에 의해 서버 식별자(NAF-Id)를 결정하고,

[0035] 애플리케이션 프로그래밍 인터페이스(JS-GBA-API)에 의해 운영 체제의 부트스트래핑 모듈로부터 서버 특정 부트스트래핑 키(Ks_NAF)를 요청하고,

[0036] 애플리케이션 프로그래밍 인터페이스(JS-GBA-API)에 의해 부트스트래핑 모듈으로부터 서버 특정 부트스트래핑 키(Ks_NAF)를 수신하고,

[0037] 애플리케이션 프로그래밍 인터페이스(JS-GBA-API)에 의해 외부 코드 특정 부트스트래핑 키 (Ks_js_NAF)를 생성

하게 하도록 더 구성된다.

[0038] 서버 식별자(NAF-Id)는 도메인 이름(FQDN) 및 보안 프로토콜 식별자를 포함 함으로써 결정될 수 있다.

[0039] 실시예에서, 보안 토큰은 전송 계층 보안(TLS) 마스터 키를 사용하여 결정될 수 있다.

[0040] 본 발명의 제 3 예시적인 양태에 따르면, 컴퓨터 실행 가능 프로그램 코드를 포함하는 컴퓨터 판독 가능한 매체 상에 내장된 컴퓨터 프로그램이 제공되고, 이러한 컴퓨터 실행 가능 프로그램 코드는 장치의 적어도 하나의 프로세서에 의해 실행될 때 장치가

[0041] 서버 특정 부트스트래핑 키(Ks_NAF)에 대한 요청을 포함하는 외부 코드를 수신하고,

[0042] 서버 식별자(NAF-Id)를 결정하고, 서버 식별자(NAF-Id)에 기초하여 서버 특정 부트스트래핑 키(Ks_NAF)를 생성하고,

[0043] 보안 토큰을 결정하고,

[0044] 서버 특정 부트스트래핑 키(Ks_NAF) 및 보안 토큰을 이용하여 외부 코드 특정 부트스트래핑 키(Ks_js_NAF)를 생성하고,

[0045] 외부 코드의 보안 메커니즘에 대해 외부 코드 특정 부트스트래핑 키(Ks_js_NAF)를 사용하게 한다.

[0046] 본 발명의 제 4 예시적인 양태에 따르면, 외부 코드에 대한 보안 메커니즘을 제공하는 방법이 제공되고, 방법은

[0047] 서버 특정 부트스트래핑 키(Ks_NAF)에 대한 요청을 포함하는 외부 코드를 송신하는 단계와,

[0048] 보안 토큰을 결정하는 단계와,

[0049] 서버 식별자(NAF-Id)를 이용하여 서버 특정 부트스트래핑 키(Ks_NAF)를 생성하는 단계와,

[0050] 서버 특정 부트스트래핑 키(Ks_NAF) 및 보안 토큰을 이용하여 외부 코드 특정 부트스트래핑 키(Ks_js_NAF)를 생성하는 단계와,

[0051] 외부 코드의 보안 메커니즘에 대해 외부 코드 특정 부트스트래핑 키(Ks_js_NAF)를 사용하는 단계를 포함한다.

[0052] 실시예에서, 방법은

[0053] 부트스트래핑 서버 기능(BSF)으로부터 서버 특정 부트스트래핑 키(Ks_NAF)를 요청하는 단계와,

[0054] 도메인 이름(FQDN) 및 보안 프로토콜 식별자를 포함하는 서버 식별자(NAF-Id)를 결정하는 단계를 더 포함한다.

[0055] 실시예에서, 방법은

[0056] 외부 코드 특정 부트스트래핑 키(Ks_js_NAF)를 수신하는 단계와,

[0057] 외부 코드의 보안 메커니즘에 대한 수신된 외부 코드 특정 부트스트래핑 키(Ks_js_NAF)와 비교함으로써 생성된 외부 코드 특정 부트스트래핑 키(Ks_js_NAF)를 검증하는 단계를 더 포함한다.

[0058] 본 발명의 제 5 예시적인 양태에 따르면, 애플리케이션 서버가 제공되고, 애플리케이션 서버는

[0059] 적어도 하나의 프로세서와,

[0060] 컴퓨터 프로그램 코드를 포함하는 적어도 하나의 메모리를 포함하고,

[0061] 적어도 하나의 메모리 및 컴퓨터 프로그램 코드는 적어도 하나의 프로세서를 사용하여 애플리케이션 서버가 적어도

[0062] 서버 특정 부트스트래핑 키(Ks_NAF)에 대한 요청을 포함하는 외부 코드를 송신하고,

[0063] 서버 식별자(NAF-Id)를 이용하여 서버 특정 부트스트래핑 키(Ks_NAF)를 생성하고,

[0064] 보안 토큰을 결정하고,

[0065] 서버 특정 부트스트래핑 키(Ks_NAF) 및 보안 토큰을 이용하여 외부 코드 특정 부트스트래핑 키(Ks_js_NAF)를 생성하고,

[0066] 외부 코드의 보안 메커니즘에 대해 외부 코드 특정 부트스트래핑 키(Ks_js_NAF)를 사용하게 하도록 구성된다.

[0067] 실시예에서, 적어도 하나의 메모리 및 컴퓨터 프로그램 코드는 적어도 하나의 프로세서를 사용하여 애플리케이션 서버가 적어도

[0068] 부트스트래핑 서버 기능(BSF)으로부터 요청함으로써 서버 특정 부트스트래핑 키(Ks_NAF)를 생성하게 하도록 더 구성된다.

[0069] 본 발명의 제 6 예시적인 양태에 따르면, 컴퓨터 실행가능 프로그램 코드를 포함하는 컴퓨터 판독 가능한 매체상에 내장된 컴퓨터 프로그램이 제공되고,

[0070] 이러한 컴퓨터 실행가능 프로그램 코드는 애플리케이션 서버의 적어도 하나의 프로세서에 의해 실행될 때 애플리케이션 서버가

[0071] 서버 특정 부트스트래핑 키(Ks_NAF)에 대한 요청을 포함하는 외부 코드를 송신하고,

[0072] 서버 식별자(NAF-Id)를 이용하여 서버 특정 부트스트래핑 키(Ks_NAF)를 생성하고,

[0073] 보안 토큰을 결정하고,

[0074] 서버 특정 부트스트래핑 키(Ks_NAF) 및 보안 토큰을 이용하여 외부 코드 특정 부트스트래핑 키(Ks_js_NAF)를 생성하고,

[0075] 외부 코드의 보안 메커니즘에 대해 외부 코드 특정 부트스트래핑 키(Ks_js_NAF)를 사용하게 한다.

[0076] 상술한 임의의 메모리 매체는 데이터 디스크 또는 디스켓, 광 저장소, 자기 저장소, 홀로그래픽 저장소, 광 자기 저장소, 상 변화 메모리, 저항성 랜덤 액세스 메모리, 자기 랜덤 액세스 메모리, 고체 전해질 메모리, 강유전성 랜덤 액세스 메모리, 유기 메모리 또는 폴리머 메모리와 같은 디지털 데이터 저장소를 포함할 수 있다. 메모리 매체는 메모리를 저장하는 것과 다른 실질적인 기능을 갖지 않은 장치 내에 형성될 수 있거나, 컴퓨터, 칩 세트 및 전자 장치의 서브 어셈블리를 포함하지만, 이에 제한되지 않는 다른 기능을 가진 장치의 부분으로서 형성될 수 있다.

발명의 효과

[0077] 본 발명의 여러 구속력없는(non-binding) 예시적인 양태 및 실시예가 위에서 설명되었다. 위의 실시예는 본 발명의 구현에서 활용될 수 있는 선택된 양태 또는 단계를 설명하기 위해서만 이용된다. 일부 실시예는 단지 본 발명의 어떤 예시적인 양태를 참조하여 제공될 수 있다. 상응하는 실시예는 또한 다른 예시적인 양태에 적용할 수 있다는 것이 이해되어야 한다.

도면의 간단한 설명

[0078] 본 발명은 첨부된 도면을 참조하여 단지 예로서 설명될 것이다.

도 1은 본 발명의 다양한 실시예가 적용될 수 있는 시스템 아키텍처의 일부 상세를 도시한다.

도 2는 본 발명의 다양한 실시예가 적용될 수 있는 시스템 요소의 일부 상세를 도시한다.

도 3은 본 발명의 실시예에 따른 메시지 다이어그램을 도시한다.

도 4는 본 발명의 다양한 실시예가 적용될 수 있는 애플리케이션 서버의 예시적인 블록도를 도시한다.

도 5는 본 발명의 다양한 실시예가 적용될 수 있는 사용자 장치의 예시적인 블록도를 도시한다.

도 6은 본 발명의 예시적인 실시예에 따라 사용자 장치에서의 동작을 보여주는 흐름도를 도시한다.

도 7은 본 발명의 예시적인 실시예에 따라 애플리케이션 서버에서의 동작을 보여주는 흐름도를 도시한다.

발명을 실시하기 위한 구체적인 내용

[0079] 다음의 설명에서, 동일한 부호는 동일한 요소를 나타낸다.

[0080] 본 발명의 다양한 실시예는 피어(peer) 인증 및 통신 보안을 위해 3GPP에서 정의된 GAA(Generic Authentication Architecture) 및 GBA(Generic Bootstrapping Architecture)의 특징을 이용한다. GAA/GBA의

변형은 OMA(Open Mobile Alliance) 및 CableLabs에 의해 표준화되어 있다. GAA/GBA는 3GPP에 대한 모바일 알고리즘 AKA(Authentication and Key Agreement)에 기초한다. GAA/GBA 절차의 원래의 목적은 사용자 장비 또는 가입자를 인증하기 위한 것이다. 이제 본 발명의 다양한 실시예에서, GAA/GBA는 애플리케이션 서버, 이동 단말기의 브라우저 및 이동 단말기의 운영 체제 사이의 보안을 개선하기 위해 사용된다. 실시예에서, 셀룰러 기반의 인증 정보의 안전한 사용은 예를 들어 웹 페이지 내에서 서버로부터 사용자 장치로 다운로드되는 외부 코드를 통해 브라우저로부터 가능하게 된다. 외부 코드는 예를 들어 자바 스크립트 코드를 포함할 수 있다.

[0081] 애플리케이션 서버를 인증하기 위해 GAA/GBA의 인증 메카니즘을 이용함으로써 달성될 수 있는 이점은 필요할 수도 있는 값비싼 PKI(Public Key Infrastructure)의 사용을 회피할 수 있다는 것이다. GAA/GBA는 예를 들어 모바일 TV 및 프레즌스(Mobile TV and presence)에 사용되는 다목적 인에이블러(multipurpose enabler)이다. 이러한 기존의 메커니즘 및 관련된 인프라를 이용함으로써, 관리 비용 및 제조될 필요가 있는 투자량이 감소될 수 있다는 이점을 달성할 수 있다.

[0082] 다음의 설명에서, 사용자 장치, 애플리케이션 서버, 부트스트래핑 서버 및 일반 인증 절차와 같은 용어는 본 발명의 다양한 실시예에 관한 다양한 요소/메커니즘을 참조하는 데 사용된다. 이러한 명시적인 요소/메커니즘 이외에 또는 대신에, 비슷한 기능을 제공하는 일부 다른 요소/메커니즘이 이용될 수 있다는 것이 주목되어야 한다.

[0083] 본 발명의 다양한 실시예에서, 애플리케이션 서버는 웹 서비스를 사용자에게 제공하는 웹 서버일 수 있다. 애플리케이션 서버는 또한 네트워크 운영자에 의해 신뢰받지 못할 수 있고, 네트워크 애플리케이션 기능(NAF)을 포함할 수 있다.

[0084] 도 1은 본 발명의 다양한 실시예가 적용될 수 있는 시스템 아키텍처(100)의 일부 상세를 도시한다. 시스템은 사용자 장비(UE)(110), 및 웹 서비스를 제공하는 애플리케이션 서버(120)와 같은 사용자 장치를 포함한다. 부가적으로, 시스템은 홈 가입자 서버(HSS) 또는 홈 위치 레지스터(HLR)와 같이 부트스트래핑 서버 기능(BSF)(130) 및 가입자 데이터베이스(140)를 포함한다. 장치(110)는 부트스트래핑 서버 기능(BSF)과 협력하도록 구성된 GBA(Generic Bootstrapping Architecture) 기능 블록(150), 및 애플리케이션 서버(120)와 협력하도록 구성된 네트워크 애플리케이션 기능(NAF) 클라이언트(160)를 더 포함한다. 네트워크 애플리케이션 기능(NAF) 클라이언트는 예를 들어 브라우저를 포함할 수 있다. GAA/GBA에서, 애플리케이션 서버는 네트워크 애플리케이션 기능(NAF)으로 지칭될 수 있다. 부트스트래핑 서버 기능(BSF)(130)과 가입자 데이터베이스(140) 사이에서 GAA/GBA에 정의된 Zh 인터페이스, 부트스트래핑 서버 기능(BSF)(130)과 애플리케이션 서버(120) 사이의 Zn 인터페이스, 및 부트스트래핑 서버 기능(BSF)(130)과 사용자 장치(110) 사이의 Ub 인터페이스가 있다. 부가적으로, HLR이 배치될 경우에 부트스트래핑 서버 기능(BSF)(130)과 HLR(140) 사이에 Zh' 인터페이스가 있을 수 있다. 부가적으로, 가입자 데이터베이스 정보를 부트스트래핑 서버 기능(BSF)에 제공하는 SLF(Subscription Locator Function)에 대한 인터페이스가 있을 수 있다.

[0085] 애플리케이션 서버(120)는 부트스트래핑 서버 기능(BSF)(130) 및 가입자 데이터베이스(140)에 비해 상이한 당사자에 의해 관리될 수 있거나, (통상적으로 해당 통신 네트워크의 운영자인) 동일한 당사자에 의해 관리될 수 있다.

[0086] 시스템(100)은 일반적으로 또한 다양한 다른 요소를 포함할 수 있지만, 명료성을 위해 본 명세서에 도시되지 않는다는 것이 주목되어야 한다.

[0087] 실시예에서, 일반적 부트스트래핑 서버 기능(BSF)(130) 및 사용자 장비(UE)(110)는 AKA(Authentication and Key Agreement) 프로토콜을 이용하여 상호 인증하며, 이후 사용자 장비와 서버(120)의 네트워크 애플리케이션 기능(NAF) 사이에 적용되는 키에 동의할 것이다. 네트워크 애플리케이션 기능(NAF)은 서비스 제공 서버(120)에 위치된 기능적 모듈이다. 대안적으로, 레거시 스마트 카드가 포함되면, 전송 계층 보안(TLS) 및 레거시 인증이 사용될 수 있다. 일반적 부트스트래핑 아키텍처(GBA)는 또한 하이퍼텍스트 전송 프로토콜(HTTP) 다이제스트 또는 세션 개시 프로토콜(SIP) 다이제스트와 같은 다른 인증 메커니즘을 이용할 수 있다. 서버(120)의 네트워크 애플리케이션 기능(NAF) 모듈의 주요 기능은 서비스/사용자 관리(예를 들어, 서비스 가입 및 미가입) 및 서비스 키 관리(예를 들어, 서비스 키 생성 및 전송)이다. 부트스트래핑 서버 기능(BSF)(130)은 키 도출 절차를 이용함으로써 서버(120)의 특정 네트워크 애플리케이션 기능(NAF)에 대한 키 재료(key material)의 적용 가능성을 제한한다. 키 도출 절차는 키 재료의 수명 동안 다수의 네트워크 애플리케이션 기능(NAF)으로 이용될 수 있다. 키 재료의 수명은 부트스트래핑 서버 기능(BSF)(130)의 로컬 정책에 따라 설정된다. 부트스트래핑 서버 기능(BSF)(130)은 홈 가입자 시스템(HSS)(140)으로부터 어떤 필요한 인증 정보, 보안 정보 및 가입자 프로파일 정보

를 인출하도록 허용된다. 레거시 네트워크에서, 부트스트래핑 서버 기능(BSF)(130)은 홈 가입자 시스템(HSS)(140) 대신에 홈 위치 레지스터(HLR)와 상호 작용할 수 있다.

[0088] 예를 들어 운영자에 의해, 예를 들어 자바 스크립트처럼 스크립트 코드와 같은 외부 코드로부터 GBA 사용에 관한 관심사(concern)가 있을 수 있다. 외부 코드는 사용자 장치로 다운로드될 수 있고, GBA 모듈의 비밀이 있는 그대로 웹 서버로 전송되는 관심사가 있을 수 있다.

[0089] 도 2는 본 발명의 다양한 실시예가 적용될 수 있는 시스템 요소의 일부 상세를 도시한다. 외부 코드는 장치로 다운로드되고, 잠재적으로 사용되거나 국부적으로 실행되는 어떤 코드를 포함할 수 있다. 외부 코드는 예를 들어 브라우저 또는 위젯과 같이 설치된 애플리케이션에서 실행될 수 있다. 외부 코드의 일례로서 자바 스크립트 코드가 있다. 단순화를 위해, 다음의 예시적인 실시예는 자바 스크립트를 이용하여 설명되지만, 실시예는 자바 스크립트로 제한되지 않으며, 어떤 외부 코드가 적용될 수 있다.

[0090] 자바 스크립트는 사용자 장비(UE)(110)에서 처리된 클라이언트측 자바 스크립트의 형식으로 사용될 수 있다. 자바 스크립트(280)의 실행은 향상된 사용자 인터페이스 및 동적 웹 사이트를 제공하기 위해 웹 브라우저(210)의 부분으로서 구현될 수 있다. 이것은 호스트 환경 내에서 계산 객체로의 프로그램 방식(programmatic)의 액세스를 가능하게 한다. 자바 스크립트(280)는 또한 예를 들어 문서, 사이트 특정 브라우저 및 데스크탑 위젯에서 애플리케이션 외부 웹 페이지에 사용될 수 있다. 자바 스크립트는 또한 서버측 웹 애플리케이션에 사용된다. 애플리케이션 프로그래밍 인터페이스(API)는 소프트웨어 프로그램이 서로 통신하도록 따를 수 있는 규칙('코드') 및 사양의 특정 세트이다. API는 서로 다른 소프트웨어 프로그램 사이의 인터페이스 역할을 하고, 이의 상호 작용을 용이하게 한다. 자바 스크립트(280)의 경우, GBA API가 생성될 수 있고, 도 2에서 JS-GBA-API(220)로 명명된다. 사용자 장비(UE)(110)의 운영 체제(OS)(230)는 사용자 장비(UE)(110)의 보안 관리를 책임지는 GBA 모듈(240)을 포함할 수 있다. 사용자 장비(UE)(110)는 또한 셀룰러 네트워크에서 이동 단말기에 사용되는 스마트 카드인 범용 집적 회로 카드(ULCC)(270)를 포함한다. 범용 집적 회로 카드(ULCC)(270)는 모든 종류의 개인 데이터의 무결성 및 보안을 보장하며, 그것은 통상적으로 애플리케이션을 포함한다. ULCC 스마트 카드는 또한 CPU, ROM, RAM, EEPROM 및 I/O 회로를 포함할 수 있다.

[0091] 사용자 장비(UE)(110)의 브라우저(210)는 예를 들어 웹 콘텐츠를 위한 애플리케이션 서비스 서버로서 동작하는 네트워크 애플리케이션 기능(NAF) 서버(120)와 통신할 수 있다. 네트워크 애플리케이션 기능(NAF) 서버(120)는 예를 들어 GBA NAF 모듈(250) 및 서버 애플리케이션(260)을 포함할 수 있다.

[0092] 실시예에서, 애플리케이션 웹 서버와의 사용자 장치 보안 관리 모듈(OS의 부분인 GBA 모듈)의 상호 작용이 제공된다. 보안 메커니즘은 외부 소스(120)로부터 나오는 자바 스크립트(280)를 가진 브라우저(210)로부터의 보안 관리 모듈(240)의 안전한 사용을 가능하게 한다.

[0093] GBA 키를 참조하면, 다음과 같은 키: Ks, 및 Ks로부터 도출되는 NAF 특정 키가 의도된다. NAF 특정 키를 참조하면, 다음과 같은 키: (GBA_U 컨텍스트에서의) Ks_ext/int_NAF 및 (GBA_ME 컨텍스트에서의) Ks_NAF, 및 이러한 키로부터 도출된 어떤 키가 의도된다. Ks_ext_NAF는 Ks_NAF, 즉 ME에 사용된 NAF 특정 키와 같은 키이다. Ks_ext_NAF는 GBA_U 컨텍스트에서의 UICC에서 도출되고, ME에 주어지며, Ks_NAF는 GBA_ME 컨텍스트에서의 ME에서 도출된다. 이들은 둘다 컨텍스트와 무관한 ME에서 동일한 방식으로 사용될 수 있다. Ks_int_NAF는 ULCC에서 도출되고, 그것은 ULCC에서 사용된다. Ks_int_NAF는 ULCC로부터 제공되지 않는다. Ks_js_NAF 키를 참조하면, Ks_NAF 또는 Ks_ext_NAF 대신에 사용되는 애플리케이션 서버 및 자바 스크립트 코드에 대한 자바 스크립트 키가 의도된다.

[0094] 실시예에서, UE와 네트워크 애플리케이션 기능(NAF) 사이의 통신이 시작하기 전에, UE 및 네트워크 애플리케이션 기능(NAF)은 먼저 GBA를 사용할 지의 여부에 동의해야 한다. UE가 네트워크 애플리케이션 기능(NAF)과 상호 작용하기를 원하지만, 네트워크 애플리케이션 기능(NAF)이 GBA에 의해 획득되는 공유 키의 사용을 필요로 하는지를 알지 못하는 경우, UE는 추가의 명령어에 대한 네트워크 애플리케이션 기능(NAF)에 접촉할 것이다.

[0095] 도 3은 본 발명의 실시예에 따른 메시징 다이어그램을 도시한다. 도시된 모든 메시지 및 항목이 수행될 필요가 없고, 메시지의 순서가 변할 수 있고, 더 많은 메시지가 수행될 수 있으며, 도 3에 도시된 메시지 및 항목으로 제한하지 않는다.

[0096] 사용자 장비(UE)와 같은 사용자 장치는 어떤 일반적 부트스트래핑 아키텍처(GBA) 관련 파라미터 없이 네트워크 애플리케이션 기능(NAF) 서버와 같은 애플리케이션 서버와 기준점(Ua)을 통해 통신을 시작할 수 있다. NAF가 GBA에 의해 획득된 공유 키의 사용을 필요로 하지만, UE로부터의 요구가 GBA 관련 파라미터를 포함하지 않는 경

우, 네트워크 애플리케이션 기능(NAF)은 부트스트래핑 개시 메시지로 응답한다. 이러한 지시(indication)의 형식은 특정 기준점(Ua)에 따라 달라질 수 있다.

[0097] 실시예에서, 웹 브라우저(210)는 사용자가 보안 관련 기능을 적절히 처리하고 비밀 번호와 같은 민감한 정보를 제 3 자에 누설하지 않도록 브라우저(210)를 신뢰한다는 의미에서 신뢰된 애플리케이션인 것으로 고려된다. 도 3에서, 웹 브라우저 (210)는 3개의 기능적 블록: 엔진 모듈(310), 자바 스크립트 모듈(320) 및 GBA-API 모듈(330)로 분할된다.

[0098] 엔진 모듈(310)은 웹 서버(120)에 전송 계층 보안(TLS)을 설정하고, 웹 자원을 다운로드하며, 사용자 인터페이스 정보를 사용자에 제공하는 것과 같은 웹 브라우저(210)에 대한 기본적 기능을 처리한다.

[0099] GBA API 모듈(330)은 애플리케이션 프로그래밍 인터페이스(API)를 웹 브라우저(210)에서 실행하는 어떤 자바 스크립트 코드를 향해 제공한다. 자바 스크립트가 명시적으로 신뢰되지 않을 때, 웹 브라우저(210) 및 GBA API(330)는 자바 스크립트에 대한 어떤 민감한 정보를 누설하지 않아야 하고, 필요 이상으로 자바 스크립트로부터 어떤 민감한 정보를 액셉트하지 않아야 한다.

[0100] 자바 스크립트 모듈(320)은 다운로드된 자바 스크립트를 실행한다. 웹 브라우저(210)에서 실행되는 어떤 자바 스크립트 코드는 신뢰되지 않는 것으로 간주되어야 하고, 민감한 자원에 대한 액세스가 승인되지 않아야 하거나 이러한 자원에 대한 액세스가 제어되어야 한다.

[0101] 도 3의 도시된 시퀀스 흐름도는 서버 인증된 전송 계층 보안(TLS) 내에서 실행될 수 있다. 또한, 웹 브라우저(210)는 링크된 자바 스크립트 자원 중 하나가 "gba.js"라고 하는 html 페이지를 다운로드하는 프로세스에서 있을 수 있다.

[0102] 항목 0에서, 브라우저 애플리케이션(210) 및 웹 서버(120)는 서버 인증된 전송 계층 보안(TLS) 터널을 확립한다.

[0103] 도 3의 항목 1에서, 콘텐츠 다운로드는 사용자 장비(UE)와 같은 사용자 장치의 브라우저 애플리케이션(210)에 의해 요청된다. 콘텐츠는 예를 들어 웹 서버와 같은 애플리케이션 서버(120)에 의해 제공된 웹 페이지일 수 있다. 항목 1의 요청은 예를 들어 HTTP 요청을 포함할 수 있다.

[0104] 도 3의 항목 2에서, 웹 서버(120)는 자바 스크립트 코드에 포함되고, 브라우저(210)의 GBA API(330)에 제공되어야 하는 서버 랜덤 챌린지 RAND1를 생성함으로써 자바 스크립트 코드 "gba.js" 파일을 동적으로 구성한다. RAND1은 또한 웹 서버(120)에 국부적으로 저장된다. 이러한 자바 스크립트 코드에서, 자바 스크립트 GBA 애플리케이션 프로그래밍 인터페이스(API)(220)는 자바 스크립트 특정 GBA 키(Ks_js_NAF)를 요청하여 획득하는 데 사용될 수 있다. 랜덤 챌린지 RAND1는 항목 2의 GBA API 요청에 포함된다. 자바 스크립트 특정 GBA 키(Ks_js_NAF) 요청은 또한 브라우저(210)에서 수신될 때에 GBA 모듈(240)로 전송될 수 있고, 추가의 처리를 위해 GBA 모듈(240)에 의해 GBA API(220)로 전송될 수 있다.

[0105] 자바 스크립트 코드(280)를 가진 웹 페이지는 예를 들어 HTTP 응답으로서, 항목 3의 서버(120)로부터 적재된다. 항목 4에서, 웹 브라우저(210)의 엔진(310)은 자바 스크립트 모듈(320)에서 자바 스크립트 코드 "gba.js"를 실행하기 시작한다.

[0106] 항목 5에서, 자바 스크립트 코드 "gba.js"는 GBA API(330)에 대한 호출이 행해지는 지점에 도달한다. 호출은 매개 변수의 하나로서 RAND1을 포함한다. 항목 6에서, 자바 스크립트 GBA API(330)는 수신된 RAND1을 저장한다. GBA API(330)는 또한 자바 스크립트 코드에 관한 관련된 정보, 예를 들어 그것이 실행하고 있는 어떤 html 페이지, 어떤 url에서 html 페이지가 다운로드되었는지, 어떤 TLS 암호 슈트가 TLS 터널에 사용되는 지를 찾는다. 웹 서버(NAF)(120)의 도메인 이름(FQDN)은 웹 페이지의 url로부터 추출될 수 있고, Ua 보안 프로토콜 식별자는 이용된 TLS 암호 슈트로부터 도출될 수 있다. NAF 서버(120)의 도메인 이름(FQDN) 및 Ua 보안 프로토콜 식별자는 네트워크 애플리케이션 기능 식별자(NAF-Id)를 형성한다.

[0107] 항목 7에서, GBA API 모듈(330)은 항목 6에서 도출된 NAF-Id로 GBA 모듈(240)에 대한 호출을 한다. 항목 8에서, GBA 모듈(240)은 유효한 GBA 마스터 키 Ks가 없는 경우에 부트스트래핑 기능(BSF)과 부트스트랩한다. Ks로부터, NAF 특정 키(Ks_ext_NAF)는 NAF-Id를 이용하여 도출된다.

[0108] GBA_ME 경우에, ULCC(270)는 CK 및 IK를 GBA 모듈(240)에 제공하며, 이는 예를 들어 CK 및 IK를 연결함으로써 이들로부터 Ks를 생성한다. 더욱이, GBA 모듈(240)은 Ks NAF-Id를 이용하여 Ks_NAF를 생성한다.

[0109] GBA_U 경우에, ULCC(270)는 그 자체로 CK 및 IK를 유지하고, Ks_ext_NAF를 생성하며, 그 후에 GBA 모듈(240)에 제공된다.

[0110] 따라서, GBA_ME 경우에, 모든 GBA 특정 기능은 ME에서 구현되고, GBA_U 경우에, GBA 기능의 부분은 ULCC(270)에서 구현된다. 주로 Ks는 ULCC(270)에 유지되고, 도출된 Ks_(ext)_NAF만이 GBA 모듈(240)에 제공된다. 다시 말하면, GBA "마스터 키" Ks는 ME(GBA_ME 경우) 또는 ULCC(270)(GBA_U 경우)에 생성된다.

[0111] 필요한 GBA 키를 획득하는 애플리케이션은 단지 GBA 모듈(240)을 처리하고, GBA 키는 제각기 GBA_ME 경우의 Ks_NAF 및 GBA_U 경우의 Ks_ext_NAF 중 어느 하나이다. 그런 다음, 애플리케이션은 소스에 관계없이 GBA 키 Ks_(ext)_NAF를 이용할 수 있다.

[0112] 항목 9에서, GBA 모듈(240)은 NAF 특정 키(Ks_ext_NAF)를 예를 들어 부트스트래핑 트랜잭션 식별자(B-TID) 및 키 수명을 가진 브라우저의 GBA API(330)로 반환한다. 항목 10에서, GBA API(330)는 클라이언트측 랜덤 챌린지 RAND2를 생성할 수 있다. 보안 토큰은 랜덤 챌린지 RAND1 및 랜덤 챌린지 RAND2를 이용하여 결정될 수 있다. 더욱이, 자바 스크립트 특정 GBA 키(Ks_js_NAF)는 서버 특정 부트스트래핑 키(Ks_ext_NAF) 및 보안 토큰(랜덤 챌린지 RAND1 및 RAND2)을 이용하여 생성된다. 키 도출 함수(KDF)는 다음과 같이 자바 스크립트 특정 GBA 키를 생성하는 데 사용될 수 있다:

[0113] Ks_js_NAF = KDF (Ks_ext_NAF, RAND1 || RAND2)

[0114] RAND1은 서버(120)로부터 수신되는 랜덤 챌린지이고, RAND2는 GBA API(330)에 의해 생성된다. Ks_(ext)_NAF는 자바 스크립트 레벨에서 GBA API(330)로 처리될 수 있다. 자바 스크립트 함수는 예를 들어 GBA.getNAFKey(RAND1)라 불리워질 수 있으며, 그 후 이러한 함수는 Ks_js_NAF 및 RAND2를 반환한다.

[0115] 항목 11에서, GBA API(330)는 자바 스크립트 특정 Ks_js_NAF 키, RAND2, B-TID 및 키 수명을 실행 자바 스크립트 모듈(320)로 반환한다. 항목 12에서, 자바 스크립트 모듈(320)은 웹 서버(120)가 (자바 스크립트 코드 "gba.js"를 통해) 지시한 식으로 Ks_js_NAF 키를 실행하고 이용하기를 계속한다.

[0116] 항목 13에서, 클라이언트 측 논리를 실행한 후, 자바 스크립트 모듈(320)은 웹 서버(120)에 대한 요청(예를 들어 HTTP 요청)을 한다. 이러한 요청은 적어도 Ks_js_NAF, RAND2 및 B-TID를 포함할 수 있다.

[0117] 항목 14에서, 웹 서버(120)는 부트스트래핑 기능(BSF)으로부터 Ks_ext_NAF를 인출하여, 수신된 RAND2 및 저장된 RAND1로 Ks_ext_NAF를 도출한다. 웹 서버(120)는 수신된 Ks_ext_NAF를 검증을 위해 국부적으로 도출된 것과 비교할 수 있다. 수신된 Ks_ext_NAF가 유효하면, 웹 서버(120)는 항목 13에서 행해진 요청을 계속 처리하고, 그 결과를 항목 15에서 웹 브라우저(120)의 자바 스크립트 모듈(320)로 반환할 것이다. 더욱이, 웹 서버(120)는 자바 스크립트 코드를 계속 실행할 수 있다.

[0118] 실시예에서, NAF 특정 키(Ks_NAF)는 이와 같이 서버로 전송되지 않으며, 이는 보안 메커니즘을 개선한다. 더욱이, RAND1 및 RAND2이 변경되기 때문에 자바 스크립트 특정 키(Ks_js_NAF)는 GBA API(330)가 사용될 때마다 변경된다. 이러한 메커니즘은 예를 들어 추가의 보안 및 재생 보호 기능을 제공한다.

[0119] 다른 실시예에서는 서로 다른 보안 토큰이 사용된다. 이러한 실시예에서, 도 3의 항목 2에서, 웹 서버(120)는 브라우저(210)의 GBA API(330)에 제공되도록 자바 스크립트 코드 "gba.js" 파일을 선택한다. 이러한 자바 스크립트 코드에서, 자바 스크립트 GBA 애플리케이션 프로그래밍 인터페이스(API)(220)는 자바 스크립트 특정 GBA 키(Ks_js_NAF)를 요청하고 획득하는 데 사용될 수 있다. 자바 스크립트 특정 GBA 키(Ks_js_NAF) 요청은 또한 브라우저(210)에서 수신될 때 GBA 모듈(240)로 전송되고, 추가의 처리를 위해 GBA 모듈(240)에 의해 GBA API(220)로 전송될 수 있다.

[0120] 자바 스크립트 코드(280)를 가진 웹 페이지는 예를 들어 HTTP 응답으로서, 항목 3의 서버(120)로부터 적재된다. 항목 4에서, 웹 브라우저(210)의 엔진(310)은 자바 스크립트 모듈(320)에서 자바 스크립트 코드 "gba.js"를 실행하기 시작한다.

[0121] 항목 5에서, 자바 스크립트 코드 "gba.js"는 GBA API(330)에 대한 호출이 행해지는 지점에 도달한다. 항목 6에서, 자바 스크립트 GBA API(330)는 자바 스크립트 코드에 관한 관련된 정보, 예를 들어 그것이 실행하고 있는 어떤 html 페이지, 어떤 url에서 html 페이지가 다운로드되었는지, 어떤 전송 계층 보안(TLS) 암호 슈트가 TLS 터널에 사용되는 지를 찾는다. 웹 서버(NAF)(120)의 도메인 이름(FQDN)은 웹 페이지의 url로부터 추출될 수 있고, Ua 보안 프로토콜 식별자는 이용된 TLS 암호 슈트로부터 도출될 수 있다. NAF 서버(120)의 도메인 이름

(FQDN) 및 Ua 보안 프로토콜 식별자는 네트워크 애플리케이션 기능 식별자(NAF-Id)를 형성한다.

[0122] 항목 7에서, GBA API 모듈(330)은 항목 6에서 도출된 NAF-Id로 GBA 모듈(240)에 대한 호출을 한다. 항목 8에서, GBA 모듈(240)은 유효한 GBA 마스터 키 Ks가 없는 경우에 부트스트래핑 기능(BSF)과 부트스트랩한다. Ks로부터, NAF 특정 키(Ks_ext_NAF)는 NAF-Id를 이용하여 도출된다.

[0123] GBA_ME 경우에, ULCC(270)는 CK 및 IK를 GBA 모듈(240)에 제공하며, 이는 예를 들어 CK 및 IK를 연결함으로써 이들로부터 Ks를 생성한다. 더욱이, GBA 모듈(240)은 Ks NAF-Id를 이용하여 Ks_NAF를 생성한다.

[0124] GBA_U 경우에, ULCC(270)는 그 자체로 CK 및 IK를 유지하고, Ks_ext_NAF를 생성하며, 그 후에 GBA 모듈(240)에 제공된다.

[0125] 따라서, GBA_ME 경우에, 모든 GBA 특정 기능은 ME에서 구현되고, GBA_U 경우에, GBA 기능의 부분은 ULCC(270)에서 구현된다. 주로 Ks는 ULCC(270)에 유지되고, 도출된 Ks_(ext)_NAF만이 GBA 모듈(240)에 제공된다. 다시 말하면, GBA "마스터 키" Ks는 ME(GBA_ME 경우) 또는 ULCC(270)(GBA_U 경우)에 생성된다.

[0126] 필요한 GBA 키를 획득하는 애플리케이션은 단지 GBA 모듈(240)을 처리하고, GBA 키는 제각기 GBA_ME 경우의 Ks_NAF 및 GBA_U 경우의 Ks_ext_NAF 중 어느 하나이다. 그런 다음, 애플리케이션은 소스에 관계없이 GBA 키 Ks_(ext)_NAF를 이용할 수 있다.

[0127] 항목 9에서, GBA 모듈(240)은 NAF 특정 키(Ks_ext_NAF)를 예를 들어 부트스트래핑 트랜잭션 식별자(B-TID) 및 키 수명을 가진 브라우저의 GBA API(330)로 반환한다. 항목 10에서, Ks_(ext)_NAF 키를 수신하면, 브라우저의 GBA API(330)는 보안 토큰을 결정할 수 있다. 보안 토큰(TLS_MK_Extr)은 익스포트된(exported) 함수를 이용하여 전송 계층 보안(TLS) 마스터 키로부터 추출될 수 있다. 익스포트된 함수에 대한 라벨은 예를 들어, "EXPORTER_3GPP_GBA_WEB"일 수 있다. 보안 토큰(TLS_MK_Extr)은 서버 인증된 TLS 터널로 바인딩되는 자바 스크립트 특정 키 Ks_js_NAF를 도출하는 데 사용될 수 있다. Ks_js_NAF는 다음과 같이 Ks_(ext)_NAF로부터 도출될 수 있다:

[0128] $$Ks\_js\_NAF = KDF (Ks\_(ext)\_NAF, TLS\_MK\_Extr)$$

[0129] 자바 스크립트 특정 GBA 키(Ks_js_NAF)는 서버 특정 부트스트래핑 키 Ks_(ext)_NAF 및 보안 토큰(TLS_ML_Extr)를 이용하여 생성된다. 키 도출 함수(KDF)는 자바 스크립트 특정 GBA 키를 생성하는 데 사용될 수 있다. Ks_(ext)_NAF는 자바 스크립트 레벨에서 GBA API(330)로 처리될 수 있다.

[0130] 항목 11에서, GBA API(330)는 자바 스크립트 특정 Ks_js_NAF 키, B-TID 및 키 수명을 실행 자바 스크립트 모듈(320)로 반환한다. 항목 12에서, 자바 스크립트 모듈(320)은 웹 서버(120)가 (자바 스크립트 코드 "gba.js"를 통해) 지시한 식으로 Ks_js_NAF 키를 실행하고 이용하기를 계속한다.

[0131] 항목 13에서, 클라이언트 측 논리를 실행한 후, 자바 스크립트 모듈(320)은 웹 서버(120)에 대한 요청(예를 들어 HTTP 요청)을 한다. 이러한 요청은 적어도 Ks_js_NAF 및 B-TID를 포함할 수 있다.

[0132] 항목 14에서, 웹 서버(120)는 부트스트래핑 기능(BSF)으로부터 Ks_(ext)_NAF를 인출하여, 항목 10에서 행한 바와 같이 보안 토큰(TLS_MK_Extr)을 결정할 수 있다. 그 후, 웹 서버(120)는 보안 토큰(TLS_MK_Extr)으로 Ks_js_NAF를 도출할 수 있다. 웹 서버(120)는 수신된 Ks_js_NAF를 검증을 위해 국부적으로 도출된 것과 비교할 수 있다. 수신된 Ks_js_NAF가 유효하면, 웹 서버(120)는 항목 13에서 행해진 요청을 계속 처리하고, 그 결과를 항목 15에서 웹 브라우저(120)의 자바 스크립트 모듈(320)로 반환할 것이다. 더욱이, 웹 서버(120)는 자바 스크립트 코드를 계속 실행할 수 있다.

[0133] 실시예에서, NAF 특정 키(Ks_NAF)는 이와 같이 서버로 전송되지 않으며, 이는 보안 메커니즘을 개선한다.

[0134] 도 4는 본 발명의 다양한 실시예가 적용될 수 있는 애플리케이션 서버(400)의 예시적인 블록도를 도시한다. 이것은 웹 서버, 파일 다운로드 서버 또는 임의의 콘텐츠 제공 서버일 수 있다.

[0135] 애플리케이션 서버(400)의 일반적인 구조는 통신 인터페이스 모듈(450), 통신 인터페이스 모듈(450)에 결합된 프로세서(410), 및 프로세서(410)에 결합된 메모리(420)를 포함한다. 장치는 메모리(420)에 저장되고, 프로세서(410)에 적재되어 실행되도록 동작 가능한 소프트웨어(430)를 더 포함한다. 소프트웨어(430)는 하나 이상의 소프트웨어 모듈을 포함할 수 있고, 컴퓨터 프로그램 제품의 형태일 수 있다.

[0136] 통신 인터페이스 모듈(450)은 본 발명의 다양한 실시예와 관련하여 논의된 데이터 전송의 적어도 부분을 구현한

다. 통신 인터페이스 모듈(450)은 예를 들어 WLAN, 블루투스, GSM/GPRS, CDMA, WCDMA 또는 LTE(Long Term Evolution) 무선 모듈과 같은 무선 인터페이스 모듈일 수 있다. 통신 인터페이스 모듈(450)은 애플리케이션 서버(400) 내에 통합될 수 있거나, 애플리케이션 서버(400)의 적절한 슬롯 또는 포트에 삽입될 수 있는 어댑터, 카드 등에 통합될 수 있다. 통신 인터페이스 모듈(450)은 하나의 무선 인터페이스 기술 또는 복수의 기술을 지원할 수 있다. 도 4는 하나의 통신 인터페이스 모듈(450)을 도시하지만, 애플리케이션 서버(400)는 복수의 통신 인터페이스 모듈(450)을 포함할 수 있다. 통신 인터페이스 모듈(450)은 예를 들어 부트스트래핑 기능(BSF), 홈 가입자 서버(HSS) 및 외부 콘텐츠 서버와의 데이터 통신을 제공한다.

[0137] 프로세서(410)는 예를 들어 중앙 처리 장치(CPU), 마이크로 프로세서, 디지털 신호 프로세서(DSP), 그래픽 처리 장치 등일 수 있다. 도 4는 하나의 프로세서(410)를 도시하지만, 애플리케이션 서버(400)는 복수의 프로세서를 포함할 수 있다.

[0138] 메모리(420)는 예를 들어 판독 전용 메모리(ROM), 프로그램 가능한 판독 전용 메모리(PROM), 소거 및 프로그램 가능한 판독 전용 메모리(EPROM), 랜덤 액세스 메모리(RAM), 플래시 메모리, 데이터 디스크, 광학 저장소, 자기 저장소, 스마트 카드 등과 같은 비휘발성 또는 휘발성 메모리일 수 있다. 애플리케이션 서버(400)는 복수의 메모리를 포함할 수 있다. 메모리(420)는 애플리케이션 서버(400)의 일부로서 구성될 수 있거나, 애플리케이션 서버(400)의 슬롯, 포트 등에 삽입될 수 있다. 메모리(420)는 데이터를 저장하는 역할을 할 수 있거나, 데이터를 처리하는 것과 같은 다른 목적을 서빙하는 장치의 일부로서 구성될 수 있다.

[0139] 일반적 부트스트래핑 아키텍처 모듈(GBA)(440)은 네트워크 애플리케이션 기능(NAF)을 포함할 수 있다. GBA는 인증을 위해 네트워크 애플리케이션 기능(NAF)과 UE 사이에 사용될 수 있고, UE와 네트워크 애플리케이션 기능(NAF) 사이에 통신 경로를 확보하기 위해 사용될 수 있다. 부트스트래핑이 완료된 후, UE 및 네트워크 애플리케이션 기능(NAF)은 메시지의 인증이 UE와 부트스트래핑 서버 기능(BSF) 사이에서 상호 인증 동안에 생성되는 이러한 세션 키에 기초하는 일부 애플리케이션 특정 프로토콜을 실행할 수 있다.

[0140] 당업자는 도 4에 도시된 요소 외에, 애플리케이션 서버(400)가 입력/출력(I/O) 회로, 메모리 칩, 주문형 반도체(ASIC), 소스 코딩/디코딩 회로, 채널 코딩/디코딩 회로, 암호화/해독화 회로 등과 같은 추가적인 회로와 같은 다른 요소를 포함할 수 있다는 것을 인식한다.

[0141] 도 5는 본 발명의 다양한 실시예가 적용될 수 있는 사용자 장치(500)의 예시적인 블록도를 도시한다. 이것은 이동 단말기, 랩톱, 태블릿 또는 다른 통신 장치와 같이 사용자 장비(UE), 사용자 장치일 수 있다.

[0142] 사용자 장치(500)의 일반적인 구조는 통신 인터페이스 모듈(550), 통신 인터페이스 모듈(550)에 결합된 프로세서(510), 및 프로세서(510)에 결합된 메모리(520)를 포함한다. 사용자 장치는 메모리(520)에 저장되고, 프로세서(510)에 적재되어 실행되도록 동작 가능한 소프트웨어(530)를 더 포함한다. 소프트웨어(530)는 하나 이상의 소프트웨어 모듈을 포함할 수 있고, 컴퓨터 프로그램 제품의 형태일 수 있다. 사용자 장치(500)는 프로세서(510)에 결합된 사용자 인터페이스 제어기(560)를 더 포함한다.

[0143] 통신 인터페이스 모듈(550)은 본 발명의 다양한 실시예와 관련하여 논의된 사용자 데이터 무선의 적어도 부분을 구현한다. 통신 인터페이스 모듈(550)은 예를 들어 WLAN, 블루투스, GSM/GPRS, CDMA, WCDMA 또는 LTE(Long Term Evolution) 무선 모듈과 같은 무선 인터페이스 모듈일 수 있다. 통신 인터페이스 모듈(550)은 사용자 장치(500) 내에 통합될 수 있거나, 사용자 장치(500)의 적절한 슬롯 또는 포트에 삽입될 수 있는 어댑터, 카드 등에 통합될 수 있다. 통신 인터페이스 모듈(550)은 하나의 무선 인터페이스 기술 또는 복수의 기술을 지원할 수 있다. 도 5는 하나의 통신 인터페이스 모듈(550)을 도시하지만, 사용자 장치(500)는 복수의 통신 인터페이스 모듈(550)을 포함할 수 있다.

[0144] 프로세서(510)는 예를 들어 중앙 처리 장치(CPU), 마이크로 프로세서, 디지털 신호 프로세서(DSP), 그래픽 처리 장치 등일 수 있다. 도 5는 하나의 프로세서(510)를 도시하지만, 사용자 장치(500)는 복수의 프로세서를 포함할 수 있다.

[0145] 메모리(520)는 예를 들어 판독 전용 메모리(ROM), 프로그램 가능한 판독 전용 메모리(PROM), 소거 및 프로그램 가능한 판독 전용 메모리(EPROM), 랜덤 액세스 메모리(RAM), 플래시 메모리, 데이터 디스크, 광학 저장소, 자기 저장소, 스마트 카드 등과 같은 비휘발성 또는 휘발성 메모리일 수 있다. 사용자 장치(500)는 복수의 메모리를 포함할 수 있다. 메모리(520)는 장치(500)의 일부로서 구성될 수 있거나, 사용자에 의해 사용자 장치(500)의 슬롯, 포트 등에 삽입될 수 있다. 메모리(520)는 데이터를 저장하는 역할을 할 수 있거나, 데이터를 처리하는 것과 같은 다른 목적을 서빙하는 장치의 일부로서 구성될 수 있다.

[0146] 범용 집적 회로 카드(UICC)(540)는 사용자 장치(500)에 사용되는 스마트 카드로서 포함될 수 있다. 범용 집적 회로 카드(UICC)(540)는 어떤 개인 데이터의 무결성 및 보안을 보장한다. 범용 집적 회로 카드(UICC)(540)는 이의 고유 일련 번호, 모바일 사용자(IMSI)의 국제적 고유 번호, 보안 인증 및 암호화 정보, 로컬 네트워크에 관련된 일시적인 정보, 사용자가 액세스하는 서비스의 목록 및 비밀 번호(일반적인 사용을 위한 PIN 및 잠금 해제를 위한 PUK)를 포함할 수 있다. 범용 집적 회로 카드(UICC)(540)는 여러 애플리케이션을 더 포함하고, 동일한 스마트 카드가 서로 다른 네트워크에 액세스할 수 있도록 하며, 또한 전화 번호부 및 다른 애플리케이션의 저장을 제공할 수 있다. 시스템은 키 저장 및 처리를 위해 내장된 보안 모듈을 이용할 수 있다.

[0147] 사용자 인터페이스 제어기(560)는 예를 들어, 키보드, 사용자 장치(500)의 디스플레이 상에 표시된 그래픽 사용자 인터페이스, 음성 인식 회로, 또는 헤드셋과 같은 부속 장치를 통해 사용자 장치(500)의 사용자로부터 입력을 수신하고, 예를 들어 그래픽 사용자 인터페이스 또는 스피커를 통해 출력을 사용자에게 제공하는 회로를 포함할 수 있다.

[0148] 당업자는 도 5에 도시된 요소 외에, 사용자 장치(500)가 마이크로폰, 디스플레이와 같은 다른 요소 뿐만 아니라 입력/출력(I/O) 회로, 메모리 칩, 주문형 반도체(ASIC), 소스 코딩/디코딩 회로, 채널 코딩/디코딩 회로, 암호화/해독화 회로 등과 같은 추가적인 회로를 포함할 수 있다는 것을 인식한다. 부가적으로, 사용자 장치(500)는 외부 전력 공급이 이용 가능하지 않을 경우에 사용자 장치(500)에 전력을 공급하는 (도시되지 않은) 일회용 또는 충전식 배터리를 포함할 수 있다.

[0149] 도 6은 본 발명의 예시적인 실시예에 따라 사용자 장치에서의 동작을 도시한 흐름도이다. 단계(600)에서, 방법이 시작된다. 단계(610)에서, 서버 특정 부트스트래핑 키(Ks_NAF)에 대한 요청을 포함하는 외부 코드가 수신된다. 단계(620)에서, 서버 식별자(NAF-Id)가 결정된다. 서버 특정 부트스트래핑 키(Ks_NAF)는 단계(630)에서 서버 식별자(NAF-Id)에 기초하여 생성된다. 단계(640)에서, 보안 토큰이 결정된다. 단계(650)에서, 외부 코드 특정 부트스트래핑 키(Ks_js_NAF)는 서버 특정 부트스트래핑 키(Ks_NAF) 및 보안 토큰을 이용하여 생성된다. 외부 코드 특정 부트스트래핑 키(Ks_js_NAF)는 단계(660)에서 외부 코드의 보안 메커니즘에 이용된다. 방법은 단계(670)에서 종료한다.

[0150] 도 7은 본 발명의 예시적인 실시예에 따라 애플리케이션 서버에서의 동작을 도시한 흐름도이다. 단계(700)에서, 방법이 시작된다. 단계(710)에서, 스크립트 코드 특정 부트스트래핑 키(Ks_js_NAF)에 대한 요청을 포함하는 스크립트 코드가 전송된다. 서버 식별자(NAF-Id)는 단계(720)에서 결정된다. 단계(730)에서, 서버 특정 부트스트래핑 키(Ks_NAF)는 서버 식별자(NAF-Id)를 이용하여 생성된다. 단계(740)에서, 보안 토큰이 결정된다. 단계(740)에서, 스크립트 코드 특정 부트스트래핑 키(Ks_js_NAF)는 서버 특정 부트스트래핑 키(Ks_NAF) 및 보안 토큰을 이용하여 생성된다. 스크립트 코드 특정 부트스트래핑 키(Ks_js_NAF)는 단계(760)에서 스크립트 코드의 보안 메커니즘에 이용된다. 방법은 단계(770)에서 종료한다.

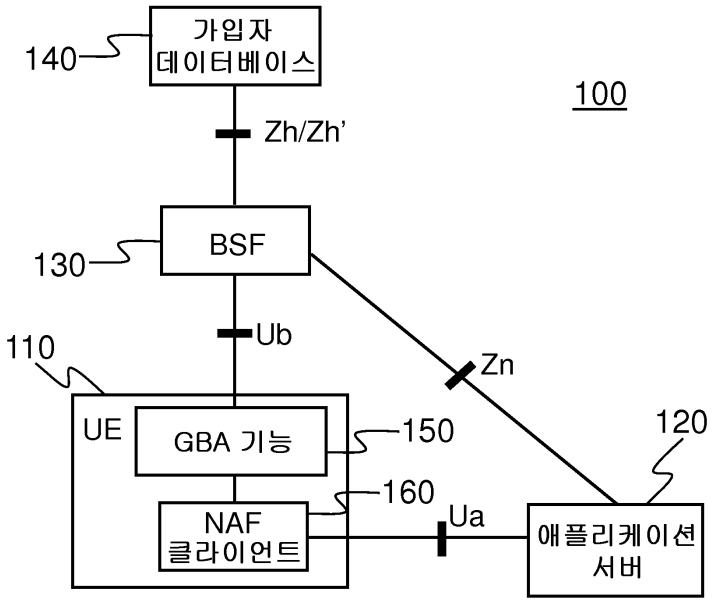
[0151] 다양한 실시예가 제공되었다. 본 문서에서 단어 "포함하다(comprise, include, 및 contain)"는 각각 의도된 배타성을 가지는 않는 개방형 종결 표현으로 사용되는 것으로 이해되어야 한다.

[0152] 상술한 설명은 본 발명의 특정 구현 및 실시예의 비제한적 예에 의해 본 발명을 실행하기 위해 발명자에 의해 현재 고려되는 최상의 모드에 대한 완전하고 정보를 제공하는 설명을 제공하였다. 그러나, 본 발명이 위에 제공된 실시예의 상세 사항에 제한되지 않고 본 발명의 특성으로부터 벗어나지 않으면서 균등한 수단을 이용한 다른 실시예 또는 실시예의 서로 다른 조합으로 구현될 수 있다는 점은 당업자에게 자명하다.

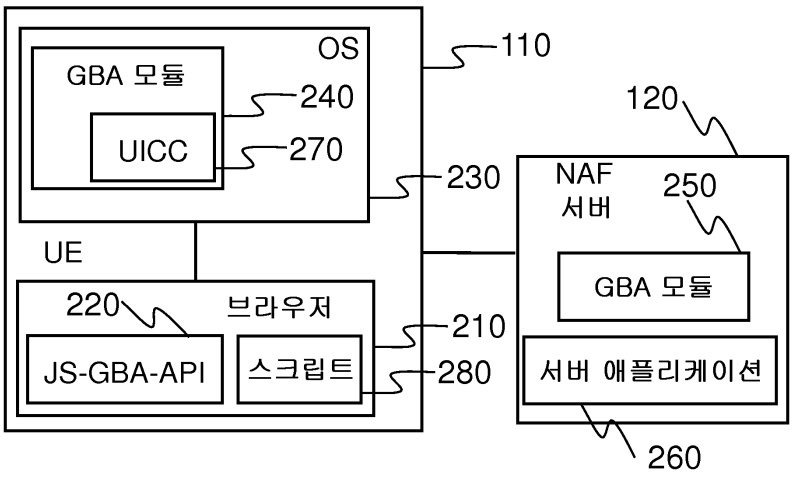
[0153] 더욱이, 본 발명의 위에 개시된 실시예의 특징의 일부는 다른 특징의 상응하는 사용 없이 유리하도록 사용될 수 있다. 이와 같이, 상술한 설명은 본 발명을 제한하지 않고 단지 본 발명의 원리의 예시하는 것으로 간주되어야 한다. 따라서, 본 발명의 범위는 첨부된 특허 청구 범위에 의해서만 제한된다.

도면

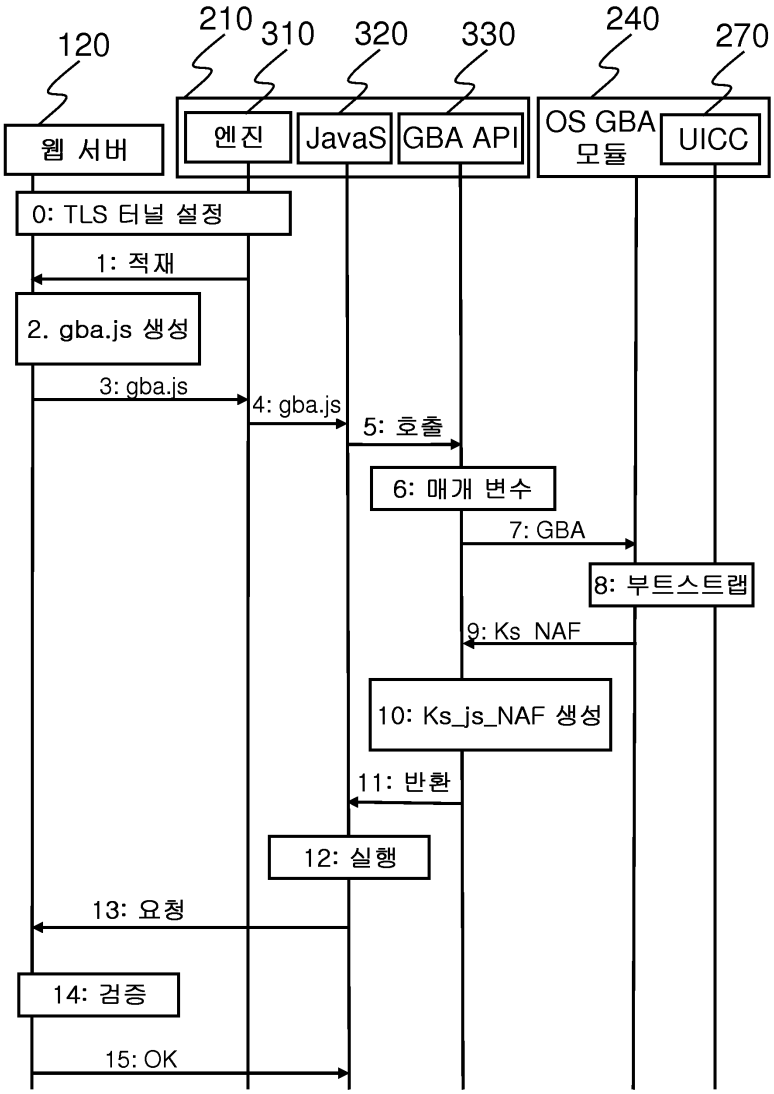
도면1


가입자
데이터베이스
140
100
Zh/Zh'
BSF
130
Ub
110
Zn
120
UE
GBA 기능
150
160
NAF
클라이언트
Ua
애플리케이션
서버

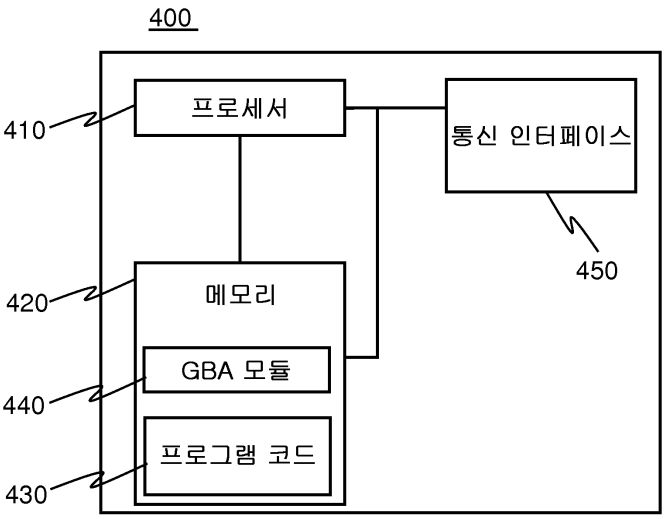
도면2


OS
110
GBA 모듈
240
UICC
270
120
230
NAF
서버
250
UE
GBA 모듈
220
브라우저
210
JS-GBA-API
스크립트
280
서버 애플리케이션
260

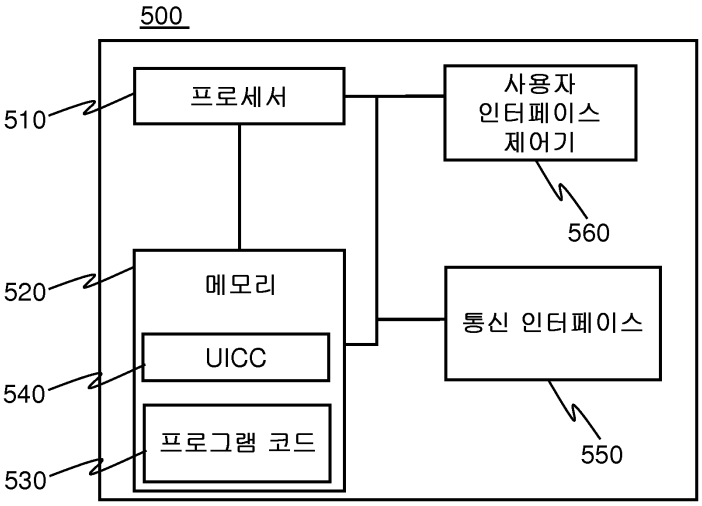
도면3


120
210
310
320
330
240
270
웹 서버
엔진
JavaS
GBA API
OS GBA 모듈
UICC
0: TLS 터널 설정
1: 적재
2. gba.js 생성
3: gba.js
4: gba.js
5: 호출
6: 매개 변수
7: GBA
8: 부트스트랩
9: Ks_NAF
10: Ks_js_NAF 생성
11: 반환
12: 실행
13: 요청
14: 검증
15: OK

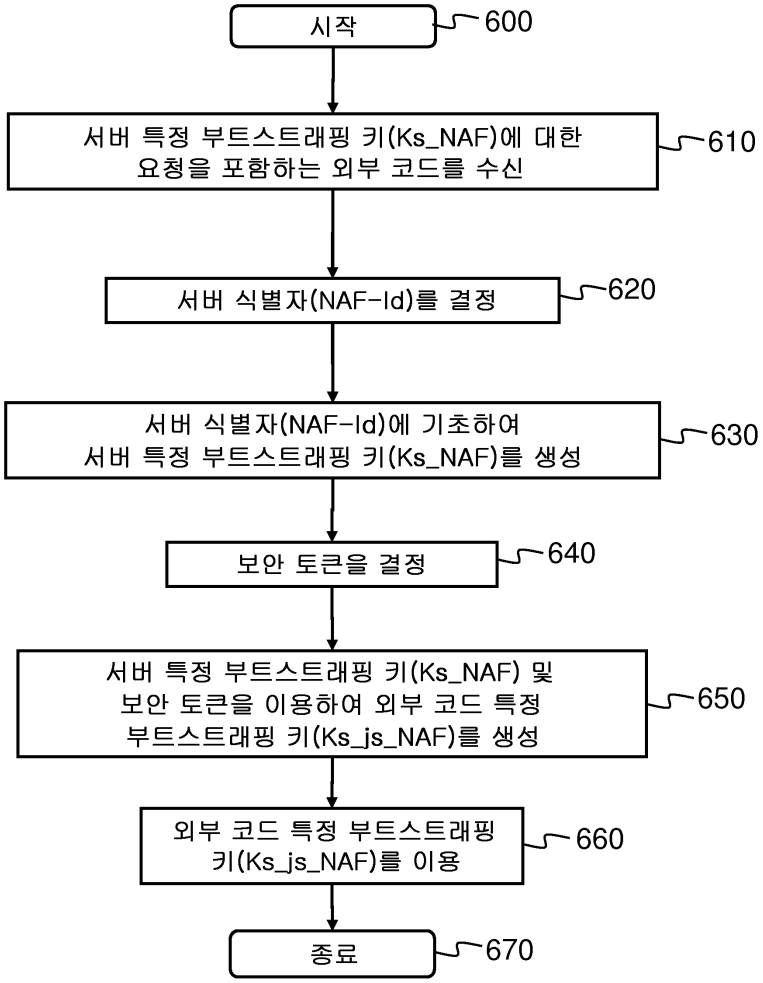
도면4


400
410
프로세서
통신 인터페이스
450
420
메모리
GBA 모듈
440
프로그램 코드
430

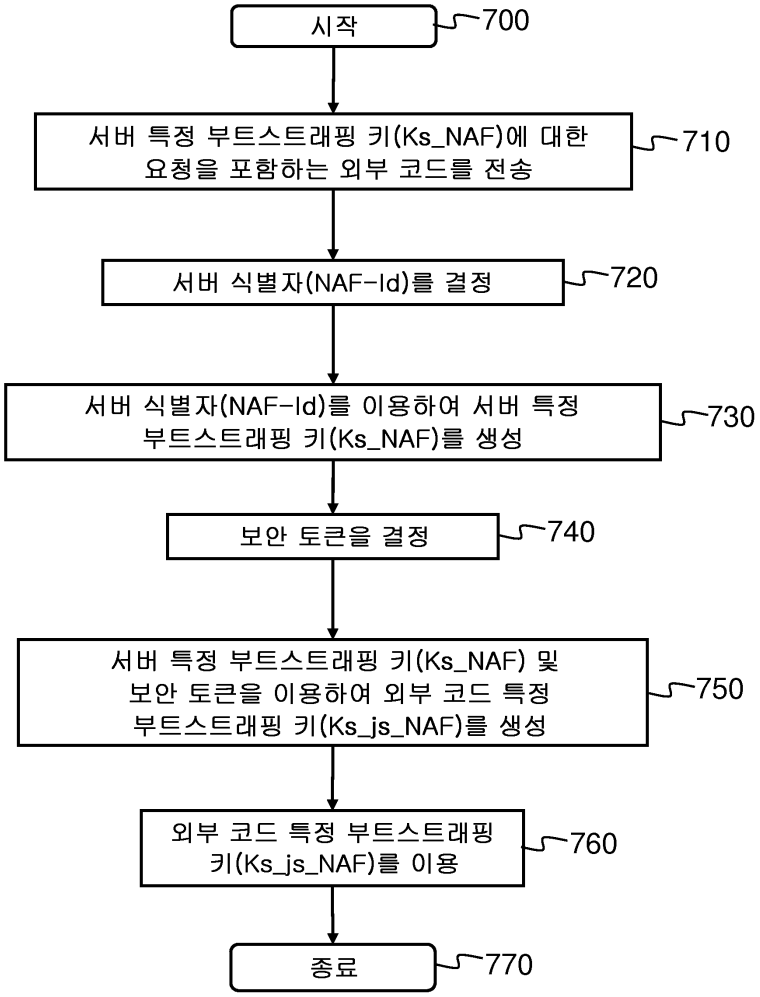
도면5


500
프로세서
510
사용자
인터페이스
제어기
560
메모리
520
UICC
540
프로그램 코드
530
통신 인터페이스
550

도면6


시작
600
서버 특정 부트스트래핑 키(Ks_NAF)에 대한
요청을 포함하는 외부 코드를 수신
610
서버 식별자(NAF-Id)를 결정
620
서버 식별자(NAF-Id)에 기초하여
서버 특정 부트스트래핑 키(Ks_NAF)를 생성
630
보안 토큰을 결정
640
서버 특정 부트스트래핑 키(Ks_NAF) 및
보안 토큰을 이용하여 외부 코드 특정
부트스트래핑 키(Ks_js_NAF)를 생성
650
외부 코드 특정 부트스트래핑
키(Ks_js_NAF)를 이용
660
종료
670

도면7


시작
700
서버 특정 부트스트래핑 키(Ks_NAF)에 대한 요청을 포함하는 외부 코드를 전송
710
서버 식별자(NAF-Id)를 결정
720
서버 식별자(NAF-Id)를 이용하여 서버 특정 부트스트래핑 키(Ks_NAF)를 생성
730
보안 토큰을 결정
740
서버 특정 부트스트래핑 키(Ks_NAF) 및 보안 토큰을 이용하여 외부 코드 특정 부트스트래핑 키(Ks_js_NAF)를 생성
750
외부 코드 특정 부트스트래핑 키(Ks_js_NAF)를 이용
760
종료
770