(19)대한민국특허청(KR)
(12) 등록특허공보(B1)

(51) 。Int. Cl.
H04L 9/30 (2006.01)
H04L 9/32 (2006.01)

(45) 공고일자 2006년11월10일
(11) 등록번호 10-0642745
(24) 등록일자 2006년10월30일

(21) 출원번호 10-2005-0093485
(22) 출원일자 2005년10월05일

(65) 공개번호
(43) 공개일자

(73) 특허권자 고려대학교 산학협력단
서울 성북구 안암동5가1 고려대학교 내

(72) 발명자 이동훈
경기 고양시 일산구 일산3동 후곡마을8단지아파트 808-804

황정연
서울 성북구 안암동5가 101-104

최규영
서울 동대문구 제기2동 133-7

(74) 대리인 현종철

(56) 선행기술조사문헌
KR1020030037145 A
KR1020040050625 A
US20050084114 A1
* 심사관에 의하여 인용된 문헌

심사관 : 양종필

(54) ＩＤ-기반의 키교환 방법 및 장치

요약

ID-기반의 키교환 방법 및 장치가 개시된다. 그 키교환 방법은 (a) 클라이언트가 ID_u, X 및 Y를 서버에게 전달하는 단계; (b) 서버는 상기 ID_u, X 및 Y를 받아 클라이언트를 인증하고, 인증이 되면 z 및 t_v를 생성하여 클라이언트에게 전달하고 세션키(sk)를 생성하는 단계; (c) 클라이언트는 서버로부터 z, t_v를 받아 서버를 인증하고, 유효한 서버로 인증되면 상기 서버의 세션키와 동일한 세션키를 생성하는 단계를 포함함을 특징으로 한다.

본 발명에 의하면, 컴퓨팅 파워가 다른 두 사용자(서버, 클라이언트) 사이의 효율적인 ID-기반의 인증된 키 교환을 할 수 있다. 특히 클라이언트 측의 계산량을 줄임으로써 낮은 파워를 가진 모바일 장치에 적합하다.

대표도

도 6

명세서

도면의 간단한 설명

도 1은 본 발명에 의한 ID-기반 키교환 장치의 구성을 블록도로 도시한 것이다.

도 2는 X&Y의 생성부(100)의 세부적인 구성을 블록도로 도시한 것이다.

도 3은 서버 인증부(120)의 세부 구성을 블록도로 도시한 것이다.

도 4는 z&t_v생성부(130)의 세부 구성을 블록도로 도시한 것이다.

도 5는 클라이언트 인증부(160)의 세부적인 구성을 블록도로 도시한 것이다.

도 6은 본 발명에 의한 ID-기반 키교환 방법을 설명하기 위한 흐름도를 도시한 것이다.

발명의 상세한 설명

발명의 목적

발명이 속하는 기술 및 그 분야의 종래기술

본 발명은 키교환에 관한 것으로서, 특히 ID-기반 키교환 방법 및 장치에 관한 것이다.

현대의 많은 협업과 분산 환경에서 인증된 키 교환은 매우 중요한 문제 가운데 하나이다. 키 교환 프로토콜은 분배된 키를 소유한 사용자들과의 비밀 통신을 목적으로 한다. 그리고 이러한 키 교환 프로토콜에 어떤 의도된 사용자들에 대한 상호 인증을 제공하는 키 교환 프로토콜을 인증된 키 교환(AKA) 프로토콜이라 한다. 다양한 인증 기법들 중 전형적인 인증서 기반의 공개키 기반 구조(Public Key Infrastructure, PKI)는 인증하고자 하는 상대방의 공개키에 대하여 신뢰 기관으로부터 발급된 인증서가 필요하다.

반면에 ID-기반 시스템은 단지 사용자의 메일 주소와 같은 평이한 공개 정보로 구성된 공개 확인자(Identity, ID)만 알면 된다. 따라서 인증서 기반의 PKI에 비해 ID-기반의 인증 시스템이 보다 효율적이고 간단한 인증방식을 제공한다.

최근 무선 환경이 급속도로 발달함에 따라 이에 적당한 키 교환 프로토콜이 요구되고 있다. 특히 서버와 저전력 모바일과 같은 불균형 컴퓨팅 환경에서의 안전한 채널의 형성은 안전한 통신을 위하여 꼭 필요하다. 그러나 불행하게도 대부분의 ID-기반의 AKA 프로토콜은 연산속도와 계산량의 복잡함 때문에 저 전력 모바일 장치와 같은 자원 제한적인 곳에 적용하기엔 적절하지 않다.

발명이 이루고자 하는 기술적 과제

본 발명이 이루고자 하는 기술적 과제는 클라이언트 측의 계산량이 적어 저 전력 모바일 장치에도 사용할 수 있으며, 하나의 서버를 중심으로 여러 클라이언트가 안전한 채널을 형성할 수 있는, ID-기반 키교환 방법 및 장치를 제공하는 것이다.

발명의 구성 및 작용

상술한 기술적 과제를 해결하기 위한 본 발명에 의한 ID-기반 키 교환 방법은, 위수가 소수 q인 gap DiffieHellman 그룹의 생성자를 P, 키생성센터(KGC)의 마스터 비밀키를 s, 상기 s와 P의 곱을 P_{pub}(= sP), 상기 P의 pairing 연산값을 g(=e(P,P)), 클라이언트 및 서버의 ID를 각각 ID_u 및 ID_v, 키생성센터로부터 받은 클라이언트 및 서버의 비밀키를 각각 S_u 및 S_v 라 할 때, (a) 클라이언트가 ID_u, X 및 Y를 서버에게 전달하는 단계; (b) 서버는 상기 ID_u, X 및 Y를 받아 클라이언트를 인증하고, 인증이 되면 z 및 t_v를 생성하여 클라이언트에게 전달하고 세션키(sk)를 생성하는 단계; (c) 클라이언트는 서버로부터 z, t_v를 받아 서버를 인증하고, 유효한 서버로 인증되면 상기 서버의 세션키와 동일한 세션키를 생성하는 단계를 포함함을

특징으로 한다. 상기 (a)단계의 X 및 Y의 생성은 위수가 q인 곱셈그룹 Zq^*로부터 임의의 값 a를 선택하는 단계; 상기 ID_V의 해쉬함수 값 $q_v(= H(ID_V))$를 생성하고, 상기 P_{pub} 와 q_vP를 합하여 $Q_v(= P_{pub} + q_vP)$를 생성하고, 상기 g의 a 멱승을 취하여 $t_u(= g^a)$를 생성하고, 상기 t_u의 해쉬함수 값 $h(=H_1(t_u))$를 생성하는 단계; 및 상기 a와 Q_v의 곱을 계산하여 X $(=aQ_V)$를 생성하고, 상기 a와 h의 합에 S_U 를 곱하여 $Y(=(a+ h)S_U)$를 생성하는 단계를 포함하여 이루어진다.

상기 (b)단계의 클라이언트 인증은, 상기 ID_U 의 해쉬함수 값 $q_u(= H(ID_U))$를 생성하고, 상기 P_{pub} 를 q_u와 P의 곱(q_uP)과 합하여 $Q_U(= P_{pub} + q_uP)$를 생성하고, 클라이언트로부터 받은 X와 S_V의 pairing 연산하여 $t_u(= e(X, S_V))$를 생성하고, 클라이언트로부터 받은 Y와 상기 Q_U 를 pairing 연산하여 $c(= e(Y, Q_U))$를 생성하는 단계; 상기 t_u 의 해쉬함수 값 $h(=H_1(t_u))$을 생성하여, 상기 t_u와 g의 h 멱승(g^h)을 곱한 값(t_ug^h)을 생성하는 단계; 및 상기 $c = t_ug^h$를 만족하면 유효한 클라이언트로 인증하는 단계를 포함하여 이루어진다. 상기 (b)단계의 t_v는, 서버가 위수가 q인 곱셈그룹 Zq^*로부터 임의로 선택함에 의해 생성되며, 상기 (b)단계의 z 및 세션키(sk)의 생성은, 상기 클라이언트가 유효한 클라이언트로 인증되면, $z = H_2(t_u, t_v, X, Y, ID_u, ID_v)$, $sk = H_3(t_u, t_v, X, Y, ID_u, ID_v)$에 의해 이루어진다. 상기 (c)단계의 서버 인증은, ID_U, ID_V, 클라이언트가 생성한 t_u, X, Y, 상기 서버로부터 받은 t_v를 해쉬함수한 값 $z'(= H_2(t_u,t_v,X,Y,ID_U, ID_V)$를 생성하는 단계; 및 상기 z'가 z와 일치(z'= z)하는지 체크하는 단계를 포함하여 이루어진다. 상기 (c)단계의 세션키(sk) 생성은 상기 서버가 유효한 서버로 인증되면, $sk = H_3(t_u, t_v, X, Y, ID_u, ID_v)$에 의해 이루어진다.

상기 (b)단계의 클라이언트 인증에서 $e(Y, Q_U)$이 t_ug^h와 동일한 값이 아니면 클라이언트 인증 실패 메시지를 클라이언트에게 전송함이 바람직하다. 상기 (c)단계의 서버 인증에서 z 와 z' 이 동일한 값이 아니면 서버 인증 실패 메시지를 서버에게 전송함이 바람직하다.

상술한 기술적 과제를 해결하기 위한 본 발명에 의한 ID-기반 키교환 장치는, 위수가 q인 덧셈그룹(G1)의 생성자를 P, 키생성센터(KGC)의 마스터 비밀키를 s, 상기 s와 P의 곱을 Ppub(= sP), 상기 P의 pairing 연산값을 g(=e(P,P)), 클라이언트 및 서버의 ID를 각각 ID_u 및 ID_v, 키생성센터로부터 받은 클라이언트 및 서버의 비밀키를 각각 S_u 및 S_v 라 할 때, 클라이언트와 서버에 설치되며, X 값 및 Y 값을 생성하는 X&Y 생성부; 클라이언트 측에 설치되며, 상기 ID_u, X 및 Y를 서버로 전송하는 클라이언트전송부; 클라이언트에 설치되며, 서버로부터 z, t_v를 받아 서버를 인증하는 클라이언트 인증부; 유효한 서버로 인증되면 상기 서버의 세션키와 동일한 세션키를 생성하는 클라이언트세션키 생성부; 서버에 설치되며, 상기 ID_u, X 및 Y를 받아 클라이언트를 인증하는 서버 인증부; 서버에 설치되며, 클라이언트가 유효한 크라이언트로 인증되면 z 및 t_v를 생성하는 $z\&t_v$ 생성부; 상기 z 및 t_v를 클라이언트로 전송하는 서버전송부; 및 서버에 설치되며, 클라이언트가 유효한 클라이언트로 인증되면 세션키(sk)를 생성하는 서버 세션키 생성부를 포함함을 특징으로 한다.

상기 X&Y의 생성부는 위수가 q인 곱셈그룹 Zq^*로부터 임의의 값 a를 선택하는 a 선택부; 상기 ID_V 의 해쉬함수 값 $q_v(= H(ID_V))$를 생성하고, 상기 P_{pub} 와 q_vP를 합하여 $Qv(= P_{pub} + q_vP)$를 생성하고, 상기 g의 a 멱승을 취하여 $t_u(= g^a)$를 생성하고, 상기 t_u의 해쉬함수 값 $h(=H_1(t_u))$를 생성하는 매개값생성부; 상기 a와 Q_v의 곱을 계산하여 $X(=aQ_V)$를 생성하는 X생성부; 및 상기 a와 h의 합에 S_U 를 곱하여 $Y(=(a+ h)S_U)$를 생성하는 Y생성부를 포함하여 이루어진다.

상기 클라이언트 인증부는, ID_U, ID_V, 클라이언트가 생성한 t_u, X, Y, 상기 서버로부터 받은 t_v를 해쉬함수한 값 $z'(= H_2(t_u,t_v,X,Y,ID_U, ID_V)$를 생성하는 z'생성부 및 상기 z'가 상기 z와 일치(z'= z)하는지 체크하여 서버를 인증하는 z'인증부를 포함한다. 상기 클라이언트 세션키 생성부는, 상기 서버가 유효한 서버로 인증되면, t_u, t_v, X, Y, ID_u, ID_v의 해쉬함수 값 $(sk = H_3(t_u, t_v, X, Y, ID_u, ID_v))$을 생성한다. 서버 인증부는 상기 ID_U 의 해쉬함수 값 $q_u(= H(ID_U))$를 생성하고, 상기 Ppub 를 q_u와 P의 곱(q_uP)과 합하여 $Q_U(= Ppub + q_uP)$를 생성하고, 클라이언트로부터 받은 X와 S_V의 pairing 연산하여

t_u(= e(X, S_V))를 생성하고, 클라이언트로부터 받은 Y와 상기 Q_U 를 pairing 연산하여 c(= e(Y, Q_U))를 생성하는 c생성부; 상기 t_u 의 해쉬함수 값 h(=H_1(t_u))을 생성하여, 상기 t_u와 g의 h 멱승(g^h)을 곱한 값($t_u g^h$)을 생성하는 $t_u g^h$생성부; 및 상기 c와 상기 $t_u g^h$를 비교하여 그 값이 동일하면 유효한 클라이언트로 인증하는 c 인증부를 포함하여 이루어진다.

상기 z&t_v생성부는, 서버가 위수가 q인 곱셈그룹 Zq^*로부터 임의로 t_v를 선택하는 t_v선택부; 및 상기 클라이언트가 유효한 클라이언트로 인증되면, t_u, t_v, X, Y, ID_u, ID_v의 해쉬함수 값 z(= H_2(t_u, t_v, X, Y, ID_u, ID_v))를 생성하는 z생성부를 포함한다. 상기 서버세션키 생성부는 상기 클라이언트가 유효한 클라이언트로 인증되면, t_u, t_v, X, Y, ID_u, ID_v를 해쉬함수한 값 sk(= H_3(t_u, t_v, X, Y, ID_u, ID_v))를 생성한다.

상기 클라이언트 인증부에서 e(Y, Q_U)이 $t_u g^h$와 동일한 값이 아니면 클라이언트 인증 실패 메시지를 클라이언트에게 전송함이 바람직하다. 상기 서버 인증부에서 z 와 z' 이 동일한 값이 아니면 서버 인증 실패 메시지를 서버에게 전송함이 바람직하다.

그리고 상기 기재된 발명을 컴퓨터에서 실행시키기 위한 프로그램을 기록한 컴퓨터로 읽을 수 있는 기록매체를 제공한다.

이하, 첨부된 도면을 참조하여 본 발명에 의한 ID-기반 키교환 방법 및 장치를 상세히 설명하기로 한다.

먼저, Bilinear Maps과 암호학적 가정들을 설명하기로 한다. 본 발명에서 G_1 을 위수가 소수q인 덧셈 연산 군이라 하고, G_2를 같은 위수 q를 갖는 곱셈 연산 군이라 하자. 그리고 P는 G_1의 생성자이다. 이 때 G_1,G_2에서의 이산 대수 문제(DLP)는 어렵다고 가정한다. 아래와 같은 조건을 만족하는 함수 e : G_1 x G_1 -> G_2를 admissible bilinear map 이라 한다.

Bilinear : 모든 P,Q ∈ G_1 이고 a,b ∈ Zq^* 에 대하여 $e(aP, bP) = e(P,P)^{ab}$

Non-degenerate : e(P,P)≠1을 만족하는 P ∈ G_1 가 존재한다.

Computable : 모든 P,Q ∈ G_1에 대하여 e(P,P)을 계산할 수 있는 효율적인 알고리즘이 존재한다.

Computational Diffie-Hellman (CDH) problem: CDH 문제는 a,b ∈ Zq^*인 P, aP, bP가 주어지면 abP를 계산하는 문제이다.

Inverse Computational Diffie-Hellman (ICDH) problem: ICDH 문제는 a∈ Zq^*인 P, aP가 주어지면 $a^{-1}P$를 계산하는 문제이다.

Modified Inverse Computational Diffie-Hellman (mICDH) problem: mICDH 문제는 a,b ∈ Zq^*인 b, P, aP 그리고 (a+ b)P가 주어지면 $(a+ b)^{-1}P$를 계산하는 문제이다.

Bilinear Diffie-Hellman (BDH) problem: BDH 문제는 a,b,c∈Zq^*인 P, aP, bP 그리고 cP가 주어지면 $e(P,P)^{abc}$를 계산하는 문제이다.

Bilinear Inverse Diffie-Hellman (BIDH) problem: BIDH 문제는 a,b,c∈Zq^*인 P, aP, bP 가 주어지면 $e(P,P)^{a^{-1}c}$를 계산하는 문제이다.

Modified Bilinear Inverse Diffie-Hellman (mBIDH) problem: mBIDH 문제는 a,b,c∈Zq^*인 b,P,aP 그리고 cP가 주어지면 $e(P,P)^{(a+b)^{-1}c}$ 를 계산하는 문제이다.

위 문제 중 CDH와 ICDH 그리고 mICDH 문제는 서로 다항식 시간(polynomial time) 동치임을 쉽게 알 수 있다.

정 리 1. BDH와 BIDH 그리고 mBIDH 문제는 서로 다항식 시간 동치이다.

(증명) BDH와 BIDH 문제는 다항식 시간 동치이다. 그러므로 단지 BIDH와 mBIDH가 서로 다항식 시간 동치임을 증명한다.

[BIDH ⇒ mBIDH] b,P,aP 그리고 cP가 mBIDH에 주어지면 a'P = aP + bP를 계산하여 P, a'P, cP를 BIDH에 입력한다. BIDH가 $e(P,P)^{a^{-1}c}=e(P,P)^{(a+b)^{-1}c}$ 를 출력하면 mBIDH 역시 같은 값을 출력한다.

[mBIDH ⇒ BIDH] P, aP, cP가 BIDH에 주어지면 a'P = aP - bP를 계산하여 b, P, a'P 그리고 cP를 mBIDH에 입력한다. mBIDH가 $e(P,P)^{(a'+b)^{-1}c}=e(P,P)^{a^{-1}c}$ 를 출력하면 BIDH 역시 같은 값을 출력한다.

CDH, BDH 그리고 BIDH 문제가 어렵다고 가정한다. 이것은 CDH, BDH 그리고 BIDH 문제를 다항식 시간 안에 의미있는 확률을 가지고 계산하는 알고리즘이 존재하지 않음을 의미한다. gap Diffie-Hellman (GDH) 가정을 만족하는 GDH 파라미터 생성기는 타원곡선 상에서의 Weil/Tate pairing으로 구성할 수 있다.

본 발명의 안전성을 증명하기 위하여 k-CAA (Collusion Attack Algorithm with k traitor) 문제와 k-mBIDH (modified BIDH with k values) 문제를 정의한다. 사실 k-mBIDH는 k-CAA 문제를 bilinear 형태로 변환한 문제이다.

정 의 1. k-CAA 문제는 아래와 같이 주어졌을 때 $h \in Zq^*$에 대하여 $(s+h)^{-1}P$를 계산하는 문제이다.

$$P, sP, \ h_1, h_2, \ldots, h_k \in Z_q^*, (s+h_1)^{-1}P, (s+h_2)^{-1}P, \ldots, (s+h_k)^{-1}P$$

정 의 2. k-mBIDH 문제는 아래와 같이 주어졌을 때 $e(P,P)^{(s+h)^{-1}t}$를 계산하는 문제이다.

$$P, sP, tP, \ h, h_1, h_2, \ldots, h_k \in Z_q^*, (s+h_1)^{-1}P, (s+h_2)^{-1}P, \ldots, (s+h_k)^{-1}P.$$

알고리즘 A에 대한 좀 더 일반적인 이점은 다음과 같이 정의된다.

$$\left|\Pr\left[A\left(\begin{array}{c} P, sP, tP, h, h_1, \ldots, h_k, \\ (s+h_1)^{-1}P, (s+h_2)^{-1}P, \ldots, (s+h_k)^{-1}P \end{array}\right) = e(P,P)^{(s+h)^{-1}t} \;\middle|\; \begin{array}{c} s,t,h,h_1,\ldots,h_k \leftarrow Z_q^*; \\ P \leftarrow G_1;\ h \neq h_i \end{array}\right]\right|.$$

다음으로 본 발명에 의한 ID기반 키교환 방법 및 장치의 안전성을 분석하기로 한다.

authenticryption 기법을 이용하여 모바일 클라이언트 U와 서버 V 사이의 ID 기반의 인증된 키 교환 방법 및 장치를 설명하고 랜덤 오라클 모델에서의 안전성을 증명한다. 본 발명에 적용된 키교환 방법을 ID-AKA라 명한다.

1. ID-기반의 인증된 키 교환 프로토콜

먼저 두 그룹 G_1,G_2와 bilinear map e를 GDH 생성기를 사용하여 생성한다. G_1 에서 mICDH 문제는 어렵다고 가정한다. 본 발명에 의한 ID-AKA 방법은 주어진 공개 확인자에 대응되는 비밀키를 생성하는 신뢰기관인 키 생성 기관 (KGC)을 이용하며, 이때 생성된 비밀키에 대한 안전성은 mICDH 문제의 어려움에 기반하게 된다.

Setup은 다음과 같이 이루어진다.

KGC는 $s \in Zq^*$ 와 G_1의 생성자 P를 선택하고 $P_{pub} = sP$를 계산한다. 또한 암호학적 일방향 해쉬 함수 $H: \{0,1\}^* \rightarrow Z_q^*$, $H_1 : G_2 \rightarrow Z_q^*$, $H_2 : \{0,1\}^* \rightarrow \{0,1\}^t$ 그리고 $H_3 : \{0,1\}^* \rightarrow \{0,1\}^k$를 선택한다. 여기서 t 는 보안상수(security parameter)이며, k는 세션키의 비트 길이이다. 그 다음에 KGC는 마스터 비밀키인 s를 비밀로 하고 공개 시스템 파라미터 $params = \{e, G_1, G_2, q, P, P_{pub}, H, H_1, H_2, H_3\}$를 공개한다.

Extract는 다음과 같이 이루어진다.

확인자가 ID_u인 클라이언트 U가 비밀키를 얻기를 원할 때, KGC는 $q_u = H(ID_U)$와 $g = e(P,P)$ 를 계산하여 비밀키 $S_U = (s + q_U)^{-1} P$를 계산한 후 (g, S_U)를 안전한 채널로 U에게 전송한다. 동일한 방법으로, 확인자가 ID_v인 서버 V도 $q_v = H(ID_V)$를 계산한 후, 비밀키 $S_V = (s + q_V)^{-1} P$와 g=e(P,P) 를 계산하여 안전한 채널로 V에게 전송한다.

도 1은 본 발명에 의한 ID-기반 키교환 장치의 구성을 블록도로 도시한 것으로서, X&Y생성부(100), 클라이언트 전송부(110), 클라이언트 인증부(160), 클라이언트 세션키 생성부(170), 서버인증부(120), z&t_v생성부(130), 서버전송부(140) 및 서버 세션키 생성부(150)를 포함하여 이루어진다.

먼저, 위수가 소수 q인 gap DiffieHellman 그룹, 예를 들어 위수가 소수 q인 덧셈그룹(G_1)의 생성자를 P, 키생성센터(KGC)의 마스터 비밀키를 s, 상기 s와 P의 곱을 Ppub(= sP), 상기 P의 pairing 연산값을 g(=e(P,P)), 클라이언트 및 서버의 ID를 각각 ID_u 및 ID_v, 키생성센터로부터 받은 클라이언트 및 서버의 비밀키를 각각 S_u 및 S_v 라 한다.

상기 X&Y생성부(100)는 상기 클라이언트에 설치되며, X 값 및 Y 값을 생성한다. 도 2는 상기 X&Y의 생성부(100)의 세부적인 구성을 블록도로 도시한 것으로서, a선택부(200), 매개값 생성부(220), X생성부(240) 및 Y생성부(260)를 포함하여 이루어진다. 상기 a 선택부(200)는 위수가 q인 곱셈그룹 Zq^*로부터 임의의 값 a를 선택한다. 상기 매개값 생성부(220)는 상기 ID_V 의 해쉬함수 값 q_v(= $H(ID_V)$)를 생성하고, 상기 P_{pub} 와 q_vP를 합하여 Q_v(= P_{pub} + q_vP)를 생성하고, 상기 g의 a 멱승을 취하여 t_u(= g^a)를 생성하고, 상기 t_u의 해쉬함수 값 h(=$H_1(t_u)$)를 생성한다. 상기 X생성부(240)는 상기 a와 Q_v의 곱을 계산하여 X(=aQ_V)를 생성한다. 상기 Y생성부(260)는 상기 a와 h의 합에 S_U 를 곱하여 Y(=(a+ h)S_U)를 생성한다.

상기 클라이언트 전송부(110)는 클라이언트 측에 설치되며, 상기 ID_u, X 및 Y를 서버로 전송한다.

클라이언트 인증부(160)는 클라이언트에 설치되며, 서버로부터 z, t_v를 받아 서버를 인증한다. 도 5는 상기 클라이언트 인증부(160)의 세부적인 구성을 블록도로 도시한 것으로서, z'생성부(500) 및 z 인증부(550)를 포함하여 이루어진다. 상기 z'생성부(500)는 ID_U, ID_V, 클라이언트가 생성한 t_u, X, Y, 상기 서버로부터 받은 tv를 해쉬함수한 값 z'(= H_2 (tu,tv,X,Y,ID_U, ID_V))를 생성한다. 상기 z 인증부(550)는 상기 z'가 상기 z와 일치(z'= z)하는지 체크하여 서버를 인증한다.

클라이언트 세션키 생성부(170)는 유효한 서버로 인증되면 상기 서버의 세션키와 동일한 세션키를 생성한다. 보다 구체적으로 설명하면, 상기 클라이언트 세션키 생성부(170)는 상기 서버가 유효한 서버로 인증되면, t_u, t_v, X, Y, ID_u, ID_v의 해쉬함수 값(sk = H_3(t_u, t_v, X, Y, ID_u, ID_v))을 생성한다.

서버인증부(120)는 서버에 설치되며, 상기 ID_u, X 및 Y를 받아 클라이언트를 인증한다. 도 3은 서버 인증부(120)의 세부구성을 블록도로 도시한 것으로서, c생성부(300), $t_u g^h$생성부(320) 및 c인증부(340)를 포함하여 이루어진다.

상기 c생성부(300)는 상기 ID_U 의 해쉬함수 값 q_u(= $H(ID_U)$)를 생성하고, 상기 Ppub 를 q_u와 P의 곱(q_uP)과 합하여 Q_U(= Ppub + q_uP)를 생성하고, 클라이언트로부터 받은 X와 S_V의 pairing 연산하여 t_u(= e(X, S_V))를 생성하고, 클라이언트로

부터 받은 Y와 상기 Q_U 를 pairing 연산하여 c(= e(Y, Q_U))를 생성한다. 상기 $t_u g^h$생성부(320)는 상기 t_u 의 해쉬함수 값 h(=$H_1(t_u)$)을 생성하여, 상기 t_u와 g의 h 멱승(g^h)을 곱한 값($t_u g^h$)을 생성한다. 상기 c인증부(340)는 상기 c와 상기 $t_u g^h$ 를 비교하여 그 값이 동일하면 유효한 클라이언트로 인증한다.

z&t_v 생성부(130)는 서버에 설치되며, 클라이언트가 유효한 클라이언트로 인증되면 z 및 t_v를 생성한다. 도 4는 상기 z&t_v생성부(130)의 세부 구성을 블록도로 도시한 것으로서, t_v선택부(400) 및 z생성부(450)를 포함하여 이루어진다. 상기 t_v선택부(400)는 서버가 위수가 q인 곱셈그룹 Zq^*로부터 임의로 t_v를 선택한다. 상기 z생성부(450)는 상기 클라이언트가 유효한 클라이언트로 인증되면, t_u, t_v, X, Y, ID_u, ID_v의 해쉬함수 값 z(= $H_2(t_u, t_v, X, Y, ID_u, ID_v)$)를 생성한다.

서버전송부(140)는 상기 z 및 t_v를 클라이언트로 전송한다.

서버 세션키 생성부(150)는 서버에 설치되며, 클라이언트가 유효한 클라이언트로 인증되면 세션키(sk)를 생성한다. 즉, 상기 서버세션키 생성부(150)는 상기 클라이언트가 유효한 클라이언트로 인증되면, t_u, t_v, X, Y, ID_u, ID_v를 해쉬함수한 값 sk(= $H_3(t_u, t_v, X, Y, ID_u, ID_v)$)를 생성한다.

한편, 클라이언트 U와 서버 V 가 세션키를 생성하는 본 발명에 의한 ID-기반 키교환 방법을 설명하기로 한다. 도 6은 본 발명에 의한 ID-기반 키교환 방법을 설명하기 위한 흐름도를 도시한 것이다. 도 6을 참조하면, 클라이언트 U는 $q_v = H(ID_V)$를 계산하고 난수 a∈Zq^* 를 선택하여 $t_u=g^a$와 Qv = Ppub + q_vP를 계산한다. 그 다음으로 U는 $h = H_1(t_u), X = aQ_V$ 그리고 $Y=(a+h)S_U$를 계산하여 $\langle ID_U, (X,Y)\rangle$를 V에게 전송한다.

서버 V는 q_u= H(ID_u)와 Q_U = P_{pub} + q_uP를 계산한 후, $t_u=g^a$와 c=e(X,Sv)를 전송받은 메시지와 자신의 비밀키 S_v를 이용하여 계산한다. 또한 h=$H_1(t_u)$를 계산하여 c = $t_u g^h$ 임을 확인한다. 만약 식이 성립하지 않으면 V는 실패를 출력하고, 식이 성립하면 난수 t_v∈ Zq^*를 선택하여 인증 메시지 $z = H_2(t_u, t_v, X, Y, ID_U, ID_V)$를 계산한 후, z와 t_v를 U에게 전송한다. 결국 서버 V는 세션키로 $sk = H_3(t_u, t_v, X, Y, ID_U, ID_V)$를 계산한다.

클라이언트 U는 $z' = H_2(t_u, t_v, X, Y, ID_U, ID_V)$을 t_u와 t_v를 이용하여 계산한 후, z=z'임을 확인한다. 만약 식이 성립하지 않으면 U는 실패를 출력하고, 식이 성립하면 세션키 $sk = H_3(t_u, t_v, X, Y, ID_U, ID_V)$를 계산한다.

위의 프로토콜에서 클라이언트 U는 t_u와 Y를 사전에 계산할 수 있다. 그러면 온-라인 상에서 U는 한번의 포인트 덧셈과 두 번의 스칼라 곱셈만을 계산하면 된다. 더구나 해쉬함수 H는 계산량이 많은 Map-To-Point 연산이 아니라 일반적인 해쉬함수이기 때문에 계산량은 매우 적다고 할 수 있다.

이를 보다 상세히 설명하면 다음과 같다. 먼저, 위수가 소수 q인 gap DiffieHellman 그룹, 예를 들어 위수가 q인 덧셈그룹(G1)의 생성자를 P, 키생성센터(KGC)의 마스터 비밀키를 s, 상기 s와 P의 곱을 P_{pub}(= sP), 상기 P의 pairing 연산값을 g (=e(P,P)), 클라이언트 및 서버의 ID를 각각 ID_u 및 ID_v, 키생성센터로부터 받은 클라이언트 및 서버의 비밀키를 각각 S_u 및 S_v 라 하자.

(a) 클라이언트가 ID_u, X 및 Y를 서버에게 전달한다. 여기서, 상기 X 및 Y의 생성은 다음과 같이 이루어진다. 위수가 q인 곱셈그룹 Zq^*로부터 임의의 값 a를 선택한다. 그 다음에 상기 ID_V 의 해쉬함수 값 q_v(= H(ID_V))를 생성하고, 상기 P_{pub} 와 q_vP를 합하여 Qv(= P_{pub} + q_vP)를 생성하고, 상기 g의 a 멱승을 취하여 t_u(= g^a)를 생성하고, 상기 t_u의 해쉬함수 값 h (=$H_1(t_u)$)를 생성한다. 그리고 나서 상기 a와 Q_v의 곱을 계산하여 X(=aQ_V)를 생성하고, 상기 a와 h의 합에 S_U 를 곱하여 Y(=(a+ h)S_U)를 생성한다.

(b) 서버는 상기 ID_u, X 및 Y를 받아 클라이언트를 인증하고, 인증이 되면 z 및 t_v를 생성하여 클라이언트에게 전달하고 세션키(sk)를 생성한다. 여기서, 상기 클라이언트 인증은 다음과 같이 이루어진다. 상기 ID_U 의 해쉬함수 값 q_u(= $H(ID_U)$)를 생성하고, 상기 P_{pub} 를 q_u와 P의 곱(q_uP)과 합하여 Q_U(= P_{pub} + q_uP)를 생성하고, 클라이언트로부터 받은 X와 S_V의 pairing 연산하여 t_u(= $e(X, S_V)$)를 생성하고, 클라이언트로부터 받은 Y와 상기 Q_U 를 pairing 연산하여 c(= $e(Y, Q_U)$)를 생성한다. 그 다음에 상기 t_u 의 해쉬함수 값 h(=$H_1(t_u)$)을 생성하여, 상기 t_u와 g의 h 멱승(g^h)을 곱한 값(t_ug^h)을 생성한다. 그리고 나서 상기 $c = t_ug^h$를 만족하면 유효한 클라이언트로 인증한다. 그리고 상기 t_v는, 서버가 위수가 q인 곱셈그룹 Zq^*로부터 임의로 선택함에 의해 생성된다. 상기 z 및 세션키(sk)의 생성은, 상기 클라이언트가 유효한 클라이언트로 인증되면, $z = H_2(t_u, t_v, X, Y, ID_u, ID_v)$, $sk = H_3(t_u, t_v, X, Y, ID_u, ID_v)$에 의해 이루어진다.

(c) 클라이언트는 서버로부터 z, t_v를 받아 서버를 인증하고, 유효한 서버로 인증되면 상기 서버의 세션키와 동일한 세션키를 생성한다. 여기서, 상기 서버 인증은 다음과 같이 이루어진다. 먼저, ID_U, ID_V, 클라이언트가 생성한 t_u, X, Y, 상기 서버로부터 받은 t_v를 해쉬함수한 값 z'(= $H_2(t_u,t_v,X,Y,ID_U, ID_V)$를 생성한다. 그 다음에 상기 z'가 z와 일치(z'= z)하는지 체크한다. 상기 세션키(sk)는, 상기 서버가 유효한 서버로 인증되면, $sk = H_3(t_u, t_v, X, Y, ID_u, ID_v)$에 의해 생성된다.

본 발명은 절반의 전방향 안전성을 제공한다. 만약 서버의 비밀키가 유출되면 공격자는 전송 메시지로부터 모든 세션키를 계산할 수 있지만, 클라이언트의 비밀키가 유출되면 이를 이용하는 공격자는 이전 세션키들에 관한 어떠한 정보도 얻을 수 없게 된다. 일반적으로 모바일 클라이언트는 낮은 파워를 가진 장치이기 때문에 서버에 비하여 공격에 취약하게 된다. 따라서 서버보다 클라이언트에서의 전방향 안전성의 제공이 더욱 효과적이라 할 수 있을 것이다.

본 발명의 안전성을 분석하면 다음과 같다. 본 발명의 안전성 분석을 위해 ID-AKA에서 클라이언트의 전송 메시지를 위조하는 공격자를 Forger라 하자. 먼저 주어진 ID_u와ID_v 로부터 클라이언트의 전송 메시지의 위조가 불가능함을 보인다.

보조정리 1. 해쉬함수 H와 H_1을 랜덤 오라클이라 하고 Forger A를 ID_u와 ID_v가 주어지고 러닝(running) 타임 t_0와 ε_0의 성공확률을 가진 위조자라 가정하자. A는 각각 q_E, q_{E_1}, q_S 그리고 q_E 개의 H, H_1, 전송 그리고 추출질의를 한다고 하자. 만약 $\epsilon_0 \geq 10q_{E_1}^2(q_S+1)(q_S+q_E)/q$라 하면 러닝 타임이 $t_1 \leq 120686q_{E_1}t_0/\epsilon_0$ 인 k-CAA문제를 계산하는 공격자 B가 존재한다.

(증명) 먼저 B에게 $k \geq q_E, q_S$인 k-CAA 문제의 입력값 $P, sP, q_1, q_2, \ldots, q_k, (s+q_1)^{-1}P, (s+q_2)^{-1}P, \ldots, (s+q_k)^{-1}P$이 주어진다. B의 목적은 임의의 q_0에 대하여 $(s+q_0)^{-1}P$를 계산하는 것이다. B는 A를 이용하기 위해 A의 공격 환경을 시뮬레이션해 준다. 먼저 B는 GDH 파라미터 $\langle e, G_1, G_2\rangle$를 생성하고 Ppub=sP와 g=e(P,P)를 계산하여 시스템 파라미터 $\langle e, G_1, G_2, P, P_{pub}, g\rangle$ 를 A에게 입력한다.

일반성을 잃지 않고, 주어진 ID에 대하여 A는 H, H_1, 전송 그리고 추출질의를 한번만 하며, 전송과 추출질의 사전에 질의를 수행한다고 가정하자. 충돌을 방지하고 질의에 대한 일관성 있는 응답을 위해 B는 목록 L_H과 L_{H1}을 유지한다. 이 목록들은 초기에 비어있다. B는 A의 질의에 대하여 다음과 같이 응답한다.

H-질의. A가 ID_i에 대한 H-질의를 했을 때, 만약 $ID_i = ID_V$이면 B는 q_0를 전송하고, 그 외의 경우에는 $<ID_i,q_i>$ 를 목록 L_H에 추가하고 q_i를 전송한다.

H_1-질의. A가 m에 대한 H_1-질의를 했을 때, B는 난수 h를 전송하고 이를 목록 L_{H1}에 추가한다.

전송-질의. A가 전송( ) 질의를 했을 때, 만약 $ID_i = ID_U$이면 B는 난수 a와 h를 선택하여 X=a(sP+ q_0P)와 Y=hP를 계산한 후, ID_u(X,Y)를 A에게 전송한다. 그 외의 경우 B는 목록 L_H에서 $<ID_i, q_i>$를 찾은 후, 난수 a와 h를 선택하고 X=a

$(sP+q_0P)$와 $Y=(a+h)(q_i+s)^{-1}P$를 계산하여 $ID_u(X,Y)$를 A에게 전송한다. 만약 A의 전송 질의가 $ID_i=ID_u$이면 B는 정당한 Y값을 생성할 수 없기 때문에 임의의 Y값을 전송하게 된다. 그러나 A는 서버의 비밀키 S_v를 알지 못하기 때문에 전송받은 (X,Y)에 대한 정당성을 확인할 수 없다. 따라서 전송 질의에 대한 B의 시뮬레이션은 타당하다.

추출-질의. A가 $ID_i \notin \{ID_U, ID_V\}$에 대한 추출 질의를 했을 때, B는 목록 L_H에서 $<ID_i, q_i>$를 찾은 후, $(s+q_i)^{-1}P$를 전송한다.

마침내 A는 새로운 정당한 메시지 $ID_u(X,Y)$를 출력하게 된다. B는 forking lemma 와 같이 위의 H_1과 다른 해쉬함수 H'_1를 사용하여 동일하게 시뮬레이션 하면 결국 A 는 $h \neq h'$인 서로 다른 정당한 두 메시지 쌍 $\langle ID_V, X=aQ_V, Y=(a+h)(q_0+s)^{-1}P \rangle$와 $\langle ID_V, X=aQ_V, Y'=(a+h')(q_0+s)^{-1}P \rangle$을 출력하게 된다. 그러면 B는 $(Y-Y')/(h-h')=(q_0+s)^{-1}P$를 계산한 후 이를 k-CAA 문제의 결과로 출력한다.

B가 h와 h'을 정확히 추측할 확률은 $1/q^2_{H1}$이며, 전체 러닝 타임 t_1은 forking lemma에 요구된 러닝 타임과 동일한 $120686 q_{H_1} t_0 / \epsilon_0$에 유계하게 된다.

정 리 2. 제안된 ID-AKA에 사용된 해쉬함수를 랜덤 오라클이라 하고 ID_U와 ID_V가 주어지고 러닝(running) 타임 t인 ID-AKA 공격자 A가 존재한다고 가정하자. 그러면 ID-AKA는 k-mBIDH 문제의 어려움에 기반하여 절반의 전방향 안전성을 제공하는 안전한 AKA 프로토콜이다. 즉,

$$Adv^{AKA-hfs}_{ID-AKA,A}(t) \leq \frac{1}{2} q_S q_{H_1} Adv^{k-mBIDH}_{G_1,G_2,e}(t) + Adv^{Forge}(t)$$

여기서 $Adv^{Forge}(t)$는 러닝 타임이 t인 임의의 Forger의 최대 이점이며, q_S, q_C그리고 q_{H_i}는 공격자 A가 생성하는 전송, 손상 그리고 해쉬 Hi 질의의 개수를 나타낸다.

(증명) A를 ID-AKA를 공격하는 능동적인(active) 공격자라 하자. 그러면 A는 두 가지 경우에 의한 공격을 시도할 수 있다. 첫 번째는 인증 메시지들의 위조. 즉, 클라이언트로의 가장 공격을 시도하거나. 두 번째는 전송 메시지의 위조나 변조하지 않고 도청만 하여 공격하는 경우이다. 공격자가 서버의 전송 메시지를 위조하는 것, 즉, 서버로 가장하는 공격은 결국 z값을 위조해야만 하고 이는 클라이언트의 비밀 정보인 t_u값을 모르고서는 성공할 수 없다. 그러나 해쉬함수 H_2의 일방향성에 의해 z값의 위조는 불가능하다. 따라서 공격자 A의 공격은 위의 두 가지 경우로 제한된다.

먼저 A가 첫 번째 경우의 공격을 시도한다면 A를 이용하여 정당한 메시지 쌍 $ID_V(X,Y)$을 위조하는 Forger F를 구성할 수 있다. 이는 F가 시스템에 필요한 모든 공개키와 비밀키를 올바르게 생성한 후, A의 질의에 대해 시뮬레이션 해줌으로써 쉽게 구성할 수 있다. Forger를 A가 새로운 정당한 메시지 쌍을 생성하는 사건이라 하면 F가 성공할 확률은 $\Pr_A[Forge] \leq Adv^{Forge}_F(t) \leq Adv^{Forge}(t)$가 된다. 보조정리 1.에 의하여 확률 $\Pr_A$[Forge]은 무시할 만한 값이다.

다음으로 A가 두 번째 경우의 공격을 시도할 때를 고려해보자. 세션키 sk의 정보를 얻기 위해서 A는 해쉬 오라클 H_3에 $\langle t_u \| t_v \| z \| \Im \| ID_U \| ID_V \rangle$를 질의해야만 한다. 따라서 A는 비밀 정보인 t_u를 계산해야만 한다. 그러므로 A를 이용하여 k-mBIDH 문제를 계산하는 공격자 B를 다음과 같이 구성할 수 있다. 먼저 B에게 $k \geq q_E q_S$인 k-mBIDH 문제의 입력값 $e, G_1, G_2, P, sP, tP, q_0, q_1, q_2, \ldots, q_k, (s+q_1)^{-1}P, (s+q_2)^{-1}P, \ldots, (s+q_k)^{-1}P$이 주어진다. B의 목적은 $e(P,P)^{(s+q_0)^{-1}t}$를

계산하는 것이다. B는 A를 이용하기 위해 A의 공격 환경을 시뮬레이션 해 준다. 먼저 B는 GDH 파라미터 <e,G_1,G_2>를 생성하고 Ppub=sP와 g=e(P,P)를 계산하여 시스템 파라미터 $\langle e, G_1, G_2, P, P_{pub}, g \rangle$를 A에게 입력한다. A의 이점을 활용하기 위해 B는 임의의 α를 선택한다. α는 A가 α번째 세션에 테스트 질의를 하는 것을 추측하기 위한 값이다.

일반성을 잃지 않고, A는 동일한 메시지에 대해서는 한번만 질의 하며, 전송과 추출질의 사전에 H질의를 수행한다고 가정하자. 충돌을 방지하고 질의에 대한 일관성 있는 응답을 위해 B는 목록 L_H, L_{H1}와 L_{H2}을 유지한다. 이 목록들은 초기에 비어있다. B는 A의 질의에 대하여 다음과 같이 응답한다.

H-질의. A가 ID_i에 대한 H-질의를 했을 때, 만약 ID_i=ID_v이면 B는q_0 를 전송하고, 그 외의 경우에는 <ID_i, q_i >를 목록 L_H에 추가하고 q_i를 전송한다.

H_1-질의. A가 목록 L_{H1}에 기록된 (m,h)와 동일한 메시지 m에 대한 H_1-질의를 하면 B는 h를 전송한다. 그 외의 경우에 B는 난수 h를 전송하고 (m,h)을 L_{H1}에 추가한다.

H_2-질의. A가 H_2-질의를 했을 때, $\langle t_u \| t_v \| \Im \| ID_U \| ID_V \rangle$는 목록에서 $(\langle t_u \| t_v \| \Im \| ID_U \| ID_V \rangle, z)$를 찾아서 z를 A에게 전송한다.

H_3-질의. A가 메시지 m에 대한 H_3-질의를 하면 B는 임의의 난수를 A에게 전송한다.

전송-질의. 전송 질의는 클라이언트에서 서버로 그리고 서버에서 클라이언트로의 두 가지 전송 질의를 고려한다. 클라이언트에서 서버로의 질의는 전송S, 서버에서 클라이언트로의 질의는 전송C라 표기한다.

A가 전송$S(\Pi_U^s, Start)$ 질의를 했을 때, 만약 이 질의가 α번째 세션에 대한 요청이면, B는 난수 γ을 선택하고, X=tP와 Y= γP를 계산한 후, ID_V (X,Y)를 A에게 전송한다. 그 외의 경우 B는 목록 L_H에서 $\langle ID_i, q_i \rangle$를 찾은 후, 난수 α와 h를 선택하고 $Q_V = sP + q_0 P,\ X = aQ_V,\ Y = (a+h)(q_u + s)^{-1} P$ 그리고 $t_u = e(P,P)^a$ 를 계산하여 t_u와 h는 목록 L_{H1}에 추가하고 ID_V(X,Y)를 에게 전송한다.

A가 전송$C(\Pi_V^i, (ID_U, X, Y))$ 질의를 했을 때, B는 난수 z와 t_v를 선택하고 이를 A에게 전송한 후 $\langle (t_u, X, Y, ID_U, ID_V), z \rangle$를 목록 L_{H2}에 추가한다.

실행-질의. A가 실행(U,V) 질의를 했을 때, B는 전송 질의의 시뮬레이션을 이용하여 메시지 $\langle (ID_U, X, Y), (z, t_v) \rangle$를 A에게 전송한다.

추출-질의. A가 $ID_i \notin \{ID_U, ID_V\}$에 대한 추출 질의를 했을 때, B는 목록$L_H$에서 $\langle ID_i, q_i \rangle$를 찾은 후, $(s+q_i)^{-1} P$를 전송한다.

손상-질의. A가 ID_V에 대한 손상 질의를 했을 때, B는 목록 L_H 에서 $\langle ID_i, q_i \rangle$를 찾은 후, $(s+q_i)^{-1} P$를 전송한다.

유출-질의. A가 유출 질의를 했을 때, B는 난수를 A에게 전송한다.

테스트-질의. A가 테스트 질의를 했을 때, 만약 그 질의가 α번째 세션에 대한 질의이면 B는 실패를 출력하고, 그 외의 경우에는 임의의 비트 b를 선택한다. 만약 b=1이면 세션키를 아니면 난수를 A에게 전송한다.

B의 성공 확률은 B가 α를 정확히 추측한 사건과 A가 비밀 값 $t_u = e(P,P)^{(s+q_0)^{-1}t}$를 해쉬 오라클에 질의한 사건에 의존한다. 위의 시뮬레이션에서 B가 α를 정확히 추측할 확률은 $1/q_S$이고, A의 이점이 ε이라면 A가 비밀 값 t_u를 H_1 해쉬 오라클에 질의할 확률은 2ε이 된다. 따라서 B가 α를 정확히 추측하였다면 t_u는 2ε의 확률로 목록 L_{H1}에 존재하게 된다. 그러므로 B는 k-mBIDH 문제를 적어도 확률 $2\epsilon/q_S q_{H_1}$로 계산할 수 있게 된다. 결국 증명의 처음 부분에서 언급한 두 번째 사건에 대한 A의 이점은 최대 $\frac{1}{2}q_S q_{H_1} Adv^{k-mBIDH}_{G_1,G_2,e}(t)$ 만큼의 이점을 가지게 된다. 마침내 우리는 위의 증명의 결과로 아래와 같은 식을 얻을 수 있다.

$$Adv^{AKA-hfs}_{ID-AKA,A}(t) \le \frac{1}{2}q_S q_{H_1} Adv^{k-mBIDH}_{G_1,G_2,e}(t) + Adv^{Forge}(t)$$

본 발명은 컴퓨터로 읽을 수 있는 기록 매체에 컴퓨터(정보 처리 기능을 갖는 장치를 모두 포함한다)가 읽을 수 있는 코드로서 구현하는 것이 가능하다. 컴퓨터가 읽을 수 있는 기록 매체는 컴퓨터 시스템에 의하여 읽혀질 수 있는 데이터가 저장되는 모든 종류의 기록 장치를 포함한다. 컴퓨터가 읽을 수 있는 기록 장치의 예로는 ROM, RAM, CD-ROM, 자기 테이프, 플로피 디스크, 광데이터 저장장치 등이 있다.

본 발명은 도면에 도시된 실시예를 참고로 설명되었으나 이는 예시적인 것에 불과하며, 본 기술 분야의 통상의 지식을 가진 자라면 이로부터 다양한 변형 및 균등한 타 실시예가 가능하다는 점을 이해할 것이다. 따라서, 본 발명의 진정한 기술적 보호 범위는 첨부된 등록청구범위의 기술적 사상에 의해 정해져야 할 것이다.

발명의 효과

본 발명에 의한 ID-기반 키교환 방법 및 장치에 의하면, 컴퓨팅 파워가 다른 두 사용자(서버, 클라이언트) 사이의 효율적인 ID-기반의 인증된 키 교환을 할 수 있다. 특히 클라이언트 측의 계산량을 줄임으로써 낮은 파워를 가진 모바일 장치에 적합하다. 본 발명의 안전성은 랜덤 오라클 모델에서의 k-CAA와 k-mBIDH 문제의 어려움에 기반한다.

Bilinear map을 이용한 ID-기반 시스템은 Weil/Tate pairing 연산과 ID를 표현한 수를 타원곡선(elliptic) 위의 한 점으로 변환시키는 Map-To-Point 연산과 같은 계산량이 많은 복잡한 연산을 필요로 한다. 따라서 이러한 복잡한 연산 특히 pairing 연산을 줄이기 위하여 많은 연구가 되어 왔음에도 불구하고 여전히 pairing 연산은 타원곡선에서의 스칼라 곱에 비하여 계산량이 훨씬 많기 때문에 여전히 기존에 제안된 ID-기반의 키교환 기법은 저전력 모바일 장치에 사용하기엔 적합하지 않았다. 이렇게 대부분의 ID-기반의 AKA는 클라이언트 측의 무거운 계산량을 수행할 수 있는 유선 네트워크 환경에 기반하고 있다.

본 발명에 의한 ID-기반 AKA 역시 bilinear map을 사용한다. 그러나 다음과 같은 특징이 있다. 첫 번째, 클라이언트 측에서의 복잡한 pairing 연산과 Map-To-Point 연산을 사용하지 않았다. 두 번째, 오프라인 사전 계산을 사용하면 온-라인 상에서 클라이언트는 단지 타원곡선에서 두 번의 스칼라 곱과 한 번의 덧셈만 수행하면 된다.

본 발명은 키 교환과 지정된 확인자(서버)만이 주어진 메시지의 검증과 복호화를 할 수 있는 authenticryption의 개념을 이용한다.

본 발명은 서버의 고정된 비밀키가 노출되면 이전의 세션키가 폭로되지만, 클라이언트의 고정된 비밀키가 노출되어도 이전의 세션키가 안전한 절반의 전방향 안전성(half forward secrecy)을 제공한다. 또한 서로 다른 도메인 영역의 사용자들끼리의 키 교환, 즉 클라이언트는 자신과 다른 도메인 영역의 서버와도 안전한 채널을 형성할 수 있다. 더욱이 본 발명에 의한 프로토콜은 하나의 서버를 중심으로 여러 클라이언트가 안전한 채널을 형성할 수 있는 ID-기반의 인증된 그룹 키 교환으로의 확장이 가능하다.

(57) 청구의 범위

청구항 1.

위수가 소수 q인 gap DiffieHellman 그룹의 생성자를 P, 키생성센터(KGC)의 마스터 비밀키를 s, 상기 s와 P의 곱을 P_{pub} (= sP), 상기 P의 pairing 연산값을 g(=e(P,P)), 클라이언트 및 서버의 ID를 각각 ID_u 및 ID_v, 키생성센터로부터 받은 클라이언트 및 서버의 비밀키를 각각 S_u 및 S_v 라 할 때,

(a) 클라이언트가 ID_u, X 및 Y를 서버에게 전달하는 단계;

(b) 서버는 상기 ID_u, X 및 Y를 받아 클라이언트를 인증하고, 인증이 되면 z 및 t_v를 생성하여 클라이언트에게 전달하고 세션키(sk)를 생성하는 단계;

(c) 클라이언트는 서버로부터 z, t_v를 받아 서버를 인증하고, 유효한 서버로 인증되면 상기 서버의 세션키와 동일한 세션키를 생성하는 단계를 포함함을 특징으로 하고,

상기 (a)단계의 X 및 Y의 생성은

위수가 q인 곱셈그룹 Zq^*로부터 임의의 값 a를 선택하는 단계;

상기 ID_V 의 해쉬함수 값 $q_v(= H(ID_V))$를 생성하고, 상기 P_{pub} 와 q_vP를 합하여 $Q_v(= P_{pub} + q_vP)$를 생성하고, 상기 g의 a 멱승을 취하여 $t_u(= g^a)$를 생성하고, 상기 t_u의 해쉬함수 값 $h(=H_1(t_u))$를 생성하는 단계; 및

상기 a와 Q_v의 곱을 계산하여 $X(=aQ_V)$를 생성하고, 상기 a와 h의 합에 S_U 를 곱하여 $Y(=(a+ h)S_U)$를 생성하는 단계를 포함하여 이루어지며,

상기 (b)단계의 클라이언트 인증은

상기 ID_U 의 해쉬함수 값 $q_u(= H(ID_U))$를 생성하고, 상기 P_{pub} 를 q_u와 P의 곱(q_UP)과 합하여 $Q_U(= P_{pub} + q_UP)$를 생성하고, 클라이언트로부터 받은 X와 S_V의 pairing 연산하여 $tu(= e(X, S_V))$를 생성하고, 클라이언트로부터 받은 Y와 상기 Q_U를 pairing 연산하여 $c(= e(Y, Q_U))$를 생성하는 단계;

상기 t_u 의 해쉬함수 값 $h(=H_1(t_u))$을 생성하여, 상기 t_u와 g의 h 멱승(g^h)을 곱한 값(t_ug^h)을 생성하는 단계; 및

상기 $c = t_ug^h$를 만족하면 유효한 클라이언트로 인증하는 단계를 포함하여 이루어지며,

상기 (b)단계의 t_v는, 서버가 위수가 q인 곱셈그룹 Zq^*로부터 임의로 선택함에 의해 생성되며,

상기 (b)단계의 z 및 세션키(sk)의 생성은, 상기 클라이언트가 유효한 클라이언트로 인증되면, $z = H_2(t_u, t_v, X, Y, ID_u, ID_v)$, $sk = H_3(t_u, t_v, X, Y, ID_u, ID_v)$에 의해 이루어지며,

상기 (c)단계의 서버 인증은,

ID_U, ID_V, 클라이언트가 생성한 t_u, X, Y, 상기 서버로부터 받은 t_v를 해쉬함수한 값 $z'(= H_2(tu,tv,X,Y,ID_U, ID_V)$를 생성하는 단계; 및

상기 z'가 z와 일치(z'= z)하는지 체크하는 단계를 포함하여 이루어지고,

상기 (c)단계의 세션키(sk) 생성은, 상기 서버가 유효한 서버로 인증되면, $sk = H_3(t_u, t_v, X, Y, ID_u, ID_v)$에 의해 이루어지는, ID-기반의 키교환 방법.

청구항 2.

제1항에 있어서,

상기 위수가 소수 q인 gap DiffieHellman 그룹의 생성자 P는

위수가 q인 덧셈그룹(G1)의 생성자 임을 특징으로 하는 ID-기반의 키교환 방법.

청구항 3.

제1항에 있어서,

상기 (b)단계의 클라이언트 인증에서 $e(Y, Q_U))$이 $t_u g^h$와 동일한 값이 아니면 클라이언트 인증 실패 메시지를 클라이언트에게 전송함을 특징으로 하는 ID-기반의 키교환 방법.

청구항 4.

제1항에 있어서,

상기 (c)단계의 서버 인증에서 z 와 z' 이 동일한 값이 아니면 서버 인증 실패 메시지를 서버에게 전송함을 특징으로 하는 ID-기반의 키교환 방법.

청구항 5.

위수가 q인 덧셈그룹(G1)의 생성자를 P, 키생성센터(KGC)의 마스터 비밀키를 s, 상기 s와 P의 곱을 Ppub(= sP), 상기 P의 pairing 연산값을 g(=e(P,P)), 클라이언트 및 서버의 ID를 각각 ID_u 및 ID_v, 키생성센터로부터 받은 클라이언트 및 서버의 비밀키를 각각 S_u 및 S_v 라 할 때,

클라이언트에 설치되며, X 값 및 Y 값을 생성하는 X&Y 생성부;

클라이언트 측에 설치되며, 상기 ID_u, X 및 Y를 서버로 전송하는 클라이언트전송부;

클라이언트에 설치되며, 서버로부터 z, t_v를 받아 서버를 인증하는 클라이언트 인증부;

유효한 서버로 인증되면 상기 서버의 세션키와 동일한 세션키를 생성하는 클라이언트세션키 생성부;

서버에 설치되며, 상기 ID_u, X 및 Y를 받아 클라이언트를 인증하는 서버 인증부;

서버에 설치되며, 클라이언트가 유효한 크라이언트로 인증되면 z 및 t_v를 생성하는 z&t_v 생성부;

상기 z 및 t_v를 클라이언트로 전송하는 서버전송부; 및

서버에 설치되며, 클라이언트가 유효한 클라이언트로 인증되면 세션키(sk)를 생성하는 서버 세션키 생성부를 포함함을 특징으로 하고,

상기 X&Y의 생성부는

위수가 q인 곱셈그룹 Zq^*로부터 임의의 값 a를 선택하는 a 선택부;

상기 ID_V 의 해쉬함수 값 $q_v(= H(ID_V))$를 생성하고, 상기 P_{pub} 와 q_vP를 합하여 $Qv(= P_{pub} + q_vP)$를 생성하고, 상기 g의 a 멱승을 취하여 $t_u(= g^a)$를 생성하고, 상기 t_u의 해쉬함수 값 $h(=H_1(t_u))$를 생성하는 매개값생성부;

상기 a와 Q_v의 곱을 계산하여 $X(=aQ_V)$를 생성하는 X생성부; 및

상기 a와 h의 합에 S_U 를 곱하여 $Y(=(a+ h)S_U)$를 생성하는 Y생성부를 포함하여 이루어지며,

상기 클라이언트 인증부는,

ID_U, ID_V, 클라이언트가 생성한 t_u, X, Y, 상기 서버로부터 받은 t_v를 해쉬함수한 값 $z'(= H_2(t_u,t_v,X,Y,ID_U, ID_V)$를 생성하는 z'생성부 및

상기 z'가 상기 z와 일치(z'= z)하는지 체크하여 서버를 인증하는 z'인증부를 포함하며,

상기 클라이언트 세션키 생성부는

상기 서버가 유효한 서버로 인증되면, t_u, t_v, X, Y, ID_u, ID_v의 해쉬함수 값$(sk = H_3(t_u, t_v, X, Y, ID_u, ID_v))$을 생성하며,

서버 인증부는

상기 ID_U 의 해쉬함수 값 $q_u(= H(ID_U))$를 생성하고, 상기 Ppub 를 q_u와 P의 곱(q_uP)과 합하여 $Q_U(= Ppub + q_uP)$를 생성하고, 클라이언트로부터 받은 X와 S_V의 pairing 연산하여 $tu(= e(X, S_V))$를 생성하고, 클라이언트로부터 받은 Y와 상기 Q_U 를 pairing 연산하여 $c(= e(Y, Q_U))$를 생성하는 c생성부;

상기 t_u 의 해쉬함수 값 $h(=H_1(t_u))$을 생성하여, 상기 t_u와 g의 h 멱승(g^h)을 곱한 값(t_ug^h)을 생성하는 t_ug^h생성부; 및

상기 c와 상기 t_ug^h를 비교하여 그 값이 동일하면 유효한 클라이언트로 인증하는 c 인증부를 포함하여 이루어지며,

상기 z&t_v생성부는,

서버가 위수가 q인 곱셈그룹 Zq^*로부터 임의로 t_v를 선택하는 t_v선택부; 및

상기 클라이언트가 유효한 클라이언트로 인증되면, t_u, t_v, X, Y, ID_u, ID_v의 해쉬함수 값 $z(= H_2(t_u, t_v, X, Y, ID_u, ID_v))$를 생성하는 z생성부를 포함하고,

상기 서버세션키 생성부는

상기 클라이언트가 유효한 클라이언트로 인증되면, t_u, t_v, X, Y, ID_u, ID_v를 해쉬함수한 값 $sk(= H_3(t_u, t_v, X, Y, ID_u, ID_v))$를 생성하며, ID-기반의 키교환 장치.

청구항 6.

제5항에 있어서,

상기 위수가 소수 q인 gap DiffieHellman 그룹의 생성자 P는

위수가 q인 덧셈그룹(G1)의 생성자 임을 특징으로 하는 ID-기반의 키교환 장치.

청구항 7.

제5항에 있어서,

상기 클라이언트 인증부에서 $e(Y, Q_U))$이 $t_u g^h$와 동일한 값이 아니면 클라이언트 인증 실패 메시지를 클라이언트에게 전송함을 특징으로 하는 ID-기반의 키교환 장치.

청구항 8.

제5항에 있어서,

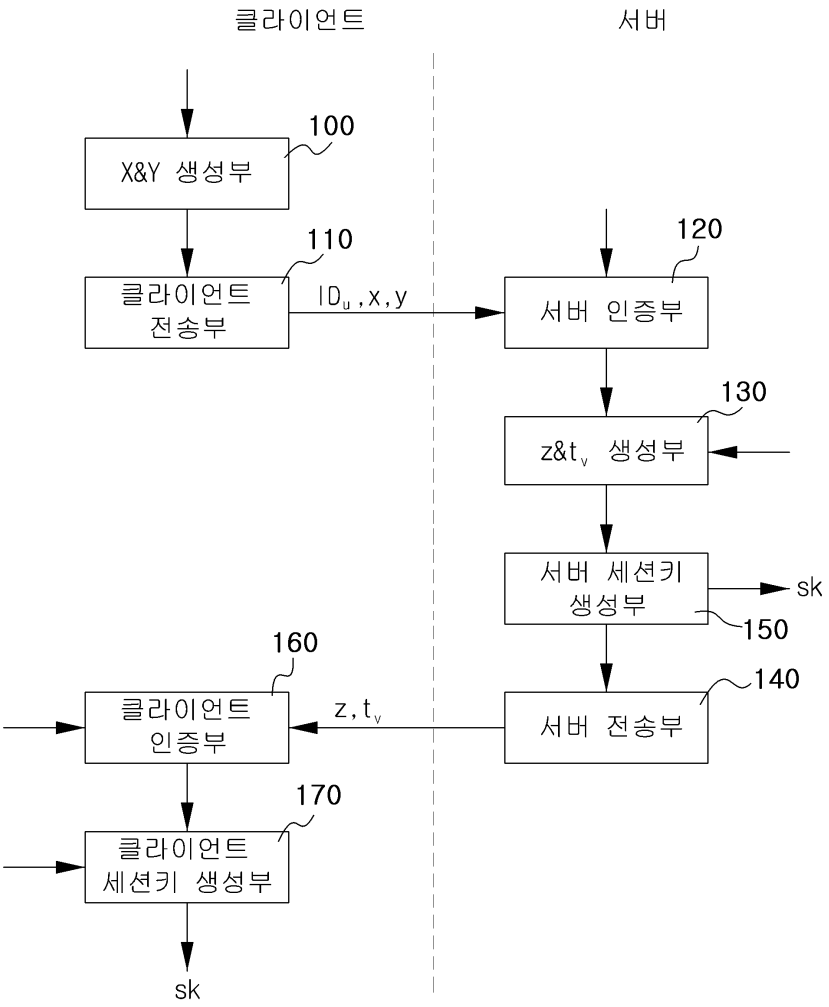
상기 서버 인증부에서 z 와 z' 이 동일한 값이 아니면 서버 인증 실패 메시지를 서버에게 전송함을 특징으로 하는 ID-기반의 키교환 장치.

청구항 9.

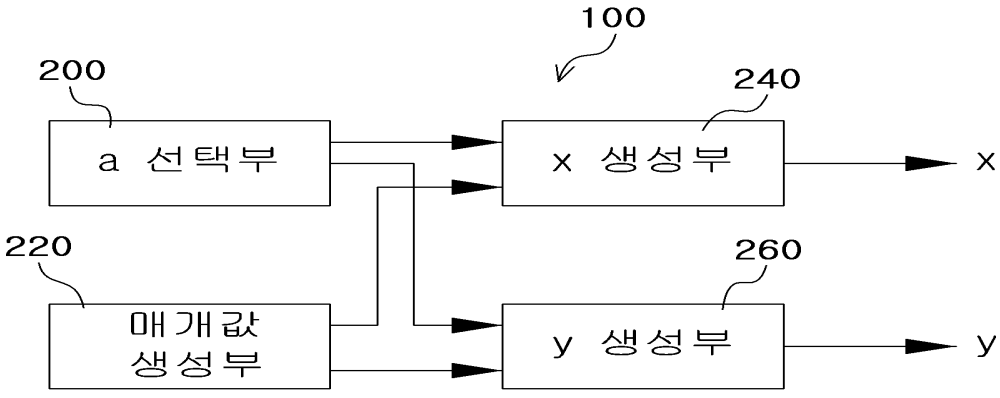
제1항에 기재된 발명을 컴퓨터에서 실행시키기 위한 프로그램을 기록한 컴퓨터로 읽을 수 있는 기록매체.

도면

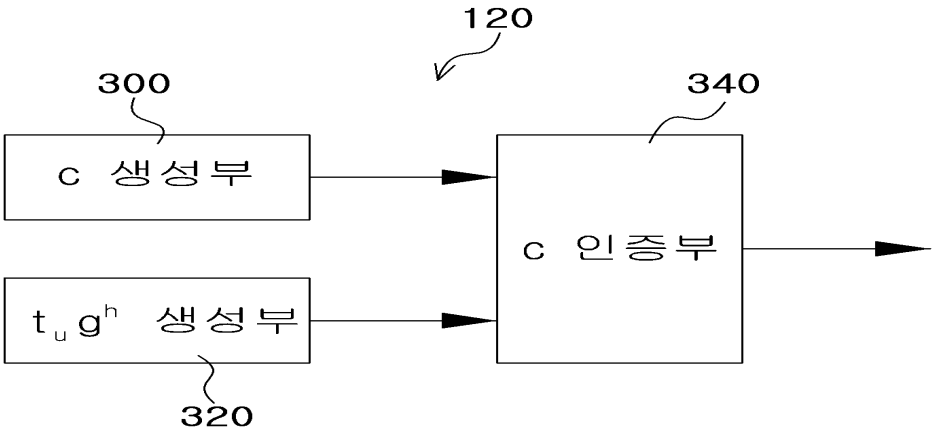
도면1


클라이언트
서버
100
X&Y 생성부
110
클라이언트 전송부
ID_u, x, y
120
서버 인증부
130
$z\&t_v$ 생성부
서버 세션키 생성부
sk
150
140
서버 전송부
z, t_v
160
클라이언트 인증부
170
클라이언트 세션키 생성부
sk

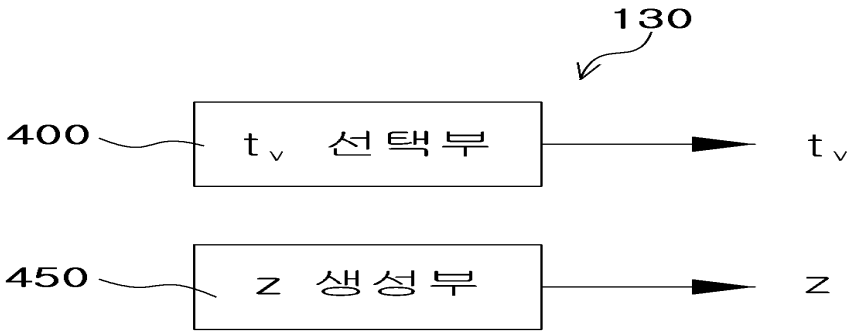
도면2


100
200
a 선택부
240
x 생성부
x
220
매개값 생성부
260
y 생성부
y

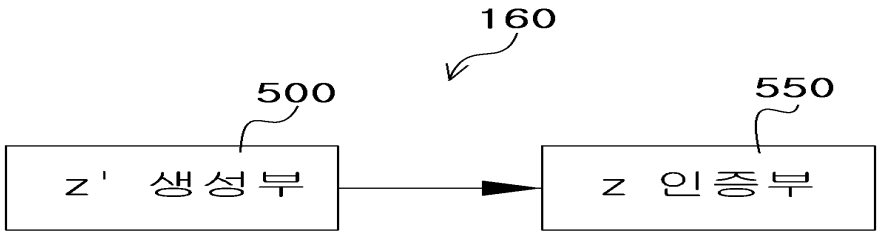
도면3


120
300
340
c 생성부
c 인증부
$t_u g^h$ 생성부
320

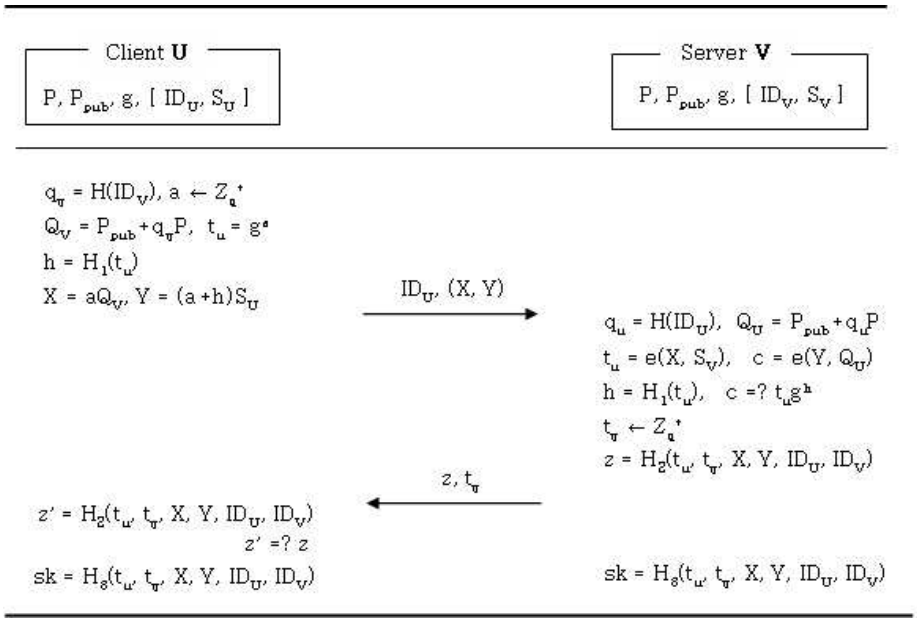
도면4


130
400
t_v 선택부
t_v
450
z 생성부
z

도면5


160
500
550
z' 생성부
z 인증부

도면6


Client U
P, P_{pub}, g, [ID_U, S_U]
Server V
P, P_{pub}, g, [ID_V, S_V]
$q_v = H(ID_V)$, $a \leftarrow Z_q^*$
$Q_V = P_{pub} + q_vP$, $t_u = g^a$
$h = H_1(t_u)$
$X = aQ_V$, $Y = (a+h)S_U$
ID_U, (X, Y)
$q_u = H(ID_U)$, $Q_U = P_{pub} + q_uP$
$t_u = e(X, S_V)$, $c = e(Y, Q_U)$
$h = H_1(t_u)$, $c =? t_ug^h$
$t_v \leftarrow Z_q^*$
$z = H_2(t_u, t_v, X, Y, ID_U, ID_V)$
z, t_v
$z' = H_2(t_u, t_v, X, Y, ID_U, ID_V)$
$z' =? z$
$sk = H_3(t_u, t_v, X, Y, ID_U, ID_V)$
$sk = H_3(t_u, t_v, X, Y, ID_U, ID_V)$