(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2009년08월26일
(11) 등록번호 10-0913783
(24) 등록일자 2009년08월18일

(51) Int. Cl.
H04L 9/32 (2006.01) *H04L 9/28* (2006.01)
(21) 출원번호 10-2007-0108156
(22) 출원일자 2007년10월26일
심사청구일자 2007년10월26일
(65) 공개번호 10-2009-0042412
(43) 공개일자 2009년04월30일
(56) 선행기술조사문헌
KR1020030037145 A
KR1020040009964 A
전체 청구항 수 : 총 14 항

(73) 특허권자
인하대학교 산학협력단
인천 남구 용현동 253 인하대학교
(72) 발명자
양대헌
서울 서초구 서초동 삼풍APT 3동 405호
(74) 대리인
이원희

심사관 : 성인구

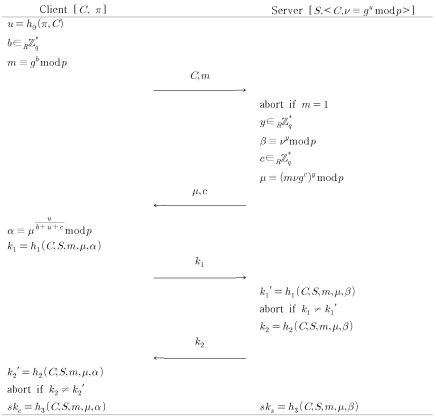
(54) 패스워드 기반 인증 및 키 교환 방법

(57) 요 약

본 발명은 인터넷 또는 이동통신 시스템과 같은 네트워크 시스템에서 최적의 보안 서비스 제공을 위해 패스워드를 사용한 인터넷 키 교환 서비스 기술 적용시 클라이언트와 서버 간에 보안 강도를 약화시키지 않으면서 기존의 AMP 보다 적은 양의 연산으로 빠르게 인증 및 키 교환 결과를 얻을 수 있으며, 전력 소비를 절감할 수 있는 패스워드 기반 인증 및 키 교환 방법을 제공하는데 그 목적이 있으며, 이를 위해 본 발명은,

네트워크를 통해 클라이언트와 서버 간에 패스워드 인증을 위한 키를 교환하는 방법에 있어서, 상기 클라이언트는 패스워드(π)와 관계없이 난수 b를 랜덤하게 생성하여 다수의 (b, $m = g^b \bmod p$)쌍을 미리 연산하여 저장하고, 상기 서버는 사용자 ID(C)와 관계없이 난수 c를 랜덤하게 생성하여 다수의 (c,$g^c \bmod p$)쌍을 미리 연산하여 저장하고, 또한 상기 서버는 사용자 ID(C)와 관계하여 난수 y를 랜덤하게 생성하여 다수의 (y, $\beta = v^y \bmod p$(β는 키 재료 정보))쌍을 미리 연산하여 저장하는 제1 단계; 상기 클라이언트가 사용자로부터 사용자 ID(C)와 패스워드(π)를 입력받으면, 상기 미리 연산된 다수의 (b, m) 중 어느 하나의 m을 선택하여 상기 서버로 사용자 ID(C)와 m을 전송하는 제2 단계; 상기 사용자 ID(C)와 m을 전송받은 서버는, 상기 미리 연산된 (y, $\beta = v^y \bmod p$) 중 어느 하나의 β와 상기 m 및 난수 c, y를 이용하여 세션 키 생성 계수 μ를 계산한 후, 상기 클라이언트로 μ와 c를 전송하는 제3 단계; 상기 μ와 c를 전송받은 클라이언트는, μ, c, b, 그리고 사용자 ID의 해시값 u를 이용하여 키 재료 정보 α를 추출하고, 상기 α를 이용하여 키 교환 요소 k_1을 연산한 후, 상기 서버로 전송하는 제4 단계; 상기 k_1을 전송받은 서버는, 키 교환 요소 k_2을 연산한 후, 그 결과 값이 상기 k_1과 같다면 상기 k_2를 상기 클라이언트로 전송하고 세션 키 sk_s를 생성하는 제5 단계; 상기 k_1을 k_2'으로 하여 상기 k_2와 비교한 후, 그 결과 값이 같다면 상기 클라이언트는 세션키 sk_c를 생성하는 제6 단계를 포함하여 구성된다.

대 표 도 - 도1



특허청구의 범위

청구항 1

네트워크를 통해 클라이언트와 서버 간에 패스워드 인증을 위한 키를 교환하는 방법에 있어서,

상기 클라이언트는 패스워드(π)와 관계없이 난수 b를 랜덤하게 생성하여 다수의 (b, m = g^b mod p)쌍을 미리 연산하여 저장하고, 상기 서버는 사용자 ID(C)와 관계없이 난수 c를 랜덤하게 생성하여 다수의 (c,g^c mod p)쌍을 미리 연산하여 저장하고, 또한 상기 서버는 사용자 ID(C)와 관계하여 난수 y를 랜덤하게 생성하여 다수의 (y, β= $υ^y$ mod p(β는 키 재료 정보))쌍을 미리 연산하여 저장하는 제1 단계;

상기 클라이언트가 사용자로부터 사용자 ID(C)와 패스워드(π)를 입력받으면, 상기 미리 연산된 다수의 (b, m) 중 어느 하나의 m을 선택하여 상기 서버로 사용자 ID(C)와 m을 전송하는 제2 단계;

상기 사용자 ID(C)와 m을 전송받은 서버는, 상기 미리 연산된 (y, β=$υ^y$ mod p) 중 어느 하나의 β와 상기 m 및 난수 c, y를 이용하여 세션 키 생성 계수 μ를 계산한 후, 상기 클라이언트로 μ와 c를 전송하는 제3 단계;

상기 μ와 c를 전송받은 클라이언트는, μ, c, b, 그리고 사용자 ID의 해시 값 u를 이용하여 키 재료 정보 α를 추출하고, 상기 α를 이용하여 키 교환 요소 k_1을 연산한 후, 상기 서버로 전송하는 제4 단계;

상기 k_1을 전송받은 서버는, 키 교환 요소 k_2을 연산한 후, 그 결과 값이 상기 k_1과 같다면 상기 k_2를 상기 클라이언트로 전송하고 세션 키 sk_s를 생성하는 제5 단계;

상기 k_1을 k_2'으로 하여 상기 k_2와 비교한 후, 그 결과 값이 같다면 상기 클라이언트는 세션키 sk_c를 생성하는 제6 단계

를 포함하는 패스워드 기반 인증 및 키 교환 방법.

청구항 2

청구항 1에 있어서,

상기 세션 키 생성 계수 μ는 하기의 식(1)에 의해 서버에서 생성되는 것을 특징으로 하는 패스워드 기반 인증 및 키 교환 방법.

식(1) : $μ = (m υ g^c)^y$ mod p

(여기서, c, y는 난수, p는 p와 서로소인 임의의 r에 대해 p=rq+1인 소수, υ는 서버에 저장된 사용자의 프로필)

청구항 3

청구항 1에 있어서,

상기 키 재료 정보 α는 하기의 식(2)에 의해 클라이언트에서 생성되는 것을 특징으로 하는 패스워드 기반 인증 및 키 교환 방법.

식(2) : $α = μ^{u/(b+u+c)}$ mod p

(여기서, b, c는 난수, u는 사용자 ID의 해시 값, p는 p와 서로소인 임의의 r에 대해 p=rq+1인 소수)

청구항 4

청구항 1에 있어서,

상기 키 교환 요소 k_1은 상기 키 생성 계수 μ, 상기 키 재료 정보 α, 상기 m 및, 사용자 ID(C)를 포함하는 해시 값인 것을 특징으로 하는 패스워드 기반 인증 및 키 교환 방법.

청구항 5

청구항 1에 있어서,

상기 키 교환 요소 k_2는 상기 키 생성 계수 μ, 상기 키 재료 정보 β, 상기 m 및, 사용자 ID(C)를 포함하는 해시 값인 것을 특징으로 하는 패스워드 기반 인증 및 키 교환 방법.

청구항 6

청구항 1에 있어서,

상기 제5 단계에서 상기 키 교환 요소 k_1과 k_2의 결과 값이 같지 않다면 프로토콜을 중지하는 것을 특징으로 하는 패스워드 기반 인증 및 키 교환 방법.

청구항 7

청구항 1에 있어서,

상기 제6 단계에서 상기 키 교환 요소 k_2과 k_2'의 결과 값이 같지 않다면 프로토콜을 중지하는 것을 특징으로 하는 패스워드 기반 인증 및 키 교환 방법.

청구항 8

네트워크를 통해 클라이언트와 서버 간에 패스워드 인증을 위한 키를 교환하는 방법에 있어서,

상기 클라이언트는 패스워드(π)와 관계없이 난수 b를 랜덤하게 생성한 후, 상기 b를 이용하여 키 재료 정보 m($m = g^b \bmod p$)을 추출한 다음, 사용자 ID(C)와 상기 m을 서버로 전송하는 제1 단계;

상기 사용자 ID(C)와 m을 전송받은 서버는 난수 y를 랜덤하게 생성한 후, 상기 y를 이용하여 키 재료 정보 β($\beta = \upsilon^y \bmod p$)를 추출하고, 상기 m의 해시 값을 난수 c로 추출한 후, 상기 m 및 난수 c, y를 이용하여 세션 키 생성 계수 μ를 계산한 다음, 상기 클라이언트로 μ를 전송하는 제2 단계;

상기 μ를 전송받은 클라이언트는 상기 m의 해시 값을 난수 c로 추출하고, 상기 μ, c, b, 그리고 사용자 ID의 해시 값 u를 이용하여 키 재료 정보 α를 추출하고, 상기 α를 이용하여 키 교환 요소 k_1을 연산한 후, 상기 서버로 전송하는 제3 단계;

상기 k_1을 전송받은 서버는, 키 교환 요소 k_2을 연산한 후, 그 결과 값이 상기 k_1과 같다면 상기 k_2를 상기 클라이언트로 전송하고 세션 키 sk_s를 생성하는 제4 단계;

상기 k_1을 k_2'으로 하여 상기 k_2와 비교한 후, 그 결과 값이 같다면 상기 클라이언트는 세션키 sk_c를 생성하는 제5 단계

를 포함하는 패스워드 기반 인증 및 키 교환 방법.

청구항 9

청구항 8에 있어서,

상기 세션 키 생성 계수 μ는 하기의 식(3)에 의해 서버에서 생성되는 것을 특징으로 하는 패스워드 기반 인증 및 키 교환 방법.

식(3) : $\mu = (m \upsilon g^c)^y \bmod p$

(여기서, c는 m의 해시 값, y는 난수, p는 p와 서로소인 임의의 r에 대해 p=rq+1인 소수, υ는 서버에 저장된 사용자의 프로필)

청구항 10

청구항 8에 있어서,

상기 키 재료 정보 α는 하기의 식(4)에 의해 클라이언트에서 생성되는 것을 특징으로 하는 패스워드 기반 인증 및 키 교환 방법.

식(4) : $\alpha = \mu^{u/(b+u+c)} \bmod p$

(여기서, b는 난수, c는 m의 해시 값, u는 사용자 ID의 해시 값, p는 p와 서로소인 임의의 r에 대해 p=rq+1인 소수)

청구항 11

청구항 8에 있어서,

상기 키 교환 요소 k_1은 상기 키 생성 계수 μ, 상기 키 재료 정보 α, 상기 m 및, 사용자 ID(C)를 포함하는 해시 값인 것을 특징으로 하는 패스워드 기반 인증 및 키 교환 방법.

청구항 12

청구항 8에 있어서,

상기 키 교환 요소 k_2는 상기 키 생성 계수 μ, 상기 키 재료 정보 β, 상기 m 및, 사용자 ID(C)를 포함하는 해시 값인 것을 특징으로 하는 패스워드 기반 인증 및 키 교환 방법.

청구항 13

청구항 8에 있어서,

상기 제4 단계에서 상기 키 교환 요소 k_1과 k_2의 결과 값이 같지 않다면 프로토콜을 중지하는 것을 특징으로 하는 패스워드 기반 인증 및 키 교환 방법.

청구항 14

청구항 8에 있어서,

상기 제5 단계에서 상기 키 교환 요소 k_2과 k_2'의 결과 값이 같지 않다면 프로토콜을 중지하는 것을 특징으로 하는 패스워드 기반 인증 및 키 교환 방법.

명 세 서

발명의 상세한 설명

기 술 분 야

<1> 본 발명은 패스워드 기반 인증 및 키 교환 방법에 관한 것으로, 보다 상세하게는 인터넷 또는 이동통신 시스템과 같은 네트워크 시스템에서 최적의 보안 서비스 제공을 위해 패스워드를 사용한 인터넷 키 교환 서비스 기술 적용시 클라이언트와 서버 간에 보안 강도를 약화시키지 않으면서 기존의 AMP 보다 적은 양의 연산으로 빠르게 인증 및 키 교환 결과를 얻을 수 있으며, 전력 소비를 절감할 수 있는 패스워드 기반 인증 및 키 교환 방법에 관한 것이다.

배 경 기 술

<2> 패스워드는 사람이 외울 수 있는 인증수단으로 많이 사용되고 있으나 근본적으로 사람이 외워야 하는 한계성 때문에 사람이 쉽게 외울 수 있는 단어들을 활용한 사전 공격(Dictionary Attack)에 약한 것으로 알려져 왔다. 패스워드를 기반으로 하는 인증 및 키 교환 프로토콜은 이러한 패스워드의 낮은 엔트로피 때문에, 안전한 프로토콜을 설계하기 위해서는 서버와 클라이언트 측에 많은 연산이 필요로 하는 방법들이 제안되어 왔다.

<3> 패스워드의 편의성을 유지하면서 이러한 단점을 극복하기 위해 낮은 엔트로피의 패스워드를 이용하여 안전한 인증 및 키 교환을 수행하는 AMP(Authentication and key agreement via Memorable Password) 프로토콜이 제안되었다. AMP 프로토콜은 이산대수문제(Discrete Logarithm Problem)에 기반한 디피-헬먼법(Diffie-Hellman)을 이

용하여 프로토콜을 완성하였다.

<4> 상기 AMP 프로토콜은 비교적 적은 연산을 수행하는 패스워드 기반의 인증 및 키 교환 프로토콜이지만, 예를 들면, 이동통신 시스템에서의 이동 단말기(클라이언트)의 계산 파워가 작으므로 이동단말기의 계산량을 더욱 더 작게 줄이므로써 소비 전력량을 줄이기 위한 기술적인 방법이 필요하다.

발명의 내용

해결 하고자하는 과제

<5> 본 발명은 전술한 기술적 필요성에 의해 고안된 것으로, 인터넷 또는 이동통신 시스템과 같은 네트워크 시스템에서 최적의 보안 서비스 제공을 위해 패스워드를 사용한 인터넷 키 교환 서비스 기술 적용시 클라이언트와 서버 간에 보안 강도를 약화시키지 않으면서 기존의 AMP 보다 적은 양의 연산으로 빠르게 인증 및 키 교환 결과를 얻을 수 있으며, 전력 소비를 절감할 수 있는 패스워드 기반 인증 및 키 교환 방법을 제공하는데 그 목적이 있다.

과제 해결수단

<6> 상기와 같은 본 발명의 목적은,

<7> 네트워크를 통해 클라이언트와 서버 간에 패스워드 인증을 위한 키를 교환하는 방법에 있어서, 상기 클라이언트는 패스워드(π)와 관계없이 난수 b를 랜덤하게 생성하여 다수의 (b, $m = g^{b} \bmod p$)쌍을 미리 연산하여 저장하고, 상기 서버는 사용자 ID(C)와 관계없이 난수 c를 랜덤하게 생성하여 다수의 (c,$g^{c} \bmod p$)쌍을 미리 연산하여 저장하고, 또한 상기 서버는 사용자 ID(C)와 관계하여 난수 y를 랜덤하게 생성하여 다수의 (y, $\beta = v^{y} \bmod p$ (β는 키 재료 정보))쌍을 미리 연산하여 저장하는 제1 단계; 상기 클라이언트가 사용자로부터 사용자 ID(C)와 패스워드(π)를 입력받으면, 상기 미리 연산된 다수의 (b, m) 중 어느 하나의 m을 선택하여 상기 서버로 사용자 ID(C)와 m을 전송하는 제2 단계; 상기 사용자 ID(C)와 m을 전송받은 서버는, 상기 미리 연산된 (y, $\beta = v^{y} \bmod p$) 중 어느 하나의 β와 상기 m 및 난수 c, y를 이용하여 세션 키 생성 계수 μ를 계산한 후, 상기 클라이언트로 μ와 c를 전송하는 제3 단계; 상기 μ와 c를 전송받은 클라이언트는, μ, c, b, 그리고 사용자 ID의 해시 값 u를 이용하여 키 재료 정보 α를 추출하고, 상기 α를 이용하여 키 교환 요소 k_1을 연산한 후, 상기 서버로 전송하는 제4 단계; 상기 k_1을 전송받은 서버는, 키 교환 요소 k_2을 연산한 후, 그 결과 값이 상기 k_1과 같다면 상기 k_2를 상기 클라이언트로 전송하고 세션 키 sk_s를 생성하는 제5 단계; 상기 k_1을 k_2'으로 하여 상기 k_2와 비교한 후, 그 결과 값이 같다면 상기 클라이언트는 세션키 sk_c를 생성하는 제6 단계를 포함한다.

<8> 상기 세션 키 생성 계수 μ는 하기의 식(1)에 의해 서버에서 생성되는 것을 특징으로 한다.

<9> 식(1) : $\mu = (m v g^{c})^{y} \bmod p$

<10> 여기서, c, y는 난수, p는 p와 서로소인 임의의 r에 대해 p=rq+1인 소수, υ는 서버에 저장된 사용자의 프로필이다.

<11> 상기 키 재료 정보 α는 하기의 식(2)에 의해 클라이언트에서 생성되는 것을 특징으로 한다.

<12> 식(2) : $\alpha = \mu^{u/(b+u+c)} \bmod p$

<13> 여기서, b, c는 난수, u는 사용자 ID의 해시 값, p는 p와 서로소인 임의의 r에 대해 p=rq+1인 소수이다.

<14> 또한, 상기 키 교환 요소 k_1은 상기 키 생성 계수 μ, 상기 키 재료 정보 α, 상기 m 및, 사용자 ID(C)를 포함하는 해시 값인 것을 특징으로 한다.

<15> 또한, 상기 키 교환 요소 k_2는 상기 키 생성 계수 μ, 상기 키 재료 정보 β, 상기 m 및, 사용자 ID(C)를 포함하는 해시 값인 것을 특징으로 한다.

<16> 또한, 상기 제5 단계에서 상기 키 교환 요소 k_1과 k_2의 결과 값이 같지 않다면 프로토콜을 중지하는 것을 특징으로 한다.

<17> 또한, 상기 제6 단계에서 상기 키 교환 요소 k_2과 k_2'의 결과 값이 같지 않다면 프로토콜을 중지하는 것을 특징으로 한다.

<18> 또한, 본 발명의 또 다른 실시예에 따른 패스워드 기반 인증 및 키 교환 방법은, 네트워크를 통해 클라이언트와 서버 간에 패스워드 인증을 위한 키를 교환하는 방법에 있어서, 상기 클라이언트는 패스워드(π)와 관계없이 난수 b를 랜덤하게 생성한 후, 상기 b를 이용하여 키 재료 정보 $m(m = g^b \bmod p)$을 추출한 다음, 사용자 ID(C)와 상기 m을 서버로 전송하는 제1 단계; 상기 사용자 ID(C)와 m을 전송받은 서버는 난수 y를 랜덤하게 생성한 후, 상기 y를 이용하여 키 재료 정보 $\beta(\beta = \upsilon^y \bmod p)$를 추출하고, 상기 m의 해시 값을 난수 c로 추출한 후, 상기 m 및 난수 c, y를 이용하여 세션 키 생성 계수 μ를 계산한 다음, 상기 클라이언트로 μ를 전송하는 제2 단계; 상기 μ를 전송받은 클라이언트는 상기 m의 해시 값을 난수 c로 추출하고, 상기 μ, c, b, 그리고 사용자 ID의 해시 값 u를 이용하여 키 재료 정보 α를 추출하고, 상기 α를 이용하여 키 교환 요소 k_1을 연산한 후, 상기 서버로 전송하는 제3 단계; 상기 k_1을 전송받은 서버는, 키 교환 요소 k_2을 연산한 후, 그 결과 값이 상기 k_1과 같다면 상기 k_2를 상기 클라이언트로 전송하고 세션 키 sk_s를 생성하는 제4 단계; 상기 k_1을 k_2'으로 하여 상기 k_2와 비교한 후, 그 결과 값이 같다면 상기 클라이언트는 세션키 sk_c를 생성하는 제5 단계를 포함한다.

<19> 상기 세션 키 생성 계수 μ는 하기의 식(3)에 의해 서버에서 생성되는 것을 특징으로 한다.

<20> 식(1) : $\mu = (m \upsilon g^c)^y \bmod p$

<21> 여기서, c는 m의 해시 값, y는 난수, p는 p와 서로소인 임의의 r에 대해 p=rq+1인 소수, υ는 서버에 저장된 사용자의 프로필이다.

<22> 상기 키 재료 정보 α는 하기의 식(4)에 의해 클라이언트에서 생성되는 것을 특징으로 한다.

<23> 식(2) : $\alpha = \mu^{u/(b+u+c)} \bmod p$

<24> 여기서, b는 난수, c는 m의 해시 값, u는 사용자 ID(C)의 해시 값, p는 p와 서로소인 임의의 r에 대해 p=rq+1인 소수이다.

<25> 또한, 상기 키 교환 요소 k_1은 상기 키 생성 계수 μ, 상기 키 재료 정보 α, 상기 m 및, 사용자 ID(C)를 포함하는 해시 값인 것을 특징으로 한다.

<26> 또한, 상기 키 교환 요소 k_2는 상기 키 생성 계수 μ, 상기 키 재료 정보 β, 상기 m 및, 사용자 ID(C)를 포함하는 해시 값인 것을 특징으로 한다.

<27> 또한, 상기 제4 단계에서 상기 키 교환 요소 k_1과 k_2의 결과 값이 같지 않다면 프로토콜을 중지하는 것을 특징으로 한다.

<28> 또한, 상기 제5 단계에서 상기 키 교환 요소 k_2과 k_2'의 결과 값이 같지 않다면 프로토콜을 중지하는 것을 특징으로 한다.

효 과

<29> 상기와 같은 본 발명에 따른 패스워드 기반 인증 및 키 교환 방법에 의하면, 인터넷 또는 이동통신 시스템과 같은 네트워크 시스템에서 최적의 보안 서비스 제공을 위해 패스워드를 사용한 인터넷 키 교환 서비스 기술 적용 시 클라이언트와 서버 간에 보안 강도를 약화시키지 않으면서 기존의 AMP 보다 적은 양의 연산으로 빠르게 인증 및 키 교환 결과를 얻을 수 있으며, 전력 소비를 절감할 수 있는 효과가 있다.

발명의 실시를 위한 구체적인 내용

<30> 이하, 첨부된 도면을 참조하여 본 발명의 실시예를 상세히 설명한다. 우선, 도면들 중 동일한 구성요소 또는 부품들은 가능한 한 동일한 참조부호를 나타내고 있음에 유의해야 한다. 본 발명을 설명함에 있어서 관련된 공지 기능 혹은 구성에 대한 구체적인 설명은 본 발명의 요지를 모호하게 하지 않기 위해 생략한다.

<31> 도 1은 본 발명에 따른 패스워드 기반 인증 및 키 교환 방법을 개념적으로 도시한 도, 도 2는 본 발명의 또 다

른 실시예에 따른 패스워드 기반 인증 및 키 교환 방법을 개념적으로 도시한 도, 도 3은 본 발명에 따른 패스워드 기반 인증 및 키 교환 방법에 있어서 프로토콜의 필요한 사전 및 실시간 연산량을 나타낸 표이다.

<32> 먼저, 아래 [표 1]에 자주 사용되는 기호에 대한 설명을 개시한다.

<33> [표 1]

<34>

C	사용자 ID
S	서버
π	패스워드
k	비밀키 암호화를 위한 보안 파라미터 (128, 160 비트 등)
l	공개키 암호화를 위한 보안 파라미터 (3072, 3840 비트 등)
h_i	랜덤 오라클
q	사이즈가 k인 소수(prime)
p	p와 서로소인 임의의r 에 대해서 p=rq+1 이며 사이즈가 l인 소수
Z_x^*	x의 곱셈 그룹(multiplicative group)
g	그룹 Z_p^*의 q-order 하위 그룹 G_q의 생성자(generator)
ε	타원 곡선 기반 암호화를 위한 보안 파라미터 (256, 512 비트 등)
E	타원 곡선(Elliptic Curve)
P	크기가 ε인 E위에서의 한 점

<35> $Z_p = \{0,1,2,...,p-1\}$로 정의하고 이를 법에 관한 완전 잉여계라고 한다. 법에 관한 완전 잉여계의 원소 $a \in Z_p$에 대해서 gcd(a,p)=1 인 원소들만 모아 놓은 집합을 Z_p^*라 한다. 이러한 성질에 의하여 만약 p가 소수일 때, $Z_p^* = \{0,1,2,...,p-1\}$로 정의된다.

<36> h_i는 h_i: $\{0,1\}^*$ -> $\{0,1\}^k$과 같은 성질을 만족한다. 즉, 어떠한 길이의 비트열(βit string)도 k크기의 비트열로 만든다. (참고로 $\{0,1\}^n$는 길이가 n인 비트열을 의미함)

<37> $h_i(M)$는 $h_i(i,M,i)$로 정의하며, h()는 MD5나 SHA-1과 같은 안전한 일 방향(one-way) 함수를 의미한다. 여기서 ‘,’ 는 비트열 간의 연결(concatenation)을 의미한다.

<38> 디피-헐먼법(DH: Diffie-Hellman), DSA(Digital Signature Algorithm)은 모두 Number field sieve와 같은 sub-exponential 알고리즘에 의해서 약하다. 그러나 타원 곡선 암호화(Elliptic Curve Cryptography)의 경우, 수학적 문제를 풀기 위해 가장 잘 알려진 알고리즘은 Pollard-rho 알고리즘으로 지수승의 시간(exponential time)이 필요하다. 그러므로 DH, DSA와 다르게 타원 곡선 암호화는 적은 사이즈의 키로써도 동등한 수준의 보안을 제공할 수 있다.

<39> 다음에 기술하는 프로토콜은 타원 곡선 암호화로 다시 기술될 수 있으며, 이 경우 더 적은 수준의 연산으로 동등한 수준의 보안을 제공할 수 있다. 타원 곡선 암호화로 다시 기술하기 위하여, Z_p^*위에서의 임의의 수 $g^x \bmod p$ 는, 타원 곡선 위의 임의의 점 xP로 다시 쓸 수 있다. 이때 g^x의 크기는 l이며 xP의 크기는 ε이다.

<40> 기존의 AMP 프로토콜 구성에 대해 설명한다.

<41> 사용자는 자신의 ID(C)와 패스워드(π)를 정한다. 서버는 사용자의 프로필(계정) <C,v>를 만든다. $v = g^u \bmod p$ 이며 $u = h_0(\pi, C)$이다. 서버는 DLP(Discrete Logarithm Problem)의 어려움에 따라 υ를 안다고 해도 u를 알기 어려우므로, 사용자의 패스워드를 알아내기 어렵다.

<42> 프로토콜 동작에 대해 설명하면, 먼저, 클라이언트는 사용자 ID(C)로부터 패스워드(π)를 입력받은 후 난수 $b \in_R Z_p^*$를 선택하고, $m = g^b \bmod p$과 $u = h_1(\pi, C)$를 계산한 후 C, m을 서버에게 전송한다.

<43> [1] 클라이언트 → 서버 : C, m

<44> 서버는 m이 1인지 아닌지를 확인한 후, 만약 1이라면 프로토콜을 중지한다. 만약 1이 아니라면 서버는 난수 $y \in_R Z_p^*$와 $c \in_R Z_p^*$를 선택하고, $\beta = \upsilon^y \bmod p$ ($\beta = y\upsilon$)와 $\mu = (m\upsilon g^c)^y \bmod p$ 를 계산한 후, μ와 c를 클라이언트에게 전송한다.

<45> [2] 서버 → 클라이언트 : μ, c

<46> μ와 c가 도착하면 클라이언트는 $\alpha = \mu^{u/(b+u+c)} \bmod p$를 계산하고 $k_1 = h_1(C,S,m,\mu,\alpha)$을 계산하여 k_1을 서버에게 전송한다.

<47> [3] 클라이언트 → 서버 : k_1

<48> 서버는 $k_1' = h_1(C,S,m,\mu,\beta)$를 계산하고, 만약 k_1'이 k_1과 같지 않다면 프로토콜을 중지한다. 만약 비교 결과가 같다면 $k_2 = h_2(C,S,m,\mu,\beta)$를 계산하여 k_2를 클라이언트에게 전송하고 세션 키 $sk_s = h_3(C,S,m,\mu,\beta)$를 계산한다.

<49> [4] 서버 → 클라이언트 : k_2

<50> 클라이언트는 $k_2' = h_2(C,S,m,\mu,\alpha)$를 계산하고, 만약 k_2'이 k_2와 같지 않다면 프로토콜을 중지한다. 만약 비교 결과가 같다면 클라이언트는 세션 키 $sk_c = h_3(C,S,m,\mu,\alpha)$를 계산한다. 서버와 클라이언트는 이후 $sk_s = sk_c$를 세션 키로 비밀 통신을 수행한다.

<51> 세션 키의 동등성에 대하여 아래 수식을 참조한다.

$$
\begin{aligned}
\alpha &\equiv \mu^{\frac{u}{b+u+c}} && \equiv (m\nu g^c)^{y\frac{u}{b+u+c}} \\
&\equiv (g^b g^u g^c)^{y\frac{u}{b+u+c}} && \equiv g^{(b+u+c)y\frac{u}{b+u+c}} \\
&\equiv g^{yu} && \equiv (g^u)^y \\
&\equiv \nu^y \equiv \beta \bmod p
\end{aligned}
$$

<52>

<53> 위의 수식에서 알 수 있듯이 만약 서버가 올바른 클라이언트의 프로파일 < C, $\upsilon = g^u \bmod p$>을 가지고 있다면 α와 β가 동일하므로 $sk_c = h_3(C,S,m,\mu,\alpha) = h_3(C,S,m,\mu,\beta) = sk_s$으로 동일한 세션 키를 생성할 수 있게 된다.

<54> 도 1을 참조하면, 본 발명에 따른 패스워드 기반 인증 및 키 교환 방법은 기존의 AMP 프로토콜과는 달리, 먼저 클라이언트는 패스워드(π)와 관계없이 난수 b를 랜덤하게 생성하여 다수의 (b, $m=g^b \bmod p$)쌍을 미리 연산하여 저장하고, 서버는 사용자 ID(C)와 관계없이 난수 c를 랜덤하게 생성하여 다수의 (c,$g^c \bmod p$)쌍을 미리 연산하여 저장하고, 또한 상기 서버는 사용자 ID(C)와 관계하여 난수 y를 랜덤하게 생성하여 다수의 (y, $\beta = \upsilon^y \bmod p$)쌍을 미리 연산하여 저장한다.

<55> 그 다음, 상기 클라이언트가 사용자로부터 사용자 ID(C)와 패스워드(π)를 입력받으면, 상기 미리 연산된 다수의 (b, m) 중 어느 하나의 m을 선택하여 상기 서버로 사용자 ID(C)와 m을 전송한 후,

<56> 상기 사용자 ID(C)와 m을 전송 받은 서버는, 상기 미리 연산된 (y, $\beta = \upsilon^y \bmod p$) 중 어느 하나의 β와 상기 m 및 난수 c, y를 이용하여 μ를 계산한 후, 상기 클라이언트로 μ와 c를 전송한다.

<57> 서버가 c를 랜덤하게 생성하여 사용하는 경우, 서버는 사용자와 독립적으로 수많은 (c,$g^c \bmod p$) 쌍을 미리 계산해놓을 수 있다. 실제 동작에서 이 수많은 쌍 중에서 하나의 (c,$g^c \bmod p$)을 가져와 사용할 수 있으므로, 서버는 μ를 생성하기 위하여 m, υ, g^c의 곱에 대해서 y의 지수승(exponentiation) 만을 수행하면 된다.

<58> 또한, 서버는 각각의 사용자에 대해서 필요한(예상되는) 수만큼의 (y, $\beta = \upsilon^{y} \bmod p$) 쌍을 클라이언트에서 실시간으로 생성한 m과 무관하게 미리 계산해놓을 수 있다.

<59> 따라서, 서버는 인증과 키 교환을 위하여 실시간으로는 한 번의 지수승(exponentiation)과 세 번의 해시 연산이 필요하다. 해시 연산은 지수승에 비하면 매우 적은 연산으로 결과를 구할 수 있다.

<60> 또한, 클라이언트는 패스워드와 무관하게 필요한(예상되는) 수만큼의 (b, $m = g^{b} \bmod p$) 쌍을 미리 계산해 놓을 수 있으며 인증 과정이 시작되면 이 중에 하나를 가져와 서버에 전송할 수 있다. 따라서 α를 계산하기 위하여 클라이언트는 각각 한 번의 역원(inverse) 연산과 한 번의 지수승(exponentiation) 연산을 필요로 한다.

<61> 이러한 사전 계산은 서버와 클라이언트가 바쁘지 않은 시간을 이용하여 계산할 수 있으며, 해당 장치의 성능을 낮추어도, 예를 들면, CPU의 클럭을 낮춘다든지 하는 식으로 성능을 낮추어도 계산할 수 있다. 이는 전력 소모에 있어서도 이득을 가져온다.

<62> 본 발명에 따른 패스워드 기반 인증 및 키 교환 방법의 다른 실시예에 따르면, 해시된 c의 사용하여 패스워드 기반 인증 및 키 교환을 수행할 수 있다.(도 2참조)

<63> 전술한 프로토콜 중에 c는 난수 형태가 아니라 c의 해시 형태로 구현될 수 있다. 즉, $c = h_4(m)$ 형태로 구하면 [2]에서 c를 함께 보낼 필요가 없으며, 클라이언트 측에서도 $c = h_4(m)$으로 바로 구할 수 있다. 그러나 만약, 이러한 방식으로 c를 구하게 되면 g^{c}를 구하기 위해서는 m을 반드시 알아야 하므로 (c, $g^{c} \bmod p$)쌍을 미리 계산해놓을 수 없게 된다.

<64> 만약, 서버에 미리 계산해놓은 (c, $g^{c} \bmod p$)쌍이 없으면 서버는 m과 관련한 c를 생성함으로써 메시지의 크기를 줄일 수 있다. 이 경우, 클라이언트는 만약 c가 따로 서버로부터 도착하지 않으면 $c = h_4(m)$를 계산하여 사용한다. 이렇게 하면 (c, $g^{c} \bmod p$)쌍을 미리 계산할 수 없다는 단점은 있지만, 클라이언트와 서버 간에 전송되는 메시지의 크기는 줄일 수 있다는 장점이 있다.

<65> 이상과 같이 본 발명에 따른 패스워드 기반 인증 및 키 교환 방법을 예시한 도면을 참조로 하여 설명하였으나, 본 명세서에 개시된 실시예와 도면에 의해 본 발명이 한정되는 것은 아니며, 본 발명의 기술사상 범위내에서 당업자에 의해 다양한 변형이 이루어질 수 있음은 물론이다.

도면의 간단한 설명

<66> 도 1은 본 발명에 따른 패스워드 기반 인증 및 키 교환 방법을 개념적으로 도시한 도,

<67> 도 2는 본 발명의 또 다른 실시예에 따른 패스워드 기반 인증 및 키 교환 방법을 개념적으로 도시한 도,

<68> 도 3은 본 발명에 따른 패스워드 기반 인증 및 키 교환 방법에 있어서 프로토콜의 필요한 사전 및 실시간 연산량을 나타낸 표이다.

도면

도면1

Client [C, π]		Server [S,< C,$\nu \equiv g^u \bmod p$>]
$u = h_0(\pi, C)$		
$b \in_R \mathbb{Z}_q^*$		
$m \equiv g^b \bmod p$		
	C, m ⟶	
		abort if $m = 1$
		$y \in_R \mathbb{Z}_q^*$
		$\beta \equiv \nu^y \bmod p$
		$c \in_R \mathbb{Z}_q^*$
		$\mu \equiv (m\nu g^c)^y \bmod p$
	⟵ μ, c	
$\alpha \equiv \mu^{\frac{u}{b+u+c}} \bmod p$		
$k_1 = h_1(C, S, m, \mu, \alpha)$		
	k_1 ⟶	
		$k_1' = h_1(C, S, m, \mu, \beta)$
		abort if $k_1 \neq k_1'$
		$k_2 = h_2(C, S, m, \mu, \beta)$
	⟵ k_2	
$k_2' = h_2(C, S, m, \mu, \alpha)$		
abort if $k_2 \neq k_2'$		
$sk_c = h_3(C, S, m, \mu, \alpha)$		$sk_s = h_3(C, S, m, \mu, \beta)$

도면2

Client [C, π]		Server [S,< C,$\nu \equiv g^u \bmod p$>]
$u = h_0(\pi, C)$		
$b \in_R \mathbb{Z}_q^*$		
$m \equiv g^b \bmod p$		
	C, m ⟶	
		abort if $m = 1$
		$y \in_R \mathbb{Z}_q^*$
		$\beta \equiv \nu^y \bmod p$
		$c = h_4(m)$
		$\mu \equiv (m\nu g^c)^y \bmod p$
	⟵ μ	
$c = h_4(m)$		
$\alpha \equiv \mu^{\frac{u}{b+u+c}} \bmod p$		
$k_1 = h_1(C, S, m, \mu, \alpha)$		
	k_1 ⟶	
		$k_1' = h_1(C, S, m, \mu, \beta)$
		abort if $k_1 \neq k_1'$
		$k_2 = h_2(C, S, m, \mu, \beta)$
	⟵ k_2	
$k_2' = h_2(C, S, m, \mu, \alpha)$		
abort if $k_2 \neq k_2'$		
$sk_c = h_3(C, S, m, \mu, \alpha)$		$sk_s = h_3(C, S, m, \mu, \beta)$

도면3

	클라이언트	서버
사전 계산량	1 exp (g^b) – 사용자 무관	1 exp (g^c) – 사용자 무관 1 exp (β) – 사용자 유관
실시간 계산량	1 inv ($u/(b+u+c)$) 1 exp (α) 2 hash (h_0, h_3) 2 hash (h_1, h_2)	1 exp (μ) 2 hash (h_1, h_2) 1 hash (h_3)