개선된 포워드 보안을 위한 인증 프로토콜

A improved authentication protocol for the forward security

원보스(Wenbo Shi)*, 장인주(InJoo Jang)*, 유형선(Hyeong Seon Yoo)**

초 록

본 논문에서는 사용자, 서비스 제공자, 그리고 키 분배센터인 KDC 사이의 인증 프로토콜과 키 분배기법에 관해 연구하였다. 본 논문에서 제안한 프로토콜은 대칭키 암호시스템과 시도-응답기법, Diffie-Hellman 기법, 해시 함수 등을 기반으로 하고 있으며, 사용자와 서버는 토큰 업데이트 과정에서 세션키를 업데이트한다. 업데이트된 세션키의 사용은 사용자에게 보다 보안성이 높은 인증과정을 수행 할 수 있도록 한다. 만일 KDC의 자료가 공격자에게 노출되었다 하더라도, 공격자가 세션키를 계산해 낼 수 없도록 설계되어 보안을 유지하게 된다. 본 논문에서는 제안된 프로토콜의 효율성과 진보된 보안성은 다른 프로토콜과의 비교와 분석을 통하여 우수함을 증명하였다.

ABSTRACT

This paper proposes a key distribution and authentication protocol between user, service provider and key distribution center (KDC). This protocol is based on symmetric cryptosystem, challenge-response, Diffie-Hellman component and hash function. In the proposed protocol, user and server update the session key under token-update operation, and user can process repeated efficient authentications by using updated session keys. And another merit is that KDC needs not to totally control the session key between user and server in proposed protocol. Even an attacker steals the parameters from the KDC, the attacker still can not calculate session key. According to the comparison and analysis with other protocols, our proposed protocol provides good efficiency and forward secure session key.

키워드 : 키 분배, 인증, 포워드 보안, 세션키
Key Distribution, Authentication, Forward Security, Session key

본 연구는 정보통신부 및 정보통신연구진흥원의 대학 IT연구센터 육성 지원사업의 연구 결과로 수행되었음.

* 인하대학교 컴퓨터정보공학 석사
** 인하대학교 컴퓨터 공학부 교수

1. 서 론

인터넷의 급증으로 많은 응용분야에서 보안 서비스의 필요성이 증가되고 있다. 사용자 인증과 키 분배 또한 신뢰성이 보장되지 않는 네트워크 상에서 매우 중요한 부분이다. Needham-Schroeder 인증 프로토콜을 기반으로 하고 있는 Kerberos 인증 시스템[1, 2]은 꽤 안정적인 표준이며, 보안적인 요소를 갖추고 있으나 많은 취약점 역시 지니고 있다. 특히, 패스워드 추측 공격과 재전송 공격, 세션키의 노출 등에 효과적으로 대처하지 못한다. 따라서 많은 공개키 기반의 프로토콜들이 Kerberos 인증시스템을 향상시키기 위해 제안되어 왔으며 중앙 중심적인 KDC에서부터 작업량을 개별적으로 분배하였으나, 구현에 있어 매우 높은 코스트가 필요하다[3-5].

KryptoKnight은 1992년, Kerberos와는 다른 키 분배 프로토콜과 새로운 인증 군의 구현으로 좀 더 유연한 프로토콜을 제안하였다[7, 8]. 1993년에 Paul Syverson은 KSL 프로토콜과 NS 프로토콜들을 향상시켰다. AUT HMAC_DH 프로토콜은 Kerberos의 결점을 극복하기 위하여 메시지 인증 코드를 사용하였다. 그러나 논의 되어진 대부분의 프로토콜에서는 사용자와 서버사이의 실제적인 세션키가 수립되지 못하였다고 할 수 있다[7-10].

Chien은 몇 가지 특별한 장점을 지니고 있는 프로토콜을 공개키와 시도-응답기법, 해시 함수등을 기반으로 제안하였다[11]. 이는 계산 비용적인 측면에서 사용자측의 부담은 줄였으나, 전체적인 계산 비용은 대칭키를 기반으로 한 다른 프로토콜에 비하여 높았다[11-13]. Hwang의 프로토콜은 Chien의 프로토콜보다 높은 효율성과 낮은 계산 비용을 지녔다[11, 13].

본 논문에서 제안하는 프로토콜은 개선된 보안성을 지닌 인증 및 키 분배 프로토콜이다. 이 프로토콜은 토큰 업데이트 과정을 통해 매번 세션키를 업데이트 한다. 또한 KDC의 작업 부담을 줄였으며, 공격자가 세션키를 계산해 낼 수 없도록 하여 보안성 향상을 가져왔다.

2. 제안된 토큰

본 논문에서 제안하는 프로토콜은 초기화 과정과 후속 과정의 두 부분으로 이루어진다. KDC는 사용자의 대칭키를 계산하기 위하여 비밀키 K_c를 유지한다. 사용자와 KDC사이에 유지되는 공유키는 K_{uc}이며, 서버와 KDC사이에 유지되는 공유키는 K_{sc}이다. 이 공유키의 생성과정은 $K_{uc} = f(K_c, U)$, $K_{sc} = f(K_c, S)$이며, $f(\cdot)$ 은 해시 함수이다.

2.1 초기화 과정

서버(S)로부터 서비스를 얻기 위해 사용자 (U)는 서버와의 세션키를 수립하는 초기화 과정을 수행해야 한다. 초기화의 과정은 다음과 같다.

(I.1) U->S $\{U, a^x \bmod p, h(a^x \bmod p, K_{uc})\}$

p 큰 소수이며, $a \in GF(p)$,

$h(\cdot)$: 일방향 해시 함수이다.

1 : 사용자 U는 난수 x를 선택하여 $a^x \bmod p$를 계산한다.

2: 사용자 U는 $h(\alpha^x \bmod p, Kuc)$를 계산한다.

3: 사용자 U는 그의 식별자(U)와, $\alpha^x \bmod p$, $h(\alpha^x \bmod p, Kuc)$를 서버에게 전송한다.

(I.2) S->KDC {U, $\alpha^x \bmod p$, $h(\alpha^x \bmod p, K_{uc})$, S, $\alpha^y \bmod p$, $h(\alpha^y \bmod p, K_{sc})$}

1: 서버 S는 값 $\alpha^x \bmod p$을 저장해 둔다.

2: 서버 S는 임의의 값 y를 선택하여 $\alpha^y \bmod p$를 계산한다.

3: 서버 S는 값 $h(\alpha^y \bmod p, K_{sc})$를 계산한다.

4: 서버 S는 사용자 U에게 값 $\alpha^x \bmod p$, $h(\alpha^x \bmod p, K_{uc})$을 그리고 KDC에게는 식별자($S$), $\alpha^y \bmod p$, $h(\alpha^y \bmod p, K_{sc})$를 전송한다.

(I.3) KDC -> S { $E_{Ksc}(\alpha^y \bmod p, n, r, E_{Kuc}(\alpha^x \bmod p, \alpha^y \bmod p, n, r))$}

1: KDC는 다음과 같이 해시 값 확인을 통하여 사용자와 서버를 인증한다.

$h(\alpha^x \bmod p, K_{uc}) = h(\alpha^x \bmod p, K_{uc})'$

$h(\alpha^y \bmod p, K_{sc}) = h(\alpha^y \bmod p, K_{sc})'$

2 : KDC 임의의 수 n을 선택하고 난수 r을 계산한다.

3: KDC는 값 $E_{Ksc}(\alpha^n \bmod p, n, r, E_{Kuc}(\alpha^x \bmod p, \alpha^y \bmod p, n, r))$를 서버에게 전송한다.

(I.4) S->U { $E_{Kuc}(\alpha^x \bmod p, \alpha^y \bmod p, n, r)$, $E_{Kn}(\alpha^x \bmod p \,||\, \alpha^y \bmod p)$}

1: S는 메시지 $EK_{sc}(\alpha^y \bmod p, n, r, EK_{uc}(\alpha^x \bmod p, \alpha^y \bmod p, n, r))$를 키 K_{sc}을 사용하여 복호화한다.

2: S는 계산 값 $\alpha^y \bmod p$를 확인하여 KDC를 인증한다.

3: S는 $\alpha^{xy} \bmod p$를 계산하여 K_n 값을 얻어내고, K_n 값을 사용하여 $\alpha^x \bmod p$, $\alpha^y \bmod p$을 암호화 한다.

4: S는 $E_{Kuc}(\alpha^x \bmod p, \alpha^y \bmod p, n, r)$, $E_{Kn}(\alpha^x \bmod p \,||\, \alpha^y \bmod p)$ 를 사용자에게 전송한다.

5: S는 U, K_n, n, r값을 유지한다.

(I.5) U receives {$E_{Kuc}(\alpha^x \bmod p, \alpha^y \bmod p, n, r)$, $E_{Kn}(\alpha^x \bmod p \,||\, \alpha^y \bmod p)$}

1: U는 메세지 $E_{Kuc}(\alpha^x \bmod p, \alpha^y \bmod p, n, r)$를 키 K_{uc}를 사용하여 복호화 한다.

2: U는 값 $\alpha^x \bmod p$를 확인함으로써 KDC를 인증한다.

3: U는 값 $\alpha^{xy} \bmod p$를 계산하여 K_n를 얻는다.

4: U는 K_n를 사용하여 ($\alpha^x \bmod p \,||\, \alpha^y \bmod p$)를 복호화 한다.

5: U는 값 $\alpha^x \bmod p$과 $\alpha^y \bmod p$를 확인하여 서버를 인증한다.

6: U는 K_n, n, r값들을 저장한다.

2.2 후속과정

후속 과정에서는 일반화 과정 없이, 사용자가 i번째 서비스를 요구하고 있는 과정을 설명한다.

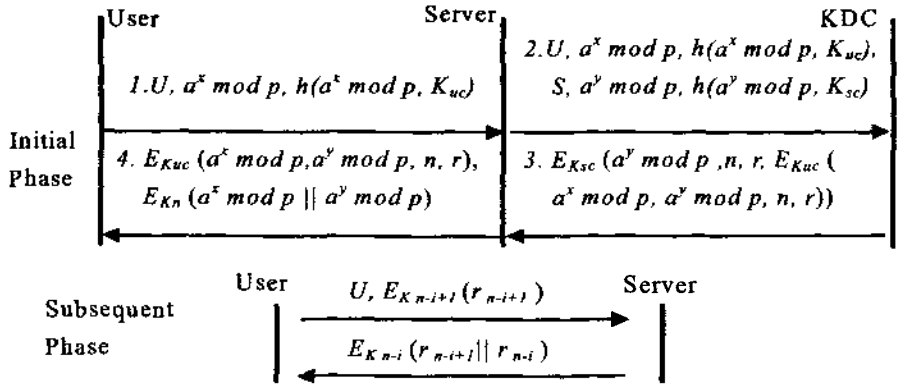
(S.1) U->S {U, $E_{Kn-i+1}(r_{n-i+1})$}

1: U는 키 K_{n-i+1}를 사용하여 r_{n-i+1}를 복호화 한다.

2: U는 서버에게 식별자 U와 $E_{Kn-i+1}(r_{n-i+1})$를 전송한다.

(S.2) S->U { $E_{Kn-i}(r_{n-i+1} \,||\, r_{n-i})$}

1: S는 키K_{n-i+1}를 사용하여 메세지 $E_{Kn-i-1}(r_{n-i+1})$를 복호화한다.



2: *S*는 값 $r_{n\,i+1}$?= r'_{n-i+1}의 확인하여 사용 자를 인증한다.

3: r_{n-i+1} == $r'_{n\,i+1}$인 조건에서 *S*는 $K_{n\,i}$ = $h(K_{n-i+1}$ || $r_{n-i+1})$를 계산한다.

4: *S*는 난수 r_{n-i}를 계산한다.

5: *S*는 값$(r_{n-i+1}$ || $r_{n-i})$를 키 K_{n-i}로 암호화 한다.

6: *S*는 값 K_{n-i}, r_{n-i},를 저장하고 *i*값을 업데이트한다.

(S.3) U receives { E_{Kn-i} (r_{n-i+1}|| $r_{n\,i}$)}

1: *U*는 값 K_{n-i} = $hash(K_{n-i+1}$ || $r_{n-i+1})$를 계산하다.

2: *U*는 값 E_{Kn-i} (r_{n-i+1} || r_{n-i})를 복호화하고, 값 r_{n-i+1} ? = r'_{n-i+1}의 확인을 통해 사용자를 인증한다.

3: r_{n-i+1} == r'_{n-i+1}인 조건에서 사용자는 K_{n-i}와 r_{n-i}를 저장하고 *i*를 업데이트 한다.

3. 보안 분석

3.1 Replaying attack

초기화 과정에서, 사용자와 서버는 임의 값 *x*와 *y*를 생성하여, α^x mod *p*와 α^y mod *p*값을 계산한다. 임의 수를 확인한 후 KDC는 대칭키를 이용하여 α^x mod *p*와 α^y mod *p*값을 각각 암호화 한다. *x*와 *y*값에 대하여는 생성자인 사용자와 서버가 매 로그인 마다 새로운 값을 선택함으로써 공격자가 얻어낸 기존 값으로는 접근 시도를 할 수 없게 된다.

후속과정에서 사용자는 K_{n-i+1}와 난수 r_{n-i+1}를 사용하여 값 K_{n-i} = $hash(K_{n-i+1}$ || $r_{n-i+1})$를 계산하고 서버에 로그인 하기 위하여 r_{n-i}를 암호화한다. 서버는 그 값을 복호화 함으로써 데이터의 유효성을 확인하고 사용자의 적합성 여부를 판단한다. 또한 세션키가 매 로그인 때 마다 업데이트 되기 때문에 재전송 공격에 관한 안전성이 유지 된다.

3.2 노출된 키에 대한 공격

제안된 프로토콜에서는 공격자가 사용자와 서버 간에 공유했던 세션키 K_{n-i+1}를 알아낸 경우라도, 매 로그인 과정에서 업데이트 되는 세션키가 보안성을 유지 시킨다. 또한 세션키를 생성하기 위해 필요한 난수가 역시 로그인 과정마다 업데이트 됨으로써 노출된 키에

대한 공격은 실패하게 된다.

제안된 프로토콜의 진보된 보안성을 증명하기 위하여, 다음의 세 가지 경우를 가정하고 살펴본다.

우선 공격자가 공유키 Kuc를 알고 있다고 가정 했을 때, 공격자는 사용자의 비밀 수인 x를 알 수 없기 때문에 $g^x \bmod p$, n ,r의 값을 얻은 경우라도 세션키의 계산이 불가능하다.

다음으로 공격자가 공유키 Ksc를 알고 있다고 가정한 경우, 공격자는 서버의 비밀 수인 y를 알 수 없기 때문에 공유키 Kuc를 알고 있는 경우와 같은 이유로 세션키는 안전하다.

만일 공격자가 공유키 Kuc와 Ksc를 알고 있다고 가정하더라도, 공격자가 x와 y를 알 수 있는 방법이 없으며, 이로 인하여 Kn을 얻을 수 없게 된다. 초기 과정에서 서버나 KDC가 impersonating attack을 받았다 하더라도 세션키는 여전히 안전함을 알 수 있다.

3.3 BAN 로직에 기반 한 인증과정 증명

이 절에서는 BAN 로직을 사용하여 제안된 프로토콜이 인증 절차와 키 분배에 있어 목적을 달성함을 증명하고자 한다.

사용자는 U, 서비스 제공자는 S, 키 분배 센터는 C로 표기 한다. 키는 K로 내용 X를 암호한 한 경우는 $\{X\}_K$로 표기한다. $<X>_Y$는 식 Y와 X를 결합함을 나타낸다. $Y \mid\Rightarrow X$ 는 Y가 X를 관단한다는 의미를 나타낸다. $Y \mid\equiv X$ 는 Y가 X를 신뢰한다는 것을 표현한다. $\#(X)$는 식 X가 신선함을 의미한다[14].

BAN 로직에 따라 다음와 같이 초기 가정을 설정한다.

1: $U \mid\equiv U \xleftrightarrow{K_{uc}} C$

2: $S \mid\equiv S \xleftrightarrow{K_{sc}} C$

3: $C \mid\equiv U \xleftrightarrow{K_{uc}} C$

4: $C \mid\equiv S \xleftrightarrow{K_{sc}} C$

5: $C \mid\equiv S \xleftrightarrow{K_{us}} U$

6: $S \mid\equiv (C \mid\Rightarrow S \xleftrightarrow{K_{us}} U)$

7: $U \mid\equiv (C \mid\Rightarrow S \xleftrightarrow{K_{us}} U)$

8: $U \mid\equiv \#(a^x \bmod p)$

9: $S \mid\equiv \#(a^y \bmod p)$

1번 가정에서 5번 가정은 사용자와 서비 키 분배 센터사이에 공유되는 키에 관한 내용이다. 가정 6번과 7번은 키분배 센터에 대한 사용자와 서버의 신뢰를 나타낸다. 가정 8번과 9번은 신선성을 고려하여 생성되는 nonce에 관하여 나타내고 있다.

본 논문이 제안하는 프로토콜이 갖추어야 할 인증과 키 분배에 목표를 BAN 로직을 사용하여 표현 하자면 아래와 같이 나타낼 수 있다.

1: $U \mid\equiv S \xleftrightarrow{K_{us}} U$

2: $S \mid\equiv S \xleftrightarrow{K_{us}} U$

3: $U \mid\equiv S \mid\equiv S \xleftrightarrow{K_{sc}} U$

4: $S \mid\equiv U \mid\equiv S \xleftrightarrow{K_{us}} U$

5: $U \mid\equiv C \mid\sim (r)$

6: $S \mid\equiv C \mid\sim (r)$

7: $C \mid\equiv U \mid\equiv (r)$

8: $C \mid\equiv S \mid\equiv (r)$

위에서 표시한 목표 중 1번에서 4번이 나

타내고 있는 것은 사용자가 분배된 키 K_n에 대해 서버가 신뢰하고 있음을 믿는다는 것과 서버 또한 사용자가 분배된 키를 신뢰함을 믿고 있음을 나타낸다. 5번과 8번은 키 분배 센터가 사용자와 서버가 분배 센터가 생성한 난수의 신선함을 믿는 다는 것을 나타낸다.

이를 표현하는 이상적인 프로토콜은 다음의 형태로 나타낼 수 있다.

Step 1: $U \rightarrow S < a^x \bmod p > k_{uc}$

Step 2: $S \rightarrow C < a^x \bmod p > k_{uc}$,
$< a^y \bmod p > k_{sc}$

Step 3: $C \rightarrow S \{ a^y \bmod p,\ S \overset{K_n}{\longleftrightarrow} U,\ r,$
$\{a^x \bmod p,\ a^y \bmod p,\ S \overset{K_n}{\longleftrightarrow} U,$
$r\}k_{uc}\}\ k_{sc}$

Step 4: $S \rightarrow U \{ a^x \bmod p,\ a^y \bmod p,$
$S \overset{K_n}{\longleftrightarrow} U,\ r\}k_{uc},\ \{a^x \bmod p,\ a^y \bmod p,$
$S \overset{K_n}{\longleftrightarrow} U\}k_n$

이를 간단히 증명하자면 아래와 같다.

위의 step 3과 가정 2로부터 message-meaning 규정을 적용하여 유도하면 아래와 같이 유도해 낼 수 있다.

"$S \models C \mid\sim (a^y \bmod p,\ S \overset{K_n}{\longleftrightarrow} U,\ r,$
$\{a^x \bmod p,\ a^y \bmod p,\ S \overset{K_n}{\longleftrightarrow} U,\ r\}k_{uc}$)" (1)

추론 (1)에 Belief-Conjuncatenation 규정을 적용하여 추론 (2)식을 얻는다.

"$S \models C \mid\sim (a^y \bmod p,\ S \overset{K_n}{\longleftrightarrow} U,\ r)$" (2)

추론 (2)에 Belief-Conjuncatenation 규정을 적용하여 추론 (3)식을 얻는다.

"$S \models C \mid\sim (r)$" (3)

추론 (3)과 가정 4에서 추론 (4)식을 얻는다.

"$C \models S \models C\ (r)$" (4)

가정 9에 Freshness-Conjuncatenation 규정을 적용하여 추론 (5)식을 얻는다.

"$S \models \#(a^y \bmod p,\ S \overset{K_n}{\longleftrightarrow} U,\ r)$" (5)

추론 (2)와 추론 (5)식에 Nounce-Verification 규정을 적용하여 추론 (6)식을 얻는다.

"$S \models C \models (a^y \bmod p,\ S \overset{K_n}{\longleftrightarrow} U,\ r)$" (6)

추론 (6)에 Belief-Conjuncatenation 규정을 적용하여 추론 (7)식을 얻는다.

"$S \models C \models (S \overset{K_n}{\longleftrightarrow} U)$" (7)

추론 (7)식과 가정 6에 Jurisdiction 규정을 적용하여 추론 (8)식을 얻는다.

"$S \models (S \overset{K_n}{\longleftrightarrow} U)$" (8)

스텝4와 가정1에 Message-Meaning 규정을 적용하여 추론 (9)식을 얻는다.

"$U \models C \mid\sim (a^x \bmod p,\ a^y \bmod p,$
$S \overset{K_n}{\longleftrightarrow} U,\ r)$" (9)

추론 (9)에 Belief-Conjuncatenation 규정을 적용하여 추론 (10)식을 얻는다.

"$U \models C \mid\sim (r)$" (10)

추론 (10)과 가정 3으로부터 추론 (11)식을 얻는다.

"$C \models U \models (r)$" (11)

가정8과 추론 (9)식에 Freshness-Conjuncatenation 규정과 Nounce-Verification 규정을 적용하여 추론 (12)식을 얻는다.

"$U \models C \models (a^x \bmod p,\ a^y \bmod p,\ S \xleftarrow{K_n} U,\ r)$" (12)

추론 (12)식과 가정 7에 Jurisdiction 규정을 적용하여 추론 (13)식을 얻는다.

"$U \models (S \xleftarrow{K_n} U)$" (13)

스텝 4와 추론 (13)식에 Message-Meaning 규정을 적용하여 추론 (14)식을 얻는다.

"$U \models S \mid\sim (a^x \bmod p,\ a^y \bmod p,\ S \xleftarrow{K_n} U)$" (14)

가정 8과 추론 (14)식에 Freshness-Conjuncatenation 규정을 적용하여 추론 (15)식을 얻는다.

"$U \models S \models (a^x \bmod p,\ a^y \bmod p,\ S \xleftarrow{K_n} U)$" (15)

추론 (15)에 Belief-Conjuncatenation 규정을 적용하여 추론(16)식을 얻는다.

"$U \models S \models (S \xleftarrow{K_n} U)$" (16)

만약 사용자가 세션키 K_n을 사용하여 암호화한 메시지를 서버에게 전송할 경우 추론 (17)식을 얻을 수 있다.

"$S \models U \models (S \xleftarrow{K_n} U)$" (17)

유도식 3, 4, 8, 10, 13, 16과 17에 의하여 우리는 본 논문에서 제안한 프로토콜이 목적하는 인증과 분배에 대한 목표를 달성할 수 있음을 증명 하였다.

4. 논의 사항

본 절에서는, 제안한 프로토콜을 기존의 다른 프로토콜들과 비교하고자 한다, 비교에 관한 요약은 <표 1>과 같다, Kerberos V5 프로토콜과 Shieh의 프로토콜은 키 분배 과정에서 대칭키를 사용하며, Chie의 프로토콜과 Huang 프로토콜 그리고 본 논문의 제안 프로토콜에서는 대칭키를 사용하지 않으며, 이로써 보안상 향상된 기능을 지니게 된다[11, 13].

KDC가 공격자에 의해 조절되고 있으며, 트로이 목마가 이식되었다고 가정할 때 각 프로토콜은 서로 다른 환경을 제공한다. Chie의 프로토콜과 Huang 프로토콜 상에서는 만일 공격자가 세션키 정보를 얻게 된다면, 공격자는 사용자와 서버간의 매 세션키를 알 수 있게 된다[11, 13]. 그러나 본 논문에서 제안하는 프로토콜의 경우는 공격자가 KDC의 모든 파라메터를 알게 되더라도, 세션키를 계산해 낼 수 없게 되어 보안상 향상된 기능을 제공한다.

<표 2>는 초기화 과정에서의 이 프로토콜들이 갖게 되는 계산 비용 부분에 대하여 비교한 것이다.

Diffie-Hellan에서의 소수 p의 길이는 1024비트로, RSA 알고리즘에서의 공개키 길이는 1024비트, AES에서의 대칭키 길이는 128비트, SHA-1의 해시함수는 160비트, DSA에서

〈표 1〉 프로토콜들의 비교표

	Kerberos V5[1]	Shieh [12]	Chie[11]	Huang[13]	제안 프로토콜
기본 암호 체계	SK	SK, Ce	PK	PK	SK, DH
초기과정/후속과정	4/2	5/3	4/2	4/2	4/2
Mounce/Clock	timestamp	nonce	nonce	nonce	nonce
대칭키 전송	yes	yes	no	no	no
Know Key 공격	Insecure[11, 13]	Insecure[11, 13]	Secure[11, 13]	Secure[13]	secure
사전 공격	Insecure[11]	Insecure[11]	Secure[11]	Secure	secure
KDC의 세션키 계산	yes	yes	yes	yes	no

주) SK : 대칭키, Ce : 인증서, PK : 공개키, DH : Diffie-Hellan.

〈표 2〉 초기 과정에서의 계산 비용에 관한 비교

	Shieh[12]			Chie[11]			Huang[13]			제안된 프로토콜		
	U	S	C	U	S	C	U	S	C	U	S	C
PKE	0	0	0	1	1	2	0	0	0	0	0	0
PKD	0	0	0	1	1	2	0	0	0	0	0	0
SKE	2	1	2	0	0	0	0	1	2	0	1	2
SKD	1	2	0	0	0	0	2	1	0	2	1	0
HF	0	0	0	1	1	n	n+2	2	n+4	0	0	0
others	0	0	0	1S, 1C	1S, 1C	1S, 2C	0	0	0	2ME	2ME	0

주) PKE : 공개키 암호화, PKD : 공개키 복호화, SKE : 대칭키 암호화, SKD : 대칭키 복호화
HF : 해시함수, S : signature, C : certification, ME : modular exponentiate.

의 사인길이는 320bit로 가정한다.

RSA의 계산 비용은 모듈러 멱승 연산으로 요약 할 수 있으며, 모듈러 멱승의 계산 비용은 약 O(|n|)시간이다. 모듈러 곱셈연산과의 비교에 있어서, 대칭 암호화, 복호화 의 시간은 무시될 수 있다[15]. 그리고 대칭 암호화 시스템이 비대칭 암호화 시스템 보다는 1000배 빠르며, 해시 함수가 대칭 암호 시스템 보다는 10배 더 빠르다[16, 17].

Shieh의 프로토콜은 초기화 과정에서 5번의 메시지 전송이 필요하며, 이것은 다른 프로토콜에 비해 효율이 떨어진다[12]. <표 2>에 따르면 제안된 프로토콜은 Chie[11] 프로토콜보다 효율적이며, Huang의 프로토콜은 Chie의 프로토콜과 제안된 프로토콜보다 초기과정을 수행함에 있어 효율적이다[11, 13]. 그러나 제안된 프로토콜은 미리 계산된 과정을 통해 통신 시간을 줄일 수 있다.

후속과정의 계산 시간 비교는 <표 3>에 정리하였다[11-13]. Shieh의 프로토콜은 후소과정 수행 시 3번의 메시지 전송이 이루어지게 된다. 이것은 다른 프로토콜에 비하여 비효율적이다. 후속 과정의 수행에 있어서는 제안된 프로토콜이 Huang의 프로토콜이나, Chie의 프로토콜에 비하여 더 효율적임을 알 수 있다[11, 13].

〈표 3〉 후속 과정에서의 계산 비용에 관한 비교

	Shieh[12]		Chie[11]		Huang[13]		제안된 프로토콜	
	U	S	U	S	U	S	U	S
SKE	n	n	n	n	n	n	n	n
SKD	n	n	n	n	n	n	n	n
HF	0	0	n(n+2)/n	3n	n(n-2)/n	3n	2n	2n
others	0	0	nPKE	nPKD,nS	0	0	0	0

주) PKE : 공개키 암호화, PKD : 공개키 복호화, SKE : 대칭키 암호화, SKD : 대칭키 복호화
HF : 해시함수, S : signature, C : certification, ME : modular exponentiate.

4. 결 론

본 논문에서는 보안성이 향상된 인증 프로토콜과 키 분배 프로토콜을 제안하였다. 이 프로토콜에서는 사용자와 서버간의 세션키 전반을 KDC가 제어했던 환경을 변화시켜, 사용자와 서버, KDC가 효율적으로 세션키를 관리하도록 하였다. 공격자가 비록 KDC로부터 이미 사용된 세션키를 얻어 냈다 하더라도, 공격자는 새로 이용할 세션키를 계산해 낼 수가 없게 된다. 이는 사용자와 서버가 세션키를 매 인증과정에서 업데이트 하기 때문이다. 제 3장과 제 4장의 분석을 통해 나타난 것과 같이 본 논문에서 제안한 프로토콜은 보안성이 향상되었으며, 좋은 효율성을 보인다.

참 고 문 헌

[1] Kohl and Neuman, "The Kerberos network authentication service (v5)," Internet Request for Comments RFC-1510, 1993.

[2] Needhamand, R. M. and Schrooder, M. D., "Using encryption for authentication in large networks of computers," Communication of the ACM, Vol. 21, No. 12, 1978, pp. 993-999.

[3] Neuman, B. C. and Ts'o, T., "Kerberos: An authentication service for computer networks," IEEE Communications, Vol. 32, No. 9, 1994, pp. 33-38.

[4] Bellovin, S. M. and Merrit, M., "Limitations of the Kerberos authentication system," Computer Communication Review, Vol. 20, No. 5, 1990, pp. 119-132.

[5] Ganesan, R., "Yaksha: augmenting Kerberos with public key cryptography," SNDSS 1995 : Proceedings of the 1995 Symposium on Network and Distributed System Security, IEEE Computer Society, 1995, pp. 132-143.

[6] Sirbu, M. A. and Chuang, J. C. I., "Distributed authentication in Kerberos using public key cryptography," Proceedings of the 1997 Symposium on Network and Distributed System Security, IEEE Computer Society, 1997, pp.

134-141.

[7] Refik Molva, Gene Tsudik, Els van Herreweghen and Stefano Zatti, "KryptoKnight authentication and key Distribution System," ESORICS, 1992, pp. 155-174.

[8] Bird, R., Gopal, I., Herzberg, A., Janson, P., Kutten, S., Molva, R., and Yung, M., "The KryptoKnight family of Light-weight protocols for authentication and key distribution," IEEE Transactions on Networking, Vol. 3, No. 1, 1995, pp. 31-42.

[9] Paul Syverson, "On key distribution protocols for repeated authentication," ACM SIGOPS Operating Systems Review, Vol. 27, No. 4, 1993, pp. 24-30.

[10] Heba, K. Aslan, "Logical analysis of AUTHMAC_DH : a new protocol for authentication and key distribution," Computers & Security, Vol. 23, No. 4, 2004, pp. 290-299.

[11] Chien, H. Y. and Jan, J. K., "A hybrid authentication protocol for large mobile network," Journal of Systems and Software, Vol. 67, No. 2, 2003, pp. 123-130.

[12] Shieh, S. P., Ho, F. S., and Huang, Y. L., "An efficient authentication protocol for mobile networks," Information Science and Engineering, Vol. 15, No. 4, 1999, pp. 505-520.

[13] Ren-Junn Hwang and Feng-Fu Su, "A new efficient authentication protocol for mobile networks," Computer Standards & Interfaces, Vol. 28, No. 2, 2005, pp. 241-25.

[14] Burrows, M., Abadi, M., and Needham, R., "A logic of authentication," ACM Transactions on Computer Systems, Vol. 8, No. 1, 1990, p. 1836.

[15] Chun-I Fan, Yung-Cheng Chan and Zhi-Kai Zhang, "Robust remote authentication scheme with smart cards," Computers & Security, Vol. 24, No. 8, 2005, pp. 619-628.

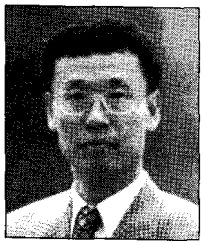
[16] Hwang, M. S., Lin, I. C., and Li, L. H., "A simple micro-payment scheme," Journal of Systems and Software, Vol. 55, No. 3, 2001, pp. 221-229.

[17] Chang, C. C., Lail, C. S., and Harn, L., *Contemporary Cryptography and its Applications, 2nd ed., Unalis Co*, 2001.

저 자 소 개

원보스 (E-mail : swb319@hotmail.com)
Chengdu university of technology in China, Department of Computer science and technology, MS
현재 인하대학교 컴퓨터정보공학 (석사)
관심분야 Applied Cryptography

장인주 (E-mail : jangij@dreamwiz.com)
현재 인하대학교 컴퓨터정보공학 (석사)
관심분야 Applied Cryptography



유형선 (E-mail : hsyoo@inha.ac.kr)
Ghent University, Belgium 기계공학 (박사)
현재 인하대학교 컴퓨터 공학부 교수
관심분야 Applied Cryptography, Scientific Computation