패스워드 기반의 커버로스 인증 메커니즘 설계

조경옥* · 김종우*· 하태진*· 한승조*
*조선대학교

Design of a Kerberos Authentication Mechanism based on Password

Kyoung-Ok Cho*, Jong-Woo Kim*, Tae-Jin Ha*, Seung-Jo Han*
*Chosun University
E-mail : sign57421@orgio.net

본 연구는 산업자원부의 지역현신 인력양성사업의 연구결과로 수행되었음

요 약

분산 네트워크 환경에서 커버로스 인증 메커니즘은 Local 영역에 있는 사용자가 다른 영역에 존재하는 커버로스 서버의 신뢰성이 있는 전제 조건에서 운용하고 있다. 하지만 커버로스 서버간의 인증 서버의 보안 정보가 누설되면 커버로스에 대한 신뢰성이 보장되지 않는 단점을 가지고 있다. 이런한 문제점을 해결하기 위하여 기존의 커버로스 메커니즘에서 제안한 인증센터와 인증서를 사용하지 않고 분할된 패스워드 사용을 통하여 패스워드 검증자의 랜덤성을 증가시켜 패스워드 추측공격이 어렵도록 하였으며 비밀 분산 기법을 적용한 패스워드 기반 인증 방법을 사용한 인증 메커니즘을 제안한다.

ABSTRACT

In a distributed network system, Kerberos certification mechanism is operated by a user in local area on the premise reliability of Kerberos server in another area. But it has a demerit. If security information of certification server between Kerberos servers is released, Kerberos server can not guarantee the reliability. To solve this problem, the proposed mechanism prevents password speculating attack by increasing the random of password certifier through use of distributed password in stead of certification center and certification which was presented by existing Kerberos mechanism. Besides, it used password based certification method which uses secret distributed technique.

키워드
커버로스, 인증, 패스워드

Ⅰ. 서 론

분산 네트워크를 통한 안전한 정보 서비스를 제공하기 위해서는 사용자와의 인증은 필수적이다. 그러나, 인증 자체만으로는 문제가 발생하여 인증(Authentication)과 함께 허가(Authorization)는 분산 시스템 환경에서 보안은 필수 항목이다. 허가는 사용자, 프로그램, 프로세스에게 허가한 권한을 의미하며 분산 네트워크에서는 권한 허가를 연결해서 사용하도록 하고 있다. 특히 분산네트워크 환경에서의 문제점은 비인가 사용자가 인가된 사용자로 가장하여 접속을 시도, 자료수정 등의 정보서비스 사용을 위협하는 대표적인 위험요소이다.

본 논문에서는 분산 네트워크에서 가장 대표적인 인증 메커니즘인 커버로스 인증에 관해서 중점적으로 다루고 있다. 커버로스 메커니즘은 각 네트워크상에서 키관리를 포함한 제반 관리 기능을 중앙 집중식으로 처리하는 커버로스 인증 서버가 있다.[1][2]

만약 인증 서버의 보안 정보가 누설되면 커버로스에 대한 신뢰성이 보장되지 않는 단점을 가지고 있다.[3] 또한 서로 다른 영역들 간에 사용되기 위해서는 좀 더 강력한 기능과 특성을 가진 커버로스 메커니즘이 있어야 할 것이다. 이러한 제약사항을

보안하기 위해 IETF(internet Engineering Task Force)의 CAT Working Group에서는 커버로스 시스템 두 영역간의 인증 서버를 공개키로 상호 서비스해 주는 PKINIT(Public Key Cryptography for Initial Authentication)/PKCROSS(Public Key Cryptography for Cross-Realm Authentication)을 발표하였다.[4][5]

본 논문에서는 기존의 커버로스 메커니즘에서 제안한 인증센터와 인증서를 사용하지 않고 분할된 패스워드 기반 인증 방법을 사용한 인증 메커니즘을 제안한다. 분할된 패스워드 사용을 통하여 패스워드 검증자의 랜덤성을 증가시켜 패스워드 추측공격이 어렵도록 하였으며 비밀 분산 기법을 적용하여 서버가 타협이 되어도 사전공격, 서버 가장 공격이 어렵도록 하였다.

본 논문의 구성은 다음과 같다. 제 2장에서는 커버로스 인증 메커니즘과 분할된 패스워드 기반의 인증방법을 설명한다. 3장은 제안되는 분할된 패스워드 기반의 인증 메커니즘을 설명하고 4장에서 기존의 인증 메커니즘과 제안된 인증메커니즘을 비교 분석하여 결론은 마지막장에서 한다.

II. 인증방법

2.1 커버로스

커버로스는 구성요소로 커버로스 서버와 티켓승인서버(TGS), 티켓(Ticket), 인증자(Authenticator)로 구성되어 있다. 커버로스 인증 절차 메커니즘은 다음 그림[1]과 같다.

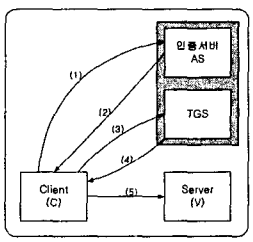

그림 5. 커버로스

(1) Client - AS 간의 교환

<메시지 1>

$Client \rightarrow AS : Options \| ID_C \| Realm_C \| ID_{TGS} \| Times1 \| Nonce1 \| AD_C$

Client는 자신의 ID와 TGS 사용을 허가하는 TGS의 ID을 AS에 전송함으로써 Ticket 승인 요청한다.

<메시지 2>

$AS \rightarrow Client : Realm_C \| ID_C \| Ticket_{TGS} \| E_{KC}[\, K_{C,TGS} \| Nonce1 \| Flag \| Times1 \| Realm_{TGS} \| ID_{TGS} \| AD_C \,]$

$Ticket_{TGS} = E_{KTGS}[\, Flag \| K_{C,TGS} \| Realm_C \| ID_C \| Times1 \| AD_C \,]$

AS는 이미 정의된 키 K_C(long-term Key)를 사용하여 TGS와 통신에 사용할 키 K_C등을 암호화하여 사용자로 전송 및 $Ticket_{TGS}$을 발급한다.

(2) Client - TGS 간의 교환

<메시지 3>

$Client \rightarrow TGS : Options \| ID_V \| Times2 \| Nonce2 \| Ticket_{TGS} \| Authenticator_C$

$Authenticator_C = E_{KC,TGS}[\, ID_C \| Realm_C \| TS1 \,]$

Client는 $Ticket_{TGS}$과 $Authenticator_C$를 TGS에게 제공하고 서비스 승인티켓을 요청한다.

<메시지 4>

$TGS \rightarrow Client : Realm_C \| ID_C \| Ticket_v \| E_{kC,TGS}[\, K_{C,v} \| Nonce2 \| Flag \| Times2 \| Realm_v \| ID_v \| AD_C \,]$

$Ticket_v = E_{kv}[\, Flag \| K_{C,v} \| Realm_C \| ID_C \| Times1 \| AD_C \,]$

TGS는 $Ticket_v$을 Client에게 발급하고 $Ticket_v$는 Client와 Server에 사용할 $K_{C,v}$을 포함하며 Client와 TGS간의 $K_{C,TGS}$는 Client와 Server간의 $K_{C,v}$을 이용하여 암호화한다.

(3) Client - Server의 교환

<메시지 5>

$Client \rightarrow Server : Options \| Ticket_v \| Authenticator_C$

Client는 $Ticket_v$과 $Authenticator_C$를 Server에 제공함으로 별도 승인 없이 Server에 접속할 수 있다.

2.2 분할 패스워드 인증방법

분할된 각각에 패스워드 지식을 랜덤한 엔트로피를 갖는 정보로 묶어 패스워드에 대한 추측을 낮추고 서버가 유지하는 패스워드 검증자를 암호화용 키가 서로 연관이 되지 않는 한 패스워드 검증자를 안전하게 보관하도록 하여 상호 인증하는 방법이이 그림[2]과 같다.[7][9]

상호인증과 상호 키확신을 위하여 TGS_i에서는 Au_i의 계산과 TGS_r에서는 Au_r을 계산하였으며 TGS_r이 정확한 패스워드 검증자(e, τ)를 알고 있고 세션값 $km_s = g^{xy}$를 계산했음을 TGS_1에게 증명한다. Au_c는 TGS_1이 정확한 패스워드 π를 알고 있고 세션값 $km_c = g^{xy}$에 대하여 계산하였음을 TGS_r에게 증명을 한다.

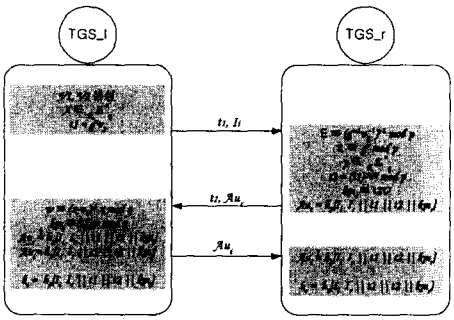

그림 6. 분할 패스워드 인증 방법

다수로 분할된 패스워드를 이용할 경우는 TGS_i는 패스워드를 n으로 분할된 형태로 확장할 수가 있는데 이는 랜덤성을 더욱 증가시키기 위한 것으로 자신의 패스워드 π를 $\pi_1 \| \pi_2 \| \cdots \| \pi_n$으로 분할 후, 각 패스워드에 대하여 $v_i = h_1(I_c \| I_s, \pi_i \| i)(i = 1, \cdots, n)$으로 계산한다.

(1) TGS_l는 $v_1, \cdots, v_n$를 계산하고 $x \in_R Z_q^*$를 생성한 후, $t_1 = g^x(\Pi_{j=a+1}^{n} v_1)$와 I_c를 TGS_r에 전달한다.

(2) y, t_2, km_r, Au_r를 생성한 후, t_2, Au_r를 TGS_l에 전달한다.

(3) TGS_l는 $w = (x - \Pi_{i=1}^{a} v_i)^{-x}$를 계산하고 km_r를 생성한후 TGS_r에게 Au_i를 전달한다.

(4) 최종적으로 TGS_l와 TGS_r 모두가 상호인증과 키확신 검사를 하면 동의된 세션키(k_i, k_s)를 생성한다.

III. 제안된 인증 메커니즘

본 논문에서는 기존의 커버로스 메커니즘의 방법을 채택하면서 기존의 공개키/개인키 인증방식이 아닌 분할된 패스워드 인증방법을 사용하여 보다 강력한 인증 메커니즘을 제안한다.

기존의 커버로스 방식은 패스워드를 교환하는데 long-term키로 만들어 티켓을 암호화하거나 통신에 세션키로 사용한다. 하지만 매번 같은 long-term 키로 암호화를 하기 때문에 사전 공격 등을 통해 키 값이 해독되거나 노출될 가망성이 있다. 또한 공개키/개인키 인증방식을 사용 할 경우 지역 커버로스와 원격 커버로스가 세션키를 다른 사람들이 모르게 공유해야 한다는 점이다. 지역 커버로스 수신자가 멀티 떨어져 있다면 두 커버로스 사이에 안전한 통신 수단을 이용하여 세션키를 전달하여야 할 것이다. 만약 공격자가 이 두 커버로스 의 통신 경로상에서 세션키를 얻어낸다면 그 후로부터 두 커버로스가 세션키를 사용하여 메세지를 암호화한다 하여도 공격자는 훔친 세션키를 이용하여 모든 메시지 복호화하여 메시지 내용을 알 수가 있게 될 것이다. 다수의 커버로스가 동시에 상호 메세지를 주고받고자 하는 경우에는 공개키의 관리가 매우 어려워진다.

본 논문에서 제안하고 있는 메커니즘은 시스템간에 패스워드를 분할하여 인증하는 방법을 사용하여 안전성을 높이고 사전공격이나 오프라인 사전추측 공격에 대한 저항성을 지닌다.

3.1 제안 인증 메커니즘

기존의 커버로스는 long-term키로 만들어 사용하는데 사용자와 지역 KDC간의 통신에 세션키와 티켓을 암호화하는 비밀키로 사용한다. 계속해서 같은 값의 세션키와 비밀키를 사용함으로써 노출되거나 사전공격의 위험성을 지닌다.

본 논문에서 제안하는 long-term키를 이용한 세션키 생성은 KDC간에 랜덤 수를 교환하고 랜덤수와 long-term키를 조합하여 통신에 사용하는 세션키를 분할된 패스워드 인증 기법을 사용하여 그림[3]와 같이 세션키를 분배한다.

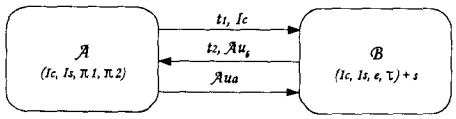

그림 7. 분할된 패스워드 인증 메커니즘

(1) $A \rightarrow B : t_1, I_c$

$t_1 = g^x v_2 \bmod p$, I_c = A의 ID

(2) $B \rightarrow A : t_2, Au_b$

$t_2 = (t_1)^y e^{xy} \bmod p$,

$Au_b = h_2(I_C \| I_s \| t_1 \| t_2 \| km_s)$

(3) $A \rightarrow B : Au_a$

$Au_a = h_3(I_C \| I_s \| t_1 \| t_2 \| km_c)$

3.2 제안 인증 프로토콜

본 논문에서 제안하는 분할된 패스워드를 이용한 인증 메커니즘의 인증 프로토콜은 그림[4]와 같다.

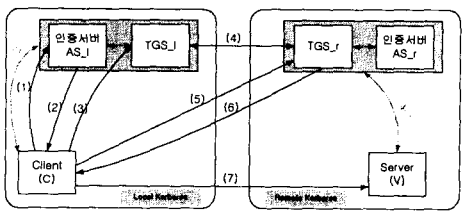

그림 8. 분할된 패스워드를 이용한 인증 메커니즘

제안한 인증 메커니즘은 4단계로 구성이 된다.

3.2.1 Client 인증 단계

Client 인증단계는 사용자와 Local 인증서버간의

인증과정을 나타낸다.

Client는 AS_l에 TGS 서버 사용허가 Ticket 승인 요청한다.

AS_l는 Client가 유효한 사용자인지 데이터베이스를 검색한 후 유효한 사용자이면 TGS_l과 통신에 사용할 키 K_C등을 암호화하여 Client로 전송 및 $Ticket_{TGS_l}$을 발급한다.

<메시지
1>
$Client \rightarrow AS\_l : Options \| ID_C \| ID_{TGS\_l} \| Times1 \| Nonce1 \| AD_C$

<메시지
2>
$AS_l \rightarrow Client : ID_C \| Ticket_{TGS\_l} \| E_{KC}[K_{C,TGS\_l} \| Nonce1 \| Flag \| Times2 \| ID_{TGS\_l} \| AD_C]$
$Ticket_{TGS\_l} = E_{KTGS\_l}[Flag \| K_{C,TGS} \| Realm_{TGS\_l} \| ID_{TGS\_l} \| Times2 \| AD_C]$

3.2.2 Client - TGS_l 간의 인증 단계

Client는 응용 서버와 통신을 하기 위하여 Client는 TGS_l에 $Ticket_{TGS_l}$과 $Authenticator_C$를 TGS_l에게 제공하고 원격 승인 티켓인 TGS_r의 티켓-승인 티켓 신청한다.

<메시지
3>
$Client \rightarrow TGS\_l : Options \| ID_{TGS\_r} \| Times3 \| Nonce3 \| Ticket_{TGS\_l} \| Authenticator_C$
$Authenticator_C = E_{KC,TGS\_l}[ID_C \| Times3 \| AD_c]$

3.2.3 TGS_l - TGS_r간의 상호 인증 단계

TGS_l은 $v_1 = h_1 (ID_{TGS\_l} \| ID_{TGS\_r} \| \pi_1 \| '1')$ $v_2 = h_2 (ID_{TGS_l} \| ID_{TGS_r} \| \pi_2 \| '2')$을 구한 후 TGS_r에 t1과 TGS_l의 ID를 TGS_r에 전달한다.

<메시지
4>
$TGS\_l \rightarrow TGS\_r : ID_{TGS\_r} \| Flag \| t_1 \| ID_{TGS\_l} \| AD_C \| Nonce4 \| Times4]$
$t_1 = g^x v_2 \bmod p$

TGS_r은 ID_{TGS_l}에 해당되는 e, r를 찾아내고 세션값 $km_s = g^{xy}$를 계산했음을 TGS_l에게 증명한다. 마찬가지로 Au_{TGS_l}은 TGS_l가 정확한 패스워드 π를 알고 있고 세션값 $km_s = g^{xy}$에 대하여 계산했음을 TGS_r에게 증명한다.

<메시지
5>
$TGS\_r \rightarrow TGS\_l : ID_{TGS\_r} \| Nonce5 \| sign_{TGS\_r}$
$sign_{TGS\_r} : E_{TGS\_l}[Nonce4 \| times4 \| t_2 \| Au_{TGS\_r}]$
$Au_{TGS\_r} = h_2(ID_{TGS\_l} \| ID_{TGS\_r} \| t_1 \| t_2 \| km_{TGS\_r})$
$t_2 = (t_1)^y e^{xy} \bmod p$

TGS_l에서 $w = (x - \Pi_{i=1}^{e} v_i)^{-x}$를 계산하고 km_r를 생성한 후 TGS_l의 공개키로 암호화하여 티켓-승 티켓과 신분을 확인할 수 있는 인증자를 TGS_r에게 전달함으로써 원격 TGS와 사용자간에 사용할 세션키를 분배 받을 수 있다.

<메시지
6>
$TGS\_l \rightarrow TGS\_r : ID_{TGS\_l} \| Flag \| Au_{TGS\_l} \| ID_{TGS\_l} \| Ticket_{TGS\_r} \| E_{TGS\_r}[Nonce_l] \| Authenticator_{cr}]$
$sign_{TGS\_l} : E_{TGS\_l}[Nonce4 \| Nonce5 \| times5 \| t_2 \| Au_{TGS\_r}]$
$Au_{TGS\_l} = h_3(ID_{TGS\_l} \| ID_{TGS\_r} \| t_1 \| t_2 \| km_{TGS\_l})$
$Ticket_{TGS\_r} = E_{TGS\_r}[K_{c,TGS\_l} \| ID_C \| AD_C \| ID_{TGS\_l} \| ID_{TGS\_r} \| Nonce6 \| Times6]$

3.2.4 Client - Remote TGS의 교환

Client는 $Ticket_{TGS\_r}$(티켓-승인 티켓)과 사용자의 인증자 정보를 TGS_r에 제공하여 유효한 사용자인지를 확인한 후 Server의 TGS_r은 K_{C,TGS_r}(세션키)을 구하고 Client와 Server에 사용할 $K_{C,v}$(세션키)을 생성하며 Client와 Server간의 사용할 $Ticket_v$을 발급한다.

<메시지
7>
$Client \rightarrow TGS\_r : Options \| ID_S \| Times7 \| Nonce7 \| Ticket_{TGS\_r} \| Authenticator_C$

<메시지
8>
$TGS_r \rightarrow Client : ID_C \| Ticket_v \| E_{KC,TGS\_r}[K_{C,V} \| Flag \| Times8 \| Nonce8 \| ID_v \| AD_C]$

(4) Client - Server의 교환

Client는 $Ticket_v$과 $Authenticator_C$를 Server에 제공함으로 별도 승인 없이 Server에 접속할 수 있다.

<메시지 9>

$Client \rightarrow Server : Options \parallel Ticket_v \parallel Authenticator_C$

IV. 분할된 패스워드 인증 메커니즘 분석

본 논문에서 제안한 분할된 패스워드 인증 기법을 사용한 커버로스 인증 메커니즘은 기존의 커버로스 메커니즘을 최대한 활용하였으며 안전성과 사전공격에 탁월한 개념의 메커니즘이다.

기존의 커버로스는 영역간의 신뢰성이 결여되어 있다. 또한 사전 공격등으로 long-term키가 노출되면 인증에 문제점이 발생하게된다. 커버로스 서버간 인증엣 사전에 약속된 long-term키를 분할된 패스워드 인증 기법을 사용한다면 성호영역간이나 시스템간에 신뢰성을 높일 수 있을 것이다. 제안된 방법은 패스워드 분할하여 사용함으로써 공격자는 더 많은 정보를 분석해야만 하며 커버로스 서버간의 메시지 교환이 수행되는 동안에는 세션값에 대한 어떠한 내용도 공격자에게 노출되지 않는다. 공격자는 참여자간에 교환되는 메시지를 도청할 수도 있고 그 사이에서 공유되는 세션키를 구하려 한다. 그러나 임의의 메시지를 바꾸거나 삭제하거나 삽입하는 것은 불가능하게 된다. 서로간의 통신하면서 발생하는 정보를 알고 있다 하더라도 공격자가 세션값 g^{xy}를 알아내는 것은 매우 어렵다.

만약 세션키가 공개되어도 패스워드는 노출되지 않을 뿐 아니라 사전공격에도 안전하다. 이전 사용된 세션키를 이용하여 공격이 불가능하다. 이는 패스워드가 아닌 t_1, t_2, km을 얻을 수 있어도 공격자가 패스워드 π 및 검증자 (e, τ)을 얻어 낼 수는 없다. 패스워드를 알아내기 위해서는 t_1, t_2, km으로부터 $(x, y, v_1 x \leftarrow Z_q^*)$, $(v_2 \leftarrow Z_q^*)$을 구별할 수 있어야 하고 DLP를 해결 할 수 있어야 한다.

제안된 프로토콜은 전방향 안전성이 제공되어지며 long-term 이전에 생성된 세션값 분실을 의미하지 않는다면 공격자에게 패스워드가 주어졌다고 가정했을지라도 공격자는 v_1, v_2만을 구할 수 있을 뿐 세션값을 구하는 것은 불가능하다. TGS서버간의 통신을 하는데 있어서 주고 받는 정보 t_1, t_2에 대한 DLP를 해결해야만 사전공격이 가능하기 때문에 논 논문에 사용된 프로토콜은 수행중에 노출되는 정보들을 통해서도 오프라인 사전추측 공격이 불가능함을 의미한다.

또한 공개키/개인키 방식은 위장 공격에 취약하다는 단점이 있다. 하지만 본 논문에서 제안하는 방식은 공격자가 양쪽 개체에 합법적으로 가장하거나 두 서버간의 메시지를 가로챈 다음 별도의 세션값을 만들어내는 공격을 할 경우 가장공격과 동일하게 모든 대화 내용을 알고 있더라도 패스워드를 모른다면 공격은 불가능하다.

본 논문에서 제안하는 메커니즘은 어떠한 공격에도 강력하게 대처 할 수 있고 공개키/개인키 기반에서의 메커니즘에 비하여 신뢰센터를 운영하는데 있어서 들어가는 통신비용을 절감할 수 있을 뿐 아니라 키 생성과 관리에 대한 문제점도 줄어들게 될 것이다.

IV. 결 론

본 논문은 분산환경의 인증 메커니즘인 커버로스를 바탕으로 커버로스 서버간에 키 교환시 패스워드 인증방식 중 분할된 패스워드 인증방식을 첨가함으로써 강력한 인증 메커니즘을 제안했다. 기존의 대칭키를 사용에 따르는 보안에 대한 문제점을 극복하였을 뿐 아니라 공개키/개인키를 사용한 메커니즘에 비하여 사전 공격에도 강한 인증방법이다. 공개키/개인키 방식은 대칭키에 비해 키관리는 편리하지만 알고리즘이 더 복잡하기 때문에 처리속도가 느려져 메시지가 커질 경우 부담은 더욱 크게 증가로 인한 대규모의 네트워크에서 사용은 무리가 갈 것이다. 사전에 교환된 패스워드 기반에서 사용되는 long-term 키의 한계를 극복하여 세션키가 공개되더라고 패스워드는 노출되지 않는 장점을 가지고 있다.

공개키/개인키 인증방법에서 사용하는 신뢰센터에서 분배되면서 발생하는 인증서 관리문제와 통신비용이 늘어나는 문제점을 안고 있다. 본 논문에서 제안된 인증 메커니즘은 외부 영역 TGS에 접근하는 방법을 단순화 시켜 서비스-승인 티켓을 얻는 인증절차를 단순화 시켰을 뿐 아니라 기존의 커버로스 시스템을 활용할 수 있는 효율적인 방법의 인증 메커니즘이다.

본 논문은 다른 어떤 커버로스보다 효율적이고 강력한 인증 메커니즘으로 앞으로 구현함에 있어서 소규모 네트워크 뿐아니라 대규모의 네트워크에서도 활용이 가능할 것이다.

참고문헌

[1] J. Kohl and C.Neuman, "The Kerberos Network Authentication Service (V5)." RFC 1510. September 1993.

[2] A. Garbitter and D. Menasce. "Performance of Public Key-Enabled Kerberos Authentication in Large Networks." Proc. 2001 IEEE Symposium on Security and Privacy, Oakland, CA, May 13-16. 2001

[3] John T. Kohl. B. Clifford Neuman, Theodore Y. T'so, "The Evolution of the Kerberos Authentication System, pages 78-94. IEEE Computer Society Press, 1994.
[4] B. Tung, B. C. Neuman, M. Gur, A.<edvinsky, S. Medvinsky, "Public Key Cryptography for Cross-Realm Authentication in Kerberos", draft- iegf- cat -kerberos-pk-closs-07.txt
[5] 신광철, 정진욱, "네트웍 환경에서 안전한 Kerberos 인증 메커니즘에 관한 연구." 정보보호학회 논문지, 제12권 2호. April 2002.
[6] P.Mackenzie, T.Shrimpton, and M.Jakobsson, "Threshold Password-Authenticated Key Exchange," CRYPTO 2002
[7] S.Bellovin and M.Merritt, "Encryted key exchange dictionary attacks," Proc. IEEE Comp. Society Symp. on Research in Security and Privacy, pp. 72-84, 1992
[8] Uecision Diffie-Hellman, and Discrete Logarithms," ISIT 1998, Cambrige, MA, USA, August 21-26
[9] Taekyoung Kwon, "Ultimate Solution to Authentication via Memorable Password, IEEE P1363 study group