키 복구를 지원하는 공개키 인증시스템

신광철*, 정진욱*
* 성균관대학교 정보공학과
e-mail : skcskc12@yahoo.co.kr

On Design of the Recoverable Public Key Authentication System

Kwang-Cheul Shin*, Jin-Wook Chung*
* Dept. of Information and Communications, Sungkyunkwan University,

요 약

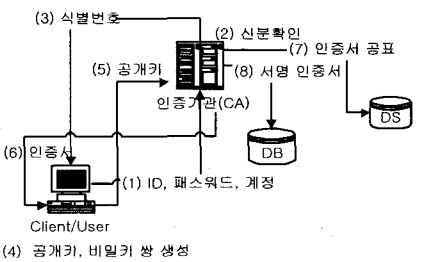
전자상거래, Single Sign On, 전자문서교환 등 웹 기반의 사이버 교환이 급증하는 오늘날 상대방의 신뢰를 보증해 줄 수 있는 수단으로 인증서의 사용이 필수적이다. 인증서는 인증기관을 신뢰한다는 가정에서 사용되기 때문에 통신개체 사이에서 보다 정확한 인증이 필요하다. 본 논문에서는 공개키 기반구조에서 인증서를 사용하는 모든 암호시스템 분야에 응용될 수 있는 인증 및 키 복구 메커니즘을 설계하였다. 서버와 사용자간에 인증서를 기반으로 공개키 암호에 의한 인증과 세션 암호 키 분배, 그리고 응용 자원서버들과의 비밀통신에서 암호 키의 유실을 고려한 키 복구 지원 프로토콜을 제안하였다.

1. 서론

공개키 기반구조는 사용자의 공개키에 대한 인증서를 발행하고 관리를 해 주며 암호학적 키와 인증서의 배달 시스템으로 여러 응용분야에서 인증서의 사용을 용이하도록 하는 정책, 수단, 도구 등을 수립하고 제공하는 개체들의 네트워이다[1][2][3]. 본 논문에서는 PKI 구조에서 X.509 인증서로 운용되는 모든 암호시스템에 적용시킬 수 있는 초기인증과 안전한 키 분배 시스템을 설계하였다. PKI를 기반으로 각 개체가 생성한 공개정보를 이용하여 사용자 개체와 인증서버(AS: Authentication Server)간의 신뢰성을 보장하는 인증과 인증에 필요한 교환정보를 기초로 하여 Diffie-Hellman(이후D-H로 표기)키 교환 방식을 이용함으로써 키 복구를 제공하는 PKI 연동 인증시스템을 설계하였다. 공개키 암호에 의한 인증과정에서 영역에 등록하는 모든 개체가 생성하여 전송한 공개키(subjectPublicKey) 정보를 보관하고 키 복구를 위해 subjectPublicKey를 교환함으로써 새로운 인증스킴을 제안한다.

2. 초기등록과 인증서 발행

PKI는 참여하는 모든 사용자들의 공개키가 공개될 수 있는 장소와 방법이 필요하게 되고 이를 안전하게 관리하여 위.변조가 되지 않는 공개키를 보장해 주어야 하며 참여자들이 사용하는 공개키에 대해 신뢰할 수 있는 인증서를 발급해 주어야 한다[4][5].



[그림 1] 공개키 등록 및 인증서 발행

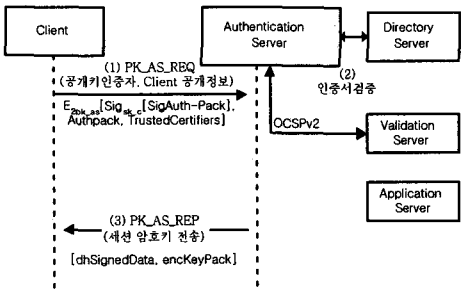
[그림 1]은 사용자가 공개키를 등록하고 인증기관으로부터 인증결과로 인증서를 분배받는 과정을 PKI 모델을 응용하여 구성한 체계도이다. 공개키 생성 및 인증서 발급과정으로 Client의 신분확인이 이루어지면 CA(Certificate Authentication)는 X.500 DN과 인증서의 고유번호(issuer and Serial)를 사용자에게 전송한다. CA는 인증서를 저장하고 디렉토리 서버에 공개한다.

3. 키 복구를 지원하는 인증시스템 설계

인증과 키 분배의 과정에서 관용 암호알고리즘인 DES를 이용하는 시스템은 가입자 수가 증가함에 따라 AS는 많은 비밀 키들을 관리해야 한다. 본 논문에서는 공개키 암호에 의한 인증 및 키 분배 과정에서 AS가 생성하는 D-H의 키 생성(D_k)과 세션 암호키(rk_c)를 동시에 채택함으로써 해결할 수 있다. 그러나 서비스 유형이나 서비스 세션마다 상이한 비밀키를 사용하기 때문에 키를 유실하는 상황이 발생하면 중요한 데이터에 대한 접근이 불가능하다는 위험이 있다. 이러한 문제점을 해결하기 위하여 키 복구 기술을 이용하여 Client와 Server간 비밀통신을 보장하면서 키 분실과 같은 상황이 발생할 경우 세션 암호 키를 복구할 수 있는 인증시스템을 제안한다.

3.1 공개키 암호 인증 및 키 분배

공개키 암호로 메시지에 암호데이터와 공개키 서명을 이용하여 사용자 인증을 함으로써 인증기관에 안전하게 접근할 수 있다. [그림 2]는 동일 Domain 내의 End Entity들에 대한 공개키 인증 서비스와 일정한 세션동안 유지되는 암호 세션 키 교환과정을 도시하였다.



[그림 2] 사용자 인증과 키 분배

공개키 암호에 의한 인증과 키 분배는 PK_AS-REQ(요청)와 PK_AS-REP(응답)을 통해서 이루어진다. 절차는 Client가 서명된 공개키와 공개키 인증정보를 전송하여 AS로부터의 인증과 함께 세션 키를 D-H의 공개 값이나 비밀 세션 키에 대해 선택적으로 요구한다. AS는 수신한 메시지를 인증자의 공개키 정보와 디렉토리서버의 인증서 정보를 검증하고 D-H 공개정보와 세션 키를 응답한다.

가. 시스템 계수 정의

- # : 개체,
- C : Client, AS : 인증서버, S : 응용서버
- $ID_{\#}$: 개체 #의 식별자
- $Realm_{\#}$: 서버 #의 영역
- sk_# : # 해당개체의 비밀키
- pk_# : # 해당개체의 공개키
- rk_# : AS가 생성한 # 해당개체의 세션 키
- D_k : D-H에 의해 생성된 AS와 Client간 세션 키
- ss_k : D-H에 의해 생성된 S와 Client간 세션 키
- A_I : 알고리즘 식별자
- E_1[] : 대칭키 암호(DES)알고리즘
- E_2[] : 공개키 암호(RSA)알고리즘
- Sig[] : 공개키 서명(SHA1)알고리즘
- h[] : 단 방향 해쉬함수
- xi_# : # 개체가 선택한 비밀키(비밀정보로 정수)
- /* */ : 주석

나. Client Request : PK_AS_REQ

인증요구에서 Client의 공개키와 암호알고리즘을 서명하고 공개키 인증정보를 AS의 공개키를 이용하여 전송한다. 초기인증은 Client의 공개키를 인증하고 AS로부터 세션 키를 교환하는 절차이다. Client는 [그림 2]의 (1)과 같이 AS의 공개키를 이용하여 PK_AS_REQ 메시지를 전송한다.

- **SigAuth-Pack** : [ID_c, A_I, pk_c] /*Client의 공개키와 알고리즘 식별자로 공개키 생성과 정확성을 인증하는 요소*/
- **Authpack** : [pkAuthenticator, clientPublicValue] /*공개키 인증자(pkAuthenticator)와 Diffie-Hellman 암호를 사용하기 위한 파라미터 값(clientPublicValue)들을 가진다.*/

-pkAuthenticator : [realm, cusec, ctime, nonce, pachecksum] /*인증자의 공개키 정보(cusec : Client의 Time, ctime : KerberosTime, nonce : D-H사용여부, pachecksum : Checksum(암호알고리즘)으로 sha1 또는 rsa-md5) */

-clientPublicValue:[algorithm, subjectPublicKey]

algorithm : [algorithm, parameters]

/*OID와 prime(p), base(g), length]*/

subjectPublicKey /*소수 p의 원시근 g에 대한 public exponent (g^{xi_c}) mod p*/

- **TrustedCertifiers** : [caname, issuerandserial]

/*AS가 보증하고 신뢰하는 인증자료로 caname(X.509에 의해 정의된 X.500 full name)과 issuerandserial(Client를 신뢰할 수 있는 이미 부여된 AS(KDC) 발행의 인증서 일련번호) */

다. AS Validation : PK_AS_REQ

인증 메시지(PK_AS_REQ)를 수신한 AS는 사용자의 인증정보를 검증하기 위해 비밀키의 생성여부, 공개키 인증자 정보, 자신이 발급한 인증서 일련번호를 검증서버(Validation Server)를 통해 확인(2)한다. 검증내용은 CA의 인증서 서명여부, SigAuth-Pack 검증 실패, 인증서 유효기간, 인증서의 취소여

부, 인증서의 Client이름, Client의 서명검증, AS와 Client의 time 등이다. Client의 subjectPublicKey로 D-H 세션키 $D_k = (g^{xi\_c})^{xi\_as\_}$ mod p를 계산한다.

라. AS Response : PK_AS_REP

AS는 Client에서 제공한 공개키 인증자의 검증 결과로 자신과 Client가 공유할 세션 키와 D-H 정보를 생성하여 Client의 공개키로 응답(3)한다.

· **dhSignedData** : [subjectPublicKey, nonce, dhKeyExpiration] /*D-H암호방식을 사용할 경우 AS에서 인증한 D-H 파라미터를 제공한다.*/

- subjectPublicKey /* $g^{xi\_as\_}$ mod p */
- nonce /* AS가 D-H 값을 사용할 때 "0"로 set */
- dhKeyExpiration /* D-H 키의 유효시간 */

· **encKeyPack** : E_{2Dk}[rk_c, nonce, E_{1}rk_c[Expiration, Trans-number]]

/*Client의 공개키를 이용하여 세션 키를 암호화하며 dhSignedData를 다시 세션 키로 암호화한 enveloped data이다.*/

- nonce /* D-H사용여부 */
- Expiration /* 세션 암호키의 유효시간 */
- Trans-number /* AS의 정책에 따라 세션 암호키를 사용할 수 있는 처리회수를 지정한다. */

AS는 세션 암호 키(rk_c), PKAuthenticator에서 사용된 값으로 요청에 대한 응답용 값(nonce), 임시 세션 암호 키의 만료시간을 나타내는 Expiration과 키의 사용횟수로 AS의 정책에 따라 결정하는 Transaction number를 세션 키로 암호화하고 Client의 공개키로 암호화한 encKeyPack을 전송함으로써 Client의 인증과 함께 암호 세션 키를 보유하게 된다. 이 암호 세션 키는 만료시간이 경과되면 새로운 세션 키를 AS로부터 발급 받기 때문에 주기적으로 갱신이 이루어진다. 이와 같이 3.1절의 과정에서 AS가 데이터베이스에 보유한 개체들의 정보는 [표 1]과 같다.

Entity	인증서	서명 정보	D-H 값
User1	$Cert_1$	$Sig_{SK\_KDC\_I}$[$User_1$, pk_User_1]	g^{xi_c1} mod p
User2	$Cert_2$	$Sig_{SK\_KDC\_I}$[$User_2$, pk_User_2]	g^{xi_c2} mod p
:	:	:	

[표 1] AS의 데이터베이스 정보

3.2 키 복구지원 메커니즘

D-H 키 교환은 이산대수 문제의 어려움을 이용한 공개키 암호방식의 개념을 도입하여 대칭키를 사용하는 두 사용자간에 안전하게 공유키를 분배하는 방식으로 참여하는 구성 개체는 다음과 같다.

· AS : 전체적인 공개요소(p, g)를 생성하며 Client의 키 복구 요청에 따라 해당 세션 키를 복구하는 기능을 가진다.

· Client : 세션 키를 분실할 경우 AS에게 키 복구 요청을 할 수 있다.

· Server : Client에게 서비스를 제공하는 개체로 키 교환을 수행한다.

가. 초기설정

· AS는 전체 공개 값으로 p와 g를 중앙 디렉토리에 저장하고 있다. 공개키 인증과정(PK_AS_REQ)에서 영역에 등록하는 모든 개체가 생성하여 전송한 subjectPublicKey 정보를 활용한다. 각 개체들은 키 복구를 위한 D-H 공개키(값)을 [표 1]과 같이 보유하고 있으며 AS는 모든 개체의 D-H 공개키를 등록받아 보유하고 있다.

나. 키 설정 단계

· 각 개체는 비밀키를 이용하여 D-H 공개정보 값을 계산한다.

· p는 Galois field 상에서 정의된 소수이다. p={1, 2, ..., p-1}

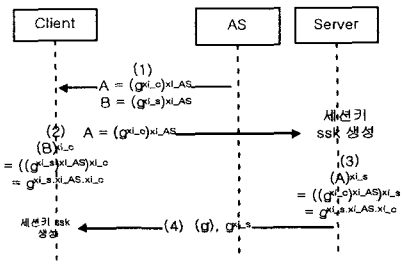
· g는 차수가 g인 곱셈군의 생성원을 나타낸다.

① AS는 공개키 초기인증과정에서 Client와 Server로부터 전송 받은 subjectPublicKey에 자신의 비밀정보(xi_AS)을 지수승하여 A와 B를 생성하고 결과를 Client에게 전송한다.

- $A = (g^{xi\_c})^{xi\_as}$ mod p
- $B = (g^{xi\_s})^{xi\_as}$ mod p

② Client는 B에 자신의 비밀키를 지수승하여 구한 B를 보관하고 값 A를 Server에 전송한다.

- $B = (B)^{xi\_c} = (g^{xi\_s})^{xi\_as})^{xi\_c}$ mod p



[그림 3] 키 설정 단계

③ Server 또한 A에 자신의 비밀키를 지수승하여 A를 구하고 랜덤 수를 구하여 생성원 g에 지수승한다.

- $A = (A)^{xi\_s} = (g^{xi\_c})^{xi\_as})^{xi\_s}$ mod p
- random number : r 생성

· $g^r \bmod p \quad (r \in R, Z)$

- 값 A에 g^r을 곱하여 세션 키 ssk 생성한다.

· $ssk = g^{xi\_s*xi\_as*xi\_c} * (g^r \bmod p) \bmod p$

$= g^{xi\_s*xi\_as*xi\_c+r} \bmod p$

④ Server는 공개정보(g^{xi_s})와 생성원 g^r을 Client로 전송한다.

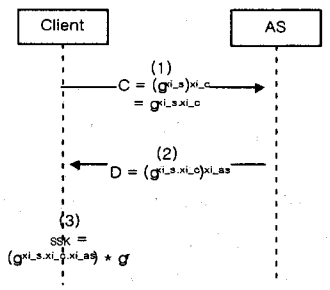
⑤ Client는 보관중인 B값에 생성원 g^r을 곱하여 세션 키를 계산한다.

- $ssk = (g^{xi\_s})^{xi\_as})^{xi\_c} * g^r \bmod p$

$= g^{xi\_s*xi\_as*xi\_c+r} \bmod p$

다. 키 복구 단계

· 사용자는 데이터를 암호화한 키가 유실되었을 때 복구필드와 대응되는 인증정보와 함께 AS에게 전송한다. Client의 키 복구는 AS의 비밀키가 있어야 해결할 수 있다. 키를 복구해야 하는 상황이 발생할 경우 Client의 요청에 의해 이루어진다.



[그림 4] 키 복구 단계

① Client는 Server로부터 전송 받은 g^{xi_s} 값에 자신의 비밀키를 지수승하여 AS로 전송한다.

- $C = (g^{xi\_s})^{xi\_c} \bmod p$

② AS는 전송된 값 C에 자신의 비밀키로 지수승한 D를 Client로 전송한다.

- $D = ((g^{xi\_s})^{xi\_c} \bmod p)^{xi\_as} \bmod p$

③ Client는 D에 Server로부터 전송된 g^r을 곱하여 ssk를 계산한다.

- $ssk = (((g^{xi\_s})^{xi\_c} \bmod p)^{xi\_as} \bmod p) * g^r \bmod p$

$= g^{xi\_s*xi\_as*xi\_c+r} \bmod p$

3.3 비교 분석

[표 2]는 기존의 인증시스템과 비교 분석하였다. Yaksha는 kerberos의 제한점을 보완한 새로운 공개키 개념의 인증시스템이다. 제안 논문에서는 공개키 시스템과 디렉토리서버, 검증서버를 운용함으로써 인증서 경로구축과 검증을 통해 다른 도메인에 대한 영역인증이 가능하고 새로운 멤버 가입 시 확장성이 용이하게 되었다.

구분	Kerberos	Yaksha	제안시스템
안전성	사전공격 약함	사전공격 견딤	사전공격 강함
서명	미 제공	제공	제공
세션키공유	조건적 가능	가능	가능
영역인증	조건적 가능	불가능	가능
확장성	제한적	인증기관 필요	용이
키 분배	비보호 채널	보호채널	비 보호채널
키 복구	기능 없슴	가능	가능

[표 2] 비교 분석표

4. 결론

본 논문에서는 공개키 기반구조의 인증서를 사용하는 암호시스템에 적용할 수 있도록 키 복구를 지원하는 PKI 연동 인증시스템을 제안하였다. 암호 알고리즘으로 관용키 암호방식의 DES와 공개키 암호방식의 RSA, 디지털 서명 알고리즘으로 SHA1을 적용하여 설계되었으며 초기 개체들의 인증과정에서 D-H 공개정보는 세션 키 생성과 키 복구를 위한 메커니즘을 설계하는데 활용된다. 기존의 인증방식과 비교하여 Dictionary Attack에 강하고 암호 키에 대한 정책을 설정하여 키의 비도를 높였다. 사용자의 인증정보는 사용자의 공개키를 서명한 메시지와 인증서를 비교함으로써 제3자는 정당한 사용자로 사칭할 수 없다. 향후 연구과제로 서로 다른 다중영역(multi-domain)간의 효율적인 인증메커니즘의 설계를 필요로 한다.

참고문헌

[1] B. Tung, C. Neuman, M. Hur, A. Medvinsky, S. Medvinsky, J. Wray, J. Trostle. "Public Key Cryptography for Initial Authentication in Kerberos". draft-ietf-cat-kerberos-pk-init-13.txt

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

[3] IETF Draft, "Internet X.509 Public Key Infrastructure Certificate and CRL profile," 1998

[4] Kwangcheul Shin, Jinwook Jung, Ilyong Chung, "An Efficent Kerberos Authentication Mechanism Associated with Drectory System ", proceedings of the international conference on security and management, SAM02, p369-375. June, 2002

[5] ITU-T(formerly CCITT) Information technology - Open Systems Interconnection - The Directory: Authentication Frame- work Recommendation X.509 ISO/IEC 9594-8.