RPC(Remote Procedure Call)에서 DES 인증을 이용한 정보보안 메카니즘의 설계

유성진, 김성진, 김성열, 정일용
조선대학교 전자계산학과

A Design of Information Security Mechanism Using DES Authenticationon on the RPC(Remote Procedure Call)

Seongjin Yoo, Seongjin Kim, Seongyeol Kim, Ilyong Chung
Dept. of Computer Science, Chosun University

요 약

분산환경에서 널리 사용되는 어플리케이션 개발도구인 RPC(Remote Procedure Call)는 정보보안을 위해서 DES 인증만을 가지고 있어 안전한 데이터 전송을 보장하지 못하고 있다. 본 논문에서는 RPC에서 제공하는 키관리 방법을 사용하여 인증, 메시지의 비밀성, 무결성과 송수신 부인봉쇄 서비스를 제공할 수 있는 개선된 RPC 기반의 정보 보안 메카니즘을 설계하고 제안된 방법의 안전도를 검증하였다.

I. 서 론

정보처리와 통신기술이 결합된 정보통신망은 공간적, 시간적 제약을 뛰어넘어 범세계적인 규모의 네트워크화로 진전되어 가고 있다. 이러한 통신망을 이용하여 전자우편 및 CALS/EC 등 기업간의 거래뿐만 아니라 가정들에게 새로운 통신 서비스가 제공되고 있다. 그러나 사회가 고도로 정보화가 될수록 개인의 프라이버시를 비롯하여, 전산망의 불법적인 해킹 등 많은 역기능적인 문제가 대두되어 이를 방지하기 위한 정보보호[1]-[4] 메카니즘 구현이 부각되고 있다.

이러한 문제점들을 해결하기 위해 여러 가지 정보보호 서비스들이 제공되어야 하며, 본 연구에서는 분산환경을 기반으로 하고 있어 네트워크 어플리케이션을 개발하는 도구가 필요하다. 사용되고 있는 개발도구는 BSD 4.1에서 소개된 소켓[5]으로서 통신 프로토콜을 지원하는 API(Application Programmer's Interface)를 만들어 클라이언트/서버간의 분산 프로그래밍을 지원한다. TLI(Transport Layer Interface)[6]는 시스템 V3.0에서 소개하고 있으며 OSI 참조모델의 Transport 계층과의 접속을 가능하게 하는 API로서 신뢰성 있는 데이터 전송을 가능하게 하고 순서적으로 데이터를 전송한다.

RPC(Remote Procedure Call)[7]-[8]는 클라이언트에게 원격 시스템에 대한 요구를 투명하게 제공하는 방법으로 현재 널리 사용되고 있고 보유하고 있는 보안 메카니즘으로는 UNIX 인증과 DES 인증이 있다. UNIX 인증은 DAC(Discretionary Access Control)[9]을 적용한 것이며, DES 인증은 키관리

메카니즘은 Diffie-Hellman 방식[10]을, 암호화 알고리즘인 DES[11]를 적용하여 인증 서비스를 구현하고 있다.

본 논문에서는 RPC 기반의 향상된 정보 보안 메카니즘을 설계하여 위해서 클라이언트/서버 환경에서 사용되는 분산 어플리케이션 제작방법과 RPC에서 제공하는 정보보안 메카니즘을 고찰하고, ISO를 비롯한 국제표준화 단체에서 제시하고 있는 인증 메카니즘[12]-[13]을 비롯한 다양한 정보보안 서비스를 분석한다. 제안된 메카니즘은 합법적인 송신자임을 증명하는 인증 서비스를 비롯하여 데이터의 비밀성, 무결성, 부인봉쇄 등 다양한 서비스들을 제공한다.

II. 안전한 전송을 위한 RPC 기반의 정보보안 메카니즘

RPC를 기반으로 하는 대부분의 어플리케이션들은 서버로부터 인증을 받지 않는 클라이언트의 요청을 막기 위해 몇 단계의 제어를 필요로 하는데 이때 전송 프로토콜이 서버를 사용하고자 하는 사용자에 대해 통합적인 제한을 하지 않기 때문에, 일반적으로 클라이언트로부터의 모든 요청은 아무런 제한 없이 서버에 전달된다. 따라서 사용자가 서버에 접근하는 것을 통제하기 위한 기능을 프로그래머가 구현해야 되는데, RPC 라이브러리는 서버로부터 인증받은 특별한 사용자나 호스트 시스템만이 RPC 호출 요청을 할 수 있도록 인증 서비스 기능을 제공한다.

인증 프로시져에 사용되는 정보로는 두 가지가 있는데, 하나는 사용자의 신원을 파악하는 증명서(credential)이고, 다른 하나는 증명서가 옳다는 것을 검증하기 위한 검증자(verifier)이다. RPC 라이브러리가 제공하는 인증서비스는 세 가지 종류가 있는데, No Authentication, UNIX 인증과 DES 인증이다.

No Authentication은 인증 구조만 설정하고 실제적으로 클라이언트와 서버사이에 인증에 관련된 정보들은 전달하지 않는다. UNIX 인증은 클라이언트가 자신의 credential을 서버에 전송함으로써 인증이 이루어지지만, 이 credential이 암호화되지 않기 때문에 중간에 위조되었는지 또는 클라이언트가 자신의 credential을 위조하여 보냈는지를 서버가 검증할 수 없다. 하지만 DES 인증은 UNIX 인증과는 달리 클라이언트의 credential을 검증할 수는 있고 timestamp를 이용하여 암호화하기 때문에 검증도 가능하다. 시스템에서 제공하는 DES 인증도 다른 인증방법들과 같이 클라이언트와 서버사이의 전송되는 메시지는 단순히 XDR로 변환만 될 뿐 메시지의 부인봉쇄, 비밀성과 무결성이 보장되지 않는다.

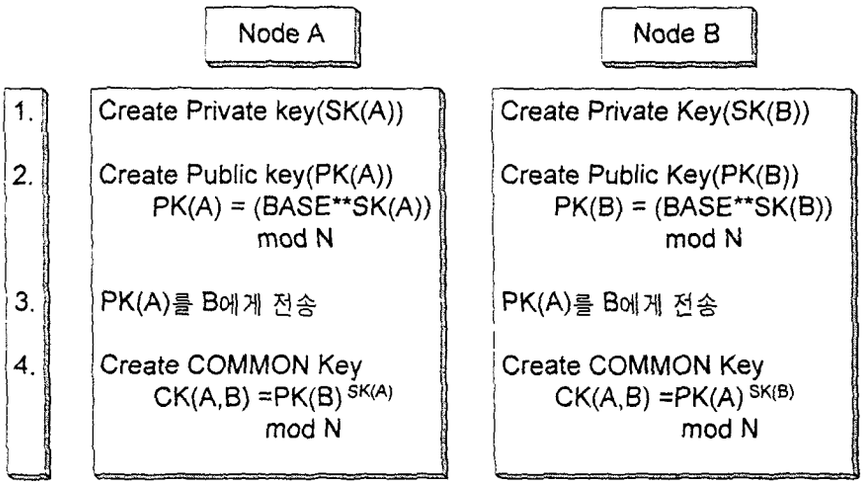
따라서 본 논문에서는 이런 문제점들을 해결하기 위해, 인증 방법은 앞에서 제시된 인증 방법 중에서 가장 우수한 DES 인증을 이용하고, 부인봉쇄 서비스와 전송되는 데이터들의 비밀성, 무결성을 보장할 수 있도록 해쉬함수를 도입한 향상된 RPC 정보보안 메카니즘을 설계한다.

II.1 RPC 기반의 정보보안 메커니즘의 설계

본 논문에서 설계하고자 하는 정보보안 메카니즘은 사용자가 정당한 사용자인가를 확인하기 위한 인증 메커니즘, 메시지의 내용이 외부에 노출되는 것을 막기 위한 메시지의 비밀성 실현, 보관중인 데이터나 전송중인 데이터가 중간에 변조되는 것을 막기 위한 메시지의 무결성 실현, 그리고 통신하는 두 사용자간의 부인봉쇄 서비스를 제공하고자 한다.

가. 인증 메커니즘의 설계

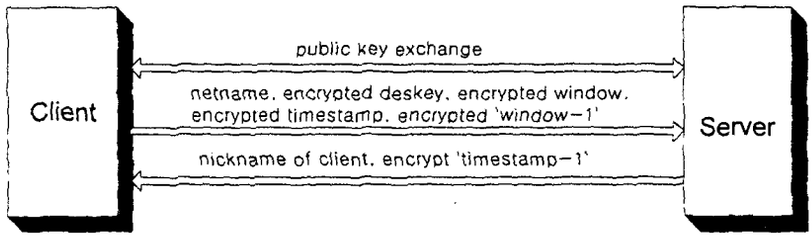
세션이 시작되기 전에 클라이언트와 서버는 네트워크 관리자에게 부여받은 자신의 공개키를 서로 교환한다. 클라이언트는 자신의 Credential과 이것을 확인하기 위한 Verifier를 생성하고Credential은 클라이언트의 netname과 암호화된 deskey를 포함하고 있다.



[그림 1] Diffie-Hellman 공개키 분배 알고리즘

[그림 1]에서 deskey를 암호화하기 위한 Common key를 생성하기 위해 Diffie-Hellman 알고리즘을 사용하는데, 이 알고리즘은 처음에 상호 교환했던 공개키를 이용한다. Verifier는 암호화된 타임스탬프를 사용하며 타임스탬프를 암호화하기 위해 DES 알고리즘을 적용하고 이때 사용되는 암호 키는 Credential에 포함된 deskey이다. 클라이언트는 자신이 생성한 Credential과 Verifier를 서버에 전송한다.

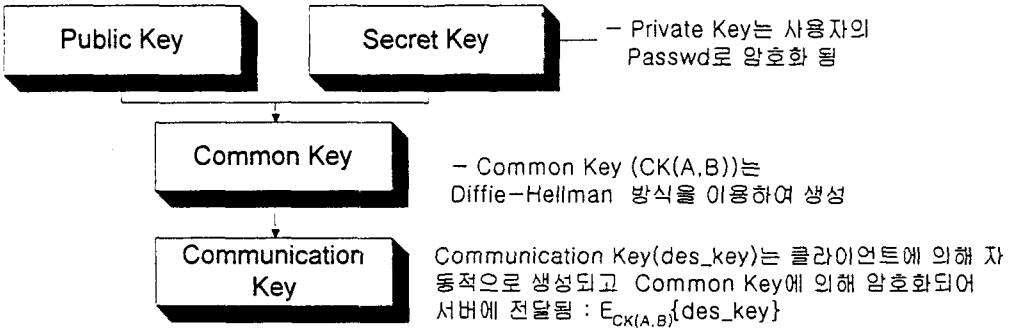
서버는 먼저 처음에 전달된 공개키를 가지고 Diffie-Hellman 알고리즘을 이용하여 Common key를 구하여 클라이언트의 Credential에 포함된 deskey를 복호화하고, 이 deskey를 이용하여 타임스탬프를 복호화한다. 그래서 이것이 자신의 시간보다 작으면 클라이언트의 Credential이 유효하다는 것을 확인하고 클라이언트의 netname으로부터 UID, GID등 클라이언트를 인증하기 위한 정보를 얻는다. 다음으로 클라이언트의 인증이 끝나면 서버는 이후 세션에서 사용할 클라이언트의 nickname과 'timestamp-1'을 암호화하여 클라이언트에 전송한다. 클라이언트는 이후 세션에서 자신의 netname 대신에 nickname을 사용하여 서버와 통신한다.



[그림 2] 클라이언트와 서버의 상호 인증 과정

나. 메시지의 비밀성

클라이언트와 서버간의 인증과정이 끝나면 메시지가 전송되는데, 메시지를 외부에 노출시키지 않기 위해 암호화과정을 거치게 된다. 메시지를 암호화하기 위해 사용되는 알고리즘은 DES 알고리즘을 사용하고, 통신키는 클라이언트가 생성하여 서버에 전달하게 된다. 통신키를 생성하고 암호화되는 과정은 [그림 3]와 같다.



[그림 3] 통신키의 생성 및 암호화

다. 메시지의 무결성

MD5는 다양한 길이의 메시지를 입력으로 받아 128-bit의 해쉬값을 출력한다. 두 개의 서로 다른 메시지를 입력으로 하여 같은 해쉬값을 취할 수 없다. 해쉬 함수는 Append Padding bit, Append length, Initialize MD buffer, Process Message in 16-word blocks와 같은 4가지 단계를 수행하여 해쉬값을 계산하여 수신 측에서 검증하여 메시지의 무결성을 확인한다.

라. 부인 봉쇄정보

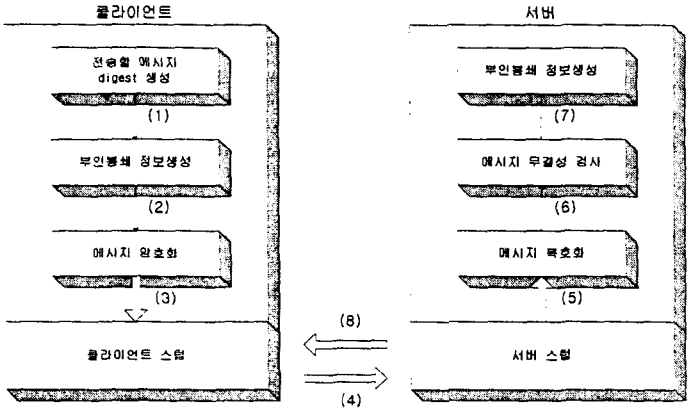
부인봉쇄에는 두 가지로 분류할 수 있는데 발신자가 보낸 데이터를 부인할 수 없도록 하는 발신 부인봉쇄와 수신자가 데이터의 수신을 부인할 수 없도록 하는 수신 부인봉쇄가 있다.

부인봉쇄 정보는 메시지의 Digest, UID, IP, Address로 구성된다. 클라이언트와 서버사이의 부인봉쇄 서비스를 보장하기 위해 각각 부인봉쇄정보를 생성해서 서로에게 전송해야 한다. 각 개체는 전송받은 부인봉쇄 정보를 생성할 수 없어야 한다. 이렇게 하기 위해 클라이언트는 생성한 메시지의 다이제스트와 자신의 네트워크 이름을 자신의 비밀키로 암호화하여 서버에 전송하고, 서버도 역시 메시지 다이제스트와 자신의 네트워크 이름을 자신의 비밀키로 암호화하여 클라이언트에게 전송한다.

- 송신 부인봉쇄 : $\{Digest, UID_A, UID_B, IP_{ADDR_A}, IP_{ADDR_B}\}SK_A$
- 수신 부인봉쇄 : $\{Digest, UID_B, UID_A, IP_{ADDR_B}, IP_{ADDR_A}\}SK_B$

마. 알고리즘

[그림 4]은 본 논문에서 제안하는 개선된 RPC 정보보호 메커니즘을 나타낸다. 클라이언트와 서버간의 상호인증과정은 [그림 2]에서 기술하였으므로 여기에서는 클라이언트와 서버간의 메시지 전송시 메시지의 비밀성, 무결성과 부인봉쇄 서비스의 실현과정을 설명하고자 한다.



[그림 4] RPC 기반 정보보안 메카니즘 구성도

먼저 해쉬함수 MD5를 이용하여 메시지의 다이제스트를 구하는데 이 정보는 메시지의 무결성을 검사하기 위해 사용된다. 다음으로 부인봉쇄정보를 생성하는데 이것은 사용자의 uid, 호스트이름, IP 주소로 이루어져 있다. 이 과정이 끝나면 전송하고자하는 메시지와 부인봉쇄정보를 암호화하는데 메시지를 암호화할 때 사용되는 암호화키는 앞에서 생성한 deskey를 사용하고 부인봉쇄정보를 암호화할 때는 자신의 비밀키로 암호화한다. 그리고 암호화된 메시지와 부인 봉쇄정보를 서버에게 전송한다. 서버는 전송받은 메시지를 복호화한 다음 클라이언트와 같은 해쉬 함수를 이용하여 메시지 다이제스트를 구하여 클라이언트로부터 전송받은 다이제스트와 같은가를 검사한다. 그리고 자신의 부인봉쇄정보를 생성하여 클라이언트에게 전송한다.

SRPC_Algorithm

1. C : $Digest_C$, $\{Repu-Info_c\}SK_C$, $\{Message\}deskey$ 생성

   /* 메시지의 Digest를 생성하여 메시지의 무결성을 검사한다. */

   · $Digest_C$: (Message Content)MD5

   /* 송신부인봉쇄 정보 : 메시지 송신에 대한 송신자의 부인 방지 */

   /* 클라이언트의 송신 부인봉쇄정보를 자신의 비밀키로 암호화*/

   · $\{Repu-Info_c\}SK_C$

   Where Repu-Info$_c$ is $Digest_C$|| UID_C|| UID_S|| IP_{ADDR_C}|| IP_{ADDR_S}

   · Message : Message Header || Message Content

   Where Message Header is "$Digest_C$||Cur_Time|| IP_{ADDR_C}|| UID_C"

2. $C \Rightarrow S$: $\{Repu-Info_c\}SK_C$, $\{Message\}deskey$

3. S : Message 복호화, $Digest_S$ 생성 및 검증, $\{Repu-Info_s\}SK_S$ 생성

· Message 복호화 : $\{\{Message\}deskey\}deskey$ =
Message Header || Message Content

· $Digest_S$ 생성 : $Digest_S$ = (Message Content)MD5

· $Digest_S$ 검증 : $Digest_S == Digest_C$

/* 수신부인봉쇄 정보: 메시지의 수신에 대한 수신자의 부인 방지 */

/* 수신부인봉쇄 정보를 서버의 비밀키로 암호화 */

· $\{Repu-Info\}SK_S$

Where Repu-Info$_s$ is " $Digest_S$|| UID_S|| UID_C|| IP_{ADDR_S}|| IP_{ADDR_C}"

/* 서버의 암호화된 수신부인봉쇄 정보를 클라이언트에게 전송 */

4. $S \Rightarrow C$: $\{Repu-Info_s\}SK_S$

II.2 *SRPC_Algorithm*의 분석

RPC에서는 정보보호 서비스 중 인증 서비스만을 제공하고 있지만 인증서비스만으로는 메시지를 안전하게 전송할 수 없다. 그러므로 본 논문에서는 기존의 RPC에서 제공하는 인증서비스에 다른 정보보호서비스를 구현하였는데; 이것들은 메시지의 무결성, 비밀성과 송수신 부인봉쇄 서비스이다.

단계 2에서 클라이언트는 서버에게 {Repu-Info$_c$}SK$_c$와 {Message}deskey를 전송하는데 {Repu-Info$_c$}는 클라이언트의 비밀키로 암호화되어 클라이언트와 Trusted Thirty Party가 아니면 복호화할 수 없고, deskey는 인증과정을 통하여 송수신자만이 가지고 있으므로 {Message}deskey는 안전하게 보내져서 비밀성을 보장할 수 있다.

단계 3에서 서버는 수신한 메시지({Message}deskey)를 인증과정중에서 얻은deskey로 복호화하여 Message Content를 얻고 이를 서버의 해쉬 함수로 계산한 값과 비교하여 동일하면 메시지 무결성을 만족한다. 만일 동일하지 않으면 메시지가 변조되었으므로 새로운 전송을 요청한다.

단계 4에서 수신부인봉쇄 정보는 수신자의 비밀키로 암호화되어 서버와 Trusted Thirty Party가 아니면 복호화할 수 없어 부인봉쇄 기능을 수행한다.

그러므로 제안된 알고리즘을 적용하여 메시지의 비밀성, 무결성, 부인봉쇄 서비스를 제공받아 메시지를 안전하게 전송할 수 있다.

III. 결론

본 논문은 RPC를 기반으로 클라이언트의 인증 및 클라이언트와 서버사이의 전송하고자하는 메시지의 비밀성 및 무결성 그리고 부인봉쇄를 보장할 수 있는 개선된 RPC 기반의 정보 보안 메카니즘을 설계하였다. 제시된 정보보안 메카니즘의 장점은 기존의 DES인증을 이용하여 클라이언트를 인증하고, 또한 메시지의 Digest, 클라이언트의 UID, IP 주소로 이루어진 부인봉쇄 정보를 생성한다. 그리고 이 부인봉쇄정보를 자신의 비밀키로 암호화하여 전송함으로서 이 부인봉쇄정보는 자신과 Trusted Third Party외에는 생성할 수 없으므로 안전한 부인봉쇄 서비스를 구현하였다. 해쉬함수인 MD5를 이

용하여 메시지의 Digest를 생성함으로서 전송되는 데이터의 무결성을 보장할 수 있으며 클라이언트와 서버간의 전송되는 메시지를 송수신자만이 인식할 수 있는 키로 암호화함으로서 비밀성을 보장할 수 있다.

본 연구에서는 정보보안 서비스 중에서 비밀성 및 무결성 그리고 부인봉쇄 서비스를 설계하였다. 향후에는 향상된 정보보안을 위해 다중서명 및 Access Control 기법 등을 적용하여 다양한 서비스를 구현하고자 한다.

[참고문헌]

[1] 컴퓨터.네트워크 보안기술 연구, 한국전자통신 연구소, 1991.2

[2] 최양희, "전산망 정보보안기술 및 동향", '96 정보보안심포지움, pp. 93-120, 1996. 7.

[3] 강명호,송주석, "전자상거래 관련기술", 통신정보보호학회지, 제7권, 제3호, pp. 5-22, 1997. 9.

[4] 김기현,은유진,이인수,이홍섭, "정보보호 기술 분류", 통신정보보호학회지, 제8권, 제1호, pp. 5-20, 1998. 3.

[5] SunOS5.3 Network Interface Programmer's Guide, Sun Microsystems, Inc., pp. 7-130, 1993

[6] Comer & Stevens, Internetworking With TCP/IP Vol III, Prentice Hall, New York, pp. 221-303, 1993

[7] RPC: Remote Procedure Call Protocol Specification version 2, IETF RFC 1057, 1988

[8] W. Stevens, Advanced Programming in the UNIX Environment, Addison-Wesley, New York, 1992

[9] B, Schneier, Applied Cryptography, 2nd Ed., John Wiley & Sons, New York, 1996

[10] Diffie & Hellman, "New directions in Cryptography", IEEE Trans. Info. Theory, IT-22, vol. 22, no. 6, pp. 644-654, Nov. 1976

[11] National Bureau of Standards, "Data Encryption Standard", Federal Information Processing Standards Publication 46, Jan. 1977.

[12] FIPS PUB 140-1, Security Requirements for Cryptographic Modules, Federal Information Processing Standards Publication 140-1, 1993

[13] ISO/IEC 7498-2, Information Technology, OSI Reference Model Part 2: Security Architecture