韓國 OA學會 論文誌
第3卷 第4號, 1998. 12.

98-3-4-3-3

Secure 클라이언트-서버 시스템 설계에 관한 연구

이 상 렬*

A Study on the Design of a Secure Client-Server System

Sang Ryul Lee*

요 약

본 논문에서는 일반적으로 많이 사용되고 있는 클라이언트-서브 시스템에 암호화 기법을 도입함으로써 클라이언트와 서버간에 교환되는 정보가 타인에게 누설되는 것을 방지할 수 있는 Secure 클라이언트-서버 시스템을 설계하였다. 초기의 로그온 단계에서는 비대칭 암호화 방식을 도입하여 상호간 사용자 확인을 가능하게 하였으며 일반 정보 교환 단계에서는 대칭 암호화 방식을 도입하여 암호화에 소요되는 시간을 최소화 시켰다. 또한 이러한 Secure 클라이언트-서버 시스템에서도 디지털 서명이 가능함을 보였으며 시스템에서 사용되는 암호키의 생성 및 배포를 안전하게 할 수 있는 효율적인 관리 방안을 제시하였다.

Abstract

In this paper we designed a secure client-server system to be able to protect messages between client and server using cryptography. We authenticated each other using a asymmetric encryption argorithm on the logon procedure and minimized the time to encrypt and decrypt messages using a symmetric encryption argorithm on exchanging messages. We proved that it is possible to make a digital signature on our secure client-server system. And we suggested the efficient key management method to generate and distribute cryptograpic key securely.

* 상지대학교병설전문대학 사무자동화과 전임강사
논문접수 : 98.11.2. 심사완료 : 98.12.5.

Ⅰ. 서 론

최근 PC통신 및 인터넷의 보급 확산으로 일반인들도 컴퓨터 통신을 손쉽게 이용할 수 있게 됨으로써 그 사용자가 급증하고 있다. 또한 컴퓨터 통신의 활용 분야도 다양하게 발전하여 전자우편, 전자상거래, 전자화폐 등으로 점차 확대되어 가고 있다. 이와같은 발전과 함께 통신내용의 도청 및 변조를 시도하려는 해커들이 대거 출현하리라 예상되고 있다. 이에 대응하기 위해서는 통신자의 신분확인 및 통신내용의 보호가 시급하다. 현재와 같이 사용자 ID 및 Password 등으로 컴퓨터 사용자를 제한하는 방법으로는 통신로 중간에서 도청하는 행위를 방지할 수 없다. 따라서 자신의 컴퓨터내에서 통신내용을 아무도 알 수 없는 형태로 암호화하여 통신로로 전송하는 것이 가장 안전한 방법일 것이다.

어떤 시스템이 안전하기 위해서는 두가지 조건이 필요한다. 첫째, 상호간의 신분확인이 우선 되어야 한다. 둘째, 상호간에 교환되는 정보 내용이 타인에게 누설되거나 타인에 의해 새로운 내용이 추가, 일부 내용 삭제 또는 변조되지 않아야 한다. 또한 이와 같은 안전한 시스템에서 발신자가 발신내용에 대해 보증해 줄 수만 있다면 전자결재와 같은 부가 서비스를 제공할 수가 있어 상거래 계약 등 여러 가지 업무를 컴퓨터 통신을 통하여 손쉽게 처리할 수 있게 될 것이다.

본 논문에서는 클라이언트-서버형 네트워크 환경 하에서 이와 같은 요건을 만족해 줄 수 있는 통신 프로토콜을 제안하고자 한다. 2장에서는 본 논문의 연구 대상이 되는 컴퓨터 네트워크의 구성 및 이에 적용할 암호기법에 대해 살펴보며 3장에서부터 5장까지는 안전한 시스템이 되기 위한 요건에 부합되는 프로토콜을 제안하고 6장에서는 암호키를 효율적으로 관리하는 방법을 제안한다.

Ⅱ. 대상 네트워크 및 적용 암호기법

클라언트-서브(Client-Server)형 네트워크는 클라이언트가 서버에 로그온하여 서버와 정보교환을 하는 형태로서 클라이언트들 사이에 정보교환이 별로 필요 없는 곳에서 많이 이용된다. 피어 투 피어(Peer to Peer)형 네트워크에서는 모든 노드가 동등한 자격으로 자신이 원하는 노드와 직접 정보교환을 할 수 있는 형태로서 분산처리 업무가 많이 발생하는 곳에서 이용된다.

본 논문에서는 전자의 클라언트-서브형 네트워크를 연구 대상으로 한다. [그림 1]은 클라언트-서브형 네트워크의 내부 계층 구조를 보여준다. 인터넷[1,2]의 경우 서버의 응용 계층은 HTTP, FTP, SMTP 등의 데몬 소프트웨어가, 클라이언트 응용 계층은 WWW 브라우즈 소프트웨어가 이에 해당된다. 그리고 네트워크 계층은 TCP/IP 통신 프로그램이 이에 해당된다. 본 논문에서 앞으로 제안하는 암호화를 위한 프로토콜은 암호화 계층에서 구현된다. 넷스케이프사에서는 이 계층에 SSL 프로토콜[3]을 자체적으로 정의해 놓았다.

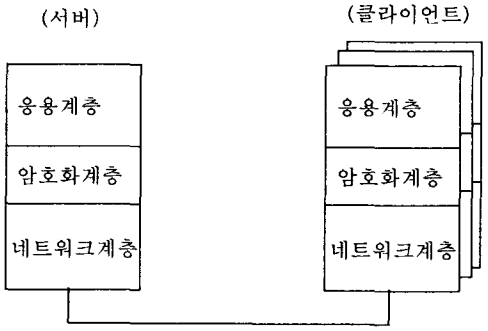

그림 1. 클라언트-서브형 네트워크의 내부 계층 구조

암호기법에는 암호화키와 복호화키가 같은 대칭키 암호기법[4]과 다른 비대칭키 암호기법[5]이 있다. 1977년 미국 표준국에서 미 연방 정보처리표준 46으로 채택된 DES(Data Encryption Standard) [6]가 대칭키 암호기법의 예이고 1977년 MIT의 R. Rivest, A. Shamir, L. Adleman이 개발한 RSA 암호[7]가 비대칭키 암호기법으로 많이 사용되는 예이다. 이들 암호기법은 각기 그 특징이 다르다. 대칭키 암호기법은 속도가 빠르고 키를 생성하기 쉽지만 수신한 내용을 수신자가 임의로 변경할 수 있음으로 발신내용의 유일성이 없다. 따라서 디지털 서명과 같은 부가서비스를 제공할 수가 없다. 반면 비대칭키 암호기법은 암호화 속도는 다소 느리지만 비밀키로 작성된 암호문은 비밀키 소유자만이 유일하게 생성할 수 있음으로 이와 같은 부가서비스를 제공할 수

있다.

본 논문에서는 클라이언트-서버 시스템을 궁극적으로 가장 빠르고 안전하고 부가서비스가 많은 시스템으로 만들기 위해서 전체 통신 내용 중 일부분에만 해당되는 사용자 신분 확인 절차 및 특정 문서에 서명할 경우에는 비대칭 암호기법을 적용하고 대부분의 일반 통신의 경우는 대칭 암호기법을 적용하도록 한다.

III. 사용자 인증

일반적으로 비밀 통신은 먼저 상대방의 신분을 확인하는 사용자 인증 단계 그리고 일반 정보를 교환하는 정보 교환 단계로 나눌 수 있다. 이 중 어느 단계라도 안전하지 못할 경우 비밀 통신은 보장되지 않는다. 처음부터 상대가 원치 않는 자일 경우에는 이후의 모든 통신 내용이 누설되는 치명적인 결과를 초래할 수 있음으로 통신을 할 때 우선적으로 쌍방이 상대의 신분을 확인할 수 있어야 한다. 또한 상대방의 신분을 확인하였더라도 정보 교환 단계에서 그 내용이 도청된다면 비밀통신의 의미가 없어진다. 이 장에서는 통신을 처음 시작할 때 상대방의 신분을 확인할 수 있는 사용자 인증 프로토콜을 제안한다.

클라이언트-서버형 네트워크에서 우선 클라이언트는 자신의 로그온 IDc와 비밀번호 PWDc 그리고 앞으로 정보 교환 단계에서 클라이언트와 서버간의 정보 암호화를 위한 비밀키 CKcs와 자신의 공개키 PKc를 서버의 공개키 PKs를 이용하여 암호화하여 다음과 같이 보낸다.

{IDc, PWDc, CKcs, PKc}PKs ·····(1)

이를 수신한 서버는 자신의 비밀키 SKs로 이를 해독하여 IDc와 PWDc로 클라이언트의 신분을 확인할 수 있다. 그리고 앞으로 정보 교환 단계에서 이용할 비밀키 CKcs와 클라이언트의 공개키 PKc를 알아낸다. 여기서 PKc를 포함시키는 이유는 서버가 모든 클라이언트의 공개키를 보관할 필요가 없고 해당 공개키를 찾는 노력을 줄일 수 있기 때문이다.

다음은 서버가 자신의 신분을 클라이언트에게 알리기 위해 다음과 같은 내용을 클라이언트에게 보낸다.

{IDc, PWDc, CKcs}PKc ·····(2)

이를 수신한 클라이언트는 자신의 비밀키 SKc로 해독하여 이전에 자신이 보낸 내용과 비교하여 일치할 경우 이 암호문을 보낸 자가 서버임을 확인할 수 있다. 그 이유는 (1)번 암호문을 해독할 수 있는 자는 서버 뿐임으로 서버만이 (2)번 암호문을 생성할 수 있기 때문이다.

이로써 서버와 클라이언트는 상호간 신분을 확인할 수 있다. 단, 서버는 자신의 비밀키 SKs를 그리고 클라이언트는 자신의 로그온 IDc, PWDc 그리고 자신의 비밀키 SKc를 잘 보관하고 있어야 한다.

IV. 도청 및 변조 방지

도청이라함은 통신하고 있는 양자 사이에서 몰래 통신 내용을 엿듣기만 하는 것을 말하고 변조라 함은 통신 내용을 변조 또는 삭제하거나 조작된 내용을 첨가하는 좀 더 적극적인 통신 방해 행위를 말한다. 따라서 안전한 시스템이 되려면 타인에 의해 도청되지도 않아야 하며 통신 내용이 변조되지 않음을 보증할 수 있어야 한다.

정보 교환 단계에서는 이미 사용자 인증 단계에서 클라이언트가 생성하여 서버에게 안전하게 전달한 비밀키 CKcs를 이용하여 서버와 클라이언트 사이의 통신 내용 M을 다음과 같이 암호화하여 상호 교환한다.

{M}CKcs ·····(3)

CKcs는 서버와 클라이언트만이 알고 있는 키임이 확실함으로 도청자가 통신 내용을 해독해 낼 수는 없다. 그러나 그 내용은 알아낼 수는 없으나 변조는 가능할 수 있음으로 통신 방해를 할 수가 있다. DES 등 일반적인 암호화 알고리즘은 일정한 크기로 쪼개어 암호화를 함으로 단위 크기로 암호화가 된다. 따라서 중간의 도청자는 암호문의 내용을 크기는 변경하지 않고 그 내용만 일부 변

경하는 방법으로 통신을 방해할 수가 있다. 이럴 경우 수신자는 암호 해독시 문제점을 발견할 수가 없고 다만 통신 내용의 문맥으로 이상 여부를 분별해 낼 수 있다. 그 내용이 너무 이상할 경우에는 다시 그 내용을 확인해 볼 수도 있겠지만 요행히 문맥에 큰 이상이 없을 경우에는 이를 그대로 받아들이는 오류를 범할 수 있다. 이와 같이 이런 방해 행위는 항상 인지되지 못함으로 이에 대한 방어책이 마련되어 있어야 한다.

이를 위해 (3)번의 암호문을 다음과 같이 변경한다.

{M, MAC}CKcs ·····(4)

여기서 MAC(Message Authentication Code)은 RSA에서 발표한 MD5 [8]와 같은 다이제스트 함수를 이용하여 구한 값으로 한다. 즉, MAC=Digest(M,CKcs)

다이제스트 함수의 특징은

- 출력 값으로부터 입력 값을 구할 수 없는 일방향 함수이다.
- 입력 값이 다를 경우 출력 값도 다르다.

네트워크 내의 모든 사용자에게 이 함수를 공개하여 사용할 수 있도록 한다.

수신자는 (4)번 암호문을 해독한 후 M의 MAC 값을 구하여 암호문 속의 MAC 값과 비교하여 같을 경우 이 M은 변조되지 않았음을 확인할 수 있다.

또한 이 방해자는 이전의 암호문을 단위 크기별로 수집해 두었다가 중간 중간에 이를 삽입한다든가 교환되고 있는 암호문을 중간에 단위 크기별로 삭제해 버리는 방법으로 통신을 방해할 수가 있다. 이 또한 수신자는 통신 내용의 문맥이 너무 이상할 경우에는 다시 그 내용을 확인해 볼 수도 있겠지만 요행히 문맥에 큰 이상이 없을 경우에는 이를 그대로 받아들일 것이다. 여시 이에 대한 방어책도 마련되어 있어야 한다. 이를 위해 (4)번의 암호문을 다음과 같이 변경한다.

{M, MAC, SN}CKcs ·····(5)

여기서 SN은 로그온한 이후 순차적으로 증가하는 일련번호로서 동일한 번호가 반복된다든가 특정 번호가 누락될 경우 통신 방해가 이루어지고 있음을 즉시 인지할 수 있다.

V. 디지털 서명

디지털 서명이란 컴퓨터와 같은 전자 정보기기를 이용하여 작성된 내용에 대하여 자신이 작성한 내용임을 확인해 주는 것을 말한다. 이와 같은 디지털 서명이 가능해지려면 작성자인 본인 외 누구도 그 내용을 작성할 수 없음이 증명되어야 한다.

만일 클라이언트가 서버에게 (5)번과 같은 암호문을 보내었다면 이를 받은 서버는 그 암호문을 해독할 수 있을 뿐만 아니라 이를 변조하여 다시 암호화할 수도 있음으로 향후 서버가 클라이언트에게 그 통신 내용에 대해 책임을 전가할 수 없다. 따라서 클라이언트가 서버에게 서명한 M을 보낼 경우에는 다음과 같은 암호문을 만들어 보낸다.

{{M, MAC, SN}SKc}CKcs·····(6)

이를 수신한 서버는 CKcs로 우선 암호문을 해독하고 (1)에서 받은 PKc로 다시 암호문을 해독하여 통신 내용을 알아낼 수 있다. SKc는 클라이언트 이 외는 누구도 알 수 없음으로 {M, MAC, SN}SKc를 발생한 자는 클라이언트가 확실하다. 따라서 이 암호문 속의 통신 내용 M에 대해서 클라이언트가 법적으로까지도 책임질 수 있다. 참고로 대표적인 디지털 서명 표준은 미국의 DSS[9]가 있다.

VI. 암호키 관리

암호 시스템의 안전성은 암호키의 관리에 상당히 많이 좌우됨으로 암호키의 생성 및 배포가 용이하고 안전해야 한다.

본 암호시스템에서는 사용자 인증 단계에서는 비대칭

키(PK, SK)를 사용하고 정보 교환 단계에서는 대칭키(CK)를 사용한다. 따라서 각각의 암호키 관리에 대해 살펴보겠다.

사용자 인증 단계에서 사용되는 클라이언트의 공개키 PK와 비밀키 SK 쌍의 생성 및 배포는 다음과 같이 세 가지 방법으로 이루어 질 수 있다.

방법1 : 서버가 PK와 SK를 생성하여 클라이언트에게 배포하는 경우

〈프로토콜〉

① 서버가 PKc와 SKc 쌍을 생성

② PKc와 SKc를 클라이언트에게 오프라인으로 은밀히 배포

③ 서버내 SKc를 삭제 (서버의 파괴로부터 시스템을 보호하기 위해)

방법2 : 클라이언트가 SK를 결정하고 서버가 이에 해당하는 PK를 생성하여 배포하는 경우

〈프로토콜〉

① 클라이언트가 SKc를 결정

② {SKc}PKs를 서버에게 온라인으로 전송

③ 서버가 SKs를 이용하여 SKc를 해독한 후 이의 PKc를 계산해 내고 이미 사용 중인 PK와의 중복여부 파악 후 클라이언트에게 결과 통보

④ {PKc}SKs를 클라이언터에게 온라인으로 전송. 만일 중복된 결과가 전송되었을 경우 앞 절차 반복

⑤ 클라이언트는 PKs로 PKc를 해독함

방법3 : 클라이언트가 PK와 SK를 생성하여 서버에게 PK를 허락 받는 경우

〈프로토콜〉

① 클라이언트가 PKc와 SKc 쌍을 결정

② {PKc}PKs를 서버에게 온라인으로 전송

③ 서버는 SKs를 이용하여 PKc를 해독해 내고 이미 사용 중인 PK와의 중복여부 파악 후 클라이언트에게 결과 통보

④ 중복되지 않을 때까지 앞 절차 반복

어느 경우에도 서버에는 모든 클라이언트의 ID, PWD 그리고 PK를 기억하고 있어야 한다. 그 이유는 방법1의 경우에는 이미 사용 중인 PK와 중복되지 않은 암호키를 생성하고자 하기 때문이며 방법2의 경우에는 클라이언트로부터 받은 SK로부터 생성한 PK를 그리고 방법3에서는 클라이언트로부터 받은 PK를 이미 사용 중인 PK와 중복여부를 파악한 후 그 결과를 클라이언트에게 통보해 주고자 하기 때문이다.

여기서 방법1은 가장 일반적으로 생각할 수 있는 방법이나 안정성 면에서 가장 취약하다. 방법3은 암호키 생성 책임이 전적으로 클라이언트 자신에게 있음으로 안전성 면에서 가장 우수하다. 하지만 암호키 생성기를 얼마나 간단하게 만들 수 있냐는 것이 문제이다. 방법3은 SK는 임의로 쉽게 지정할 수 있으나 이로부터 PK를 생성하기가 힘들 경우에 사용하는 것이 좋다.

다음은 정보 교환 단계에서 사용되는 대칭키 CK는 사용자 인증 단계에서 클라이언트가 임의로 결정하여 사용함으로 별도의 보관 및 관리가 필요하지 않다. 다만 클라이언트에서 임의의 CK를 생성하는 장치가 필요할 뿐이다.

Ⅶ. 결 론

정보 누설로부터 안전한 시스템이 되기 위해서는 상대방의 신분확인이 우선 되어야 한다. 그리고 교환되는 정보 내용이 제3자에게 누설되거나 그 내용이 변조되지 않아야 한다. 본 논문에서는 클라이언트-서버 시스템에 암호화 기법을 도입하여 이러한 요건을 만족시켜 줌으로써 클라이언트와 서버 사이의 정보를 보호할 수 있는 Secure 클라이언트-서버 시스템을 설계하였다.

이 시스템에서는 처음 사용자 인증 단계에서는 비대칭 암호화 방식을 사용하였고 그 다음 정보 교환 단계에서는 대칭 암호화 방식을 사용하였다. 그렇게 함으로써 상호간의 신분 확인도 가능하게 하였으며 암호화 처리 속도도 빠르게 할 수 있었다. 특히 비대칭 암호 방식을 도입하여 자신의 비밀키를 이용하여 암호화하여 보낼 경우 암호문 생성자의 유일성이 보장됨으로 디지털 서명 기능이 가능하게 되어 본 Secure 클라이언트-서버 시스템을 전자상거래 등 여러 분야에 활용할 수가 있다.

또한 시스템의 안전성에 큰 영향을 미칠 수 있는 암호키의 관리방안에 대하여 세 가지를 제시하고 각각의 특징을 면밀히 분석하였다. 각기 그 장단점이 있음으로 차후 암호화 시스템을 실제 구현할 때 암호키 생성기의 특징에 따라 적당한 것을 선택한다면 도움이 될 것이다.

표 1. 암호키 생성 및 배포 방법에 따른 장단점

	장 점	단 점
방법1	- 서버만이 암호키 생성기를 보유하면 된다.	- 서버의 파괴는 암호 시스템 전체의 파괴를 의미한다. - 서버의 관리자도 도청자가 될 수 있다. - SK 배포를 오프라인으로 은밀히 해야 한다.
방법2	- SK를 클라이언트 스스로 결정할 수 있고 서버에서의 노출을 최소화 할 수 있다. - 서버로부터 PK를 온라인으로 받아 볼 수 있다.	- 모든 클라이언트가 SK 생성기를 보유해야 한다.
방법3	- SK를 클라이언트 스스로 결정할 수 있고 서버를 포함하여 어느 누구도 알아낼 수 없다.	- 모든 클라이언트가 암호키 생성기를 보유해야 한다.

본 논문에서는 클라이언트-서버 시스템에 대해 연구하였으나 시스템의 구성에 따라 효율적인 암호화 방식이 다를 수 있음으로 앞으로 피어 투 피어 시스템에 대한 연구도 필요하리라 본다.

참 고 문 헌

[1] A. Cain, "Security, Authentication, and Privacy on the Web", Fourth International WWW Conference, 1995.

[2] Virginia Polytechnic Institute & State University, WWW: Beyond the Basics, Dec. 1996.

[3] A.O. Freier, P. Karton, P.C. Kocher, The SSL Protocol V3.0, Transport Layer Security Working Group, Nov. 1996

[4] G.J. Simmons, "Symmetric and Asymmetric encryption", Comput. Surv. 11, 4, Dec. 1979.

[5] W. Diffie and M.E. Hellman, "New Directions in Cryptography", IEEE Trans. on IT, Vol. 1 IT-22, No.6, Nov. 1976.

[6] Data Encryption Standard, Federal Information Processing Standard (FIPS) Publication 46, National Bureau of Standard, U.S. Department of Commerce, Washington DC, Jan. 1977.

[7] R. Rivest, A. Shamir, L. Adleman, "A method for obtaining digital signature and public-key cryptosystem", Commun. ACM 21, 2, Feb. 1978.

[8] R. Rivest, The MD5 Message-Digest Algorithm, MIT Laboratory for Computer Science and RSA Data Security Inc., Apr. 1992.

[9] Digital Signature Standard, Federal Information Processing Standard (FIPS) Publication 186, National Bureau of Standard, U.S. Department of Commerce, Washington DC, May 1994.

저자소개

이상렬

1981.3 ~ 1983.2 한양대학교 전자공학과 공학석사

1983.1 ~ 1993.2 삼성전자 컴퓨터연구실 선임연구원

1993.3 ~ 1997.8 경인여자전문대학 사무자동화과 조교수

1995.9 ~ 1998.8 한양대학교 전자공학과 박사수료

1997.9 ~ 현재 상지대학교병설전문대학 사무자동화과 전임강사