(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(51) Int. Cl.[6]
G06F 15/16

(11) 공개번호 특1998-050938
(43) 공개일자 1998년09월15일

(21) 출원번호 특1996-069786
(22) 출원일자 1996년12월21일

(71) 출원인 한국전자통신연구원 양승택
대전광역시 유성구 가정동 161한국전기통신공사 이준
서울특별시 종로구 세종로 100번지
(72) 발명자 최준혁
대전광역시 유성구 전민동 나래아파트 106동 601호
고병도
대전광역시 유성구 어은동 한빛아파트 115동 501호
류재철
대전광역시 유성구 궁동 충남대학교 컴퓨터 과학과
김태갑
대전광역시 유성구 궁동 충남대학교 과학과
(74) 대리인 박해천, 원석희

심사청구 : 있음

(54) 인터넷 상에서 암호환된 문서 전송방법

요약

1. 청구범위에 기재된 발명이 속한 기술분야

인터넷 상에서의 암호화된 문서 전송방법.

2. 발명이 해결하려고 하는 기술적 과제

인터넷상에서 데이터의 전송시 발생할 수 있는 불법적인 정보 유출이나 데이터 변조 등을 막고자 함.

3. 발명의 해결방법의 요지

IDEA(International Data Encryption Standad) 암호화 알고리즘을 이용한 암호화 기법과 MD를 이용한 메세지 인증 기법을 이용하여 문서의 기밀성 보장 및 전송시문서 변조를 막기 위해 서버와 클라이언트간 세션키를 공유하고, 사용자 데이타를 읽어 암호화된 세션키와 해쉬값을 생성하여 서버로 전송하며, 서버는 수신된 세션키를 해독하여 해쉬값을 검사하고, 응답 메세지를 생성하여 클라이언트로 전송하고, 클라이언트는 해쉬값이 정상이면 화일을 읽어 암호화하고, 제어데이타와 함께 서버로 전송하며, 서버는 암호화된 데이타를 복호하여 해쉬값을 검사하고, 응답 메세지를 클라이언트로 전송하는 과정을 통해 이루어짐.

4. 발명의 중요한 용도

인터넷에서의 기밀 유지를 위한 문서 전송에 이용됨.

대표도

도3

명세서

도면의 간단한 설명

도 1 은 본 발명이 적용되는 암호화 문서 전송시스템의 구조도,

도 2 는 본 발명에 따른 세션키 공유 절차 설명도,

도 3 은 본 발명에 따른 암호화된 문서 송신 흐름도,

도 4 는 본 발명에 따른 암호화된 문서 수신 흐름도,

도 5 는 본 발명에 다른 키 관리 모듈의 구성도,

도 6 은 본 발명에 따른 키관리 흐름도.

발명의 상세한 설명

발명의 목적

발명이 속하는 기술분야 및 그 분야의 종래기술

본 발명은 현재 전세계적으로 관심의 초점이 되고있는 WWW(World Wide Web) 인터페이스로 이용하여 특정 사용자 구룹 내에서의 보안이 요구되는 문서 전송시 암호화된 문서 전송 방법에 관한 것이다. WWW은 인터넷에서 다루어지는 모든 데이터 형식(그림, 동화상, 소리)을 처리하여 보여 줄 수 있는 단일화된 인터페이스로서 그 기능의 확장이 무한하다고 할수 있다. 그러나, 데이터의 전송시 발생할 수 있는 불법적인 정보 유출이나 데이터 변조 등을 막는 방법이 제공되고 있지 않기 때문에 비공개를 원칙으로 하는 문서나 데이터의 경우 전송시 보안상의 치명적인 문제점을 안고 있다.

현재 이러한 암호화와 관련하여 SSL(Security Socket Layer)나 S-HTTP(Secured Hypertext Transfer Protoco1)등의 기술을 이용하여 만들어진 상품들이 있기는 하지만 모두가 고가이며, 각 서버에 대해 별도의 인증을 받아야 하는 등, 그 사용법이 불편한 문제점이 있었다.

발명이 이루고자 하는 기술적 과제

따라서, 상기 문제점을 해결하기 위한 본 발명은 현재 전세계적으로 가장 많이 사용되고 있는 WWW을 기반으로 하여 암호화된 문서 송수신을 수행하는 기능으로 IDEA(International Data Encryption Standad) 암호화 알고리즘을 이용한 암호화 법과 MD5를 이용한 메세지 인증 기법을 이용하여 문서의 기밀성 보장 및 전송시문서 변조를 방지할 수 있는 인터넷 상에서의 암호화된 문서 전송방법을 제공하는데 그 목적이 있다.

발명의 구성 및 작용

상기 목적을 달성하기 위한 본 발명은, 인터넷을 통한 통신 시스템에 적용되는 암호화 문서 전송방법에 있어서, 클라이언트와 서버는 암호화에 사용될 세션키를 교환하고, 공유하는 제 1 단계, 클라이언트는 전송하고자 하는 사용자 데이타를 읽어 암호화된 세션키와 해쉬값을 생성하여 서버로 전송하는 제 2 단계, 서버는 수신된 세션키를 해독하여 해쉬값을 검사하고, 응답 메세지를 생성하여 클라이언트로 전송하고, 세션키는 저장하는 제 3 단계, 클라이언트는 응답메세지를 분석하여 해쉬값의 정상 유무를 확인하는 제 4 단계, 해쉬값이 에러이면 종료하고, 정상이면 화일을 읽어 암호화하고, 제어데이타와 함께 서버로 전송하는 제 5 단계, 서버는암호화된 데이타를 복호하여 해쉬값을 검사하고, 응답 메세지를 클라이언트로 전송하는 제 6 단계, 및 클라이언트는 응답메세지를 분석하여 해쉬값이 정상이면 데이타 전송을 종료하는 제 7 단계를 포함하여 이루어진 것을 특징으로 한다.

본 발명은 현재 전세계적으로 가장 많이 사용되고 있는 WWW을 통하여 특정 그룹 내에서 이루어지는 문서 전송을 암호화 함으로써, 허가 받지 않은 사용자로부터 그 내용을 보호함은 물른 불법적인 변조롤 막을 수 있는 기법을 제공한다. 암호화된 문서 송수신을 수행하는 기능으로 IDEA(International Data Encryption Standard) 암호화 알고리즘을 이용한 암호화 기법과 MD5를 이용한 메세지 인증 기법을 이용하여 문서의 기밀성 보장 및 전송시 문서 변조의 방지 기능이있다. 이러한 방법을 이용한 경우 개개인이 자기 키를 관리할 수 있어 중앙에서 키를 관리하는 부담을 줄일수 있으며, 클라이언트 쪽에서 전송하려는 문서 자체를 완전히 암호화 하여 전송하므로 이를 해독하기는 거의 불가능하다. 일반적인 WWW상에서의 데이터 전송의 경우 전송되는 데이터가 허가 받지 않은 사용자에게도 그대로 노출이 될 수 있을 뿐 아니라 전송도중 변경될 수도 있다.

본 발명은 IDEA 알고리즘을 이용하여 데이터를 암호화하고, MD5 해쉬함수를사용하여 데이터를 암호화하여 전송하고, 이를 다시 복호화하는 구조롤 가지고 있다. 이를 위해서는 전송시 사용될 여러 가지 키의 정의 및 관리, 변경하는 작업이 추가적으로 요구된다. 본 발명에서 이루어지는 문서의 전송은 크게 서버로부터 각사용자에게 문서가 전송되는 경우와 각 사용자가 서버에 문서를 등록하는 두 가지 경우로 구분된다.

이하, 첨부된 도면을 참조하여 본 발명의 일실시예를 상세히 설명하기로 한다.

도 1 은 본 발명이 적용되는 암호화 문서 전송시스템의 구조도를 나타낸다.

도 1 에 나타낸 바와 같이 사용자 측은 크게 브라우저와 암호화/인증 모듈, 그리고 키 관리 모듈로 구성되며, PC를 사용하는 사용자를 대상으로 한다. 브라우저와 암호화/인증 모듈간의 데이터 교환을 위해서는 네스캐프사에서 제공하는 NCAPI함수를 이용한다. 여기서의 키관리 모듈의 역할은 디스켓에 암호화된 형태로 저장되어 있는 사용자의 키를 복호화해서 암호화 프로그램에 제공하거나 사용자 키를 관리하는데 사용된다.

실제적인 문서의 송수신을 수행하는 것은 WWW 브라우저이며, 암호화 프로그램은 단지 브라우저와의 데이터 교환을 통한 데이터의 암호화/복호화 만을 수행한다. 따라서 본 발명에서 제공되는 문서 송수신은 HTTP를 기반으로 수행된다. 서버측은 크게 HTTPD 프로세스와 암호화/인증 모듈, 키관리 프로그램으로 구성되며, 유닉스(UNIX) 시스템을 대상으로 한다. HTTPD는 일반적으로 사용되는 WWW 서버를 의미하며, CGI 프로그램을 이용해서 암호화/인증 모듈을 수행한다. 서버 측 키 관리 프로그램은 전송을 요구한 사용자의 키를 찾아서 CGI 프로그램에 전달하거나 사용자 측의 키 관리 프로그램의 요구에 따른 키의 변경이나 인증 작업을 수행한다. 사용자 측과 서버 측의 키 관리 프로그램 사이의 데이터 전송은 유닉스(UNIX) 소켓 인터페이스를 기반으로 한다. 한편 시스템의 보안 기능의 강화와 키노출 방지를 위해서는 여러 가지 복합적인 키들을 사용하게 되는데 다음은 본 발명에서 제시된 키들에 대한 설명이다.

첫째, 기본키(Kb)는 암호화 통신에 가장 기본이 되는 64비트 키로써 사용자와 서버간에 공유되어야 하며, 키의 비밀성 유지와 효율적인 관리를 위하여 암호화에 직접 사용이 되지는 않지만 실제 암호화를 위해 사

용자와 서버간에 공유되어 사용되는 일종의 임시키인 세션키의 생성과 전달을 위해서 사용된다. 본 암호화 전송시스템에 등록하는 모든 사용자는 시스템 관리자로부터 하나의 기본 키를 받아서 사용하게 되는데 이때 보안상의 비밀성을 높이기 위해서 3.5인치 플로피 디스크에 기본 키를 기록해서 전달한다. 서버는 사용자들의 문서 송수신 요청에 응답하여야하기 때문에 각 사용자들의 기본 키를 모두 가지고 있어야 한다. 서버에서의 기본키의 저장 형태는 ID:Encryptedkey:이며 ID는 유일한 것이어야 한다. 기본키는 암호화되어 저장되어야 하는데 이때의 암호화를 위해서는 서버키가 사용된다. 사용자에게 전달되는 기본 키는 사용자키로 암호화되어 저장된다. 기본키의 암호화에 사용되는 알고리즘은 IDEA이다.

둘째, 서버키(Kv)는 128비트 크기로 사용자들의 기본키를 암호화 하기 위하여 사용된다. 서버 키는 시스템 관리자 만이 알고 있어야 하며, 암호화 되지 않은 형태로 따로 저장되지는 않는다. 서비키도 다른 기본 키들처럼 ID:EncryptedKey:의 형태로 기본키들과 함께 저장되지만 root''라는 ID를 가지며 UNIX의 패스워드메커니즘을 이용해서 암호화 된다. UNIX 패스워드 메커니즘은 단 방향(1-way) 해쉬(Hash) 기능(Function) 이므로 사실상 복호화가 불가능하다.

세째, 메세지키(km)는 암호화 통신에 사용되는 세션키를 생성하는데 사용되는 일종의 임시 키로써 64비트의 크기를 갖는다. 일종의 랜딤키이기 때문에 따로 보관되지는 않으며, 기본 키와 합성되어 세션키를 생성한다.

네째, 세션 키(ks)는 사용자와 서버간의 암호화 통신에 직접 사용되는 일종의 임시키로써 문서 전송이 끝나면 사라진다. 올바른 암호화/복호화를 위해서 사용자와 서버에 의해서 공유 되어야 하며, 문서 전송이나 수신을 요청하는 사용자 측에서 생성하는 것을 원칙으로 하며, 기본키로 암호화 되어서 서버로 전달된다. 이때 서버는 모든 사용자들의 기본 키를 알고 있기 때문에 사용자가 사용하고자 하는 세션키에 대한 인증을 할수 있으며, 인증된 사용자의 요청일 경우 이를 파일에 보관해서 사용한 후 문서 전송이 종료되면 삭제하게 된다. 세션 키는 64비트의 기본키와 64비트의 메세지키의 조합으로 생성된다.

다섯째, 사용자키(kc)는 기본키를 허가받지 않은 사용자로부터 보호하는 것은 암호화 송수신의 가장 기본이 되는 일이기 때문에 시스템 관리자는 사용자에게 분배되는 기본키를 플로피 디스크에 저장해서 전달하는 것을 원칙으로 한다. 그러나 일반적인 시스템 사용자는 보안에 대한 인식이 부족할수 있고 디스크의 분실에 대비하여 사용자키라는 사용자만이 알고 있는 키로의 암호화가 필요하다. 사용자와 서버간의 문서 송수신은 크게 두 부분으로 구분된다. 첫번째는 암호화에 사용될 세션키의 교환과 공유이며, 두번째는 공유된 세션키를 이용한 암호화된 문서 송수신이다. 세션키나 문서 전송의 경우 모두 암호화와 메세지 인증을 기본으로 한다.

도 2 를 통해 우선 세션 키의 교환 과정을 살펴본다. 클라이언크가 서버에게 문서를 송신하거나 수신하고자 할 때 우선 요청 메세지를 보내게 되면 서버는 요청(Request)의 종류에 따라 특별한 내용 형식을 되돌린다. 브라우저는 서버로부터 전송되어온 내용 형식에 따라 도움 응용 프로그램을 구동하게 되는데 이는 사용자에 의해 미리 등록되어 있어야 한다. 일단 도움 응용(helper application) 프로그램이 구동 되면 도움 응용 프로그램은 NCAPI를 이용하여 브라우저와 데이터를 교환한다. 클라이언트에서 자신의 식별자(ID)와 기본 키로 암호화한 세션키를 해쉬값과 함께 서버로 전송하면 서버는 사용자의 식별자(ID)를 보고 해당 기본키를 찾아서 세션키를 복호화 해낸 후, 이를 다시 기본 키와 조합해서 암호화한 후 되돌린다. 클라이언트는 서버가 전송한 메세지를 자신의 기본키로 복호화한 후, 자신의 기본키와의 조합한 결과가 이전에 만들어서 전송한 세션키와 일치할 경우 서버와 성공적으로 키가 공유되었음을 확인한다.

세션키의 공유가 이루어지면 실제 전송할 데이터들의 전송이 이루어지는데, 도 3 은 본 발명에 따른 암호화된 문서를 송신하는 과정을 나타낸다. 우선 사용자가 서버로 문서를 송신하는 경우와 서버로부터 문서를 수신하는 2가지 경우로 구분되며, 각 과정은 세션키의 공유 과정과 유사한 형태로 진행된다. 여기서 제어정보(Control info.)란 현재 전송되고 있는 데이터 정보를 나타내는 부분으로 이를 통해서 전송을 제어하게 된다. 먼저, 도움 프로그램을 통하여 파일을 송수신 하는 프로그램을 띄운다. 이프로그램은 해당 사용자에 대한 각종 정보를 읽어 드린 뒤(101) 암호화 된 세션키와 해쉬(Hash) 값을 생성한다(102). 이 값들은 네스케이프를 통하여 세션키를 해독하고(103), 해쉬 값을 검사한 후(104) 이 검사에 대한 응답을 하고 세션키를 저장한다(105). 검사 결과 만약 해쉬(Hash) 값이 틀리면(106) 프로그램은 종료되며, 해쉬 값이 올바른 경우는(106) 파일을 읽어서(107) 암호화된 데이타와 제어 데이타를 전송한다(108). 전송된 데이터는 암호가 풀리게 되고(109), 해쉬(Hash) 값을 검사하여(110)응답 메세지를 전송하고(111), 올바르면 전송을 종료하고, 해쉬(Hash) 값이 틀리면 프로그램을 빠져나간다.

도 4 는 본 발명에 따른 암호화된 문서 수신 처리 흐름도를 나타낸다. 먼저 도움 프로그램을 통하여 파일을 송수신 하는 프로그램을 띄운다. 이 프로그램은 해당 사용자에 대한 각종 정보를 읽어 드린 뒤(201) 암호화 된 세션키와 해쉬(Hash)값을 만들어 낸다(202). 이 값들은 네스케이프를 통하여 세션키를 해독하고(203), 해쉬(Hash) 값을 검사한 후(204) 이 검사에 대한 응답을 하고, 세션키를 저장한다(205). 만약 해쉬(Hash)값이 틀리면(206) 프로그램은 종료되며, 해쉬 값이 올바른 경우는 서버는 제어 데이터를 전송하고, 파일을 전송한다(207). 클라이언트는 파일을 모두 수신한 뒤에(208,209) 제어 데이터를 서버로 전송하며(211) 서버는 이들 데이터를 해독하고(212), 해쉬(Hash)값을 검사하여(213) 이상이 없으면 파일 전송을 종료한다.

도 5 는 암호화 시스템에서 사용되는 각종 키들을 관리하기 위한 프로그램들에 대한 설명이다. 사용자는 기본키 관리기를 이용하여 자신의 기본키를 서버로부터 인증 받을수 있을 뿐만 아니라 노출되었다고 의심될 경우 이를 변경할 수도 있다. 물론 기본키의 확인 및 변경은 암호화, 메세지 인증을 기반으로 수행되기 때문에 불법적인 사용자의 접근으로부터 보호받을 수 있다. 사용자의 관리기는 자신의 기본키를 암호화해서 저장하는데 사용된 사용자키를 변경하는 기능을 제공하며, 서버키 관리기 서버키의 변경에 사용된다. 서버키의 변경의 경우에는 현재 저장된 모든 사용자들의 기본키를 다시 암호화해야 하며, 변경 도중에는 사용자의 요청에 서비스를 제공할 수 없다는 점을 고려해야 한다. 세션키 생성기는 기본키와 메세지키를 이용하여 세션키를 생성하며, 메세지키 생성기와 Nonce 생성기 등은 쉽게 추측될 수 없는 랜덤키를 생성한다. 서버측의 기본키 생성기를 이용해서 만들어진 기본키는 기본키 분배기를 통해서 디스켓에 저장된 후, 사용자들에게 분배된다. 시스템 관리자는 기본키 분배기를 사용해서 서버 시스템에 원격으로 접속

해서 서버키를 확인 받은 후에 디스컷을 만들 수 있다.

도 6 은 본 발명에 따른 키 관리 방법의 흐름도를 나타낸다. 키관리 윈도우가 수행된 후, 사용자 데이타를 읽어(301) 암호화된 메세지와 해쉬값을 생성하여 전송한다(302). 암호화된 키는 소켓 인터페이스를 통해 해독되고(303), 해쉬값을 검사한 후(304), 키를 생성하고(305), 이는 다시 소켓 인터페이스를 통해 전송되고, 응답 메세지를 검사하여(306) 검사 결과를 출력한다(307). 이와 같은 과정은 키 값을 만들때마다 계속해서 반복된다.

이상에서 설명한 본 발명은 본 발명이 속하는 기술분야에서 통상의 지식을 가진자에게 있어 본 발명의 기술적 사상을 벗어나지 않는 범위내에서 여러가지 치환, 변형 및 변경이 가능하므로, 전술한 실시예 및 도면에 한정되는 것이 아니다.

발명의 효과

상기와 같이 이루어지는 본 발명은 웹(Web) 상에서 문서 전송시 암호화 및 메세지 인증 과정을 수행함으로써, 허가받지 않은 사용자로부터 문서의 내용을 보호할 수 있고, 이를 통해 사용자는 자신이 가지고 있는 데이터를 안전하게 전송할수 있는 효과가 있다.

(57) 청구의 범위

청구항 1

인터넷을 통한 통신 시스템에 적용되는 암호화 문서 전송방법에 있어서, 클라이언트와 서버는 암호화에 사용될 세션키를 교환하고, 공유하는 제 1 단계,

클라이언트는 전송하고자 하는 사용자 데이타를 읽어 암호화된 세션키와 해쉬값을 생성하여 서버로 전송하는 제 2 단계,

서버는 수신된 세션키를 해독하여 해쉬값을 검사하고, 응답 메세지를 생성하여 클라이언트로 전송하고, 수신된 세션키를 저장하는 제 3 단계,

클라이언트는 응답메세지를 분석하여 해쉬값의 정상 유무를 확인하는 제 4단계,

해쉬값이 에러이면 종료하고, 정상이면 화일을 읽어 암호화하여 제어 데이타와 함께 서버로 전송하는 제 5 단계,

서버는 암호화된 데이타를 복호하여 해쉬값을 검사하고, 응답 메세지를 클라이언트로 전송하는 제 6 단계,및

클라이언트는 응답메세지를 분석하여 해쉬값이 정상이면 데이타 전송을 종료하는 제 7 단계를 포함하여 이루어진 인터넷 상에서의 암호화된 문서 전송방법.

청구항 2

제 1항에 있어서,

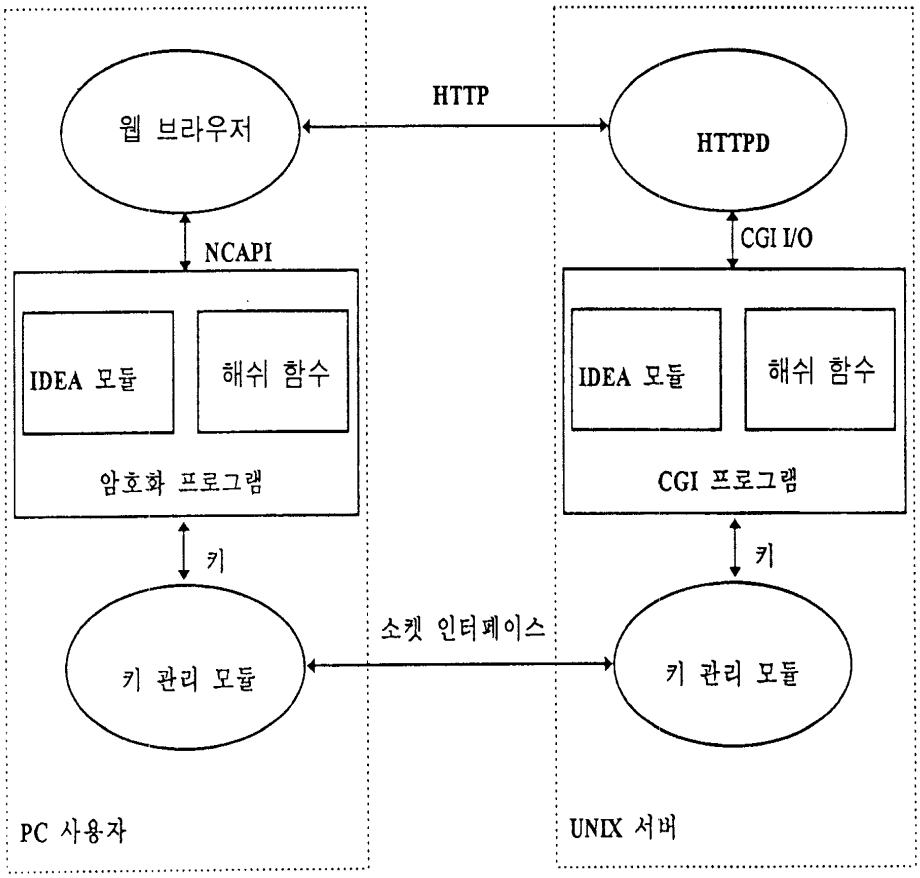
상기 제 1 단계는, 클라이언트에서 자신의 식별자와 기본 키로 암호화한 세션키를 해쉬값과 함께 서버로 전송하는 단계,

서버는 사용자의 식별자를 보고 해당 기본키를 찾아 세션키를 복호화한 후,이를 다시 기본 키와 조합해서 암호화하여 클라이언트로 전송하는 단계, 및

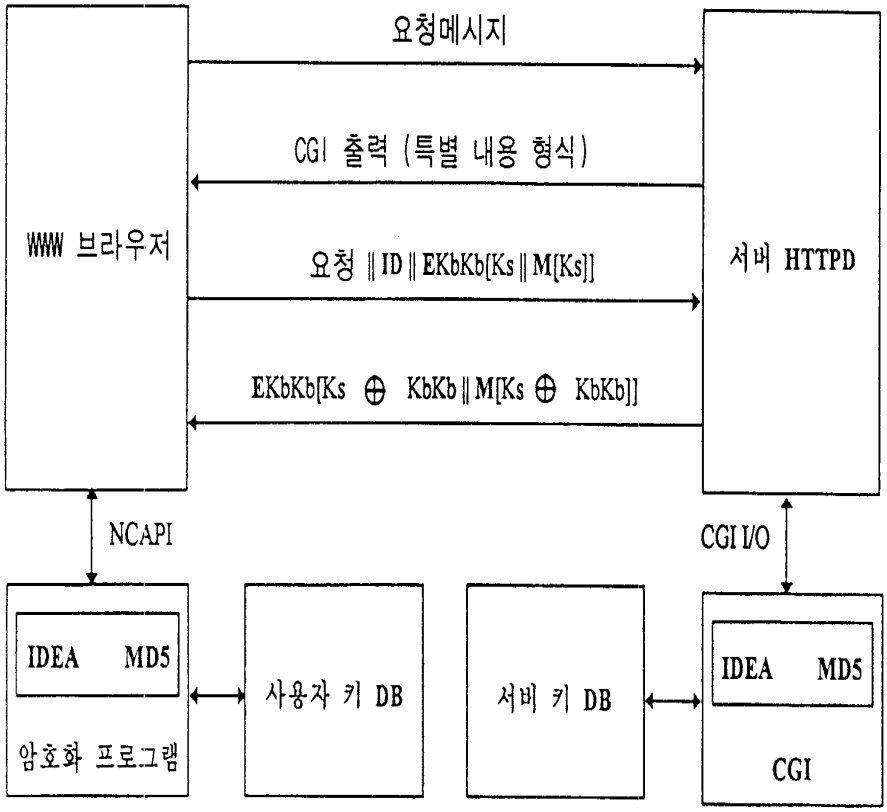
클라이언트는 서버가 전송한 메세지를 자신의 기본키로 복호화한 후, 자신의 기본키와의 조합한 결과가 이전에 만들어서 전송한 세션키와 일치할 경우 서버와 성공적으로 키가 공유되었음을 확인하는 단계를 포함하여 이루어진 것을 특징으로하는 인터넷 상에서의 암호화된 문서 전송방법.

도면

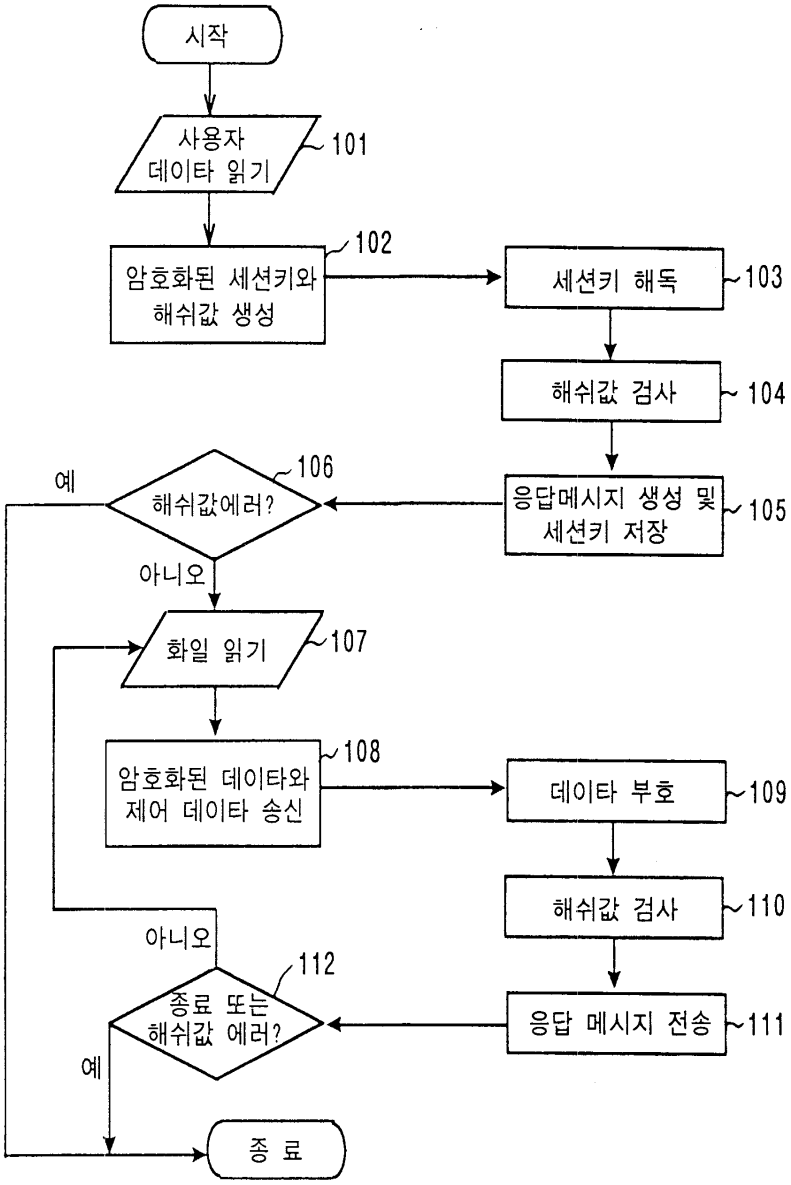
도면1


웹 브라우저
HTTP
HTTPD
NCAPI
CGI I/O
IDEA 모듈
해쉬 함수
암호화 프로그램
IDEA 모듈
해쉬 함수
CGI 프로그램
키
키
소켓 인터페이스
키 관리 모듈
키 관리 모듈
PC 사용자
UNIX 서버

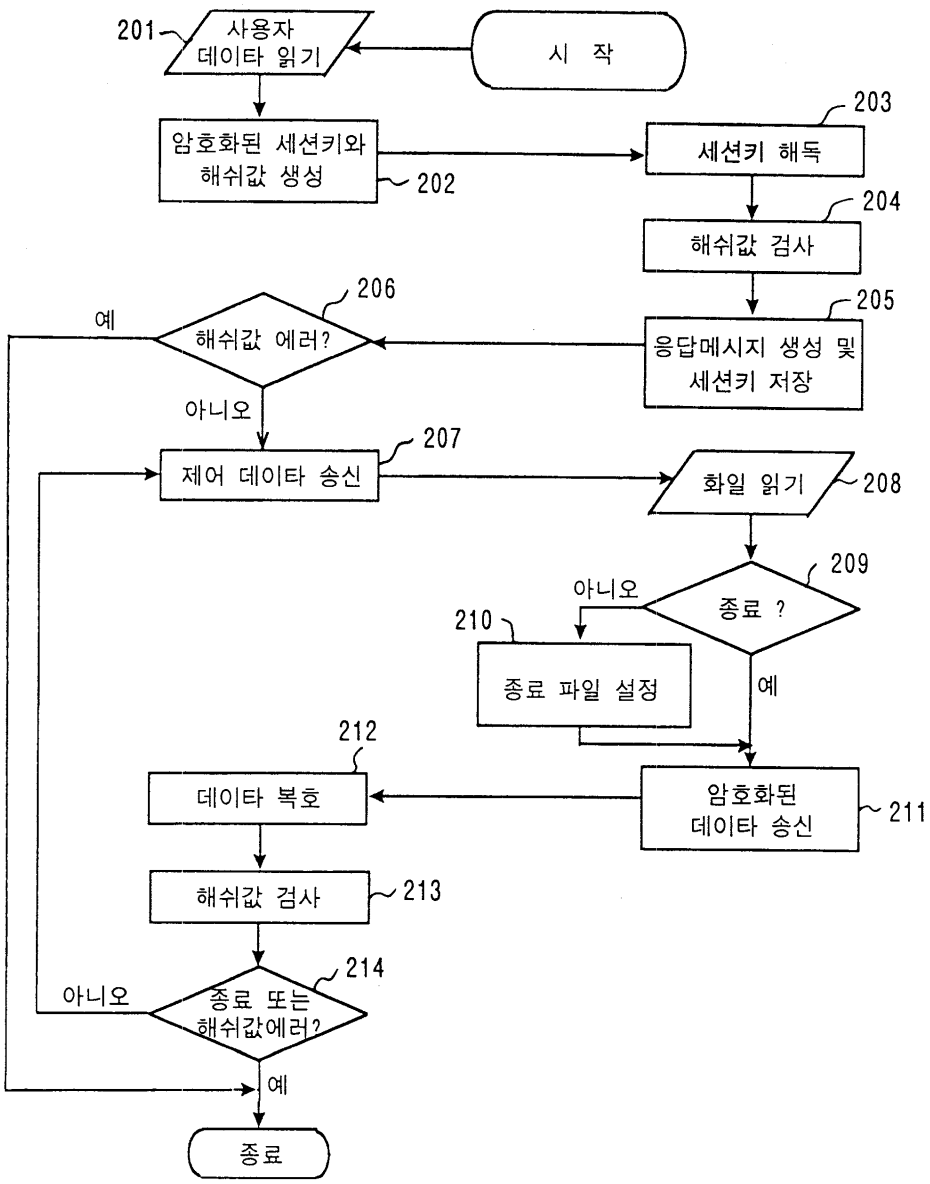
도면2


요청메시지
CGI 출력 (특별 내용 형식)
WWW 브라우저
요청 ‖ ID ‖ EKbKb[Ks ‖ M[Ks]]
서버 HTTPD
EKbKb[Ks ⊕ KbKb ‖ M[Ks ⊕ KbKb]]
NCAPI
CGI I/O
IDEA MD5
암호화 프로그램
사용자 키 DB
서버 키 DB
IDEA MD5
CGI

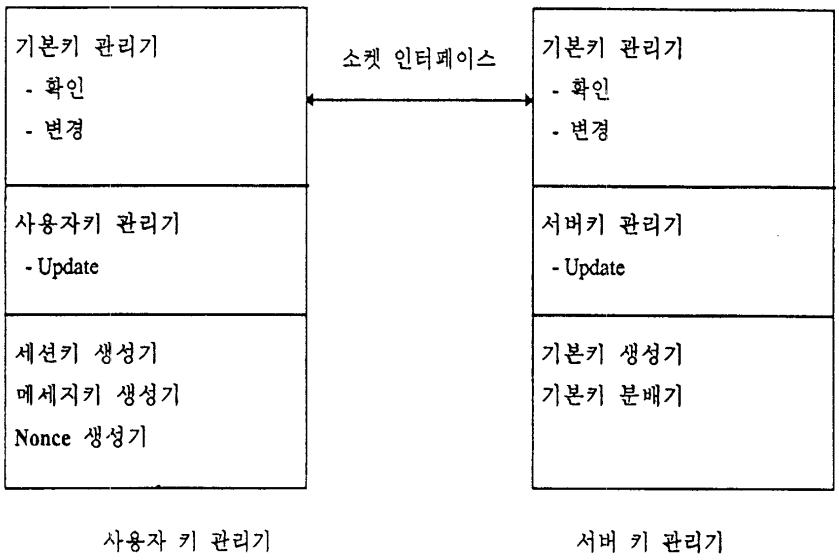
도면3


시작
사용자 데이타 읽기
101
암호화된 세션키와 해쉬값 생성
102
세션키 해독
103
해쉬값 검사
104
응답메시지 생성 및 세션키 저장
105
해쉬값에러?
106
예
아니오
화일 읽기
107
암호화된 데이타와 제어 데이타 송신
108
데이타 부호
109
해쉬값 검사
110
응답 메시지 전송
111
종료 또는 해쉬값 에러?
112
아니오
예
종 료

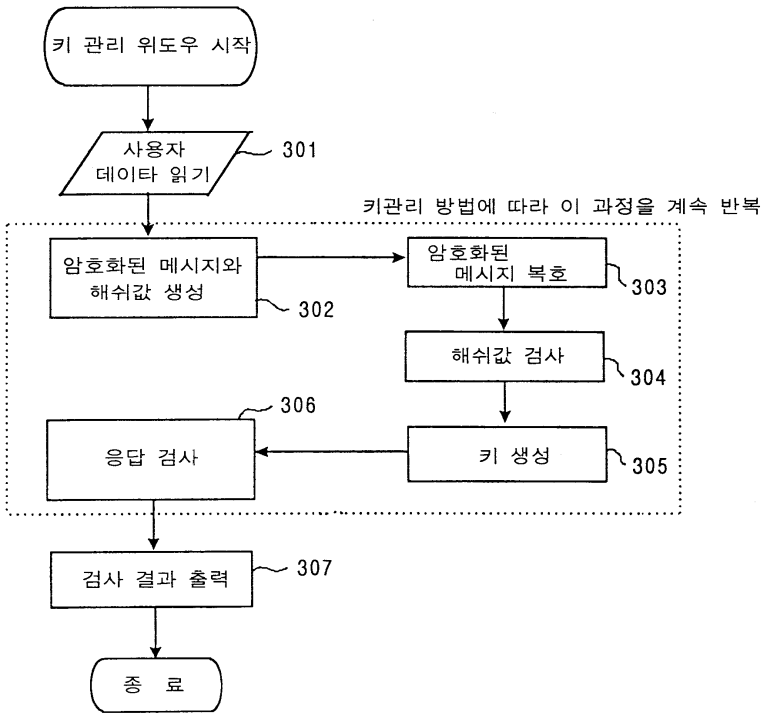
도면4


시 작
201
사용자 데이타 읽기
암호화된 세션키와 해쉬값 생성
202
203
세션키 해독
204
해쉬값 검사
205
응답메시지 생성 및 세션키 저장
206
해쉬값 에러?
예
아니오
207
제어 데이타 송신
208
화일 읽기
209
종료 ?
아니오
210
종료 파일 설정
예
212
데이타 복호
암호화된 데이타 송신
211
213
해쉬값 검사
214
종료 또는 해쉬값에러?
아니오
예
종료

도면5


기본키 관리기
- 확인
- 변경
소켓 인터페이스
기본키 관리기
- 확인
- 변경
사용자키 관리기
- Update
서버키 관리기
- Update
세션키 생성기
메세지키 생성기
Nonce 생성기
기본키 생성기
기본키 분배기
사용자 키 관리기
서버 키 관리기

도면6


키 관리 위도우 시작
사용자 데이타 읽기
301
키관리 방법에 따라 이 과정을 계속 반복
암호화된 메시지와 해쉬값 생성
302
암호화된 메시지 복호
303
해쉬값 검사
304
키 생성
305
응답 검사
306
검사 결과 출력
307
종 료