2000 년 한국정보처리학회 춘계 학술발표논문집 제 7 권 제 1 호

암호 라이브러리를 이용한 안전한 인터넷 뱅킹 시스템 설계 및 구현

김진묵*, 유황빈*
*광운대학교 컴퓨터과학과
ggang@chollian.net

Design and Implementation of Secure Internet Banking System using Cryptography Library

Jin-Mook Kim*, Hwang-Bin Ryou*
*Dept. of Computer Engineering, Kwang-woon University

요 약

많은 사람들이 관심을 가지고 급속도로 발전하는 인터넷 환경의 웹 서비스 중에서 인터넷 뱅킹 시스템은 반드시 필요한 서비스 중의 하나지만, 아직까지 많은 보안상의 문제점을 내포하고 있다.

본 논문에서는 이런 보안상의 문제들 중에서 사용자 인증에 관한 부분, 데이터 암호화에 관한 부분, 키 분배 문제에 관한 부분을 해결할 수 있는 방안을 제시하려 한다.

이를 위해 공개적으로 사용이 가능한 암호 라이브러리인 Crypto++3.1 을 이용하여 인터넷 환경에서 보안 서비스를 제공할 수 있는 안전한 인터넷 뱅킹 시스템인 SIBS(Secure Internet Banking System)을 설계 및 구현하였다.

SIBS 는 빠른 데이터 암호화 처리를 위해 IDEA 암호 알고리즘을 사용하였다. 데이터 암호화에 사용할 키를 분배하기 위해서 Diffie-Hellaman 키 분배 알고리즘을 이용한다. 또한, 사용자의 인증을 위해 X.509 형식의 인증서를 이용하기 위해서 SSLeay 를 설치하여 인증서(Certificate)를 발급 받는다.

그러므로, 사용자는 인터넷에서 SIBS 의 GUI(Graphic User Interface)를 이용해 빠르고 편리한 접근이 용이하고, 암호 알고리즘에 대한 지식이나 특별한 조치가 없이도 빠른 데이터 암호화 처리와 인증서를 이용한 확실한 사용자 인증을 보장 받을 수 있다.

1. 서론(Introduction)

1.1.연구배경

현대 사회는 인터넷 시대이다. 전 세계적으로 모든 컴퓨터 사용자들이 정보의 공유를 목적으로 만들어진 인터넷은 이제는 컴퓨터 환경에서 없어서는 안 될 지경에 이르게 되었다. 현실적으로 인터넷이라는 말을 제외한 컴퓨터 환경은 이제 컴퓨터 사용자들에게 있어서 무용지물과 같이 느껴질 수도 있다. 이처럼 발달한 인터넷은 초기 발전 단계에는 교육, 과학적인 목적을 지닌 과학자들만이 관심을 가지고 사용하였다. 이런 웹 서비스가 조금 더 그래픽적이고 사용자에게 친숙한 형태의 웹-브라우저(Web-Browser) 프로그램이 개발되면서 많은 사람들에게 널리 사용되게 되었다.

널리 사용되기 시작하게 된 인터넷 환경은 급기야 많은 사람들이 상업적인 분야까지 이용하기를 원하게 되었다. 예를 들어, 인터넷 상에서 많은 정보를 가지고 있어 필요로 하는 정보를 얻을 수 있도록 해 주는 정보 은행(Data-Bank), 필요로 하는 물건을 백화점이나 상점까지 직접 방문하지 않고도 가정에서 주문을 해서 구매할 수 있는 인터넷 쇼핑몰(Shopping-Mall), 원하는 물건을 구매하고 구매한 물건에 대한 댓가를 지불하기 위한 지불 시스템(Payment-system)[5], 문서 결재를 기존의 방식과는 다르게 인트라넷(Intranet)상에서 해결 하려는 전자결재(Eletronic-Approval) 시스템[4] 등과 같은 시스템들이 현재 개발되어 사용 중이거나, 개발에 많은 투자가 이루어지고 있다. 그 밖에도 다양한 형태로 상업적인 목적으로 인터넷은 이용되고 있다.

상업적인 목적으로 인터넷 환경을 이용하기를 원하는 사람들에 의해 인터넷 뱅킹(Internet-Banking)이라는 가상의 은행에 대한 욕구가 생겨나게 되었다. 이와 같은 욕구로 인해 현재는 폰-뱅킹(Phone-Banking) 서비스, 홈-뱅킹(Home-Banking)서비스라는 이름으로 인터넷 뱅킹 서비스[6,7]의 중간 형태인 서비스들이 은행 이용 고객들의 욕구에 대해 불충분하지만 해결해 주고 있다. 하지만, 이런 중간 형태의 서비스들은 진정으로 인터넷 환경을 이해하고 구현되어진 것이 아닌 사용자의 욕구를 만족시키기 위해 인터넷 환경이 아니거나 약간의 인터넷 서비스를 활용하여 구현된 서비스들에 불과하다.

현실적으로 많은 은행의 고객들이 인터넷 뱅킹 시스템을 간절하게 원하고 은행측에서도 인터넷 뱅킹 서비스를 제공해야 한다. 하지만, 보안 서비스에 대한 확신을 가질 수 없기 때문에 이를 기피하는 경향이 있다.

이러한 보안상의 문제점들과 사용자의 편리성을 제공하기 위해 본 논문에서는 공개된 암호 라이브러리[16,17]를 이용하여 보안 응용 프로그램과 웹 서비스 상에서의 사용자 인증을 위한 인증 시스템[8], 인터넷 환경에서 웹 브라우저를 통한 사용자 인터페이스를 설계 및 구현하였다.

1.2.연구 목적

본 논문에서는 공개된 암호 라이브러리를 이용하여 고객과 은행이 원하는 확실한 보안 서비스를 제공할 수 있고, 편의성을 제공하는 안전한 인터넷 뱅킹 시스템을 설계 및 구현하는 것을 목적으로 한다. 기존의 인터넷 뱅킹 시스템이 가지는 보안상의 문

제점들을 보완하고 고객에게 편의성을 제공할 수 있도록 연구하였으며 이를 위해 독립적인 응용 프로그램 모듈 및 웹 서비스 상에서의 사용자 인터페이스 부분에 관한 연구를 한다

1.3.연구 내용 및 구성

본 논문은 공개된 암호 라이브러리를 이용하여 안전한 보안 서비스를 제공하는 안전한 인터넷 뱅킹 시스템의 설계 및 구현에 관한 것이다.

안전한 인터넷 뱅킹 시스템을 설계 및 구현하기 위해서 보안 기반지식과 인터넷 뱅킹 시스템에 대해 연구하였고 [1,2,9,10,11,12,13,14,15], 이러한 기초 지식을 바탕으로 안전한 인터넷 뱅킹 시스템을 설계 및 구현하였다.

안전한 인터넷 뱅킹 시스템을 설계하기 위해 설계 모듈을 세분화 하여 시스템을 3 개 부분으로 나누어 구성하였다. 각각의 시스템에 대해 살펴보면, 데이터 암호 처리 시스템은 공개된 암호 라이브러리를 이용하여 독립된 응용 프로그램의 형태로 설계하였다. 사용자 인증 시스템은 외국의 수출 문제에 걸리지 않는 Eric Young의 SSLeay[5]를 설치하여 독립적인 인증서를 발행할 수 있는 CA 시스템을 구축하였다. 마지막으로 사용자 인터페이스 시스템은 Windows 환경에서 사용자에게 친숙한 그래픽 환경을 제공하도록 하기 위해 웹 브라우저를 통해 독립적으로 처리된 결과를 보여주도록 구성하였다.

이처럼 각각의 독립적인 시스템을 하나로 묶어 서버와 클라이언트로 시스템으로 구현하였으며, 사용자 편의성과 확실한 신분 인증, 믿을 수 있는 보안 서비스를 제공할 수 있도록 설계 및 구현하였다.

2.기반 지식

본 장에서는 인터넷 상에서 행해지는 정보 전송들에 대해 가해지는 보안 공격들과 이러한 보안 공격들을 막기 위한 보안 서비스들에 대해서 기술한다. 또한 SIBS를 설계하는데 사용한 사용자 인증에 관한 사항과 키 분배를 위한 Diffie-Hellman 알고리즘, 데이터 암호화에 사용하는 IDEA알고리즘, 인터넷 뱅킹 시스템에 대해서 기술한다.[12,13,14,15]

2.1.보안 공격

흔히 사용하고 있는 인터넷 서비스는 초기 설계시에 보안을 고려하지 않고 만들어졌다. 이러한 인터넷 서비스가 주로 사용하는 HTTP 는 TCP/IP를 기반으로 하고 있다. 그러므로, 대부분의 보안 공격들은 TCP/IP를 이용한 것들이 많다. 인터넷 상에서 발생할 수 있는 보안 공격들의 유형을 분류하여 살펴보면 다음과 같다[1,12,13].

- 가로막기(Interruption)
- 가로채기(Interception)
- 위조(Modification)
- 신분위장(Fabrication)

보안 공격들을 간단히 그림으로 나타내보면 그림 2.1과 같다.

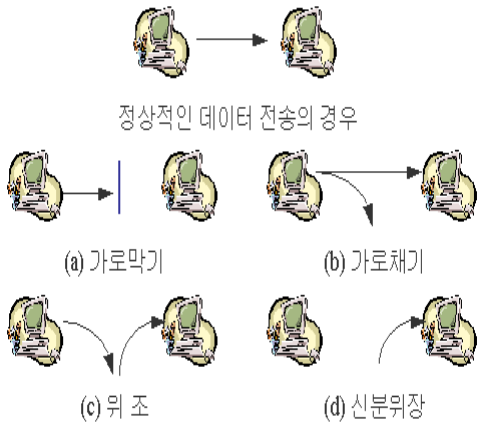


그림 2.1 보안 공격의 유형

이처럼 나쁜 목적을 가진 공격자에 의해서 인터넷을 사용하는 정당한 사용자들이 자신의 정보를 도청 당하거나, 중요한 정보를 노출 당하게 된다. 그렇다면, 이처럼 나쁜 의도를 지닌 공격자로부터 정당한 사용자들 의 정보 전송을 보호하기 위한 보안 서비스들이 있다. 다음 절에서는 나쁜 목적을 지닌 공격자들로 부터 정보를 보호하기 위한 보안 서비스에 대해 기술한다.

2.2.보안 서비스

보안 공격에 대해 제공되는 보안 서비스는 다음과 같다.[1,2,12,13,15]

- 비밀성 서비스
- 무결성 서비스
- 사용자 인증 서비스
- 부인 봉쇄 서비스

2.3.기반 암호 알고리즘들

본 절에서는 SIBS를 설계 및 구현하는데 기반이 되는 암호 알고리즘들에 대해 기술하도록 하겠다[2,12,14]. 우선은 사용자 인증을 위한 SSLeay를 사용한 CA 시스템에 관해 설명을 하고, 다음으로는 고객과 은행 사이에서 데이터의 암호화를 위해 사용할 KEY값을 분배할 수 있도록 해 주는 Diffie-Hellman 키분배 알고리즘[2,16,17]에 대해 설명하도록 하겠다. 마지막으로 나누어 가진 암호화 키를 가지고 데이터를 실제로 암호화 하는 비 대칭키 방식의 하나로 DES(Data Encryption Standard) 알고리즘이 갖는 짧은 키 길이의 문제를 해결하고 빠른 속도를 제공할 수 있는 알고리즘으로 개발된 IDEA 알고리즘에 대해 기술한다.

가.사용자 인증 시스템

고객과 은행 사이에서 거래에 대한 정보를 주고 받기 위해서는 먼저 고객과 은행 모두 상대방의 신분을 확인할 수 있어야 한다. 이를 위해 확실하게 믿을 수 있는 제 3 의 기관으로부터 인증받은 사용자 인증서가 필요하다.

사용자는 인증서를 발급 받기 위해서 인증 기관의 대리 기관인 RA에 인증서를 신청하면, RA는 지역적인 인증 기관인 CA를 대신해서 인증서 신청에 관한 처리 과정을 수행한다. 지역적인 CA 기관은 인증서를 신청한 고객의 신분에 대해 조사를 하고 정당한 사용자일 경우 인증서를 발급해 준다. 그리고, 자신보다 상위의 Root CA에 이를 통보하여 글로벌 인증 구조에 따라 인증서를 유지한다.

사용자 인증 시스템을 그림으로 나타내면 아래의 그림 2.2 와 같다.

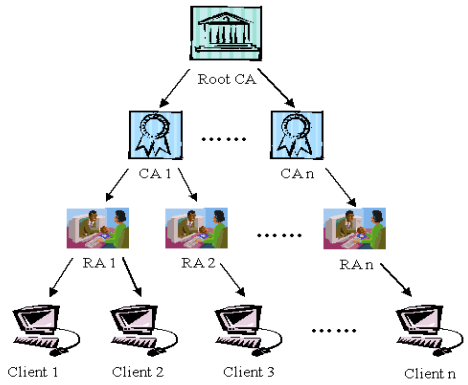


그림 2.2 인증시스템의 구조

위의 그림과 같이 인증서를 발급 받기를 원하는 다수의 클라이언트들이 존재한다. 최하위 계층의 바로 위에는 CA에 인증서를 발급받기 위해 거치는 등록 절차를 대신해서 수행해 주는 RA 기관이 존재한다. RA 기관은 사용자의 편의성을 제공하기 위해 사용자의 인증서 발급 신청을 모아서 대신 처리해 주는 역할을 한다. RA기관의 상위에는 지역 CA 기관이 존재하게 된다. CA 기관은 클라이언트의 인증서 신청을 수렴하여 인증서를 발급해 준다. 최상위 기관인 Root CA는 지역 CA들을 인증하고 발급된 인증서를 관리하는 역할을 수행한다.

CA 시스템은 확실하게 믿을 수 있는 제 3의 기관으로부터 서버측과 클라이언트측이 자신의 정보를 전송하여 정당한 사용자인지, 인증서의 사용 목적이 무엇인지를 밝히는 정보를 인증 서버로 보내게 된다. 그렇게 하면, 인증 서버는 클라이언트의 정보들을 조사하여 진위 여부를 파악하고 정당한 사용자라면 사용 목적에 맞는 한도 범위 내에서 인증서를 자유롭게 사용할 수 있도록 발급해 준다.

이렇게 발급 받은 인증서를 클라이언트는 자신의 컴퓨터에 안전하게 보관을 한 상태로 다른 서버에 접속할 때 자신의 인증서 요구를 받게 되면 확실하게 믿을 수 있는 상위 기관으로부터 발급 받은 인증서를 전송해 주게 되어 자신의 신분을 인증하게 되고, 이처럼 전송 받은 인증서를 서버 측에서는 확실한 인증 기관으로부터 클라이언트의 인증서의 진위 여부를 판정해서 자신의 시스템에 접근 가능 여부를 판단하게 된다.

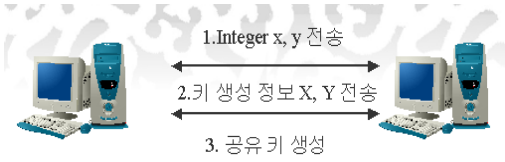
인증 시스템은 물론 처음에 인증서를 발급 받아야만 하기 때문에 불편함을 내포하고 있다. 하지만, 본 논문에서 제안하는 안전한 인터넷 뱅킹 서비스를 제공하기 위해서는 인증 시스템을 이용함으로서 보안 서비스 중 사용자 인증 서비스에 대한 확실한 서비스를 제공하게 된다. 또한 인증 시스템을 이용함으로서 인증서 발급시에 발급된 클라이언트의 비밀키를 사용하여 전송하고자 하는 메시지에 암호화를 하여 데이터를 전송하면 부가적으로 사용자의 데이터 전송 행위에 대한 부인 봉쇄 서비스도 제공받을 수 있다.

나.Diffie-Hellman의 키 분배 알고리즘

데이터를 전송하는데 있어서 비밀성 서비스를 제공하기 위해서는 평문(Plain-Text)을 암호화 키(Encryption-Key)를 사용하여 암호화 하게 된다. 이처럼 암호화 작업을 수행하기 위해서는 먼저 클라이언트와 서버 사이에서 평문을 암호화 하는데 사용하는 암호화 키에 대해 사전에 약속이 되어 있어야 한다. 본 절에서는 암호화를 위한 사전 작업으로 암호화 키를 나누어 갖기 위해 필요한 키 분배 알고리즘 중에서 Diffie-Hellman 키 분배 알고리즘에 대해 살펴본다.

Diffie-Hellman 키 분배 알고리즘은 공개

된 채널 상에서 송신자와 수신자가 암호화와 복호화를 할 때 필요한 비밀키를 생성하는 알고리즘이다. Diffie-Hellman 알고리즘은 이산대수 문제의 어려움을 바탕으로 하여 설계된 알고리즘이다. Diffie-Hellman 키 분배 알고리즘을 다음과 같이 그림으로 간략하게 나타낼 수 있다.



단계1] 데이터를 전송하고자 하는 사용자 A와 B는 임의의 정수 x, y를 선택하고 이를 자신만이 알고 있는다.

단계 2] 사용자 A는 $X = g^x \bmod n$ 을 계산한 결과값 X를 B에게(또는 공개 키들의 저장소에) 보내고, 사용자 B는 $Y = g^y \bmod n$ 을 계산한 결과값 Y를 A에게(또는 공개 키들의 저장소에) 전송한다.

단계 3] 사용자 A와 B는 각자 자신이 소유하고 있는 비밀 정보와 단계 2 과정에서 전송 받은 정보를 사용하여 다음과 같이 서로 공유할 수 있는 키 값, K와 K'을 생성한다. 이제 사용자 A와 사용자 B는 앞으로 데이터를 암호화 하여 전송할 때 이 키 값을 사용하면 된다.

그림 2.3 Diffie-Hellman의 키 분배 알고리즘

지금까지 간단하게 사용자 인증 시스템과 Diffie-Hellman 키 분배 알고리즘에 대해 살펴 보았다. 마지막으로, 실제 전송되는 데이터를 암호화 하기 위한 대칭키 암호방식인 IDEA 알고리즘을 다음 절에서 논의하도록 한다.

다.IDEA(International Data Encryption Algorithm)

데이터를 암호화하는 방법에는 크게 분류하여 볼 때 두 가지로 나누어 생각할 수 있다. 첫 번째 방법은 대칭키(Symmetric-Key) 방법이고, 두 번째 방법은 공개키(Public-Key) 방법이다.

이 중에서 본 논문에서는 대칭키 방법인 IDEA 암호 알고리즘을 적용하도록 한다. IDEA 암호 알고리즘은 DES 암호 알고리즘이 갖는 짧은 키 값으로 인한 공격을 보완하여 개발된 암호 알고리즘이다. DES 알고리즘이 데이터 암호화를 위해 56 비트 암호화 키를 사용하는데 비해 IDEA 알고리즘은 128 비트의 암호화 키를 사용한다. IDEA 알고리즘은 Block 단위로 데이터를 암호화 한다.

IDEA암호 알고리즘은 64 비트의 평문과 128 비트의 키 값을 입력값으로 사용하여 데이터를 암호화 한다. 출력값으로는 암호화한 64 비트 암호문을 생성해 낸다. DES와 마찬가지로 8 라운드 동안 반복하여 동작한다. 16 비트 서브블록 4 개를 입력값으로 하며 라운드 별로 4 개의 16 비트 결과 블록을 생성해 내는 과정을 8 라운드 동안 수행한다. 마지막 라운드를 제외하고 1 라운드부터 8 라운드까지는 4 개의 16 비트 평문 블록과 128 비트의 암호화 키 값을 16 비트씩 잘라 순서대로 6 개의 키값을 선택하여 서브키 값으로 사용한다. 한 라운드를 수행하고 나면 서브키 값은 순환하는 왼쪽 쉬프트 연산을 통해서 25 비트 쉬프트하여 다음 6 개의 서브키 값을 선택한다. 위와 같은 과정을 암호화 처리 과정을 8 라운드 수행한 후, 마지막으로 4 개의 서브키를 이용하여 데이터 변환을 수행하게 된다. 마지막으로 변환 과정을 거친 출력값이 최종적인 암호문이 된다. 복호화 과정은 위의 암호화 과정을 역순으로 수행한다.

데이터의 암호화를 위한 IDEA 의 처리 수행과정을 그림으로 표시하면 아래의 그림 2.4 와 같다.

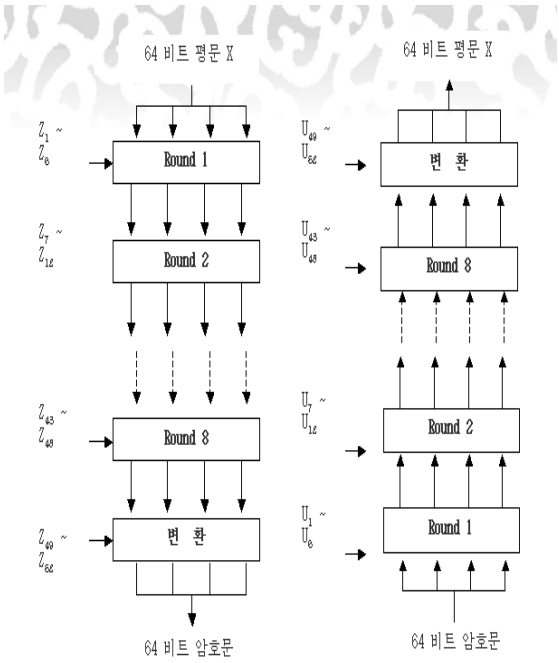


그림 2.4 IDEA의 동작과정

다음 절에서는 IDEA 암호 알고리즘의 암호 강도 및 특성, 연산 방법, 구조, 서브키 생성방법, 처리 절차, DES 암호 알고리즘과의 비교에 대해 서술한다.

IDEA 암호 알고리즘의 강도 및 특성

- 블록길이 : 블록의 길이는 통계적 분석을 예방할 수 있을 만큼 길어야 한다. 그렇지 않다면 특정 블록이 다른 블록들에 비하여 자주 나오는 현상이 발생하기 때문이다. 일반적으로 64 비트 블록 크기는 충분히 강력하다고 인식되어 사용되고 있다.

- 키 길이 : 키는 무차별공격(Brute Force Attack)에 효율적으로 대응 할 수 있을 만큼 충분히 길어야 한다. 그러므로 IDEA는 DES가 사용하는 56 비트 보다 더욱 긴 128 비트 키 길이를 사용하고 있다.

- 혼돈(Confusion) : 암호문은 평문과 추측해 내기 어려운 방법으로 생성된 키 값에 의해 암호의 강도가 결정되게 된다. 그러므로 혼돈의 성질을 이용하여 어떻게 암호문의 통계적 성격을 지니지 않고 복잡하게 만들어 내는가에 대한 방법론이다.

- 확산(Diffusion) : 각각의 평문 비트는 모든 암호문 비트에 영향을 주어야 한다. 키 또한 모든 암호문 비트에 영향을 주어야 한다. 하나의 평문 비트를 다수의 암호문 비트에 확산시켜 통계적 성질을 감춘다.

IDEA에서 Confusion을 위한 3 가지 연산

- XOR연산 : ⊕ 로 나타낸다.

- 16 비트 unsigned integer 로 취급되는 입력과 출력에 대한 정수 나머지 연산 2^{16} (mod 65536)의 덧셈, 로 표시한다.

- unsigned integer 인 16 비트 정수로 취급되는 입력과 출력에 대한 정수 나머지 연산 $2^{16} + 1$ (mod 65537) 의 곱셈, 단, 모든 비트가 '0'일때는 2^{16} 으로 나타낸다. ⊙로 나타낸다.

위의 3 가지 연산의 예외 법칙은 다음과 같다.

- 3 가지 연산의 조합에서

다음과 같이 분배법칙이 성립하지 않는다.

$$a \boxplus (b \odot c) \neq (a \boxplus b) \odot (a \boxplus c)$$

- 3 가지 연산의 조합에서 다음과 같이 결합법칙이 성립하지 않는다.

$$a \boxplus (b \oplus c) \neq (a \boxplus b) \oplus c$$

위에서 보듯이 이 3 가지 연산을 모두 조합하여 사용함으로써 XOR 연산만을 사용하는 DES보다 암호 해독이 더 어려워지게 된다.

IDEA의 기본적인 덧셈/곱셈 구조

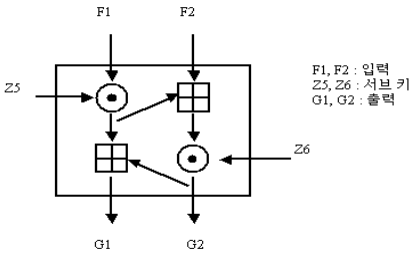


그림 2.5 IDEA의 연산구조

IDEA는 위의 그림 2.5 와 같이 2 개의 덧셈과 2 개의 곱셈 연산을 한 라운드에 수행하게 된다. 블록 단위 연산으로 한 연산으로부터 생성된 결과값이 다시 다른 연산의 입력값이 되는 형태로 연산은 수행되게 되며 2 개의 입력값과 2 개의 서브키를 사용하여 2 개의 결과값을 생성해 낸다. 즉, 2 개의 평문 16 비트 블록과 2 개의 16 비트 서브 키를 입력으로 받아 2 번의 덧셈과 곱셈 연산을 순차적으로 수행해서 2 개의 암호문을 생성해 낸다.

서브 키 생성

IDEA에서는 처음에 주어진 128 비트의 암호키를 사용하여 16 비트씩 구성된 52 개의 서브키를 생성해 낸다. 생성 방식은 Z1, Z2, ..., Z6 로 명명되는 처음 8 개의 서브 키들은 Z1 은 최상위 16 비트이다. Z2 는 다음의 16 비트 키 값을 선택하게 된다. 이와 같은 과정을 차례로 Z8 까지 서브키를 설정하고, 다음부터는 순환 왼쪽 쉬프트 연산을 25 비트 수행한 후 처음과 같이 적용하여 그 다음의 Z7, Z8, ... Z12 의 서브키를 생성해 낸다. 52 개의 서브키 값이 생성될 때까지 이와 같은 과정을 반복하여 수행한다.

반대로 복호화 키는 U1, U2, ..., U6 으로 명명되며 곱셈의 역원과 덧셈의 역원을 이용하여 생성해 낸다. 곱셈의 역원은 Z_j^{-1} 로 나타내며 다음식에 의해 구한다.

$$Z_j \odot Z_j^{-1} = 1$$

덧셈의 역원은 $-Z_j$로 나타내며 다음식에 의해 구한다.

$$-Z_j \boxplus Z_j = 0$$

위와 같이 덧셈의 역원과 곱셈의 역원을 구해 복호화 키를 생성해 낸다.

지금까지 살펴 본 IDEA 암호 알고리즘에서 사용하는 암호화 키 값과 복호화 키 값을 계산하여 표로 나타내면 표 1 과 같다.

표 1 암호화/복호화 키 값 테이블

	암호화		복호화	
상태	선정	Bit 위치	선정	같은 값
1 R	$Z_1 \sim Z_6$	1..96	$U_1 - U_6$	Z_{49}^{-1}, $-Z_{50}$, $-Z_{51}$, Z_{52}^{-1}, Z_{47}, Z_{48}
2 R	$Z_7 \sim Z_{12}$	97..128, 26..89	$U_7 - U_{12}$	Z_{43}^{-1}, $-Z_{45}$, $-Z_{44}$, Z_{46}^{-1}, Z_{41}, Z_{42}
3 R	$Z_{13} \sim Z_{18}$	90..128, 1..25, 51..82	$U_{13} - U_{18}$	Z_{37}^{-1}, $-Z_{39}$, $-Z_{38}$, Z_{40}^{-1}, Z_{35}, Z_{36}
4 R	$Z_{19} \sim Z_{24}$	83..128, 1..50	$U_{19} - U_{24}$	Z_{31}^{-1}, $-Z_{33}$, $-Z_{32}$, Z_{34}^{-1}, Z_{29}, Z_{30}
5 R	$Z_{25} \sim Z_{30}$	76..128, 1..43	$U_{25} - U_{30}$	Z_{25}^{-1}, $-Z_{27}$, $-Z_{26}$, Z_{28}^{-1}, Z_{23}, Z_{24}
6 R	$Z_{31} \sim Z_{36}$	44..75, 101..128, 1..36	$U_{31} - U_{36}$	Z_{19}^{-1}, $-Z_{20}$, $-Z_{21}$, Z_{22}^{-1}, Z_{17}, Z_{18}
7 R	$Z_{37} \sim Z_{42}$	37..100, 126..128, 1..29	$U_{37} - U_{42}$	Z_{13}^{-1}, $-Z_{15}$, $-Z_{14}$, Z_{16}^{-1}, Z_{11}, Z_{12}
8 R	$Z_{43} \sim Z_{48}$	30..125	$U_{43} - U_{48}$	Z_7^{-1}, $-Z_9$, $-Z_8$, Z_{10}^{-1}, Z_5, Z_6
변환	$Z_{49} \sim Z_{52}$	23..86	$U_{49} - U_{52}$	Z_1^{-1}, $-Z_2$, $-Z_3$, Z_4^{-1}

IDEA는 DES의 암호화에 사용하는 키 길이가 짧아서 발생하는 보안상의 약점을 보완하여 암호화 키 길이가 길어졌으며 암호화 처리 속도가 빨라 졌으며, 연산의 복잡도 또한 높아져서 불법적으로 복호화 하기가 더욱 어려워졌다. 그렇다면 기존의 DES 암호 알고리즘에 비해 IDEA암호 알고리즘이 얼마나 향상되었는지 비교해 보면 표 2 와 같다.

표 2 DES와 IDEA의 비교

	DES	IDEA
키 길이	56 bit	128 bit
암호문 Type	block	block
키 운용 방식	대칭키 (symmetric key)	대칭키 (symmetric key)
sub-block 크기	32 bit	16 bit
sub-block 갯수	2	4
연산 반복 횟수	16	8
sub-key 갯수	16	52
혼돈 연산	XOR연산 S-Box에 의한 전치, 대치, 확장 연산을 수행	XOR 연산 덧셈의 법 (⊞) 연산 곱셈의 법 (⊙) 연산

2.4.인터넷 뱅킹 시스템의 이해

본 절에서는 SIBS를 제안하기 위해서 인터넷 뱅킹 시스템에 대해 살펴 본 후, 일반적인 인터넷 뱅킹 시스템을 좀 더 인터넷 환경에서 안전하게 설계하기 위해 필요로 하는 요구 사항들을 분석하도록 한다.

라.인터넷 뱅킹 시스템의 개요

인터넷 뱅킹 시스템이란, 전 세계적으로 사용하고 있는 인터넷을 매체로 사용하여 가정이나 사무실에서 직접 은행 업무를 처리할 수 있는 시스템을 말한다. 즉, 은행에 직접 고객이 왕래하여 처리하던 잔액조회, 계좌이체, 입출 거래 명세확인, 금융 업무 사고 처리, 환율 조회 등과 같은 은행 업무들을 고객이 은행까지 직접 찾아가지 않고도 가정이나 사무실에서 인터넷이라는 전 세계적으로 구성된 글로벌 네트워크와 PC(Personal Computer)를 사용하여 처리할 수 있는 시스템을 말하는 것이다.

인터넷 뱅킹 시스템에 대한 개념적인 형

태를 그림 2.6 과 같이 나타낼 수 있다.

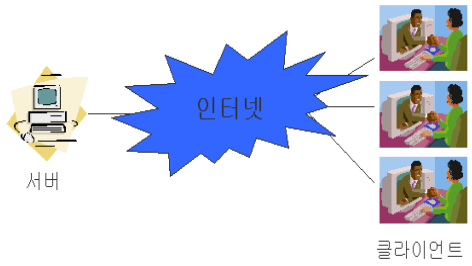


그림 2.6 인터넷 뱅킹 시스템

현재 인터넷 뱅킹 시스템은 많은 사람들이 관심을 가지고 사용하기를 원하고 있다. 또한 은행의 관점에서도 좀 더 다양하고 편리한 서비스를 제공하기 위해서 인터넷 뱅킹 시스템은 꼭 제공해야 하는 서비스이다. 기존에는 점외 CD기계에서나, ATM기계를 통하여, 또는 ARS를 통해 처리하던 다양한 은행업무를 이제는 은행의 고객들이 쉽게 인터넷 뱅킹을 통해 해결 하기를 원한다는 것이다. 그러므로, 은행 또한 인터넷 뱅킹 시스템에 많은 관심을 가지고 서비스를 제공하기 위해서 최선을 다하고 있다.

인터넷 뱅킹 시스템이 갑자기 사람들의 관심을 끌게 된 것은 아니다. 인터넷 뱅킹 시스템이 나오기 이전에 이미 은행의 고객들은 폰 뱅킹 서비스, PC를 사용한 홈 뱅킹 서비스를 이용하였다. 폰 뱅킹 서비스는 전화를 사용하여 ARS 서비스와 같이 전화 안내에 따라서 필요로 하는 정보를 전화로 입력하여 필요한 정보를 얻는 것으로 전화를 사용하여 정보를 입력하고 음성으로 필요로 하는 데이터를 받게 되기 때문에 제약 사항이 많고, 중간에 도청 당할 위험성이 크다는 단점을 지니고 있다. PC를 이용한 홈 뱅킹 서비스도 공중망인 전화선과 PC를 사용하여 은행업무를 수행하는 방법으로 공중망인 전화선망을 사용함으로서 약한 보안 서비스로 인해 보안 위험성이 크다. 이러한 보안상의 위험성과 은행 고객의 편의성을 만족 시키기 위해 현재 인터넷 뱅킹 시스템이 개발중이거나 또는, 보안상으로 완전하지 않은 형태로 개발된 후 사용되고 있는 것이다.

마.인터넷 뱅킹 시스템의 요구사항 분석

본 절에서는 이처럼 고객에게 편의성과 보안상의 안전성을 제공하기 위한 인터넷 뱅킹 서비스를 제공하기 위해서 필요로 하는 사항들을 살펴보고 분석한다.

인터넷 뱅킹 시스템은 기존에 사용되던 과거의 다른 시스템들과 마찬가지로 사용자가 직접 은행에 찾아가지 않고서 집이나 사무실에서 은행업무를 해결하고자 하는 욕구를 만족시키기 위해 만들어진 서비스이다. 그러므로 사용자 편의성을 최대한 고려하여야 할 것이다.

인터넷상에서 은행 업무를 실제 은행에 찾아가서 은행업무를 처리할 때와 똑같은 서비스를 제공하면서 강력한 보안 서비스를 제공함으로서 고객에게 신뢰성을 제공하는 것이 인터넷 뱅킹 시스템의 최대 요구사항이다. 만약, 보안 서비스가 빈약하다면 고객의 다른 다수의 요구 조건들을 만족 시킨다 하더라도 고객은 아마 이 시스템을 이용하지 않을 것이다.

인터넷 뱅킹 시스템에 보안 서비스를 제공하기 위해서 필요로 하는 요구사항들은 다음과 같다.

- 고객의 신분 정보에 대한 확실성 보장 – 사용자 인증

- 은행과 고객 사이에서 전송되는 정보에 대한 비밀성 보장
- 은행과 고객 사이에서 전송되는 정보에 대한 무결성 보장
- 처리 업무에 대한 부인 봉쇄

위의 요구사항들을 만족시킬 수 있는 인터넷 뱅킹 시스템만이 고객을 만족시킬 수 있을 것이다. 물론 위의 요구사항 이외에도 더욱 다양하고 편리함을 제공할 수 있는 요구 사항들이 많겠지만, 이외의 다양한 모든 요구사항들은 중요도나 학문적인 가치도 보다는 고객의 편의성 제공에 초점을 맞추고 있기 때문에 본 논문에서 해결하고자 하는 과제는 아니다.

그러므로, 본 논문에서는 보안상의 기본이 되는 위의 요구사항들에 대해서만 언급하도록 하겠다.

지금까지 안전한 인터넷 뱅킹 시스템인 SIBS를 설계 및 구현하기 위해서 기초가 되는 기반지식들과 인터넷 뱅킹 시스템의 개요에 대해 살펴 보았다. 본 장에서는 지금까지 살펴 본 기반 지식들과 인터넷 뱅킹 시스템의 정의, 고객의 요구사항들에 따라 이를 만족시킬 수 있는 안전한 인터넷 뱅킹 시스템인 SIBS를 설계한다.

3.SIBS 설계

3.1.SIBS 전체구조

우선 안전한 인터넷 뱅킹 시스템인 SIBS의 전체적인 설계에 대해 서술하도록 하겠다. SIBS의 전체적인 구조를 살펴 보면, 인터넷 환경에서 웹 서비스를 이용하여 사용자에게 좀 더 그래픽적이고 편리성을 제공할 수 있는 사용자 인터페이스 부분과 하부에서 보안 서비스를 제공하기 위해서 독립적인 프로그램 형태로 구성된 암호 프로그램 부분으로 구성된다. 하부의 응용 프로그램 부분은 암호 부분, 키 분배 부분, 인증서 발급 및 관리에 관한 부분으로 세분화되어 구성된다.

사용자 인터페이스 부분은 웹 브라우저를 사용하여 좀 더 그래픽적이고 사용자에게 친숙한 형태로 설계하도록 노력하였다. 응용 프로그램 모듈은 사용자가 시스템의 전반적인 사항이나 운영에 관한 것들에 대해 신경쓰지 않고도 원활하게 사용할 수 있도록 설계하였다. 단지 사용자는 응용 프로그램 모듈을 초기에 다운로드 받은 후에 자동 설치를 수행하기만 하면 되는 것이다.

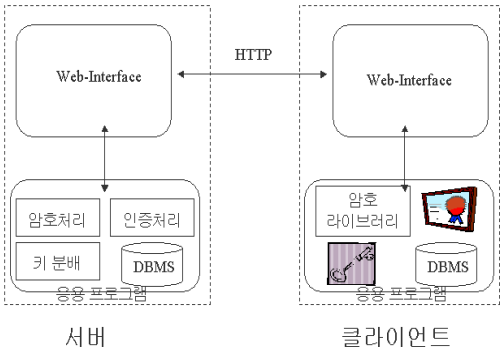


그림 3.1 SIBS 전체 구성도

응용 프로그램 모듈에 대해 자세하게 살펴 보면 크게 3 가지 모듈로 구성된다. 하나는 사용자 인증을 위한 CA 시스템이다. CA 시스템 부분은 UNIX 시스템에 SSL을 설치하여 인증서를 발급해 주는 형태로 구성하도록 하였다. CA로 부터 발급된 인증서는 사용자의 컴퓨터상에 파일로 보관되어 진다.

두번째는 암호화 처리 모듈이다. 암호화 처리 모듈은 기반지식에서 살펴 본 IDEA 암호 알고리즘을 라이브러리 형태로 제공받아 응용 프로그램으로 구현하도록 한다. 세번째는 키 분배 모듈이다. 키 분배 모듈도 기반지식에서 살펴 본 Diffie-Hellman의 키 분배 알고리즘은 라이브러리를 응용 프로그램 형태로 구현하도록 한다. 이처럼 3 개의 모듈이 각각 독립적인 모듈로 구성된 후 하나의 응용 프로그램 형태로 구현하도록 한다. 각 모듈의 관계는 아래의 그림 3.2 와 같다.

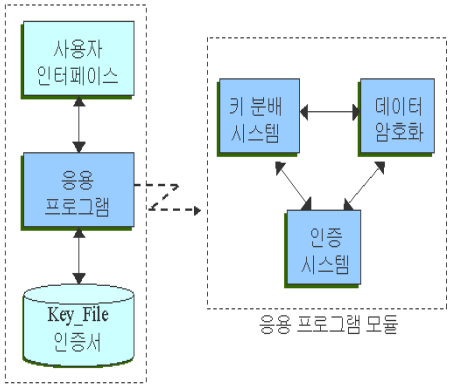


그림 3.2 SIBS 응용 프로그램 모듈의 구성

지금까지 SIBS의 전체적인 설계에 대해 살펴 보았다. 각각의 독립된 형태의 모듈로 구성된 SIBS에서 핵심이 되는 응용 프로그램 모듈의 설계에 대해서 좀 더 구체적으로 살펴 볼 필요가 있다. 그러므로, 다음 절에서는 응용 프로그램 모듈의 내부 구조에 대한 상세적인 설계 방안에 대해 서술하도록 하겠다.

3.2.모듈별 상세 설계

SIBS 는 크게 2 개의 부분으로 나뉘어 구성된다. 그 중에서 사용자 인터페이스 부분은 Windowns 환경에 친숙한 사용자를 위해 WindowsNT서버의 IIS(Internet Information Server)를 웹 서버로 사용하여 구현할 수 있다. 클라이언트 사용자는 IE(Internet Explorer) 웹 브라우저를 통해 서버에 접속하여 자유로이 SIBS를 사용할 수 있다. 사용자 인터페이스 부분은 고객의 접속에 편의성을 제공하기 위해서 웹 서비스를 제공하는 점이 본 시스템의 특징이 된다.

두 번째 모듈인 응용 프로그램 모듈은 하부에서 사용자 인터페이스를 통해 입력되거나 수신된 데이터를 처리하는 모듈이다. 응용 프로그램 모듈은 다시 3 개의 하부 모듈로 구성된다. 암호 처리 모듈, 키 분배 모듈, 사용자 인증 모듈로 구성된다. 본 절에서는 하부의 응용 프로그램 모듈 3 개 모듈에 대해 자세히 살펴 본다.

바.암호 처리 모듈

데이터 암호 처리 모듈은 고객과 은행 사이에서 또는 은행과 다른 금융기관 사이에서 주고 받는 데이터를 암호화 하는 모듈이다. 기반 지식에서 살펴 본 IDEA 암호 알고리즘을 라이브러리화 되어 제공되는 암호 알고리즘을 사용하여 금융기관과 고객 사이에서 송.수신 하는 데이터를 암호화 하여 중간에서 가로채기 공격을 해도 전송되는 데이터 자체를 지킬 수 있게 해 주는 보안 서비스이다.

En : 데이터 암호화 **De** : 데이터 복호화
Pub : 공개키 **Pri** : 비밀키
DH : **Diffie-Hellman** 방식으로 나눈 공유키
S_k : 일회용으로 사용하는 **Session key**

$$En\ pub\_a(En\ dh(s\_k)\ ||\ En\ s\_k(M))$$

그림 3.3 데이터 암호처리 모듈

우선, 데이터를 암호화 하기 위해 사용하는 용어와 연산 과정을 위와 같이 표현할 수 있다. 암호화 처리 과정은 아래의 그림과 같다. 전송하고자 하는 데이터를 일회용 세션키를 생성하여 암호화 한 값과 키 분배 알고리즘에 의해 나누어 가진 공유 키를 사용하여 암호화에 사용한 일회용 키 값을 암호화 한다. 마지막으로 데이터를 수신하는 상대방의 공개키를 사용하여 데이터 꾸러미를 암호화 한 후에 전송하게 된다. 복호화는 위와 역순으로 수행하면 된다.

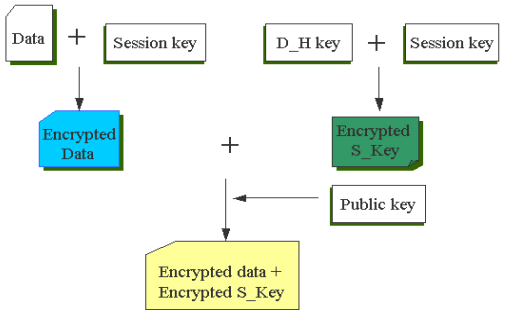


그림 3.4 데이터 암호화 처리 과정

사.사용자 인증 모듈

실제 은행에서 고객이 은행업무를 처리하기 위해서는 사전에 반드시 신분을 확인하는 과정을 거쳐야만 한다. 하지만, 이러한 현실 세계의 사용자 인증 과정을 가상의 공간인 인터넷 상에서 적용하기란 쉽지가 않다.

그러므로, 본 논문에서는 SSL을 이용한 UNIX 서버상에서 CA 시스템을 설치하여 은행의 영역 내에서만 인증서를 발급하고, 관리하며, 처리하는 것을 제한 사항으로 두고 설계하도록 하였다.

그림 4.1 인증서 발급 및 사용자 인증 과정

고객은 은행과 거래를 위해 먼저 가상의 공간인 인터넷 상에서 자신의 신분을 보장하기 위해 인증서를 발급 받기 위해 신청한다. 인증 서버는 인증서 발급에 앞서 사용자의 신분 확인 절차를 거친다. 신분 확인 절차에서 정당한 사용자임이 밝혀지면 인증서버로부터 인증서를 생성하게 된다. 생성된 인증서는 클라이언트에 파일로 전송되어 보관되어 진다.

발급된 인증서는 해당 서버와 연결시 필요에 의해 서버가 클라이언트의 인증서를 확인 하기를 요청하면, 클라이언트에 저장되어 있는 인증서를 서버로 전송함으로서 자신의 신분 인증을 거친다. 사용자 인증을 마친 후에는 연결이 끊어질 때까지는 사용자의 신분을 인정하게 된다. 다시 새로운 연결이 발생하면 사용자 인증 과정을 다시 거치게 됨으로써 매번 연결마다 사용자의 신분을 인증하여 데이터 송.수신이 이루어지게 된다.

4.SIBS 구현 및 고찰

본 장에서는 안전한 인터넷 뱅킹 시스템인 SIBS의 실제 구현 형태와 시스템의 특징에 대해 서술하고, 또한 본 시스템의 제약

사항들에 대해 고찰한다.

SIBS를 구현하는데 있어 안전성을 제공하기 위해서 운영체제는 마이크로소프트사의 Windows NT 를 사용하였고, 사용 언어는 VC++6.0 을 사용하였다. 또한 암호 라이브러리는 Crypto++ 3.1 라이브러리를 사용하였다.

표 3 개발 환경 및 언어

개발환경	Windows NT4.0
구현언어	VC++, ASP
암호 라이브러리	Crypto3.1
특징	사용자 인터페이스를 위해 GUI 운영체제 적용한 웹 서비스 제공 사용자 인증을 위한 SSL내의 CA 시스템을 이용한 인증서 발급 데이터 암호화에 공개키 방식에 비해 속도가 빠른 대칭키 암호 방식 적용

이러한 개발 환경하에 사용자 인터페이스 부분과 서버 부분, 응용 프로그램 부분으로 나누어 SIBS를 구현했다.

4.1.사용자 인터페이스 구현

사용자 인터페이스는 편의성을 최대한 고려해 웹 브라우저를 통해 그래픽 인터페이스로 접근할 수 있도록 구현하였다. Windwos 환경에 익숙한 사용자들을 위해서 Windows NT서버에 IIS 웹서버를 이용하여 사용자 인터페이스를 제공한다.

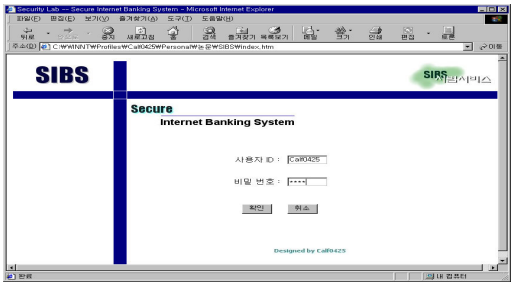


그림 4.2 SIBS 초기 접속화면

사용자는 위의 그림 4.1 과 같이 초기 인터넷 접속을 통해 편리하게 안전한 인터넷 뱅킹 시스템에 접속하게 된다. 웹 브라우저를 통해 SIBS에 접속하여 로그인 과정을 통해 사용자 신분 확인과정을 거치게 된다. 만약, 처음 접속하는 사용자라면 다음 과정에서 인증서 발급을 받도록 할 것이다.

사용자 인터페이스의 제약 사항으로는 ActiveX기법을 사용하여 하부의 응용 프로그램과 연계되기 때문에 IE 4.0 이상을 사용해야 하는 제약 조건을 갖는다. Netscape 웹 브라우저가 ActiveX기법을 제공하지 않기 때문에 발생하는 제약 사항이다.

4.2.응용 프로그램 구현

본 절에서는 SIBS의 핵심이 되는 응용 프로그램 모듈에 대해 서술하도록 한다. 응용 프로그램은 3 개의 모듈로 나뉘어 구현되어진다. 물론, 3 개의 모듈은 1 개의 완성된 응용 프로그램으로 구성된다. 각각의 모듈은 공개된 암호 라이브러리를 이용하여 구현하였다.

아.암호 처리와 키 분배 알고리즘

암호 처리 모듈은 암호 라이브러리인 Crypto3.1 이 제공하는 많은 암호화 라이브러리 중에서 빠른 데이터 암호화를 위해

IDEA 알고리즘을 이용하여 구현한다.

암호화 처리 과정과 키 분배 처리 과정은 라이브러리인 DLL(Dynamic Linked Library) 형태로 구현되어지기 때문에 실제로 사용자가 볼 수 없다. 단지 DLL형태의 파일을 ActiveX 기법을 사용하여 웹 상에서 사용할 수 있게 한다.

표 4 IDEA 암호화 함수 예제

```
void IDEA::EnKey (const byte *userKey)
{
   int i, j;
   word *Z=key;

   for (j=0;j<8;j++)
          Z[j] = (userKey[2*j]<<8) + userKey[2*j+1];
   for (i=0;j<IDEA_KEYLEN;j++)
   {
          i++;
          Z[i+7]=low16((Z[i&7] << 9) | (Z[i+1 & 7] >> 7));
          Z+=i&8;
          i&=7;
   }
}



void IDEA::DeKey()
{
    word *Z=key;
    int j;
    word t1,t2,t3;
    SecBlock<word> tempKey(IDEA_KEYLEN);
    word *p=tempKey+IDEA_KEYLEN;
    t1=inv(*Z++);
    t2=low16(0-*Z++);
    t3=low16(0-*Z++);
    *--p=inv(*Z++);
    *--p=t3;
    *--p=t2;
    *--p=t1;
    for (j=1;j<ROUNDS;j++)
    {
           t1=*Z++;
           *--p=*Z++;
           *--p=t1;
           t1=inv(*Z++);
           t2=low16(0-*Z++);
           t3=low16(0-*Z++);
           *--p=inv(*Z++);
           *--p=t2;
           *--p=t3;
           *--p=t1;
    }
    t1=*Z++;
    *--p=*Z++;
    *--p=t1;
    t1=inv(*Z++);
    t2=low16(0-*Z++);
    t3=low16(0-*Z++);
    *--p=inv(*Z++);
    *--p=t3;
    *--p=t2;
    *--p=t1;
    /*copy and destroy temp copy*/
    memcpy(key, tempKey, IDEA_KEYLEN*sizeof(word));
}
```

사용자는 입력폼에 필요한 데이터를 입력하게 되면 입력된 사용자 ID와 패스워드로 정당한 사용자인지 확인한 후에 데이터 암호화 과정을 처리하게 된다.

암호화를 위한 IDEA 함수는 라이브러리화 되어 있기 때문에 사용자가 볼 수 없다. 하지만 위와 같이 구현된 암호 함수들을 라이브러리로 만든 후 ActiveX 제어기로 연결하면 사용자는 하부에서 일어나는 암호화 과정에 대해서는 신경쓰지 않아도 된다. 다음과 같이 HTML 문서에 파라미터로 객체의 ID를 넘겨주면 암호화 처리가 이루어지게 된다.

표 5 ActiveX 제어기 구현

```
<HTML>
<HEAD>
    <TITLE>SIBS -- 암호처리 모듈</TITLE>
</HEAD>
<BODY>
<OBJECT ID="SIBS_CLIENT.SibsClientCtrl.1"
     CLASSID="CLSID:255B07E2-1871-B958-ER457E3-009238EA0984">
    <PARAM NAME="SessionID" VALUE="1">
    <PARAM NAME="ServerIPAdd" VALUE="Securtiy.paichai.ac.kr">
</OBJECT>
</BODY>
```

사용자 인증 알고리즘

SIBS 를 사용하기 위해서는 제일 먼저 사용자 인증을 위한 과정을 거쳐야 할 것이다. 안전한 인터넷 뱅킹 서비스를 제공하기 위해서는 선행되어야 하는 과정이다.

먼저 사용자 인증을 위해서 사용자 인증서를 신청하게 된다. 사용자는 웹 상에서 사용자 인증서를 생성하기 위해서 입력 양식에 맞추어 필요한 데이터를 입력하게 된다.

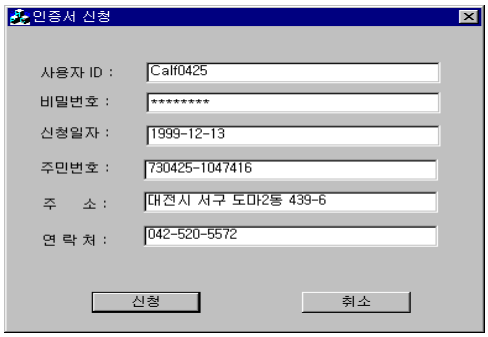


그림 4.3 인증서 신청 폼

입력된 데이터는 암호화 되어 서버에 전송되게 되고, 전송된 입력 데이터를 가지고 인증 서버는 사용자의 인증서를 생성하게 된다. 사용자가 정당하다면, 인증 서버는 IETF 표준인 X.509 포멧 파일 형식으로 인증서를 생성하여 사용자에게 전송한다.

4.3.고찰

본 절에서는 안전한 인터넷 뱅킹 서비스를 제공하기 위해 제안한 SIBS의 특징에 대해 서술한다. 그리고, SIBS와 유사한 기존의 인터넷 뱅킹 서비스를 제공하는 타 시스템과의 비교를 통해 장.단점을 비교해 보도록 한다. 마지막으로 SIBS 의 응용 분야에 대해 고찰하도록 한다.

본 논문에서는 안전한 인터넷 뱅킹 서비스를 제공하기 위해서 빠른 데이터 암호화 알고리즘으로 IDEA를 적용하였다. 그리고, 고객의 인증을 위해 X.509 의 포멧에 따른 사용자 인증서를 발급하여 사용자 인증을 제공하도록 하였다. 또한, 데이터 암호화에 사용하기 위한 암호화 키의 보안을 위해 키 분배 알고리즘으로 Diffie_Hellman의 키분배 알고리즘을 적용하였다.

이렇게 함으로써 빠른 데이터 암호처리 속도와 고객의 확실한 인증 서비스를 제공할 수 있으며, 키 분배에 대한 보안성도 제공할 수 있다는 장점을 지니게 됐다. 하지만, 인증 서비스를 제공하는 SSLeay의 CA 시스템을 구성함에 있어 SIBS에서 사용하기 위한 소규모의 인증 시스템을 구현함으로써 기타의 용도로 범용적이게 사용하는데 대한 제약 사항을 지니고 있다. 사용자에게 GUI 환경의 웹 브라우저를 통해 SIBS에 접속하게 하고 보안서비스를 제공하는 응용 프로그램을 하부에서 구동하도록 설계 및 구현함으로 인해 이를 연계 시키기 위한 방법으로 ActiveX 기법을 사용하였는데, ActiveX 기법이 자체적으로 보안상에 있어서 약간의 문제점을 내포하고 있다는 단점을 지니기도 한다.

표 6 SIBS와 타 시스템들의 비교

	SIBS	C 시스템	H시스템
알고리즘	IDEA, Diffie_Hellman	SSL	SSL
키 길이	128	128	128
인증방식	인증서	인증서	인증서
사용방식	웹 방식	보안프로그램 웹 방식	전자지갑 웹 방식
특징	IDEA를 사용하여 빠른 암호화 인증서를 사용한 확실한 신분인증 웹 브라우저를 통한 쉬운 접근	기타 보안 프로그램 다운로드 필요	기타 보안 프로그램 다운로드 필요

이러한 특징을 가지는 SIBS는 인터넷 뱅킹 시스템에 보안 서비스의 제공, 사용자 인증 시스템을 통한 신분 인증과 같은 분야에 사용 할 수 있다. 또한 좀 더 확대해서

활용한다면, 인터넷에서의 데이터 전송에 보안 서비스를 필요로 하는 분야에 적용할 수 있을 것이다.

5.결 론(Conclusion)

본 논문은 인터넷 사회라고 불리는 현대 사회에서 고객이 직접 은행에 왕래하지 않고 은행 업무를 가정이나 사무실에서 처리할 수 있도록 해 주는 인터넷 뱅킹 시스템을 좀 더 안전하고 편의성을 제공할 수 있는 SIBS 시스템에 대한 것이다.

SIBS는 보안상으로 128 비트의 암호키 값을 사용하여 암호 강도를 높였다. 또한 사용자 인증 시스템을 구성하여 인증서를 발급해 확실한 사용자 인증 서비스를 제공한다. 또한 Diffie-Hellman 키 분배 알고리즘을 이용해 데이터 암호화에 사용하는 세션 키를 암호화 하여 전송하도록 구성하여 보안 강도를 더욱 높였다.

ActiveX기법을 사용하여 공개된 암호 라이브러리를 도입해 설계 및 구현한 응용 프로그램 부분을 웹 브라우저 상에서 고객이 사용하기에 편리하도록 최대한의 편의성을 제공할 수 있다.

하지만, SIBS 도 현실적으로 몇 가지 단점을 가지고 있다. Diffie-Hellman 키 분배 알고리즘을 사용함으로 인해 Man-in-the-Middle-Attack에 약점을 내포하고 있다. 또한 인증 시스템이 은행 업무를 처리하는 데 사용하는 부분까지만 책임을 지기 때문에 자체 인증 시스템을 사용함으로 인해서 발생할 수 있는 보안상의 취약점을 지닐 수 있다.

SIBS의 활용분야로는 뱅킹 업무만이 아닌 주식 및 부동산 등과 같은 금전적인 목적물을 인터넷상에서 매매할 때에도 보안상의 서비스를 제공하기 위해서 사용 가능하다. 기타의 보안상의 데이터 전송을 위한 응용분야에 활용하기 위해서 암호처리 알고리즘과 신분 인증알고리즘을 적용하여 응용한다면 많은 웹 브라우저 응용 어플리케이션으로 활용 가능하다.

향후에 SIBS를 보완하여 좀 더 다양한 암호 처리 알고리즘을 제공하고, ActiveX 가지니는 자체적인 보안상의 문제점을 해결하여 안전하고 빠른 인터넷 뱅킹 시스템이 될 수 있도록 향후 연구한다. 사용자 인증을 위한 인증 시스템을 글로벌 인터넷 개념에서 제공한다면 하나의 인증서를 사용하여 인터넷 뱅킹만이 아닌 타 금융업무에도 사용 가능하다. 마지막으로, 현재 파일이나 문자열의 데이터 전송에만 암호 처리 알고리즘을 적용하는 점을 보완해서 멀티미디어 데이터 전송에도 가능하도록 확장할 수 있을 것이다.

참고문헌(Reference)

[1] “암호학의 이해”, 김철, 영풍문고, 1996

[2] “인터넷 보안 메커니즘에 관한 연구”, 조인준, 정회경, 김동규, 통신보호학회 학술지 1998.6.

[3] “웹 환경 구축 및 운영을 위한 보안 관리 지침서(안)”, 한국 전산원, 1997.12.

[4] “인터넷 환경에서 암호 라이브러리를 이용한 전자결재 시스템의 설계 및

구현", 황의주, 배재대학교 석사학위 논문, 1999.6.

[5] "SSL 기반 전자 상거래 지불 프로토콜 설계", 채송화, 아주대학교 석사학위논문, 1999.7.

[6] "일회용 세션키를 이용한 인터넷 홈뱅킹 프로토콜" 민덕기, 이용환, 건국대학교 석사학위논문, 1997.

[7] "인터넷을 이용한 홈뱅킹 시스템의 보안에 관한 연구", 홍석종, 건국대학교 석사학위논문, 1997.

[8] "웹 보안을 위한 사용자 인증과 암호화 통신 구현", 송상헌, 박정수, 강신각, 김재명, 안은미, 류재철

[9] 특집(1) 국내 사이버 금융현황, 데이터베이스 월드, 1999 년 10 월호, http://www.dpc.or.kr/dbworld/document/9910/spec-1.html

[10] 특집(2) 사이버 금융의 미래, 데이터베이스 월드, 1999 년 10 월호, http://www.dpc.or.kr/dbworld/document/9910/spec-2.html

[11] 특집(3) 사이버 금융 시스템 구축방안, 데이터베이스 월드, 1999 년 10 월호, http://www.dpc.or.kr/dbworld/document/9910/spec-3.html

[12] "Applied Cryptography, Second Edition",Bruce Schneier, John Wiley & Sons, 1996

[13] "Cryptography and Network Security, Second Edition", Willian Stallings, Prentice-Hall, 1999

[14] "Security in Computing, Second Edition", Charles P. Pfleeger, Prentice-Hall, 1997

[15] "Web Security: a step-by-step reference guide",Stein D. Lincoln, Addison Wesley Longman, 1997

[16] Crypto++ 3.1 Crypto Library, http://www.spinnaker.com/crypt/libraries

[17] Crypto++3.1 Crypto Library, http://www.eskimo.com/~weidai/cryptlib.html

[18] X.509, RFC2459. http://www.ietf.org/RFC/rfc2459.txt