Journal of Digital Contents Society Vol. 9 No. 1 March. 2008(pp. 125-130)

웹 어플리케이션 콘텐츠 보호를 위한 분산 보안

허진경*

요 약

인터넷 기술의 발전으로 웹을 통해서 이루어지는 사용자 서비스가 증가하고 있다. 또한 암호화 기술의 발전과 함께 네트워크를 통해 전달되는 암호화 데이터의 양이 증가하고 있다. 이로 인해 웹 어플리케이션 시스템에서 사용자에게 보내지는 인터넷에서의 콘텐츠에 대한 신뢰성 및 불법적인 복재에 대한 문제가 발생하게 된다. 네트워크를 통해 전달되는 데이터를 제3자가 가로채도 그 내용을 알 수 없게 하는 암호화 기술, 현재 사용자가 올바른 사용자인지를 구분하는 인증기술, 원본 데이터와 복제된 데이터를 구분할 수 있게 하는 디지털 서명 등의 기술은 웹상에서 콘텐츠 보호를 위해 사용되는 기술들이다. 웹 어플리케이션 시스템에서 발생할 수 있는 보안상 취약점을 해결하기 위한 방법으로 전달되는 메시지의 암호화와 공개키를 관리가 필요하다. 본 논문은 웹 어플리케이션 시스템에서 데이터의 기밀성 및 사용자 인증을 제공함과 동시에, 다수의 클라이언트 접속시 암호화 서버의 병목현상으로 인한 성능저하를 방지하고 서비스 질을 향상시키기 위한 방법으로 분산 보안 시스템을 제안한다.

Distributed Security for Web Application Contents Protection

Jin-Kyoung Heo*

Abstract

User web service is increasing by development of internet technology. Quantity of encrypted data that transmitted through the network are increasing by development of encipherment technology. We have many problems; it is caused by technical development and service increase of user requests. It is like that, we have reliability of contents and illegality copy problem of internet contents in web application system. It is contents protection skills in web that encipherment technology, authentication and digital signature. We need message encoding and secret key for solve vulnerability of encipherment in web application system. In this paper, we propose a distributed secure system that can data confidentiality and user authentication. It prevent performance degradation from bottle neck in encipherment server, and improve service quality.

Keywords : 암호화(Encipherment), 인증(Authentication), 디지털서명(Digital Signature)

1. 서론

인터넷상에서 데이터를 안전하게 전송하기 위해서는 데이터가 제3자에게 악의적인 방법으로 노출되더라도 그 내용을 알지 못하도록 하는 방법이 필요하다. 그러한 방법으로 암호화 기법이 사용된다. 평문을 전송하는 것은 내용이 공개되는 위험에 노출되기 때문에 아주 위험하다. 그러므로 암호화된 문서를 전송하면 보다 안전하다. 송신자는 평문을 암호화키로 암호문을 만들고, 이 암호문을 전달하면 수신자는 암호문을 복호화 하여 평문을 만들어 내용을 확인하게 된다. 네트워크를 통해 지나다니게 되는 것은 평문이 아닌 암호문이다. 해커가 암호문을 잡을 수 있지만 해커는 암호화키가 없으므로 암호문을 가로채도 복호화 시킬 수 없다. 이와 함께 사용자가 올바른 사용자인지를 구분하는 인증기술, 원본 데이터와 복제된 데이터를 구분할 수 있게 하는 디지털 서명 등의 기술은 웹상에서 콘텐츠 보호를 위해 사용되는 기술들이다. 웹 어플리케이션 시스템에서 발생할 수 있는 보안상 취약점을 해결하기 위한 방법으로 전달되는 메시지의 암호화와 공개키를 관리가 필요하다. 본 논문은 웹

※ 제일저자(First Author) : 허진경
접수일자:2008년01월21일, 완료:2008년02월15일
* 호원대학교 사이버수사경찰학부
heojk@howon.ac.kr

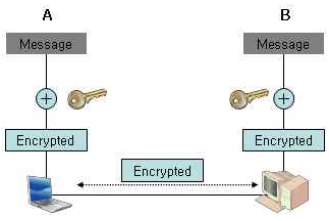
어플리케이션 시스템에서 데이터의 기밀성 및 사용자 인증을 제공함과 동시에, 다수의 클라이언트 접속시 암호화 서버의 병목현상으로 인한 성능저하를 방지하고 서비스 질을 향상시키기 위한 방법으로 분산 보안 시스템을 제안한다. 이를 위해 2장에서는 암호화 방법에 대하여 언급하였고, 3장에서는 분산처리 기술과 객체 활성화 기술을 이용해서 분산 보안 시스템을 구축하였다. 4장에서는 제안한 시스템 유효성 검증을 위하여 실험 및 결과를 제시하였다.

2. 암호화

정보 보안을 위협하는 요소에는 도청(盜聽), 변조(變造), 위조(僞造), 부인(否認) 4가지가 있을 수 있다. 이들 각각의 위협 요소들과 관련된 보안 요소는 내용이 노출되지 않는 기밀성(Confidentially), 내용이 변경되지 않는 무결성(Integrity), 정당한 사용자임을 확인 하는 인증(Authentication), 그리고 사용 증거를 확보하는 부인방지(Non-repudiation)가 있다. 암호화 알고리즘은 크게 대칭키 알고리즘, 해쉬 알고리즘, 공개키 알고리즘이 있는데, 대칭키 알고리즘은 기밀성에는 사용되고, 해쉬 알고리즘은 무결성에 사용된다. 그리고 공개키 알고리즘은 기밀성, 인증, 부인방지에 사용된다.

2.1 대칭키 암호화

대칭키 암호 알고리즘은 암호화 할 때 사용하는 키를 복호화 할 때 반대로 사용하면 복호화 되는 알고리즘이다. 메시지를 전송하는 송신측에서 사용하는 암호화키와 수신측에서 사용하는 복호화 키가 대칭을 이룬다고 해서 대칭키 알고리즘이라고 하며 비밀키(Secret Key) 방식이라고도 한다. 대칭키 알고리즘에는 DES, Blowfish, 3DES, ARS, RC4, SEED등이 있다.



(그림 1) 대칭키 암호 방식

(그림 1)은 대칭키 암호방식의 암호화 및 복호화를 보여주고 있다. (그림 1)에서 보여주고 있는 것처럼 대칭키 암호 방식은 메시지를 보내는 송신측과 받는 수신측이 같은 키를 사용하여 암호화/복호화를 수행 한다. 대칭키 알고리즘은 다양한 알고리즘 개발이 용이하며, 안전성 검장 방법이 비교적 정형화 되어 있다. 그리고 암호화 및 복호화 속도가 매우 빠른 장점이 있다. 그러나 키 관리 및 키 분배의 어려움이 있다. 그 이유는 Entry 쌍의 개수만큼의 키가 필요한데, N명의 사용자가 대화하기 위해서는 n(n-1)/2 쌍의 비밀키가 필요하다. 예를 들면 1천명의 사용자가 대화하기 위해서는 50만 여개의 키가 필요하게 된다. 뿐만 아니라 디지털 서명, 부인방지 등의 기능이 불가능하게 되고, 어느 한쪽이 키를 분실하면 시스템에 위험을 초래하게 되는 키 공유의 문제가 발생하는 등의 단점이 존재하게 된다. 대칭키 알고리즘은 주로 대용량 데이터 암호화에 사용한다.

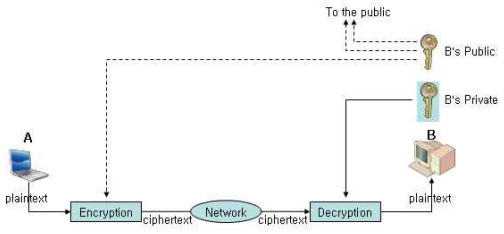
2.2 공개키 암호화

공개키 암호 알고리즘이 나타나기 전에는 대칭키 암호 알고리즘식이 사용되었다. 그러나 이러한 방식은 송신자의 비밀키를 사용하여 메시지를 암호화한 후 수신자에게 전달하는 방식으로, 수신자가 암호화된 메시지를 해독하기 위해서는 송신자의 복호화 키를 알고 있어야 한다. 따라서 이 방식은 복호화 키를 아는 사람은 누구라도 암호문을 복호화 할 수 있으므로 복호화 키를 수신자에게 전달할 때에는 각별한 주의가 필요하다.

이러한 단점을 극복한 방식이 바로 공개키 암호 알고리즘인데 이 방식은 복호화 키를 공유해야 하는 어려움을 해결하였다. 공개키 암호 알고리즘에서는 암호화키와 복호화와 키 중 암호화 키를 외부에 공개한다. 그러나 이 공개된 암호화 키로 복호화 키를 알아낼 수는 없다. 따라서 공개키 암호방식을 이용하여 특정인에게 비밀 메시지를 보내고자 하는 사람은 공개된 특정인의 암호화키를 이용하여 메시지를 암호화해서 보낸다. 그러면 그 특정인은 자신만이 아는 복호화 키로 메시지를 복호화 할 수 있다. 이런 이유 때문에 공개키 암호방식에서는 암호화키를 공개키라고 부른다. 다시 말해서 공개키 암호방식은 암

호화키와 복호화 키가 서로 다른 암호 방식을 말한다.

공개키 암호 알고리즘은 암호화 할 때 사용하는 키와 복호화 할 때 사용하는 키가 대칭을 이루지 않는다. 따라서 비대칭 키 방식이라고도 부른다. 공개키 암호 알고리즘은 비대칭이므로 1쌍, 즉 두 개의 키가 필요하다. 두 개의 키는 각각 공개키(Public Key)와 개인키(Private Key)로 부른다. 공개키는 인증기관을 통해 공개하는 키이며, 비공개로 본인만이 소유하게 된다.



(그림 2) 공개키 암호 방식

(그림 2)는 공개키 암호방식을 보여주고 있다. (그림 2)에서는 송신측에서 암호화 할 때의 키와 수신측에서 암호화된 데이터를 복호화 하는데 사용하는 키가 서로 다르게 되어 있다. 공개키 암호방식의 경우는 키의 크기가 크고, 또한 각 키는 특수한 성질을 요구하기 때문에 비밀키 암호화 방식의 키와 같이 사용자가 직접 원하는 키를 만들지는 못한다. 많이 사용되는 공개키 암호방식의 키 크기는 높은 안전성을 갖도록 하기 위하여 현재 사용되는 공개키 암호 방식 보안 제품은 중 RSA는 1,024bits의 키를 사용하고 있다

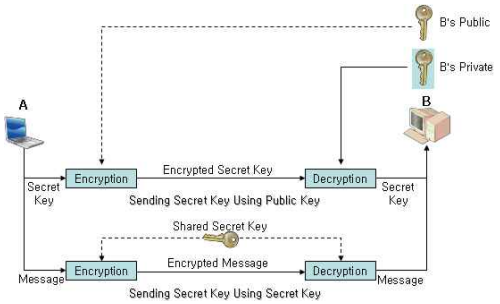
공개키 암호 알고리즘은 기밀성과 인증을 제공하며, 부인불가에 사용될 수 있다. 기밀성은 공개키는 인증기관을 통해 알 수 있으므로, 수신자의 공개키를 이용하여 메시지를 암호화 하면 수신자 외의 누구도 메시지를 복호화 할 수 없다. 즉, 사용자 A가 사용자 B에게 평문 메시지를 사용자 B의 공개키를 사용하여 암호화 하여 보내고, 사용자 B는 자신의 비밀키로 암호화된 메시지를 복호화 하여 평문 메시지를 얻는다. 사용자 B가 복호화에 사용되는 비밀키는 자신만이 가지고 있으므로 자신 외에는 아무도 자신의 비밀키로 암호화 된 메시지를 복호화 하여 볼 수 없다. 따라서 사용자 B만 메시지를 볼 수 있는 기밀성을 제공한다. 인증은 송신자가 자신의 개인키로 메시지를 암호화 하면 수신자가 송신자의 공개키로 복호화 하여 송신자에 대한 확인이 가능하다. 즉, 사용자 A가 사용자 B에게 평문 메시지를 자신의 비밀키를 사용 암호화 하여 보내고, 사용자 B는 사용자 A의 공개키로 암호화된 메시지를 복호화 하여 평문 메시지를 얻는다. 사용자 A가 암호화에 사용하는 비밀키는 자신만이 가지고 있으므로 자신 외에는 아무도 메시지를 암호화 할 수 없다. 따라서 암호화된 메시지는 사용자 A로부터 왔다는 인증을 제공한다. 이 경우에는 수신자를 제외한 제3자(해커)도 복호화 할 수 있다.

공개키 방식의 특징은 서로 연관된 키 쌍(개인키/공개키)이 필요하고, 하나의 키로 암호화한 경과는 반드시 쌍이 되는 키로만 복호화가 가능하다. 그리고 구조상 복잡한 수학연산이 필요하고, 안전성이 이러한 수학적 이론에 근거한다. 또한 N명의 사용자가 있으면 N개의 공개/개인키 쌍으로 충분하기 때문에 키 관리 문제를 해결 할 수 있다. 또한 서명기능, 기밀성 등의 기능이 가능하다. 그러나 공개키 방식의 단점으로는 동일한 양의 데이터를 암호화 시 대칭키 암호화에 비해 매우 느린데, DES에 비하여 하드웨어 구현에서 1000배가량 더 시간이 소요된다.

공개키 방식 알고리즘으로는 RSA(Rivest Shamir Adleman)와 ECC (Elliptic Curve Cryptography: 타원곡선 암호)가 있다.

2.3 대칭키와 공개키 조합

공개키 방식은 기밀성 및 인증이 가능하므로 공개키 방식만 사용해도 된다. 그러나 공개키 방식은 암호 알고리즘이 복잡하여 연산속도가 느려지므로, 대용량 데이터를 암호화 하는 데는 부적절하다. 반면, 대칭키 방식은 빠른 속도로 데이터를 암호화/복호화 수행한다. 따라서 일반적으로 메시지는 대칭키 방식으로 암호화 하고, 비밀키만 공개키 방식으로 암호화 하게 되면, 효율적인 방식으로 암호화/복호화를 할 수 있다. 다음 그림 3은 A와 B사이에서 메시지와 비밀키의 암호화를 보여주고 있다.



(그림 3) 대칭키와 공개키 조합

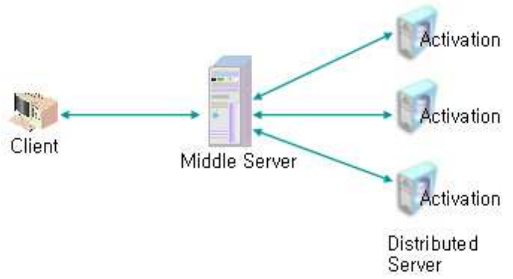
(그림 3)에서 송신자는 수신자에게 보낼 메시지를 대칭키 암호 방식을 이용하여 암호화 한다. 이때 수신자가 메시지를 복호화 하기 위해 필요로 하는 키는 공개키 암호 방식으로 암호화 한 후 수신자에게 전송된다. 수신자는 자신의 개인키를 이용해 송신측에서 전송되어진 비밀키를 복호화 한다. 그리고 비밀키를 이용해 메시지를 복호화 하는 방법을 사용하게 된다. 이러한 방법을 사용하면 웹 어플리케이션에서 전송되어야 할 대량의 데이터는 대칭키 암호 방법을 통해 보다 빠르게 암호화 할 수 있고, 그 외에 비밀키 또는 기밀성과 인증이 요구되는 데이터는 공개키를 사용하여 암호화 할 수 있다.

3. 분산 암호화 핸들링

3.1 분산처리

분산처리 기술의 가장 큰 장점은 작업을 분산시켜 작업함으로써 시스템 과부하 및 데이터 병목현상을 줄일 수 있다. (그림 4)는 클라이언트의 요구를 미들 서버 통해서 분산 처리하는 것을 나타내고 있다. 그러나 일반적인 분산처리가 아닌 객체 활성화 기능을 갖는 분산 처리 그림을 보여주고 있다.

(그림 4)에 나타나 있는 분산처리 서버들은 각각 객체 활성화 능력을 보유하고 있다. 객체 활성화 능력은 서비스하기 위한 자원 또는 프로세스를 저장해 둔 다음 임의의 서버에서 서비스를 제공하지 못할 경우에 대신 서비스를 해줄 수 있게 한다. 이는 하나의 보조 서버의 중단으로 인해 전체 시스템의 서비스 중지를 방지할 수 있다.

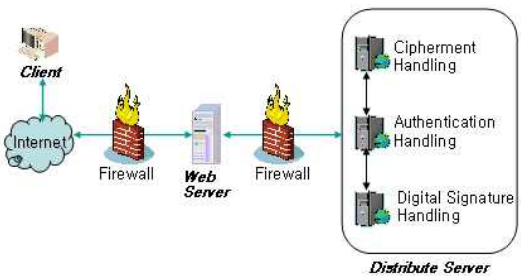


(그림 4) 분산 객체 활성화

분산처리 시스템에서 아주 가끔씩 호출되는 작업이 있을 경우에 이를 메모리에 상주해 놓는 것이 아니고, 디스크에 저장해 놓았다가 필요하면 가져다 쓰는 방식이다[2].

3.2 분산 암호화 키 활성화

(그림 5)는 보안과 관련된 작업을 분산서버에서 각각 나누어 하도록 설정 해 놓았다. 기본적으로 클라이언트에서 요구하는 모든 데이터는 웹서버에 전달되기 전에 방화벽을 거치게 된다. 이후 웹서버는 클라이언트의 요구에 따라서 분산 서버에서는 각각 암호화, 인증, 디지털 서명 등의 작업을 담당하게 된다. 이때 발생되는 비밀키들은 각각의 분산서버들 사이에서 통신이 이루어지게 된다. 각각 분산서버들에 존재하는 비밀키들은 다른 분산 서버가 정상적으로 동작하지 않을 경우에 대비하여 비밀키를 객체 활성화 상태로 만들어 놓는다.

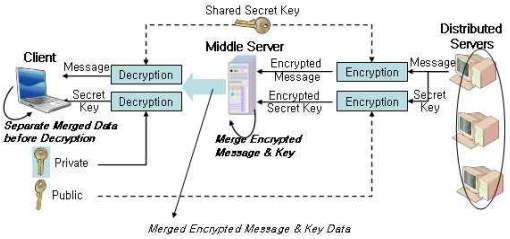


(그림 5) 분산 암호화 키 활성화

3.3 분산 암호화 처리

각 분산서버에서는 각각의 보안 처리 메시지와 암호화로 인한 비밀키가 생성되어 잇다. 이후 클라이언트에 보내지기 위한 분산 서버의 메시

지 암호화 처리는 대칭형 알고리즘 사용하여 빠른 속도로 데이터를 암호화/복호화를 수행시킬 수 있다. 분산 서버는 신뢰성 있는 미들웨어에 암호화 된 데이터와 메시지를 복호화 하기 위한 비밀키를 공개키 방식으로 암호화 하여 전송한다.



(그림 6) 분산서버에 의한 암호화

미들서버는 암호화 된 데이터를 조합한다. 미들서버에서 암호화된 데이터를 조합할 때 다시 조합 코드를 공개키 방식으로 암호화한다. 암호화된 후 조합된 메시지에는 분산서버에 의하여 처리된 메시지들과 이 메시지들을 복호화 하기 위한 비밀키가 포함되어 있다. 미들서버는 조합된 메시지에 추가로 메시지들을 풀기 위한 해제키를 공개키 방식으로 암호화 하여 전송한다. 클라이언트는 메시지를 이해하기 위해서는 1차로 공개키 방식에 의하여 조합코드를 이해한다. 조합 코드에 의해 미들서버에 의해 조합된 메시지를 처리하여 분산서버에서 생성된 암호화 코드와 암호화 코드를 해석하기 위한 비밀키를 얻는다. 이때 암호화 코드를 해석하기 위한 비밀키는 공개키 방식으로 암호화 되었으므로, 사용자 인증 및 기밀성 등을 제공 할 수 있다.

4. 실험결과

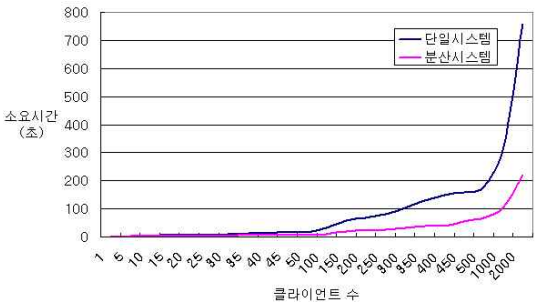
해킹 기술의 발전과 암호화 처리 데이터의 양은 증가하는 상황에서의 웹 어플리케이션 시스템 구축은 보안상 많은 취약점을 나타낼 수 있다. 뿐만 아니라 사용자수에 비례하는 네트워크 트래픽은 보안 시스템에 심각한 병목 현상을 초래할 수 있다. 본 논문은 웹 어플리케이션 시스템에서 암호화와 인증을 위한 시스템의 구현을 제안하였다. 이를 위해 관련 암호화 기술의 소개와 분산처리, 그리고 이들의 장/단점에 대하여 정의하였다. 이들을 바탕으로 하여 웹 어플리케이션에서 콘텐츠 보호를 위한 보안시스템의 구조를 설계하였다.

실험은 사용자 요구 처리에 사용되는 소요시간 계산을 단일 시스템에서 처리할 때와 분산 시스템에서 처리할 때를 비교하였다. 실험은 각 클라이언트 수에서 10회 응답 속도를 체크하여 평균값으로 처리하였다.

단일 시스템에서 암호화 처리 실험을 위한 시스템 사양으로 펜티엄 D 3.40Ghz CPU, 2G Ram PC 1대를 사용하였다.

분산 시스템에서 암호화 처리 실험을 위한 시스템 사양으로 미들서버 1대(펜티엄 D 3.40Ghz CPU, 2G Ram PC)와 분산서버 3대(펜티엄 D 3.40Ghz CPU, 2G Ram PC)를 이용하였다.

(그림 7)에서는 단일 시스템에서 암호화 처리는 암호화를 요구하는 클라이언트 수가 많아질수록 서버의 응답속도는 현저하게 떨어지는 것을 볼 수 있다. 그러나 분산 시스템에서는 스트레스테스트를 하기 위해 클라이언트 수를 증가시켰을 때 일정 수준까지 서버의 응답속도를 유지하는 것을 볼 수 있다.



(그림 7) 클라이언트 수에 따른 응답ㅂ시간

5. 결론

본 논문에서는 사용자의 요구를 분산처리 하였다. 콘텐츠의 암호화 처리는 대칭키 알고리즘을 사용하여 암호화 전송함으로써 데이터 무결성 처리를 하였고, 암호화에 소요된 처리 시간을 단축 시켰다. 그리고 사용자 인증을 위해 복호화 키를 공개키 방식으로 암호화하여 전송함으로써

데이터의 기밀성 및 인증작업을 처리하였다.

향후 SSL통신 및 PKI를 적용하여 메시지를 보다 안전하게 전송하기 위한 시스템을 구현하고자 한다.

참 고 문 헌

[1] 한국전산원, “웹 환경 구축 및 운영을 위한 보안 기술 연구”, NCA III-RER-97052, 1997. 12

[2] 서영규, “웹 프락시를 이용한 웹 입력 오류 공격 탐지 및 차단 연구”, 동국대학교 국제정보대학원 석사학위 논문, 2004.

[3] 김신규, 한광택, “안전한 웹 응용프로그램 개발에 관한 연구”, 2004

[4] Nam-Deok Cho, Eun-ser Lee, Hyun Gun Park, "Security Intelligence: Web Contents Security System for Semantic Web", KES, LNCS, ISBN 978-3-540-46537-9, Volume 4252/2006, pp.819-828, 2006

[5] 한국썬교육센터, "웹 어플리케이션 보안", 2007.

[6] 황순일, 김광진, "웹 해킹 패턴과 대응", 사이텍미디어, 2005.

[7] Roger Fournier저 유해영 역, "웹 어플리케이션 개발 방법론", 이한출판사, 2002

[8] Mike Shema McGraw-Hill Hacknotes, "Web Security Portable Reference", Companies Inc, 2003

[9] Distributed Programming With Java Technology, Revision C.2, Sun Microsystems Inc, April 2001.

[10] 정보통신부 한국정보보호진흥원, 웹 어플리케이션 보안 템플릿, 2006.

[11] 정보통신부 한국정보보호진흥원, 홈페이지 개발 보안 가이드, 2005

[12] 과학기술부, 새로운 방식의 공개열쇠 암호의 제작과 기존 방식의 공개열쇠 암호의 연구, 2002.

[13] 정보통신부, 타원곡선 암호시스템을 이용한 차세대 정보보호기술의 연구 및 개발, 2000.

[14] 정보통신부 한국 정보보호센터, 공개키 암호알고리즘 개발에 관한 연구 : 최종연구개발결과보고서, 1999

허 진 경

1998년 : 호원대학교 전자계산학과 (이학사)

2000년 : 조선대학교 전산통계학과 (이학석사)

2004년 : 조선대학교 전산통계학과 (이학박사)

2006년～현 재: 호원대학교 사이버수사경찰학부 연구교수

관심분야 : 정보보호(Information Security), 암호화(Encipherment)