情 報 保 護 學 會 誌
第20卷 第6號, 2010. 12

전자금융거래에서의 문서변조 취약점 분석 및 대응방법 고찰

맹 영 재*, 신 동 오*, 김 성 호**, 양 대 헌*

요 약

전자금융거래는 사용자의 컴퓨터에 악성 프로그램이 설치될 수 있다는 환경에서도 신뢰성 있는 서비스가 요구된다. 하지만 국내의 전자금융거래는 아직까지 MITB(Man-In-The-Browser)공격에 취약한 상태이다. 이 논문에서는 MITB 공격의 동작원리와 그 대응방법에 대해 논의하며, 이를 바탕으로 QR코드를 활용한 승인방법을 제안한다.

Ⅰ. 서 론

국내 전자금융거래의 이용률은 인터넷과 웹의 출현 이후로 꾸준히 증가세를 유지하고 있다. 한국은행이 발표한 자료에 따르면 2010년도 2/4분기 중 국내 인터넷 뱅킹 일평균 이용실적은 3,291만건, 29조 9,548억원으로 꾸준한 증가세를 유지하고 있고, 전체 입출금 및 자금이체의 34.1%, 조회서비스의 66.1%로 금융서비스 전달채널 중 가장 높은 비중을 차지하고 있는 것으로 조사되었다[3]. 인터넷뱅킹 뿐만 아니라 증권거래, 카드결제, 보험 등의 금융업무와 전자세금 계산서, 전자입찰, 전자계약 등의 기업 조달업무, 정부에서 제공하는 전자민원, 전자정부 업무 등도 웹 브라우저를 통해 서비스되고 있으며 웹을 이용한 중요업무는 앞으로도 증가할 것으로 보인다.

전자금융은 비대면 거래이기 때문에 사용자 인증과 사용자의 거래를 공격자로부터 보호하는 것이 중요하다. 다양한 형태의 비밀과 이 비밀을 보호하는 보안 프로토콜이 사용자 인증을 안전하게 하고 각종 보안 프로그램은 다른 위협들로부터 사용자의 비밀과 컴퓨터를 보호하여 전자금융거래를 가능하게 하고 있다. 그럼에도 불구하고 현재 사용되고 있는 보안 프로그램들은 MITM(Man-In-The-Middle)의 일종인 MITB공격에 무방비 하다는 사실이 실험을 통해 밝혀졌다.

MITB는 2005년 Augusto에 의해 처음 발표되었고 2007년 Philipp에 의해 이름 붙여진 공격으로 국내에서는 2008년 임형진 등의 논문에서 언급된 바 있다[2,8,11]. MITB는 사용자의 컴퓨터에 악성 프로그램이 설치될 수 있다는 가정 하에 가능한 공격이다. Sharek등의 연구[6]에서는 사용자가 '예'라는 버튼에 익숙하다는 사실을 진짜와 가짜 팝업 에러 창을 구분할 수 있는지에 대한 실험을 통해 보인바 있다. 73%의 사용자가 가짜 팝업 에러 창을 구별하지 못했고, 'OK'버튼을 선택한 사용자 중에서 12%는 '에러 내용에서 그렇게 지시했기 때문에', 23%의 사용자는 '에러 메세지를 볼 때면 언제나 OK를 누른다', 42%의 사용자는 '에러 창을 없애고 싶어서'라고 답한 것으로 조사되었다. 이에 더해, 국내에서 ActiveX 또는 BHO, Toolbar 형태의 플러그인(plug-in)의 사용이 빈번하다는 사실은 사용자들에게 프로그램 설치 요구를 선택의 문제이기 보다는 서비스를 제공받기 위한 하나의 통과의례로 인식하도록 만들었고 그만큼 악성프로그램의 배포 또한 쉬워졌다고 할 수 있다. 국내의 웹은 그만큼 MITB 공격이 어렵지 않게 발생할 수 있는 환경에 놓여있는 것이다.

이 논문에서는 MITB 공격과 관련하여, 2장에서 국내 전자금융거래 보안서비스 현황을 소개하고 3장에서는 문서변조 공격 원리와 대응방안 논의, 4장에서는 MITB 대응방법 동향을 분석한다. 5장에서는 MITB에

이 논문은 2010년도 정부(교육과학기술부)의 재원으로 한국연구재단의 기초연구사업 지원을 받아 수행된 것입니다.(2010-0013254)
* 인하대학교 정보공학과 정보보호연구실 ({brendig,mannershin}@isrl.kr, nyang@inha.ac.kr)
** 인하대학교 정보통신대학원 (night12th@isrl.kr)

대한 보안성과 사용자 편의성을 고려하여 QR코드를 이용한 승인방법을 제안하고 6장은 MITB 공격 대응방법의 비교를 보이고 7장은 결론을 담는다.

Ⅱ. 국내 전자금융거래 현황

이 장에서는 전자금융거래에 요구되는 보안서비스와 그에 대한 국내 현황을 알아본다.

2.1. 사용자 인증에 사용되는 비밀

국내 전자금융거래의 인증과정에는 고정된 비밀 중에서 사용자가 외우는 형태의 비밀과 파일 형태의 비밀이 함께 사용되고 있다(Multi-factor authentication)[표 2]. 고정된 형태의 비밀은 모두 복제가 가능하지만 유동적인 형태의 비밀은 OTP는 서버와 공유되어 있는 seed를 얻어내지 못하면 복제가 불가능하다. 그 때문에 고정된 비밀은 한번 노출되면 더 이상 비밀로서의 가치가 없다. OTP의 경우에는 비밀이 매번 바뀌고 제한된 시간 안에 사용해야 하거나 또는 일회용이기 때문에 고정된 형태의 비밀보다는 안전하지만 그 제한적인 시간 안에 공격자가 악용할 가능성이 있다. 이러한 취약점은 키보드를 통해 입력하는 비밀의 이용 방법이 모두 동일하기 때문이다.

인증 프로토콜을 직접 공격하여 비밀을 얻어내는 것보다 사용자가 비밀을 입력하는 단계에서 공격하는

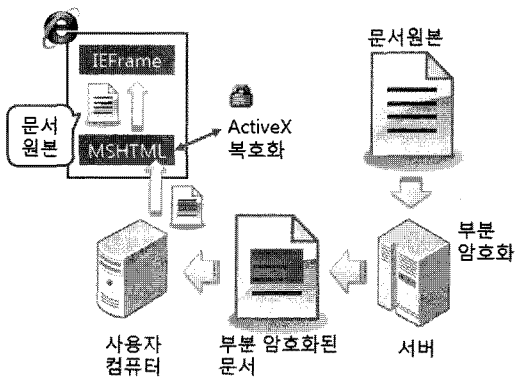
〔표 1〕 전자금융거래에 요구되는 보안 서비스

사용자/서버 인증 (2.1장)	전자금융은 비대면 거래이기 때문에 사용자가 정상적인 서버에 접근하였는지, 그리고 서버는 접속한 사람이 주장하는 사용자 본인이 맞는지를 안전한 인증과정을 통해 확인할 수 있어야 한다.
데이터 기밀성 (2.2장)	제 3자에게 사용자의 민감한 정보가 노출되어서는 안 된다.
데이터 무결성 (2.3장)	사용자와 서버가 주고받는 데이터가 중간에 변조되었을 경우 사용자와 서버는 이를 감지할 수 있어야 한다.
거래 부인방지 (2.4장)	사용자와 서버는 거래가 승인된 이후에 이를 부인할 수 없도록 제 3자에게 거래 사실을 입증할 수 있어야 한다.
서비스 가용성 (2.5장)	전자금융거래 시스템은 사전에 지정해 놓은 점검시간 외에는 항상 서비스가 가능해야 한다.

〔표 2〕 사용자 비밀의 종류

대분류	소분류	사용자 비밀의 종류	비고
고정된 비밀	외우는 형태의 비밀	로그인 패스워드	영문,숫자 혼합 6~10자리
		공인인증서 패스워드	암호화되어있는 공인인증서를 사용하기 위한 패스워드
		계좌(거래) 비밀번호	계좌이체 등을 위한 패스워드
	파일 형태의 비밀	공인인증서	하드디스크, 이동식저장장치, 휴대폰 등에 저장될 수 있음 외우는 형태의 비밀과 함께 사용됨
		ISP (Internet Secure Payment)	
	소유하는 형태의 비밀	보안카드	challenge-response형태로 사용됨. 비밀의 수가 제한됨(예,4자리 숫자 35개)
유동적인 비밀	소유하는 형태의 비밀	OTP (One Time Password)	매번 다른 6자리 숫자를 생성해냄. 복사가 불가능함

것이 더 쉽다고 생각하는 공격자는 사용자의 컴퓨터에 악성 프로그램을 설치하여 사용자가 입력하는 비밀을 훔쳐보거나(키로거: Key-logger 또는 어깨너머 훔쳐보기: Shoulder-surfer) 가짜 사이트에 비밀을 입력하도록 유도하는 공격을 시도한다. 이러한 공격들에 대비하여 사용자의 컴퓨터에 설치되는 보안 프로그램은 키보드 후킹(Hooking)을 막아 사용자의 비밀입력을 보호하거나 웹 브라우저가 공격자의 사이트로 이동되는 것을 방지하는 피싱/파밍방지 서비스 등을 제공하고 있다.



〔그림 1〕 국내 전자금융의 부분 암호화 및 복호화 과정

2.2. 기밀성을 제공하기 위한 암호화

국내 전자금융은 TLS/SSL을 이용하기도 하지만 민감한 데이터만을 부분적으로 암호화하는 방식을 주로 채택하고 있는 것이 특징이다. 사용자가 인증을 시도할 때 서버와 클라이언트에 설치되어 있는 플러그인이 세션키를 공유하고, 서버가 데이터를 부분적으로 암호화하여 웹 브라우저에 전송하면 자바스크립트가 플러그인을 통해 복호화 한 뒤 IEFrame에 출력하도록 하는 구조를 가진다[그림 1]. SSL/TLS는 TCP/IP수준에서 암호화 및 복호화를 처리하지만 플러그인을 이용한 방법에서는 데이터가 응용계층까지 올라가 암호기능을 수행하기 때문에 보다 많은 구간에 대해 기밀성을 제공하고 있다[그림 2]. 여기서의 암호화는 네트워크 수준에서의 기밀성을 제공하는 것이 목적으로 응용계층에서 동작하는 악성 프로그램에 대해서는 기밀성이 제공되지 않는다.

2.3. 부분적인 암호화와 문서의 무결성에 대해

HTML문서(이하 문서)를 부분적으로 암호화한 경우에는 문서전체의 무결성을 확인하는 것이 더욱 중요해진다. 암호키 없이 암호화된 부분을 수정하는 것은 불가능하지만 그 외의 부분에 악성코드를 삽입하여 악의적인 행동을 하는 것이 가능해지기 때문이다. [그림 2]는 웹 브라우저에 악성코드가 침입 가능한 경로를 보여준다.

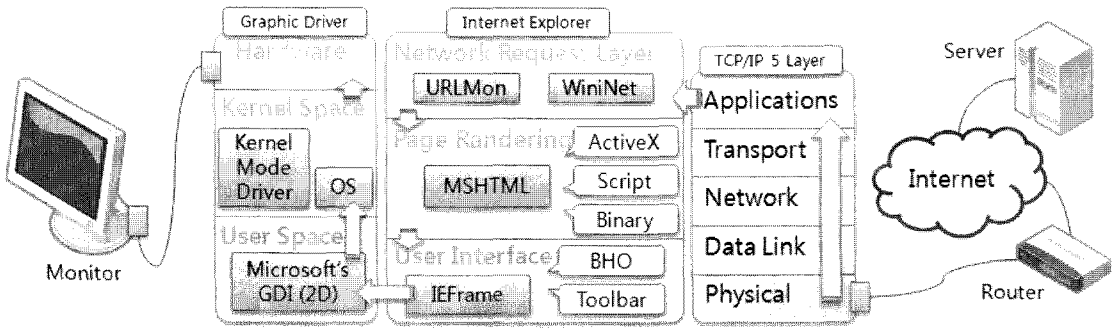
2.2장에서 언급하였듯이, 암호화는 네트워크 수준에서 기밀성을 제공하기 위함이다. 문서가 복호화 되어 IEFrame에 출력된 이후로는 기밀성뿐만 아니라 무결성 또한 기대할 수 없다. 악성 프로그램이 복호화 된 문서를 브라우저의 Page Randering 또는 User Interface 수준에서 수정할 수 있기 때문이다. 예로, 시스템의 자원을 이용하여 웹의 응용을 가능하게 하는 BHO(Browser Helper Object)나 Toolbar는 User Interface 수준에서 문서를 수정할 수 있다.

2.4. 부인방지를 위한 승인과정

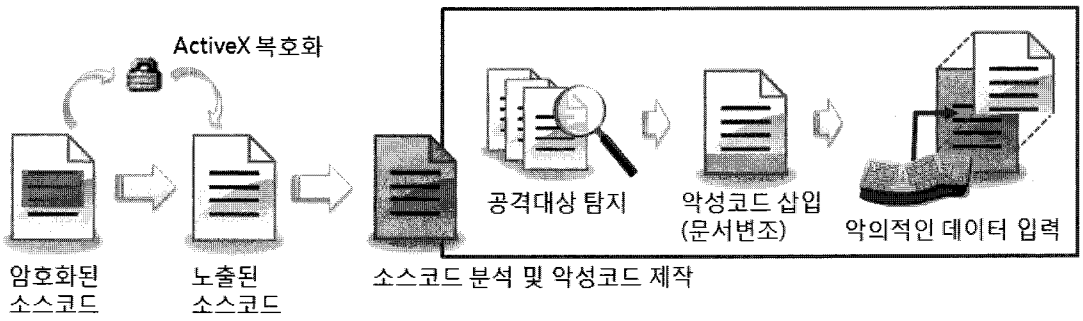
승인과정에서는 사용자가 거래하고자하는 내용을 확인하고 그에 대한 사용자의 응답으로 외우는 형태의 비밀, 파일 형태의 비밀, 소유하는 형태의 비밀[표 2]을 복합적으로 이용하고 있다. 이는 인증과정에 비해서 비밀의 소유를 추가적으로 증명해야하기 때문에 인증과정보다 높은 보안성을 제공하는 것이다. 하지만 사용자의 비밀이 증명하는 것이 사용자의 거래내역에 이상이 없었음을 보장해주지는 않는다. 사용자가 입력하는 현재 사용 중인 비밀은 승인하고자 하는 거래내역과 연관되어 있지 않기 때문이다. 이 경우 악성 프로그램이 실시간으로 변조한 문서에 사용자는 의심 없이 비밀을 입력하게 되고 결국 공격자의 의도대로 거래가 완료되는 부정승인이 가능하기 때문에 MITB 공격이 가능한 상태에서는 부인방지 서비스가 정상적으로 제공되고 있다고 보기 어렵다. 사용자가 바라본 거래내역이 정상적인 것이었는지 또한 확인할 수 있을 때 부인방지가 정상적으로 서비스 된다고 할 수 있다.

2.5. MITB 공격과 서비스 가용성

전자금융거래에 대해 공격이 탐지되었다 하더라도



〔그림 2〕 서버에서 생성된 문서가 사용자의 화면으로 출력되기까지의 데이터 이동 경로

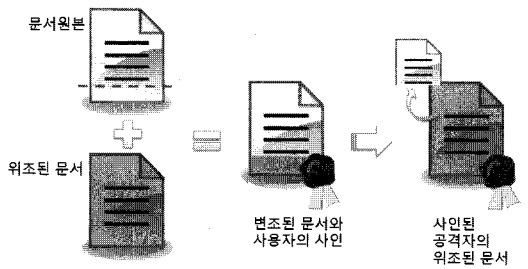


〔그림 4〕 MITB를 이용한 문서변조 공격 순서

서비스는 계속 이루어져야 한다. 금융거래는 서비스의 신뢰성이 특히 중요하기 때문이다. 서비스의 가용성을 저해시키는 것이 목적인 DDoS(분산서비스공격: Distribute Denial of Service)와는 다르게 MITB는 서비스가 정상적으로 이루어지고 있을 때 공격을 하고 피해를 발생시킨다. 부인방지가 이상적으로 제공되지 않는 상태에서 MITB공격이 어느 한 순간에 다수 발생한다면 서비스 가용성은 오히려 더 많은 피해를 유도할 수도 있다. 이러한 피해를 방지하기 위해서는 MITB 대응방법이 빠른 시일 내에 마련되어야 할 것이다.

Ⅲ. MITB를 이용한 문서변조 공격 동작원리 및 대응방안 분석

문서변조 공격은 문서에 대한 무결성이 제공되지 않을 때 가능한 공격으로, 화면에는 사용자가 거래하고자 하는 화면을 보여주되 사용자가 입력하는 비밀은 건드리지 않은 상태에서 공격자의 의도대로 거래를 완료시키는 것이 목표이다. 이는 사용자의 눈을 속인다는 점에서 피싱 또는 파밍과 비슷하다고 볼 수 있으나 비밀을 단순히 얻어내는 형태의 공격이 아니고 사용자 의심 없이 입력된 비밀을 악용한다는 점에서 더욱 지능화된 공격이라 할 수 있다.



〔그림 3〕 사인된 공격자의 위조된 문서

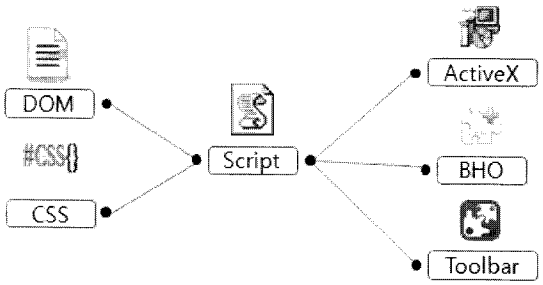
[그림 4]를 보면 문서원본과 위조된 문서(공격자의 문서)가 있다. 공격자는 자신의 문서 위에 문서원본의 내용부분만이 겹치도록 만들고 이렇게 변조된 문서를 사용자에게 보여준다. 문서의 내용에 이상이 없음을 확인한 사용자는 문서에 사인을 한다. 사인된 문서를 받은 공격자는 겹친 부분을 떼어내어 사용자가 사인한 위조된 문서를 가지게 된다. 사용자는 자신이 거래하고자 하는 내용을 눈으로 확인하고 비밀을 입력하기 때문에 사용자의 눈을 속일 수 있으면 사용자의 의심 없이 비밀을 얻어낼 수 있는 것이다. 때문에 단순히 비밀을 증명하는 방법은 문서변조 공격 앞에서는 의미가 없다.

문서를 변조하는 공격은 프로그램을 통한 자동화가 요구되는데 이를 위해서 공격자는 [그림 3]과 같이 공격대상 시스템을 분석할 수 있어야 하고 제작한 악성코드를 원하는 시점에 삽입할 수 있어야 한다.

3.1장부터 3.4장까지는 MITB를 이용한 문서변조 공격의 구동 원리와 그 대응방법을 논의하고 4.5장은 문서변조 공격이 전자금융거래에 미칠 수 있는 영향을 알아본다.

3.1. 소스코드가 노출되는 스크립트

웹에서 전자금융거래를 가능하게 하는 데에는 자바스크립트의 역할이 크다. 정적인 HTML문서의 각 개체에 DOM(Document Object Model)형태로 접근하고 각종 플러그인과 통신하여 동적으로 문서를 구성하는 것이 자바스크립트(Javascript 또는 VB script, 이하 스크립트)이기 때문이다. 이 스크립트는 클라이언트에 컴파일 된 형태로 제공되지 않고 소스코드가 인터프리터에 의해 직접 해석되기 때문에 HTML에 소스코드가 그대



〔그림 5〕 웹 문서를 구성하는데 중요한 역할을 하는 스크립트

로 노출되어 있다. 2.3장에서 언급했듯이 암호화된 형태로 전달되기도 하지만 이를 복호화 하고 출력하는 것 또한 스크립트가 제어하고, 스크립트는 평문 형태로 IEFrame에 출력되어야 해석될 수 있기 때문에 공격자가 소스코드를 얻어내는 것은 어렵지 않다.

공격자가 소스코드를 얻어내는데 성공했다면 다음 단계는 소스코드를 분석한 이후에 이를 바탕으로 악성 스크립트를 제작하는 것이다. 스크립트는 매번 동일한 것이 사용되기 때문에 소스코드 분석 후 작성된 공격 스크립트가 동작할 수 있는 것이다. 소스코드가 노출되는 것을 막을 수 없다면 분석이 어렵도록 또는 분석된 코드가 더 이상 의미가 없도록 소스코드가 요청될 때마다 혼잡화(Code Obfuscation)하는 방법이 있지만 이 방법은 근본적인 해결책이 될 수 없다[10]. 코드 혼잡화가 소스코드의 해석을 불가능하게 만든다는 것을 의미하지 않으며 공격자는 한번 분석한 코드를 이용하여 거래에 필요한 최소한의 스크립트만을 유지, 문서 자체를 재구성할 수도 있기 때문이다. 거래에 필요한 최소한의 스크립트란 거래에 필수적으로 요구되는 플러그인과 관련된 스크립트, 폼 전송에 사용될 변수명을 유지한 것으로 공격자는 서버가 전송한 문서대신 이 문서가 화면에 출력되도록 하여 부정거래를 완료시킬 수 있다.

3.2. 공격대상 문서의 탐지

악성 프로그램이 공격하고자 하는 시점을 파악하기 위해서는 웹 브라우저에 출력된 문서가 공격대상인지를 파악할 수 있어야 한다. 일반적으로 서버에서 문서를 구분하는 방법이 URL(Uniform Resource Locator)이라는 것을 고려하면, 악성 프로그램은 단순히 URL을 읽어 공격 대상을 구분해낼 수 있다. 예로, onDocumentComplete이벤트를 이용하여 다운로드가 완료된 문서의 URI를 얻어낼 수 있다. 이에 대비하여 URL을 임의적으로 생성하는 방법을 적용해볼 수 있겠지만 특정한 패턴(예, 그림파일이나 내용)을 통해 문서가 구분될 수 있고 이 모두를 혼잡화 하여도 실시간으로 캡처한 화면을 비교하여 공격대상을 파악하려 할 수도 있다.

3.3. 스크립트의 삽입

악성 프로그램이 공격대상 문서를 탐지해내는데 성공하면 사전에 작성해놓은 악성코드를 삽입한다. 악성코드가 삽입될 수 있는 경로는 TLS/SSL이 적용된 문서의 경우 전송(Transport)계층 이후의 구간에서 스크립트를 삽입할 수 있고 플러그인을 통한 부분적인 복호화가 요구되는 경우에는 인터넷 익스플로러(Internet Explorer)의 유저 인터페이스(User Interface)까지 모든 구간에서 스크립트 삽입이 가능하다. 예로, 웹 브라우저 컨트롤 IHTMLDocument2의 execScript함수를 이용하여 스크립트를 실행시킬 수 있다.

스크립트 삽입을 통해 문서를 변조하는 방법은 두 가지가 있다. 가짜 문서를 제작하여 CSS의 z-index속성을 이용, 문서 원본에 계층구조 형태로 덮어씌우는 방법과 스크립트의 함수 오버라이딩(Function Overriding)을 통해 서버와 통신하는 데이터만을 변조하는 방법이다. 스크립트의 삽입을 탐지하는 방법은 문서가 IEFrame에 로드되기 전과 후로 나누어 생각해볼 수 있다. 문서가 IEFrame에 출력되기 전에는 유저 인터페이스(ActiveX는 페이지 렌더링 단계이기 때문에 안 됨)수준에서 무결성을 확인하고 문서가 IEFrame에 출력된 이후에는 특정한 이벤트가 발생할 때 또는 주기적으로 문서가 변조되었는지를 확인한다. 참고로 보호하고자 하는 문서가 정적인 경우에는 IEFrame에 문서가 출력된 이후에 onPropertyChange 이벤트로 문서의 변조여부를 탐지할 수 있다. 인터넷뱅킹 계좌이체의 경우, 승인단계에서 공인인증서 패스워드를 입력할 때 창 상단에 위치한 거래내역은 IEFrame을 통해 출력된다. 이 거래내역은 정적인 문서이고 문서 수정과 관련한 이벤트가 발생하면 에러메시지와 함께 거래를 더 이상 진행할 수 없도록 되어있다. 하지만 동적인 문서에 대해 이 이벤트를 사용하려면 정상적으로 거래할 때 속성이 변할 수 있는 모든 태그(HTML tag)의 리스트를 유지해야 하고 변환된 내용 또한 사전에 정의된 규칙대로 변환되었는지 여부 등

을 확인해야 한다. 변환된 내용이 사전에 정의해놓은 규칙을 통과했다 하여도 문서는 여전히 오버라이딩이 허용되는 스크립트에 의해 변조될 수 있다. 또한 공격자는 API 후킹, DLL 인젝션(Injection), DMA(Dynamic Memory Allocation) 또는 IAT(Import Address Table) 변조와 같은 해킹 기법으로 OpenProcess, ReadProcessMemory, WriteProcessMemory, VirtualProtectEx등의 함수를 이용하여 메모리상에 존재하는 문서에 접근하고 또 수정할 수 있다.

3.4. 공격자가 의도하는 데이터 입력

화면을 조작하는데 성공했다 하여도 문서에 공격자가 원하는 값을 입력할 수 없다면 공격은 완료될 수 없다. 이는 키보드 입력 데이터와 관련이 있다. 사용자가 입력하는 키 데이터는 키보드 보안 프로그램이 지니고 있다가 요청이 있을 때 암호화 된 값만을 이용할 수 있도록 되어있다. 즉, 키보드 보안 프로그램에 키 데이터를 입력할 수 없다면 공격자의 의도대로 거래가 이루어질 수 없도록 되어있다. 하지만 화상키보드나 원격데스크톱, 타블렛 노트북으로도 인터넷뱅킹이 가능하다는 점은 키 이벤트를 통해 키보드 보안 프로그램에 원하는 데이터를 입력하는 것이 가능하다는 것을 보여준다. 예로, 키보드 드라이버의 인터럽트 핸들러가 호출하는 keybd_event(Winuser.h에 정의되어 있음)를 이용하면 키보드 보안 프로그램에 공격자가 원하는 데이터를 입력할 수 있었다.

3.5. 공격 시나리오

이체가능전액을 공격자의 계좌로 이체: 사용자가 계좌이체를 할 때 화면에는 사용자가 입력한 수신자 정보와 이체금액을 출력하고, 실제로는 공격자의 계좌에 이체가능전액을 이체하도록 한다. 사용자가 입력한 이체정보와 공격자의 계좌에 나머지 전액이 동시에 이체되도록 하는 것 또한 가능하다. 사용자의 보안수단(보안카드 또는 OTP등)에 따른 보안등급이 높을수록 피해액이 커진다.

주가조작: 공격자의 악성 프로그램에 감염된 컴퓨터가 다수이거나 큰 금액을 보유한 사용자가 그 중에 포함되어 있는 경우, 공격자가 특정 주식을 강제로 매수하거나 매도하도록 하여 간접적으로 이득을 취하려 할 수 있다. 공격자가 피해자의 주식거래와 직접적으로 관련되지 않기 때문에 주가조작과 관련한 증거를 찾는 것이 쉽지 않을 수 있다.

부정결제: 공격자가 의도하는 방향으로 결제를 유도하여 사용자에게 금전적인 피해를 입힐 수 있다. 결제과정에는 고정된 형태의 비밀만이 사용되고 있기 때문에 MITB가 문서를 변조하는 공격 대신 비밀을 훔치는 공격을 하여 비밀을 얻어낸다면 공격자가 원할 때 사용자의 신용을 악용할 수 있다.

Ⅳ. MITB 대응방법 동향

문서변조 공격은 사용자의 눈을 속이는 공격이기 때문에 이에 대응하기 위해서는 사용자가 정상적인 거래내역을 보고 승인하였는지 여부를 서버가 확인할 수 있어야 한다. 이는 승인과정이 현재와 같이 단순하게 비밀을 증명하는 형태가 아니라, 사용자의 거래내역을 변조할 수 없는 추가적인 장치를 이용하는 방법 또는 거래내역을 확인하는 단계에서 악성코드가 자동화된 공격을 할 수 없도록 거래내역이 포함된 CAPTCHA (Completely Automated Public Turing test to tell Computers and Humans Apart)를 이용하는 방법이 적용되어야 함을 뜻한다.

CAPTCHA는 사람과 로봇의 인지능력에 차이가 있다는 사실에 기초하여 자동화된 로봇을 방지하는 보안기법이다. 온라인 커뮤니티의 회원가입이나 글을 작성할 때 찾아볼 수 있으며 최근에는 DDoS 공격을 차단하기 위한 수단으로 연구가 이루어지고 있다[9]. 하지만 위와 같은 환경에서 사용되는 CAPTCHA는 CAPTCHA의 내용을 그대로 입력하는 단순한 형태 즉, 단순히 사람과 로봇을 구분하기 위한 용도이기 때문에 MITB 공격에 대응하기에는 부족한 면이 있다. 변조된 문서를 바라보는 사용자는 아무런 의심 없이 CAPTCHA에 응답할 수 있기 때문이다. ArcotVPS와 MS워터마크는 OTP를 CAPTCHA로 표현하여 MITB 공격에 대응할 수 있도록 하였다[13,7]. 하지만 두 방법 모두 CAPTCHA를 해석하지 않더라도 공격이 가능하다. ArcotVPS의 경우 배경과 문자열을 분리시킬 수 있는 경우 1/4의 확률로 공격이 가능하고 MS워터마크 또한 문서위에 출력된 CAPTCHA 문자열 추출할 수 있으면 공격자의 의도대

로 거래를 완료시킬 수 있다. CAPTCHA에서 특정한 색상을 분리시키거나 문자열의 굵기, 투명도 등으로 문자열을 추출하는 것은 컴퓨터의 이미지 처리(Image Processing)를 통해 가능하다. 이는 [표 5]에서 더욱 자세히 다루도록 한다.

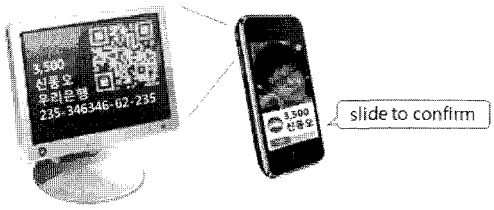
부가장치를 이용한 방법은 신뢰할 수 있는 장치를 통해 거래내역을 전달하고 사용자에게 승인하도록 하는 것이 목적이다. 현재 사용 중인 MITB 공격 대응방법으로는 인터넷뱅킹 계좌이체 과정에서의 전화승인서비스가 유일하다. 이는 계좌이체 승인단계에서 ARS 시스템을 이용해 사용자에게 전화를 걸어 거래내역을 확인시켜 주는 방법으로 이 과정에서 거래를 승인하거나 취소할 수 있다. 트랜잭션 서명 기법은 사용자 토큰이라 불리는 추가적으로 발급받은 기기에 거래내역의 일부를 입력하면 토큰에 출력된 MAC을 사용자가 컴퓨터에 수작업으로 입력하는 방법이다[5]. 이 연구에 기초한 임형진 등의 연구[2]는 사용자 토큰에 인증서를 저장한 형태 또는 사용자가 입력한 MAC을 컴퓨터에서 사인하는 방법으로 부인방지를 추가적으로 제공한다. 트랜잭션 서명 기법은 사용자 입장에서는 일부이기는 하지만 거래내역을 한 번 더 입력해야 하기 때문에 편의성이 떨어진다. IBM의 ZTIC은 USB 장치의 디스플레이를 통해 거래내역을 출력하고 두 개의 버튼을 통해 승인 또는 취소를 결정하도록 한다[14]. ZTIC은 컴퓨터에 연결되면 USB 저장장치로 인식이 됨과 동시에 사전에 정의되어있는 사이트(은행)으로 연결되도록 하는 프록시를 설정하고 웹 브라우저와 해당 사이트와의 통신은 모두 ZTIC을 통해 이루어지며 이 세션을 암호화하는 TLS/SSL의 키는 악성 프로그램이 접근할 수 없도록 되어있다. 내용을 확인하고 버튼으로 승인여부를 결정하면 되기 때문에 편의성이 좋지만 추가적인 장치를 발급해야 한다는 단점이 있다.

5장에서는 MITB에 효과적으로 대응할 수 있으며 편의성이 좋은 승인방법을 제안한다.

V. QR코드를 활용한 승인방법

이 논문에서 제안하는 승인방법은 휴대폰과 QR코드를 활용한 승인방법으로 사용자의 휴대폰을 이용하기 때문에 추가적인 기기의 발급이 요구되지 않는다.

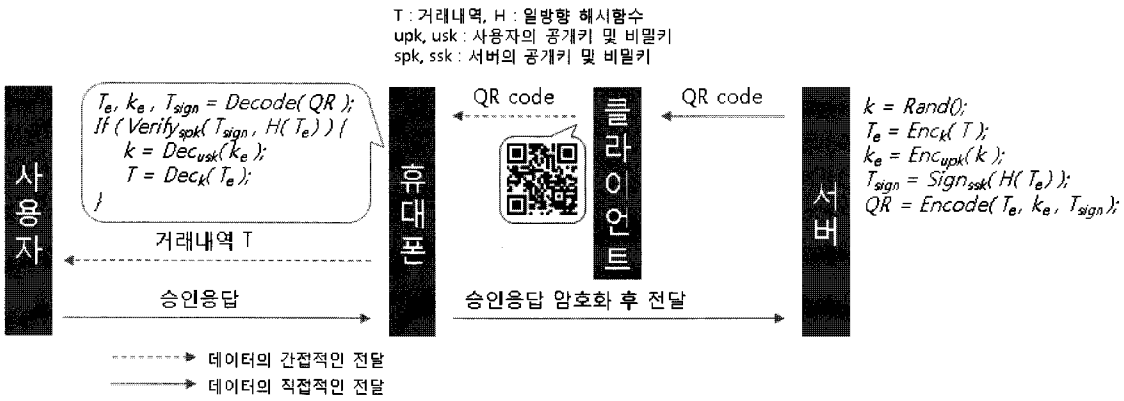
공인인증서와 모바일금융거래: 국내의 휴대폰 보급률은 2010년 3월말 처음으로 100% 돌파한데 이어 꾸준히 상승하고 있다[4]. 현재 휴대폰은 사용자 인증이나 휴대폰결제 등을 위한 SMS 승인번호 서비스, 각종 거래내역 확인 등의 용도로 활용되고 있다. 최근에는 휴대폰에서의 공인인증서를 이용한 모바일 뱅킹, 주식거래와 같은 모바일금융거래가 크게 활성화 되고 있으며 특히 모바일 뱅킹은 2010년 2/4분기에 일평균 이용건수와 금액은 262만건, 4,078억원으로 전분기대비 각각 13.2%, 14.0%의 증가율을 기록하였다[3]. 휴대폰에서 공인인증서의 사용이 가능하다는 것은 이를 이용하여 다양한 보안서비스를 제공할 수 있다는 것을 의미한다. 제안하는 기법에서는 사용자의 휴대폰에서 공인인증서를 이용할 수 있다고 가정한다.



〔그림 6〕 QR코드를 활용한 승인방법

휴대폰의 보안성: 휴대폰에도 악성코드가 침입할 가능성이 있는 만큼 휴대폰을 신뢰할 수 있는 장치라고 볼 수는 없다. 웹 브라우저 기반의 전자금융거래에서 스크립트 소스코드가 쉽게 노출되어 코드분석이 용이한 것과는 다르게 모바일금융거래는 컴파일 된 실행파일로 이루어져있어 디버그(Debug) 또는 역공학(Reverse Engineering)를 통해 코드를 분석한다.

예로, 해킹된 아이폰에서는 모바일금융거래가 동작하지 않도록 프로그램 되어있다. 이를 해킹된 아이폰에서 동작하게 하는 방법은 암호화되어있는 앱(App)이 실행 직전 복호화 된 것을 메모리 덤프(Memory Dump)하여 디버그 과정을 통해 해킹을 감지하는 함수를 건너뛰도록 하는 것이다. 이 과정에서 공격자는 단순히 해킹감지 함수를 무시하는 것이 아니라 악성코드를 삽입할 수도 있다. 따라서 아이폰의 경우 자체적으로 제공하는 앱스토어(Appstore)를 통해서만 전자금융거래 앱을 다운로드 해야 하며 안드로이드의 경우에도 신뢰할 수 있는 곳에서 앱을 설치해야 한다.



〔그림 7〕 QR코드를 활용한 승인방법

취약할 수 있음에도 불구하고 휴대폰을 이용한 방법을 제안하는 것은 사용자의 컴퓨터에 비해서 보안성이 떨어지지 않기 때문이다. 사용자의 컴퓨터의 경우 신뢰할 수 없는 프로그램(웹에서의 파일 다운로더, 동영상 코덱, 무료 프로그램, 블로그에서의 파일 다운로드 등)의 사용이 익숙하고 이를 통해 악성 프로그램이 설치될 수 있지만, 휴대폰의 경우 일반적으로는 신뢰할 수 있는 앱스토어를 통해서만 앱을 설치하도록 권고하고 있기 때문에 악성 프로그램의 침입이 비교적 어렵다고 할 수 있다. 따라서 전자금융거래의 승인과정에 휴대폰을 추가적으로 이용하는 것은 공격자의 공격시도를 사전에 차단하거나 공격 성공률을 낮추는 효과를 볼 수 있다.

QR코드의 활용: 휴대폰에 거래내역을 전달하는 방법으로는 QR(Quick Response)코드를 사용한다[12]. QR코드는 matrix형태의 바코드로 QR 스캐너 또는 카메라가 달린 휴대폰 등을 통해 읽을 수 있으며, 국내에서는 이를 활용한 마케팅 사례가 있다. 특히 몇몇의 스마트폰에서 QR코드를 해석하는 기능이 기본으로 탑재되면서 앞으로의 이용도가 높아질 것으로 기대된다. QR코드는 Reed-Solomon 에러교정 방식(8bits/codewords에 대해)을 사용하고 있으며 이에 따른 최대 데이터 표현양은 [표 3]과 같다. 이는 거래내역을 포함하기에 충분하다.

5.1. QR코드의 생성

QR코드에 표현될 데이터는 거래내역 T를 사용자 공인인증서의 공개키 upk를 이용하여 암호화 한 것이다. 전자금융거래 서비스의 종류에 따라 암호화할 데이터는 아래와 같이 추가한다.

거래내역을 사용자에게 전달하는 방법은 사람이 직접 개입되는 과정이니 만큼 최대한 인식능력을 이용하도록 한다. 여기서의 인식능력이란 사람의 이미지를 읽는 능력을 의미하며 문자열을 읽는 것보다 이미지를 인식하는 것이 편리하다는 사실을 이용하고자 함이다. 예로, 인터넷뱅킹 계좌이체의 경우 수신자의 전화번호를 거래내역에 포함시켜 휴대폰에 전송하면 이 전화번호를 통해 휴대폰에 등록되어 있는 수신자의 사진을 보여주어 사용자에게 수신자를 인식하도록 유도한다. 추가정보가 이미지인 경우에는 이미지를 인코딩(예, base64)하여 전송한다.

〔표 4〕 전자금융서비스에 따른 거래내역

전자금융거래	거래내역 T	
인터넷뱅킹 계좌이체	거래내역	이체금액, 수신자 은행명, 수신자 계좌번호, 수신자명 등
	추가정보	사용자 보안수단이 보안카드인 경우 이를 사용하기 위한 질문. (수신자가 정보제공에 동의한 경우)수신자에 대한정보가 서버에 존재하는 경우 수신자를 식별할 수 있는 부분정보(전화번호, 이메일 주소 등)
주식거래	거래내역	종목명, 종목코드, 주문유형, 주문수량, 주문단가, 수수료 등
	추가정보	거래하고자 하는 주식의 CI(Company Identity)
전자결제	거래내역	품목명, 가격, 수량, 구매옵션 등
	추가정보	품목의 이미지 또는 품목의 브랜드 이미지

5.2. QR코드를 활용한 승인방법

1) 서버는 거래내역 T를 임의적으로 생성한 대칭키 k를 이용하여 암호화 값 T_e, k를 사용자 공인인증서의 공개키 upk를 이용하여 암호화한 값 k_e, 그리고 T_e의 해시값을 서버의 비밀키 ssk로 서명한 T_{sign}을 QR코드로 표현하여 클라이언트에 전달한다.
2) 사용자는 휴대폰의 프로그램(또는 앱)을 구동시키고 공인인증서 패스워드를 입력한 이후에 카메라를 통해 QR코드를 인식한다.
3) 휴대폰은 T_{sign}을 서버의 공개키 spk와 T_e의 해시값을 통해 서명검증을 통과한 경우에만 usk를 이용하여 k를 복호화하고, 이 k로 T_e를 복호화한 다음 화면에 출력한다. 이때 T에 이미지가 포함되어 있다면 이미지 또한 출력하도록 한다.
   3-1) 사용자의 보안수단이 보안카드인 경우 보안카드 질의 또한 출력한다.
4) 사용자는 T를 확인하고 승인요청을 한다. 휴대폰은 승인요청을 사용자 공인인증서의 비밀키 usk와 서버의 공개키 spk를 이용, 암호화하여 서버에 전송한다.
   4-1) 사용자의 승인응답(보안카드나 OTP)이 있을 경우 이를 암호화하여 서버에 전송한다.
5) 서버는 수신된 응답을 확인하여 거래의 승인여부를 결정한다.

Ⅵ. MITB 공격 대응방법 비교

CAPTCHA는 사람과 로봇의 능력을 구분하여 자동화된 공격을 막기 위한 방법이지만 ArcotVPS와 MS워터마크는 이미지 처리를 통해 공격자가 여전히 자동화된 공격이 가능하다. 김성호 등이 제안한 기법의 경우에는 이미지 처리를 사용해도 CAPTCHA를 읽지 못하는 이상 보안성이 떨어지지는 않는다. 또한 CAPTCHA의 문자열을 인식하고 입력하는 것이 아니라 선택하는 형태이기 때문에 더욱 복잡도가 높은 CAPTCHA를 사용할 수 있다는 것이 장점이다. 하지만 공격자가 직접 공격에 실시간으로 참여하는 경우에는 CAPTCHA가 제 역할을 하지 못하기 때문에 보안성을 기대할 수 없다.

반면에 부가장치와 같은 접근 방법은 공격자가 실시간

〔표 5〕 MITB 공격에 대응하는 방법들의 비교

부가장치	전화 승인 서비스	보안성	사용자의 전화기와 그 통신 채널의 보안성에 의존한다. 사용자가 구리선을 이용한 유선전화기를 사용하는 경우에는 공격자가 ARS응답을 사용자 대신 할 수 있기 때문에 MITM 공격에 취약하다.
		편의성	사용자가 가진 전화기를 사용하여 추가적인 장치가 요구되지 않고 ARS 확인 후 승인여부만을 판단하면 되기 때문에 편리하다.
	트랜잭션 서명 기법 [2,5]	보안성	임형진 등의 연구[2]에서는 공인인증서를 사용하여 부인방지를 제공하지만 결과적으로는 추가적으로 발급한 장치의 보안성에 의존한다.
		편의성	추가적인 장치를 소지해야 하고 사용자가 부분적인 거래내역을 장치에 입력해야 하기 때문에 불편한 측에 속한다.
	ZTIC [14]	보안성	추가적으로 발급한 장치의 보안성에 의존한다.
		편의성	추가적인 장치를 소지해야 하지만 화면을 통해 거래내역 확인 후 승인여부를 판단하면 되기 때문에 편리하다.
	제안 방법	보안성	사용자 휴대폰의 보안성에 의존한다.
		편의성	사용자가 가진 휴대폰을 사용하기 때문에 추가적인 장치가 요구되지 않고 화면을 통해 거래내역 확인 후 승인여부를 판단하면 되기 때문에 편리하다.
CAPTCHA	Arcot VPS [13]	보안성	배경과 문자열의 색이 다르고 이미지에서 문자가 차지하는 비율이 낮다는 사실을 근거로 Kmeans++를 사용하여 문자열을 분리시킬 수 있다. 악성 프로그램이 캡차를 읽을 수 없더라도, 추출된 4개의 문자열 중 하나를 정해 추출한 다음 변조된 거래내역에 대한 승인이미지를 생성할 수 있다. 따라서 악성 프로그램은 CAPTCHA를 읽을 수 없어도 1/4의 확률로 공격에 성공할 수 있다.
		편의성	캡차를 인식하고 사용자가 수동적으로 입력해야 하기 때문에 편의성이 높지는 않다.
	MS 워터 마크 [7]	보안성	MS워터마크는 일반폰트로 거래내역이 표현된 배경 위로 CAPTCHA가 표현되어 있다. CAPTCHA의 폰트가 더욱 두껍고 크기 때문에 윤곽선 검출(Edge Detection)의 Canny 필터(가우시안 마스크를 통해 잡영 제거 후 Sobel필터 적용)를 통해 강한윤곽선만을 남긴 후에 이를 추출해내면 CAPTCHA 문자열을 분리시킬 수 있다. 특정 두께에 대한 영역을 세선화(Thinning)하여 문자열을 얻어낸 다음이 이를 다시 Blur 필터 등으로 확장하면 더욱 선명한 문자열을 얻어낼 수 있다. 분리된 캡차 이미지는 변조된 거래내역에서 악용될 수 있다.
		편의성	캡차를 인식하고 사용자가 수동적으로 입력해야 하기 때문에 편의성이 높지는 않다.
	김성호 등의 연구 [1]	보안성	6개의 콘텐츠(수신은행, 수신계좌, 수신자명, 이체금액, 송신자, 송신은행)와 44개의 더미 캡차로 구성되어 있을 경우, 무차별 선택에 대한 보안성은 $1/(50\times\cdots\times45)=8.74\times10^{-11}$이다. 공격이 가능한(사용자가 입력한 수신은행과 이체금액은 그대로 사용) 최소한의 콘텐츠 변경은 수신계좌와 수신자명 두 가지를 변경하는 것으로, 이때의 보안성은 $1/(50\times49)=4.08\times10^{-4}$이다.
		편의성	사용자는 여러 개의 캡차를 인식해야 하기 때문에 편의성이 높지는 않다. 하지만 사용자가 CAPTCHA를 읽고 입력하는 형태가 아닌 선택하는 용도로 사용하였기 때문에 더욱 복잡도가 높은 CAPTCHA를 이용할 수 있다. 예로, 심하게 뒤틀려 일부 글자를 인식하기 힘든 형태가 되어도 CAPTCHA를 인식하고 선택하는 데에는 지장이 없다.

으로 공격에 참여하고 있어도 장치를 제어하지 못하는 이상 부정승인을 완료할 수 없기 때문에 MITB와 같은 공격에서 CAPTCHA에 비해 안전하다. 트랜잭션 서명기법은 MITB에 효과적으로 대응할 수 있지만 거래내역에 대한 부분정보를 장치에 입력해야 하기 때문에 편의성이 떨어진다. ZTIC은 디스플레이 장치로 거래내역을 확인하고 두 개의 버튼을 통해 승인여부를 결정하도록 선택하도록 되어있어 편의성이 뛰어나다. 하지만 추가적인 장치발급으로 인한 비용은 여전히 부담으로 남는다.

MITB에 대응하는 기법이 실용화에 주로 영향을 미치는 것은 사용자의 편의성과 추가적인 기기의 발급비용 부담이라는 점에서, 제안하는 승인방법은 보급률이 뛰어난 사용자의 휴대폰을 이용하기 때문에 추가적인 장치의 발급이 필요하지 않고 거래내역 확인 후 승인여부만을 결정하면 되기 때문에 편의성이 좋다고 할 수 있다.

Ⅶ. 결 론

이 논문에서는 MITB를 이용한 문서변조 공격과 그 대응방안에 대해 논의하였으며 QR코드를 활용한 승인방법을 제안하였다. 제안하는 방법은 보급률이 높은 휴대폰을 이용하여 추가적인 기기발급으로 인한 부담이 없고, 휴대폰에 QR코드 인식을 통해 거래내역을 전달하기 때문에 장치에 거래내역의 일부를 수동적으로 입력해야 하는 트랜잭션 서명기법에 비해 사용자 편의성이 높다.

휴대폰에도 악성코드가 잠입할 가능성이 있는 만큼 제안하는 방법에 약점이 없는 것은 아니다. 하지만 소스코드를 숨기는 것이 불가능하고 태생적으로 수정이 허용되는 스크립트를 중심으로 서비스되는 인터넷뱅킹에 비해 안전성이 떨어지지는 않는다. 더해서 승인과정에 휴대폰을 이용하도록 하는 것은 공격자에게 두 시스템을 모두 공격해야 한다는 부담을 주어 공격시도를 사전에 차단하거나 공격확률을 낮추는 효과를 기대할 수 있다. 그러므로 제안하는 방법은 전자금융거래에서 하나의 보안(승인)수단이 될 수 있을 것이라 기대한다.

참고 문헌

[1] 김성호, 강전일, 김기태, 양대헌, "문맥 기반의 캡차를 이용한 신뢰성 있는 인터넷 계좌 이체 방법", *한국정보보호학회 동계학술대회*, 제19권 제2호, 319-323, 2009

[2] 임형진, 이정근, 김문성, "안전한 인터넷 뱅킹을 위한 트랜잭션 서명기법에 관한 연구", *인터넷정보학회논문지*, 제9권 제6호, 73-79, 2008

[3] 한국은행, "2010년 2/4분기 국내 인터넷뱅킹서비스 이용현황", http://bok.or.kr/contents/total/ko/boardView.action?boardBean.categorycd=0&boardBean.sdt=&boardBean.edt=&boardBean.searchColumn=title&boardBean.searchValue=&menuNaviId=559&boardBean.menuid=559&boardBean.brdid=72506&boardBean.rnum=24&boardBean.cPage=3, 2010

[4] 한국통신사업자연합회, http://stat.ktoa.or.kr/default_client.asp, 2010

[5] A. Hiltgen, T. Kramp and T. Weigold, "Secure Internet Banking Authentication", *IEEE Security and Privacy*, 2006

[6] D. Sharek, C. Swofford and M. Wogalter, "Failure to Recognize Fake Internet Popup Warning Messages", *PROCEEDINGS of the HUMAN FACTORS AND ERGONOMICS SOCIETY 52nd ANNUAL MEETING*, 557-560, 2008

[7] D.J. Steeves and M.W. Snyder, "Secure online transactions using a CAPTCHA image as a watermark", US 2007/0005500 A1, US Patent, Jan. 2007

[8] Gühring Philipp (2007). "Concepts against Man-in-the-Browser Attacks". http://www2.futureware.at/svn/sourcerer/CAcert/SecureClient.pdf

[9] NVHServer, http://www.nvhserver.com/

[10] Obfuscated code, http://en.wikipedia.org/wiki/Obfuscated_code

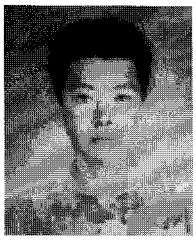
[11] P. d. Barros and Augusto, "O futuro dos backdoors-opior dos mundos", http://www.paesdebarros.com.br/backdoors.pdf, 2005

[12] QR code. http://en.wikipedia.org/wiki/QR_Code

[13] R.A. Gopalakrishna, "Authentication using a turing test to block automated attacks", US 2009/0199272 A1, US Patent, Aug. 2009

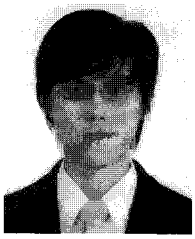
[14] ZTIC, http://www.zurich.ibm.com/ztic/

〈著者紹介〉



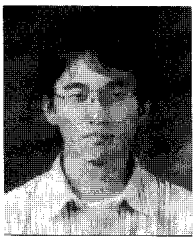
맹 영 재 (YoungJae Maeng)
학생회원

2006년 8월: 인하대학교 컴퓨터 공학과 졸업
2008년 8월: 인하대학교 정보통신대학원 석사
2008년 9월~현재: 인하대학교 정보공학과 박사 과정
<관심분야> 인터넷 보안, 네트워크 보안



신 동 오 (DongOh Shin)
학생회원

2010년 2월: 인하대학교 컴퓨터 공학과 졸업
2010년 3월~현재: 인하대학교 정보공학과 석사 과정
<관심분야> 인터넷 보안, 네트워크 보안



김 성 호 (Sung-Ho Kim)
학생회원

2009년 2월: 인하대학교 컴퓨터 공학과 졸업
2009년 9월~현재: 인하대학교 정보통신대학원 석사 과정
<관심분야> 시스템 보안, 네트워크 보안

양 대 헌 (DaeHun Nyang)
종신회원

1994년 2월: 한국과학기술원 과학기술대학 전기 및 전자 공학과 졸업
1996년 2월: 연세대학교 컴퓨터 과학과 석사
2000년 8월: 연세대학교 컴퓨터 과학과 박사
2000년 9월~2003년 2월: 한국전자통신연구원 정보보호연구본부 선임연구원
2003년 2월~현재: 인하대학교 정보통신대학원 부교수
<관심분야> 암호이론, 암호프로토콜, 인증프로토콜, 무선 인터넷 보안