情報保護學會論文誌
第 21 卷 第 1 號, 2011. 2

모바일 뱅킹에서 비밀퍼즐을 이용한 비밀증명방법과 거래승인방법*

맹 영 재[1†], 양 대 헌[1], 이 경 희[2‡]
[1]인하대학교, [2]수원대학교

Password Authentication and Transaction Confirmation Method Using Secret Puzzle on Mobile Banking*

YoungJae Maeng[1†], DaeHun Nyang[1], KyungHee Lee[2‡]
[1]INHA University, [2]University of Suwon

요 약

모바일뱅킹에서 사용자인증과 거래승인을 보호하는 것은 매우 중요하다. 스마트폰에 설치된 악성 프로그램은 사용자가 입력하는 비밀을 얻어내거나 거래내용을 조작하여 사용자에게 승인하도록 유도할 수도 있다. 이 논문에서는 사용자의 비밀과 거래승인을 보호하기 위한 연구들의 보안성과 편의성을 분석하고, 이를 바탕으로 사용자의 비밀을 보호하고 문서의 내용을 조작하는 공격에 대응하는 방법을 제안한다.

ABSTRACT

Securing user authentication and transaction confirmation is very critical in mobile banking. Malicious software, which is installed in user's smart phone, can either steal user's password or induce user to confirm manipulated transaction by handling transaction resource. In this paper, we propose schemes, that are aimed to secure user's password or to secure transaction confirmation, based on the security and usability analysis of existing schemes.

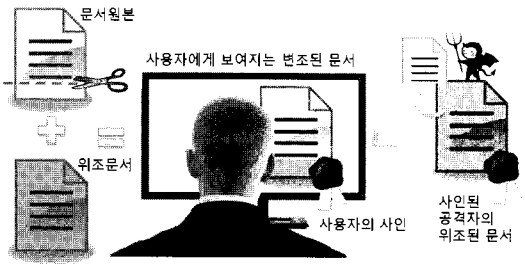
Keywords: MITB, Password Authentication, Transaction Confirmation, Secret Puzzle, Mobile Banking

I. 서 론

스마트폰과 태블릿PC(이하 스마트폰)가 보급화 되면서 이러한 휴대기기와 관련한 보안이슈 또한 증가하고 있다. 이들 기기의 공통점은 대부분이 과거 모바일 기기에 비해 비교적 높은 성능과 디스플레이 해상도 그리고 무선랜 또는 3G 무선통신을 기본적으로 탑재하고 있으며, 그러한 기능들을 활용할 수 있도록 해주는 각종 어플리케이션(이하 앱)을 손쉽게 설치하고 관리할 수 있도록 되어있다는 점이다.

하지만 스마트폰은 일반적인 컴퓨터에서와 마찬가지로, 사용자의 의도와는 다르게 악의적인 코드가 설치되고 실행될 수 있다. 이 악성 프로그램은 스마트폰에 담긴 연락처, 사진, 소셜 네트워크 데이터, 위치정보 등과 같은 사생활 정보를 얻어내려 하거나 봇(Bot)형태로 이 기기가 가진 기능을 악용하는 등, 컴퓨터에서 가능했던 공격뿐만 아니라 기기에 저장된 사생활 정보 또는 사용자의 권한을 이용하여 더욱 지능적인 공격 또한 시도할 수 있다. 그러한 공격은 전자금융거래에 대한 공격도 포함한다.

접수일(2010년 12월 3일), 게재확정일(2011년 2월 1일)
* 이 논문은 2010년도 정부(교육과학기술부)의 재원으로 한국연구재단의 기초연 구사업 지원을 받아 수행된 것입니다.(2010-0013254)
† 주저자, brendig@isrl.kr
‡ 교신저자, khlee@suwon.ac.kr



〔그림 1〕 사용자에게 보여지는 변조된 문서

모바일 기기에서의 전자금융거래(이하 모바일뱅킹)는 기존의 IC칩 방식, VM 방식과 스마트폰에서 사용되는 앱 방식으로 나누어진다. 한국은행의 2010년 3/4분기 국내 인터넷뱅킹서비스 이용현황에 따르면 모바일뱅킹 이용실적이 전체 인터넷뱅킹에서 차지하는 비중이 지속적으로 상승하고 있으며, 특히 앱 방식의 모바일뱅킹 사용자가 전분기 대비 153.5% 증가하였다고 한다〔1〕.

인터넷뱅킹이 그러하듯이, 모바일뱅킹 또한 비대면 거래이기 때문에 사용자 인증과 거래승인 과정을 보호하는 것이 매우 중요하다. 이를 위해 다양한 형태의 비밀과 보안 프로토콜, 보안 프로그램으로 사용자의 전자금융거래를 보호하고 있으나 그럼에도 불구하고, 문서를 변조하여 공격자가 의도한 형태로 거래를 완료시키는 공격에 취약하다는 것이 맹영재 등의 연구〔2〕에 의해 분석된 바 있다.

문서변조는 악성 프로그램이 거래내용을 조작하지만 화면에는 거래내용이 조작되지 않은 것처럼 보이도록 하여 사용자로 하여금 조작된 거래를 완료하도록 유도하는 공격이다 〔그림 1〕. 사용자가 입력하는 비밀을 그대로 이용하고 거래가 완료되기까지 시각적인 속임수를 이용하여 현재의 전자서명과 보안프로그램을 무색하게 만들었다는 점에서 지능화된 공격이라 할 수 있다. 특히 계좌이체와 같은 거래를 승인할 때는 고정된 비밀의 단점을 극복하기 위해 보안카드(질의-응답 형태로 이용)나 매번 다른 응답을 생성해내는 OTP(One-Time-Password)와 같이 소유하는 형태의 비밀(이하 승인수단)을 추가적으로 이용하도록 하였음에도, 이는 비밀의 소유를 증명하는 것이 목적이기 때문에 거래내용의 조작 여부와는 무관하게 승인이 될 수 있다.

만약 스마트폰에 악의적인 명령을 가진 앱이 유입된다면 위의 공격이 모바일뱅킹에서도 동일한 형태로 발생할 수 있게 된다. 맥아피(McAfee)가 발표한 2010년 4분기 위협보고서에 따르면 2010년에 새롭게 등장한 모바일 멀웨어의 수는 2009년 대비 46%증가하였다고 한다〔24〕. 이 논문에서는 그러한 공격에 대한 대응하기 위해 안전한 비밀입력 방법과 거래 승인방법에 대한 요구사항을 알아보며, 이를 바탕으로 비밀을 복합적으로 사용하는 비밀퍼즐과 CAPTCHA를 응용한 안전한 서명퍼즐을 제안한다.

이 논문의 2장에서는 현재의 비밀이용 방법이 가지는 문제점의 도출 및 공격자를 가정하고, 3장에서는 비밀을 보호하기 위한 연구와 거래승인을 보호하기 위한 연구를 소개한다. 이 연구들의 분석을 바탕으로, 4장에서는 안전한 비밀입력 방법인 비밀퍼즐을, 5장에서는 부정승인 방지기법인 서명퍼즐을 제안한다. 6장에서는 다른 연구들과의 비교를 보이며 7장에서는 결론을 담는다.

II. 비대면 거래에서의 인증과 승인, 그리고 공격자

비밀은 용도에 따라서는 사용자 인증을 위한 것과 거래 승인을 위한 것, 공격자의 입장에서는 고정된 형태와 유동적인 형태로 나누어 볼 수 있다. 사용자 인증과정에는 고정된 형태의 비밀만이 사용되고 있는데, 고정된 형태의 비밀은 한번 노출되면 사용자가 해당 비밀을 바꾸기 전까지 공격자가 악용할 수 있다는 단점이 있다. 그 때문에 사용자 인증만으로 할 수 있는 것은 조회서비스로 한정되고, 그 외의 중요정보 조회나 정보변경, 거래승인 등에서는 매번 다른 응답(이하 승인응답)을 입력하도록 하는 소유하는 형태의 비밀을 이용하였다. 매번 달라지는 응답은 분명 공격자에게 부담이 되는 것이다.

2.1 공격자 가정

이 논문에서의 공격자는 사용자의 컴퓨터에 악성 프로그램을 설치할 수 있다고 가정한다. 국내의 전자금융거래에서 악성 프로그램을 의식하여 보안 프로그램을 설치하고 있는 만큼, 그러한 가정은 현실적으로도 받아들여지고 있는 가정이다. 악성 프로그램은 공격자가 사전에 지시해놓은 자동화된 행동을 할 수 있으며, 설치된 보안 프로그램을 무력화 할 수는 없다고 가정한다. 또한 이 악성 프로그램은 일종의 자동화된 프로그램이기 때문에 공격자(사람)가 실시간으로 공격에 참여하지는 않으며 따라서 CAPTCHA와 같은

문제는 스스로 해결할 수 없다고 가정한다.

2.2 비밀전달방법의 취약점

위의 모든 비밀의 전달방식은 평문을 입력하는 것이다. 그 비밀을 얻어내기 위해 공격자는 키보드 후킹(Hooking)과 같은 공격을 시도하거나 피싱(또는 파밍)공격을 통해 공격자가 만든 입력 폼에 비밀을 입력하도록 유도한다. 키보드 보안 프로그램이나 피싱방지 프로그램은 그러한 공격들을 어느 정도 방지하지만, 근본적인 해결방안이 되지는 못한다. 예로, 문서변조와 같은 공격에서는 사용자의 비밀을 보호하지 못한다[2,23]. 공격자는 키보드 보안 프로그램이 보호대상이 아닌 입력 폼까지 보호하지는 않는다는 사실을 악용하여, 자신이 생성한 입력 폼을 화면에 보이도록 하고 사용자에게 비밀을 입력하도록 유도할 수 있다.

2.3 거래승인방법의 취약점과 부인방지

인터넷뱅킹에서는 거래승인을 위해 승인수단과 공인인증서를 동시에 사용한다. 승인수단은 소유하는 형태의 비밀로, 4~6자리의 임의적인 숫자로 이루어져 있으며 인터넷뱅킹에서 사용되는 비밀 중 유일하게 동적인 형태이다. 승인수단이 제공하는 보안성은 $10^4 \sim 10^6$정도로, 암호학 측면에서는 낮은 편이지만 무차별 공격 등을 대응하는 것에는 효과적인 수준으로 받아들여지고 있다. 승인수단과 함께 사용되는 공인인증서 서명은 공개키 암호수준에서의 부인방지를 제공한다. 공인인증서 파일은 사용자가 정한 패스워드로 대칭 암호화되어 보관되기 때문에 공인인증서를 사용하기 위해서는 매번 사용자가 정한 패스워드를 입력해야 한다.

그럼에도 불구하고, 악성 프로그램이 승인수단과 공인인증서를 동시에 악용하는 방법은 크게 두 가지로 나뉜다. 첫 번째 방법은 2.2장에서 언급한 방법 등을 통해 승인응답과 공인인증서 패스워드를 얻어내고 이를 실시간으로 악용하는 것이다. 두 번째 방법은 공격자가 위의 두 가지 비밀을 얻어내는 것이 목적이 아니라, [그림 1]에서와 같이 비밀 입력 폼들은 그대로 두고 거래내용만을 조작하여 사용자가 입력한 비밀들이 공격자의 의도대로 사용되도록 하는 것이다. 이러한 취약점들은 승인응답이 거래내용과 연관되어 있지 않기 때문에 발생한다. 거래내용 출력화면이 악성 프로그램에 의해 조작될 수 있는 구간이 다수 존재하는 반면에, 현재 제공되고 있는 보안 프로그램들은 출력화면을 조작하는 공격들을 대응하고 있지 못하는 실정이다.

특히 공인인증서 서명이 악용될 수 있다는 사실은, 부인방지 보안 서비스가 사용자를 오히려 곤란한 상황에 이르게 할 수 있다는 것을 뜻한다. 사용자의 비밀키로 서명된 조작된 거래는 적어도 서버 측에서 발생한 문제는 아니기 때문에 서버(back-end)에서는 조작 여부를 확인할 수 없으며, 이는 사용자의 컴퓨터에서 악성 프로그램이 발견되기까지 사용자의 책임일 수 있다는 것을 의미한다. 문제는 악성 프로그램이 발견되더라도 조작된 거래가 발생한 이후의 대책을 위한 증거 수준에 그친다는 점이며, 만약 이 증거가 거래취소사유 등에 해당한다면 이러한 과정자체를 악용하는 사례가 발생할 수 있게 된다.

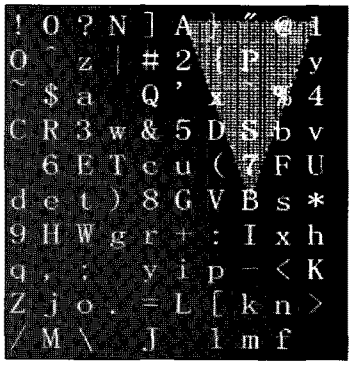
III. 비밀보호 또는 승인방법 연구동향 및 분석

사용자 인증을 보호하기 위한 연구는 주로 비밀의 종류나 그 이용방법을 바꾸어 비밀이 노출되기 어렵도록 한다. 거래승인을 보호하기 위한 연구는 자동화된 공격을 방지하기 위해 CAPTCHA를 응용하는 방법과 부가장치를 이용하는 방법으로 나눌 수 있다. 3.1장은 사용자의 인증을 보호하는 방법을, 3.2장부터 3.3장은 거래승인 보호와 관련한 연구를 소개하고 분석한다.

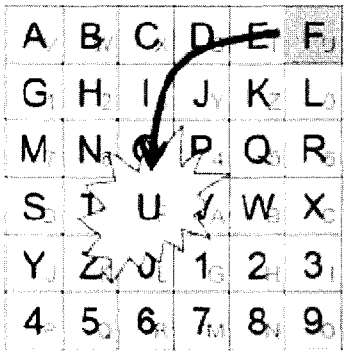
3.1 비밀증명 방법에 대한 연구

비밀증명 방법들이 가정하는 공격자는 어깨너머 훔쳐보기(Shoulder Surfing)로, 사용자의 인증과정을 사용자 뒤에서 육안으로 또는 녹화장치 등으로 훔쳐보고 비밀을 얻어낸다. 이 공격은 비밀이 시스템에 입력되는 과정을 훔쳐보는 것으로, 악성 프로그램보다 한층 더 강력한 공격이다. 다시 말해, 비밀증명 방법에 관한 연구들은 악성 프로그램에도 강한 내성을 가지므로 그러한 방법들을 분석하여 볼 필요가 있다.

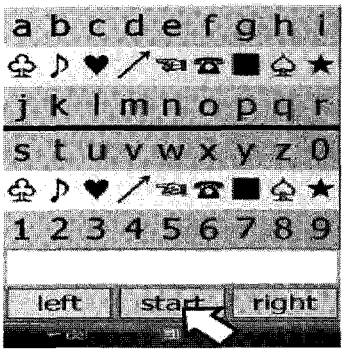
강문설 등이 제안한 기법은 어깨너머 훔쳐보기 공격자를 육안으로만 비밀입력과정을 훔쳐보는 사람으로 한정하고, 그 공격에 대응하는 방법에 대해 연구하였다[3]. 키 배열을 매번 다르게 나타나도록 하고 입력과정에서는 키가 보이지 않도록 한 것으로, 비밀을 아는 사용자는 비밀이 위치한 위치만을 기억하면 되지만 공격자는 화면에 나타난 모든 키의 위치를 기억하



〔그림 2〕 S3PAS
(A1B로 만든 삼각형내의 문자선택)



〔그림 3〕 행렬 상에서의 인증
(교차점 F를 임시비밀 U에 위치시킴)



〔그림 4〕 모바일에서 패스워드 인증방법
(비밀을 pass-object에 위치시킴)

고 있어야 한다는 정보의 차이를 이용한 기법이다. 이 기법은 육안으로 비밀을 얻어내려는 공격을 효과적으로 차단하지만, 현실적으로는 공격자가 녹화장치를 이용하는 것이 간편해졌기 때문에 어깨너머 훔쳐보기 공격자를 가정할 때 녹화장치 또한 고려되어야 한다(녹화장치의 예로, 대부분의 휴대폰에 녹화 가능한 카메라가 기본으로 탑재되어 있음).

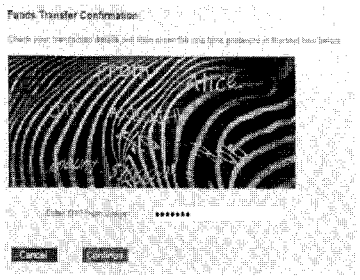
어깨너머 훔쳐보기에 안전한 비밀증명 방법들은 질의-응답(challenge-response) 방식을 사람이 참여할 수 있는 형태로 만든 것이라고 할 수 있다. pass-icon〔4〕은 화면에 보이는 아이콘들(질의에 해당) 중에서 자신이 외운 아이콘 세 개를 꼭짓점으로 가지는 삼각형을 머릿속으로 그리고 그 삼각형 안의 아이콘을 선택(응답에 해당)하도록 하였다. 아이콘이 임의적인 순서로 나타나는 pass-icon과는 다르게 S3PAS〔5〕〔그림 7〕는 사용자가 외우는 n자리의 패스워드 $P=p_1, p_2, \cdots, p_n$를 i번째 인증 단계에서 세 자리씩 $p_i, p_{(i+1)\%n}, p_{(i+2)\%n}$ 잘라 사용하였다. 하지만 앞과 뒤 라운드에서 두 자리의 패스워드가 동시에 사용된다는 점은, 사용자가 선택한 문자를 중심으로 삼각형을 재구성하는 패스워드 후보군의 추측 가능성을 높이게 된다. 공격자가 이 패스워드 후보군을 3개 모으면 교차공격(비밀 후보군들의 교집합)을 통해 패스워드를 알아낼 수 있다는 분석이 신동오 등의 연구〔6〕에서 연구된 바 있다.

강전일 등의 연구〔7〕〔그림 3〕에서는 두 겹으로 전달되는 6x6행렬(알파벳 26개, 숫자 10개로 이루어져 있으며 초기단계에서는 두 겹의 행렬에 표시된 문자가 같다, 질의에 해당)을 서버가 전송하는 질의로 사용하고, 이 행렬상의 움직임(응답에 해당)을 통해 인증하는 방법을 소개하였다. 인증은 총 n(패스워드의 길이)단계이며 사용자는 홀수 번째 인증단계에서 임시비밀을 행렬 상에서 임의적으로 선택하고 짝수 번째 인증단계에서는 이전(홀수 번째) 인증단계에서 선택한 임시 비밀을 이용하도록 되어있다. n번째 인증단계에서 사용자는 행렬 상에서 n번째 패스워드가 위치한 행과 $(n+1)\%n$번째 패스워드가 위치한 열이 겹치는 교차점을 찾고, 이 교차점(상위계층)을 임시비밀(하위계층)에 방향키로 이동하여 위치시킨다. 이 기법은 무차별 공격과 재전송 공격, 교차공격을 적절히 방어하지만 머릿속에서 패스워드를 두 자리씩 잘라 기억해내고 임시비밀 또한 기억해야 한다는 점은 사용자에게 부담이 될 수 있다.

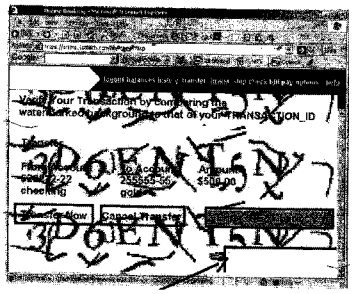
김창순 등의 연구〔8〕〔그림 4〕에서는 사용자에게 안전한 채널을 통해 질의전달이 가능한 경우, 첫 번째 패스워드 문자와 같은 열에 위치하는 pass-object를 임시비밀로 사용하는 경우, 그리고 안전한 채널을 통해 pass-object를 전달받는 경우에 대한 인증기법을 보였다. 방향버튼을 통해 비밀을 pass-object에 위치시키는 형태이며 이 기법 또한 적절한 보안성을 가진다. 하지만 첫 번째와 세 번째 기법의 경우는 질의를 전달할 수 있는 안전한 채널이 요구되는데, 악성 프로그램을 가정했을 때 안전한 채널을 가정하는 것은 쉽지 않은 문제이다. 예로, 음성을 통해 질의를 전달하는 방법은 악성 프로그램이 음성인식 기법을 통해 질의를 알아낼 수 있다.

3.2 CAPTCHA를 이용한 부정승인 방지방법

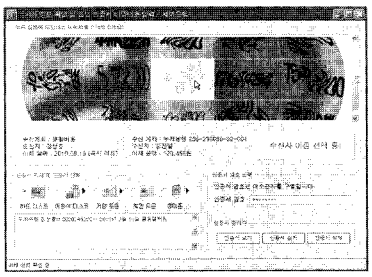
CAPTCHA는 무작위한 회원가입이나 스팸메일과 같은 자동화된 악의적인 행동을 막기 위해 사용된다. 문서변조 공격이 자동화된 공격이라는 점에서



〔그림 5〕 ArcotVPS



〔그림 6〕 MS워터마크



〔그림 7〕 문맥기반의 CAPTCHA

CAPTCHA를 응용하면 그 자동화된 공격을 방지하기 위한 용도로도 사용될 수 있다. 여기서 유의해야 할 점은 일반적인 형태의 CAPTCHA가 문서변조 공격 앞에서는 무의미하다는 사실이다. 조작된 거래를 인지하지 못하고 승인하려는 사용자가 직접 CAPTCHA마저 응답할 수 있기 때문이다. 대신에, CAPTCHA의 내용에 거래내용이 포함되도록 하면 사용자가 이에 응답할 때 그 내용을 확인하게 되기 때문에 문서변조 공격에 대응할 수 있게 된다.

ArcotVPS[9]의 경우 거래의 승인단계에서 거래내용과 OTP가 하나의 CAPTCHA로 표현된 이미지를 사용자에게 보여주고 이 중에서 OTP를 입력하도록 하여 거래내용이 변조된 경우 사용자가 인지할 수 있도록 하였다[그림 5]. MS워터마크[10]는 거래내용이 출력 된 화면상에 CAPTCHA를 겹쳐진 형태로 출력시켜 사용자가 CAPTCHA에 응답할 때 거래내용 또한 확인할 수 있도록 하였다. 위의 두 기법이 CAPTCHA에 표시된 문자를 입력하도록 한 것과는 다르게, 문맥기반의 CAPTCHA[11]는 CAPTCHA에 표현된 문자 중에서 거래내용과 관련된 것만을 선택하는 형태로 CAPTCHA문제를 바꾸었다.

선택하는 형태의 CAPTCHA가 적용된 환경에서 거래내용을 수정하려면, 공격자는 CAPTCHA에서 자신이 변조한 내용에 해당하는 부분을 사용자가 의도했던 내용으로 바꾸어 놓는 작업을 해야 한다. 이때 공격자는 CAPTCHA를 해석하는 것이 불가능하고 가정하기 때문에 조작할 대상을 임의적으로 선택하여 내용을 바꾸는 수밖에 없다. 예로, 문맥기반의 CAPTCHA가 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.

한편, 국내 인터넷뱅킹에서 거래를 완료하기 위해서는 승인응답을 서버에 전달해야 한다. 위의 기법 중 하나를 적용하여 계좌이체를 보호하였다 하더라도 승인응답을 입력받는 것은 별개의 과정인 것이다. 이때 악성 프로그램은 승인응답을 얻어내 승인응답이 필요한 다른 용도에 악용할 수 있게 된다. 예로, 악성 프로그램은 서버에 인증된 상태의 사용자 권한을 이용하여 위와 같은 CAPTCHA가 적용되어있지는 않지만 승인응답이 필요한 메뉴를 이용할 수 있다.

3.3 부가장치를 이용한 거래승인방법

부가장치를 이용한 거래승인방법은 악성 프로그램의 존재로 신뢰할 수 없는 모니터의 화면대신 비교적 안전한 부가장치를 통해 거래내역을 확인하고 승인할 수 있도록 한다. 현재 사용 중인 부가장치를 이용한 유일한 승인방법은 인터넷뱅킹에서의 전화승인서비스로, 계좌이체 승인단계에서 서버가 ARS 시스템을 이용해 사용자에게 전화를 걸어 거래내역을 확인시켜 주고 거래를 승인하거나 취소할 수 있도록 하고 있다.

트랜잭션 서명 기법은 사용자 토큰이라 불리는 추가적으로 발급받은 기기에 거래내역의 일부를 입력하도록 하여 거래내용을 사용자에게 다시 한 번 인지시켜주고 이 결과로 토큰에 출력된 MAC을 컴퓨터에 입력, 서버에 전달하도록 하여 잘못된 거래내용이 승인되는 것을 방지하였다[12]. 이 연구에 기초한 임형진 등의 연구[13]는 사용자 토큰에 인증서를 저장한 형태 또는 사용자가 입력한 MAC을 컴퓨터에서 공인인증서를 통해 사인하는 방법으로 부인방지 보안서비스를 추가적으로 제공한다. 위 두 기법의 승인방법-추가

적으로 발급받은 기기에 거래내용의 일부분을 입력하고 그에 대해 생성된 응답을 사용자의 컴퓨터에 입력하는 것-은 발급받은 기기와 사용자의 상호작용을 통해 거래내용을 확인할 수 있도록 하여 보안성을 한층 강화하였지만, 발생하는 거래의 수만큼 거래내용의 일부분을 입력하는 작업을 반복해야 한다는 점은 단점으로 남는다.

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 하였다. 디스플레이에 출력된 내용 확인 후 승인여부만을 결정하면 되기 때문에 위의 기법들보다 편의성이 높아 비교적 실용성이 높은 거래승인 방법이 될 수 있을 것이다.

하지만 거래승인만을 위해서 이러한 장치를 추가적으로 발급하는 것은 부담이 될 수 있고 모바일 뱅킹에서는 컴퓨터에서의 USB와 같은 통일된 규격단자가 제공되고 있지 않기 때문에 모바일뱅킹에서 ZTIC을 적용하는 것은 어려움이 있어 보인다. 무선랜이나 블루투스 같은 무선통신기술을 이용한 방법을 생각할 수 있지만 무선통신장치 탑재로 인한 기기발급 비용 상승과 통신에 소요되는 배터리 등 고려해야 할 사항들이 추가적으로 발생한다.

IV. 비밀을 복합적으로 이용하는 비밀증명방법

이 장에서는 3장에서의 분석을 바탕으로 비밀을 복합적으로 이용하는 비밀증명방법을 소개한다.

4.1 비밀증명방법 고려사항

교차공격에 비교적 강한 강전일 등의 연구[7]와 김창순 등의 연구[8]는 공통적으로 사용자에게 임시비밀을 정하도록 하고 이것을 비밀에 위치시키는 등의 연산을 사용한 것을 알 수 있다. 이는 마치 질의-응답 프로토콜에서의 난수와 해시함수를 연상케 하는데, 이 두 요소의 사용대상이 사람일 때는 다음과 같은 사항들을 고려해야 한다.

4.1.1 난수를 대체하기 위한 임시비밀

프로토콜에서 난수는 예측할 수 없는 결과를 만들어내고 재전송 공격 등을 방지하기 위해 사용된다. 난수 대신 사용되는 임시비밀 또한 매번 다른 응답이 생성되도록 하여 재전송공격에 대비하는 필수불가결한 요소이므로 임의성을 가져야한다. 하지만 사람에게 임시비밀을 정하도록 하는 것은 아래와 같은 어려움이 있다.

- 긴 문자를 외우기가 쉽지않은 사용자: Miller의 연구에 따르면 사람이 가장 잘 기억할 수 있는 알파벳-숫자의 숫자는 7에서 2를 빼거나 더한 수인 5~9이라고 한다[18]. 따라서 이보다 더 긴 임시비밀의 길이와 개수는 편의성에 직접적인 영향을 준다.
- 사용자가 정한 임의성이 떨어지는 난수: 프로토콜에서는 보안성을 위해 사용자가 임시비밀을 임의적으로 정할 것을 기대하지만, 사용자는 어떤 문자를 정하고 항상 그 문자만을 임시비밀로 사용할 수도 있다. 게임에 참여하는 사용자로부터 일정 수준 이상의 임의성을 얻어낼 수 있다는 연구결과[15]도 있지만 게임이 아닌 목적에서 사용자로부터 충분한 임의성을 가져오는 것은 쉽지 않다. 예로, [16]의 연구에서는 사용자가 사진에서 임의적인 공간을 비밀로 사용하길 바라지만 실제로 사용자들이 선택한 곳을 분석하여보면 특정한 패턴이 존재한다는 것을 알 수 있다.

위의 두 내용을 종합하여보면 사용자가 정하는 임시비밀로부터 일정 수준의 임의성을 보장받기가 쉽지 않다는 것을 알 수 있다. 이는 인증 기법의 보안성이 사용자가 정하는 임시비밀에만 의존해서는 안 된다는 것을 말해준다.

4.1.2 일방향성을 가지지 않는 연산의 활용

사용자에게 요구할 어떤 연산이 일방향성을 가지도록 하는 것 또한 쉽지 않은 문제이다. 예로, 시각적인 정보를 가지고 할 수 있는 가장 간단한 연산 중의 하나는, [3]에서와 같이 눈으로 어떤 개체의 위치를 확인하고 선택하도록 하거나 이 개체를 다른 개체(예로, 임시비밀)의 위치로 이동시키는 것이다. 하지만 그 과정에서 이동정보가 노출되고, 공격자는 얻어낸 이동정보를 바탕으로 거꾸로 추적할 수 있기 때문에 일방향성을 가지지 못하게 된다. 이때 보안성을 가지기 위한

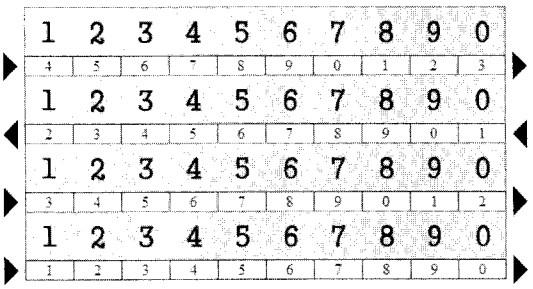
방법으로는 공격자가 이동정보를 통한 분석을 시도하더라도 비밀이 노출되는 것이 아니라 비밀의 후보군만이 노출되도록 설계하는 것이다. 4.2.1장에서 이동정보를 통한 분석이 가능한 경우와 그에 대한 대응방법에 대해 보였다.

4.1.3 고도의 집중력을 요구하지 않는 간단한 연산

사용자는 시스템에 안전하게 로그인하기 위한 것이라 하여도 고도의 집중력이 요구되는 연산을 원하지는 않을 것이다. 아무리 간단한 사칙연산이라 하여도 이러한 연산에 익숙하지 않은 사람들도 고려해야 하기 때문에 결국 비교적 부담이 적은 사람의 인지능력 또는 인식능력을 최대한 활용하여 연산을 설계하는 것이 중요하다.

4.2 비밀퍼즐을 통한 비밀증명방법

이 장에서는 4.1장의 분석을 바탕으로 비밀을 복합적으로 이용하는 비밀증명방법을 제안한다. 이 비밀증명방법은 모바일뱅킹과 같이 다양한 비밀을 사용하는 시스템을 목적으로 한다. 전자금융거래의 거래승인에 사용되는 이체비밀(계좌비밀번호 또는 이체비밀번호)과 승인수단(보안카드와 OTP)은 공통적으로 4자리 또는 6자리의 숫자로 이루어져있다. 이 중에서 승인수단은 재전송 공격에 대비하기 위해 매번 다른 응답을 입력할 수 있도록 하였고 그 응답은 서버가 평문을 알 수 있도록 설계되어 있다. 보안카드의 경우 2자리씩 두 번, 질의-응답형태로 사용되기 때문에 보안카드의 비밀 모두 또는 그 비밀들을 복원시킬 수 있는 인자를 서버가 저장하고 있을 것으로 예상할 수 있다. OTP의 경우에는 질의-응답 방식, 시간 동기화 방식, 이벤트(카운터) 동기화 방식, 조합(시간 및 카운터) 방식이 있으며[17], 이들 방식 모두 응답을 생성하기 위해 필요한 인자가 서버와 공유 또는 동기화 되어있기 때문에 서버가 응답을 평문상태로 알 수 있다. 특히, 시간으로 동기화된 OTP는 조회하는 도중에 그 응답이 바뀔 수 있기 때문에 사전에 정의한 t만큼의 오차를 허용한다. 서버가 평문을 알 수 있는 승인응답과는 다르게 이체비밀번호는 고정된 형태의 비밀로, 서버에 암호화 또는 일방향 해시함수의 결과 값이 저장되어 있을 수 있기 때문에 서버에서 평문을 알아내지 못할 수 있다. 이체비밀과 승인수단을 복합적으로 이용하면 아래와 같은 질의-응답 형태의 비밀퍼즐을 만들 수 있다.



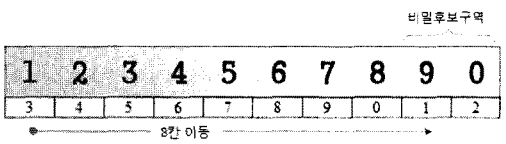
〔그림 8〕 비밀을 복합적으로 증명하기 위한 비밀퍼즐

비밀퍼즐이 사용자에게 주어졌을 때는 이체비밀번호 및 승인응답 모두가 오름차순으로 정렬되어 있으며, 비밀퍼즐에 응답하는 방법은 다음과 같다. 〔그림 8〕에서 i번째 자리의 이체비밀번호를 i번째의 회색바탕 행에서 찾고, 그 아래에 i번째 자리의 승인응답을 화살표가 가리키는 방향으로 이동하여 위치시킨다. 이체비밀번호가 '3869'이고 승인응답이 '6989'라면 첫 번째 승인응답은 오른쪽으로 7칸, 두 번째 승인응답은 왼쪽으로 1칸, 세 번째 승인응답은 오른쪽으로 8칸, 네 번째 승인응답은 오른쪽으로 0칸 이동하는 것이다. 이체비밀번호를 아는 사용자는 승인응답의 숫자를 쉽게 이동시킬 수 있는 반면에 공격자는 모든 경우의 수를 추측해야 한다.

이체비밀번호는 고정된 형태의 비밀이기 때문에 편의성을 위해서 일종의 색인역할을 하도록 고정시켰고 매번 다른 응답을 사용하도록 하는 승인응답은 이동이 가능한 형태로 만들어 놓았다. 서버에는 이동정보가 전달된다. 이동정보를 받은 서버는 먼저 승인응답을 이에 이동정보를 적용하여 해당 위치의 숫자를 알아낸다. 서버에서 이체비밀을 해시하여 저장하고 있는 경우, 앞에서 알아낸 숫자의 해시값을 데이터베이스에 저장된 이체비밀과 비교, 검증한다. 이체비밀과 승인응답 두 비밀 모두를 알지 못하면 퍼즐을 풀 수 없다.

4.2.1 이동방향의 제한

승인응답은 키보드의 방향키, 마우스 또는 터치스크린을 통해 이동될 수 있다. 이동할 때 승인응답 양쪽에 화살표가 가리키는 방향으로만 이동되도록 하였다. 만약 이동방향을 제안하지 않고 사용자에게 자유롭게 이동방향을 정할 수 있도록 한다면 사용자는 〔그림 9〕와 같이 직관적으로 이동방향을 정할 수 있게 되고 이때의 이동정보는 공격자가 비밀을 얻어내기 위한 단서가 될 수 있다.



〔그림 9〕 사용자가 승인응답을 직관적으로 이동했을 때 발생할 수 있는 취약점

예로 이체비밀번호가 '9'이고 승인응답이 '1'일 때, 왼쪽으로는 2번 움직이면 되지만 사용자는 오른쪽 방향으로 움직이는 것이 더욱 직관적이기 때문에 오른쪽으로 8번 움직일 수 있는 것이다. 이렇게 직관적인 이동방향의 선택은 공격자에게 비밀후보구역을 지정할 수 있게 한다. 공격자가 여러 세션을 반복하여 바라보면 여러 개의 비밀후보구역을 얻을 수 있게 되고 여기에 교차공격을 하면 이체비밀번호를 얻어낼 수 있는 것이다. 이러한 공격에 대비하여 이동방향을 시스템에서 정하고 그 방향으로만 승인응답을 이동할 수 있도록 하면, 직관적인 이동방향에서 발생하는 비밀후보구역은 발생하지 않기 때문에 교차공격에 내성을 가지게 된다.

4.2.2 비밀퍼즐의 적용범위

비밀퍼즐은 키보드 후킹과 피싱 공격을 동시에 대응할 수 있다. 사용자가 입력하는 이동정보만으로는 이체비밀이나 승인수단을 얻어낼 수 없고, 비밀퍼즐을 사용하기로 약속한 사용자는 피싱사이트에 유도되더라도 평문으로 비밀을 입력하지 않을 것이기 때문이다. 하지만 공격자의 관심이 비밀을 얻어내는 것이 아니라 위조된 문서에 승인을 유도하기위한 것이라면 비밀퍼즐은 대응방안이 되지는 않는다. 비밀퍼즐은 비밀의 소유를 증명하는 것이 목적이기 때문이다.

V. CAPTCHA를 응용한 문서변조 방지기법

이 장에서는 3장에서의 CAPTCHA에 대한 분석과 4장에서 소개한 퍼즐을 응용하여, CAPTCHA를 응용한 문서변조 방지기법을 제안한다.

5.1 문서변조 공격과 자동화

〔2〕에 따르면, 문서변조 공격이 이루어지기 위해서 공격자는 우선 소스코드의 분석이 가능해야 한다. 이를 바탕으로 제작된 악성 프로그램은 스스로 공격대상 문서를 탐지할 수 있어야 하고, 문서를 변조할 수 있어야 하며, 거래내용을 조작할 수 있어야 한다. 다시 말하면, 이 중 한 가지를 불가능하도록 하는 것은 악성 프로그램의 자동화된 공격을 차단하여 문서변조 공격을 방지할 수 있다는 것으로 해석할 수 있다. 이 논문에서 제안하는 문서변조 방지기법은 자동화를 방지하는 CAPTCHA를 응용한 방법이다. CAPTCHA를 응용하여 자동화를 방지하는 것은 다양한 곳에 적용될 수 있으며, 계좌이체에 대해 자동화된 문서변조를 방지하는 방법에 대한 예는 다음과 같다.

5.1.1 계좌이체 과정

사용자의 입장에서 계좌이체 과정은 다음과 같다.

1) 계좌이체 요청: 이체금액 및 수신계좌정보를 입력하고 계좌이체를 요청한다.
2) 수신계좌 확인: 수신자명 및 수신계좌정보를 확인한다. 여기서 1)의 과정이 반복될 수 있다.
3) 승인응답 입력 및 공인인증서 서명: 이체금액 및 수신계좌 확인 후 승인응답을 입력하고 공인인증서 패스워드를 입력하여 거래승인을 요청한다.
4) 이체결과 확인: 이체 결과를 확인한다.

CAPTCHA를 응용한 방법은 서버와 사용자 사이의 질의-응답 프로토콜이라 볼 수 있다. 사용자가 정상적인 정보를 확인하였는지를 서버가 확인하는 것으로, 서버가 질의에 해당하는 CAPTCHA를 전송하고 사용자가 이에 응답하는 형태가 된다. 위의 계좌이체 과정에서 거래가 실제로 승인되기 전까지, 서버가 질의를 보낼 수 있는 단계는 2단계 및 3단계이다.

5.1.2 CAPTCHA와 승인수단의 조합

계좌이체의 2단계 및 3단계에서 문서변조 공격은 사용자가 의도한 거래내용이 보이도록 조작하여 사용자에게 승인응답을 입력하고 공인인증서를 이용해 서명하도록 유도한다. 이에 제안하는 기법은 CAPTCHA와 승인수단을 조합하여 거래내용을 비밀(승인수단)을 가진 사용자에게만 확인받도록 하는 것이다. 거래내용을 바탕으로 CAPTCHA를 생성하고 이에 대한 응답에 승인수단을 이용하도록 하면 사용자에게 거래내용을 전달시켜 주는 것과 동시에 서버에서는 사용자가 확인한 거래내용과 승인응답을 확인할 수 있게

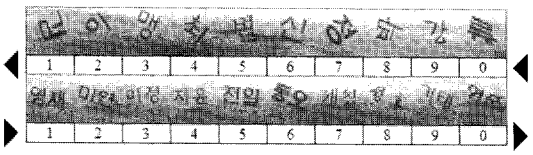
된다. 이는 4장에서 보인 비밀퍼즐과 같이 질의-응답 형태로 동작하게 되며, 그에 요구되는 임의성은 승인수단을 통해 충족되는 형태이다.

5.2 CAPTCHA가 적용된 서명퍼즐

서버는 사용자에게 확인받고자 하는 내용과 임의적으로 선택한 내용들을 CAPTCHA로 표현한다. [그림 8]의 퍼즐에서 이체비밀대신 거래내용을 CAPTCHA로 표시하고 그 아래 승인수단을 위치시키도록 한 것이다. 이렇게 승인용도로 변경한 퍼즐은 계좌이체의 2단계 또는 3단계에서 적용될 수 있다.

5.2.1 계좌이체를 요청할 때

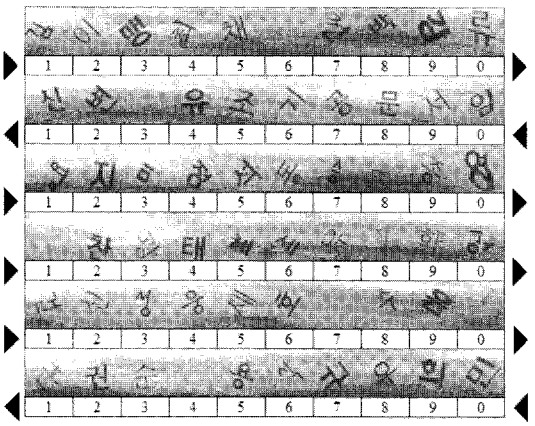
계좌이체의 첫 번째 단계는 계좌비밀번호, 금액, 수신계좌정보를 입력하고 이를 서버에 전송하는 것이다. 서버에서는 계좌비밀번호, 이체가능금액 여부, 수신계좌의 존재유무 등을 확인한 이후에 이체정보를 클라이언트에 전달한다. 이때 클라이언트에 전달되는 이체정보에는 수신자명이 포함되며 이는 화면에 출력되어 사용자에게 수신자명을 확인하도록 한다. 수신계좌가 조작된 경우 서버는 해당 계좌의 수신자명을 클라이언트에 전송하지만 악성 프로그램이 이 또한 조작하여 사용자가 의도했던 계좌의 수신자명을 출력하기 때문에 서버는 사용자가 수신자명을 제대로 확인하였는지를 알 수 없게 된다. 이 과정에 [그림 10]과 같은 승인퍼즐을 적용하면 악성 프로그램이 수신자명을 자동으로 조작하는 것을 방지할 수 있다. 예로, 사용자가 예상한 수신자명이 '김기태'이고 승인수단의 앞 두 자리가 '57'이라면 위의 숫자 행을 좌측으로 4칸, 그리고 아래 숫자 행을 우측으로 2칸 이동하였을 때에만 계좌이체 추가나 다음단계로 넘어갈 수 있도록 하는 것이다.



[그림 10] 편의성을 위해 간소화된 퍼즐

5.2.2 전자서명을 할 때

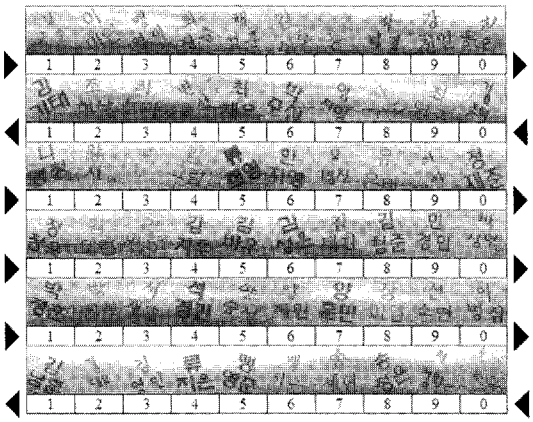
문서변조를 통한 치명적인 공격들은 전자서명과 관련되어있다. 현재의 전자서명방법이 가지는 취약점은



[그림 11] 더미가 포함된 퍼즐

서명할 내용의 조작유무에 관련 없이 서명 될 수 있도록 설계되어있다는 것이다. 서명할 내용이 악성 프로그램으로부터 조작되는 것을 막을 수 있다면 부인방지 보안서비스가 현재보다 한층 강화되고 서명이 악용되는 피해를 예방할 수 있을 것이다. [그림 11]과 [그림 12]와 같은 형태의 CAPTCHA는 거래를 최종적으로 승인하는 과정인 전자서명 과정에서 사용될 수 있다.

사용자가 서버에 거래내용을 전송하면 서버는 거래확인을 위한 내용(예제에서는 수신자명)을 이용하여 [그림 11]과 같이 서명퍼즐을 작성한다. 서명퍼즐의 보안성은 사용할 승인수단의 길이로 조절할 수 있다. 예로 [그림 11]에서, '김성재'는 첫 번째, 세 번째 그리고 네 번째 행에서 찾아볼 수 있으며 그 외의 행에는 빈칸에 승인수단을 위치하도록 하여 6자리의 승인수단에 대해 맞게 구성할 수 있다. 이렇게 빈칸을 추가하는 방법 외에, 한글을 초성, 중성, 종성을 분리하여



[그림 12] 6건을 동시에 이체하는 경우의 비밀퍼즐2

〔표 1〕 비밀증명기법의 비교분석

		S3PAS[5]	강전일 등의 기법[7]	김창순 등의 기법[8] (SRS의 경우)	비밀퍼즐
비밀	종류	alphanumeric, 특수문자	알파벳, 숫자	알파벳, 숫자	숫자
	길이	4자리 이상	≥ 6자리	≥ 8자리	≥ 4자리(6자리)
	술어	3자리 비밀	2자리 비밀	1자리 비밀	1자리 비밀
사용성	화면표시정보	10x10행렬	≥ 6x6행렬	≥ 9x4행렬	≥ 10x4행렬(6행렬)
	임시비밀	-	≥ 1	≥ 1	0 (승인수단사용)
보안성	라운드 수	≥ 4회	≥ 6회	≥ 8회	1회
	무작위 시도	$\leq 10^{-5}$	$10^{-8} \sim 10^{-5}$	$\leq 9^{7}$	$t/10^{6} \sim t/10^{4}$
	훔쳐보기 공격	-	$10^{-8} \sim 10^{-5}$	$10^{-7.2}$	$t/10^{6} \sim t/10^{4}$
	교차공격	3회[6]	≥ 12~49회	2회[8]	-

CAPTCHA를 생성하는 방법 등을 고려할 수 있다.

한 번에 다수의 거래를 서명하고자 하는 경우에는 전자서명과정에서 다음과 같은 서명퍼즐〔그림 12〕을 사용할 수 있다.

서명할 거래의 수가 승인수단의 길이(예로, OTP는 6자리)를 초과하는 경우에는, 하나의 서명퍼즐에 그 만큼의 거래가 포함되도록 하여 여러 개의 서명퍼즐을 이용할 수도 있다. 이러한 서명퍼즐은 자동화를 방지하는 것이 목적인만큼, 서명에 포함될 내용이 많은 경우에는 그 수를 간추려 승인수단이 가지는 보안성 수준으로 구성하여 사용할 수 있다. 현재 거래승인 과정에 받아들여지는 보안수준은 $10^{-4} \sim 10^{-6}$으로, 4행 또는 6행〔그림 12〕으로 구성된 서명퍼즐 하나로 문서변조 공격에 대응할 수 있다.

VI. 보안성 분석

이 장에서는 비밀퍼즐과 서명퍼즐이 가지는 보안성에 대해 분석하여 보인다.

6.1 비밀퍼즐의 보안성 분석

〔표 1〕은 비밀을 보호하는 기법들과의 비교를 보인다. [8]에서 SRS기법을 제외한 다른 기법들은 안전한 채널을 통해 임시비밀을 전달받도록 하여, 가정이 다르기 때문에 비교대상에서 제외하였다. 이론적으로 4자리의 이체비밀과 4자리의 승인수단을 함께 사용하는 경우 무작위 시도에 대해 이론상으로는 10^{4*2}만큼의 보안성을 제공한다. 하지만 비밀퍼즐을 사용하는 경우에는 이체비밀을 일종의 색인으로 사용하여 10^{4}만큼의 보안성만을 제공하게 된다. 더해서, 시간이 동기화된 OTP를 승인수단으로 사용하는 경우에는 사용자가 입력할 때의 시간차를 고려해야 하기 때문에 사전에 정의해놓은 허용범위 t만큼의 오차를 허용할 수 있다. 이때의 이론적인 보안성은 $t/10^{n}$가 된다(n은 승인응답의 길이이며, 보안카드를 사용할 때 $t=1$이다). 이는 계좌비밀번호나 이체비밀번호, 보안카드나 OTP에서 제공하고 있는 보안성으로, 제한된 횟수의 무차별 공격에 대응 가능한 보안수준으로 받아들여지고 있다.

비밀퍼즐은 비밀을 키보드에 그대로 입력하는 방법에 비하여 무작위 시도에 대한 보안성을 희생하지만 키로거, 피싱 공격 등에서도 비밀을 노출시키지 않는다는 이점을 가진다. 또한, 비밀퍼즐은 공격자가 사용자의 인증과정을 여러 번 바라보아도 비밀 후보군의 수가 줄어들지 않기 때문에 교차공격에 강한 내성을 가진다. 이는 4.2.1장에서 언급한바와 같이 사용자의 이동방향을 제한하여 얻게 된 특징이다. 이 기법은 숫자를 대상으로 하여 화면에 표시되는 정보의 복잡성이 비교적 낮고 인증을 위해 다수의 라운드를 요구하지 않아 다른 기법들에 비해 비교적 편의성이 높다고 할 수 있다.

6.2 서명퍼즐이 가지는 보안성

3.2장에서 소개한 기법들이 거래내용을 사람에게 확인 받도록 한 것에 비해 제안하는 기법은 거래내용을 승인수단을 가진 사람에게 확인 받도록 하였다. 3.2장에서 소개한 기법들을 이용하는 경우 승인수단을 독립적으로 사용해야 하며 악성 프로그램은 이 승인응답을 얻어내 다른 용도에 악용할 수 있다. 하지만 제안하는 기법의 경우 CAPTCHA를 읽을 수 있어야 승인응답을 얻어낼 수 있기 때문에 사용자가 입력한 승인응답의 악용을 사전에 방지하는 효과를 가진다.

2.1장에서 악성 프로그램은 CAPTCHA를 해결할 수 없다고 가정하였다. 하지만 실제로는 구현방법에 따라 왜곡률이 낮은 CAPTCHA는 글자를 인식하는 OCR(Optical Character Recognition)이나 신경망(Neural Network) 기술을 역으로 이용하면 쉽게 무력화 될 수 있음이 알려져 있다[19,20,21,22]. 예로, Arcot VPS나 MS워터마크와 같은 방법에서는 CAPTCHA에 표현된 문자를 읽고, 입력하도록 되어있다. 다시 말해, 사용자가 글자 모두를 읽을 수 있는 수준의 CAPTCHA가 요구되는 것이다. 만약 글자가 심하게 뒤틀려 사용자가 읽기 힘든 경우에는 CAPTCHA를 다시 제공하거나 보안수준이 떨어지는 것을 감안하고서라도 뒤틀림 정도를 낮춘 CAPTCHA를 제공해야 한다. 결국 제공되는 CAPTCHA의 보안수준은 사용자가 각각의 글자를 읽을 수 있는 수준으로 제한되는 것이다. 하지만 [11]에서 보인 CAPTCHA에서는 사용자가 같이 글자 하나를 완전히 읽지 못하였더라도 자신이 의도했던 거래내용을 선택하는 것에는 크게 지장을 주지 않기 때문에 뒤틀림이 비교적 심한-비교적 보안강도가 높은- CAPTCHA를 사용할 수 있게 된다. 예로, '홍?동'과 같이 한 글자를 읽지 못하였더라도 이를 인식하고 선택하는 것에는 무리가 없다. 더해서, CAPTCHA에 표현된 내용이 사전에 노출된 것이라 하더라도 이미지의 왜곡률이 높기 때문에 이를 해석하는 것이 쉽지는 않을 것으로 사료된다(적어도, 기존의 CAPTCHA 보다는 보안성이 높다). 서명퍼즐에서는 왜곡률이 높은 CAPTCHA를 사용하기 때문에 그 퍼즐을 구성하는 내용이 추측이 가능한 것이라 하더라도 주어진 퍼즐에서 조작할 위치를 찾는 것은 쉽지 않은 일이 될 것이다.

거래내용을 조작하는 공격은 사용자가 서명할 거래내용을 확인할 수 있는 장치 또는 비밀이 어느 용도로 사용되었는지를 확인할 장치가 마련되어있지 않아서 발생한다. 서명퍼즐은 $t/10^6 \sim t/10^4$의 확률로 퍼즐을 조작되는 것을 방지하는 것과 동시에 승인응답을 직접적으로 노출시키지도 않기 때문에 서명단계에서의 보안성을 한층 강화시켜줄 수 있다. $t/10^6 \sim t/10^4$의 확률은, 승인수단이 가진 보안성이다. 또한 3.3장에서 소개한 기법들이 문서변조 공격에 대응하기 위해 추가적인 장치를 요구하는 반면에, 제안하는 기법은 그러한 장치를 요구하지 않고 현재 사용되고 있는 승인수단을 활용한 것이기 때문에 비교적 호환성이 높다.

VII. 결 론

이 논문에서는 사용자의 비밀 및 서명을 보호하는 연구들에 대해 분석하였으며, 이를 바탕으로 모바일뱅킹에서 사용가능한 비밀퍼즐과 문서변조 공격에 대응하는 서명퍼즐을 소개하였다. 비밀퍼즐은 사용자의 응답에 충분한 임의성과 교차공격에 대한 내성을 가지며, 복잡도가 낮은 숫자만을 이용하고 다수의 라운드를 요구하지 않기 때문에 비교적 높은 편의성을 가진다. 서명퍼즐은 악성 프로그램의 자동화를 차단할 뿐만 아니라 사용자에게 거래내용을 직접적으로 확인시켜주고, 승인응답이 악용되는 것 또한 방지하기 때문에 문서변조와 같은 공격을 효과적으로 방지하고 전자서명의 보안성을 한층 향상시켜줄 수 있을 것이라 기대한다.

이 연구의 결과물은 스크립트 수준에서 구현될 수 있을 정도로 구현방법이 어렵지 않아 모바일뱅킹 외에 다른 형태의 전자금융거래에도 적용될 수 있다. 특히, 웹에 적용될 경우에는 플러그인을 추가적으로 요구하지 않으면서도 사용자의 비밀을 보호하고 문서변조 공격을 자동화할 수 없도록 하여 더욱 수준 높은 보안서비스를 제공하는데 도움이 될 수 있을 것이다.

참고문헌

[1] 한국은행, "2010년 3/4분기 국내 인터넷뱅킹서비스 이용현황," http://bok.or.kr/contents/total/ko/boardView.action?boardBean.categorycd=0&boardBean.sdt=&boardBean.edt=&boardBean.searchColumn=title&boardBean.searchValue=%C0%CE%C5%CD%B3%DD&menuNaviId=559&boardBean.menuid=559&boardBean.brdid=74592&boardBean.rnum=1&boardBean.cPage=1, 2010

[2] 맹영재, 신동오, 김성호, 양대헌, "전자금융거래에서의 문서변조 취약점 분석 및 대응방법 고찰," 한국정보보호학회 학회지, 20(6), pp. 17-27, 2010년 12월.

[3] 강문설, 김용일, "비밀번호 훔쳐보기로부터 안전한 기술을 내장시킨 비밀번호 입력기의 설계 및 구현," 한국정보처리학회 논문지, 17-D(2), pp. 167-174, 2010년 04월.

[4] S. Wiedenbeck, J. Waters, L. Sobrado,

and J.Birget, "Design and evaluation of a shouldersurfing resistant graphical password scheme," Proceedings of the Advanced Visual Interfaces, pp. 177-184, May 2006.

[5] H. Zhao and X. Li, "S3PAS: a scalable shoulder-surfing resistant textual-graphical password authentication scheme," Proceedings of the 21st IEEE International Conference on Advanced Information Networking and Applications Workshops, vol. 2, pp. 467-472, May 2007.

[6] 신동오, 강전일, 맹영재, 양대헌, "S3PAS의 교차 공격에 대한 취약성 분석," 한국정보보호학회 동계 학술대회, 19(2), pp. 409-412, 2009년 12월.

[7] 강전일, 맹영재, 양대헌, 이경희, 전인경, "행렬 상에서 문자 간 연산을 수행하는 패스워드 인증 기법," 한국정보보호학회 논문지, 19(5), pp. 175-188, 2009년 10월.

[8] 김창순, 윤선범, 이문규, "모바일 환경에서 엿보기 공격에 강한 패스워드 입력방법," 한국정보보호학회 논문지, 20(3), pp. 93-104, 2010년 06월.

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

[10] 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.

[11] 김성호, 강전일, 김기태, 양대헌, "문맥 기반의 캡차를 이용한 신뢰성 있는 인터넷 계좌 이체 방법," 한국정보보호학회 동계학술대회, 19(2), pp. 319-323, 2009년 12월.

[12] A. Hiltgen, T. Kramp and T. Weigold, "Secure internet banking authentication," IEEE Security and Privacy, Mar./Apr. 2006

[13] 임형진, 이정근, 김문성, "안전한 인터넷 뱅킹을 위한 트랜잭션 서명기법에 관한 연구," 인터넷정보학회논문지, 9(6), 73-79, 2008

[14] T. Weigold, T. Kramp, R. Hermann, F. Höring, P. Buhler and M. Baentsc, "The zurich trusted information channel - an efficient defence against man-in-the-middle and malicious software attacks," TRUST 2008, LNCS 4968, pp. 75 - 91, 2008.

[15] R. Halprin and M. Naor, "Games for Extracting Randomness," SOUPS 2009.

[16] S. Wiedenbeck, J. Waters, J.C. Birget, A. Brodskiy and N. D. Memon, "Authentication using graphical passwords: effects of tolerance and image choice," SOUPS 2005.

[17] 서승현, 강우진, "OTP 기술현황 및 국내 금융권 OTP 도입사례," 한국정보보호학회 논문지, 17(3), pp. 18-25, 2007년 06월.

[18] G. A. Miller, "The magical number seven, plus or minus two: some limits on our capacity for processing information," The Psychological Review, vol. 63, pp. 81-97, 1956

[19] G. Mori and J. Malik, "Recognizing objects in adversarial clutter: breaking a visual CAPTCHA," In Proceedings of the Computer Vision and Pattern Recognition (CVPR) Conference, IEEE Computer Society, Vol. 1, pp. 134-141, 2003.

[20] K. Chellapilla and P. Y. Simard, "Using machine learning to break visual human interaction proofs (HIPs)," Advances in Neural Information Processing Systems 17 (NIPS'2004), MIT Press, 2004.

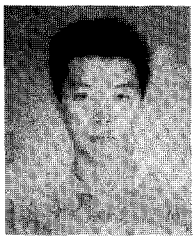
[21] K. Cellapilla, K. Larson, P. Y. Simard, and M. Czerwinski, "Computers beat human at single character recognition in reading based human interaction proofs (HIPs)," In Proceedings of International Conference on Email and Anti-Spam (CEAS), 2005.

[22] PWNtcha Project, http://sam.zoy.org/pwntcha/

[23] 맹영재, 신동오, 김성호, 변제성, 양대헌, "키보드 보안 요구사항 재고," 한국정보보호학회 동계학술대회, 20(2), pp. 86-89, 2010년 12월.

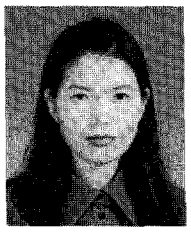
[24] McAfee Threats Report: Fourth Quarter 2010, http://www.mcafee.com/us/resources/reports/rp-quarterly-threat-q4-2010.pdf

〈著者紹介〉



맹 영 재 (YoungJae Maeng) 학생회원
2006년 8월: 인하대학교 컴퓨터공학부 졸업
2006년 8월: 인하대학교 정보통신대학원 석사
2008년 9월~현재: 인하대학교 컴퓨터정보공학과 박사과정
〈관심분야〉 인터넷 보안, 네트워크 보안

양 대 헌 (DaeHun Nyang) 종신회원
1994년 2월: 한국과학기술원 과학기술 대학 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



이 경 희 (KyungHee Lee) 정회원
1989년: 서울대학교 식품영양학과 학사
1993년: 연세대학교 전산과학과 학사
1998년: 연세대학교 컴퓨터과학과 석사
2004년: 연세대학교 컴퓨터과학과 박사
1993년 1월~1996년 5월: LG소프트(주) 연구원
2000년 12월~2005년 2월: 한국전자통신연구원 선임연구원
2005년 3월~현재: 수원대학교 조교수
〈관심분야〉 영상처리, 컴퓨터비젼, 인공지능, 패턴인식, 생체인식, 얼굴인식, 다중생체인식