情報保護學會論文誌
第18卷 第4號, 2008. 8

복수의 이미지를 합성하여 사용하는 이미지 기반의 캡차와 이를 위한 안전한 운용 방법*

강 전 일[1†], 맹 영 재[1], 김 군 순[1], 양 대 헌[1‡], 이 경 희[2]
[1]인하대학교 정보통신대학원, [2]수원대학교 전기공학과 조교수

Image-based CAPTCHA Using Multi-Image Composition and Its Secure Operation*

Jeonil Kang[1†], YoungJae Maeng[1], KoonSoon Kim[1], DaeHun Nyang[1‡], KyungHee Lee[2]
[1]Graduate School of IT&T, INHA University, [2]The University of Suwon

요 약

현재 인터넷의 발달과 봇의 사용이 활발해짐에 따라 컴퓨터와 사람을 분류할 수 있는 수단인 캡차(CAPTCHA)가 많이 활용되고 있다. 글자를 이미지 형태로 출력한 후 이를 변형시키는 방법이 많이 사용되는 캡차는 많은 연구 활동에 의하여, 인공지능 기법을 사용하면 쉽게 무력화 될 수 있음이 알려져 있다. 이에 대한 대안으로 이미지를 활용하는 캡차가 주목받고 있고 그에 따라 여러 형태의 이미지 기반 캡차가 제안 및 구현되었다. 그러나 이미지를 활용하는 캡차 또한 각각의 여러 다른 문제가 있는 것도 사실이다. 이 논문에서는 이러한 문제점에 대해서 짚어보고 이를 해결하기 위한 방안으로 복수의 이미지를 합성하는 캡차를 제안하였다. 또한 안전한 캡차의 운용을 위하여 가상 세션을 사용하지 않는 통신 프로토콜을 제안하였으며, 이에 따른 세부 사항에 대해서 논의하였다.

ABSTRACT

According to the growth of the internet and the usage of software agents, the CAPTCHA that is a method for taking apart humans and computers has been widely deployed and used. As the results of many research activities, the CAPTCHA, which is spoken for a distorted image material including random text, has known to be easily breakable via artificial intelligence techniques. As one of alternatives for those text-based CAPTCHAs, methods using photos are concerned and various image-based CAPTCHAs are suggested. However, image-based CAPTCHAs still have some problems. In this paper, we discuss what are the problems in each image-based CAPTCHA and propose a new image-based CAPTCHA using image composition as the solution of those problems. Furthermore, for the secure operation of the CAPTCHA, we suggest a communication protocol that works without the virtual session and consider possible security and usability problems in the protocol.

Keywords : CAPTCHA, Turing Test, Image Composition, Human Identification

Ⅰ. 서 론

캡차(CAPTCHA, Completely Automated Public Turing test to tell Computers and Humans Apart 또는 HIP, Human Interactive Proof)는 컴퓨터와 사람을 구별할 수 있게 해주는 공개된 튜링 테스트를 의미한다.[1,5] 사람은 쉽게 풀 수 있지만 현재의 컴퓨터가 풀기 힘든 문제라면 무엇이든 캡차에 이용될 수 있기 때문에 주로 인공지능(artificial intelligence)에 관련된 문제들이 사용된다. 캡차는 현재 사용자의 서비스 가입, 게시판의 게시물 등록, 특정 파일이나 이메일 주소로의 접근, 스팸 메일 차단, 투표 봇의 차단 등에 널리 사용되고 있으며, 인터넷의 발달과 봇의 사용이 증가함에 따라 사용 빈도가 점점 늘어나고 있는 추세이다.

2005년 W3C의 Matt May는 기술 자료(Technical Report)[2]를 통하여 이러한 캡차의 여러 문제점을 지적하고 다른 대안들을 사용하는 것이 좋겠다는 의견을 내었다. 그러나 그 대안이라는 것이 온톨로지(Ontology) 기술로 쉽게 무력화될 수 있는 논리적인 문제라든가, 개인의 프라이버시에 해당하는 생체 정보나 신용 카드, 공인 인증서 등의 정보를 이용하라는 것이어서 그러한 대안들에 대해서는 부정적일 수밖에 없다. 현재 캡차가 많이 사용되는 분야는 어느 정도의 익명성이 보장되어야하는 측면이 있기에 위와 같은 개인의 민감한 정보의 요구는 현실에 맞지 않는다. 그러한 이유에서 현재까지 캡차를 대체할 만한 마땅한 수단이 없으며, 캡차의 수요는 더 늘어날 것이다.

대부분의 캡차는 무작위한 글자를 이미지 형태로 출력하고 변형하는 방식을 취하고 있는데, 이는 여러 개인적인 프로젝트나 학문적인 노력에 의하여 높은 확률로 무력화될 수 있음이 알려져 있다. 그에 대한 해결책으로 주목할 만한 것은 이미지를 활용한 캡차이다. 이미지를 활용한 캡차는 초기에 연구되었고, 사용되지 않고 있다가 최근에서야 다시 몇몇 사람에 의하여 연구되고 있다. 이러한 이미지 기반의 캡차로는 카네기멜론 대학 팀의 PIX CAPTCHA[1,5]와 Oli Warner의 KittenAuth[12], Chew와 Tygar의 이미지 기반 캡차[6], Microsoft Research 팀의 Asirra[11]가 있다. 하지만 이미지를 사용하였다고 하더라도 해결해야할 문제는 여전히 남아 있다. 위와 같은 이미지 기반 캡차는 충분한 보안성을 위해서 많은 이미지가 사용되어야 하는데, 이는 시스템을 구축하는데 있어 많은 부담이 된다. 따라서 상대적으로 적은 수의 비용에 더 높은 보안성을 갖는 캡차의 필요성은 여전히 존재한다.

또한 캡차가 가지고 있는 자체의 문제뿐만 아니라 실제 구현하고 운용하는 데에서 발생하는 문제도 중요하다. 아무리 캡차가 쉽게 깨지지 않는다고 하더라도, 보안성을 고려하지 않은 통신은 캡차를 쉽게 우회할 수 있는 문제점이 있다.[13] 특별히 HTTP와 같이 비상태 유지의 통신 환경에서 파일(서버)과 쿠키(클라이언트)를 사용하는 가상 세션에 의존한 통신 방식은 캡차를 우회하지 못하게 코드를 꼼꼼히 작성한다고 하더라도, 외부적인 보안 취약점에 노출될 가능성이 있기 때문에 충분히 안전하다고 말하기 어렵다. 따라서 가상 세션을 사용하지 않고 캡차를 안전하게 테스트할 수 있는 방법이 필요하다 할 수 있다.

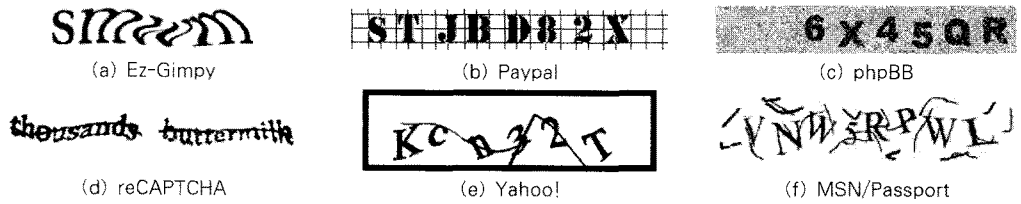
이 논문에서는 기존의 글자 및 이미지 기반의 캡차가 갖는 문제점을 살펴보고, 이러한 문제점들을 해결하거나 보완하기 위하여 새로운 방식의 이미지 기반 캡차를 소개하고 안전한 운용을 위한 통신 프로토콜을 제안한다. 2장에서는 기존의 캡차와 관련된 여러 연구나 활동에 대해서 살펴보고, 이 논문에서 제안하는 캡차와 크게 관련이 있는 이미지 기반 캡차의 문제점과 이를 공격하기 위한 방법들을 살펴본다. 3장에서는 2장에서 살펴보았던 문제점을 보완한 새로운 이미지 기반 캡차를 설계하고 이를 실제로 네트워크를 통해 운용하는 방법에 대해서 알아본다. 4장에서는 이 논문에서 제안한 캡차를 실제로 구현한 데모를 사용하여 피실험자들에게 실험한 결과를 간략히 소개한다. 5장에서는 이 논문에서 제안한 이미지 기반 캡차에 관련된 여러 주제에 대해서 알아보고, 6장에서 이 논문을 마무리한다.

접수일 : 2008년 1월 9일; 수정일 : 2008년 5월 11일;
채택일 : 2008년 6월 23일
* 본 연구는 정보통신부 및 정보통신연구진흥원의 IT신성장 동력핵심기술개발사업의 일환으로 수행하였음. [2008-F-036-01, 익명성기반의 U-지식 정보보호기술개발]
† 주저자, dreamx@seclab.inha.ac.kr
‡ 교신저자, nyang@inha.ac.kr

Ⅱ. 기존의 캡차와 관련한 연구 및 활동

2.1. 캡차의 종류

[그림 1]과 같은 텍스트 기반의 캡차는 무작위로 문자열을 생성하고 이를 왜곡된 형태의 이미지로 만들어



(a) Ez-Gimpy (b) Paypal (c) phpBB

(d) reCAPTCHA (e) Yahoo! (f) MSN/Passport

〔그림 1〕 여러 가지 텍스트 기반의 캡차

(a) animal-pix (b) naming CAPTCHA (c) KittenAuth

〔그림 2〕 여러 가지 이미지 기반의 캡차

사용자에게 보여주어 이미지에 어떠한 문자가 적혀 있는지 물어보는 형식을 가지고 있다. 이러한 형태의 캡차는 언뜻 컴퓨터가 풀기 어려울 것처럼 보이나, 글자를 인식하는 OCR(optical character recognition)이나 신경망(neural network) 기술을 역으로 이용하여 캡차의 많은 구현들이 쉽게 무력화될 수 있음이 알려져 있다.[4,7,9,13] 이러한 문제를 해결하기 위하여 텍스트 기반의 캡차는 보다 복잡한 형태를 갖도록 발전하였다.[8,10] 그러나 컴퓨터가 풀 수 없도록 잘 설계된 떫떫 캡차의 경우 사람조차 잘 인식하지 못하는 경우가 발생하게 되는 문제점이 발생하기도 한다.

[그림 2]와 같이 이미지를 사용하는 캡차의 경우 무작위로 선택하고 왜곡한 이미지가 무엇인지 물어보거나, 동일한 특징을 갖는 복수의 이미지를 선택하게 한다거나, 복수의 이미지에서 동일한 개체의 이름을 물어보는 등의 방식을 사용하고 있다.[1,5] 이러한 방식은 현재까지 텍스트 기반의 캡차와 같이 그 자체가 쉽게 무력화되지 않지만, 올바른 태그가 달리고 사용에 적합한 많은 수의 이미지를 확보해야하는 어려움이 있다. 또한 웹페이지에서의 구현 시 캡차가 차지하는 크기가 텍스트 기반의 캡차에 비해서 훨씬 많은 공간을 차지한다는 단점이 있다.

그 외에 소리를 이용하거나 논리적인 문제를 이용하는 캡차가 있다. 소리를 이용하는 캡차는 범용적인 사용을 위해서는 다양한 언어를 위한 데이터베이스를 구축해야하는 하는 문제가 있다. 논리적인 문제를 이용하는 캡차의 경우 복잡한 논리의 문제를 만들 수 없기 때문에 텍스트 기반의 캡차가 갖는 문제점을 가지고 있을 뿐만 아니라, 온톨로지 기법에 취약한 면이 있다.

글자는 인간이 글자를 인식하기 위한 최소한의 범위가 존재하여 글자에 작은 변화를 줄 수 있을 뿐이다. 반면, 어떠한 사물에 대한 이미지는 동일한 사물에 대해서도 다양한 외곽선이나 색상, 배경 등을 가지고 있지만 인간이 그 사물을 인식하는데 별다른 어려움이 없다. 이미지를 사용하기 때문에 사물의 이름은 제한된 분량의 사전에 존재하게 되어 응답의 다양성은 떨어지게 되지만, 컴퓨터에게 해당하는 사물의 이름을 맞추라는 것은 매우 어려운 일이다. 따라서 이미지 기반의 캡차에 인공지능의 기술을 사용하여 이를 무력화 하기란 쉽지 않다. 그렇다고 하더라도 이미지를 사용하여 캡차를 구현하는 것은 여전히 여러 문제가 남아 있다.

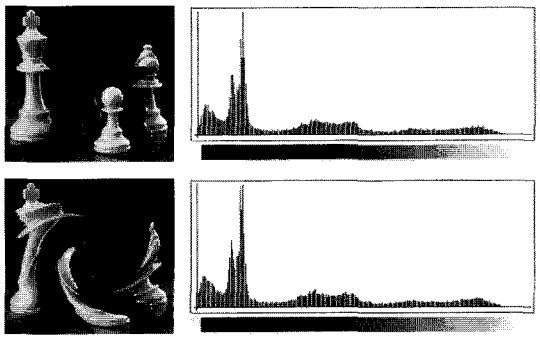
2.2 이미지 기반 캡차와 문제점

떫떫 캡차의 경우, 공격자가 생성자가 가진 이미지를 가지고 있지 못해도 반복적인 문제의 요구를 통하여 생

성자로부터 이미지를 모을 수 있다. 이때, 공격자는 생성자가 가지고 있는 이미지의 메타 정보를 얻어올 수 없지만 생성자가 캡차를 만드는 비용보다 적은 비용으로 메타 정보를 구축할 수 있다. 예를 들자면, naming CAPTCHA의 경우 하나의 문제에서 동일한 개체로 어떠한 이미지들이 분류되었는지 알 수 있다. 이는 하나의 캡차 문제에 노출된 이미지 수에 반비례하는 노력으로 공격자가 생성자와 동일한 이미지 데이터베이스를 구축할 수 있음을 알 수 있다. 이와 비슷하게, 동일한 분류에 속하지 않는 이미지를 선택하도록 하는 anomaly CAPTCHA는 메타 정보를 모른다고 하더라도 반복적인 문제 요구만으로, 문제 안에 노출된 중복 이미지들을 힌트로 하여 생성자의 전체 이미지 데이터베이스의 분류 체계를 알 수 있게 된다. 이러한 문제가 발생하는 근본적인 이유는 캡차의 보안성이 이미지 데이터베이스의 크기가 아니라 이미지의 분류에 있기 때문이다.

12장의 이미지 중에서 모든 고양이 그림을 선택하게 하는 Asirra는 '이미지의 분류'와 더불어 '경우의 수'를 이용한다.[11] Asirra는 잃어버린 애완동물을 찾아주는 Petfinder.com의 도움을 받아 사람들이 올린 자신의 애완동물 사진 300만장 이상의 이미지를 사용할 수 있어 안전하다고 한다. 그러나 단순하게 계산하더라도 1초에 12장의 이미지(한 번의 테스트)를 가져온다고 하면 300만장은 30일이면 소스 이미지를 모두 가져올 수 있다. 추가적인 서버에서의 보안 기법을 사용하지 않을 때, 무작위로 컴퓨터가 응답한 대답에 Asirra가 사람이라고 판단 할 확률을 2^{-12} =1/4096에 해당한다. 이 수치는 이미지의 수에 비하면 매우 작은 수치이므로 일정 수준이상의 이미지의 수는 최소한의 요구 조건임을 알 수 있다. 또한 개와 고양이를 컴퓨터가 분류할 수 없을 것이라고 기대하지만 개와 고양이를 특징지을 수 있는 특징(예를 들면 눈동자의 생김새, 귀의 모양)이 분명히 존재하므로, 컴퓨터가 이를 활용할 경우 컴퓨터가 캡차를 무사히 통과할 확률은 더 높아지게 되는 문제점이 있다.

떫떫 캡차는 이미지 데이터베이스의 크기를 이용하여 보안성을 확보하려고 한다. 예를 들자면, animal-pix와 같은 캡차는 하나의 동물 이미지에 변형을 주어 사용자에게 무슨 동물의 이미지인지 묻는다. 공격자는 통신을 엿듣고 이미지를 모아 과거에 사용되었던 이미지인지 확인함으로써 캡차를 무력화시킬 수 있다. 이러한 경우 생성자는 공격자에게 문제의 이미지가 과거에 사용되었던 이미지가 아니라고 믿게 하거나, 또는 과거에



〔그림 3〕 원본 이미지와 변형 이미지의 히스토그램 분석

사용했던 이미지를 다시 사용하지 않음으로써 높은 보안성을 가진 캡차를 만들 수 있다. 그러나 전자의 경우, [그림 3]에서 볼 수 있듯이 이미지의 단순 변형만으로는 공격자를 속일 수 없다. 이미지의 단순한 변형은 이미지만의 고유한 특성이 남아 있는 경우가 많다. 이미지를 회전을 시키거나 부분을 왜곡 변형을 한다고 하더라도 히스토그램 정보는 거의 그대로 남아 있으며, 노이즈를 섞는다 하더라도 사람이 인식할 수 있는 한 외곽선은 여전히 추출 가능하다. 따라서 해당 캡차가 매우 빈번히 사용되는 경우에는 충분한 보안성을 보장하기 위해서 구축해야하는 이미지 개수가 많아야 할 필요가 있다. 그러한 이유에서 떫떫 기존의 연구에서는 필요한 다수의 이미지를 구글(Google)과 같은 다른 사이트에서 구축해놓은 천문학적 크기의 이미지 데이터베이스의 사용을 제안하였다.[6] 그러나 메타 정보가 정확한지, 저작권에 위배는 되지 않는지, 이미지가 실제 획득 가능한지 모른다는 점에서 여전히 문제의 소지가 있다고 할 수 있다. 실제로 Luis von Ahn이 제안했던 EPS 게임에서 메타 정보의 정확도는 83% 정도로 알려져 있다.[3]

2.3 이미지 기반 캡차에 대한 공격 방법

2.3.1 사람의 협력과 인공 지능 기법을 응용한 공격 방법

이미지의 변형을 이용하거나 이미지 분류를 이용하는 떫떫 이미지 기반의 캡차에 대해서, 컴퓨터는 사람의 도움을 얻어 비교적 쉽게 이미지 기반 캡차를 무력화할 수 있다. 만약 충분한 수의 이미지가 사용되지 않는다면, 컴퓨터는 테스트에 사용되는 떫떫 이미지를 모으고 이 이미지들에 메타 정보를 달아줄 것을 협력자(사람)에게 요구할 수 있을 것이다. 컴퓨터는 이렇게 메타 정

보를 얻은 이미지들이 테스트에 다시 사용될 때까지 여러 번의 재시도 끝에 캡차 테스트를 통과할 수 있을 것이다. 이 때, 이미지의 변형은 이러한 경우 컴퓨터에게 자신이 가지고 있는 이미지가 사용되었는지 확인하는 과정을 다소 어렵게 할 수 있지만 불가능하게 하기는 힘들다. 이미지 x를 받아 변형된 이미지 x'을 생성하는 함수를 D라고 하면, 공격자는 변형된 이미지 x'과 y'이 같은 지 확인하는 함수 A를 만들 수 있다면 이러한 캡차 시스템을 무력화시킬 수 있다.

$$A(x',y') = A(D(x),D(y)) = \begin{cases} true & \text{if } x = y \\ false & \text{if } x \neq y \end{cases} \quad (1)$$

위의 식에서 함수 A은 역함수 D^{-1}를 구할 수 있다면 쉽게 만들 수 있다. 그러나 이미지의 변형이 복합적이고 무작위로 일어나게 한다면 역함수 D^{-1}를 구하는 것은 쉽지 않다. 따라서 함수 A은 입력 값들로부터 변하지 않는 공통된 요소를 추출해서 비교할 수 있는 추상화(abstraction) 함수이어야만 한다. 이러한 추상화 함수는 인공 지능 영역에서 쉽게 찾아볼 수 있다. 알려져 있기로 이러한 문제를 해결하기 위하여 PCA(Principal Component Analysis)나 LDA(Linear Discriminal Analysis)와 같은 선형 판별 분석법이나 인간의 뇌세포를 모델로 한 신경망(Neural Network), SOFM(Self Organized Feature Map), HMM(Hidden Markov Model), SVM(Support Vector Machine) 등이 사용될 수 있다.

2.3.2 사회 공학적 공격

캡차에서의 '사회 공학적 공격(social engineering attack)'은 실제 사람이 동원되어 이루어지는 공격을 의미한다. 값싼 다수의 인력을 고용하여 캡차를 풀게 할 수 있다. 무력화하고 싶은 캡차를 자신이 생성한 캡차로 위장하고 다른 종류의 대가를 지불하는 방식으로 사람들에게 풀도록 하는 방법이다.[2] 이러한 사회 공학적인 공격은 캡차를 처음 고안해낸 연구팀이 있는 카네기멜론 대학교의 한 학생에 의해 처음 제시된 이래, 아직까지 이를 막기란 불가능하다고 여겨지고 있다. 캡차가 생성된 시간과 사용자가 이에 대한 응답을 주는 시간을 짧게 강제하는 방법은 공격자가 원하는 시간에 캡차를 제공하는 시스템의 자원에 접근하기 힘들게 하지만, 특정한 시간대에 집착하지 않는 공격자에 대해서는 여전히 해결책이 되지 못한다.

2.3.3 안전하지 않은 네트워크 환경을 이용한 캡차의 공격

또 다른 가능한 문제점으로는 캡차가 많이 사용하는 네트워크 환경에 있다. 캡차는 현재 HTTP를 사용한 월드 와이드 웹(World Wide Web, WWW) 서비스 위에서 많이 사용되고 있다. HTTP는 비상태유지형(stateless) 네트워크 프로토콜로써, 이 위에서 캡차를 구현하기 위해서 가상적인 연결 상태를 만들어주는 세션(session) 메커니즘이 널리 사용되고 있다. 가상 세션은 고유한 아이디를 가지고 있으며, 서버 쪽에는 세션 아이디와 세션 정보가 파일의 형태로 저장되고 클라이언트 쪽에서는 세션 아이디가 쿠키(cookie)나 URL 첨부 형태로 저장된다. 각종 PHP나 ASP, JSP 같은 많은 웹 스크립트 언어가 이러한 가상 세션을 지원해준다.

문제는 많은 경우의 웹 프로그램이 가상 세션을 위하여 보안 설정을 준비하여 사용하는 경우는 거의 없다는 것이다. 또한 자체적으로 서버를 운영할 능력이 없는 중소 업체나 개인, 또는 단체는 보안성을 충분히 고려하지 않은 시스템의 초기 설정들과 프로그래밍 방법론을 사용하고 있다.

한 홈페이지 관리자는 대부분의 캡차 구현에서 테스트에 성공한 세션을 만료시키지 않는 문제를 이용하여 캡차 테스트를 회피하는 방법에 대해서 기술하였다.[14] 그에 따르면 현재의 캡차 테스트에 과거에 사용되었고 해답을 알고 있는 테스트 이미지에 해당하는 세션 아이디를 재전송함으로써, 현재의 캡차 테스트를 통과할 수 있다고 한다. 이것은 사람이 한 번 캡차 테스트를 통과한 뒤에 컴퓨터가 사람이 입력한 정보와 세션 아이디를 서버 쪽에 지속적으로 재전송함으로써 이 후의 많은 캡차 테스트를 통과할 수 있는 공격 방법이 있을 수 있음을 시사한다.

위와 같은 예에서는, 프로그래머가 세션을 만료시키는 코드를 집어넣음으로써 보안 문제를 완벽히 해결할 수 있을 것처럼 보인다. 그렇지만 문제는 여전히 남아 있다. 많은 시스템에서는 세션을 만든 뒤 사용자가 입력을 줄 때까지 세션에 다시 접근하지 않는다. 이는 공격자가 시스템의 자원을 잠식시킬 수 있는 공격이 가능하다는 의미와 같다. 가상 세션은 실제 연결을 사용하지 않기 때문에 파일이나 메모리를 사용하여 세션 정보를

저장한다. 세션이 수집(garbage collection) 당하지 않는 시간(PHP의 초기 설정은 1440초) 동안 공격자는 수많은 세션을 만들 수 있다. 이렇게 많은 세션 파일이 서버에 저장되면 쓰레기 수집에 걸리는 시간이 점점 지체되게 되고 시스템에 대한 서비스 거부 공격을 수행할 수 있다. 또한 세션 정보는 설정에 따라 세션을 만든 사용자에게만 보여주는 것이 아니고 쿠키나 URL을 위조함으로써 세션을 가로챌 수도 있기 때문에, 웹 페이지의 다른 보안 취약점을 이용하여 이를 공격할 수 있는 등의 보안 문제가 발생할 가능성을 배제할 수 없다. 다시 말해, 일반적인 웹 환경에서 캡차만 배타적으로 세션을 사용할 수 없는데, 캡차가 세션을 사용하는 것이 과연 안전한가에 대한 문제인 것이다.

Ⅲ. 제안 기법 : 복수의 이미지 합성을 이용한 캡차

3.1 위협 모델 및 시스템 요구 사항

공격자는 컴퓨터를 이용하여 대규모로 어떤 시스템에서 제공하는 리소스에 접근하거나 서비스를 요청하고 싶어 한다. 해당 시스템은 캡차를 통하여 캡차 테스트를 통과한 경우에만 이를 허용하도록 설계되었다. 공격자는 인공지능 기법을 사용할 수 있으며, 네트워크에서 다른 사람의 패킷을 도청하거나 자신의 패킷을 전송할 수 있다. 그러나 다른 사람이 전송중인 패킷을 가로채 이를 자신의 패킷처럼 보이게 만드는 시도는 수행하지 않는다. 왜냐하면 캡차의 응답에 해당하는 패킷은 일반적인 상황에서 빈번히 발생하는 패킷이 아니어서 대규모로 캡차를 풀기 위한 방법과는 거리가 멀고, 게다가 이를 위해서는 수많은 패킷의 관찰이 필요할 뿐만 아니라 패킷을 가로채는 것이 실제로는 쉽지 않기 때문이다.

시스템은 이 논문에서 제안하는 이미지를 사용한 캡차와 운용 방법을 사용하며, 소스 이미지는 자체적으로 제작하거나 배타적 사용권을 양도받아 사용하여 외부에 노출되지 않도록 한다. 소스 이미지의 수는 보안 수준에 따라 달라지지만 일정 수준 이상을 유지한다.

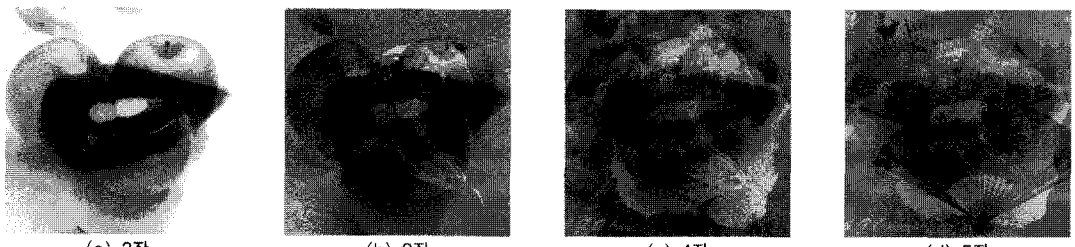
〔표 1〕 논문에서 사용하는 기호 및 설명

기호	설명
$A \rightarrow B:m$	A는 B에게 m을 보낸다.
$h(m)$	m에 대한 해시를 계산한다.
$E_K(m)$	비밀키 K로 m을 비밀키 암호화 방식으로 암호화한다.
$M(a,b)$	이미지 a와 b를 합성한다.
$D^a(b)$	이미지 b을 임의의 방법으로 a번 변형한다.
$HTML_{req}$	html 문서의 요청 형식
IMG_{req}	img 파일의 요청 형식
$A=B$	B를 A에 할당한다.
$A+=B$	A는 B를 포함한다.
U	사용자 (User)
C	클라이언트 (Client)
S	서버 (Server)

[표 1]은 이 논문에서 사용하는 기호와 그에 따른 설명이다.

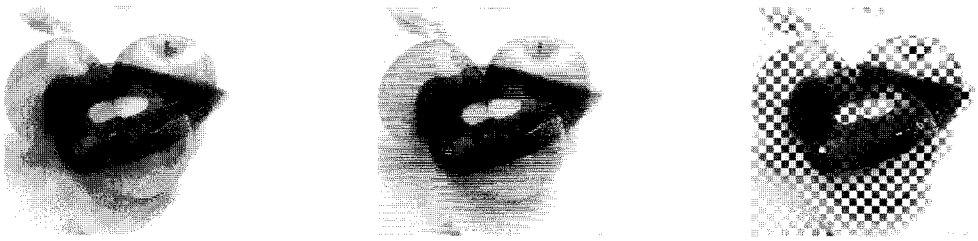
3.2 테스트 이미지의 합성

이 논문에서는 기존의 이미지 기반 캡차 시스템의 다양한 문제를 해결하기 위하여, 복수의 이미지를 합성하여 사용할 것을 제안한다. 이미지 합성에는 여러 방법이 사용될 수 있지만 그 중에서 가장 대표적인 방법은 이미지의 투명도를 조절하여 오버레이 하는 방법일 것이다. 복수의 이미지를 오버레이 하였을 때에도 사람은 여전히 하나 이상의 개체에 대하여 인식을 수행할 수 있다. [그림 4]는 차례대로 입술, 사과, 뱀, 무궁화 꽃, 부채를 섞은 결과로 위와 같은 주장을 실험적으로 증명한다. 그러나 너무 많은 이미지를 오버레이하면 한 지점



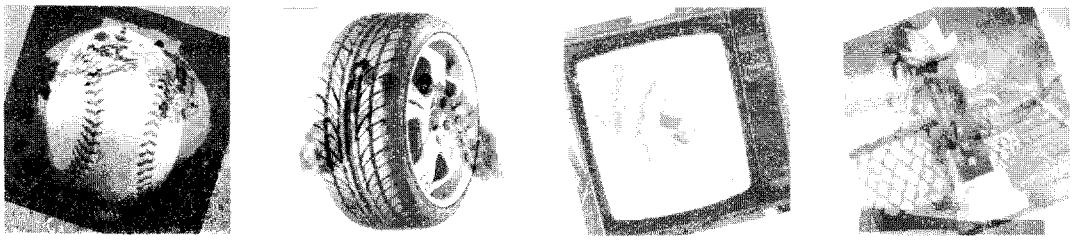
(a) 2장 (b) 3장 (c) 4장 (d) 5장

〔그림 4〕 복수의 이미지 오버레이와 그에 따른 인식



(a) 오버레이 (b) 인터라인 (c) 정사각형

[그림 5] 여러 가지 이미지 합성 방법, 인터라인과 정사각형과 같은 합성 방법을 사용해서는 안 된다.



(a) 도자기+야구공 (b) 개미+타이어 (c) TV+오른손 (d) 비석+공룡

[그림 6] 최종 테스트 이미지

(픽셀)에서 한 이미지가 최종적인 이미지를 만드는데 기여하는 정도가 지나치게 낮아져 형태가 보이지 않게 된다.

이외에도 상호 치환 같은 방법이 있을 수 있으나, [그림 5]의 (b)와 (c) 같은 단순한 상호 치환의 경우 두 이미지를 분리하는 데에는 별다른 어려움이 없다는 점에 유의해야한다. 복수의 이미지가 각자의 이미지로 분리될 수 있다면 그러한 방법은 기존의 하나의 이미지를 사용하는 이미지 기반의 캡차와 크게 다르지 않다.

기존의 이미지 캡차에서 이미 사용되는 이미지 변형은 여러 가지 변형을 동시에 사용함으로써 좋은 효과를 거둘 수 있다. 단일의 이미지 변형에 대해서는 몇몇의 경우를 제외하고는 거의 정확하게 역변형을 구할 수 있고 변형한 상태에서도 동일함을 보일 수 있는 방법이 존재하지만, 다중의 이미지 변형은 그렇지 못하다. 다중의 변형 과정에서 중첩된 정보의 손실과 역변형 과정에서 순서에 따라 올바른 역변형을 구하기 힘들 수도 있고, 변형한 상태에서 동일함을 보이던 몇몇 방법의 유효성을 떨어뜨릴 수 있다. 하지만 지나친 이미지의 변형은 사람 또한 인식하기 어렵게 하는 문제점이 있기 때문에 적당한 수준에서 변형의 정도가 정해져야 한다. 대표적인 이미지 변형은 노이즈 첨가, 굴곡, 회전, 확대(축소), 흐리기, 대비 변화, 밝기 변화, 색상화(colourize) 등이 될 수 있다.

최종적으로 이 논문에서 제안하는 캡차의 테스트 이미지는 [그림 6]과 같은 형태를 갖게 된다.

3.3 복수의 캡차 테스트

일반적으로 테스트를 위하여 이미지를 사용자에게 제공하였을 때, 시스템이 사용자에게 기대할 수 있는 것은 이미지 안에 사람이 인식할 수 있는 개체들의 이름이다. animal-pix와 같은 경우 기대할 수 있는 개체의 이름은 단지 하나밖에 없고, 그것도 대중적으로 알려진 이름이거나 개체군을 의미하는 단어일 것이다. 예를 들자면, '장미'를 보고서는 많은 사람이 '장미'라고 답할 수 있지만 같은 꽃 종류인 '용담(龍膽, Gentiana scabra)'을 보고서는 대부분의 사람들은 높은 수준으로 추상화된 단어인 '꽃'이라고 할 것이다. 사전에는 매우 많은 개체를 지칭하는 단어가 있지만, 개개인이 가진 지식과 관심의 차이에 따라 알고 있는 정도가 다르기 때문에 실제 사람들에게서 기대할 수 있는 단어의 수는 매우 한정적이다. 이는 컴퓨터가 이미지를 인식하지 못한다고 하더라도 단일 테스트를 통과할 수 있는 확률이 일반적인 기대 이상이라는 것을 의미한다.

초기의 KittenAuth 같은 경우 개체의 이름을 물어보진 않지만 이와 비슷한 문제가 있었다. 9개의 그림 중에서 3개의 그림을 선택하는 문제로는 컴퓨터의 재시도

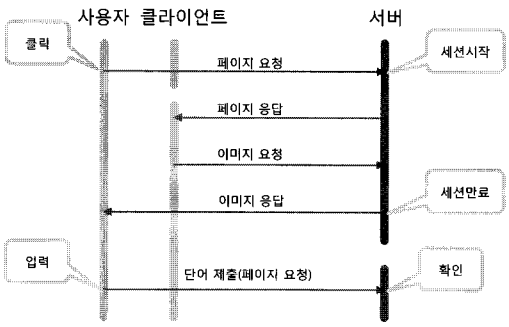
공격에 대해서 전혀 안전하지 못하다. ${}_9C_3=84$이기 때문에 컴퓨터가 한 가지 경우를 선택하고 이 답이 맞을 때까지 단지 84번의 재시도면 충분하다. 굳이 컴퓨터가 고양이 그림을 인식하려고 하지 않아도 되는 것이다. 이를 해결하기 위한 단순한 방법은 그림의 수를 늘리고 선택하는 경우를 고정하지 않는 것이라고 한다. 이는 Asirra에서 사용하였던 방법이다.

이와 같이 문제를 해결하기 위한 다른 접근 방법으로는 다수의 캡차 테스트를 수행하거나 응답하는 단어가 갖는 구체성 정도에 따라서 테스트의 수를 조절하는 것이다. 다수의 캡차 테스트를 수행하였을 때, 컴퓨터가 사람의 도움 없이 이미지를 인식하지 않고 이미지를 사용하는 캡차 시스템을 공격할 수 있는 확률은 극히 낮아지게 된다. 물론, 이미지나 글자, 단어의 세분화와 추상화 등의 복잡한 문제와 상관없이 복수의 캡차 테스트는 시스템이 가질 수 있는 보안 수준을 높게 조절할 수 있는 좋은 방법임은 틀림이 없다.

반면, 이러한 복수의 캡차 테스트는 사용자의 입장에 있어서 번거로울 수 있다. 사용자의 편의성의 측면에서, 합성된 이미지를 사용한 캡차 시스템의 경우 동시에 사용자에게 두 개 이상의 단어를 입력받을 수 있다. 이는 곧, 테스트의 횟수를 줄일 수 있는 한 가지 방법이 될 수 있음을 의미한다. 하지만 어느 정도로 테스트의 횟수를 줄일 수 있는지는 사용자가 사용하는 단어의 세분화와 추상화 수준에 따라 결정된다.

3.4 가상 세션을 사용하지 않는 캡차 운용

가상 세션은 보안성이 필요한 프로그램에서는 사용하지 않는 것이 좋은데, 가상 세션을 사용하지 않고 캡차 테스트를 비연결형 네트워크에서 안전하게 수행하기 위해서는 시스템의 비밀키가 필요하다. 서버는 사용자의 요청에 따라 수행할 캡차에 관한 정보를 자신만이 아는 비밀키를 이용하여 암호화하고 이를 사용자에게 돌려준 뒤, 후에 사용자가 입력하는 정보와 함께 사용자가 입력한 정보가 올바른지 확인하는 과정을 거친다. [그림 7]은 웹에서 일반적으로 사용되는 캡차 테스트의 순차적인 순서를 보여준다.



[그림 7] 웹에서의 순차적인 캡차 테스트 순서

위의 그림에 따라 가상 세션을 사용하지 않는 두 장의 이미지 합성을 이용한 캡차의 운용은 다음과 같이 될 수 있다. 여기서는 두 장의 이미지를 합성하여 사용하는 방법에 대해서 설명한다. 두 장 이상의 이미지 합성은 아래의 과정에서 쉽게 변형될 수 있다.

단계 1. (페이지 요청) 사용자가 페이지의 특정 링크를 클릭함으로써 서버로 페이지가 요청된다.

$$U(C)\rightarrow S\colon HTML_{req}$$

단계 2. (페이지 응답) 페이지 요청을 받은 서버는 무작위로 두 장의 이미지 x와 y를 선택하고 각각의 이미지에 해당하는 메타 정보의 집합 $I_x=\{\epsilon_{x,1},\epsilon_{x,2},\cdots,\epsilon_{x,m}\}$와 $I_y=\{\epsilon_{y,1},\epsilon_{y,2},\cdots,\epsilon_{y,m}\}$, 테스트 시작 시간 t, 임의의 솔트 s, 그리고 자신만이 알고 있는 비밀키 K를 이용하여 다음과 같은 메시지를 포함하는 페이지를 만들어 클라이언트로 전송한다.

$$\begin{aligned}&S\rightarrow C\colon MSG=t,s,h(\epsilon_{x,1},s),\cdots,h(\epsilon_{x,m},s),\\&h(\epsilon_{y,1},s),\cdots,h(\epsilon_{y,m},s)\\&SIG=E_K(h(MSG))\\&IMG_{req}\mathrel{+}=E_K(x,y,t)\end{aligned}$$

단계 3. (이미지 요청) 서버로부터 메시지를 받은 클라이언트는 IMG_{req}에 해당하는 정보를 서버로 재전송함으로써 서버에게 이미지를 요청한다.

$$C\rightarrow S\colon IMG_{req}$$

단계 4. (이미지 응답) 클라이언트로부터 이미지 요청을 받은 서버는 비밀키 K를 이용하여 메시지를 풀고, 테스트 시작 시간 t를 확인하여 유효한 값이 적혀 있고 각각의 이미지 x와 y가 존재한다면 이 둘을 합성하여 클라이언트에게 돌려준다. 그렇지 않다면 이미지 생성을 중지한다. 아래 식에서 D^a와 D^b는 이미지 변형을 각각 a번, b번 수행함을 의미한다. a와 b는 1보다

크고 일정 수준보다 낮은 영역에서 무작위로 선택된다.

$$S \rightarrow C: IMG_{file} = M(D^a(x), D^b(y))$$

단계 5. (단어 제출) 사용자는 이미지 IMG_{file}을 보고 사용자는 이미지에 해당하는 단어 ϵ_u를 입력하고 이를 MSG, SIG와 함께 서버로 전송한다.

$$U \rightarrow C: \epsilon_u$$
$$C \rightarrow S: \epsilon_u, MSG, SIG$$

단계 6. (단어 확인) 서버는 MSG와 SIG가 유효한지 확인한다. 우선 테스트 시작 시간 t가 유효한지, SIG가 사용된 테스트 리스트(used test list)에 존재하는 지, $E_K(h(MSG))$와 SIG가 일치하는지 인지 확인한다. 유효하다면 사용자가 입력한 단어 ϵ_u와 s를 이용하여 $h(\epsilon_u, s)$가 MSG 안에 존재하는 지 확인한다. 확인 결과 존재한다면 사람이고 그렇지 않다면 컴퓨터이다. 캡차 테스트 결과 사람인 경우 서버는 t와 SIG를 사용 리스트에 저장하고 관리한다.

단계 3에서 웹에서는 클라이언트가 IMG값을 이미지 태그(tag)의 URL을 요청하는 방식으로 이루어진다. 단계 6에서 서버는 통과한 SIG 값을 사용 리스트를 통해 관리하는데, t가 유효한 시간 동안만 유지하면 된다. 위와 같은 통신이 성공적으로 완료된 뒤에는 서버는 바로 사용자가 필요로 하는 리소스나 서비스에 1회에 한하여 연결해주어야 한다. 만약 캡차를 통과했다는 사실을 증명할 수 있도록(즉, 리소스나 서비스에 복수의 접근이 가능하도록) 인증서와 같은 추가적인 정보를 클라이언트에게 건네줄 경우, 공격자는 다시 이를 엿듣고서 이를 사용할 수 있는 가능성이 존재하기 때문이다.

위의 통신 프로토콜은 솔트 s를 평문으로 전송함으로써, 클라이언트 측에서 사용자의 입력 단어가 올바른지 확인할 수 있도록 되어 있다. 만약 솔트를 사용하지 않고 단순한 메타 정보의 해시 값만을 건네줄 경우, 공격자는 이 해시 값을 이용하여 미리 만들어두었던 단어의 해시 값들을 이용하여 매우 쉽게 단어를 추측할 수 있다. 클라이언트 측에서 사용자의 입력이 올바른지 확인할 수 있다며 페이지의 변환 없이 사용자의 입력이 틀렸음을 알려 내용을 다시 적는 불편을 방지할 수 있다. 이와 관련된 보안성 분석은 5장에서 논의한다.

Ⅳ. 제안 기법의 구현 및 사용자 테스트

이 장에서는 앞서 제안한 기법 중에서 테스트 이미지를 생성하고 이를 실제 사용자에게 실험한 내용을 소개한다. 실험의 목적은 실제 사용자가 테스트 이미지에 대해서 어떻게 응답하는지에 확인하고 이를 텍스트 기반의 캡차와 비교하고자하는 것으로, 통신 프로토콜의 구현은 하지 않았다. 피실험자는 22명의 22세에서 27세까지의 대한민국 국적의 남·여로 이루어져 있다. 피실험자들은 원래의 이미지를 모르는 상태에서 실험에 참여하였으며 300×300 픽셀 크기의 5개의 테스트 이미지에 대해서 인식한 사물의 이름을 적도록 하였다. 모든 소스 이미지의 수는 200개이며, 각각의 이미지에 대한 메타 정보는 프로그램 개발자에 의해서 정해졌다. 또한 비교 대상으로 글자의 변화를 주지 않은 텍스트 기반 캡차가 수행되었다.

[표 2]에서 알 수 있듯, 제안 기법은 텍스트 기반의 캡차에 비해서 테스트 성공률이 낮았다. 하지만 평균 입력 타수를 고려한다고 하더라도 응답 속도가 더 빠르다는 점을 비추어보았을 때, 피실험자들이 제안 기법의 테스트 이미지를 올바르게 인식하는 것에는 별다른 어려움이 없는 것으로 보인다. 실제로 피실험자의 응답을 살펴보면 테스트 실패 요인 중에는 피실험자의 오타 이외에도 메타 정보가 미비하게 준비된 측면이 있음을 알 수 있었다.

실험에서는 변형되지 않은 글자를 사용한 텍스트 기반 캡차가 사용되었지만, 변형된 글자가 사용되었을 때는 더 낮은 테스트 성공률을 보일 것으로 기대할 수 있으므로 이 논문에서 제안하는 이미지 기반 캡차가 상대적으로 텍스트 기반의 캡차에 비해서 우수함을 알 수 있다. 변형된 글자를 사용한 텍스트 기반의 캡차와 관련된 사용자 평가는 Cellapilla 등 작성한 논문에서 찾아볼 수 있다.[8,9]

〔표 2〕 사용자 실험 결과 및 비교

	텍스트 기반 캡차	제안 기법
테스트 성공률	89.52%	86.36%
평균 입력 시간 (성공 시)	7.236초	3.161초
평균 입력 시간 (실패 시)	9.429초	4.397초
평균 입력 타수	8	5.76

이 실험 결과에서는 인문 과학에 관련된 사용자들이 입력한 단어의 세분화와 추상화 정도의 분석은 이루어지지 않았으며, 이를 앞으로의 연구 과제로 남겨둔다.

V. 제안 기법의 평가 및 토론

5.1 사회 공학적인 공격과 이미지 기반 캡차

모든 캡차는 사회 공학적인 공격에 약하다. 이에 대한 피해를 줄이기 위해서는 테스트의 유효 시간을 줄여 되도록 빨리 입력하게 하는 방법이 적당하지만 완벽한 해결책은 될 수 없다. 한 가지 가능성 있는 방법으로는, 이미지를 이용하여, 사람에게 사물에의 인식력뿐만이 아니라 '지식'을 이용하도록 하는 것이다. 만약 특정한 모집단을 대상으로 하는 캡차의 경우 그 모집단이면 누구나 알 수 있는 문제나 사물을 활용 할 수 있다. 예를 들자면, 백범 김구 선생님의 얼굴은 한국 사람이라면 누구나 알 수 있지만 보통의 외국인이라면 알기 어렵다. 이러한 지식을 활용하는 캡차는 글자로 만들기 어려울 뿐만 아니라 내용 또한 간단해질 수밖에 없지만 이미지로는 매우 다양하고 쉽게 구축할 수 있는 장점이 있다. 만약 텍스트로 이러한 캡차를 구현한다고 하였을 때, 그 캡차의 문제가 몰랐던 지식을 묻는 문제라고 할지라도 사람들은 제시된 문제의 몇몇 키워드를 이용하여 웹에서 쉽게 검색하여 해답을 얻을 수 있을 것이다. 하지만 캡차의 문제가 이미지일 경우 해당 이미지를 가지고 일반인들이 역으로 무언가를 검색하기란 현재까지 쉬운 일은 아니다.

이처럼 이미지와 사람의 지식을 이용하는 이미지 기반 캡차는 사회 공학적 공격에 어느 정도의 이점을 가지고 있지만, 완벽히 이를 막기란 쉬운 일이 아니다. 그것은 사람들이 그룹 안에서 공유하지만 그룹 외에는 배타적인 지식이 그렇게 많다고 볼 수 없기 때문이다. 또한, 사회 공학적 공격은 해당 그룹 안에서도 일어날 수 있고, 캡차가 현재 사용되는 대부분의 경우는 일반인들을 위한 것이라는 것을 감안하면 위와 같은 방법은 매우 제한적일 수밖에 없다.

5.2 협력자(사람)가 있는 공격자(컴퓨터)

제안하는 캡차를 성공적으로 공격하기 위하여 공격자는 두 장의 테스트 이미지 $\Gamma_1 = M(D^a(x_1), D^b(y_1), \cdots)$로부터 $D^a(x_1), D^b(y_1), \cdots$, 를 추출하는 함수 M^{-1}를 만들려고 할지 모른다.

$$M^{-1}(\Gamma_1) = M^{-1}(M(D^a(x_1), D^b(y_1), \cdots)) = D^a(x_1), D^b(y_1), \cdots \quad (2)$$

공격자가 만약 M^{-1}을 만들 수 있다면 제안하는 캡차는 함수 A을 다수 수행하는 것으로 어떠한 이미지들을 포함하고 있는지 확인할 수 있다. 만약 공통된 이미지를 확인할 수 있고, 기존에 메타 정보가 알려진 변형된 이미지 $D^c(x_1), D^d(y_1), \cdots$를 가지고 있다면 새로운 캡차 테스트 이미지들이 어떤 이미지를 포함하고 있는지 알 수 있다. 그러나 100×100픽셀의 그레이스케일 이미지를 가정한다고 하더라도 약 $128^{10,000}$ 이상의 경우의 수로 이미지를 나눌 수 있으므로 M^{-1}과 같은 단순한 분리는 불가능하다. 현재까지 합성된 이미지를 분류하는 방법은 존재하지 않는다. 이는 공격자가 메타 정보가 알려진 변형된 이미지를 가지고 있기 힘들다는 것을 의미하기도 한다.

여전히 공격자는 충분한 수의 테스트 이미지를 모아 이미지들에 메타 정보를 입력하고, 이를 학습시켜 새로운 테스트 이미지가 어떻게 분류되는지 확인할 수 있다. 하지만 이는 협력자(사람)가 반드시 필요한 공격방법이다. 공격자가 모든 이미지를 가지고 있지 않다고 하더라도 공격자(컴퓨터)는 테스트 이미지를 모아서 이를 협력자에게 풀도록 요구할 수 있다. 제안하는 이미지 기반의 캡차는 하나의 테스트 이미지에 복수의 메타 정보가 달릴 수 있으므로, 소스 이미지의 수만큼의 테스트 이미지가 있으면 하나의 메타 정보에 다른 두 장의 테스트 이미지를 연관 시킬 수 있다. 이를 인공 지능 기법을 이용하여 학습시켜 공격에 사용할 수도 있다. 이러한 공격이 어느 정도의 성공 가능성이 있는지는 실제로 실험을 해보아야 하겠지만, 이 또한 절대적인 수치는 아니다. 이에 대한 내용은 앞으로 수행할 연구에 남겨두고자 한다.

이러한 공격 가능성이 존재할 때, 소스 이미지의 수는 보안성에 크게 영향을 미치게 된다. 실제 시스템에서 소스 이미지가 수천, 수만 장을 사용된다고 가정하면 이에 대한 공격은 결코 쉽지 않다. 왜냐하면 협력자가 들여야 하는 노력은 소스 이미지의 수에 비례하여 증가하게 될 뿐만 아니라, 현재의 인공 지능 기법에서는 클래스 수가 많고 클래스 별 학습 데이터가 적은 경우 높은 인식률을 기대하기란 힘들다.

5.3. 모든 이미지와 메타 정보를 가지고 있는 공격자(컴퓨터)

이미지 기반의 캡차에 있어서 가장 강력한 공격자(컴퓨터)는 시스템이 가지고 있는 모든 이미지와 그에 해당하는 메타 정보를 가지고 있는 공격자이다. 이러한 경우, 기존의 이미지 변형을 사용하는 캡차의 경우 $D^a(x)$(테스트 이미지)와 y(학습 이미지)를 비교하는 문제로, 이 논문에서 제시하는 이미지 합성을 사용하는 캡차의 경우 $M(D^a(x_1), D^b(y_1), \cdots)$(테스트 이미지)와 x_2(학습 이미지)를 비교하는 문제로 단순화된다.

전자의 문제는 인공지능의 영역에서 얼굴 인식 등의 문제와 비슷하며, 앞서 밝힌 바와 같이 이를 해결하기 위한 많은 기법들이 존재한다. 후자의 문제는 분명 전자의 문제에 비해서는 더 어려워 보인다. 그러나 멀티레이블 분류법을 사용해야할 필요가 없으므로, 전자에서 수행하는 동일한 인공지능 기법을 동원하여 테스트 이미지를 분류해볼 수 있다. 테스트 이미지는 전자보다 더 복잡해져 있으므로 더 낮은 인식률을 기대할 수 있다 하더라도, 이를 증명할 수 있는 방법은 현재로써는 실험적인 것 이외에 마땅히 찾을 수 없는 실정이다.

5.4 네트워크 프로토콜의 안전성

제안하는 기법에서 사용하는 네트워크 프로토콜은 앞서 밝혔던 것과 같이 오프라인 사전 탐색 공격의 가능성이 있다. 메타 정보를 솔트와 함께 전송해주기 때문에, 공격자는 사전을 뒤져 올바른 응답을 찾아낼 수 있다. 이러한 위협을 감수하면서도 이렇게 하는 이유는 클라이언트 측에서 즉시 사용자가 입력한 단어가 올바른지 판단하기 위해서이다. 캡차의 답변과 함께 많은 사용자 입력이 들어간 페이지를 서버에 넘겨주었을 때, 캡차의 답변이 잘못되었을 경우 사용자의 입력이 사라질 위험이 존재한다. 이러한 경우, 사전에 캡차의 답변을 포함한 사용자의 입력을 클라이언트 측에서 확인해줄 수 있다면 이러한 불편함을 줄일 수 있다.

오프라인 사전 탐색 공격을 막기 위하여 입력 시간을 되도록 짧게 유지한다 하더라도 만약 사용자가 입력하는 단어가 한정적이라면(즉, 사전의 단어 수가 적다면) 공격자는 그 짧은 시간에서도 충분히 오프라인 사전 탐색을 수행할 수 있다. 앞의 실험에서처럼 200개의 이미지를 사용한 캡차 시스템의 경우 공격자가 사용하는 사전은 단지 200개 정도의 단어로 유지될 것이고(공격자가 네트워크를 도청하는 등의 방법을 사용하여 사전을 만들 수 있다), 솔트가 있다고 하더라도 200번의 해시 계산에 소용되는 시간은 사용자가 단어를 입력하는 시간보다 짧다. 따라서 위와 같이 사용자 편의성을 중시한 프로토콜을 사용하면서, 보안성을 확보하기 위해서는 1) 많고 다양한 이미지를 사용하고, 2) 사용자의 입력 시간을 짧게 하며, 3) 사용자의 응답을 도청하지 못하도록 SSL(Secure Socket Layer) 등의 보안 통신 프로토콜을 사용하고, 4) 복수의 테스트를 수행하는 등의 조치가 필요하다.

위와 반대로 클라이언트 측에서 사용자의 입력을 검증할 필요가 없다면 솔트 s를 MSG에서 빼고 SIG 안에 $SIG' = E_K(h(MSG), s)$처럼 넣음으로써 서버만이 사용자의 입력이 올바른지 아닌지 확인하도록 할 수 있다. 솔트 s를 모르는 공격자는 오프라인 사전 탐색 공격이 불가능하다. 그러나 이 경우 클라이언트가 사용자가 올바른 해답을 입력하였는지 알 수 없기 때문에 사용자의 입력을 서버로 넘겨야 한다. 이 과정에서 서버에게 특별하게 추가되는 연산은 존재하지 않지만, 잘못된 응답에 대해서도 같은 연산을 반복해야 하기 때문에 서버의 부담이 조금 증가할 것으로 예상할 수 있다.

한 편, 공격자는 위의 프로토콜을 이용하여 서비스 거부 공격을 수행할 수도 있을 것이다. 잘못된 응답이라 하더라도 이를 지속적으로 서버에 전송함으로써, 서버에게 계속적인 연산을 강요하는 것이다. 위 프로토콜을 사용하였을 때, 그렇지 않은 경우보다 서버가 처리해야 하는 연산이 더 많은 것은 틀림이 없다. 그러나 SSL과 같은 보안 통신 프로토콜을 사용한 환경에서 필요한 기본적인 연산에 비한다면 크게 문제되진 않을 것으로 보인다.

5.5 이미지의 메타 정보와 관련된 보안성

일반적인 사진 이미지를 캡차에 사용하였을 때, 이미지 안에서 사람이 인식할 수 있는 개체는 여러 가지가 있다. 또한 동일한 개체에 대해서도 다른 여러 표현이 존재할 수 있으며, 추상화 된 단어로 대표될 수도 있다. 이미지의 메타 정보는 사람이 입력해줘야 하는데, 이때에 다른 사람이 응답할 수 있는 다른 응답에 대해서도 최대한 많이 입력해놓을 필요성이 있다. 이 과정에서 고려해야 되는 문제는 잘못된 맞춤법, 띄어쓰기, 단어의 수 등이다.

사용자가 잘못 입력할 수 있는 단어에 대해서 모두 입력해놓으면 사용자의 편의성이 높아진다. 문제는 이러한 모든 경우를 고려해주기 위해서는 이미지에 메타 정보를 만들 때 이를 모두 입력해주어야 하지만 이는 결코 쉽지만은 않은 일이다. 띄어쓰기 또한 사람들이 틀리기 쉬운 부분이지만 비교의 수월성을 위해서 공백문자를 무시하는 것으로 해결 가능하다.

또 다른 문제가 되는 것은 하나의 이미지를 기술하기 위한 메타 정보의 수에 있다. 어떤 이미지는 메타 정보의 수가 많고, 어떤 이미지는 메타 정보의 수가 적게 되는데, 공격자는 네트워크에서 운용하는 캡차 시스템에서 *MSG*에 기술된 메타 정보의 수를 보고 이미지에 대한 힌트를 얻을 수 있다. 이 경우 서버는 무작위 비트열을 필요한 만큼 메타 정보에 삽입함으로써, 공격자에게 합성된 두 이미지가 갖는 단어의 수를 알 수 없도록 해야 한다.

Ⅵ. 결론 및 앞으로의 연구 계획

이 논문에서는 복수의 이미지를 합성하여 사용하는 이미지 기반의 캡차를 구현하는 이를 안전하게 운용할 수 있는 방법에 대해서 제안하였다. 이미지 기반의 캡차는 텍스트 기반의 캡차에 비해서 인식 문제의 어려움을 장점으로 하지만, 기존의 방법은 어느 정도의 보안성을 가질지, 실제로 구현 가능한지 등에 대한 충분한 고찰 없이 제시되어 왔다. 기존의 방법과는 다르게 두 이미지를 변형하고 이를 서로 합성함으로써 더 높은 보안성과 실제로 구현 가능한 타당한 수준으로 이미지의 수를 줄일 수 있을 것으로 기대할 수 있었다.

그러나 캡차 테스트 이미지 이외에 다른 여러 곳에서 다양한 문제점이 잠재되어 있을 수 있으며, 이를 보완하기 위하여 여러 보완 기법들과 함께 사용되어야 바람직할 것으로 보인다. 전체적인 캡차 시스템의 보안성을 높이기 위해서는 한 번의 캡차 테스트가 아닌 다수의 캡차 테스트를 수행해야했다. 이는 인공지능을 이용한 공격이나, 메타 정보의 취약점을 이용한 공격 등에 대처할 수 있을 것으로 보인다. 네트워크와 관련하여, 가상 세션은 여러 보안 문제를 낳을 수 있으므로 사용하지 않는 것이 바람직하며, 사용자의 편의를 위해서는 사용자 입력을 짧게 강제하거나 도청하지 못하도록 SSL 같은 추가적인 보안 설정이 필요할 것으로 보인다. 그렇지 않다면 사용자의 편의를 희생하고 캡차 테스트를 수행해야 안전할 것으로 보인다.

비록 이 논문에서는 인공지능과 관련되거나 사회공학적인 부분에 있어서 자세히 논의할 수 없었지만, 실제로 이미지 합성을 이용한 캡차 시스템을 구현하여 사용하기 위해서는 이러한 부분에 있어서 추가적인 실험이나 조사가 뒷받침 될 필요성이 있다. 앞으로, 인공 지능 기법들을 사용하여 앞서 논문에서 밝혔던 것과 유사한 형태로 제안 기법을 공격해보고 과연 충분한 안전성이 있는지 확인해볼 계획이다. 또한, 사용자들이 응답으로 주로 사용하는 단어의 추상화 정도에 따라, 동적인 횟수의 캡차 테스트가 가질 수 있는 보안성에 대한 분석을 수행할 예정이다.

참고문헌

[1] Luis von Ahn, Manuel Blum, Nicholas J. Hopper, and John Langford, "CAPTCHA: Using Hard AI Problems For Security", In Proceedings of Eurocrypt, pp. 294-311, 2003.

[2] Matt May, "Inaccessibility of CAPTCHA: Alternatives to Visual Turing Tests on the Web", W3C Working Group Note 23, http://www.w3.org/TR/turingtest/November 2005

[3] Luis von Ahn, "Human Computation," CMU-CS-05-193, 2005.

[4] Greg Mori and Jitendra 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.

[5] Luis von Ahn, Manuel Blum, and John Langford, "Telling Computers and Human Apart (Automatically) or How Lazy Cryptographers do AI", Communications of the ACM, Vol. 47, No. 2, pp. 56-60, 2004.

[6] Monica Chew and J. D. Tygar, "Image Recognition CAPTCHAs", In Proceedings of the 7th International Information Security Conference (ISC 2004), pp. 268-279, 2004.

[7] Kumar Chellapilla and Patrice Y. Simard,

"Using Machine Learning to Break Visual Human Interaction Proofs (HIPs)", Advances in Neural Information Processing Systems 17 (NIPS'2004), MIT Press, 2004.

[8] Kumar Cellapilla, Kevin Larson, Patrice Y. Simard, and Mary Czerwinski, "Designing Human Friendly Human Interaction Proofs (HIPs)", In Proceedings of Conference on Human Factors in Computing systems (CHI), 2005.

[9] Kumar Cellapilla, Kevin Larson, Patrice Y. Simard, and Mary 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.

[10] Kumar Cellapilla, Kevin Larson, Patrice Y. Simard, and Mary Czerwinski, "Building Segmentation Based Human-friendly Human Interaction Proofs (HIPs)", In Proceedings of International Workshop on Human Interaction Proofs (HIP), 2005.

[11] Jeremy Elson, John R. Douceur, Jon Howell, and Jared Saul, "Asirra: a CAPTCHA that exploits interest-aligned manual image categorization", In Proceedings of ACM Conference on Computer and Communications Security, pp. 366-374, 2007.

[12] Oli Warner, http://www.thepcspy.com/kittenauth, KittenAuth Project

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

[14] http://www.puremango.co.uk

〈著者紹介〉

강 전 일 (Jeonil Kang) 학생회원
2003년 2월 : 인하대학교 컴퓨터 공학과 졸업
2006년 2월 : 인하대학교 정보통신대학원 석사
2006년 3월~현재 : 인하대학교 정보공학과 박사 과정
<관심분야> RFID 보안, 생체 인식 보안, 무선 센서 네트워크 , 무선 인터넷 보안, 웹 인증

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, 웹 인증 보안

김 군 순 (KoonSoon Kim) 학생회원
2006년 8월 : 인하대학교 컴퓨터 공학과 졸업
2008년 8월 : 인하대학교 정보통신대학원 석사
<관심분야> 생체 인식 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 조교수
<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 전임강사
<관심분야> 영상처리, 컴퓨터비젼, 인공지능, 패턴인식, 생체인식, 얼굴인식, 다중생체인식