Ihmisen ja koneen erottaminen toisistaan Internetissä

Antti Mattila

Tampereen yliopisto
Informaatiotieteiden yksikkö / vt
Pro gradu -tutkielma
Ohjaaja: Roope Raisamo
25.10.2011

Tampereen yliopisto
Informaatiotieteiden yksikkö / vt
Antti Mattila: Ihmisen ja koneen erottaminen toisistaan Internetissä
Pro gradu -tutkielma, 65 sivua
Lokakuu 2011

Tiivistelmä

CAPTCHA (*Completely Automated Public Turing test to tell Computers and Humans Apart*) on automatisoitu keino erottaa ihminen tietokoneesta. Nämä Internet-palveluiden väärinkäyttöä estävät tekstipohjaiset haasteet kyetään kuitenkin murtamaan tehokkaasti. Tähän asti ongelman ratkaisuksi on riittänyt haastekuvien monimutkaistaminen, mutta haasteiden vaikeuttamisella päädytään lopulta siihen, että haasteita eivät kykene ratkaisemaan koneet eivätkä ihmiset.

Tässä tutkielmassa käydään läpi useita visuaalisia ja auraalisia CAPTCHA-haasteita. Haasteista esitellään vahvuudet ja heikkoudet, ja tämän tiedon perusteella muodostetaan CAPTCHA-kehys auttamaan tulevaisuuden CAPTCHA-suunnittelua. Kehyksen mukaan suunnitellaan uusi sosiaaliseen kuvapalveluun perustuva CAPTCHA, TAGTCHA. Visuaalisissa CAPTCHA:ssa on usein ongelmana haastekuvien automaattinen generointi ja tarkistus. TAGTCHA ehdottaa tähän ratkaisuksi kuvapalvelun käyttäjien kuvien ja niihin merkittyjen tunnisteiden käyttöä haasteina ja niiden ratkaisuina. TAGTCHA:ssa on kaksi erilaista haastetyyppiä: tekstiversio, jossa käyttäjä kirjoittaa tekstikenttään kuvaa kuvaavia sanoja, ja monivalintaversio, jossa käyttäjä valitsee pudotusvalikosta kuvaa parhaiten kuvaavan sanan.

Suoritettujen käyttäjätestien perusteella TAGTCHA:n tekstiversio toimii huomattavasti paremmin kuin monivalintaversio, mutta tekstiversiokin onnistui tunnistamaan ihmisen vain hieman alle 50% suorituskerroista. Kuvapalveluun perustuvan CAPTCHA:n suurimmaksi ongelmaksi havaittiinkin vastausten tarkistus; kuvapalvelusta kuvien ohella hankitut oikeat vastaukset ovat vain yhden ihmisen mielipide kuvan sisällöstä, eivätkä välttämättä vastaa sitä, mitä kuva todellisuudessa esittää.

Asiasanat: CAPTCHA, HIP, kuvapohjainen CAPTCHA

Kiitokset

Tiedemaailmalle ehkä vaatimattoman mutta itselleni sitäkin suuremman merkityksen omaavan työni valmistumisesta tahtoisin esittää kiitokseni rakkaalle vaimolleni Merjalle jatkuvan tsemppauksen ja tukemisen johdosta.

Kiitos lopputyöni ohjauksesta Roope Raisamolle. Kiitos kuuluu myös työnantajalleni *Futuricelle* tutkielman kirjoittamisen mahdollistamisesta sekä Risto Sarvakselle ohjauksesta ja eteenpäin potkimisesta. Kiitoksen sana myös inspiraationlähde Eduard Khilille musiikin saralla tekemänsä elämäntyön johdosta.

Stay hungry, stay foolish.

Nokialla, Lokakuu 2011
Antti Mattila

Sisällys

1 Johdanto

Monet Internet-palvelut, kuten ilmaissähköpostit ja pokeripalvelut, ovat muodostuneet niistä saatavan suoran tai epäsuoran taloudellisen hyödyn takia houkuttteleviksi kohteiksi haittaohjelmille. Oikeiden ihmiskäyttäjien erottamiseksi tarvitaan testejä, jotka ovat ihmisille helppoja suorittaa mutta samalla koneille lähes mahdottomia. Paradoksaalisesti siis tietokoneet generoivat testejä, joita ne itse eivät osaa ratkaista. Tällä hetkellä yleisin tehtävätyyppi ovat tekstipohjaiset CAPTCHA:t (*Completely Automated Public Turing test to tell Computers and Humans Apart*), joissa käyttäjän tulee tulkita ja kirjoittaa tekstikenttään haastekuvassa oleva eri tavoin tarkoituksella vaikealukuiseksi tehty teksti (kuva 1.1). Ensimmäinen ihmisyyttä testaava sovellus tehtiin vuonna 1997 AltaVistalla [Lillibridge *et al.*, 2001], mutta CAPTCHA-termi itsessään kehitettiin vasta vuonna 2000 [Captcha.net, 2011].

Kuva 1.1 Google CAPTCHA.

Tekstipohjaisten CAPTCHA:jen murtamiseksi on tehty paljon tutkimustyötä (muun muassa [El Ahmad *et al.*, 2010], [Chellapilla & Simard, 2004], [Yan & El Ahmad, 2008a], [Chandavale & Sapkal, 2010]), jonka tuloksena on saatu selville nykyisten tekstipohjaisten ratkaisuiden heikko vastustuskyky hyökkäyksiä vastaan. Chellapilla ja muut [2005c] vertailivat koneiden ja ihmisten kykyä tunnistaa kirjaimia eri tavoin hankalalukuiseksi tehdystä kuvasta ja totesivat, että mikäli kone kykenee segmentoimaan kirjaimet omiksi kokonaisuuksikseen, se erittäin todennäköisesti kykenee myös tulkitsemaan kirjaimet usein jopa paremmin kuin ihminen. Useat teksti-CAPTCHA:t ovatkin murrettu onnistuneesti; esimerkiksi suosittu reCAPTCHA [von Ahn *et al.*, 2008] on onnistuttu murtamaan jopa 30 prosentin onnistumistodennäköisyydellä [Higgins, 2010].

Tekstipohjaisten haasteiden vaihtoehdoksi on kehitetty useita erilaisia kuvapohjaisia ratkaisuja (kuvassa 1.2 Asirra [Elson *et al.*, 2007], lisäksi esimerkiksi [Gossweiler *et al.*, 2009], [Rui & Liu, 2003], [Chew & Tygar, 2004b]). Kuvapohjaisissa CAPTCHA:ssa on usein ongelmana haasteen generointi; kuvat ja niiden ratkaisut joudutaan syöttämään sovelluksen tietokantaan käsin, koska tietokoneet eivät luonnollisesti kykene tulkitsemaan kuvista merkitystä. Tämän käsin teh-

tävän työn takia haastekokoelma voi jäädä pieneksi, mikä johtaa CAPTCHA:n heikkoon turvallisuuteen hyökkääjän selvittäessä kaikki haasteet käsin ja käyttäessään hyökkäyksessä esimerkiksi haastekuvien tarkisteita (engl. *hash*). Lisäksi koska kaikki haasteet ja ratkaisut ovat tietokannassa, tietokannan sisällön joutuminen vääriin käsiin tekisi koko haasteen merkityksettömäksi.

Kuva 1.2 Asirra.

1.1 Tutkimuskysymykset

Tekstintunnistusalgoritmien parantuessa teksti-CAPTCHA:ja joudutaan jatkuvasti monimutkaistamaan. Koska ihmisten kirjainten hahmotuskyky ei kuitenkaan vastaavasti jatkuvasti parane, tämä vaikeuttaa huomattavasti teksti-CAPTCHA:jen ratkaisemista. Tästä johtuen on kehitetty uudenlaisia CAPTCHA-sovelluksia perustuen tekstin lisäksi muun muassa kuviin, ääneen ja videoon. Tutkielmassani selvitän uusien sovellusten vahvuudet ja mahdolliset ongelmat sekä tutkin, onko niistä vanhojen menetelmien korvaajiksi.

Käyn olemassaolevat CAPTCHA-ratkaisut tutkielmassani kirjallisuuskartoituksenomaisesti läpi, luokittelen ne ja kokoan niistä vahvuudet sekä mahdolliset saavutettavuus- ja/tai käytettävyysongelmat ja puutteet turvallisuudessa. Tästä tiedosta muodostan hyvän CAPTCHA:n kehyksen (engl. *framework*), jota apuna käyttäen esittelen ja arvioin oman CAPTCHA-ehdotukseni.

Tekstintunnistustekniikoiden kehittyessä tekstipohjainen CAPTCHA on tiensä päässä. Tarvitaan jokin uusi tapa erottaa ihminen koneesta Internetissä, mutta ongelmia on useita: Miten CAPTCHA:sta saadaan mahdollisimman huomaamaton ja helppokäyttöinen siten, että se on myös turvallinen? Miten voidaan suun-

nitella sellainen CAPTCHA, että se voidaan ratkaista riippumatta käyttäjän laitteistosta, taidoista tai kyvyistä? Tätä varten ehdotan tutkielmassani CAPTCHA-kehystä. Tähän kehykseen perustuen ehdotan myös omaa CAPTCHA-sovellustani. Tutkielmani tutkimuskysymykset ovat seuraavat:

- *Millaisia vaihtoehtoisia CAPTCHA-toteutuksia on olemassa?* Käyn läpi aiemmin kehitetyt vaihtoehdot sekä tunnistan niiden vahvuudet ja heikkoudet.

- *Mitä ominaisuuksia vaaditaan hyvältä CAPTCHA-toteutukselta?* Tunnistamieni vahvuuksien ja heikkouksien perusteella muotoilen CAPTCHA-kehyksen hyvän CAPTCHA-toteutuksen suunnittelun avuksi.

- *Soveltuuko sosiaalinen kuvapalvelu CAPTCHA-ratkaisun haastemateriaalin toimittajaksi?* Luon esimerkki-CAPTCHA-toteutuksen, joka käyttää haastemateriaalin hankintaan Flickr-kuvapalvelua, testaan sitä testikäyttäjillä ja analysoin testin tulokset.

1.2 Tutkielman rakenne

Tutkielman luvussa 2 esittelen tarpeen erottaa tietokoneet ihmisistä verkkopalveluiden käyttäjinä sekä käyn läpi olemassa olevat keinot suorittaa tämä erottelu. Luvussa 3 tarkastelen tarkemmin erilaisten CAPTCHA-toteutusten ominaisuuksia käytettävyys- ja saavutettavuusnäkökulmista. Luvussa 4 esittelen hyvän CAPTCHA-toteutuksen vaatimuskehyksen ja luvussa 5 käyn läpi oman CAPTCHA-ehdotukseni, TAGTCHA:n, toimintaperiaatteen, toteutuksen ja testauksen. Luvussa 6 esittelen tutkielmani tulokset sekä pohdin kaikkia CAPTCHA:ja toteutusriippumattomasti koskevia ongelmia. Tutkielmani päättää yhteenveto luvussa 7.

Tässä tutkielmassa käytän ihmisen ja tietokoneen erottamista varten kehitetyistä tekniikoista nimitystä CAPTCHA. Näistä tekniikoista käytetään myös nimitystä HIP (Human Interaction Proof).

2 Kirjallisuuskatsaus ja olemassa olevat toteutukset

Internet-palveluiden luonne on viime vuosien aikana muuttunut. Aiemmin vain staattista informaatiota esittänyt väline on aiheuttanut web 2.0 -ilmiön myötä erilaisten sovellustyylisten verkkopalveluiden yleistymisen. Nämä Internet-palvelut, kuten esimerkiksi ilmaisia sähköposteja tarjoavat sivustot ja keskustelupalstat, haluavat rajoittaa palveluidensa käyttäjät ainoastaan yksityishenkilöihin, jotka käyttävät palvelua palvelun käyttöehtojen mukaisesti. Esimerkiksi Googlen Gmail- ja Microsoftin Hotmail-sähköpostipalveluiden tunnukset ovat roskapostittajien keskuudessa haluttuja, sillä näistä sähköpostidomaineista lähtevät viestit tulkitaan helpommin aidoiksi, ihmisten lähettämiksi viesteiksi kuin roskapostiksi.

Marraskuussa 1999 tietotekniikkaan erikoistunut sivusto slashdot.com julkaisi kyselyn, jossa haluttiin selvittää paras tietojenkäsittelytiedettä opettava yliopisto. Kyselyyn saattoi vastata vain kerran per IP-osoite, mutta Carnegie Mellon -yliopiston opiskelijat havaitsivat, että on mahdollista kiertää rajoite ja kirjoittaa tietokoneohjelma, joka äänestää toistuvasti heidän oppilaitoksensa puolesta. Välittömästi seuraavana päivänä MIT:n opiskelijat kehittivät vastaavanlaisen ohjelman, ja lopputulos oli 21 156 ääntä MIT:n puolesta, 21 032 ääntä Carnegie Mellonin puolesta ja kaikilla muilla yliopistoilla alle 1 000 ääntä per yliopisto [von Ahn *et al.*, 2003].

Näiden väärinkäytösten johdosta havaittiin, että ihmisen ja tietokoneen erottamiseen tarvittiin jokin keino. Ensimmäinen käytännön toteutus tehtiin vuonna 2001 AltaVistalla [Lillibridge *et al.*, 2001], tarkoituksena estää automaattisten bottisovellusten rekisteröitymistä verkkopalveluihin. Järjestelmä perustui periaatteeseen jossa käyttäjä tulkitsee tekstiä kuvasta, jossa on venytettyjä, kierrettyjä tai muulla tavoin vääristettyjä kirjaimia. Tämä haaste oli kuitenkin murrettavissa helposti jatkokehittämällä senhetkistä tekstintunnistusteknologiaa [von Ahn *et al.*, 2003].

Ei-toivottujen rekisteröitymisien lisäksi CAPTCHA:t ovat tarpeen, jos halutaan estää sivuston läpikäynti hakukoneiden boteilta, estää roskapostin lähettäminen tai torjua sanakirjahyökkäyksiä pyytämällä käyttäjää vahvistamaan salasanan syöttö CAPTCHA:lla muutaman epäonnistuneen yrityksen jälkeen [Captcha.net, 2011]. Haittasovellukset voivat huijata Internet-peleissä, kuten pokeri [Schluessler *et al.*, 2007], tai aiheuttaa mainostajille merkittäviä taloudellisia haittoja turhilla

mainosklikeillä [BusinessWeek, 2006].

Tätä väärinkäyttöä varten Internet-palveluihin on alettu sisällyttää erilaisia tapoja erottaa ihmiset tietokoneista. Edellä mainittu AltaVistan ratkaisu oli eräs, mutta yleisesti nämä CAPTCHA:t ovat erilaisia haasteita, jotka ovat suurelle osalle ihmisistä ratkaistavissa, mutta joita tietokoneet eivät kykene ratkaisemaan [von Ahn *et al.*, 2003].

Käytännössä kaikki [Yan & El Ahmad, 2008b] Internet-palveluissa käytettävät CAPTCHA-ratkaisut ovat edelleen sukua jo yli kymmenen vuotta vanhalle AltaVistan tekstipohjaiselle haasteelle, jossa käyttäjän tulee kirjoittaa tekstikenttään kuvassa oleva eri tavoin vaikealukuiseksi tehty teksti. Ajatuksena on ihmisten pystyvän helposti ratkaisemaan tehtävän, koska Gestalt-psykologian mukaan ihmiset ovat hyviä muodostamaan kokonaisuuksia pienistä osasista, kun taas ohjelmistoteknisesti tämä nähdään haastavana asiana [Yan & El Ahmad, 2008b].

2.1 CAPTCHA:n määritelmä

Vuonna 1950 Alan Turing tutki kysymystä "Voivatko tietokoneet ajatella?" Hän esitti kuuluisan testinsä, jossa haastattelijana toimivan ihmisen piti kysymyksiä esittämällä päätellä, haastatteleeko hän tietokonetta vai ihmistä [Turing, 1950]. Ihmisen ja tietokoneen erottamisessa Internetissä tilanne on päinvastainen; koneen tulee tunnistaa, onko käyttäjä ihminen vai tietokone. Ensimmäiset tähän tarkoitukseen kehitetyt ajatukset kirjoitti Moni Naor [1996] julkaisemattomassa käsikirjoituksessaan jo vuonna 1996, mutta siinä ei vielä ollut konkreettista ideaa ongelman ratkaisemiseksi. Ensimmäisen käytännön sovelluksen kehitetti AltaVista vuonna 2001 [Lillibridge *et al.*, 2001].

Itse CAPTCHA-käsite on peräisin vuodelta 2000 [Captcha.net, 2011] ja se on lyhenne sanoista *Completely Automatic Public Turing test to tell Computers and Humans Apart*. Määritelmänsä mukaan [Chew & Tygar, 2004b] CAPTCHA-testin on oltava

- helposti ihmisen ratkaistavissa,
- helposti luotavissa ja tarkistettavissa koneellisesti, sekä
- vaikea ratkaista koneellisesti.

Mielenkiintoisesti CAPTCHA:ssa on käänteinen ongelma kuin tietoturvassa yleensä; tavallisemmin hyökkääjillä on etu, koska hyökkääjät etsivät ensin kohteestaan haavoittuvuuden, ja vasta hyökkäyksen jälkeen kohdejärjestelmä voidaan paikata haavoittuvuuden varalta. CAPTCHA:ssa esitetään haaste hyökkääjälle, ja kun hyökkääjä saa haasteen murrettua, usein pienikin haasteen toteutuksen muuttaminen saa aikaan hyökkäyksen toimimattomuuden ja tarpeen muokata haittasovellusta uutta haastetta vastaavaksi.

Olemassa olevat CAPTCHA-ratkaisut voidaan jakaa karkeasti kolmeen pääryhmään. Pääryhmiä ovat visuaaliset CAPTCHA:t, äänipohjaiset CAPTCHA:t sekä muihin tekniikoihin perustuvat CAPTCHA:t. Visuaaliset jakautuvat puolestaan teksti-CAPTCHA:hin, kuva-CAPTCHA:hin sekä muihin visuaalisiin toteutuksiin.

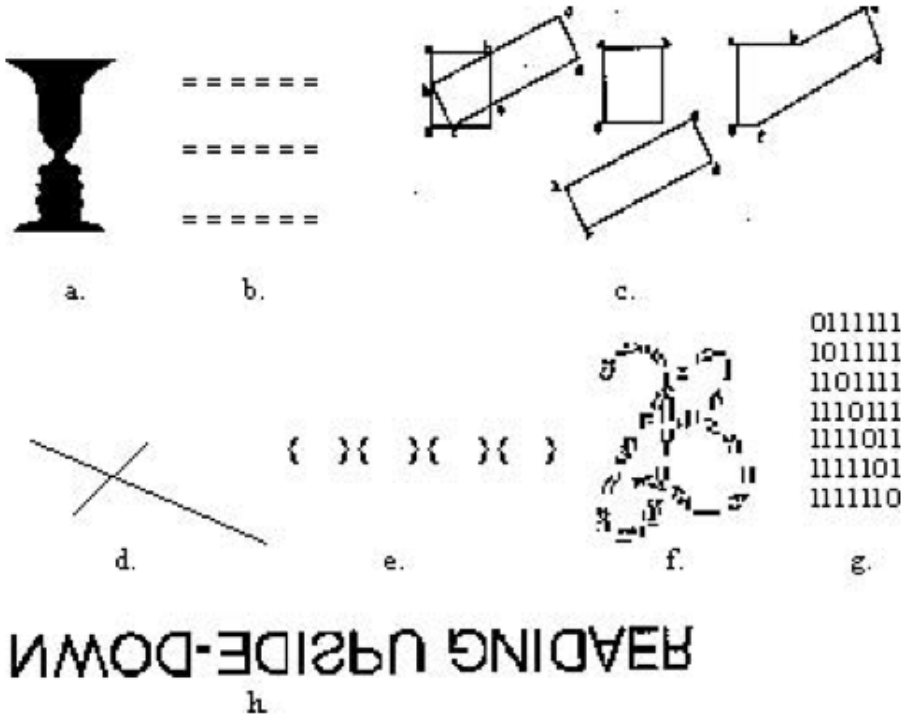
2.2 Visuaaliset CAPTCHA:t

Visuaaliset CAPTCHA:t perustuvat kuvan tulkitsemiseen ja sen perusteella haasteeseen vastaamiseen. Suurin osa CAPTCHA-ratkaisuista voidaan lukea tähän kategoriaan. Visuaalisuudella tarkoitetaan tässä muutakin kuin vain esimerkiksi valokuvia; esimerkiksi vääristetyn tekstin tulkitseminen kuuluu tähän kategoriaan.

2.2.1 Teksti-CAPTCHA:t

Yleisin, käytännössä ainoa laajassa käytössä oleva CAPTCHA-tyyppi on kuvapohjainen teksti-CAPTCHA. Näissä käyttäjä haastetaan tulkitsemaan haastekuvassa oleva teksti todistaakseen olevansa ihminen. Testi on tehokas, koska se hyödyntää Gestalt-psykologian mukaisesti ihmisen ominaisuutta kyetä hahmottamaan kokonaisuuksia paremmin kuin mitä koneellisesti on mahdollista [Chew & Baird, 2003]. Gestalt-psykologia perustuu havaintoon siitä, kuinka usein koemme asioita, joita emme kuitenkaan erikseen havaitse. Sen, mitä näemme, uskotaan olevan enemmän kuin vain osiensa summa. Gestalt-psykologia sisältää monia eri konstruktivistisia hahmolakeja [Rusu & Govindaraju, 2005]: läheisyyden, samankaltaisuuden, symmetrian, sulkeutuvuuden, jatkuvuuden, tuttuuden sekä kohde-alustan lait (kuva 2.1).

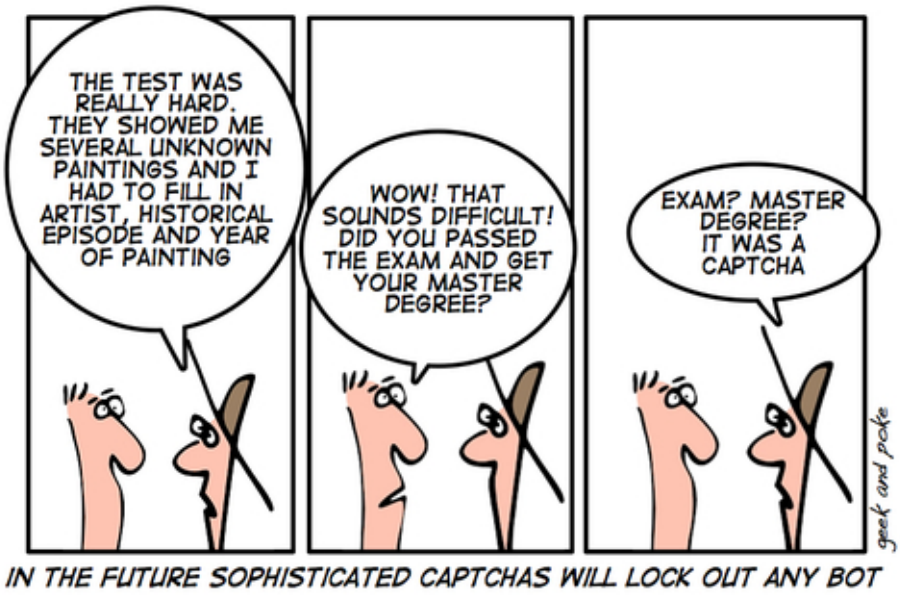
Tekstipohjaisten CAPTCHA:jen suosioon on useita syitä; haasteita voidaan luoda



Kuva 2.1 Hahmolait: kohde-alusta (a), läheisyys (b), tuttuus (c), jatkuvuus (d), symmetria (e), sulkeutuvuus(f), samankaltaisuus (g) ja muisti (h) [Rusu & Govindaraju, 2005]

nopeasti, tehtävä on helposti ymmärrettävissä ja kulttuurieroilla ei useinkaan ole suurta vaikutusta. Kirjainten tunnistus on asia, jota ihmiset ovat harjoitelleet koko elinikänsä, ja haasteesta voidaan luoda helposti useita miljardeja eri vaihtoehtoja jo kahdeksan merkin mittaisena. Lisäksi tekstintunnistusteknologian kenttä on hyvin tutkittu alue, ja näin ollen teksti-CAPTCHA:a suunniteltaessa voidaan helposti ottaa huomioon teknologian rajoitteet [Chellapilla *et al.*, 2005c].

Kuvapohjaisten teksti-CAPTCHA:jen suurin ongelma ovat segmentaatiohyökkäykset ja siitä johtuva haasteen vaikeustason nousu (kuva 2.2). Segmentaatiohyökkäyksessä hyökkäävän ohjelman tavoitteena on ensin saada haastekuvasta ylimääräinen kohina pois, sitten segmentoida eli ositella haastekuva yksittäisiksi kirjaimiksi ja sen jälkeen syöttää yksittäiset kirjaimet tekstintunnistusohjelmistolle, joka yrittää tunnistaa kirjaimen. Tätä prosessia voidaan vaikeuttaa monilla tavoilla, kuten sotkemalla, venyttämällä ja kiertämällä kirjaimia tai pilkkomalla kirjaimet osiin vaikeuttaen segmentointia, mutta haasteen vaikeuttaminen vaikeuttaa sitä samaan aikaan niin ihmisille kuin koneillekin.



Kuva 2.2 Geek & Poke: The Future of CAPTCHAs. Käyttöön saatu lupa.

Tekstipohjaisten CAPTCHA:jen tulkinta

Tekstipohjaiset CAPTCHA:t hyödyntävät usein edellä esiteltyjä hahmolakeja; Rusu ja Govindaraju [2005] ja Chellapilla ja muut [2005] ovat tutkineet, kuinka vääristää tekstiä siten, että se on edelleen vaivattomasti ihmisen tulkittavissa, mutta mahdollisimman hankala ratkaista tekstintunnistussovelluksia käyttäen. Chellapilla ja muut [2005] havaitsivat, että neuroverkkoihin perustuvalla tekniikalla pystyttiin tunnistamaan yksittäisiä kirjaimia jopa silloin, kun kirjaimet olivat erittäin vääristettyjä ja ylimääräisten sotkujen määrä oli suuri. Käyttäjäkokein he huomasivat ihmisten kirjaintentunnistuskyvyn pysyvän hyvänä kun kirjaimia siirreltiin, käännettiin tai skaalattiin, mutta vain vähäiset kirjainkohtaiset tai sanakohtaiset väännökset (engl. *warp*) heikensivät ihmisten kykyä tunnistaa kirjaimia. Ihmiset kykenivät myös selvittämään haasteita, joita oli vaikeutettu ohuilla viivoilla, paksuilla kirjaimiin osumattomilla viivoilla tai taustalla olevilla viivoilla. Ihmisten kyky ratkaista haasteita heikkeni huomattavasti, kun paksut viivat olivat kirjainten päällä.

Chellapilla ja muut [2005] huomasivat myös, että tietokoneet ovat ihmisiä parempia tunnistamaan yksittäisiä kirjaimia, kunhan kirjainten segmentointi on suoritettu onnistuneesti. Segmentointi on haastavaa sekä tietokoneille että ihmisille,

mutta testeissään he huomasivat, että ihmiset puolestaan taitavat segmentoinnin huomattavasti tietokoneita paremmin.

Tekstipohjaisten CAPTCHA:jen yhteiset tekijät

Tekstipohjaisista CAPTCHA:sta voidaan eritellä tiettyjä yhteisiä tekijöitä; toimintaperiaatteen lisäksi teksin tunnistamisen estämiskeinot voidaan ryhmitellä tiettyihin kategorioihin [Chellapilla *et al.*, 2005b] [Chellapilla & Simard, 2004]. Merkkeihin voidaan kohdistaa koko sanaan vaikuttava vääntö (*global warp*), yhteen merkkiin kohdistuva vääntö (*local warp*), siirto (*translation*), kääntö (*rotation*) tai skaalaus (*scale*). Tunnistusta voidaan hankaloittaa lisäämällä tekstin päälle pisteitä tai viivoja, jotka joko osuvat tai eivät osu kirjaimiin. Lisäksi haasteissa on usein käytetty taustaväriä, taustavärejä tai taustakuviota.

Teksti-CAPTCHA:ja, joissa ei ole käytetty muita tekniikoita kuin edellä luetellut, ovat muun muassa Yahoon Gimpy-R ja EZ-Gimpy, Microsoftin ja Googlen CAPTCHA:t, PessimalPrint [Coates *et al.*, 2001] sekä Baffletext [Chew & Baird, 2003]. Nämä ja useat muut ainoastaan edellämainittuja keinoja käyttävät CAPTHCA:t on onnistuttu murtamaan merkittävillä onnistumisprosenteilla [Chellapilla & Simard, 2004], [Mori & Malik, 2003], [Yan & El Ahmad, 2008a], [Chandavale & Sapkal, 2010].

Seuraavassa esittelen lyhyesti sellaiset eri teksti-CAPTCHA-ratkaisut, joiden toimintaperiaate on jollain tapaa kehittyneempi kuin edellä luetellut.

Tekstipohjaisia CAPTCHA:ja

- *reCAPTCHA*: Arvioiden mukaan maailmassa ratkaistaan yli sata miljoonaa CAPTCHA:a päivittäin [von Ahn *et al.*, 2008]. Tällä työmäärällä ei saavuteta muuta kuin käyttäjien pääsy kulloinkin kohteena olevaan järjestelmään. reCAPTCHA (kuva 2.3) käyttää hyväkseen tätä ilmaistyötä luodakseen uutta sisältöä haasteisiinsa digitoiden samalla vanhoja skannattuja kirjoja, joita tekstintunnistusohjelmistot eivät ole osanneet tulkita [von Ahn *et al.*, 2008]. Haasteessa käyttäjälle näytetään kirjoitettavaksi kaksi sanaa, joista toinen, tarkistesana, on järjestelmän tiedossa ja toinen on toistaiseksi tunnistamaton. Mikäli tarkistesana on vastattu oikein, järjestelmä ottaa toisen sanan sanavarastoonsa, kunhan se on varmistettu muilla käyttäjäl-

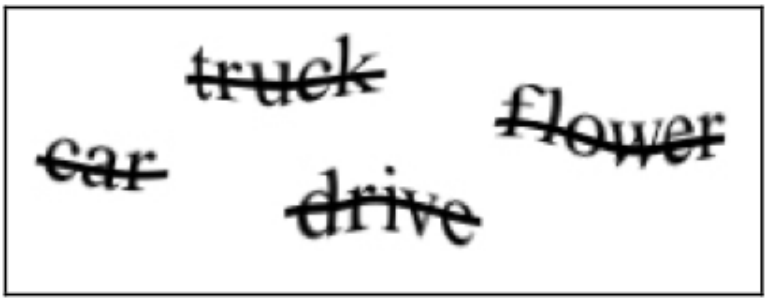
lä. reCAPTCHA, jota pidetään tällä hetkellä suositeltavimpana CAPTCHA:na [Captcha.net, 2011], on kuitenkin onnistuttu murtamaan jopa 30% onnistumistodennäköisyydellä [Higgins, 2010].

Kuva 2.3 reCAPTCHA-haaste.

- *AssoCAPTCHA* (kuva 2.4) perustuu tekstintunnistuksen lisäksi koneiden oletettuun kyvyttömyyteen assosioida sanojen merkityksiä. Haasteessa käyttäjälle esitetään neljä vääristettyä sanaa, joista yksi ei merkitykseltään kuulu joukkoon. Käyttäjän tulee lukea sanat, päätellä joukkoon kuulumaton sana ja kirjoittaa se tekstikenttään. Tekijät vakuuttavat AssoCAPTCHA:n olevan vahva suoja haittaohjelmia vastaan [Kulkarni, 2008], koska oletuksen mukaan 8-merkkisten teksti-CAPTCHA:jen onnistumisprosentti on alle 10%. Tästä he jatkavat päättelyä: jotta AssoCAPTCHA voidaan ratkaista, haittaohjelman täytyy onnistua ratkaisemaan kaikki neljä annettua sanaa oikein, jolloin todennäköisyys laskee $0,1^4$:ään, josta haittaohjelman pitäisi vielä pystyä päättelemään oikea sana, todennäköisyyden laskiessa vielä alemmaksi kuin 0,0001.

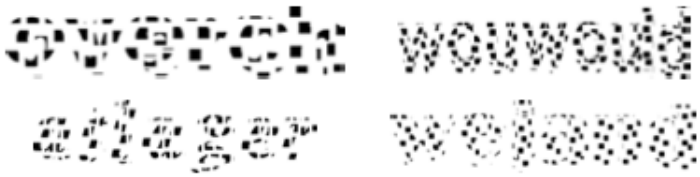
  Vaikka yksittäisen sanan onnistuneen tunnistuksen todennäköisyydeksi oletaisikin vain 10 prosenttia, pitäisi AssoCAPTCHA:n yhteydessä silti ottaa huomioon myös toinen näkökulma: entä jos haittaohjelma selvittääkin vain ensimmäisen näkemänsä sanan ja antaa sen vastaukseksi? Näin tehden AssoCAPTCHA:n voisi läpäistä 2,5% yrityskerroista; $10\%/4 = 2{,}5\%$. 2,5% taas menee reilusti yli 0,01%:n, jota pidetään toimivan CAPTCHA-haasteen koneellisena maksimiläpäisyrajana [Chellapilla *et al.*, 2005b]. Lisäksi AssoCAPTCHA:a heikentää taustalla toimiva tietokanta, josta järjestelmä hakee ja tarkistaa haasteet. Tämän tietokannan paljastuminen hyökkääjälle tekisi koko CAPTCHA:n täysin voimattomaksi. AssoCAPTCHA:n perustuminen tiettyyn sanavarastoon tekee siitä myös kieliriippuvaisen.

- *ScatterType*: Teksti-CAPTCHA:jen kaksi haastetta ovat segmentointi ja



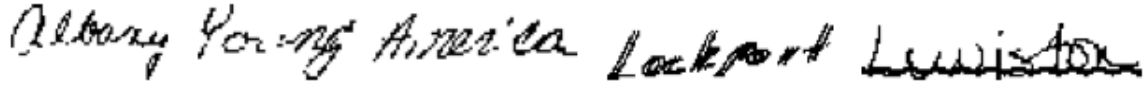
Kuva 2.4 AssoCAPTCHA-haaste.

kirjaimen tunnistus. Segmentointivaiheessa haastesana ositellaan erillisiksi merkeiksi, ja tunnistusvaiheessa nämä merkit tulkitaan. ScatterTypen (kuva 2.5) toiminta perustuu segmentoinnin vaikeuttamiseen leikkelemällä ja siirtelemällä kirjaimia lähelle toisiaan. ScatterTypen ongelmiksi havaittiin kuitenkin haasteen vaikeus ihmisille; pahimmillaan haasteen selvitti vain 53% testikäyttäjistä [Baird *et al.*, 2005].



Kuva 2.5 ScatterType, käyttäjien arvioimana keskitason vaikeusasteinen

- *Käsinkirjoitetun CAPTCHA:n* ajatus on käyttää haasteina skannattuja, käsin kirjoitettuja sanoja (kuva 2.6), joiden tunnistaminen on tietokoneelle hankalaa [Rusu & Govindaraju, 2005]. Näitä sanoja voidaan tarpeen mukaan myös vääristää aiemmin esiteltyjen periaatteiden mukaisesti. Ongelmana tässä lähestymistavassa on käyttäjien haasteenselvitysprosentti, joka parhaimmillaan oli 87,5% ja huonoimmillaan 67,7%. Käyttäjäystävällisyyttä suurempi ongelma on kuitenkin taustatietokannan tarve; järjestelmä ei kykene generoimaan uusia eikä tarkistamaan nykyisiä haasteita automaattisesti ilman, että haastekuvat ja niiden sisältämä teksti on tallennettu tietokantaan.



Kuva 2.6 Käsinkirjoitettu CAPTCHA

2.2.2 Kuva-CAPTCHA:t

Kuvapohjaiset CAPTCHA:t perustuvat usein kuvan sisällön tulkintaan, kuvassa olevan kohteen merkityksen selvittämiseen tai yhtenäisyyksien löytämisen usean kuvan kesken. Tietokoneilla on vaikeuksia esimerkiksi erottaa kissan kuva koiran kuvasta ([Elson *et al.*, 2007]), vaikka ihmiselle tehtävä onkin yksinkertainen.

Kuvapohjaisten kuva-CAPTCHA:jen murtamisesta ei ole juuri menestyksekkäitä menetelmiä julkaistu, mutta kuva-CAPTCHA:t eivät silti ole saavuttaneet suosiota. Tähän voi vaikuttaa monta seikkaa, suurimpana ihmisten tottuminen teksti-CAPTCHA:ihin, mutta muitakin syitä voi helposti päätellä: kuvat vievät paljon enemmän siirtokaistaa kuin teksti, aiheuttaen sivulatauksen hidastumista ja palvelinkuorman kasvua. Kuviin perustuvat CAPTCHA:t ovat usein myös ulkomitoiltaan suuria, eivätkä sovi yhtä hyvin Internet-palveluiden lomakkeille (mitoiltaan tai tyyliltään — tuskin esimerkiksi kukaan vakavasti otettava asianajotoimisto haluaisi yhteydenottolomakkeelleen kissanpentujen kuvia).

Kuvapohjaisia CAPTCJA:ja

- *ARTiFACIAL* perustuu ihmisten kykyyn tunnistaa ihmisen kasvot, vaikka ne olisivat vääristyneet, vain osittain näkyvissä tai huonosti valaistuja [Rui & Liu, 2003]. ARTiFACIAL-haasteessa on abstrakti kuvakollaasi, johon on piilotettu kasvot (kuva 2.7). Läpäistäkseen haasteen, käyttäjän on klikattava kuvaa kuudesti: neljä silmäkulmaa ja kaksi suun reunaa.

- *Asirra* esittää käyttäjälle kissojen ja koirien kuvia, pyytäen käyttäjää valitsemaan joukosta kaikki kissat. Kun yhdessä testissä valinta pitää tehdä 12 kuvan kohdalla (kuva 2.8), pelkän oikeinarvaamisen todennäköisyys on 0,024%. Koska tämä on enemmän kuin vaadittu 0,01%, Asirraan on rakennettu IP-osoitekohtainen yrityskertarajoitus, joka pienentää todennäköisyyden 0,002 prosenttiin [Elson *et al.*, 2007]. Asirra-haasteet ladataan ai-

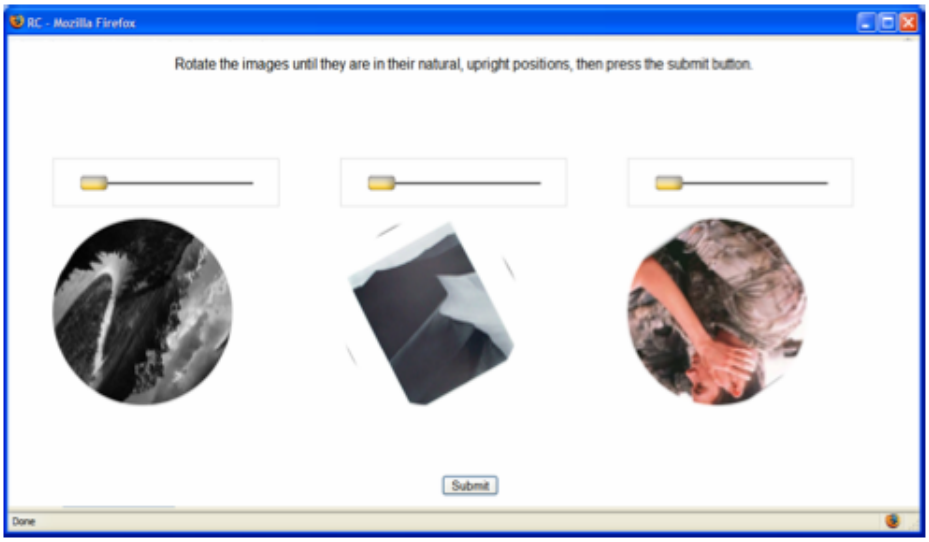
Kuva 2.7 ARTiFACIAL-haaste.

na dynaamisesti tietyltä Asirra-palvelimelta, joka käyttää taustalla PetFinder.comin [PetFinder, 2011] kuvatietokantaa. Näin ollen Asirrasta voidaan tunnistaa kaksi heikkoutta: toiminta vaatii selaimelta JavaScript-tuen sekä PetFinderin tietokannan vuotaminen vääriin käsiin peittoaisi koko CAPTCHA:n. Lisäksi Asirra on myöhemmin kyetty murtamaan jopa 82,7%:n todennäköisyydellä, mikä osoittaa kissan ja koiran erottamisen olevan koneellisesti mahdollista [Golle, 2008].

Kuva 2.8 Asirra-haaste.

- *What's Up CAPTCHA?* on Googlen kehittämä kuvapohjainen CAPTCHA [Gossweiler *et al.*, 2009]. CAPTCHA:ssa käyttäjälle esitetään kolme pyöreää kuvaa, joka käyttäjän täytyy pyörittää oikein päin oheista säädintä käyttäen (kuva 2.9). Pyöräytyksessä on 16 asteen virhemarginaali, joten näin ollen arvaamalla yhden kuvan oikeaan asentoon asettamisen todennäköisyys on 22,5%. Kolmelle kuvalle todennäköisyys on $0,225^3 \approx 1,1\%$, joka

on suurempi kuin yleisenä vaatimuksena pidetty 0,01%. Lisäksi CAPTCHA:n käyttöönotto vaatii tarkasti käsin valitun kuvakirjaston, koska kuvantunnistusohjelmat kykenevät tunnistamaan esimerkiksi maisemakuvien orientaation oikein. Näin ollen What's Up ei ole CAPTCHA:n määritelmän mukaisesti tietokoneella generoitavissa (Completely Automated), ja lisäksi se vaatii selaimelta toimiakseen JavaScript-tuen, joka rajaa osan käyttäjistä sen ulottumattomiin.



Kuva 2.9 What's Up CAPTCHA?

- *IMAGINATION* on kaksivaiheinen: ensimmäisessä vaiheessa käyttäjälle esitetään 800 × 600 pikselin kokoinen kuvakollaasi, joka on koostettu kahdeksasta kuvasta. Käyttäjän tulee hahmottaa kahdeksasta osittain päällekkäin asetellusta kuvasta mikä tahansa ja klikata sen keskikohtaa päästäkseen vaiheeseen kaksi, jossa hänelle esitetään vääristetty yksittäinen kuva. Yksittäisen kuvan vieressä on noin kymmenen sanaa, joista käyttäjän tulee valita kuvaa kuvaavin vaihtoehto [Datta *et al.*, 2005]. CAPTCHA:n murtamisesta ei ole julkaistu tutkimustuloksia, mutta 800 × 600 pikselin kokoisena se ei ole kovin helposti käyttöön otettavissa.

- *Klikattava CAPTCHA* haastaa käyttäjän valitsemaan ruudukkoon sijoitelluista sanoista oikeat englanninkieliset sanat pelkkien kirjainjonojen joukosta. Ruudukon sanat ovat tehty vaikealukuiseksi samalla tavalla kuin teksti-CAPTCHA:t [Chow *et al.*, 2008]. Arvaamalla CAPTCHA:n läpäisemisen

todennäköisyyttä voi säädellä kasvattamalla ruudukon kokoa ja säätämällä oikeiden sanojen lukumäärää. Näin tekemällä päästään alle 0,01%:n todennäköisyyteen. CAPTCHA kärsii kuitenkin samoista heikkouksista kuin teksti-CAPTHCA:t; se voidaan murtaa tekstintunnistussovelluksilla, ja tämän estämiseksi suoritettava tekstin vaikeuttaminen vaikeuttaa myös ihmiskäyttäjiä.

- *Sarjakuviin perustuva CAPTCHA* esittää käyttäjälleen neljäruutuisen sarjakuvan, jossa ruudut ovat väärässä järjestyksessä. CAPTCHA luottaa ihmisten taitoon ymmärtää huumoria. Käyttäjän tehtävänä on järjestää ruudut oikeaan järjestykseen [Yamamoto *et al.*, 2010a], mutta neljä ruutua voidaan järjestää vain 24 eri tavalla. Näin ollen pelkällä arvauksella haittaohjelmat voivat läpäistä CAPTCHA:n lähes 4,2%:n todennäköisyydellä.

- *SoylentGrid*-CAPTCHA haastaa käyttäjän tunnistamaan tekstiä kuvista. Testityyppejä on kolme: annotaatio, tunnistus ja molemmat yhdessä. Annotaatiossa käyttäjän tulee merkata kuvista tekstiä hiirellä raahaamalla, tunnistuksessa kirjoittaa kuvassa oleva teksti tekstikenttään ja yhdistetyssä testissä kirjoittaa valittu sana tekstikenttään (kuva 2.10) [Faymonville *et al.*, 2009]. reCAPTCHA-tyyliin haasteessa on kaksi kuvaa; toinen on kontrollikuva, toinen varsinainen haaste. Näin toimiessaan CAPTCHA kykenee luomaan uusia haasteita perustuen uusin tunnistettaviin kuviin. CAPTCHA ei kuitenkaan ole varsinaisesti “completely automated”, sillä uusien kuvien lisääminen haasteeseen tapahtuu käsin. Lisäksi CAPTCHA toimii JavaScriptin avulla, ja tarkkojen valintasuorakulmien piirtäminen on mahdotonta selaimilla, joissa ei ole JavaScript-tukea.

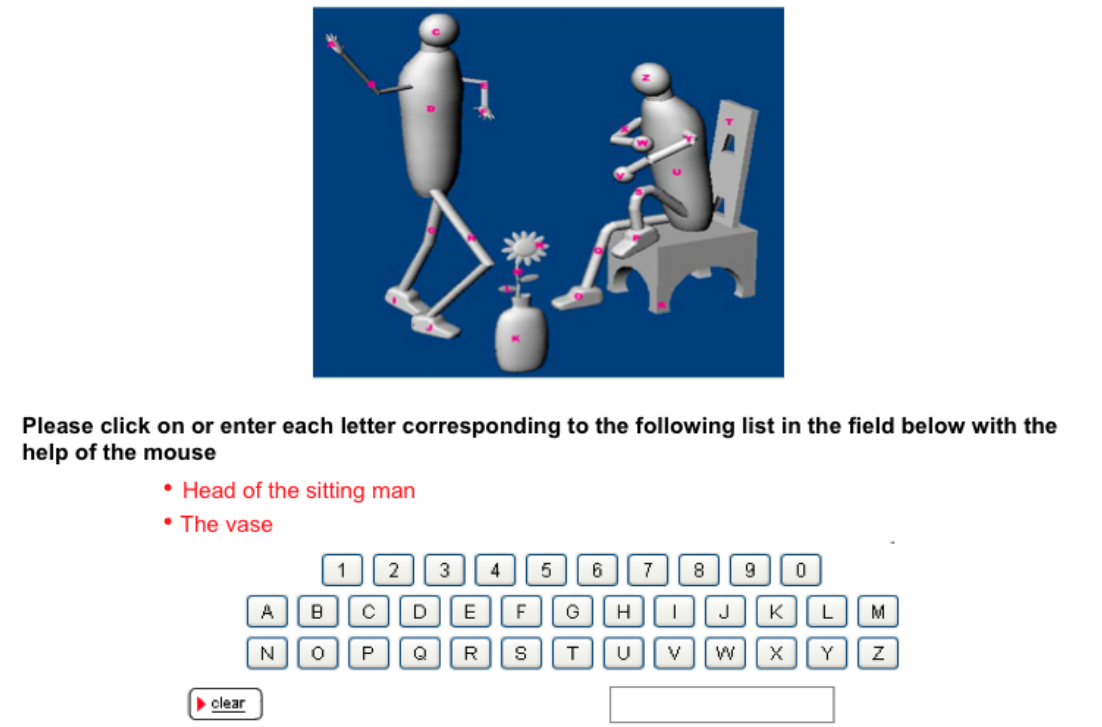
Kuva 2.10 SoylentGrid: Yhdistetty testi.

- *TagCaptcha* [Morrison *et al.*, 2009] toimii vastaavalla tavalla kuin SoylentGrid, mutta TagCaptcha:ssa on käytössä ainoastaan SoylentGridin tyylinen tunnistusmenetelmä. Jälleen reCAPTCHA:n henkeen käyttäjälle esitetään kaksi kontrollikuvaa ja yksi tuntematon. Jos käyttäjä vastaa kontrollikuviin oikein, tuntematon kuva saa merkitysehdotuksen. Kun tarpeeksi moni käyttäjä tunnistaa tuntemattoman kuvan samalla tavalla, siitä tulee yksi kontrollikuvista. Kuitenkin kuten SoylentGridissä, myös TagCaptchassa kuvat tulee syöttää sovellukseen yksitellen ihmisen toimesta.

- *Palapeli*-CAPTCHA:ssa käyttäjälle esitetään 3×3 – 5×5 palan kokoinen palapeli (kuva 2.11), jossa kahden palan paikkaa on vaihdettu. Käyttäjän tulee vaihtaa väärissä paikoissa olevien palojen paikkaa todistaakseen ihmisyytensä. Kuitenkin parhaimmillaankin 5×5 -kokoisena palapelissä on vain 300 vaihtoehtoa, joten arvaamallakin CAPTCHA:n voi läpäistä 0,33%:n todennäköisyydellä. Lisäksi tekijät itse kykenivät murtamaan CAPTCHA:n 9,4%:n todennäköisyydellä käyttäen koneellista kuvanlukusovellusta [Gao *et al.*, 2010b].

Kuva 2.11 Palapeli-CAPTCHA-haaste.

- *Multimedia-CAPTCHA* esittää käyttäjälle 3D-kuvan, jonka tietyt pisteet on merkitty aakkosin ja numeroin (kuva 2.12). CAPTCHA pyytää käyttäjää syöttämään tietyt pisteet (esimerkiksi istuvan miehen pää) ja näin vahvistaa käyttäjän olevan ihminen [Al-Sudani *et al.*, 2010]. CAPTCHA kuitenkin vaatii toimiakseen tietokannan, jossa nämä kuvat ja niihin liittyvä informaatio on tallennettu. Tuon tietokannan murtamalla hyökkääjä voi ohittaa CAPTCHA:n. CAPTCHA on myös kielisidonnainen, sillä tehtävän haasteet ovat esitetty englanniksi.



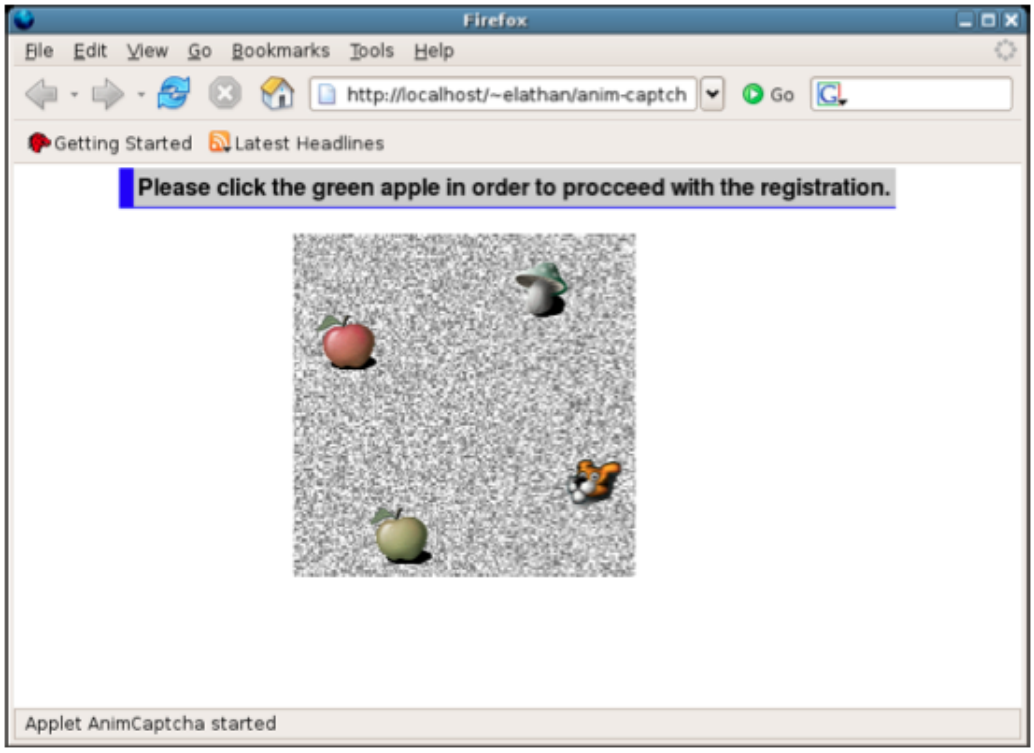
Kuva 2.12 Multimedia-CAPTCHA-haaste.

2.2.3 Muut visuaaliset toteutukset

- *Video-CAPTCHA* perustuu YouTube-videoiden tageihin. Käyttäjälle esitetään YouTubesta poimittu video, ja käyttäjää pyydetään kuvailemaan videota kolmella sanalla. CAPTCHA vertaa näitä sanoja YouTubessa videoon liitettyihin tageihin. Mikäli selkeä yhtäläisyys löytyy, käyttäjä tulkitaan ihmiseksi [Kluever & Zanibbi, 2009]. Tekijät onnistuivat kuitenkin murtamaan CAPTCHA:n jopa 13%:n todennäköisyydellä. Lisäksi CAPTCHA vaatii toimiakseen Flash-tuen, jota ei kaikissa selaimissa ole. CAPTCHA:n ratkaiseminen vie videon katselun verran aikaa (tosin tekijät huomauttavat, että videota ei ole pakko katsoa kokonaan).

- *Enhanced CAPTCHA* kehitettiin vastustamaan erityisesti eri sivustoilta tulevia CAPTHCA-pesuhyökkäyksiä (alakohta 6.2.2) vastaan. Animaatioon perustuva CAPTCHA esittää käyttäjälle neljä eri asiaa esittävää liikkuvaa kuvaa (kuva 2.13) ja pyytää käyttäjää klikkaamaan tiettyä niistä [Athanasopoulos & Antonatos, 2006]. CAPTCHA on toteutettu Internet-sivustolle upotettavana Java-appletina, ja näin ollen sen käyttökelpoisuus on rajoittunut käytännössä ainoastaan Javaa tukeviin tietokoneisiin, eikä se toimi

esimerkiksi kännyköissä.



Kuva 2.13 Enhanced CAPTCHA

- 3D-animatioon perustuva CAPTCHA esittää käyttäjälle tekstuaalisen haasteen samaan tapaan kuin teksti-CAPTCHAT:t, mutta kirjaimet näytetään yksi kerrallaan ja liikkuvana GIF-animaationa [Cui *et al.*, 2009] [Cui *et al.*, 2010]. CAPTCHA:n luojat ovat kehittäneet tavan animoida kirjaimia siten, että yksittäisestä animaation kuvasta ei voi päätellä haasteen ratkaisua. Kokonaiset kirjaimet ovat tulkittavissa vain animaation liikkuessa. CAPTCHA:n tekijät uskovat tämän takaavan hyvän suojan haittaohjelmia vastaan, mutta heillä ei ole toistaiseksi tutkittua tietoa väitteensä tueksi.

2.3 Äänipohjaiset CAPTCHA:t

Koska kaikki ihmiset eivät kykene ratkaisemaan visuaalisia CAPTCHA-haasteita, näkövammaisia varten on kehitetty äänipohjaisia haasteita. Äänipohjaiset CAPTCHA:t ovat tyypillisesti periaatteeltaan vastaavia kuin tekstipohjaiset CAPTCHA:t: käyttäjä kuulee ääninäytteen jossa luetellaan kirjaimia. Ääninäytteen koneellisen tulkinnan vaikeuttamiseksi siihen on lisätty usein taustalle hälyääniä,

taustamusiikkia tai kohinaa. Kuultuaan ääninäytteen, käyttäjän tulee kirjoittaa kirjaimet tekstikenttään, samaan tapaan kuin tekstipohjaisissa CAPTCHA:ssa. Tyypillisimmät äänipohjaiset haasteet kärsivät myös samantyylisistä ongelmista kuin tekstipohjaiset CAPTCHA:t: suosituimmat äänipohjaiset CAPTCHA:t on onnistuttu murtamaan jopa 71%:n todennäköisyydellä [Tam *et al.*, 2008], ja jos tekstintunnistusohjelmistot rinnastetaan käsitteellisesti äänentunnistusohjelmistoihin, päästään samaan varustelukilpaan kuin teksti-CAPTCHA:jen kanssa, kun haasteita vaikeutetaan sitä mukaa kuin äänentunnistusohjelmat kehittyvät. Äänipohjaisten haasteiden ratkaisu kestää käyttäjältä tyypillisesti kauemmin kuin yleisimpien tekstipohjaisten [Bigham & Cavender, 2009], ja äänipohjaiset vaativat käyttäjältä luonnollisesti äänentoistoon kykenevän laitteiston ja kuuntelun mahdollistavan ympäristön.

Perinteisestä audio-CAPTCHA:sta eroava CAPTCHA perustuu äänien tunnistamiseen [Holman *et al.*, 2007]: haasteessa käyttäjälle näytetään kuva objektista joka tulee tunnistaa. Näkökykyinen käyttäjä pystyy ratkaisemaan haasteen kuvan perusteella, mutta haasteen voi ratkaista myös äänen avulla, sillä jokaista kuvaa varten järjestelmässä on ääninäyte siitä, miltä kuvan objekti kuulostaa (esimerkiksi linnun laulua lintukuvalle). Holman ja muut totesivat näkövammaisten käyttäjien kykenevän ratkaisemaan äänentunnistus-CAPTCHA:nsa käytännössä virheettömästi. Ainoa ongelma heidän ratkaisussaan onkin taustatietokannan tarve; ääninäytteet, kuvat ja niiden merkitys täytyy olla kirjattuna tietokantaan, jotta koneellinen tarkistus olisi mahdollista. Jälleen tämän tietokannan murtuminen tai haaste haasteelta selvittäminen tekee CAPTCHA:n merkityksettömäksi.

2.4 Muut CAPTCHA-ratkaisut

Edellä kategorisoitujen CAPTCHA:jen lisäksi akateemisessa maailmassa on kehitetty myös muunlaisia CAPTCHA:ja. Etenkin täysin tekstipohjaisia, tekstin sisällön ymmärtämistä testaavia CAPTCHA:ja on kehitetty.

- *Vitseihin perustuvat CAPTCHA:t*: Knock-knock -CAPTCHA [Ximenes *et al.*, 2006] ja vitsi-CAPTHCA [Chew & Tygar, 2005] ovat kokonaan tekstipohjaisia, luetun ymmärtämiseen perustuvia CAPTCHA:ja. Molemmissa käyttäjän ihmisyyttä mitataan vitsien hauskuuden tulkitsemisella. Knock-knock -CAPTCHA esittää käyttäjälle etenkin Yhdysvalloissa suosittuja olevia kop-kop -vitsejä ("Knock knock? Who's there? Kenya. Kenya who? Ke-

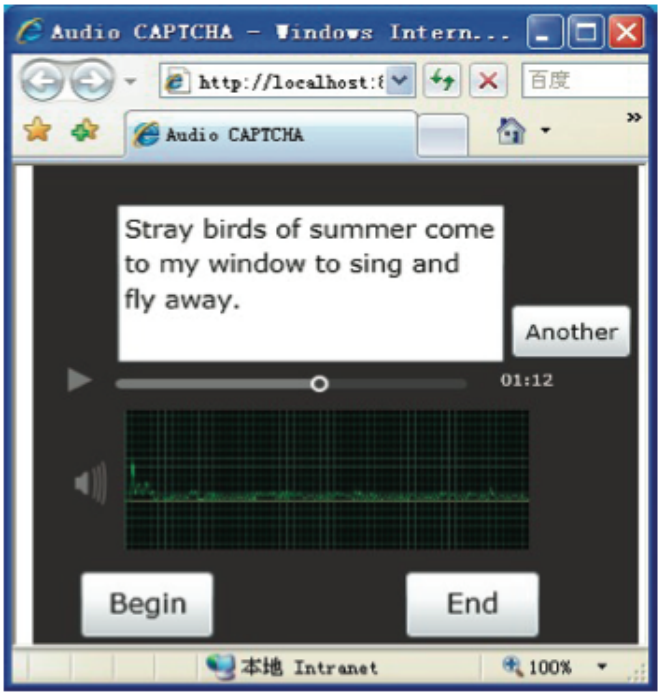
nya give me a hand"), jotka leikittelevät englanninkielisten sanojen ääntämisen samankaltaisuudella. Edellisessä esimerkissä "Kenya give me a hand?" vastaa lausetta "Can you give me a hand". Vitsi-CAPTCHA taas luetuttaa käyttäjälle vitsin tai vitsejä ja pyytää käyttäjää valitsemaan parhaan tai huonoimman vitsin, tai arvioimaan yksittäisen vitsin hauskuuden. Näissä lähestymistavoissa on jokunen ongelma: ensinnäkin ne ovat tiukasti sidottuja englannin kieleen, ja niiden suorittamiseen menee kauan aikaa, koska luettavaa tekstiä on paljon. Lisäksi Knock-knock -CAPTCHA:ssa vitsit täytyy olla tallennettu tietokantaan, ja koska haaste esittää käyttäjälle vain viisi vitsiä, arvaamalla valitsee 20% todennäköisyydellä oikean vaihtoehdon.

- Baird ja Bentley [2005] ehdottavat neljän kohdan suunnittelumallia *epäsuorille CAPTCHA:lle* (Implicit CAPTCHA): Haasteiden tulisi olla naamioitu tavanomaisiksi Internetin selailuun tarkoitetuiksi linkeiksi, haasteisiin tulisi voida vastata turvallisesti yhdellä hiiren klikkauksella, haasteisiin voi vastata onnistuneesti vain ymmärtäessään sen kontekstin, jossa haaste esitetään, ja haasteet ovat ihmiselle niin helppoja, että väärä vastaus voidaan luotettavasti tulkita haittaohjelmaksi. Baird ja Bentley esittävät erilaisia esimerkkejä, jotka kaikki perustuvat kuvan tietyn kohdan klikkaamiseen. Kuitenkin parhaimmillaankin nämä kuvat on mahdollista ohittaa arvaamalla noin 1,5%:n todennäköisyydellä, ja osa haasteista on sellaisia, joissa hyväksyttävästi klikattavat kohdat joudutaan tallentamaan tietokantaan.

- *SS-CAPTCHA* perustuu kirjoitetun kielen "outouteen", jonka ihminen tunnistaa helposti [Yamamoto *et al.*, 2010b]. Tällaisia outoja lauseita saadaan aikaiseksi tekstin automaattisilla käännöksillä kielestä toiseen. SS-CAPTCHA esittää käyttäjälle 15 lausetta, joista käyttäjän tulee valita viisi oikeaa lausetta. Näistä viidestätoista lauseesta kymmenen on käännetty ohjelmallisesti ensin käyttäjän äidinkielestä toiseen kieleen ja sitten taas takaisin äidinkieleen ja viisi on oikein kirjoitettuja. SS-CAPTCHA on siis kieliriippuvainen, tarvitsee tietyn tietokannan, josta hakea lauseita kääntämistä ja esittämistä varten, sekä vaatii käyttäjältä noin puolitoista minuuttia aikaa.

- *Puheentunnistus-CAPTCHA* perustuu audioon, mutta käyttäjän tuottamana. Gao ja muut [2010a] esittävät tutkimuksessaan tekniikan, jolla he voivat erottaa ihmisen äänen koneella tuotetusta. Haasteessa ihmiselle esite-

tään lause (kuva 2.14), joka käyttäjän tulee lukea tietokoneensa mikrofoniin. Sovellus erottelee ihmiset koneista äänen luonteenpiirteitä tutkimalla. Kuitenkin jo omassa tutkimuksessaan Gao ja muut onnistuivat murtamaan haasteensa jopa 48%:n todennäköisyydellä.



Kuva 2.14 Puheentunnistus-CAPTCHA.

2.5 Yhteenveto erilaisista toteutuksista

Kirjoittajalla on tiedossa edellä listattujen lisäksi myös muita CAPTCHA-toteutuksia, mutta ne ovat jätetty tämän tutkielman ulkopuolelle merkityksettöminä joko huonon suunnittelun, triviaalin koneellisen ratkaisun tai samankaltaisuuden takia. Lopuksi taulukossa 2.1 on tiiviisti esitelty kaikki edellä läpikäydyt CAPTCHA-ratkaisut. L/M%-sarake tarkoittaa läpäisy- ja murtoprosentteja, mikäli ne ovat tiedossa. Ominaisuus-sarakkeeseen on lyhyesti koottu ratkaisujen suurimmat puutteet ja lisäarvoa tuottavat ominaisuudet.

Taulukko 2.1: Yhteenveto esitellyistä ratkaisuista.

Nimi	**Tyyppi**	**L/M%**	**Ominaisuudet**
reCAPTCHA [von Ahn *et al.*, 2008]	Teksti	L: 77% M: 30%	Joukkoistaa (engl. *crowdsource*) käyttäjät luomaan lisää haasteita. Murrettavissa koneellisesti.
AssoCAPTCHA [Kulkarni, 2008]	Teksti	L: - M: 2,5%	Haasteet ja ratkaisut tietokannassa.
ScatterType [Baird *et al.*, 2005]	Teksti	L: 53% M: -	Vaikeuttaa koneellista segmentointia, mutta on hankala ihmisille.
Käsinkirjoitettu [Rusu & Govindaraju, 2005]	Teksti	L: 67,7% M: -	Haastekuvat ja ratkaisut tietokannassa.
ARTiFACIAL [Rui & Liu, 2003]	Kuva	L: 99,7% M: 0,2%	Vaatii tarkat klikkaukset kasvojen eri kohdissa, ja on täten hankala ratkaista pieninäyttöisillä mobiililaitteilla.
What's Up? [Gossweiler *et al.*, 2009]	Kuva	L: 81-94% M: 1,1%	Vaatii Javascript-tuen. Käytetyt kuvat täytyy valita huolellisesti käsin.
IMAGINATION [Datta *et al.*, 2005]	Kuva	L: 86% M: -	Suurikokoinen; koko 800×600 pikseliä. Haastekuvat ja ratkaisut tietokannassa.
Klikattava [Chow *et al.*, 2008]	Kuva	L: - M: -	Voidaan murtaa käyttäen samoja tekstintunnistusalgoritmeja kuin perusteksti-CAPTCHA:jen murtamiseen.
Sarjakuva [Yamamoto *et al.*, 2010a]	Kuva	L: - M: 4,2%	Vahvasti kulttuurisidonnainen. Haastekuvat ja ratkaisut tietokannassa.
SoylentGrid [Faymonville *et al.*, 2009]	Kuva	L: - M: -	Vaatii Javascript-tuen. Haastekuvat lisättävä käsin.

Nimi	**Tyyppi**	**L/M%**	**Ominaisuudet**
TagCaptcha [Morrison *et al.*, 2009]	Kuva	L: - M: -	Toimii samaan tyyliin kuin reCAPTCHA: joukkoistaa käyttäjät luomaan lisää kuvahaasteita.
Palapeli-CAPTCHA [Gao *et al.*, 2010b]	Kuva	L: - M: 9,4%	Tekijät itse kykenivät murtamaan haasteensa lähes kymmenen prosentin onnistumistodennäköisyydellä. Haastekuvat tietokannassa.
Multimedia-CAPTCHA [Al-Sudani *et al.*, 2010]	Kuva	L: - M: -	Kielisidonnainen. Haastekuvat ja ratkaisut tietokannassa.
Video-CAPTCHA [Kluever & Zanibbi, 2009]	Video	L: - M: 13%	Vaatii Flash-tuen. Ratkaisu vie videon katselun verran aikaa. Tekijät kykenivät murtamaan CAPTCHA:nsa jopa 13% todennäköisyydellä.
Enhanced CAPTCHA [Athanasopoulos & Antonatos, 2006]	Anim.	L: - M: -	Vaatii Java-tuen. Vastustaa erityisesti eri sivuilta tulevia CAPTCHA-pesuhyökkäyksiä (alakohta 6.2.2).
3D-animoitu [Cui *et al.*, 2009]	Anim.	L: - M: -	Esittää haasteen GIF-animaationa. Tekijät eivät ole julkaisseet testituloksia CAPTCHA:n toimivuudesta.
Äänipohjaiset [Tam *et al.*, 2008]	Ääni	L: 71% M: -	Suosituimmat äänipohjaiset CAPTCHA:t on onnistuttu murtamaan jopa yli 70 prosentin todennäköisyydellä.
Äänen tunnistus [Holman *et al.*, 2007]	Ääni	L: - M: -	Ääninäytteet ja ratkaisut tietokannassa.
Vitsi-CAPTCHA [Chew & Tygar, 2005]	Muu	L: - M: 20%	Vahvasti kulttuuri- ja kielisidonnainen. Vain viittä vaihtoehtoa tarjoavana liian helposti arvaamalla ratkaistavissa.

Nimi	**Tyyppi**	**L/M%**	**Ominaisuudet**
Epäsuora CAPTCHA [Baird & Bentley, 2005]	Muu	L: - M: 1,5%	Osa haasteista ja ratkaisuista tietokannassa.
SS-CAPTCHA [Yamamoto *et al.*, 2010b]	Muu	L: - M: -	Kielisidonnainen. Haasteet tietokannassa. Vaatii käyttäjältä noin puolitoista minuuttia aikaa.
Puheentunnistus [Gao *et al.*, 2010a]	Muu	L: - M: 48%	Tutkijat itse kykenivät murtamaan haasteensa lähes 50% todennäköisyydellä.

3 CAPTCHA:jen käytettävyys ja saavutettavuus

Tässä luvussa analysoin edellä esitellyt CAPTCHA-toteutukset käytettävyys- ja saavutettavuusnäkökulmista. Tätä analyysia hyväksi käyttäen esittelen luvussa 4 CAPTCHA-suunnittelukehyksen.

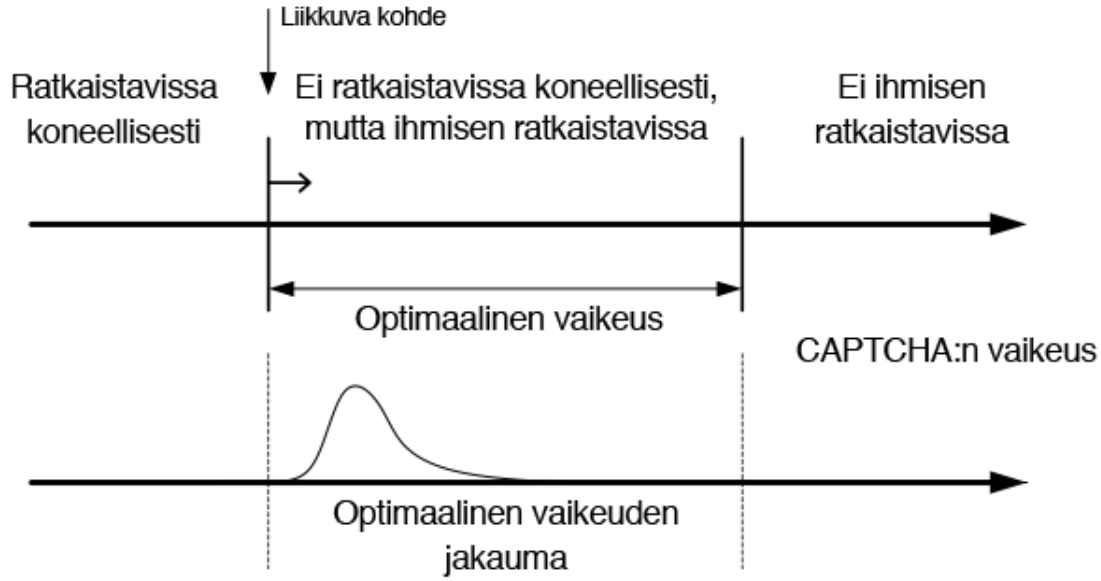
3.1 Tekstipohjaiset CAPTCHA:t

Tekstintunnistussovellusten kehittyessä tekstipohjaisten CAPTCHA:jen täytyy muuttua koko ajan vaikeaselkoisemmiksi. Valitettavasti tämän seurauksena CAPTCHA:sta tulee vaikeampia myös ihmisille; esimerkiksi Googlen omistamasta reCAPTHCA:sta pääsee läpi enää vain 77% ja Microsoftin käyttämästä CAPTCHA:sta 80% ihmisistä, vaikka testatut puhuivat äidinkielenään englantia [Bursztein *et al.*, 2010]. Fidasin ja muiden [2011] mukaan taas vain 48,5% käyttäjistä kykenee ratkaisemaan teksti-CAPTCHA:n ensimmäisellä yrittämällä. Jotta CAPTCHA voidaan katsoa toimivaksi, ihmiskäyttäjien ratkaisuprosentin pitäisi lähestyä vähintään 90 prosenttia [Chellapilla *et al.*, 2005b].

Yleisesti teksti-CAPTCHA:n ratkaisuun kuluu keskimäärin 9,8 sekuntia, mikä on Burszteinin ja muiden [2010] mukaan vielä käyttäjän hyväksyttävissä oleva aika, mikäli CAPTCHA:ja esitetään käyttäjälle vain harvoin [Bursztein *et al.*, 2010]. Alhaiseen keskimääräiseen suoritusaikaan vaikuttaa varmasti se seikka, että viimeisen kymmenen vuoden aikana, kun teksti-CAPTCHA:ja on ollut käytössä, ihmiset ovat ehtineet tottua niihin [Yan & El Ahmad, 2008b].

Kun teksti-CAPTCHA:ja vaikeutetaan riittävästi esimerkiksi venyttämällä ja kiertämällä kirjaimia sekä sotkemalla sanoja ylimääräisillä elementeillä tekstin päällä, päästään väistämättä ennemmin tai myöhemmin siihen tilanteeseen, että haastetta ei pysty ratkaisemaan ihminen eikä tietokone. Koska CAPTCHA-ongelmien ratkominen tietokoneella on sekä tiedemaailman että väärinkäyttäjien yhteinen intressi, tämä hetki lähestyy lähestymistään jatkuvasti. Tämä lähestyminen on kuvattu kuvassa 3.1 [Chellapilla *et al.*, 2005c], jossa esitetyn optimaalisen vaikeuden vasen raja liikkuu vääjäämättä kohti oikeaa tietokoneiden CAPTCHA-ratkaisukyvyn parantuessa ja ihmisen ratkaisukyvyn pysyessä kuitenkin paikallaan. Näinollen oikean haastavuuden löytäminen on alati vaikeampaa, ja haaste, joka on sopivan haasteellinen koneille tällä hetkellä, ei välttämättä ole enää tur-

vallinen hetken päästä.



Kuva 3.1 CAPTCHA:jen optimaalinen haastavuus

Sitä mukaa, kun kehitetään uusia keinoja tehdä tekstistä mahdollisimman vaikealukuista, osa tutkijoista on keskittynyt CAPTCHA:jen murtamiseen. CAPTCHA:jen murtamisella saavutetaan kaksi etua: Mikäli CAPTCHA saadaan murrettua, se saadaan murrettua ehkä ennen kuin jokin väärinkäyttäjä ehtii murtamaan sen. Toisaalta, kun tekstipohjainen CAPTCHA on murrettu, ollaan kenties ratkaistu jokin tekstintunnistukseen liittyvä ongelma.

Aiemmat tutkimukset CAPTCHA:jen murtamisen osalta osoittavat tekstipohjaisten CAPTCHA:jen olevan helposti murrettavia. Murrettu on ainakin EZ-Gimpy, Asirra, Microsoftin CAPTCHA, Googlen CAPTCHA ja Yahoo!:n CAPTCHA ([Chandavale & Sapkal, 2010], [Chellapilla & Simard, 2004], [Yan & El Ahmad, 2008a], [Chellapilla *et al.*, 2005a], [Mori & Malik, 2003], [Golle, 2008]).

Lisäksi reCAPTCHA, jota pidetään tällä hetkellä suositeltavimpana CAPTCHAna [Captcha.net, 2011], on onnistuttu murtamaan 30% onnistumistodennäköisyydellä [Higgins, 2010]. On osoitettu, että OCR-ohjelmistot kykenevät jopa parempaan kirjaintunnistukseen kuin ihmiset [Chellapilla *et al.*, 2005a]. Näin ollen voitaneenkin todeta tekstipohjaisten CAPTCHA-ratkaisuiden ajan olevan ohi.

Varsinaista tutkimusta CAPTCHA:jen käytettävyyden syistä ja seurauksista eivät ole tehneet muut kuin Yan ja El Ahmad [2008b]. He esittivät yksinkertaisen arviointikehyksen suunnaten sen pääsääntöisesti tekstipohjaisiin CAPTCHA:hin. He esittivät oletuksen tekstipohjaisten CAPTCHA:jen toimintaperiaatteen olevan käyttäjille helppo ymmärtää ja muistaa, ja että teksti-CAPTCHA:jen käytettävyyteen vaikuttavat tekijät voidaan jakaa kolmeen kategoriaan: vääristymä, sisältö sekä esitystapa. Tekijät ovat esitettynä taulukossa 3.1, ja nec käydään seuraavassa lyhyesti läpi.

Taulukko 3.1: Teksti-CAPTCHA:jen käytettävyyteen vaikuttavat tekijät.

Kategoria	**Ongelma**	
Vääristymä	Vääristymätapa ja -taso	
	Toisiinsa sekoittuvat merkit	
	Kielen ymmärtämättömyys	
Sisältö	Merkistö	
	Haasteen pituus	Kuinka pitkä?
		Ennustettavissa vai ei?
	Satunnaisia merkkejä vai sana sanakirjasta?	
	Loukkaava sana	
Esitystapa	Kirjasintyyppi ja -koko	
	Haastekuvan koko	
	Värien käyttö	
	Upotus websivuun	

Taulukossa ensimmäisenä mainitut vääristymätavat on kuvattu alakohdassa 2.2.1. Toisiinsa sekoittuvilla merkeillä tarkoitetaan merkkejä, jotka samannäköisyytensä puolesta voivat sekoittua toisiinsa, kuten O ja 0, tai vääristyksen yhteydessä yhdistyneitä merkkejä, kuten vv, joka näyttää w:ltä. Lisäksi tekstintunnistusta haittaavat ylimääräiset viivat saattavat näyttää kirjaimilta kuten I tai J. Suurin osa, 61,4% ihmisistä, kokee kirjainten vääristyksen olevan suurin CAPTCHA:n ratkaisua haittaava tekijä [Fidas *et al.*, 2011].

Tekstipohjaisia CAPTCHA:ja on usein ajateltu olevan hyvin kulttuuri- ja kieliriippumattomia. Kuitenkin ihmiset, joiden äidinkieltä ei kirjoiteta latinalaisin aakkosin, kokevat huomattavia hankaluuksia selvittää latinalaisin aakkosin toteu-

tettuja haasteita [Yan & El Ahmad, 2008b]. Samaan lopputulokseen päätyivät myös [Banday & Shah, 2011].

Sisältöön liittyvissä ongelmissa merkistöllä tarkoitetaan merkistön laajuutta; mikäli toisiinsa helposti sekoittuvat merkit jätetään pois, merkistö supistuu ja näin ollen vastausvaihtoehtojen määrä pienenee. Haasteen pituus vaikuttaa haasteen vaikeuteen niin ihmisille kuin koneillekin. Haasteen pituuden merkitys saattaa kuitenkin pienentyä, mikäli haasteen ratkaisuna on jokin oikea sana; ihmisille riittää tunnistaa osa sanasta, ja Gestalt-periaatteen myötä ihmiset kykenevät tunnistamaan oikean sanan tunnistamatta kuitenkaan jokaista yksittäistä kirjainta [Yan & El Ahmad, 2008b]. Ihmiset myös kirjoittavat oikeita sanoja nopeammin kuin sattumanvaraisia kirjaimia, joten CAPTCHA:n ratkaisunopeuskin paranee oikeita sanoja käytettäessä. Toisaalta oikeisiin sanoihin perustuva CAPTCHA on altis sanakirjahyökkäyksille. Sanakirjaan perustuvissa CAPTCHA:ssa on usein sopimattomat sanat poistettu.

Haasteen esitystavan suhteen kirjasintyyppi, -koko ja haastekuvan koko ovat selvästi asioita, jotka voivat aiheuttaa ihmisille enemmän hankaluuksia kuin koneille. Mikäli haaste on liian pieni, kone voi skaalata haastetta suuremmaksi, toisin kuin ihminen. Värien käyttö sen sijaan ei ole triviaali asia; kuten [Yan & El Ahmad, 2010] tutkimuksesta ilmenee, värien käyttö on usein ongelmallista käytettävyyden kannalta ja saattaa tehdä CAPTCHA:sta jopa helpommin murrettavan, jos kuvasta pystyy pelkän väri-informaation perusteella eristämään tekstin taustasta (kuva 3.2). Monet eniten käytetyistä CAPTCHA:sta, kuten reCAPTCHA, käyttävätkin vain mustavalkoisia värejä. Hieman yli viidennes (21,4%) ihmisistä kokee tekstin taustavärin tai -kuvion suurimpana ratkaisua haittaavana tekijänä, vaikka kuten väri, se aiheuttaa vain vähän tai ei ollenkaan lisäturvaa koneellista ratkaisua vastaan [Fidas *et al.*, 2011].

Kuva 3.2 EZ-Gimpy ja värien vaikutus murrettavuuteen. Molemmat versiot molemmista haasteista ovat yhtä helposti koneellisesti ratkaistavissa.

Yhteenvetona taulukossa 3.2 esitetään teksti-CAPTCHA:jen yleiset käytettävyys- ja saavutettavuusominaisuudet.

Taulukko 3.2: Teksti-CAPTCHA:jen käytettävyys ja saavutettavuus

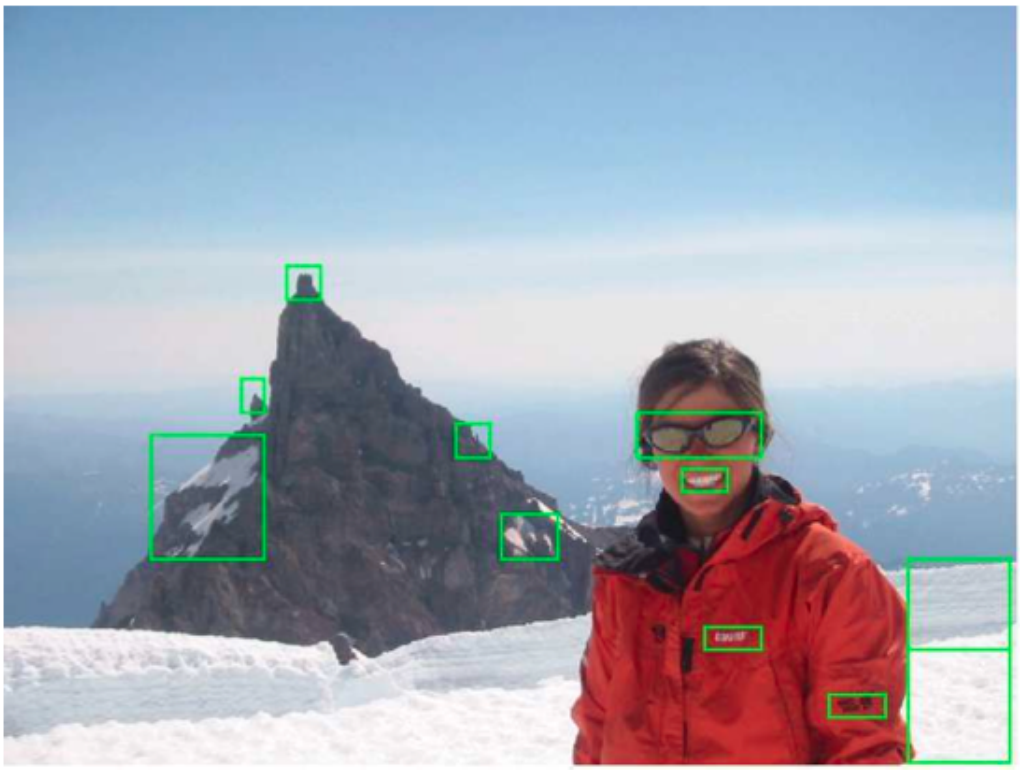
Matalaresoluutioinen näyttö	Ei vaikutusta
Sokea käyttäjä	Mahdoton ratkaista
Kuuro käyttäjä	Ei vaikutusta
Ei Flash-tukea	Ei vaikutusta
Ei JavaScript-tukea	Ei vaikutusta
Kulttuurisidonnainen	Ei
Kielisidonnainen	Mahdollisesti
Ratkaisuun kuluva aika	Lyhyt

3.2 Kuvapohjaiset CAPTCHA:t

Kuvapohjaiset CAPTCHA:t perustuvat usein haastekuvan tulkitsemiseen, kuten esimerkiksi yhdessä kuvassa olevan esineen tai asian nimeämiseen, tai yhteisen asian ymmärtämiseen useiden haastekuvien keskuudesta. Chew ja Tygar [2004a] esittivät oman toteutuksensa kummastakin edellämainitusta periaatteesta; nimeämis-CAPTCHA, jossa käyttäjän tulee kuuden kuvan perusteella kirjoittaa kuville yhteinen tekijä, erilaisuus-CAPTCHA, jossa käyttäjän tulee valita kuvien joukosta sinne kuulumaton, ja erottelu-CAPTCHA, joka perustuu eri kuvaryhmien erotteluun toisistaan.

Kuten Chew ja Tygar [2004a] huomasivat, sellaiset kuviin perustuvat CAPTCHA:t, joissa käyttäjän ei tarvitse kirjoittaa mitään, ovat vähemmän virheitä aiheuttavia ja käyttäjille mieluisampia kuin kirjoitusta edellyttävät CAPTCHA:t. Kirjoitusta vaativissa CAPTCHA:ssa ongelmia havaittiin sanojen oikeinkirjoituksen lisäksi synonyymeissa ja useaa asiaa tarkoittavien sanojen yhteydessä. Lisäksi erilaisuus-CAPTCHA:n havaittiin olevan kieliriippumaton, koska englantia äidinkielenään puhumattomat menestyivät testissä yhtä hyvin kuin englantia puhuvatkin. Kieliriippumattomuus esiintyy myös muissa kuvapohjaisissa CAPTCHA:ssa, kuten ARTiFACIALissa [Rui & Liu, 2003] ja What's Up CAPTCHA:ssa [Gossweiler *et al.*, 2009].

Kuvapohjaiset CAPTCHA:t voivat helposti muodostua hankakäyttöisiksi sellaisissa laitteissa, joissa näytön resoluutio on matala tai osoittimena toimii jokin muu kuin hiiri. Useat alakohdassa 2.2.2 läpikäydyistä kuvapohjaisista CAPTCHA:sta on suunniteltu normaalille tietokonekäyttjälle sillä oletuksella, että käytössä on tavallinen monitori ja hiiri. Tämä näkyy monessa toteutuksessa haasteiden kokoina; esimerkiksi IMAGINATION-haasteen koko on 800×600 pikseliä [Datta *et al.*, 2005]. Se tekee haasteen sijoittamisen kohdesivustolle erittäin vaikeaksi. Osoitinlaitteiden rajoittuneisuus tiettyjen CAPTCHA:jen ratkaisemisessa käy ilmi esimerkiksi Bairdin ja Bentleyn [2005] ehdottamissa ratkaisuissa: kun käyttäjän tulisi klikata jotain tiettyä kohtaa kuvasta, tarkka osoittaminen on hankalaa sormen peittäessä osan kosketusnäytön pintaa (kuva 3.3).



Kuva 3.3 Klikattavien kohteiden pieni koko kuvapohjaisessa CAPTCHA:ssa.

Osa kuvapohjaisista CAPTCHA:sta on toiminnallisuudeltaan monimutkaisempia kuin teksti-CAPTCHA:t, joissa usein riittää haastekuva ja tekstikenttä. Kuvapohjaisissa CAPTCHA:ssa toiminnallisuus saattaa riippua JavaScript-tuesta, kuten Asirrassa [Elson *et al.*, 2007] tai What's up CAPTCHA:ssa [Gossweiler *et al.*, 2009] tai selainliitännäisistä, kuten Flash-tukivaatimus video-CAPTCHA:ssa [Kluever & Zanibbi, 2009] tai Java-tukivaatimus Enhanced CAPTCHA:ssa [At-

hanasopoulos & Antonatos, 2006]. Mikäli CAPTCHA-toteutuksen toimivuus riippuu jostakin näistä teknisistä seikoista, sen saavutettavuus esimerkiksi matkapuhelinten selaimilla saattaa heikentyä merkittävästi.

Kuvahaasteiden suunnittelussa yhdeksi hankalimmista asioista nousee usein haastekuvien hankinta. Koska CAPTCHA:n määritelmän mukaan haasteiden tulee olla automaattisesti generoituja, on selvä tarve saada runsas määrä kuvia automaattisesti, mutta toisaalta mitkä tahansa kuvat eivät kelpaa haasteisiin. Esimerkiksi haasteissa, joissa käyttäjän tulee syöttää vastaukseksi kuvassa olevan asian tai esineen nimi, kuvien täytyy luonnollisesti esittää jotain yksiselitteistä asiaa tai esinettä. Lisäksi kuvista täytyy suodattaa pois sopimattomat, käyttäjille mahdollisesti vastenmieliset kuvat. Joissakin CAPTCHA:ssa, kuten What's Up CAPTCHA:ssa, on lisärajoitteina vielä haasteen toteutukseen liittyvät seikat; What's Up CAPTCHA:n tapauksessa vain sellaiset kuvat, jotka voidaan kääntää oikein päin, kelpaavat mukaan [Gossweiler *et al.*, 2009].

Yhteenvetona taulukossa 3.3 on esitelty kuvapohjaisten CAPTCHA:jen yleiset käytettävyys- ja saavutettavuusominaisuudet.

Taulukko 3.3: Kuvapohjaisten CAPTCHA:jen käytettävyys ja saavutettavuus

Matalaresoluutioinen näyttö	Vaikeuttaa ratkaisua
Sokea käyttäjä	Mahdoton ratkaista
Kuuro käyttäjä	Ei vaikutusta
Ei Flash-tukea	Saattaa vaikeuttaa ratkaisua
Ei JavaScript-tukea	Saattaa vaikeuttaa ratkaisua
Kulttuurisidonnainen	Mahdollisesti
Kielisidonnainen	Mahdollisesti
Ratkaisuun kuluva aika	Toteutusriippuvainen

3.3 Äänipohjaiset CAPTCHA:t

Vaikka valtaosa nykypäivänä ratkaistuista CAPTCHA:sta onkin tekstipohjaisia, niiden vaihtoehtona on usein näkövammaisille tarkoitettu äänipohjainen CAPTCHA. Äänipohjaiset CAPTCHA:t ovatkin merkittävässä roolissa tekstipohjaisten

CAPTCHA:jen vaihtoehtona; lähes 1% kaikista CAPTCHA:sta ratkaistaan äänimuodossa [Bursztein *et al.*, 2010].

Pääosa audio-CAPTCHA:sta toimii soittamalla käyttäjälle nauhoitteen, jossa toistetut kirjaimet ja numerot käyttäjän tulee kirjoittaa haasteen vastauskenttään. Periaate on siis vastaava kuin tavanomaisimmissa tekstipohjaisissa CAPTCHA:ssa, mutta koska äänentoisto on luonteeltaan sellaista, että se alkaa ja loppuu, CAPTCHA:n ratkaisu saattaa edellyttää käyttäjää toimimaan eri tavalla kuin teksti-CAPTCHA:ssa. Siinä missä teksti-CAPTCHA:n ratkaisussa käyttäjä voi minä ajanhetkenä hyvänsä tarkistaa haasteen minkä tahansa kohdan vain vilkaisemalla haastekuvaa, äänipohjaisessa CAPTCHA:ssa käyttäjä joutuu aloittamaan, pysäyttämään ja mahdollisesti kelaamaan äänitettä päästäkseen tiettyyn kohtaan haastetta.

Audio-CAPTCHA:n käyttäjät ovat usein näkövammaisia, jotka selaavat Internetiä ruudunlukijoilla (engl. *screen reader*). Äänipohjaisten CAPTCHA:jen käyttö ruudunlukijoilla on hyvin erilaista kuin hiiren, näppäimistön ja tavanomaisen näytön kanssa. Kun käyttäjä näkee haastesivun, hän voi siirtää kursorin vastauskenttään valmiiksi ja käynnistää sitten audiohaasteen toiston. Haastetta toistettaessa käyttäjän on helppo kirjoittaa kirjaimia sitä mukaan, kun niitä kuulee. Ruudunlukijoilla käytettäessä asia on hieman toisin; koska äänipalaute on kaikki, mitä ruudunlukijat sivustosta kertovat, voi helposti käydä niin, että siirtyessään toiston käynnistyspainikkeelta tekstikentälle haasteen jo toistuessa ruudunlukija puhuu haasteen päälle, tehden haasteen ratkaisusta vaikeaa tai jopa mahdotonta [Bigham & Cavender, 2009]. Bigham ja Cavender ehdottivatkin pieniä parannuksia audio-CAPTCHA-toteutuksiin; haasteen ratkaisua helpottaisi huomattavasti, mikäli äänentoistoa pystyisi ohjailemaan näppäimistöltä silloin, kun kursori on haasteen vastauskentässä (kuva 3.4).

Kuva 3.4 Parannettu versio äänipohjaisesta CAPTCHA:sta

Audion toiston lineaarisuudesta ja ruudunlukijaohjelmistojen käytöstä johtuen äänipohjaisen CAPTCHA:n ratkaiseminen kestää huomattavasti kauemmin kuin muuntyyppisten CAPTCHA:jen; käyttäjillä kului aikaa keskimäärin lähes puoli

minuuttia (28,4 sek.) [Bursztein *et al.*, 2010]. Bursztein ja muut [2010] havaitsivat lisäksi kolmen käyttäjän päätyvän samaan ratkaisuun keskimäärin vain 31,2 prosentissa ratkaisukerroista. Toisin kuin kuvapohjaisissa CAPTCHA:ssa, myös käyttäjän äidinkieli vaikuttaa ääni-CAPTCHA:jen ratkaisuaikaan merkittävästi; muuta kuin englantia äidinkielenään puhuvilla kestää ääni-CAPTCHA:n ratkaisemisessa keskimäärin 57% kauemmin kuin englantia äidinkielenään puhuvilla [Bursztein *et al.*, 2010].

Yhteenvetona taulukossa 3.4 on esitelty äänipohjaisten CAPTCHA:jen yleiset käytettävyys- ja saavutettavuussominaisuudet.

Taulukko 3.4: Audiopohjaisten CAPTCHA:jen käytettävyys ja saavutettavuus

Matalaresoluutioinen näyttö	Ei vaikutusta
Sokea käyttäjä	Ei vaikutusta
Kuuro käyttäjä	Mahdoton ratkaista
Ei Flash-tukea	Saattaa vaikeuttaa ratkaisua
Ei JavaScript-tukea	Saattaa vaikeuttaa ratkaisua
Kulttuurisidonnainen	Ei
Kielisidonnainen	Kyllä
Ratkaisuun kuluva aika	Pitkä

4 CAPTCHA-suunnittelukehys

Luotettavan, käytettävän ja helposti saavutettavan CAPTCHA:n täytyy selvästi täyttää tietyt ehdot. Esitän seuraavassa ehdotuksen suunnittelukehyksestä, jota voi käyttää apuvälineenä CAPTCHA:ja suunniteltaessa. Kehys perustuu CAPTCHA:n määritelmään [Chew & Tygar, 2004b] sekä tämän tutkielman lukuihin 2 ja 3.

Teksti-CAPTCHA:jen käytettävyyteen vaikuttavat tekijät on koottu taulukkoon 4.1.

Taulukko 4.1: Teksti-CAPTCHA:jen käytettävyyteen vaikuttavat tekijät.

Tehokkuus	Helposti, taloudellisesti ja satunnaisesti generoitava
	Helposti koneellisesti tarkistettavissa
	Alusta- ja teknologiariippumaton
Turvallisuus	Vaikea ratkaista koneellisesti
	Avoin
Käytettävyys	Ihmisen helposti ratkaistavissa
	Saavutettava
	Helppokäyttöinen
	Mahdollisimman vähän virheitä aiheuttava
	Kulttuuri- ja kieliriippumaton

Seuraavassa käydään läpi taulukossa 4.1 mainitut käytettävyyteen vaikuttavat tekijät:

- *1. Helposti, taloudellisesti ja satunnaisesti generoitava.* Haaste pitää olla mahdollista tuottaa koneellisesti mahdollisimman tehokkaasti ja nopeasti; esimerkiksi kuvapohjaiset CAPTCHA:t ovat usein ongelmissa, mikäli kuvamateriaali pitää koostaa itse.

- *2. Helposti koneellisesti tarkistettavissa.* Käyttäjän syöte pitää olla mahdollista verifioida välittömästi, ja palaute onnistumisesta tai epäonnistumisesta pitää antaa käyttäjälle heti.

- *3. Alusta- ja teknologiariippumaton.* CAPTCHA:n tulee olla käytettävissä myös mobiililaitteissa, screenreadereissa ja muissa rajoitetuissa ympäristöissä. Tämä sulkee pois muun muassa Javascriptin, Flashin, kuvat ja monimutkaiset käyttäjän toimet (ei esimerkiksi raahattavia elementtejä).

- *4. Vaikea ratkaista koneellisesti.* Oikein arvaamisen todennäköisyys tulee olla pieni (alle 0,01%), sillä haittaohjelmille CAPTCHA:n ratkaisuyritykset ovat halpoja ja nopeita [Chellapilla *et al.*, 2005b].

- *5. Avoin.* CAPTCHA:n tulee olla murtamaton myös siinä tapauksessa, että CAPTCHA:n lähdekoodi pääsee hyökkääjän käsiin. Näin ollen CAPTCHA:n lähdekoodi voi olla avointa alusta alkaen. CAPTCHA ei voi myöskään toimia suljetun tietokannan varassa, sillä tietokannan murtaminen tekee CAPTCHA:n triviaalisti ohitettavaksi.

- *6. Ihmisen helposti ratkaistavissa.* Käyttäjien ratkaisuprosentti tulisi olla lähes 90% [Chellapilla *et al.*, 2005b].

- *7. Saavutettava.* Myös aistirajoitteisten tulee olla mahdollista käyttää CAPTCHA:a.

- *8. Helppokäyttöinen.* CAPTCHA:n käyttöliittymän tulee olla helposti opittavissa ja omaksuttavissa, sekä sen tulee olla helposti muistettava. Itse haasteen tulee olla ratkaistavissa ilman, että käyttäjä joutuu keskeyttämään varsinaisen tehtävänsä (lomakkeen täytön) ja pysähtyä miettimään.

- *9. Mahdollisimman vähän virheitä aiheuttava.* Suunnittelussa tulee ottaa huomioon käyttäjän virheet, esimerkiksi syötteen oikeamuotoisuuden validointi (kuitenkin CAPTCHA:n turvallisuuden rajoissa) jo ennen lomakkeen lähetystä. Tehtävien täytyy olla vaivattomia ja suhteellisen nopeita ratkaista.

- *10. Kulttuuri- ja kieliriippumaton.* CAPTCHA pitää olla ratkaistavissa, vaikka käyttäjä ei ymmärtäisi haasteessa käytettyä kieltä tai haasteen yhteiskunnallisia viitteitä.

Tässä luvussa ehdottamani kehystä käyttöä voi käyttää CAPTCHA-suunnittelussa apuna. Kuitenkin tulee pitää mielessä, että se on vain kuvaus

optimaalisimmasta tavoitteesta; kaikkien kohtien täyttäminen ei välttämättä ole tarpeellista, ja toisaalta kaikkien kohtien täyttäminen ei takaa onnistunutta ja hyvää CAPTCHA:a. CAPTCHA:n toimivuuteen vaikuttaa suuresti esimerkiksi se konteksti, missä sitä on tarkoitus käyttää; Mikäli tiedetään järjestelmän käyttäjien olevan ainoastaan suomalaisia ja järjestelmä on tehty käytettäväksi vain matkapuhelimilla, nämä asiat voidaan ottaa huomioon CAPTCHA:a suunniteltaessa, ja näin saada siitä esimerkiksi suomalaisille käyttäjille helpommin ratkaistava.

5 TAGTCHA

Ratkaisuni idea on rakentaa CAPTCHA, joka hyväksikäyttää sosiaalisen kuvapalvelun käyttäjien tuottamaa massiivista kuva- ja teksti-informaatiomäärää. Esimerkkisovelluksessani käytetään Flickr-nimisen kuvapalveluun [Flickr, 2011b] käyttäjien lataamia kuvia ja niiden merkitystä kuvaavia sanoja (engl. *tag*). Ratkaisuni, jota kutsun nimellä TAGTCHA, hakee Flickristä sattumanvaraisesti näitä kuvia tietyin kriteerein ja esittää ne käyttäjälle, pyytäen käyttäjän tulkitsemaan kuvien merkityksen.

TAGTCHA:n toteuttamiseksi on kaksi mahdollisuutta: käyttäjälle esitetään yksi kuva, joka käyttäjän pitää tunnistaa ja kirjoittaa ratkaisu tekstikenttään yhdellä tai useammalla sanalla. Toinen vaihtoehto perustuu monivalintaan: käyttäjälle esitetään viisi kuvaa, joissa jokaisen vieressä on kymmenen vaihtoehdon pudotusvalikko. Käyttäjän tulee valita pudotusvalikosta kuvaa kuvaava sana, ja kaikkien viiden sanan ollessa oikeita käyttäjä tulkitaan ihmiseksi. Testaan näiden kahden vaihtoehdon toimivuutta koekäyttäjillä.

TAGTCHA:ssa on selvät yhtymäkohdat videopohjaiseen CAPTCHA:an [Kluever & Zanibbi, 2009]. Klueverin ja Zanibbin ehdotuksessa on kuitenkin havaittavissa puutteita, jotka TAGTCHA ottaa huomioon: toteutus käyttää Flash-pohjaisia videoita, jotka eivät toimi ilman selaimeen asennettua Flash-lisäosaa. Useissa mobiililaitteissa tuki on joko vajaa (esimerkiksi Android- ja Symbian S60-puhelimet) tai puuttuu kokonaan (Applen tuotteet), ja näin ollen CAPTCHA:a ei voida läpäistä. Lisäksi videomuotoisen CAPTCHA:n ratkaiseminen edellyttää käyttäjältä videon katsomisen, johon kuluu ratkaisun kirjoittamisen lisäksi videon keston verran aikaa. Näitä ongelmia ei ole kuvapohjaisessa TAGTCHA:ssa, sillä kuvat toimivat kaikissa nykyisissä mobiiliselaimissa.

Kuvapohjaisissa CAPTCHA:ssa, kuten TAGTCHA:n tekstinsyöttöversion kaltaisessa TagCaptchassa [Morrison *et al.*, 2009], yhtenä ongelmakohtana on kuvamateriaalin hankinta. Koska kuvapohjaisten CAPTCHA:jen vahvuus perustuu tietokoneiden tämänhetkiseen kyvyttömyyteen tulkita kuvia yhtä hyvin kuin ihmiset, tietokoneet eivät kykene generoimaan haasteita itse. Näin ollen kuvamateriaali ja sen ratkaisut täytyy syöttää järjestelmään käsin. Tämä taas on ristiriidassa CAPTCHA:n määritelmän kanssa: CAPTCHA:n tulisi olla automaattisesti generoitavissa ja koneellisesti tarkistettavissa. TAGTCHA tarjoaa tähän ratkaisuna sosiaalisen kuvapalvelun käyttämistä kuvalähteenä. Flickr perustettiin vuonna 2004, ja elokuussa 2011 palvelussa oli 6 miljardia kuvaa [Flickr, 2011a], joten kuvia

on tullut keskimäärin yli 200000 kappaletta päivässä. Näin suuri volyymi mahdollistaa Flickrin tulkitsemisen käytännössä automaattisena, kuvia ja ratkaisuja generoivana lähteenä.

5.1 Toimintaperiaate

Järjestelmä suorittaa tietyin väliajoin ajastetun prosessin, jossa se lataa omaan tietokantaansa uudet kuvat Flickristä. Mitä enemmän tai mitä useammin kuvia päivittäin ladataan, sitä enemmän vaihtelua tuleviin haasteisiin saadaan, mutta luonnollisesti sitä enemmän rasitetaan Flickriä ja sitä kauemmin kuvien päivitysoperaatio kestää. Tämä päivitysoperaatio suoritetaan kuitenkin taustaprosessina, jolloin se ei haittaa haasteiden tarjoamista käyttäjille.

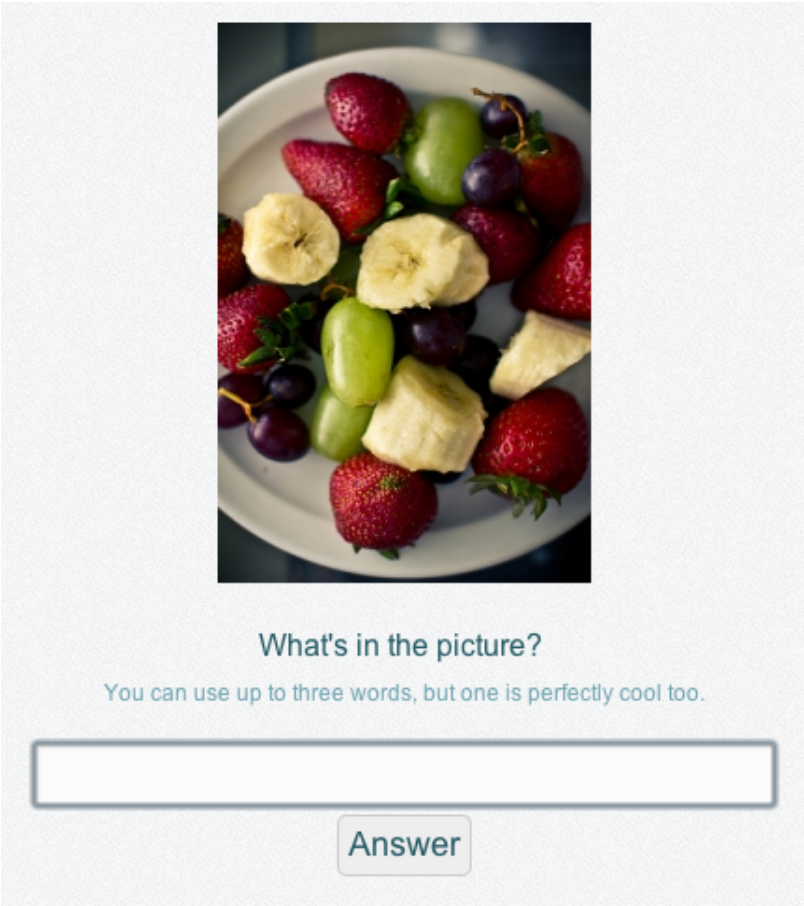
Järjestelmään voidaan määrittää kriteerit kuvien hakuun; on esimerkiksi mahdollista määritellä, että kuvalla on tietty määrä tageja (oletuksena haetaan vain sellaisia kuvia, joissa on vähintään viisi tagia). Koska Flickrissä käyttäjät voivat merkitä kuvansa mielivaltaisilla tageilla, tämä toimintatapa voi tuottaa odottamattomia tuloksia. Esimerkiksi lemmikkikoiraa esittävässä kuvassa tageina on ainoastaan lemmikin nimi tai jotain muuta sellaista, jota kuvan kohdetta tuntematon ei pysty pelkän yksittäisen kuvan perusteella tulkitsemaan. Tämän ongelman välttämiseksi on mahdollista määritellä tietyt tagit, jotka kuvista on pakko löytyä, mutta tätä ominaisuutta käytettäessä on huomattava käyttää niin laajaa sanalistaa, että pelkän sanalistan selvittämällä ei satunnaisesti arvaamalla pääse yli 0,01 prosentin onnistumistodennäköisyyteen. Kun tekstinsyöttöversioon voi syöttää maksimissaan kolme sanaa, täytyisi sanalistassa näinollen olla vähintään 30 000 sanaa. Tästä sanalistasta valitaan kuvapäivitysprosessin yhteydessä tietty määriteltävissä oleva määrä sanoja, joiden perusteella kuvatietokanta päivitetään.

Sanalistalla TAGTCHA:sta voidaan myös suodattaa pois ei-halutuntyyppiset kuvat; jos TAGTCHA esimerkiksi haluttaisiin sisällyttää tietotekniikkaan keskittyneeseen blogin kirjoitusten kommenttikentän yhteyteen, voitaisiin sanalista vaihtaa teknisempiä sanoja sisältäväksi.

5.1.1 Tekstinsyöttöön perustuva versio

Ensimmäinen TAGTCHA-versio toimii tekstinsyöttöön perustuen. Käyttäjälle näytetään haastekuva, jonka vieressä on tekstikenttä, johon ratkaisu kirjoitetaan

(kuva 5.1). Käyttäjä voi syöttää yhdestä kolmeen sanaa; syöte käsitellään melko löyhästi, joten käyttäjää ei tarvitse ohjeistaa sen enempää. Esimerkiksi syötteet “Man, Mouse, Car”, “man mouse car”, “man. mouse,,, car”, “MAN MouSE car” ja “man mouse car blue” tulkitaan kaikki samalla tavalla, sanoiksi man, mouse ja car.

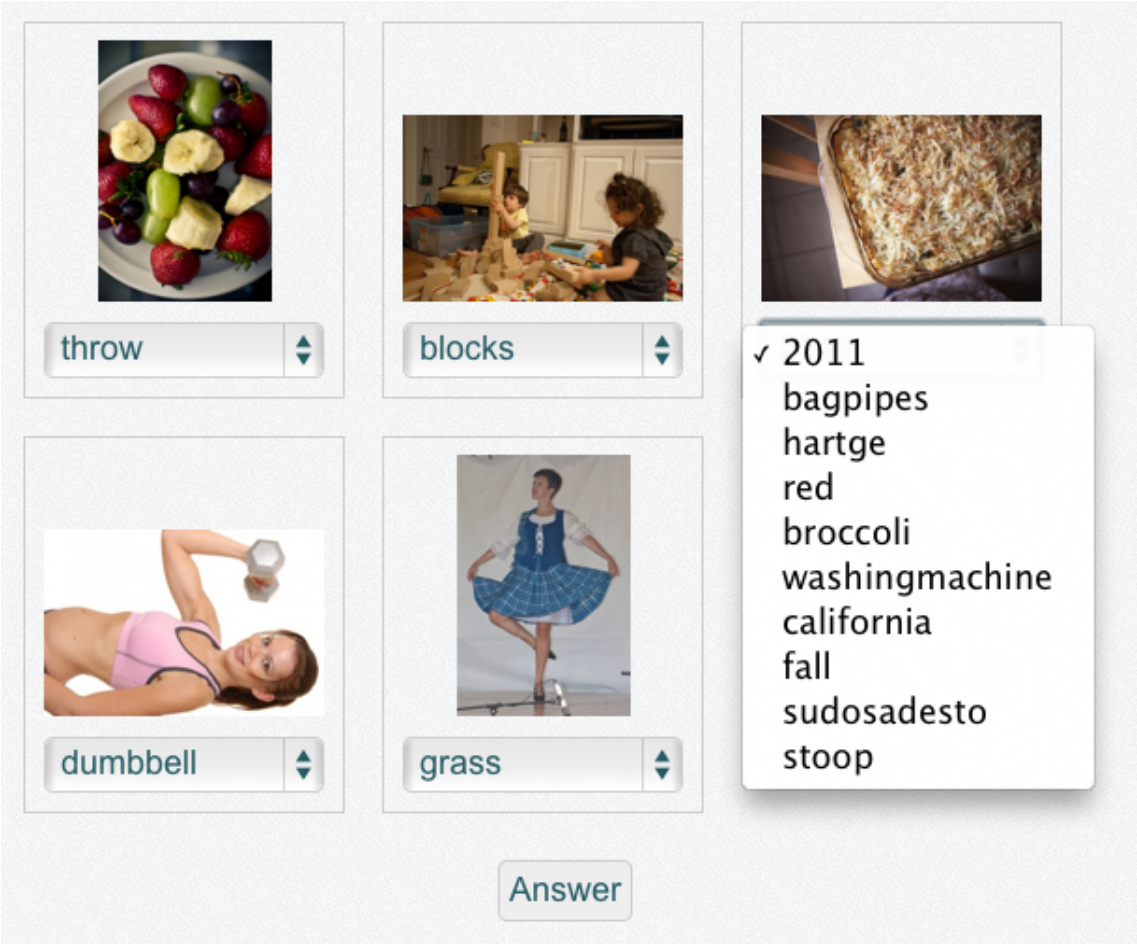


Kuva 5.1 TAGTCHA: Tekstinsyöttöhaaste.

Käyttäjän syöttämiä sanoja verrataan kuvaan liitettyihin Flickr-tageihin. Sovellus suorittaa sanojen vertailun muuntaen vertailtavat sanat pieniksi kirjaimiksi, jonka jälkeen vertailu tapahtuu sumeasti PHP-kielen *similar_text*-metodia [PHP, 2011] käyttäen. Mikäli yksikin käyttäjän syöttämä sana täsmää täysin tai vähintään 90-prosenttisesti mihin tahansa kuvaan liitetyistä tageista, käyttäjä voidaan tulkita ihmiseksi.

5.1.2 Monivalintaan perustuva versio

Monivalintaan perustuva versio TAGTCHA:sta hyödyntää samoja kuvia ja tageja kuin tekstipohjainenkin, mutta toisin kuin tekstipohjaisessa versiossa, monivalintaversiossa käyttäjän tulee valita oikea kuvaa kuvaava sana kirjoittamisen sijaan. Käyttäjälle esitetään viisi kuvaa, joiden jokaisen yhteydessä on kyseiseen kuvaan liittyvä pudotusvalikko (kuva 5.2).



Kuva 5.2 TAGTCHA: Monivalintahaaste.

Jokaisessa pudotusvalikossa on kymmenen vaihtoehtoa; yksi oikea ja yhdeksän väärää. Oikea vaihtoehto on valittu kuvaan liittyvistä tageista, ja yhdeksän muuta valitaan sattumanvaraisesti muihin kuviin liittyvistä tageista. Kuvien ja tagivaihtoehtojen määrä perustuu CAPTCHA-frameworkin kohtaan 3, *"Vaikea ratkaista koneellisesti"*: jotta CAPTCHA on turvallinen, arvaamalla läpäisemisen

todennäköisyyden tulee olla pahimmassa tapauksessa enintään 0,01%. Näinollen tuon todennäköisyyden saavuttamiseksi tarvitaan viisi kuvaa, joissa jokaisessa kymmenen tagivaihtoehtoa: $5 \times 10 = 10000$ vastausvaihtoehtoa.

5.2 TAGTCHA CAPTCHA-kehyksen näkökulmasta

TAGTCHA:n suunnittelussa käytin hyväkseni aiemmin luvussa 4 esittelemääni CAPTCHA-kehystä. Tässä luvussa käydään läpi TAGTCHA:n toiminnallisuuden CAPTCHA-kehyksen näkökulmasta.

TAGTCHA käyttää kuvalähteenään todella suuria määriä kuvadataa tuottavaa sosiaalista web-palvelua, Flickriä. Koska Flickrin käyttäjät lataavat ja merkitsevät kuvia palveluun yli 200 000 kuvan päivävauhdilla, TAGTCHA-haasteet ovat helposti, taloudellisesti ja satunnaisesti generoituvia. Koska TAGTCHA ratkaistaan syöttämällä tai valitsemalla tekstiä, myös TAGTCHA:n koneellinen tarkistus on triviaalia suorittaa tekemällä pelkkää tekstivertailua. TAGTCHA perustuu kuvien käyttöön, ja käytännössä kaikki nykypäivänä käytetyt Internet-selaimet riippumatta alustasta tukevat kuvia, ja täten mahdollistavat TAGTCHA:n ratkaisemisen alusta- ja teknologiariippumattomasti. Näinollen CAPTCHA-frameworkin näkökulmasta TAGTCHA täyttää kaikki tehokkuusehdot.

TAGTCHA:n turvallisuus perustuu tämänhetkisten kuvantunnistussovellusten kyvyttömyyteen tunnistaa kuvia. Kuvantunnistusohjelmat kykenevät tiettyihin spesifisiin tehtäviin, kuten kissojen erottaminen koirista [Golle, 2008], mutta yleisempi kuvantunnistus kuvasta riippumatta ei ole tutkimuksesta huolimatta vielä tarpeeksi luotettavalla tasolla [Li & Wang, 2008]. TAGTCHA:n lähdekoodin tutkiminen ei paljasta ratkaisusta muuta kuin toimintaperiaatteen, ja koska itse haasteet ja niiden ratkaisut syntyvät Flickr-käyttäjien toimesta, ratkaisun voidaan tulkita olevan avoin. Näin TAGTCHA täyttää CAPTCHA-frameworkin turvallisuusehdot.

TAGTCHA on kuviin perustuva CAPTCHA. Näin ollen sen ratkaisu on riippuvainen käyttäjän näkökyvystä. Mikäli haluttaisiin saada aikaan CAPTCHA, joka olisi kaikkien aistirajoitteisten ja -rajoittamattomien ratkaistavissa, tarvitsisi TAGTCHA:n yhteyteen liittää audiopohjainen CAPTCHA. Näin ei ole kuitenkaan tätä testiä varten tehty, vaan seikka huomioidaan mahdollisessa jatkokehityksessä.

Koska tekstipohjaisia CAPTCHA:ja on ollut jo kymmenen vuotta, ihmiset ovat tottuneet niihin ja osaavat odottaa sellaisen ratkaisemista esimerkiksi rekisteröityessään uuden palvelun käyttäjäksi. TAGTCHA:n tekstipohjainen haaste on periaatteeltaan hyvin samanlainen kuin perinteinen tekstipohjainen CAPTCHA; myös TAGTCHA:ssa käyttäjän tulee tulkita kuvasta sana ja kirjoittaa se tekstikenttään. Näin ollen TAGTCHA voidaan ajatella olevan ainakin yhtä helppokäyttöinen kuin tekstipohjainen CAPTCHA.

TAGTCHA on suunniteltu mahdollisimman vähän virheitä aiheuttavaksi. Tekstintunnistusversio hyväksyy käyttäjän syötteen useassa eri muodossa. Mikäli käyttäjä haluaa vastata monisanaisesti haasteeseen, sanojen erottimeksi kelpaa mikä tahansa ei-aakkosnumeerinen merkki, kuten välilyönti tai pilkku. Sanojen ei tarvitse myös olla täysin samassa muodossa kuin oikeat vastaukset ovat, sillä tarkistuksessa käytetään hyväksi sumeaa logiikkaa. Jo 90-prosenttisella vastaavuudella sanojen kesken voidaan todeta käyttäjän ihmisyys.

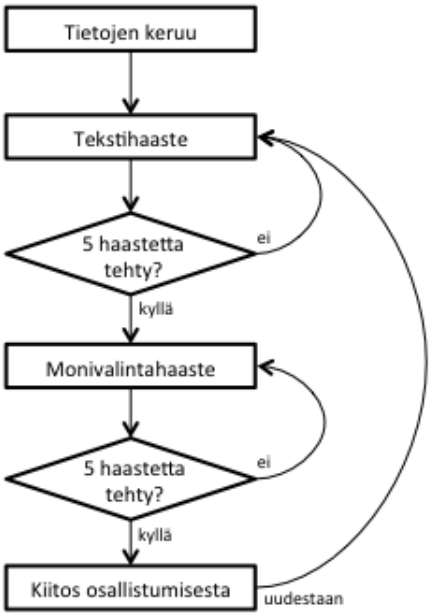
Koska Flickr-palvelun käyttäjät ovat valtaosin englantia puhuvia, sisältö on luonnollisesti myös englanninkielistä, mikä tekee haasteesta vaikean tai mahdottoman englantia ymmärtämättömille käyttäjille. TAGTCHA:n periaatteella toimiva CAPTCHA on kuitenkin mahdollista rakentaa käyttämään taustapalveluna jotakin toista sosiaalista kuvapalvelua, mikäli tarpeellista.

Tässä kohdassa on käyty läpi TAGTCHA:n ominaisuudet CAPTCHA-kehyksen näkökulmasta. Voidaankin todeta TAGTCHA:n täyttävän lähes kaikki kehyksen kohdat.

5.3 Toteutus

Järjestelmä on toteutettu PHP:lla ja se käyttää MySQL-tietokantaa tarpeellisen tiedon säilyttämiseen. Tietokantaan on tallennettu Flickristä haettujen kuvien nimet sekä kuviin liitetyt tagit.

Kun käyttäjä saapuu haastesivulle, hän saa eteensä kuvan ja tekstikentän, johon kirjoittaa yhdestä kolmeen sanaa siitä, mitä kuva esittää. Lomakkeen lähettämisen jälkeen järjestelmä pilkkoo käyttäjän syötteen osiin (mikäli sanoja on enemmän kuin yksi), vertaa käyttäjän syötettä esitetyn kuvan tietokannasta saatuihin tageihin, ja mikäli syötteestä löytyy yksikin täsmäävä sana, järjestelmä päättelee käyttäjän olevan ihminen. Prosessi on esitetty kaaviona kuvassa 5.3.



Kuva 5.3 Prosessikaavio järjestelmän toiminnasta.

Monivalintahaasteessa käyttäjä valitsee viidestä monivalintavaihtoehdosta kutakin kuvaa vastaavan vaihtoehdon ja lähettää lomakkeen. Sovellus vertaa käyttäjän valintoja oikeisiin ratkaisuihin, ja mikäli kaikki viisi vaihtoehtoa ovat oikein, käyttäjän ihmisyys on varmistettu.

PHP-sovellus on rakennettu yksinkertaisuudestaan johtuen erittäin kevyen BareBones MVC-kehyksen päälle [BareBones, 2009]. Flickr-rajapintatoteutus käyttää hyväkseen Flickrin suosittelemaa [Flickr, 2011c] phpFlickr-kirjastoa [Coulter, 2011].

5.4 Testaus

TAGTCHA:a testattiin luomalla yksinkertainen verkkosovellus, joka tarjosi käyttäjille peräkkäin yhteensä kymmenen haastetta per käyttäjä, viisi tekstipohjaista ja viisi monivalintapohjaista. Näistä haasteista tallennettiin tieto käytetystä kuvasta, käyttäjän syöte, järjestelmän tulkinta käyttäjän ihmisyydestä sekä kyseisen haasteen ratkaisuun kulunut aika. Lisäksi jokaisesta käyttäjästä tallennettiin vapaaehtoista tietoa: ikäluokka (alle 21, 21–25, 26–30, 31–40 ja yli 40 vuotta) ja englannin kielen taito asteikolla sujuva, hyvä, keskinkertainen, huono.

Testien suorituksen tarkoituksena oli selvittää TAGTCHA:n eri versioiden toimi-

vuus CAPTCHA-ratkaisuna:

- kuinka onnistuneesti käyttäjät kykenevät ratkaisemaan haasteet,
- haasteiden ratkaisuun kuluva aika sekä
- sosiaalisen kuvapalvelu Flickrin sopivuus haastelähteeksi.

Järjestelmää testattiin 30.9.2011-5.10.2011 välisenä aikana osoitteessa http://puerimor.futupeeps.com. Testaajat käyttivät testisuorituksessa omaa tietokonettaan, valitsemaansa Internet-selainta sekä tekivät testit haluamaansa aikaan haluamassaan paikassa. Testattavia ei valvottu eikä ohjeistettu testien aikana.

5.5 Järjestelmän konfiguraatio

Testauksessa käytettiin seuraavia asetuksia:

- Sallittujen sanojen määrä tekstihaastevastauksissa: 1-3
- Haasteeseen hyväksyttyjen kuvien minimitagimäärä: 5
- Kuvia vain tiettyjen tagien perusteella: kyllä

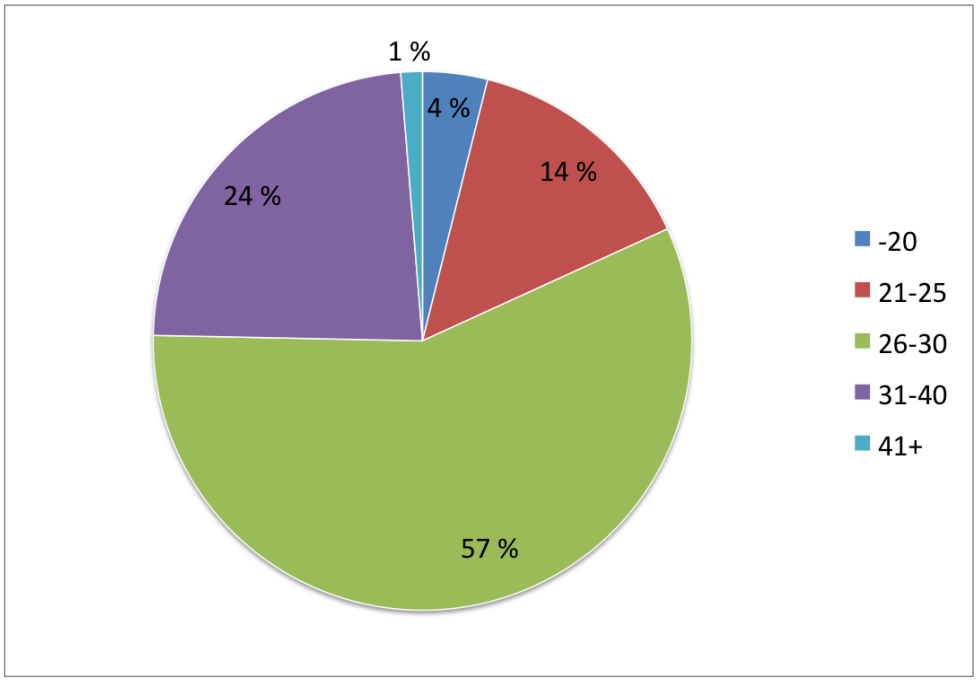
TAGTCHA-testiversion sanalistassa oli 632 vaihtoehtoa. On huomioitava, että sanalista on liian lyhyt, mikäli käytettäisiin TAGTCHA:n tekstinsyöttöversiota ja haluttaisiin riittävä turvallisuus sanalistan julkisuuteen vuotamista vastaan, mutta TAGTCHA:n testausta varten lyhyemmän sanalistan katsottiin olevan riittävä.

Sekä tekstinsyöttö- että monivalintahaasteissa käytettiin samaa kuvakokoelmaa. Sovellus piti huolen siitä, että käyttäjälle ei esitetty samaa kuvaa kahdesti.

5.6 Testikäyttäjät

Testikäyttäjiä oli 77 kappaletta. Suurin osa käyttäjistä oli 26-30-vuotiaita, hyvin tai erinomaisesti englantia osaavia. Ikäluokat jakaantuivat diagrammin 5.4

mukaan. Englannin kielen taitojakauma on esitetty kuvaajassa 5.5. Testikäyttäjät on hankittu kirjoittajan työpaikalta ja Facebook-verkkopalvelussa julkaistun ilmoituksen avulla.



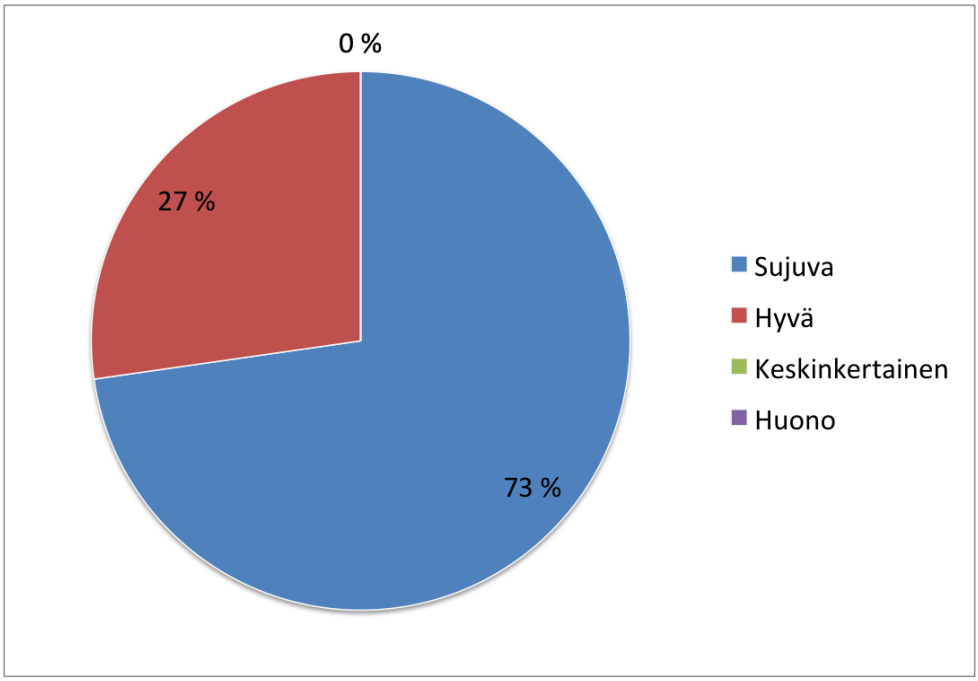
Kuva 5.4 Testikäyttäjien ikäjakauma.

5.7 Testitulokset

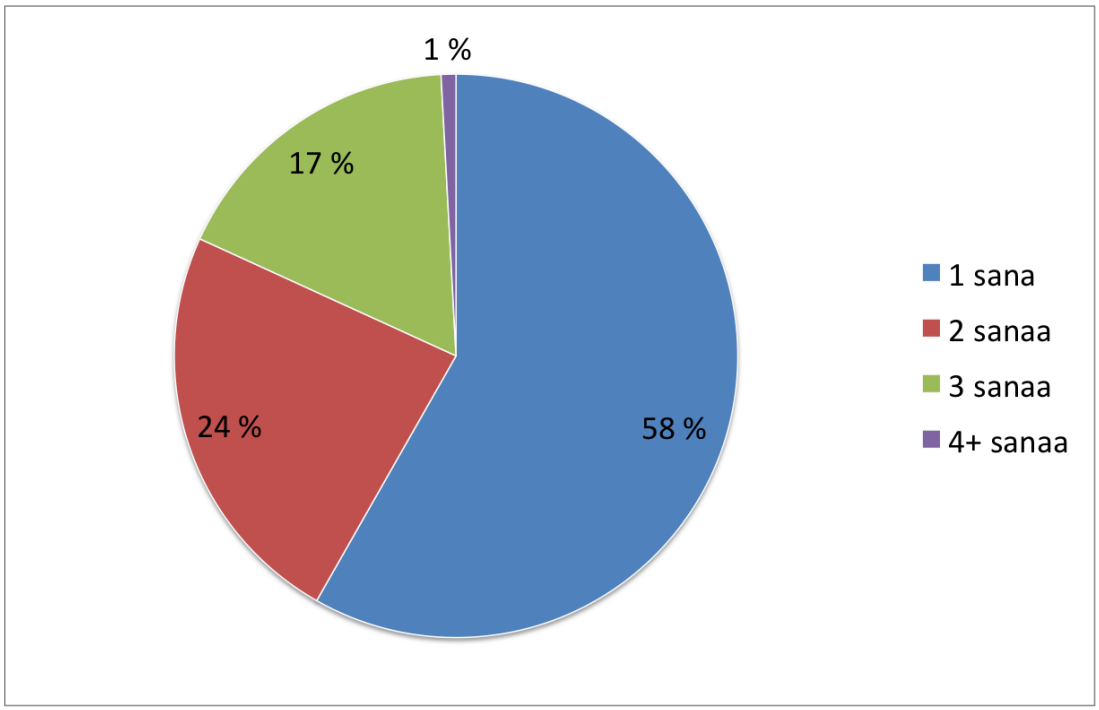
Testien suoritusten jälkeen tietokannassa oli 352 tekstihaastevastausta ja 301 monivalintahaastevastausta. Tekstihaastevastauksista 45,7% oli oikein, kun monivalintahaasteissa kaikki viisi kuvaa oikein tunnistaneita vastauksia oli vain 1,9% kaikista vastauksista. Tekstihaasteiden keskimääräinen suoritusaika oli 19 sekuntia keskihajonnalla 14,5. Monivalintahaasteissa keskimääräinen vastausaika oli huikea 71 sekuntia keskihajonnan ollessa myös erittäin korkea 448,3. Käyttäjä saattoi syöttää tekstihaasteessa vastauksena yhdestä kolmeen sanaa; suurimpaan osaan haasteista käyttäjät olivat vastanneet yhdellä sanalla. Sanamäärien jakauma on esitetty kuvassa 5.6.

Tuloksista voidaan tulkita seuraavat johtopäätökset:

- *Tekstiversio on käyttäjille huomattavasti helpompi kuin monivalintaversio.* Koska tekstihaasteiden läpäisyprosentti on selkeästi suurempi kuin moni-



Kuva 5.5 Testikäyttäjien englannin kielen taitojakauma.



Kuva 5.6 Tekstihaastevastausten sanamäärät.

Taulukko 5.1: Tekstihaasteen tulokset

Sanojen lkm	**Vastauksia**	**Joista oikein**
1 sana	205 kpl	43%
2 sanaa	83 kpl	51%
3 sanaa	61 kpl	52%
4+ sanaa	3 kpl	67%

valintahaasteen, voidaan tulkita tekstihaasteen olevan käyttäjille vaivattomampi.

- *Tekstiversio on käyttäjille huomattavasti vaivattomampi suorittaa kuin monivalintaversio.* Tekstihaasteessa käyttäjillä kului keskimäärin 19 sekuntia, kun monivalintaversion kanssa keskimääräinen aika oli minuutti ja 11 sekuntia. Epäformaalit haastattelut muutaman testikäyttäjän kanssa paljastivat ongelman: monivalintahaasteissa esiintyi sellaisia kuva- ja vaihtoehtoyhdistelmiä, joista käyttäjien mielestä mikään vaihtoehto ei sopinut kuvaan.

- *Tekstihaastevastauksissa sanojen määrällä on pieni merkitys vastauksen oikeellisuuteen.* Kuten taulukosta 5.1 ilmenee, vastaussanojen lukumäärällä on merkitys vastauksen oikeellisuuteen, mutta ero ei ole merkittävä. Neljä tai enemmän sanoja vastanneet vastaukset tulkittiin kuten kolmen sanan vastaukset, ottaen vastauksen alusta laskien kolme ensimmäistä sanaa huomioon.

- *Flick:n käyttö kuvalähteenä sellaisenaan on huono idea, koska kuvien tagit ovat vain yhden ihmisen mielipide.* Vaikka TAGTCHA:n käyttämät kuvat onkin haettu helposti visualisoitavissa olevalta sanalistalta olevien sanojen perusteella ja hyväksytyissä kuvissa on vähintään viisi tagia, ovat kuvaan liitety tagit kuitenkin vain yhden ihmisen mielestä asiaan sopivia. Esimerkiksi kuvaan, joka on haettu tagilla “car”, ei käyttäjä välttämättä vastaa sanalla “car” vaan vaikkapa auton värillä tai muulla kuvassa ehkä enemmän esillä olevalla sanalla. Flickristä poimitut kuvat eivät myöskään välttämättä kuvaa vain yhtä kohdetta.

Näin ollen TAGTCHA:n voidaan todeta olevan jatkokehitystä kaipaava voidak-

seen olla käyttökelpoinen CAPTCHA-toteutus. Käyttäjät eivät kykeneet ratkaisemaan TAGTCHA:a tarpeeksi suurella onnistumistodennäköisyydellä ja haasteiden ratkaisuun kului keskimäärin melko pitkä aika. Flickriä ei sellaisenaan voi pitää hyvänä haastelähteenä, sillä Flickr-kuvien tagit ovat aina vain yhden ihmisen tulkintoja omista kuvistaan, eivätkä välttämättä kuvaa itse kuvan sisältöä lainkaan.

5.8 Avoimet asiat ja jatkokehitystarpeet

Mikäli TAGTCHA:n tyylisen, sosiaalista kuvapalvelua käyttävän CAPTCHA-toteutuksen haluaisi saada käyttökelpoiseksi, tulisi suurimpana ongelmana ratkaista kuvien alkuperäisten tagien vahvistukseen liittyvä ongelma. Mikäli kuvat olisivat merkitty esimerkiksi usean kuvaan neutraalisti suhtautuvan ihmisen (eli ei esimerkiksi kuvassa esiintyvän lemmikkikissan tuntevia henkilöitä) toimesta, testikäyttäjilläkin olisi parempi mahdollisuus tulla samaan lopputulokseen. Toisaalta koska haasteiden tulee olla automaattisesti generoitavissa, tämä kuvien yhteisöllinen merkitseminen tulisi hoitaa CAPTCHA-haasteita ratkottaessa käyttäjien toimesta. Tätä reCAPTCHA-tyylistä lähestymistapaa kuva-CAPTCHA:jen toteuttamiseen ovat tutkineet ainakin Morrison ja muut [2009]. Ongelmaksi lähestymistavassa tulee kuitenkin paikallisen tietokannan tarve, johon tulisi tallentaa käyttäjien ehdotukset uusien kuvien sisällöstä.

TAGTCHA:n suunnittelussa käytettiin hyväksi luvussa 4 esiteltyä CAPTCHA-suunnittelukehystä. Vaikka TAGTCHA täyttääkin lähes kaikki kehyksen kohdat, voidaan TAGTCHA:n testituloksista silti todeta, ettei TAGTCHA ole riittävän tehokas ollakseen hyvä CAPTCHA. Suunnittelukehystä tuleekin käyttää vain apuvälineenä CAPTCHA:n suunnittelussa, eikä edes kehyksen täydellinen noudattaminen välttämättä takaa toimivaa CAPTCHA-toteutusta.

6 Tulokset

Olen selvittänyt CAPTCHA:jen nykytilanteen esittelemällä olemassa olevat CAPTCHA-ratkaisut sekä käymällä läpi CAPTCHA:jen murtamiseen liittyvät tutkimukset. Tämän tiedon valossa voidaankin sanoa nykyisin suosittujen tekstipohjaisten CAPTCHA:jen olevan poikkeuksetta murrettavissa, ja mikäli haasteiden monimutkaisuutta edelleen nostetaan, niistä tulee liian vaikeita ratkaista sekä tietokoneille että ihmisille. CAPTCHA-toteutuksesta ja sen haastavuudesta riippumatta se voidaan kuitenkin kiertää käyttämällä maksettua työvoimaa. CAPTCHA toimiikin lopulta vain ihmisen ja koneen erottavana, mutta ei Internet-palveluiden väärinkäyttöä estävänä tekijänä.

Kehittämäni CAPTCHA-kehyksen avulla CAPTCHA:a voidaan arvioida tehokkuus-, turvallisuus- ja käytettävyysnäkökulmasta. Kolmea CAPTCHA-tyyppiä, teksti-, kuva- ja audiopohjaisia, voidaankin lyhyesti arvioida seuraavasti: Tekstipohjaiset ovat heikoimmillaan turvallisuusnäkökulmasta katsoen; käytännössä kaikki tekstipohjaiset CAPTCHA:t voidaan murtaa koneellisesti. Kuvapohjaiset CAPTHCA:t ovat hankalia generoida ja tarkistaa automaattisesti, ja audiopohjaiset CAPTCHA:t taas ovat hitaita käyttää ja aiheuttavat paljon virheitä.

6.1 TAGTCHA-tulokset

CAPTCHA-kehyksen avulla suunnittelin uuden CAPTCHA-ratkaisun, TAGTCHA:n. TAGTCHA on kuvapohjainen CAPTCHA, joka testaa Internetissä toimivien sosiaalisten kuvapalveluiden soveltuvuutta haastelähteenä generoiden automaattisesti lisää haasteita ja niiden ratkaisuja hakemalla haastekuvat Flickr-kuvapalvelusta. TAGTCHA toimii joko pyytäen käyttäjää kirjoittamaan esitettyä haastekuvaa kuvaavia sanoja tai esittäen käyttäjälle viisi kuvaa, joihin jokaiseen liittyypudotusvalikko, jossa on kymmenen sanavaihtoehtoa. Järjestelmää testattiin 77 testikäyttäjällä. Lopputuloksena voidaan todeta tekstinsyöttöversion olevan huomattavasti parempi kuin monivalintaversion. Kuitenkin tekstinsyöttöversiossakin oikeita vastauksia oli vain 45,7% kaikista vastauksista, kun hyvässä CAPTCHA-toteutuksessa ihmisyyden onnistuneen todentamisen todennäköisyyden on oltava (lähes) 90 prosenttia. Näinollen TAGTCHA ei sellaisenaan käy uudeksi CAPTCHA-järjestelmäksi, mutta se esittää uuden tavan ammentaa kuvahaasteita käytännössä loputtomasta ja alati uusiutuvasta lähteestä, sosiaalisesta

kuvapalvelusta.

6.2 CAPTCHA:jen toteutusriippumattomat ongelmat

Koska CAPTCHA:jen murtamiseen käytetyt menetelmät kehitetään nimenomaan kyseinen murrettava CAPTCHA ja sen ominaisuudet huomioon ottaen, murtosovellukset voidaan usein rampauttaa helposti tekemällä vain pieniä muutoksia CAPTCHA:an [Motoyama *et al.*, 2010]. Lisäksi useissa verkkopalveluissa on otettu käyttöön IP-osoitteeseen perustuva rajoitin, joka estää tai hidastaa uusien CAPTCHA:jen ratkaisuyrityksiä, kun tietystä IP-osoitteesta on havaittu tulevan nopeasti useita virheellisiä yrityksiä. Näinollen palvelujen väärinkäyttäjille jää kaksi tehokasta ja taloudellista vaihtoehtoa: käyttäminen maksettua työvoimaa ja CAPTCHA:jen murto murtamisesta tietämättömien ihmisten avulla.

6.2.1 CAPTCHA:jen murto maksetulla ihmistyövoimalla

Kalliiden murtosovellusten vaihtoehdoksi markkinoille on ilmestynyt useita CAPTCHA:jen murtamista tarjoavia verkkopalveluita, jotka käyttävät CAPTCHA:jen selvittämiseen ihmistyövoimaa usein Kiinasta, Intiasta, Vietnamista tai muista maista, joissa työvoima on halpaa. Esimerkiksi BeatCaptchas.com [BeatCaptchas.com, 2011] myy CAPTCHA:n ratkaisupalvelua halvimmillaan hintaan $0,006 per CAPTCHA. Murtopalvelut pystyvät tarjoamaan palveluita hyvin lyhyellä vasteajalla, eräs palvelu lupaa jopa 5–15 sekunnin vasteajan vuorokaudenajasta riippumatta [Bursztein *et al.*, 2010]. Mikäli CAPTCHA:n ratkaisemalla saavutetun hyödyn, oli se sitten käyttäjätunnus verkkopalveluun tai mainoskommentti blogiin, arvo on suurempi kuin halvin CAPTCHA:n ratkaisuhinta, näiden palveluiden käyttö on hyökkääjälle kannattavaa.

6.2.2 CAPTCHA-pesuhyökkäys ja CAPTCHA-salakuljetus

CAPTCHA-pesuhyökkäys (engl. *laundry attack*) on saanut nimensä rahanpesusta. Hyökkäyksessä hyödynnetään tavallisia, pahaa aavistamattomia Internetin käyttäjiä ratkaisemaan CAPTCHA:ja hyökkääjän puolesta kuvitellen, että he ratkaisevat CAPTCHA:n saavuttaakseen jotakin muuta.

Pesuhyökkäys voi tapahtua esimerkiksi seuraavasti: hyökkääjällä on suosittu

Internet-sivusto, joka tarjoaa esimerkiksi aikuisviihdettä tai tiedostonjakopalveluita. Jotta käyttäjä voi tehdä haluamansa toiminnon, oli se sitten alastonkuvan katsominen tai tiedoston lataaminen, hänen täytyy selvittää CAPTCHA-haaste. Todellisuudessa kuitenkin haaste on peräisin hyökkäyksen kohdesivustolta, mutta näin hyökkääjä saa hyvin varmasti oikean ratkaisun CAPTCHA:an ja voi toistaa haasteiden ratkaisun vaatimista käyttäjältä esimerkiksi ennen jokaista tiedostonlatausta ilman, että kohdesivusto pystyy selvittämään kyseessä olevan palvelunsa väärinkäyttö [Athanasopoulos & Antonatos, 2006].

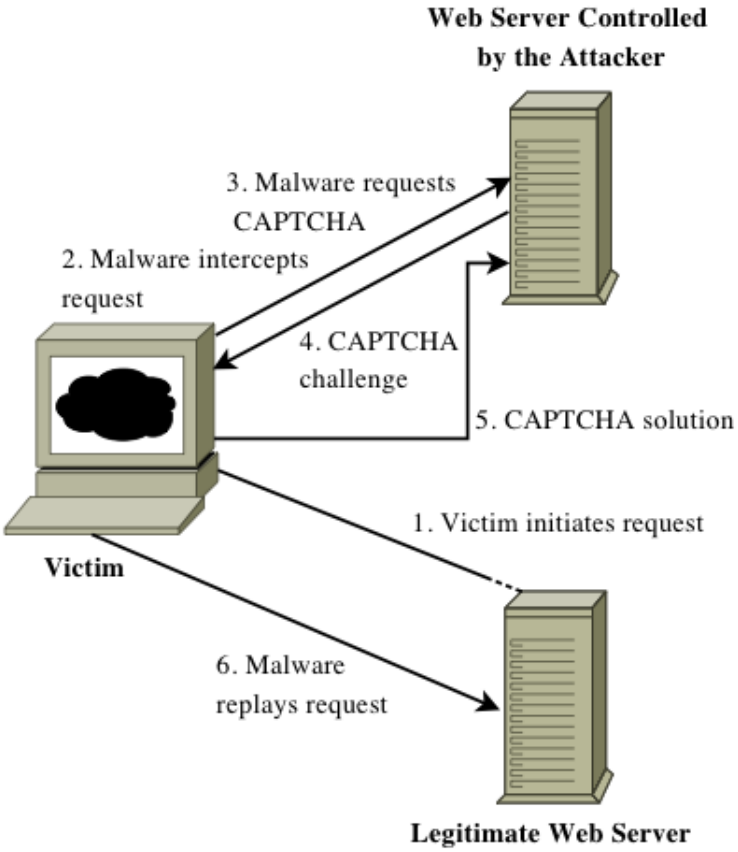
Koska käyttäjä voi kuitenkin helposti kyllästyä CAPTCHA:jen selvittämiseen ja vaihtaa palvelua, kehittivät Egele ja muut [2010] toisen samantyylisen hyökkäyksen, tällä kertaa hyökäten sellaisten verkkopalveluiden avulla, jotka ovat tietämättömiä väärinkäytöstä ja joista käyttäjä ei voi niin helposti vaihtaa pois. Lähestymistavasta käytetään nimeä CAPTCHA-salakuljetus (engl. *CAPTCHA smuggling*). Tässä hyökkäyksessä haittaohjelma asentuu käyttäjän tietokoneelle, tässä tapauksessa tarkemmin Firefox-selaimen lisäosana, ja monitoroi käyttäjän verkkoliikennettä. Hyökkäys on kuvattu kuvassa 6.1: käyttäjä suorittaa jonkin normaalin palvelupyynnön, kuten kirjautumisen Facebook-sivulle. Lisäosa havaitsee pyynnön, keskeyttää sen ja hakee hyökkääjän palvelimelta CAPTCHA:n. Käyttäjä luulee tämän CAPTCHA:n tulevan Facebookista ja olevan normaali osa palvelun käyttöä, joten hän ratkaisee sen. Todellisuudessa ratkaisu palautuu hyökkääjän palvelimelle, ja pahaa aavistamattoman käyttäjän palvelupyyntö suoritetaan loppuun normaalisti.

6.3 Ovatko CAPTCHA:t toimiva ratkaisu?

Jos kerran CAPTCHA:t pystytään joka tapauksessa kiertämään, viimeistään ihmistyövoimaa apuna käyttäen, herääkin kysymys: ovatko CAPTCHA:t toimiva ratkaisu? Motoyama ja muut [2010] toteavat vastauksen riippuvan siitä, mitä CAPTCHA:lla halutaan saada aikaiseksi. CAPTCHA:lla voidaan

- *Erottaa ihmiset ja tietokoneet toisistaan.* Vaikka osa CAPTCHA:sta pystytäänkin CAPTCHA-kohtaisesti ohittamaan koneellisesti, vielä ei ole kehitetty yleistä menetelmää, joka osaisi automaattisesti ohittaa kaikki CAPTCHA:t. Tässä mielessä CAPTCHA:t ovat menestyneet.

- *Estää kaikki automatisoidut sivunlataukset.* Koska palveluiden väärinkäyt-



Kuva 6.1 CAPTCHA-salakuljetushyökkäyksen toiminta [Egele et al., 2010].

täjät voivat ostaa CAPTCHA-ratkaisupalveluja halpaan hintaan ja näin saada CAPTCHA-suojatut palvelut käyttöönsä, CAPTCHA:t ovat epäonnistuneet.

- *Rajoittaa automatisoituja sivunlatauksia.* Motoyaman ja muiden mielestä on lyhytnäköistä ajatella CAPTCHA:ja ainoana puolustuskeinona hyökkääjiä vastaan. Niitä tulisi mieluummin ajatella tekijänä, joka nostaa hyökkääjien kustannuksia onnistua hyökkäyksessään, ja näin ollen saada hyökkääjät harkitsemaan hyökkäyksensä kannattavuutta.

CAPTCHA:sta on muodostunut “one size fits all”-ratkaisu, työkalu jonka kuvitellaan sopivan kaikenlaisiin verkkopalveluihin. Voidaan kuitenkin kysyä, voisiko olla muitakin keinoja ehkäistä verkkopalveluiden väärinkäyttö, kuin vain CAPTCHA:t?

Vaihtoehtoja on useita. Esimerkiksi WordPressiä [WordPress Inc., 2011] käyttävissä blogeissa on usein kommenttisuotimena roskakommentteja vastaan lisäosa nimeltä Akismet [Akismet Inc., 2011]. Akismet on keskitetty verkkopalvelu, joka suodattaa mainoksia ja muuta vältettävää sisältöä sisältävät käyttäjien lisäämät kommentit pois. Käyttäjä kirjoittaa blogiin kommentin, WordPress-blogi lähettää kommentin sisällön automaattisesti Akismetin palvelimelle, joka tulkitsee viestin sisällön, ja mikäli viesti on sisällöltään asiallinen, se palauttaa viestin WordPress-blogiin, joka taas voi jatkossa näyttää uuden kommentin lukijoillensa. Ratkaisussa ei tarvita CAPTCHA:a, mutta silti pystytään suodattamaan käytännössä kaikki haitalliset viestit. Samaa periaatetta käyttävät myös useat keskustelupalstasovellukset ja sähköpostipalvelimet [May, 2005].

Toinen tapa on tutkia käyttäjien palvelunkäyttöä. Verkkopalvelun käyttäjätietoja tarkasti tulkiten voidaan usein havaita väärinkäyttäjien tekevän asioita eri tavalla kuin normaalien käyttäjien. Esimerkiksi sähköpostipalvelussa roskapostittajat lähettävät selkeästi enemmän sähköpostia kuin tavalliset käyttäjät. Näin palvelu voisi joko sulkea palvelusta pois sellaiset käyttäjätunnukset, jotka on tunnistettu roskapostittajiksi, tai ennaltaehkäistä roskapostittajien kiinnostuksen palveluun jo etukäteen asettamalla esimerkiksi sadan lähtevän sähköpostin rajoituksen per päivä per käyttäjätili tai tietyn minimiajan asettaminen kahden viestin lähetykselle. Tällöin palvelusta ei ole käytännössä hyötyä massapostittajille ollenkaan (etenkin jos tilin luonnissa esitetyn CAPTCHA:n murtamisesta joutuisi maksamaan).

Nämä ratkaisut voivat parhaimmillaan olla erittäin toimivia, ja samalla kuitenkin lähes tai täysin näkymättömissä tavalliselle käyttäjälle. Tällainen ratkaisu on kuitenkin erittäin syvälle sidottu siihen verkkopalveluun, jossa se on käytössä. Heuristiikka tarvitsee suunnitella nimenomaan kyseistä verkkopalvelua varten, eikä se ole CAPTCHA:n tyylinen yleisratkaisu. Toisaalta mikäli järjestelmä pyrkisi erottelemaan sitä väärinkäyttävät käyttäjät oikeista käyttäjistä pelkkien heuristiikkojen perusteella, voisi sopivassa tapauksessa vahinko olla jo tapahtunut, sillä jotta heuristiikkoja voidaan käyttää, täytyy olla dataa mitä arvioida.

Ihmisyyden tunnistamiseen voidaan käyttää myös jotakin viranomaisteitse vahvistettua henkilöllisyystodistusta, kuten esimerkiksi Suomessa pankkien tarjoamat Internet-pankkikäyttäjätunnukset ja niitä hyödyntävä Tupas-varmennepalvelu [Finanssialan Keskusliitto, 2011]. Tällöin voidaan suorittaa käyttäjän niin sanottu vahva tunnistaminen, jonka tuloksena palveluntarjoaja saa

tietoonsa käyttäjän henkilötunnuksen ja nimen, jotka ovat tärkeää ja välttämätöntä tietoa esimerkiksi luoton myöntämiseen keskittyneiden palveluiden käytössä. Vahvat tunnistamismenetelmät voivat kuitenkin olla palveluntarjoajalle maksullisia, ja ne poistavat Internetissä usein vallitsevan anonymiteetin palveluiden käyttäjiltä. Keskustelu siitä, pitäisikö Internetin käyttäjät olla aina yksilöitävissä, jätetään tämän tutkielman ulkopuolelle.

7 Lopuksi

Tässä tutkielmassa käsiteltiin nykyisten CAPTCHA-ratkaisuiden ongelmakohtia. Useimmin käytössä olevat tekstipohjaiset CAPTCHA:t ovat tiensä päässä; nykyaikaisin tekstintunnistusohjelmistoin niiden kiertäminen on mahdollista. Koska uusia ratkaisuja tarvitaan, tutkielmassa läpikäytyjen nykyisten ratkaisuiden pohjalta kehitettiin CAPTCHA-kehys uusien CAPTCHA:jen suunnittelua varten. Tämän kehyksen perusteella suunnittelin TAGTCHA:n, joka testaa sosiaalisten kuvapalveluiden käyttökelpoisuutta kuvapohjaisen CAPTCHA:n haasteja ratkaisulähteenä. TAGTCHA:n tekstinsyöttöversio osoittautui käyttäjätesteissä huomattavasti paremmaksi kuin monivalintaversio, mutta tekstinsyöttöversiokin onnistui tunnistamaan ihmisen onnistuneesti ainoastaan noin joka toisessa tapauksessa.

Tutkimuksessani selvisi, että suurin TAGTCHA:n epäonnistumiseen johtava ongelma on haastekuvien ratkaisuiden subjektiivisuus. Haastekuvat ja niihin liitetyt tagit ovat vain yhden ihmisen tulkinta itse palveluun lataamastaan kuvasta. Mikäli Internetin sosiaalisia kuvapalveluita haluttaisiin käyttää CAPTCHA-haasteiden pohjana, tulevaisuudessa täytyisi kehittää jokin sellainen toimintamalli, jossa kuvien merkitys luotaisiin jotenkin useamman ihmisen yhteistyönä.

CAPTCHA on nimensä mukaisesti automatisoitu testi, jolla testataan käyttäjän ihmisyys. Koska testi on usein kertaluonteinen ja suoritetaan Internet-palveluun kirjauduttaessa tai rekisteröidyttäessä, se ei välttämättä takaa sitä, että palvelua lopulta käyttävä käyttäjä olisi ihminen. Internetissä toimii erityisesti halpatyövoimaa käyttäviä CAPTCHA:jen kiertoon keskittyneitä palveluita, jotka tarjoavat CAPTCHA-murtoa ihmisvoimin erittäin halpaan hintaan. Näin CAPTCHA:t toimivat oikein erotellessaan ihmiset koneista, mutta alkuperäinen ajatus palveluiden väärinkäytön estämisestä on virheellinen. Jos palveluiden väärinkäyttö halutaan estää, tulisi ennemmin kehittää jokin keino tarkkailla käyttäjän toimia palvelussa. Tämän tarkkailun perusteella tavallisuudesta poikkeavat käyttäjät voidaan tarpeen tullen erottaa palvelusta. Tällaiset säännöt tulee kuitenkin kehittää palvelukohtaisesti, eivätkä ne toimi CAPTCHA:jen tavoin kontekstiriippumattomasti.

A SQL-taulujen luontilauseet

cached_image

```
CREATE TABLE ‘cached_image‘ (
  ‘id‘ int(11) NOT NULL AUTO_INCREMENT,
  ‘filename‘ varchar(512) NOT NULL,
  ‘original_tag‘ varchar(255) NOT NULL,
  PRIMARY KEY (‘id‘)
);
```

tag

```
CREATE TABLE ‘tag‘ (
  ‘id‘ int(11) NOT NULL AUTO_INCREMENT,
  ‘image_id‘ int(11) NOT NULL,
  ‘tag‘ varchar(255) NOT NULL,
  PRIMARY KEY (‘id‘)
);
```

user

```
CREATE TABLE ‘user‘ (
  ‘id‘ int(11) NOT NULL AUTO_INCREMENT,
  ‘age‘ varchar(255) NOT NULL,
  ‘english‘ varchar(255) NOT NULL,
  ‘timestamp‘ timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,
  ‘ip‘ varchar(255) NOT NULL,
  PRIMARY KEY (‘id‘)
);
```

result

```
CREATE TABLE ‘result‘ (
  ‘id‘ int(11) NOT NULL AUTO_INCREMENT,
  ‘user_id‘ int(11) NOT NULL,
```

```
  ‘timestamp‘ timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,
  ‘time_taken‘ int(11) NOT NULL,
  ‘type‘ varchar(64) NOT NULL,
  PRIMARY KEY (‘id‘)
);
```

result_line

```
CREATE TABLE ‘result_line‘ (
  ‘result_id‘ int(11) NOT NULL,
  ‘image_id‘ int(11) NOT NULL,
  ‘answer‘ varchar(1024) NOT NULL,
  ‘result‘ tinyint(4) NOT NULL
);
```

Viiteluettelo

[Al-Sudani *et al.*, 2010] Wesam Al-Sudani, Amit Gill, Chen Li, Jidong Wang, & Fei Liu. Protection through multimedia CAPTCHAs. In *Proceedings of the 8th International Conference on Advances in Mobile Computing and Multimedia*, (MoMM '10), pages 63–68, New York, NY, USA, 2010. ACM.

[Athanasopoulos & Antonatos, 2006] Elias Athanasopoulos & Spiros Antonatos. Enhanced CAPTCHAs: Using Animation to Tell Humans and Computers Apart. *Lecture Notes in Computer Science*, 4237:97–108, 2006.

[Baird & Bentley, 2005] Henry S. Baird & Jon L. Bentley. Implicit CAPTCHAs. In *Proc., IS & T/SPIE Document Recognition & Retrieval XII Conf.*, volume XII, pages 191–196, January 2005.

[Baird *et al.*, 2005] Henry S. Baird, Michael A. Moll, & Sui-Yu Wang. A Highly Legible CAPTCHA That Resists Segmentation Attacks. *Lecture Notes in Computer Science*, 3517:27–41, 2005.

[Banday & Shah, 2011] M. Tariq Banday & Nisar A. Shah. Challenges of CAPTCHA in the accessibility of Indian regional websites. In *Proceedings of the Fourth Annual ACM Bangalore Conference*, (COMPUTE '11), pages 31:1–31:4, New York, NY, USA, 2011. ACM.

[BareBones, 2009] BareBones. BareBones MVC, 2009. Available at: `http://code.google.com/p/barebonesmvc-php/`. Last checked: 28.9.2011.

[BeatCaptchas.com, 2011] BeatCaptchas.com. BeatCaptchas.com, 2011. Available at: `http://www.beatcaptchas.com/prices.html`. Last checked: 14.9.2011.

[Bigham & Cavender, 2009] Jeffrey P. Bigham & Anna C. Cavender. Evaluating existing audio CAPTCHAs and an interface optimized for non-visual use. In *CHI '09: Proceedings of the 27th International Conference on Human Factors in Computing Systems*, pages 1829–1838, New York, NY, USA, 2009. ACM.

[Bursztein *et al.*, 2010] Elie Bursztein, Steven Bethard, Celine Fabry, John C. Mitchell, & Dan Jurafsky. How Good Are Humans at Solving CAPTCHAs? A Large Scale Evaluation. In *Proceedings of the 2010 IEEE Symposium on*

Security and Privacy, (SP '10), pages 399–413, Washington, DC, USA, 2010. IEEE Computer Society.

[BusinessWeek, 2006] BusinessWeek. Click fraud, 2006. Available at: `http://www.businessweek.com/magazine/content/06_40/b4003001.htm`. Last checked: 5.9.2011.

[Captcha.net, 2011] Captcha.net. captcha.net, 2011. Available at: `http://www.captcha.net`. Last checked: 1.9.2011.

[Chandavale & Sapkal, 2010] A. A. Chandavale & A. M. Sapkal. Algorithm for Secured Online Authentication Using CAPTCHA. In *Proceedings of the 2010 3rd International Conference on Emerging Trends in Engineering and Technology*, (ICETET '10), pages 292–297, Washington, DC, USA, 2010. IEEE Computer Society.

[Chellapilla & Simard, 2004] Kumar Chellapilla & Patrice Simard. Using Machine Learning to Break Visual Human Interaction Proofs (HIPs). In *Advances in Neural Information Processing Systems 17, Neural Information Processing Systems (NIPS'2004)*. MIT Press, 2004.

[Chellapilla *et al.*, 2005a] Kumar Chellapilla, Kevin Larson, Patrice Simard, & Mary Czerwinski. Computers beat Humans at Single Character Recognition in Reading based Human Interaction Proofs (HIPs). In *Second Conference on Email and Anti-Spam (CEAS'2005)*, 2005.

[Chellapilla *et al.*, 2005b] Kumar Chellapilla, Kevin Larson, Patrice Simard, & Mary Czerwinski. Designing human friendly human interaction proofs (HIPs). In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, (CHI '05), pages 711–720, New York, NY, USA, 2005. ACM.

[Chellapilla *et al.*, 2005c] Kumar Chellapilla, Kevin Larson, Patrice Y. Simard, & Mary Czerwinski. Building Segmentation Based Human-Friendly Human Interaction Proofs (HIPs). *Lecture Notes in Computer Science*, 3517:1–26, 2005.

[Chew & Baird, 2003] Monica Chew & Henry S. Baird. BaffleText: a human interactive proof. In *DRR*, pages 305–316, 2003.

[Chew & Tygar, 2004a] Monica Chew & J. D. Tygar. Image Recognition CAPTCHAs. Technical Report UCB/CSD-04-1333, EECS Department, University of California, Berkeley, Jun 2004.

[Chew & Tygar, 2004b] Monica Chew & J.D. Tygar. Image Recognition CAPTCHAs. In *Proceedings of the 7th International Information Security Conference (ISC 2004)*, pages 268–279. Springer, September 2004.

[Chew & Tygar, 2005] Monica Chew & J.D. Tygar. Collaborative filtering CAPTCHAs. In Henry S. Baird & D. Lopresti, editors, *Human Interactive Proofs: Second International Workshop (HIP 2005)*, pages 66–81. Springer, May 2005.

[Chow *et al.*, 2008] Richard Chow, Philippe Golle, Markus Jakobsson, Lusha Wang, & XiaoFeng Wang. Making CAPTCHAs clickable. In *Proceedings of the 9th Workshop on Mobile Computing Systems and Applications*, (HotMobile '08), pages 91–94, New York, NY, USA, 2008. ACM.

[Coates *et al.*, 2001] Allison L. Coates, Henry S. Baird, & Richard J. Fateman. Pessimal Print: A Reverse Turing Test. In *IAPR 6th Int'l Conf. on Document Analysis and Recognition*, pages 1154–1158, 2001.

[Coulter, 2011] Dan Coulter. phpFlickr, 2011. Available at: `http://phpflickr.com/`. Last checked: 28.9.2011.

[Cui *et al.*, 2009] Jing-Song Cui, Jing-Ting Mei, Xia Wang, Da Zhang, & Wu-Zhou Zhang. A CAPTCHA Implementation Based on 3D Animation. In *Proceedings of the 2009 International Conference on Multimedia Information Networking and Security - Volume 02*, (MINES '09), pages 179–182, Washington, DC, USA, 2009. IEEE Computer Society.

[Cui *et al.*, 2010] Jing-Song Cui, Jing-Ting Mei, Wu-Zhou Zhang, Xia Wang, & Da Zhang. A CAPTCHA Implementation Based on Moving Objects Recognition Problem. In *Proceedings of the 2010 International Conference on E-Business and E-Government*, (ICEE '10), pages 1277–1280, Washington, DC, USA, 2010. IEEE Computer Society.

[Datta *et al.*, 2005] Ritendra Datta, Jia Li, & James Z. Wang. IMAGINATION: a robust image-based CAPTCHA generation system. In *Proceedings of the 13th*

Annual ACM International Conference on Multimedia, (MULTIMEDIA '05), pages 331–334, New York, NY, USA, 2005. ACM.

[El Ahmad *et al.*, 2010] Ahmad Salah El Ahmad, Jeff Yan, & Lindsay Marshall. The robustness of a new CAPTCHA. In *Proceedings of the Third European Workshop on System Security*, (EUROSEC '10), pages 36–41, New York, NY, USA, 2010. ACM.

[Elson *et al.*, 2007] Jeremy Elson, John R. Douceur, Jon Howell, & Jared Saul. Asirra: a CAPTCHA that exploits interest-aligned manual image categorization. In *Proceedings of the 14th ACM Conference on Computer and Communications Security*, (CCS '07), pages 366–374, New York, NY, USA, 2007. ACM.

[Faymonville *et al.*, 2009] Peter Faymonville, Kai Wang, John Miller, & Serge Belongie. CAPTCHA-based image labeling on the Soylent Grid. In *Proceedings of the ACM SIGKDD Workshop on Human Computation*, (HCOMP '09), pages 46–49, New York, NY, USA, 2009. ACM.

[Fidas *et al.*, 2011] Christos A. Fidas, Artemios G. Voyiatzis, & Nikolaos M. Avouris. On the necessity of user-friendly CAPTCHA. In *Proceedings of the 2011 Annual Conference on Human Factors in Computing Systems*, (CHI '11), pages 2623–2626, New York, NY, USA, 2011. ACM.

[Finanssialan Keskusliitto, 2011] Finanssialan Keskusliitto. Tupas, 2011. Saatavissa `http://www.fkl.fi/teemasivut/sahkoinen_asiointi/tupas/Sivut/default.aspx`. Viitattu 15.9.2011.

[Flickr, 2011a] Flickr. 6,000,000,000, 2011. Available at: `http://blog.flickr.net/en/2011/08/04/6000000000/`. Last checked: 25.9.2011.

[Flickr, 2011b] Flickr. Flickr, 2011. Available at: `http://www.flickr.com`. Last checked: 4.10.2011.

[Flickr, 2011c] Flickr. Flickr Services: The App Garden, 2011. Available at: `http://www.flickr.com/services/api/`. Last checked: 28.9.2011.

[Gao *et al.*, 2010a] Haichang Gao, Honggang Liu, Dan Yao, Xiyang Liu, & Uwe Aickelin. An Audio CAPTCHA to Distinguish Humans from Computers. In

Proceedings of the 2010 Third International Symposium on Electronic Commerce and Security, (ISECS '10), pages 265–269, Washington, DC, USA, 2010. IEEE Computer Society.

[Gao *et al.*, 2010b] Haichang Gao, Dan Yao, Honggang Liu, Xiyang Liu, & Liming Wang. A Novel Image Based CAPTCHA Using Jigsaw Puzzle. In *Proceedings of the 2010 13th IEEE International Conference on Computational Science and Engineering*, (CSE '10), pages 351–356, Washington, DC, USA, 2010. IEEE Computer Society.

[Golle, 2008] Philippe Golle. Machine learning attacks against the Asirra CAPTCHA. In *Proceedings of the 15th ACM Conference on Computer and Communications Security*, (CCS '08), pages 535–542, New York, NY, USA, 2008. ACM.

[Gossweiler *et al.*, 2009] Rich Gossweiler, Maryam Kamvar, & Shumeet Baluja. What's up CAPTCHA?: a CAPTCHA based on image orientation. In *Proceedings of the 18th International Conference on World Wide Web*, (WWW '09), pages 841–850, New York, NY, USA, 2009. ACM.

[Higgins, 2010] Kelly Jackson Higgins. reCAPTCHA Broken, 2010. Available at: `http://www.darkreading.com/authentication/167901072/security/vulnerabilities/226700514/index.html`. Last checked: 6.9.2011.

[Holman *et al.*, 2007] Jonathan Holman, Jonathan Lazar, Jinjuan Heidi Feng, & John D'Arcy. Developing usable CAPTCHAs for blind users. In *Proceedings of the 9th International ACM SIGACCESS Conference on Computers and Accessibility*, (Assets '07), pages 245–246, New York, NY, USA, 2007. ACM.

[Kluever & Zanibbi, 2009] Kurt Alfred Kluever & Richard Zanibbi. Balancing usability and security in a video CAPTCHA. In *Proceedings of the 5th Symposium on Usable Privacy and Security*, (SOUPS '09), pages 1–11, New York, NY, USA, 2009. ACM.

[Kulkarni, 2008] Chinmay Eishan Kulkarni. Assocaptcha: designing human-friendly secure CAPTCHAs using word associations. In *CHI '08 Extended Abstracts on Human Factors in Computing Systems*, (CHI '08), pages 3705–3710, New York, NY, USA, 2008. ACM.

[Li & Wang, 2008] Jia Li & James Z. Wang. Real-Time Computerized Annotation of Pictures. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(6):985–1002, 2008.

[Lillibridge *et al.*, 2001] M. D. Lillibridge, M. Abadi, K. Bharat, & A. Z. Broder. Method for selectively restricting access to computer systems. U.S. Patent No. 6 195 698, February 2001.

[May, 2005] Matt May. Inaccessibility of CAPTCHA, 2005. Available at: `http://www.w3.org/TR/turingtest/`. Last checked: 14.9.2011.

[Mori & Malik, 2003] G. Mori & J. Malik. Recognizing Objects in Adversarial Clutter: Breaking a Visual CAPTCHA. In *CVPR*, volume 1, pages 134–141, 2003.

[Morrison *et al.*, 2009] Donn Morrison, Stéphane Marchand-Maillet, & Éric Bruno. TagCaptcha: annotating images with CAPTCHAs. In *Proceedings of the ACM SIGKDD Workshop on Human Computation*, (HCOMP '09), pages 44–45, New York, NY, USA, 2009. ACM.

[Motoyama *et al.*, 2010] Marti Motoyama, Kirill Levchenko, Chris Kanich, Damon McCoy, Geoffrey M. Voelker, & Stefan Savage. Re: CAPTCHAs: understanding CAPTCHA-solving services in an economic context. In *Proceedings of the 19th USENIX Conference on Security*, (USENIX Security '10), pages 28–28, Berkeley, CA, USA, 2010. USENIX Association.

[Naor, 1996] Moni Naor. Verification of a human in the loop or Identification via the Turing test. Available at: `http://www.wisdom.weizmann.ac.il/~naor/PAPERS/human.ps.gz`. Last checked: 17.10.2011., September 1996.

[PetFinder, 2011] PetFinder. PetFinder, 2011. Available at: `http://www.petfinder.com`. Last checked: 12.9.2011.

[PHP, 2011] PHP. PHP Manual: similar_text, 2011. Available at: `http://php.net/manual/en/function.similar-text.php`. Last checked: 26.9.2011.

[Rui & Liu, 2003] Yong Rui & Zicheg Liu. ARTiFACIAL: automated reverse turing test using FACIAL features. In *Proceedings of the Eleventh ACM Inter-*

national Conference on Multimedia, (MULTIMEDIA '03), pages 295–298, New York, NY, USA, 2003. ACM.

[Rusu & Govindaraju, 2005] Amalia Rusu & Venu Govindaraju. Visual CAPTCHA with Handwritten Image Analysis. *Lecture Notes in Computer Science*, 3517:42–52, 2005.

[Schluessler *et al.*, 2007] Travis Schluessler, Stephen Goglin, & Erik Johnson. Is a bot at the controls? detecting input data attacks. In *Proceedings of the 6th ACM SIGCOMM Workshop on Network and System Support for Games*, (NetGames '07), pages 1–6, New York, NY, USA, 2007. ACM.

[Tam *et al.*, 2008] J. Tam, J. Simsa, S. Hyde, & L. Von Ahn. Breaking Audio CAPTCHAs. *Advances in Neural Information Processing Systems*, 21, 2008.

[Turing, 1950] Alan Turing. Computing Machinery and Intelligence. *Mind*, 49:433–460, 1950.

[von Ahn *et al.*, 2003] Luis von Ahn, Manuel Blum, Nicholas J. Hopper, & John Langford. CAPTCHA: Using Hard AI Problems for Security. *Lecture Notes in Computer Science*, 2656:646, 2003.

[von Ahn *et al.*, 2008] Luis von Ahn, Benjamin Maurer, Colin McMillen, David Abraham, & Manuel Blum. reCAPTCHA: Human-Based Character Recognition via Web Security Measures. *Science*, 321, 2008.

[Ximenes *et al.*, 2006] Pablo Ximenes, André dos Santos, Marcial Fernandez, & Juaquim Celestino Jr. A CAPTCHA in the Text Domain. *Lecture Notes in Computer Science*, 4277:605–615, 2006.

[Yamamoto *et al.*, 2010a] Takumi Yamamoto, Tokuichiro Suzuki, & Masakatsu Nishigaki. A Proposal of Four-Panel Cartoon CAPTCHA: The Concept. In *Proceedings of the 2010 13th International Conference on Network-Based Information Systems*, (NBIS '10), pages 575–578, Washington, DC, USA, 2010. IEEE Computer Society.

[Yamamoto *et al.*, 2010b] Takumi Yamamoto, J. D. Tygar, & Masakatsu Nishigaki. CAPTCHA Using Strangeness in Machine Translation. In *Proceedings of the*

2010 24th IEEE International Conference on Advanced Information Networking and Applications, (AINA '10), pages 430–437, Washington, DC, USA, 2010. IEEE Computer Society.

[Yan & El Ahmad, 2008a] Jeff Yan & Ahmad Salah El Ahmad. A low-cost attack on a Microsoft CAPTCHA. In *Proceedings of the 15th ACM Conference on Computer and Communications Security*, (CCS '08), pages 543–554, New York, NY, USA, 2008. ACM.

[Yan & El Ahmad, 2008b] Jeff Yan & Ahmad Salah El Ahmad. Usability of CAPTCHAs or usability issues in CAPTCHA design. In *Proceedings of the 4th Symposium on Usable Privacy and Security*, (SOUPS '08), pages 44–52, New York, NY, USA, 2008. ACM.

[Yan & El Ahmad, 2010] Jeff Yan & Ahmad Salah El Ahmad. Colour, usability and security: a case study. Technical Report CS-TR-1203, Newcastle University, May 2010.