Pós-Graduação em Ciência da Computação

Rodrigo Alves Costa

"Uma Abordagem para Combate da Fraude de Clique Baseada em CAPTCHAs Clicáveis"

Tese de Doutorado

Universidade Federal de Pernambuco
posgraduacao@cin.ufpe.br
www.cin.ufpe.br/~posgraduacao

RECIFE
2016

RODRIGO ALVES COSTA

“UMA ABORDAGEM PARA COMBATE DA FRAUDE DE CLIQUE BASEADA EM CAPTCHAS CLICÁVEIS "

TESE DE DOUTORADO APRESENTADA À PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO DO CENTRO DE INFORMÁTICA DA UNIVERSIDADE FEDERAL DE PERNAMBUCO COMO REQUISITO PARCIAL PARA OBTENÇÃO DO GRAU DE DOUTOR EM CIÊNCIA DA COMPUTAÇÃO.

ORIENTADOR: Ruy José Guerra Barreto de Queiroz

RECIFE
2016

Catalogação na fonte
Bibliotecário Jefferson Luiz Alves Nazareno CRB 4-1758

C837a Costa, Rodrigo Alves.
Uma abordagem para combate da fraude de clique baseada em CAPTCHAs clicáveis / Rodrigo Alves Costa – 2016.
164f.: fig., tab.

Orientador: Ruy José Guerra Barreto de Queiroz.
Tese (Doutorado) – Universidade Federal de Pernambuco. CIn. Ciência da Computação, Recife, 2016.
Inclui referências.

1. Anúncios pela internet 2. Fraude clique. 3. CAPTCHAs clicáveis I. Queiroz, Ruy José Guerra Barreto de. (Orientador). II. Titulo.

005.8 CDD (22. ed.) UFPE-MEI 2016-165

Rodrigo Alves Costa

Uma Abordagem para Combate da Fraude de Clique Baseada em CAPTCHAs Clicáveis

Tese de Doutorado apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de Pernambuco, como requisito parcial para a obtenção do título de Doutora em Ciência da Computação

Aprovado em: 04/03/2016.

_____

Orientador: Prof. Ruy José Guerra Barretto de Queiroz

BANCA EXAMINADORA

_____

Prof. Dr. Vinicius Cardoso Garcia
Centro de Informática / UFPE

_____

Prof. Dr. Kelvin Lopes Dias
Centro de Informática / UFPE

_____

Prof. Dr. Wellington Candeia de Araújo
Centro de Ciências Exatas e Sociais Aplicadas / UEPB

_____

Prof. Danilo Cesar Maganhoto Doneda
Centro de Estudos e Pesquisas no Ensino do Direito / UERJ

_____

Prof. Dr. Carlos Alberto Kamienski
Centro de Matemática, Computação e Cognição / UFABC

Dedico este trabalho ao Espírito Santo, porque Ele nunca me deixa só, nunca me abandona, nunca desampara e nunca vai embora.

Agradecimentos

O que parecia impossível, Deus tornou possível e por isto estou feliz e grato. Não tenho palavras para definir o que é terminar esse trabalho. Só Deus seria capaz de tornar isso possível, de verdade. Assim, dedico esse trabalho única e exclusivamente ao Deus do impossível. Só Ele sabe o que passei nos anos que antecederam essa entrega e como para mim teria sido natural, aceitável e justificável desistir.

Por isso, acima de tudo, agradeço a Ele, o Senhor Deus de toda a terra, o Soberano, YHWH, mas que é tão maravilhoso que amou o mundo de tal maneira que deu Seu Filho unigênito, para que todo aquele que crer neste Filho não pereça, mas tenha a vida eterna. Mesmo assim, me fez companhia, logo a mim, que sou tão falho e insignificante e não me deixou desistir, e me permitiu defender esta tese no último dia possível.

Agradeço aos meus pais, Hianto (*in memoriam*) e Núbia, por terem tantas vezes abdicado de suas vidas para que minhas irmãs e eu pudéssemos ter a melhor educação possível. Eles sempre foram tão amorosos. Tenho muitas saudades do meu pai e certeza que a recompensa da minha mãe está no Senhor.

Agradeço de maneira muito especial ao professor Ruy Guerra, por sua disponibilidade e receptividade constante. Tenho consciência que as minhas falhas como orientando são muito mais um reflexo delas mesmas do que de sua imensa capacidade como professor, tutor e pesquisador. Espero que este seja apenas o início de nossa parceria.

Agradeço a todos os professores do Centro de Informática da UFPE, pela dedicação e estímulo à pesquisa, com quem já convivo até essa data há dezesseis anos. O Centro de Informática é definitivamente parte da minha história.

A todos os que tinham certeza que eu terminaria. Àqueles que nunca duvidaram. Que sabiam que seria difícil, mas não me subestimaram. Eu consegui. Por fim, um agradecimento particular para todos aqueles que me incentivaram com uma palavra de ânimo sincera, ou com algo que colaborou para a realização deste trabalho. Muitas pessoas me estimularam de diversas formas ao longo dos anos. Posso dizer que pequenas palavras de estímulo acumuladas definitivamente fizeram toda a diferença no fim das contas.

Resumo

No atual clima desafiador da economia global, anunciantes e suas agências de publicidade encontram-se em busca contínua por oportunidades de negócios que os levem a aumentar significativamente a exposição de suas marcas e de seus produtos a custos cada vez menores. Recursos de marketing têm sido redirecionados a campanhas na Internet do tipo Pagamento Por Clique (PPC), com remunerações associadas à realização de alguma ação potencialmente lucrativa para o anunciante, como o clique em um anúncio por parte de um usuário. Nessa estrutura, um fraudador pode aparecer como alguém que busca aumentar os lucros de um publicador de anúncios ao clicar nos anúncios exibidos sem ter a real intenção de comprar os produtos, ou ainda como uma empresa concorrente que clica nos anúncios de determinada organização para gerar custos indevidos e esse processo é exponencialmente mais oneroso para as vítimas da fraude se for realizado de maneira automatizada, por meio de *scripts* e *bots*. Combater a fraude de clique é, portanto, de fundamental importância para a continuidade do mercado de anúncios *online*, já que anunciantes mais conservadores acabam se afastando desta forma de fazer negócios por sua aversão a riscos e, no mundo ideal, no qual a fraude de clique ou inexiste ou possui influência mínima no faturamento das companhias, é necessário desenvolver soluções computacionais que suportem o crescimento do mesmo. Nesse sentido, esta pesquisa tem por objetivo produzir contribuições acadêmicas e desenvolver soluções com foco no negócio de PPC, incluindo a disponibilização de um material inédito em língua portuguesa sobre a sistemática, histórico e evolução dos anúncios *online*, bem como suas fraudes, casos legais e formas de detecção, e a elaboração de uma abordagem de combate que alia formas clássicas de resposta à fraude de clique com uma abordagem inovadora para a sua prevenção, baseada em CAPTCHA clicáveis, algo que não foi encontrado na literatura, nem no mercado. Apresentamos o Google NoCAPTCHA reCAPTCHA como uma forma de prevenir a automatização da fraude de clique, o que caracteriza uma mudança de paradigma no combate à fraude de clique, que passa a ser de prevenção proativa, ao contrário das abordagens clássicas, que buscavam identificar a fraude por meio de heurísticas de análise de histórico de cliques após as mesmas já terem ocorrido. Com efeito, este trabalho contribui na aquisição de conhecimento no domínio desse negócio, através de um estudo extensivo de anúncios *online*, apoiando a identificação de requisitos de sistemas e do negócio, e sua disposição de mercado. Como um benefício, tal estudo viabilizou a proposição de uma abordagem de segurança para as redes de anúncios, inspirada em casos legais e no conjunto de métodos utilizados para fraudes de clique.

Palavras-chave: Anúncios *online*. Fraude de clique. CAPTCHA Clicáveis.

Abstract

In the current challenging climate of the global economy, advertisers and their advertising agencies are in continuous search for business opportunities that lead them to significantly increase the exposure of their brands and their products at increasingly lower costs. Marketing resources have been gradually redirected to marketing campaigns such as Pay Per Click (PPC), which consists of associating payment to some potentially lucrative action for the advertiser, such as clicking on an ad by a user. Within this market structure, a fraudster may appear as someone who seeks to increase the profits of an ad publisher by merely clicking on the ads displayed on the site of the publisher without real intention to buy the product, or as a competitor clicking on the ads of a particular organization in order to generate undue costs. This entire process is exponentially more costly for the victims if it is done in an automated manner through scripts and bots, such that combating, or even preventing, click fraud is of fundamental importance for the continuity of the online marketing advertising, as more conservative advertisers end up avoiding this business structure for their risk aversion profile. To achieve the ideal world, where click fraud either does not exist or has minimal impact on the companies revenues, it is necessary to develop computational techniques and security solutions that support the growth of this market. In this sense, this research aims to produce contributions and develop solutions for online ads, focusing on the PPC business, including the provision of academic material in the Portuguese language about the systematic, history and evolution of online advertising as well as their frauds, legal cases and forms of dectecion, and the development of an approach to combating click fraud that combines classic forms of detection with an innovative approach to prevention, based on clickable CAPTCHA, something neither found in the literature on past researchs nor on the current solutions available in the market. Thus, the developed prototype introduced Google NoCAPTCHA reCAPTCHA as a way to prevent automation of click fraud: by clicking on an ad, the user will need to respond to a challenge and, once identified as a human, proceed. The authentication aspect ensures continuous use of the system without the need to respond additional challenges through the use of temporary coupons represented as cookies. This characterizes a paradigm shift in the fight against click fraud, which happens to be from reactive detection to proactive prevention, since traditional approaches used to identify fraud through heuristics that performed historical analysis of clicks after the fraud has already occurred. Indeed, this work contributes to the acquisition of knowledge of the business, through an extensive study of online ads, significantly supporting the identification of business and system requirements, as well as their current market provision. As a benefit, this study made it possible to propose a security approach to online advertising networks, inspired by legal cases and the set of methods used for click fraud.

Keywords: Online advertising. Click Fraud. Clickable CAPTCHA.

Sumário

1 Introdução

Os mercados e a economia global, local de operação e competição dos negócios globais, têm sido alvos de constantes mudanças nos últimos anos. O desenvolvimento de sites comerciais acompanhou a popularização da própria Internet, desde meados dos anos 1990. Anunciar *online* é hoje uma das formas mais rentáveis de realizar campanhas de *marketing*, para empresas que possuam o objetivo de atingir diversos tipos de clientes, sem fronteiras geográficas ou de fuso horário.

Além dessas, há outras vantagens. Por exemplo, anunciar *online* permite a personalização de anúncios, de seu conteúdo e localização. Tanto o Google *AdWords*/*AdSense* (SAGIN, 2015) e o *Yahoo! Search Marketing* (O'REILLY, 2015) permitem que anúncios sejam mostrados tanto em páginas web quanto em resultados de pesquisas de palavras-chave relacionadas. Anunciantes e suas agências encontram-se em contínua busca por oportunidades de negócios que os levem a aumentar significativamente a exposição de suas marcas e de seus produtos a custos cada vez menores (OU, 2012).

Nesse cenário, recursos anteriormente destinados a campanhas de *marketing* tradicionais, que normalmente têm sua estrutura de pagamento associada ao número de pessoas expostas aos anúncios, têm sido redirecionados a campanhas de *marketing* na Internet do tipo Pagamento Por Plique (PPC) ou pagamento por ação, associados à realização de algumas ações, como o clique em um anúncio. São diversas as tendências de mercado (HENDRICKS, 2013), (INTRIERI, 2014) mostrando uma recuperação de investimentos associada a uma maior alocação de recursos com anúncios *online* a partir de 2014. Em contrapartida, fundos associados anúncios por exibição foram reduzidos em 6% pelo mesmo período.

Na Figura 1.1, é possível ver um exemplo que contém anúncios do tipo PPC, no qual o Google realiza papel duplo, o de publicador e o de rede de anúncios. Essa relação tornou-se comum nos primeiros anos de estabelecimento das redes de anúncio online, com organizações como o próprio Google Adwords/Adsense (GOODMAN, 2008), o Yahoo MarketSearch (O'REILLY, 2007) e a Doubleclick (BROWSERMEDIA, 2008). Pode-se identificar as seguintes estruturas na Figura 1.1: 1) palavras-chave que foram buscadas na consulta, 2) anúncios do tipo PPC (exibidos baseados na busca), e 3) resultados da busca.

Figura 1.1 - Exemplo de Anúncios do Tipo Pagamento por Clique

Fonte: google.com, modificado pelo autor

Podemos perceber uma rede de anúncios *online* (doravante, referenciada simplesmente como rede de anúncios) como uma organização que conecta anunciantes a páginas na Internet que desejam publicar anúncios em seus *sites* e subpáginas. O principal negócio de tais redes é vender espaço para que um subconjunto do seu inventário total de

anúncios apareça. Esse subconjunto pode ser visualizado de formas diferentes, incluindo espaços sublocados em *websites* de terceiros, em *feeds* RSS (SOMMERVILLE, 2015), em *blogs*, em aplicações de mensagens instantâneas, em *e-mails* ou em quaisquer outros recursos na Internet.

1.1 Relevância

Grandes portais de internet que publicam anúncios vendem apenas parte do seu portfólio de anúncios através das redes de anúncio, enquanto páginas menores normalmente associam todo o seu conjunto de anúncios através das mesmas.

As projeções da Research and Markets (2015) dão conta que o lucro global que a indústria desse mercado deve se aproximar de 140 bilhões de dólares em 2018. É importante notar que as cinco maiores empresas da indústria concentravam 69% do mercado total de anúncios em 2012, cuja previsão de crescimento foi de 18% em 2010, o que levou a grandes investimentos na área por parte de diversas organizações. Em 2007, o Google comprou a DoubleClick por 3,1 bilhões de dólares, empresa que era a segunda maior rede de anúncio em número de clientes e em faturamento. De acordo com Hargreaves (2014), cerca de 90% do lucro do Facebook procede de anúncios – em termos de negócio, a tendência é se perceber cada vez mais o uso dessa rede social como uma rede de anúncios, já que o seu lucro com o PPC possui uma tendência de 69% de crescimento em 2016 (BRADY, 2016).

Em contrapartida, ao mesmo tempo em que os números associados ao negócio são relevantes, é impossível ignorar o incentivo natural que o modelo PPC em redes de anúncios parece fornecer para a fraude. Os servidores na rede que hospedam os anúncios e os publicadores que os exibem percebem cada clique como uma simples requisição HTTP (SOMMERVILLE, 2015), não possuindo a capacidade objetiva de determinar se foi realmente uma pessoa que de fato iniciou a requisição ou ainda, no caso de se tratar de fato de um usuário

humano, ou se aquela pessoa estava agindo de forma honesta quando a iniciou. Nesse sentido, o ato de gerar cliques como uma tentativa de fraudar a disposição desse modelo de negócios é denominado de “fraude de clique” (OENTARYO, 2014), e também será explorado em detalhes no Capítulo 3.

Conforme descrito por Sagin (2015), a fraude de clique pode beneficiar os fraudadores de diversas formas. A sua forma mais simples ocorre quando um fraudador se utiliza dos anúncios que aparecem em seu *site* com os objetivos de inflar o seu próprio lucro, sem ter o objetivo de realmente visualizar os produtos exibidos. Além disso, a fraude de clique pode ser empregada por concorrentes de uma organização que querem prejudicá-la, através da geração de custos indevidos, uma vez que se identifique a URL associada ao seu anúncio e se aumente exageradamente a quantidade de cliques realizadas nele, gerando cobranças. Muitas vezes essas duas formas da fraude utilizam ferramentas automatizadas, como *scripts* e *bots*, para extrapolar ainda mais os seus efeitos (SAGIN, 2015).

Todas as partes envolvidas no negócio podem ser prejudicadas direta ou indiretamente pelas fraudes de clique. Embora seja até possível vislumbrar um benefício financeiro inicial para as redes de anúncio – pois no fim das contas as mesmas recebem o valor da cobrança associado à fraude –, no longo prazo, quando os seus clientes passam a se tornar sensíveis às perdas, a fraude de clique pode colocar em risco a relação entre essas entidades e os seus anunciantes. Desta forma, para garantir a credibilidade do negócio de anúncios online, toda e qualquer forma de fraude precisa ser combatida.

Tipicamente, para combater a fraude de clique, as entidades envolvidas, especialmente as redes de anúncios adotam uma postura de detecção da fraude de clique para impedir que os valores de cobrança indevidamente gerados sejam de fato computados junto aos anunciantes. Essa detecção normalmente consiste da realização de buscas por cliques fraudulentos, nos seus históricos de navegação e registros de anúncios requisitados, nos custos

das palavras-chave que foram buscadas, nos endereços IP que originaram as requisições e no número de requisições desses endereços.

Embora essas técnicas sejam eficientes contra a fraude de clique menos especializada, elas possuem limitações contra fraudadores mais sofisticados, que conseguem usar redes sofisticadas de fraude de *bots* (DANCHEV, 2014) mascarar cliques através de *proxies* (DOYLE, 2003), e assim burlar os filtros mais básicos de detecção, enquanto realizam a fraude de clique de larga escala. Sagin (2015) recentemente relatou que *bots* clicando em *links* constituem até 85% do tráfego obtido em uma campanha mensal típica de pagamento por clique no Google Ad Words (CICILIANO, 2015).

Além disso, em termos práticos, a abordagem de detecção não pode garantir que a fraude de clique deixará de acontecer, pois ela sempre será uma medida contingencial. Tomando por base a premissa de que a melhor forma de segurança em um modelo de negócios proeminente como esse em termos financeiros seria a prevenção de cliques fraudulentos, propomos neste trabalho a complementação da detecção através de uma mudança de paradigma: ao invés de apenas realizar buscas para a detecção da fraude, propomos tomar medidas para considerar caminhos para antecipá-la e impedir que a mesma ocorra, ao menos em larga escala (automatizada), através da realização de testes para identificar que os usuários clicando nos anúncios são, de fato, humanos.

Impedir a fraude de clique parece ser de fundamental importância para a continuidade e o crescimento do mercado de anúncios online. Anunciantes mais conservadores acabam se afastando desta forma de *marketing* por sua natural aversão ao risco (ZELLER, 2004) e, assim, a possibilidade de se desenvolver novos negócios acaba impactada, inclusive através de redes de anúncios, mesmo sendo essas benéficas para o consumidor.

Para se alcançar o mundo ideal, no qual a fraude de clique influencia minimamente o negócio de anúncios online, o surgimento de técnicas de segurança que suportem o crescimento deste negócio é necessário. Nesse cenário, a academia desempenha papel fundamental, por se tratar da forma de pesquisa e desenvolvimento mais indicada para a predileção de cenários para a exploração e resolução de situações tecnicamente desafiadoras.

Nesse contexto, a hipótese desse trabalho é que prevenir a fraude de clique automatizada é possível e que essa prevenção, associada a uma abordagem de detecção, é a configuração ideal, em termos de segurança, para o combate à fraude de clique, por tornar a sua realização extremamente custosa. Para alcançar esse objetivo, é necessário realizar um estudo sobre a sistemática e fraudes em anúncios *online*, de maneira a se construir um arcabouço teórico e científico sobre as necessidades técnicas e de negócio dos participantes desse nicho de mercado. Entendemos que, por um estudo como esse ser inédito em língua portuguesa, esta tese de doutorado encontra uma oportunidade de disponibilizar esse material em vernáculo, ao construir tal corpo técnico-científico de conhecimentos.

Paralelamente, o PPC tem importância significativa para a forma atual da realização de negócios pela Internet e pode ser visto como uma das principais formas de anúncios através das redes de anúncios. Nesse sentido, esta pesquisa apresenta também uma contribuição para anunciantes na Internet, que possuem o natural interesse de elevar a segurança e a garantia de negócios legítimos de pagamento por clique. Com efeito, esta pesquisa vislumbra o atual estado da arte dos sistemas de redes de anúncios e busca descrever uma proposta de segurança com o objetivo de avançar o conhecimento científico nessa área, ao levar em consideração elementos desse estado.

Por fim, o trabalho também será recomendado a usuários de Internet em geral, que são consumidores em potencial dos serviços disponibilizados pelas redes de anúncios e

publicadores. Ela os auxiliará a entender a disposição atual do negócio de pagamento por clique na Internet e a buscar, de maneira proativa, dentre as diversas opções disponíveis, a utilização de serviços que disponibilizam níveis aceitáveis de segurança. Estes serviços devem contribuir, de maneira ética, para a evolução contínua da utilização de negócios online.

1.2 Objetivos

A visão desta pesquisa é produzir contribuições para a área de segurança para a web com o foco específico no domínio de anúncios *online* e, especificamente, desenvolver mais o estado da arte das tecnologias envolvidas no negócio de PPC. Tais contribuições apontam para o objetivo geral da pesquisa, que consiste em realizar uma proposta para a prevenção da fraude de clique automatizada por meio de CAPTCHA clicáveis.

Nesse sentido, o primeiro passo para a prevenção da fraude de clique automatizada é a necessidade da utilização de métodos para diferenciação entre usuários humanos e não-humanos, que precisam ser empregados, o objetivo geral desta pesquisa converge na proposição de uma abordagem baseada em CAPTCHA (*Completely Automated Public Turing test to tell Computers and Humans Apart*) clicáveis que vislumbra adotar uma estratégia de prevenção para essa fraude automatizada em redes de anúncios.

Mais especificamente, o objetivo geral pode ser detalhado nos seguintes objetivos específicos, que contemplaram trabalhos relacionados à:

• Sistemática dos anúncios *online* e de sua atual disposição de negócios, para garantir que a abordagem proposta se adequa a esse modelo de negócios;

• Métodos de fraude neste nicho de negócios, especialmente fraude do clique. Estudo de casos legais, como o relatório de Tuzhilin (2006);

• Estudo de CAPTCHA, histórico, classificação, análise de benchmarks, *Optical Character Recognition* (OCR), e a introdução de CAPTCHA clicáveis como uma forma para prevenir a automatização da fraude de clique;

• Referencial teórico sobre métodos de autenticação pela Internet e a sua utilização para aumentar a confiabilidade de abordagem a ser proposta;

• Estudo de abordagens de detecção e a proposta de adaptação do algoritmo *Similarity-Seeker* como um método complementar aos CAPTCHA clicáveis;

• Proposta de uma abordagem baseada na utilização de CAPTCHA clicáveis, incluindo uma mudança de paradigma no combate à fraude de clique, se propondo a ser de prevenção. Tal proposta, no entanto, não ignorará os aspectos clássicos de detecção da fraude de clique já consagrados no mercado e na academia.

1.3 Metodologia

Levando os objetivos expostos na seção 1.1 em consideração, um conjunto de foram elaboradas como parte da metodologia deste projeto de pesquisa. Assim, a realização do mesmo pode ser dividida em quatro momentos de referência pertinentes a diferentes métodos e técnicas que utilizamos para investigação, descritos abaixo:

1. Revisão bibliográfica: momento inicial da pesquisa, quando organizamos e compilamos as pesquisas que apresentavam técnicas de segurança da informação na Internet, especialmente aquelas voltadas ao combate da fraude de clique, análise das redes de anúncios online e seus atuais métodos de combate à fraude;

2. Revisão da tecnologia: momento quando analisamos as tecnologias do Google NoCAPTCHA reCAPTCHA (SHET, 2014) e do *Similarity-Seeker* (KITTS et al, 2013)

(METWALLY et al, 2007). Estudamos as especificações técnicas desses projetos e suas aplicabilidades ao problema em questão, incluindo uma análise detalhada de segurança, considerando sua aplicabilidade em PPC;

3. Desenvolvimento do *software*: momento quando, baseados na revisão de tecnologia, realizamos o desenvolvimento de uma camada de *software* integrada a um provedor de serviços de redes de anúncios. Teve por objetivo permitir o aumento da segurança do serviço oferecido, de maneira a permitir a análise e visualização de redes de anúncios de forma acessível para todos os usuários do provedor de serviços, isso se cujas identidades houverem sido validadas por um teste CAPTCHA.

Para realização da etapa de desenvolvimento, identificamos as subtarefas:

3.1 Proposta de um protótipo de rede de anúncios que implementa a abordagem proposta, introduzindo componentes e subsistemas tais como:

3.1.1 Subsistema gerador de anúncios e um estudo de heurísticas para construção de anúncios a serem exibidos nos sites de Internet;

3.1.2 O sistema gerador de CAPTCHA e a proposta do tipo de CAPTCHA a ser utilizado nesta abordagem, incluindo um esquema de desenvolvimento do fluxo de comunicações associado à resolução dos desafios, bem como o estudo e escolha de algoritmos para a criação de desafios CAPTCHA;

3.1.3 Modelo de dados para possibilitar as funcionalidades associadas aos componentes propostos na abordagem - de geração de anúncios ao armazenamento de cliques;

3.1.4 Proposta da sistemática de comunicação entre usuários e rede de anúncios - inclui troca de *cookies* (PUTTACHAROEN; BUNYATNOPARAT, 2011), resposta a desafios e comunicações;

3.1.5 Análise e prototipação do algoritmo *Similarity-Seeker* para detecção.

4. Validação da pesquisa: o momento onde aplicamos um conjunto de testes controlados de dados para analisar a eficácia e segurança da solução. Esta fase incorporou a validação da pesquisa, e encontra-se detalhada na seção 6. Uma vez validado o ambiente, realizamos uma avaliação das redes de anúncios no mercado existentes, sobretudo aplicando medidas de centralidade para identificar as principais providências de segurança que precisariam ser aplicadas em cada um dos provedores de serviço para garantir o seu uso de maneira computacionalmente segura.

Esta atividade incluiu as subtarefas:

4.1. Análise técnica da segurança computacional da abordagem como um todo;

4.2. Análise da usabilidade do produto a ser desenvolvido com base na literatura e em *softwares* e projetos existentes, que nos levou a escolher pelo uso de autenticação por cupons através de *cookies* (JUELS et al, 2006);

4.3. Uso de cenários diversos de fraude e descrição de como o sistema se comportava em tais cenários, com o objetivo de validar a solução como um todo.

1.4 Estrutura

Esta tese está organizada como a seguir: no Capítulo 2, introduzimos os anúncios online, seu histórico e estado atual, e também um número de fraudes que neste ambiente de mercado, incluindo a fraude de clique, tema central dessa pesquisa. Apresentamos métodos de detecção e combate atuais, problemas e como uma abordagem de prevenção é necessária. No Capítulo 3, introduzimos o conceito de CAPTCHA como uma possível solução para a fraude de clique automatizada e introduzimos possibilidades de autenticação para dirimir problemas

associados à usabilidade de tais sistemas. No Capítulo 4, realizamos a proposta da arquitetura de rede de anúncios que utiliza CAPTCHA clicáveis para prevenção, implementada em forma de protótipo, cujos detalhes arquiteturais são expostos no Capítulo 6. Explicamos também porque uma abordagem de detecção é importante como aspecto complementar dos módulos de prevenção e dissertamos, no mesmo Capítulo 6, dentre diversos métodos consagrados de detecção, a razão pela qual adotamos o algoritmo *Similarity-Seeker* para possibilitar a detecção de cliques fraudulentos com especial foco na fraude do publicador. Por fim, no Capítulo 7, apresentamos as considerações finais e trabalhos futuros.

2 Anúncios e Fraudes Online

Anunciar pela Internet é uma das formas mais rentáveis de realizar campanhas de *marketing* com o objetivo de atingir categorias diferentes de clientes, tanto para pequenas quanto para grandes empresas (INTRIERI, 2014). Um dos grandes benefícios de anunciar *online* é a possibilidade de publicar informação e conteúdo sem fronteiras geográficas – na verdade, é possível personalizar tal conteúdo em função de variáveis como essa, variando-se o conteúdo dos mesmos em função dos lugares e do público alvo onde serão exibidos. Por exemplo, é muito comum grandes redes de anúncios, como o Google AdWords/AdSense e o Yahoo! MarketSearch permitam que anúncios sejam exibidos em páginas *web* relevantes e em resultados de pesquisas. Com esse fim, a área emergente de anúncios interativos apresenta alguns novos desafios para anunciantes, como veremos neste Capítulo.

2.1 Histórico

De acordo com Sagar (2009), a Internet começou a ser utilizada efetivamente para fins comerciais em meados de 1997. Nesse período, milhares de novas páginas foram disponibilizadas, os grandes portais de Internet nasceram e bilhões de dólares em capital de ventura fluíram através deles. Essa foi a primeira vez que os *websites* se dividiram em categorias diferentes: os de comércio eletrônico, com o objetivo principal de venda de produtos, e os de oferta de conteúdo, que são criados para disponibilizar conteúdo, como palavras, imagens e vídeos, para que pessoas interessadas e leitores os acessem.

Para suportar a própria atividade dos *sites* de oferta de conteúdo e atrair mais clientes, a tecnologia de anúncios para eles passou por um processo de evolução ao longo dos anos. Se considerarmos o período inicial, ou seja, meados dos anos 90, anunciar na Internet

significava elaborar anúncios em *banners*, imagens que tipicamente possuíam 728x90 *pixels* e eram exibidas em praticamente todas as páginas na Internet de então.

No fim dos anos 90, essa forma de anunciar na *web* era considerada muito lucrativa (HOFFMAN; NOVAK, 2000), e portais populares como o Yahoo cobravam entre 30 e 100 dólares a cada mil exibições de um determinado anúncio em um *banner* em suas páginas, que contavam com cerca de 100 milhões de acessos por mês e eram inspirados em exibições de anúncios em revistas de grade circulação (RYAN, 2006).

Figura 2.1 – Anúncio em banner padrão no topo da página

Fonte: http://www.howstuffworks.com (2012)

De acordo com Zeff e Aronson (1999), a partir de um determinado ponto, os anunciantes perceberam que o retorno gerado por anúncios em *banners* era muito baixo. Os grandes portais removeram os anúncios em *banner* da sua oferta, e organizações empresariais como a DoubleClick (O'REILLY, 2007) surgiram, com o objetivo de centralizar e otimizar a função de vender o melhor anúncio possível, com respeito ao formato e outras customizações.

De forma indireta, o princípio econômico da oferta e da procura (STONE, 2004) começou a ser aplicado na Internet, e as taxas pagas por anúncios em *banners* começaram a flutuar.

Quando o ambiente de negócios é a Internet, exibir anúncios acontece tipicamente por duas razões (HOFFMAN; NOVAK, 2000): desenvolvimento da marca (*branding*) (KOTLER; KELLER, 2009), e possiblidade de vendas diretas (STONE, 2004). Como os anunciantes começaram a retirar os anúncios dos grandes portais, o preço para alocar anúncios em *banner* começou a ficar mais barato (KANG; LEE, 2003). As organizações de anúncio da Internet procuraram soluções para que seus anúncios possuíssem mais influência em *branding* e vendas diretas potenciais (DIRECT MARKETING ASSOCIATION, 2006).

Percebeu-se então que havia uma necessidade de mudança de formato e, inspirados nessas necessidades, outros formatos de anúncios na Internet começaram a ser desenvolvidos. Um desses foram os anúncios em barras laterais, que são similares aos anúncios em *banner*, mas orientados verticalmente. A altura da barra lateral pode atingir 600 pixels, com uma largura de normalmente 120 pixels (KANG; LEE, 2003), e tem mais impacto do que um anúncio em *banner* por ficar mais tempo "visível" ao usuário. Em um *banner* clássico, rolar a tela pra baixo em apenas 60 pixels é suficiente para remover o anúncio da tela do usuário. A exposição do usuário em barras laterais é cerca de dez vezes mais longa, pois eles se adaptam à forma como os usuários utilizam os navegadores de Internet.

Por causa desse impacto positivo para a estratégia de *marketing*, os anúncios em barras laterais têm um poder de desenvolvimento de marca maior e originam uma maior taxa de cliques por anúncio. Segundo Chaffey et al (2006), "um anúncio em barra lateral típico apresenta uma taxa média de 1% de cliques por exibição", ou de cem cliques a cada mil exibições, o que o torna cerca de três vezes mais eficiente que um anúncio de *banner*. Como se pode imaginar, os portais e páginas de internet, ao perceberem essa tendência de favorecimento

de consumo, aumentaram o valor de cobrança para os anunciantes para disponibilizar esse tipo de anúncio em suas páginas (CHAFFEY et al, 2006).

Enquanto anúncios em *banner* e em barras laterais possuem tamanhos "padronizados", nos últimos anos, tem-se visto na internet uma proliferação de diferentes tipos de anúncios online, incluindo tamanhos e locais de exibição próprios. Um exemplo pode ser visto na Figura 2.2, que contém um anúncio no topo, além de um *banner* padrão, um quadro no centro direito da página, e quatro anúncios menores na parte de baixo.

Figura 2.2 – Diversos anúncios diferentes na mesma página

Fonte: http://www.mapquest.com

É possível observar também algumas outras formas interessantes utilizadas por redes de anúncios como descritas por Goodman (2008), como colocar mais de um anúncio da mesma marca na mesma página, com o objetivo de aumentar a probabilidade de cliques em um deles. Redes de anúncios são estudadas mais detalhadamente na seção 2.4.

Um anúncio em janelas de *pop-up* consiste na exibição de um anúncio em uma janela não requisitada quando um usuário acessa uma determinada página. Esse tipo de anúncio

normalmente sobrepõe a página que o usuário está tentando acessar, de maneira que o mesmo tem que fechar a janela contendo o anúncio ou movê-la do seu caminho. Anúncios *pop-under* são similares, mas se localizam dentro do conteúdo que o usuário está tentando ler e são menos intrusivas. Na Figura 2.3 pode-se ver um exemplo de anúncio deste tipo.

Figura 2.3 –Anúncios *pop-up* na frente da página principal

Fonte: http://www.britannica.com (2014)

Anúncios *pop-up* são considerados “irritantes” para a maioria dos usuários porque enchem a área de trabalho e demandam tempo e ações dos usuários para serem fechadas (O’TOOLE, 2014). Entretanto, em termos de resultados de campanha *marketing*, eles são mais eficientes que os anteriormente descritos, podendo chegar a uma média de trinta cliques em mil

exibições, ou seja, 3% de prospecção (BRIAN, 2002), sendo esta a razão pela qual tantos anúncios desta categoria podem ser vistos na *web* até os dias de hoje.

Os anúncios flutuantes (em inglês, *floating ads*) aparecem tipicamente quando o usuário realiza o primeiro acesso a uma página na Internet e eles parecem "voar" por algum tempo. Quando estão na tela de exibição, eles sobrepõem a visão da página que o usuário busca ler. Muitos deles são, ainda, configurados para ignorar os comandos que o usuário dá pelo *mouse* ou teclado, para garantir sua exibição completa. A Figura 2.4 mostra um exemplo de um anúncio flutuante animado, com quatro partes que se movem independentemente.

Figura 2.4 – Anúncio flutuante para um produto da Norton

Fonte: Norton.com (2013)

As altas taxas de cliques (média de 3%) associadas ao maior poder de *branding* são naturalmente traduzidas nos preços dos mesmos. É natural que eles tenham atraído interesse

dos anunciantes e que os portais tenham se interessado a nesse tipo de anúncios quando os mesmos adentraram o mercado em meados dos anos 2000.

Finalmente, anúncios em vídeo introduzem uma mudança de perspectiva de anúncios *online* muito significativa e tem se tornado cada vez mais populares na Internet, com taxas de clique extremamente satisfatórias. Funcionam como um comercial de TV que roda em uma janela de *pop-up*, ou em uma janela previamente aberta, como a de um vídeo. O site Unicast.com traz vários exemplos desse tipo de anúncio e eles têm se popularizaram em meados de 2014 no Youtube.com antes das exibições dos vídeos requisitados pelos usuários. Estima-se que um anúncio dessa natureza tem a capacidade de desenvolvimento de marca equiparada a de um comercial de TV (INTRIERI, 2014), com o diferencial de oferecer a possibilidade de clicar no anúncio para obter mais informações. Números iniciais sobre as taxas de clique nesse tipo de anúncio chegam a 5%.

Sobre taxas de reclamação, muitos anúncios são considerados “chatos” pelos usuários e por isso eram evitados em alguns portais. Os navegadores de Internet disponíveis do mercado chegaram a políticas de bloqueadores de *pop-up*. No entanto, após um determinado tempo e com o surgimento de outros tipos de anúncio, auxiliados também pela implementação de políticas de privacidade, o que permitiu uma maior possibilidade de seleção de conteúdo, os usuários acabaram se acostumando com os mesmos e as taxas de reclamação foram cada vez mais sendo reduzidas (ALSUHIBANY, 2011).

2.2 Estado da prática

Com o passar do tempo, será possível ver cada vez mais variações de anúncios online. Já podemos citar algumas destas abaixo:

- O *site* HowStuffWorks (http://www.howstuffworks.com), em fevereiro de 2013, ofereceu pela primeira vez o que foi denominado de "*takeover campaign*" (em tradução livre, campanha pública de aquisição) (IAB, 2015). Na primeira visita ao site, os visitantes recebiam um anúncio, e a mensagem era reiterada em praticamente todas as subpáginas através de *banners* e barras laterais. Essencialmente, o anunciante "adquiriu" os anúncios do portal por um ou mais dias. A ideia da *takeover campaign* é oferecer ao anunciante todos os espaços reservados para *marketing* em determinado site durante o período da campanha, tornando os anúncios do mesmo visíveis em todo o *site*;

- Anúncios em *banners* suspensos apareceram em alguns sites. Ao se colocar o *mouse* em cima do anúncio, o mesmo expande para preencher praticamente toda a página, e depois retorna ao tamanho normal. Nas Figuras 2.5 e 2.6 vemos um exemplo no qual o banner é expandido por alguns segundos e depois é "encolhido" para o tamanho normal. O site em questão utiliza *banners* com área de 725 x 70 pixels, e a largura do site é determinada pelos anúncios. Botões no anúncio possibilitam aumenta-lo caso os usuários assim desejem.

Figura 2.5 – Anúncio suspenso exibido por alguns segundos

Fonte: http://www.zdnet.com (2015)

Figura 2.6 – Anúncio suspenso após retornar para o tamanho original

Fonte: http://www.zdnet.com (2015)

- Anúncios por *e-mail*: anunciar por *e-mail* é uma forma de marketing direto (STONE, 2004) que comunica mensagens comerciais para uma determinada audiência. No

sentido mais genérico desta forma de anúncio, cada *e-mail* enviado para clientes pode ser considerado uma forma de anunciar. De acordo com a Direct Marketing Association (2015), empresas norte-americanas investiram cerca de 400 milhões de dólares com anúncios por *e-mail* em 2015. Esse número se reduziu consideravelmente nos últimos anos, mas ainda se encontra em posição de destaque, com 75% das organizações ainda investindo em mensagens personalizadas (DIRECT MARKETING ASSOCIATION, 2015).

- *Marketing* de afiliados: tipo de marketing em que uma organização compensa um ou mais parceiros, denominados afiliados, por cada novo visitante ou cliente referenciado ao negócio devido aos esforços de *marketing* de tal afiliado. Exemplos desta disposição incluem alguns serviços online de recompensa, nos quais usuários são compensados financeiramente ou com alguma recompensa ao completarem uma ordem de serviço ou indicarem novos usuários para o serviço (SHEKHAR, 2009).

- Anúncios contextuais: muitas redes mostram determinados anúncios motivados pela busca de conteúdo na Internet realizada previamente, ou baseados no conteúdo exibido nas páginas onde o anúncio é mostrado. Acredita-se que esses anúncios possuem uma probabilidade maior de atrair os usuários, pois têm a tendência de compartilhar um contexto similar com o seu interesse.

- Anúncios semânticos: utiliza tecnologias da *web* semântica (ZAMANZADEH, 2013) a soluções de anúncios *online*. A função dessa abordagem é analisar semanticamente a página em exibição, interpretar o seu significado, e garantir que ela contenha o anúncio mais apropriado. A ideia é utilizar a web semântica para aumentar as chances que os usuários cliquem nos anúncios ao aumentar a relevância e o interesse do que está sendo exibido para os usuários.

Todas essas diferentes formas de anunciar na Internet são tentativas de se achar a combinação ideal para fornecer aos anunciantes o que eles desejam – altas taxas de clique e

poder de desenvolvimento de marca. Em retorno, os anunciantes possuem consciência de que o *marketing* na Internet funciona e, assim, propõem-se a pagar pela exibição dos seus anúncios. Dentre os modelos de pagamento adotados, o mais relevante para a fraude de clique é o chamado pagamento por clique, que detalhamos a seguir.

2.3 Pagamento por clique

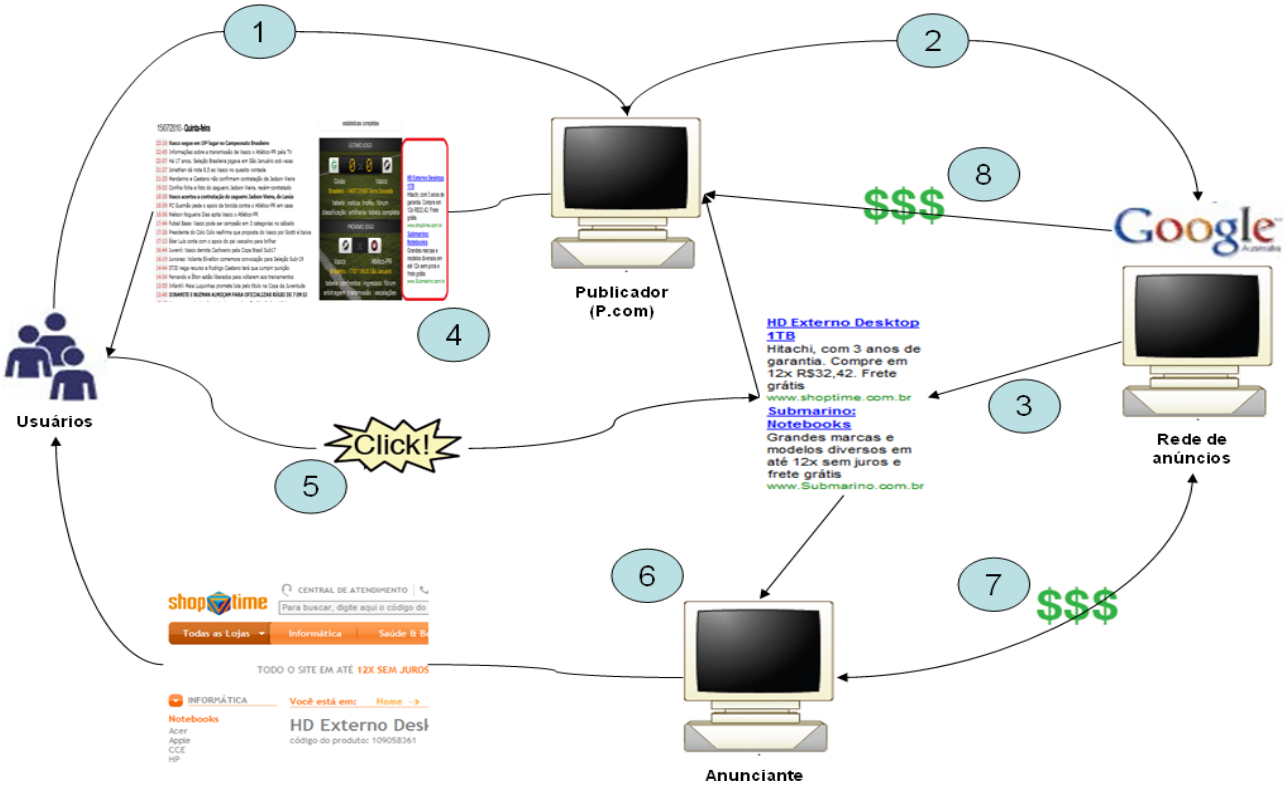
O pagamento por clique é, em síntese, um acordo entre empresas. O primeiro grupo de empresas, os publicadores, exibe em suas páginas na Internet alguns *links* clicáveis, que são anúncios do segundo grupo, os anunciantes. Em troca, há uma cobrança por cada clique realizado naqueles *links*, por parte do publicador para com o anunciante. Com o crescimento dessa indústria, desenvolveu-se uma terceira entidade, denominada rede de anúncios, que passou a agir como mediadora entre os dois grupos iniciais, publicadores e anunciantes, e que cresceu de maneira muito rápida. Nesse novo modelo, sempre que um usuário da *web* clica em um anúncio, o anunciante paga um valor à rede de anúncios, que repassa uma parte desse pagamento ao publicador. A vantagem de existir essa entidade adicional é que anunciantes possuem uma maior exposição e publicadores contratam um *portfólio* maior de anúncios de uma entidade única, o que em teoria pode aumentar seus lucros com cliques nos anúncios exibidos em suas páginas.

Conforme esquematizado na Figura 2.7, sempre que um usuário da Internet visita a página de um publicador, esse usuário é associado a um dos servidores da rede de anúncios. O servidor escolhe um conjunto de anúncios e o repassa ao publicador, que exibirá os anúncios juntamente com o conteúdo da página no navegador do usuário. Se o usuário decidir clicar em um anúncio, haverá o registro de cobrança, e a mesma será registrada no momento em que o

usuário clicar em um anúncio. Parte dessa cobrança será repassada para o publicador, já que o usuário acessou o anúncio a partir dos seus conteúdos.

O sistema de compartilhamento de receitas entre a rede de anúncios e os publicadores pode ser visto como um incentivo natural para a fraude de clique. Uma das razões para isso é que as redes de anúncios podem agir também como publicadores de anúncios através de suas ferramentas de busca.

Figura 2.7 – Sequência de ações de uma solicitação PPC clássica



Fonte: Próprio Autor

De acordo com alguns críticos (MANN, 2006), (HADADDI, 2010), (DE QUEIROZ; COSTA, 2013), (OENTARYO et al, 2014), essa relação complexa acaba gerando um conflito de interesses: ao mesmo tempo em que a rede perde dinheiro quando não consegue

detectar a fraude, pois tem que pagar o publicador, ela também ganha ao realizar cobranças originárias dos cliques fraudulentos dos anunciantes. Devido ao fato de as taxas pagas pelos anunciantes serem compartilhadas entre a rede e o publicador, e já que a rede de anúncios só repassa ao publicador uma parte geralmente menor do que ganha, a rede de anúncios acaba ganhando dinheiro com a fraude.

2.4 Redes de anúncio online

Uma rede de anúncios *online*, ou simplesmente rede de anúncios, é uma empresa que conecta anunciantes para os *sites* que querem hospedar anúncios publicitários, que chamamos neste trabalho de publicadores de anúncios. Elas agregam a oferta existente no mercado através do espaço publicitário dos publicadores, combinando-a com a demanda existente do lado dos anunciantes. Na verdade, a expressão "rede de anúncios" por si só indica neutralidade de mídias, no sentido que poderia haver uma "rede de anúncios televisiva" ou ainda uma "rede de anúncios impressa". No entanto, a diferença fundamental entre as redes tradicionais de anúncios e redes de anúncios *online* é que essas últimas usam servidores de anúncios central para fornecê-los para os consumidores, e cada anúncio está associado a um determinado site na Internet. Um trecho de código em um *iFrame*, por exemplo, é invocado do servidor de anúncios pelo site que publica o anúncio, e a resposta a essa requisição é, normalmente, um *banner* contendo um ou mais anúncios, conforme exibido na Figura 2.8.

Figura 2.8 – Destaque para um banner de rede de anúncios em um site de jogos online

Fonte: http://www.sokker.org, com destaques do autor

Tipologia

Há basicamente duas formas de se classificar redes de anúncios online. A primeira classificação, que diz respeito à distribuição dos anúncios, separa as redes em três tipos:

1. **Redes verticais**: essas redes atuam como representantes dos locais onde o seu portfólio de anúncios será publicado. Nesse caso, há transparência total do anunciante a respeito dos locais onde seus anúncios irão aparecer. Essas redes visam a qualidade do serviço, oferecendo tráfego diferencial de dados a preços acessíveis, e são usadas extensivamente por empresas interessadas em construir suas marcas. Seu modelo de negócio é normalmente baseado no compartilhamento anteriormente acordado das receitas obtidas com os anúncios, objetivando exposição, o que favorece *branding*.

2. **Redes cegas**: essas empresas se propõem a oferecer preços mais baratos para empresas interessadas em vendas diretas que não necessitam de ter controle do local onde os seus anúncios irão aparecer (STONE, 2004). Elas são bem mais baratas do que os outros tipos de redes de anúncios e conseguem isso através da compra no atacado de locais de anúncio combinadas com a utilização de tecnologia de orientação de anúncios e otimização de

conversão (KING, 2008). Desta forma, o modelo de negócios desse tipo de rede não segue um padrão bem definido (DE JONG; ROSENTHAL; VAN DIJK, 2009).

3. **Redes orientadas**: essas redes focam em tecnologias de orientação e adequação de buscas, tais como análise contextual de navegação e comportamental de usuários. Redes orientadas são especializadas em utilizar dados do consumidor para aumentar o valor do inventário de anúncios aos quais eles serão submetidos.

Uma outra forma menos utilizada de tipificar redes de anúncio *online* é quanto à proximidade do relacionamento comercial da rede com os anunciantes, e consiste em subdividir as redes de anúncios *online* em redes de primeiro de segundo nível (IAB, 2015). As redes de primeiro nível possuem uma quantidade maior de anunciantes cadastrados como seus clientes diretos e possibilitam tráfego de rede de alta qualidade, servindo anúncios e tráfego de rede para grandes portais de internet e para redes de anúncios de segundo nível. Exemplos de redes de anúncios de primeiro nível são as ferramentas de busca que também utilizam de serviços próprios de divulgação. Em contrapartida, redes de anúncios de segundo nível podem até ter alguns clientes cadastrados como anunciantes diretos, mas sua principal fonte de receita se origina do repasse de anúncios de outras redes.

Servidores

De acordo com Weiner (1998), o primeiro servidor de anúncios foi desenvolvido pela empresa FocaLink Media Services. Introduzido em 17 de julho de 1995, o servidor controlava anúncios online em *banners*. A empresa, com sede em Palo Alto, Califórnia, foi fundada por Dave Zinman e Jason Strober. Em 1998, ela foi renomeada para AdKnowledge e foi adquirida pela empresa CMGI em 1999 (WEINER, 1998).

As organizações que desenvolvem tecnologias para veiculação de anúncios *online* fornecem *software* para que redes possam servir os anúncios, realizar contabilidade, escolher quais anúncios devem ser exibidos – e que vão gerar mais receitas para os anunciantes e publicadores – e monitorar o progresso de diferentes campanhas de anúncio. Um aspecto da tecnologia de veiculação de anúncios é a utilização de métodos automáticos e semiautomáticos para otimizar os preços dos anúncios, sua localização e outras características. Podemos citar os métodos a seguir (KHAN, 2007):

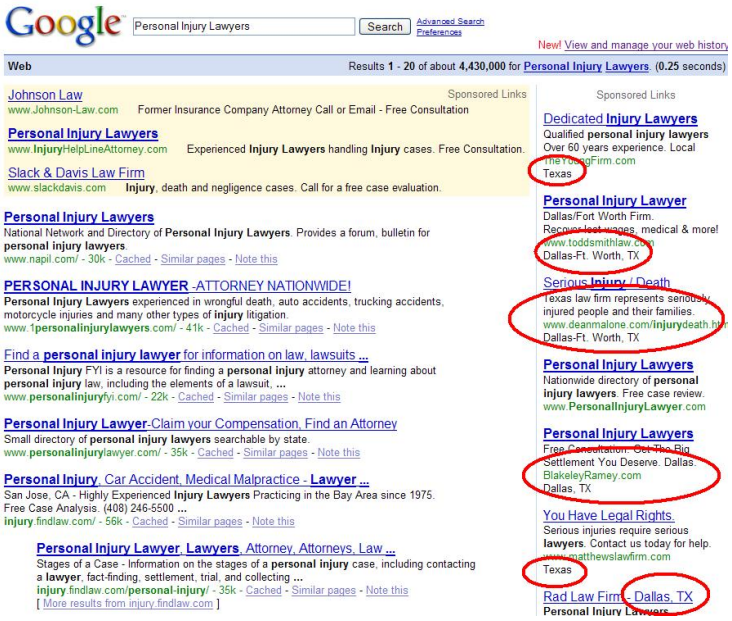
- Orientação comportamental de anúncios: consiste em usar um perfil de comportamento anterior do usuário para determinar qual anúncio deve ser exibido durante uma dada visita. Por exemplo, a exibição de anúncios para venda de carros em um portal de notícias para um usuário que tenha anteriormente visitado a seção de compras de automóveis de uma página na Internet;

- Orientação contextual de anúncios: consiste em inferir o melhor local onde se deve colocar um determinado anúncio a partir da informação contida na página onde o anúncio será exibido. Por exemplo, exibir um anúncio de bicicletas em uma página que contenha um artigo sobre ciclismo.

- Orientação por geolocalização: consiste na utilização de técnicas para identificar a localização real dos usuários de um publicador e, a partir dessa informação, exibir anúncios que sejam interessantes naquela região específica. Um exemplo deste tipo de anúncio é mostrado na Figura 2.9, quando a busca textual no Google por "*Personal Injury Lawyers*" (tradução livre: advogados especializados em danos físicos), realizada da cidade de Dallas, Texas, retorna um número de escritórios de advocacia daquela região. Existe uma boa quantidade de empresas especializadas em geolocalização na Internet e uma lista interessante pode ser encontrada em Smith (2007). Na verdade, através de uma consulta ao *American Registry for Internet Numbers* (ARIN) (SMITH, 2007), qualquer usuário de Internet pode obter

informações (cidade, estado, país e, até mesmo, CEP), associadas no momento a um determinado IP.

Figura 2.9 – Anúncios orientados por geolocalização



Fonte: www.google.com, com destaques do autor

- Otimização criativa: consiste em usar métodos experimentais e preditivos para explorar a melhor forma de se exibir um dado anúncio, e explorar o que foi determinado em exibições posteriores.

- O estudo técnico aprofundado de cada uma das metodologias acima explicadas, bem como da tecnologia e serviços envolvidos na disponibilização de anúncios em páginas e portais de Internet, incluindo os requisitos técnicos e de negócios associados a tais serviços não são o objetivo primordial do escopo desta tese. Mais detalhes sobre estes tópicos podem ser encontrados na matéria de veiculação de anúncios em trabalhos tais como Jakobsson, Mackenzie, Stern (1999), Stone (2004), Goodman (2008), Shekhar (2009), Kotler e Keller (2009), Zamanzadeh (2013) e IAB (2015).

2.5 Considerações do Capítulo

Neste Capítulo 2 foram analisadas algumas técnicas para a sistemática de anúncios na Internet e algumas das diversas formas existentes de disposição do mercado de anúncios. Exploramos em particular a estrutura de pagamento por clique na seção 2.3, e sua natural disposição ao modelo de mercado de compartilhamento de receitas o que favorece o surgimento de modelos tais como as redes de anúncios *online*, que também foram apresentadas neste Capítulo, mais especificamente na seção 2.4.

A abordagem de segurança cuja proposição será realizada como parte desta tese está fundamentada na realização de um estudo extensivo do domínio de negócios de anúncios *online*, de maneira a entender o que se pode fazer para prevenir a fraude de clique a qual tem sido largamente discutida na academia e no âmbito de mercado, e representa um verdadeiro desafio para este negócio, de maneira que fundamentar de maneira sólida é de importância para entender todas as necessidades desse nicho de mercado.

Uma das finalidades principais deste trabalho está relacionada com a proposta de uma abordagem para o combate efetivo à fraude de clique. Para que a construção da abordagem que seja realizável no modelo de negócios existente, ela adotará um conjunto de métodos já propostos anteriormente e utilizados em organizações. Tais métodos foram implementados em projetos de grande porte, tendo apresentado resultados satisfatórios sob a perspectiva dos participantes do negócio e dos demais envolvidos em pesquisas nesta área, no âmbito da academia. Após a proposição desta abordagem, acreditamos que a comunidade contará com uma ferramenta poderosa para trabalhos futuros, que podem conter a especialização desta abordagem no nível de arquitetura e desenvolvimento.

3 Fraude de Clique

Uma rede de anúncios *online* pode sofrer abusos de diferentes formas. Na verdade, cada um dos modelos de receita referenciados anteriormente está sujeito a fraudes que têm como principal consequência a redução do lucro dos anunciantes. Alguns exemplos dessas fraudes seriam a fraude de exibição, quando requisições para sites anunciantes são realizadas sem a exibição efetiva dos anúncios, a fraude de conversão, quando publicadores fraudulentos tentam obter receita a partir de ações produzidas artificialmente, e a fraude de clique, que é o foco de estudo deste Capítulo (CHO et al, 2015).

Fraudes *online* têm como principal consequência a redução do faturamento e do lucro dos anunciantes. O foco deste Capítulo é analisar as fraudes *online* como uma das muitas variáveis que podem ocasionar prejuízos a anunciantes e, dessa forma, uma taxonomia dos tipos diferentes de fraude que podem impactar uma rede de anúncios será apresentada.

Uma dessas fraudes é fraude de exibição, que consiste no resultado de requisições HTTP transparentes para os usuários, enviadas para páginas na *web* que geram custos para os anunciantes, que por sua vez estão pagando pela exibição que nunca chega para os usuários. A fraude de exibição afeta campanhas de anúncio baseadas em pagamento por exibição, pois, na prática, os anunciantes não deveriam ser cobrados por tais requisições HTTP. Com efeito, a fraude de exibição afeta a taxa de cliques recebidos pelos anúncios, também denominada de *click-through rate* (CTR)[1], uma vez que o denominador comum no cálculo do valor é o número de exibições da página (IMMORLICA et al, 2005). Como o *ranking* entre os diferentes anúncios de uma rede pode depender do CTR, um anunciante malicioso pode manipular essa

[1] *Clickthrough rate* ou "taxa de cliques por impressões" é uma forma medir o sucesso de uma campanha de anúncios online e é calculada pela razão entre o número de usuários que clicou em um anúncio em uma página web e o número de vezes que o anúncio foi exibido, o que é também conhecido como o número de impressões.

taxa para alterar o resultado de um leilão de anúncios. O tipo de modelo de negócios para pagamento mais utilizado para o modelo de impressão é o chamado *cost per mile* (CPM), ou custo por milha, em tradução livre. A métrica de aquisição típica de um anúncio nessa abordagem é a sua exibição para mil usuários – mil exibições do anúncio (FARRIS et al, 2010).

Por sua vez, a fraude de conversão ocorre quando requisições HTTP são submetidas com o objetivo de produzir conversões que não são geradas por usuários reais. Uma conversão ocorre quando uma requisição HTTP por uma URL pré-determinada pelo anunciante acontece – qualquer que seja ela. Ou seja, quando uma ação acordada entre o anunciante e uma terceira parte, publicador ou rede de anúncios, acontece por parte de um usuário, há uma conversão (CHAFFEY et al, 2010). Essa ação pode ser a visualização de uma página, a submissão de um formulário, o início do *download* de algum arquivo ou até mesmo a finalização de uma venda *online*. Da mesma forma como há um incentivo para a fraude de exibição em anúncios do tipo custo CPM, existe um para a fraude de conversão em anúncios do tipo pagamento por clique (PPC), de maneira a inflar a quantidade de cliques relevantes (OENTARYO et al, 2014).

Em sua forma automatizada, basta programar um *script* para simular o clique em uma URL de anúncio do site do publicador e ser redirecionado para o site do anunciante e, após isso, realizar uma ação na página destino como, por exemplo, realizar o download de um arquivo, como uma versão de avaliação de um pacote de software, em uma tentativa de simular o comportamento de um usuário legítimo.

3.1 Fraude de Clique

Em termos conceituais, a fraude de clique ocorre quando requisições pelos endereços dos anúncios em exibição na *web* são realizadas sem intenção legítima por usuários que eram considerados legítimos inicialmente. No momento da detecção da fraude, tais usuários

passaram a ser considerados fraudulentos e esses cliques devem ser marcados como “inválidos” (TUZHILIN, 2006). Conforme a forma clássica de lidar com a fraude de clique pelas redes de anúncios (CHO et al, 2015), (INTRIERI, 2014), (KITTS et al, 2013), (HADDADI, 2010), (GOODMAN, 2008), (ANUPAM et al., 1999), quando cliques são classificados como inválidos, o usuário que fez a requisição ainda é redirecionado para o site do anunciante, mas este não é cobrado pelo clique.

Em contrapartida, um clique fraudulento é aquele que foi realizado com intenção maliciosa, e a fraude de clique pode ser simplesmente definida como a prática de realizar cliques fraudulentos (COSTA; DE QUEIROZ; CAVALCANTI, 2012). Como a intenção por trás de um clique em um anúncio online é algo subjetivo, que só pode ser defendida pelo usuário realizando o clique – ou pelo autor do software que realiza cliques de maneira automática –, nos sistemas de redes de anúncios atuais, é impossível ter certeza absoluta de que qualquer clique seja é fraudulento (TUZHILIN, 2006).

Nas abordagens atualmente encontradas no mercado e entre as soluções na academia que encontramos através desta pesquisa (CHO et al, 2015), (OENTARYO et al, 2014), (KITTS et al, 2013), (HADDADI, 2010), (METWALLY et al, 2005), (JUELS; STAMM; JAKOBSSON, 2006), (METWALLY; et al, 2007), as redes de anúncios buscam armazenar sinais e evidências que serão posteriormente avaliadas, normalmente por heurísticas que buscam similaridades em padrões de cliques, caso haja suspeita de uma ação maliciosa. Assim, cliques suspeitos podem acabar sendo marcados como inválidos após esta análise, mas a garantia probabilística acerca da completude do número de cliques fraudulentos que serão detectados e classificados como inválidos depende de inúmeros fatores matemáticos que variam de caso a caso, incluindo o tipo de fraude que o algoritmo de detecção se propõe a combater (TUZHILIN, 2006).

Tipicamente, como já explicado anteriormente, quando cliques suspeitos são marcados como inválidos, o usuário que executou o clique ainda é redirecionado ao site do anunciante. Essa escolha de prosseguir com o redirecionamento oferece dois benefícios para uma abordagem exclusiva de detecção:

- Um fraudador não recebe nenhum retorno se foi detectado como tal;

- Se um clique suspeito na verdade for legítimo, ou seja, se a avaliação futura de detecção retornar um falso positivo, então a experiência do usuário com a rede de anúncios e com o anúncio em si não é impactada negativamente.

Quanto a esse último aspecto, um número alto de falsos positivos é algo pouco lucrativo para o publicador, uma vez que os mesmos podem implicar receitas para os anunciantes. É de vital importância que uma rede de anúncios realize todos os esforços possíveis para minimizar o número de falsos positivos e assim balancear o aumento do lucro dos anunciantes com a prospecção de novos publicadores e relacionamento de qualidade com eles. Nesse âmbito, é importante notar que nem todos os cliques marcados como inválidos são necessariamente fraudulentos. Por exemplo, muitos cliques são marcados como inválidos devido a ações de usuário – como cliques duplos ocasionados por um usuário que teve temporariamente problemas de conexão, ou por outras razões técnicas.

Há duas fontes de cliques inválidos que ocorrem por intenção maliciosa, a saber, cliques originados por concorrentes de anunciantes e cliques de publicadores desonestos. Uma vez que os publicadores lucram com os eventos de clique nos anúncios exibidos em seus sites, é possível observar um incentivo para que publicadores desonestos aumentem o número de cliques que seus sites (CHO et al, 2015), de maneira a gerar receita através de PPC. Além disso, concorrentes de anunciantes também acabam se sentindo incentivados a simularem cliques nos anúncios de seus concorrentes com o objetivo de esgotar o orçamento dos departamentos de

marketing desses últimos. Como se pode perceber, em ambos os casos, a fraude de clique resulta em má reputação para as redes de anúncios, e existem diversos casos de pagamento de multas para anunciantes (KITTS et al, 2013), de maneira que a fraude de clique coloca em risco toda a indústria de anúncios pela Internet (OENTARYO et al, 2014).

A fraude de clique tem sido uma preocupação para representantes de anúncios desde a sua concepção (ZELLER JR, 2004). Os números envolvidos na mesma são difíceis de quantificar e existem diversas formas de se estimar a proporção de cliques falsos, que chegam a variar de 10% a 50%, dependendo da forma como a proporção de cliques inválidos é medida. De acordo com um estudo da MarketingExperiments.com (RESEARCHANDMARKETS, 2014), 29,5% dos cliques em três campanhas experimentais do Google eram fraudulentos. Mesmo com números tão expressivos, as empresas de busca e muitos dos seus clientes vêm argumentando que o problema em suas redes está sob controle. Entretanto, alguns observadores do mercado de cliques online (DIRECT MARKETING ASSOCIATION, 2015), (IAB, 2015), acreditam que a fraude de clique traz prejuízos da ordem de bilhões de dólares e possuem potencial para destruir a indústria inteira (CHO et al, 2015). Independentemente do número exato, a fraude de clique está impregnada no negócio de anúncios pela Internet e, embora as ferramentas de busca procurem se defender de diferentes maneiras, os fraudadores tornam-se cada vez mais sofisticados e os programas utilizados para automatização da fraude são cada vez mais complexos, disfarçando, inclusive, a origem dos cliques.

Por partes não contratadas

Uma forma alternativa de realização da fraude de clique é através de partes não contratadas, ou seja, pessoas ou organizações que não fazem parte do acordo pré-estabelecido de pagamento por clique. Definir regras de punição para esse tipo de fraude é ainda mais difícil,

pois o criminoso, na maioria das vezes, não pode ser processado por quebra de contrato nem criminalmente acusado de fraude. Exemplos de partes não contratadas são (PEARCE, 2014):

- Concorrentes de anunciantes: conforme já descrito anteriormente, essas partes podem querer prejudicar um concorrente do mesmo mercado, ao clicar nos seus anúncios. Os criminosos, nesses casos, não lucram diretamente, mas forçam seus concorrentes a pagar por cliques irrelevantes com o objetivo de enfraquecê-los ou eliminá-los economicamente;

- Concorrentes de publicadores: essas partes podem querer difamar um publicador. Esta fraude é feita de maneira a parecer que o publicador está clicando nos seus próprios anúncios – e a rede de anúncios pode querer terminar a relação. Alguns publicadores têm apenas os anúncios como fonte de renda, e um ataque como esse pode simplesmente buscar retirar uma empresa do mercado;

- "Amigos" dos publicadores: já aprendemos que um publicador ganha dinheiro quando os anúncios são clicados do seu site. Então, um "amigo" do publicador, como um fã, familiar, ou amigo pessoal acaba clicando em diversos anúncios para "ajudar". Entretanto, isto pode acabar sendo ruim para o publicador e o mesmo (ao invés do "amigo") pode acabar sendo acusado de fraude de clique;

- Outras partes maliciosas: além de vandalismo, existe uma infinidade de razões para prejudicar tanto um anunciante como um publicador, mesmo que o criminoso não tenha nada a ganhar com o ato. Os motivos podem variar de vinganças políticas a pessoais. Normalmente, tais casos são os mais difíceis de lidar, uma vez que é complicado rastrear o acusado e, caso ele venha a ser encontrado, existem poucas providências legais que podem ser efetivamente tomadas contra o mesmo.

Por mais que as redes de anúncios busquem combater todas as fraudes possíveis, levando em consideração os cenários acima, é muito difícil saber quais cliques são legítimos.

Além disso, com a exceção das fraudes cometidas pelos publicadores, é difícil definir quem deve pagar quando uma fraude cometida no passado é descoberta. Se levarmos em consideração o objetivo de parceria comercial de longo prazo com credibilidade de mercado, é de certa forma prejudicial para os publicadores ter que reembolsar cliques cujas fraudes, muitas vezes, não são realmente sua responsabilidade. Compreensivelmente, os anunciantes são irredutíveis e se negam a arcar por custos relacionados a cliques falsos.

Conforme Tuzhilin (2006), a forma mais simples de fraude de clique é quando um usuário, ao iniciar um pequeno empreendimento na *web*, se torna um publicador de anúncios e passa a clicar nos *links* dos anúncios que aparecem em seu site para gerar receita. Por vezes o número de cliques e o seu valor são tão pequenos que a fraude não é nem detectada e, quando detectados, alguns os publicadores argumentam que alguns de tais cliques foram acidentais, problema conhecido como "clique duplo" (BINDU, 2015). No entanto, o principal gargalo da fraude de clique é realmente quando a mesma acontece em larga escala e de maneira automatizada, por *bots*, o que chega a constituir até 85% do tráfego de uma campanha mensal típica de pagamento por clique (RESEARCHANDMARKETS, 2013), (PEARCE et al., 2014), (SAGIN, 2015), (CICILIANO, 2015).

Os envolvidos em fraude de clique em larga escala normalmente rodam *scripts* para fingir que um usuário humano está clicando nos anúncios (OENTARYO et al, 2014). Obviamente, uma quantidade enorme de cliques originados de um único computador ou de um pequeno grupo de computadores, ou ainda de uma única região geográfica, parecerá extremamente suspeito para rede de anúncios e anunciantes. É o que Oentaryo et al (2014) denominam de "similaridades". Cliques que possuem origem em um computador que reconhecidamente pertence ao publicador também parecerão suspeitos àqueles que vigiam contra fraude de clique. Dessa forma, uma pessoa que tentar realizar fraude em larga escala sozinha, certamente estará correndo grande risco de ser descoberta.

A fraude que transforma o tráfego real de usuários em cliques inválidos dificilmente é detectada, mesmo quando são aplicados métodos de filtragem de padrões de endereços IP repetidos (CHO et al, 2015). Tal ataque pode ser escondido dos usuários ao se usar *iFrames* (ANA; WHITEOPS, 2015) de tamanho 0 para acessar anúncios por meio de *javascript*. Um *iFrame* é um recurso muito utilizado nas páginas da Internet, que consiste na inserção de páginas *web* dentro de outras páginas. Elas se diferenciam dos *frames* pois esses são divisões da mesma página, ou seja, os frames subdividem a mesma página em seções – já os *iFrames* são páginas dentro de páginas. Dessa forma, utilizando essa tecnologia, é possível camuflar a navegação dos anunciantes e das redes de anúncio, mesmo que os mesmos possuam rastreadores ou indexadores do tipo *web crawlers* (KITTS et al, 2013). Já que os usuários estão acessando uma página *web* legítima e realizando a fraude por meio de *iFrames* embutidos de tamanho 0, que na verdade são transparentes e invisíveis para eles, nem os rastreadores nem mesmo os usuários conseguirão detectar que estarão recebendo nos navegadores um *script* que realiza a fraude.

A técnica acima e outras técnicas que usam visitantes reais podem ser combinadas com o chamado tráfego incentivado, em que membros de determinados sites do tipo Paid to Read (PTR) (INTRIERI, 2014) recebem pequenas quantidades de dinheiro para, centenas de vezes durante um dia, simplesmente visitar um *site* (concorrente ou não), clicar em uma palavra-chave ou em um resultado de pesquisa (MANN, 2006). Naturalmente, alguns donos de sites PTR também são membros de ferramentas PPC, e podem enviar anúncios por *e-mail* para atrair usuários que realizam as pesquisas, ao mesmo tempo em que enviam pequenos anúncios para os que não realizam. Isso acontece porque, muitas vezes, a cobrança por cliques é a única fonte de renda do site.

Existem também formas automatizadas para se realizar a fraude de clique em larga escala. Organizações de segurança e a polícia cada vez mais têm tratado essas formas da fraude

em larga escala como crime organizado (MOSCARITOLO, 2011). Em sua forma mais simples, podemos imaginar uma rede de computadores geograficamente dispersos com conexões independentes à Internet, mas que se comunicam por meio de uma *botnet* (KITTS et al, 2013) ou por um mecanismo semelhante para realizar a fraude de clique.

Muitas vezes, aliado às *botnets,* os fraudadores e as redes de crime organizado utilizam de códigos maliciosos e outras formas de *malware*, como *trojan horses* (BINDU, 2015) para transformar o computador dos usuários em uma espécie de "computador zumbi" para, esporadicamente, realizar ações que lhe beneficiam. Encontrar similaridades em casos como esse é complicado para todas as partes interessadas no negócio e lidar com casos envolvendo redes de pessoas espalhadas em diferentes países é muito difícil para anunciantes, redes de anúncios e autoridades.

Existe ainda outra forma de fraude, denominada *impression fraud* ou fraude de impressão (IMMORLICA, 2005), em tradução livre, que acontece quando impressões de anúncio geradas de maneira fraudulenta afetam a conta de um anunciante. Existem redes que utilizam modelos através dos quais o anunciante pode ser penalizado se tiver um nível de aceitação (cliques) muito baixo em uma determinada palavra-chave. A fraude consiste em realizar inúmeras pesquisas sobre uma mesma palavra-chave, inclusive de maneira automatizada, sem nunca se clicar no anúncio. Tal anúncio, com o passar do tempo, perde relevância no *ranking* de impressões, e é desabilitado, fazendo com que os anúncios mais caros (aqueles que aparecem nas primeiras páginas de pesquisa) não sejam exibidos, em detrimento do anúncio mais barato, que o fraudador busca beneficiar, que passa a aparecer nas primeiras páginas quando a mesma palavra-chave for pesquisada.

3.2 Implicações legais da Fraude de Clique

Os problemas legais em torno da fraude de clique têm se proliferado desde os primórdios do PPC. Normalmente, quando algum fraudador é descoberto, a sua conta de PPC é desabilitada junto à rede de anúncio e, como resultado, ele perde os pagamentos pendentes e potenciais. Entretanto, dependendo dos valores envolvidos, o caso pode acabar sendo direcionado para a justiça.

Quando alguém concorda com os termos de utilização de serviço de uma empresa de PPC, essa pessoa está assinando um contrato. Através desse acordo, as partes são obrigadas a cumprir as regras e condições estabelecidas, e evitar quaisquer comportamentos proibidos. Se uma das partes comete uma fraude, abre-se uma brecha no contrato e, consequentemente, um precedente para disputa legal. Além disso, dependendo de certas circunstâncias, a violação consciente do contrato pode implicar em penalidades criminais, como no caso de flagrante de delito no exercício da fraude de clique. Este seria o caso da detecção da fraude por meio da utilização, por parte da autoridade policial, de programas que exponham a perpetração. Os resultados dessa ação podem ser usados como evidências contra o fraudador.

Em algumas jurisdições internacionais, como a do Estado da Califórnia, cometer fraude de clique é considerado um ato criminoso (BINDU, 2015). Em outras palavras, isso significa que qualquer pessoa que seja flagrada no ato de realizar a fraude de clique pode ser processada e condenada a uma pena de prisão. A duração da pena vai variar com a soma obtida ilegalmente: quanto mais dinheiro o fraudador conseguir, maior será a sentença. Normalmente, as possibilidades de sucesso em um caso de fraude de clique estão associadas ao uso de *scripts* de computador para automatizar o clique nos anúncios, pois, como já falamos anteriormente, a fraude de clique manual tende a deixar muitas similaridades em sua ocorrência. Assim, quando

apresentadas evidências geradas por programas de rastreamento, tais como *web crawlers*, as cortes dificilmente olharão de maneira favorável ao réu. Tais evidências são consideradas indicações fortíssimas de que o réu cometeu a fraude premeditadamente.

Semelhantemente, quando se descobre que a obtenção de grandes lucros decorrentes de uma campanha de PPC deve-se ao uso de fraude, os anunciantes provavelmente iniciarão um processo civil para recuperar os pagamentos impróprios. É possível que as redes de anúncio, por uma questão de reputação, também busquem ressarcimento, muito embora elas tipicamente sejam a parte que mais lucram com a fraude. O caso pode resultar em um julgamento extenso, especialmente se as partes que processam também buscarem sanções punitivas de caráter criminal.

Existe ainda a perspectiva do anunciante na fraude de clique, que muitas vezes é ignorada na literatura sobre o assunto. O anunciante pode processar a rede de anúncios ou a empresa de PPC, no caso de uma contração direta, para recuperar os pagamentos realizados como resultado de uma fraude de clique por parte de terceiros. Foi o que aconteceu no caso do acordo de 90 milhões de dólares entre o *Google* e a *Lane's Gifts & Collectibles*, depois que a essa última provou ter pagado ao site milhões de dólares por cliques de origem fraudulenta. Uma empresa gigante do *marketing online* como o Google certamente tem uma motivação em perseguir agressivamente na justiça aqueles que cometem a fraude de clique no seu ambiente de anúncios, já que pode acabar sendo réu em processo que tenha como vítima seus clientes, os anunciantes. Na prática, o Google tem se mostrado relutante em disponibilizar informações específicas sobre o tráfego de dados associado aos cliques (SAGIN, 2015). Isso certamente deve ser uma preocupação para os anunciantes, que podem estar sendo cobrados arbitrariamente sem receber informações devidas, acerca do detalhamento da cobrança nem sobre como proteger suas campanhas no futuro. Normalmente, as cobranças envolvem grandes quantias de

dinheiro e são realizadas diretamente na fatura de cartões de crédito dos anunciantes, com uma simples linha de identificação de cobrança (DAVIS, 2005).

O problema com as soluções baseadas na colaboração de terceiros está no fato de que tais soluções podem enxergar apenas uma parte do tráfego na rede inteira. Consequentemente, elas dificilmente identificarão a amplitude total de padrões de fraude de clique, afetando muitos anunciantes ao mesmo tempo. Além disso, devido a limitações na parte do tráfego à qual têm acesso, se comparadas com ferramentas de busca, os terceiros podem ter dificuldades para julgar um determinado tráfego como fraudulento.

3.2.1 Teoria dos Contratos e a Fraude de Clique

De acordo com Maria Helena Diniz (2015), pode-se definir "contrato" como o "acordo de duas ou mais vontades, na conformidade da ordem jurídica, destinado a estabelecer uma regulamentação de interesses entre as partes". Juridicamente falando, podemos dizer que um contrato é um negócio jurídico que gera obrigações para as partes. Na fraude de clique envolvendo participantes de uma rede de anúncios, a teoria dos contratos desempenha papel fundamental quanto à definição de mecanismos de reembolso e de cursos de ação cíveis, no caso de condutas fraudulentas.

Os contratos para utilização de serviços de PPC são de natureza virtual, firmados através da aceitação de termos exibidos em uma página de Internet e por meio da interação com o potencial usuário do serviço, seja ele o publicador de anúncios, o anunciante, e a rede de anúncios. No Brasil, ainda não há legislação específica para os contratos virtuais, de maneira que é necessário utilizar as leis constantes no nosso ordenamento jurídico e as adaptar para o universo dos anúncios na Internet, procurando a identificar possíveis lacunas. É interessante

também analisar a identificação das partes no universo virtual, tanto no momento da efetivação do contrato como em uma eventual lide.

O momento da efetivação de um contrato virtual depende de alguns fatores comuns aos contratos regulares (VENOSA, 2014): vontade das partes de modificar deveres e obrigações; terem as partes pressupostos jurídicos para exercer o contrato; o proponente, após a proposição, deve receber a aceitação do aceitante; as partes devem se guiar sobre os princípios de boa-fé e probidade, respeitando a função social do contrato. Os contratos virtuais podem ser considerados contratos nos quais as partes estão presentes por ocasião de comércio na Internet, já que trocam informações em tempo real. Reforçando esse entendimento, podemos afirmar que o código do consumidor se aplica aos contratos virtuais de maneira análoga, de maneira que ele é válido desde que não seja contrário ao direito (VENOSA, 2014).

Uma das questões principais para se aferir a veracidade, legitimidade e regularidade do vínculo obrigacional em um contrato virtual consiste na identificação das partes envolvidas na operação remota (DIAS, 2014). Em geral, as redes de anúncios são empresas de grande porte que normalmente possuem escritórios fixos, sendo, portanto, de fácil localização para uma eventual lide. No entanto, existem sites que não possuem lojas ou escritórios fixos, sendo somente virtuais. Nesse caso, o interessado em realizar qualquer tipo de negociação na Internet precisa tomar cuidados para garantir que suas informações chegarão efetivamente aos proprietários da loja virtual, através da utilização de um certificado digital (PEREIRA, 2011).

A grande implicação da teoria dos contratos na fraude de clique relaciona-se com a possibilidade de se encerrar e de se executar um contrato por ocasião que o torne anulável. Um contrato anulável é aquele que foi celebrado por incapaz, ou que tenha sido viciado por um erro, dolo, coação, estado de perigo, lesão ou fraude contra credores (MIRANDA, 2009). Nesse sentido, chamamos atenção para o dolo e para fraude contra credores, verificáveis comumente

em casos de fraude de clique. O dolo significa o uso de astúcia com o objetivo de enganar. Assenta-se na má fé e na indução ao erro. As ações dolosas têm por objetivo o não cumprimento da promessa. O agente visa obter o resultado ilícito, que seja contrário ao direito firmado por acordo. Ou seja, o objetivo seria conduzir a outra parte ao erro. Verifica-se a possibilidade de dolo na fraude de clique com posterior execução e anulação do contrato quando houver, por exemplo, má fé por parte da rede de anúncios na definição dos termos do contrato, possivelmente induzindo os anunciantes ou publicadores à fraude.

No entanto, a forma mais natural de anulação e execução de contratos decorrente de fraude de clique estaria associada com a categoria de fraude contra credores. Essa ocorreria no ato de falseamento ou de ocultação da verdade com intenção de prejudicar ou enganar a outra parte. Nesse caso, a manobra seria realizada com o objetivo de fraudar terceiros, com o intuito de fugir à incidência da lei e aos seus efeitos. A fraude contra credores é o artifício malicioso empregado para prejudicar terceiros despidos de garantias reais. Para caracterizar uma fraude, basta que o devedor tenha consciência de que seu ato irá prejudicar ou trazer prejuízos a um terceiro. Os atos viciados por fraude são anuláveis por meio da Ação Pauliana (MIRANDA, 2009), na qual os bens transferidos de maneira fraudulenta retornam ao patrimônio do credor. Os casos mais comuns da fraude de clique envolvendo contratantes se encaixam nessa categoria.

Vale ressaltar mais uma vez que a fraude de clique não se configura apenas quando há contrato entre o fraudador e o fraudado. É possível, por exemplo, que o concorrente de determinada organização que participe de um esquema de PPC realize a fraude de clique contra uma rede da qual ele não faça parte para gerar prejuízos para o seu concorrente, conforme detalhamos anteriormente.

3.3 Casos legais

Disputas sobre fraude de clique resultaram em um grande número de processos. Em um determinado caso, descrito por Davis (2005), o Google, que agia tanto como anunciante quanto como rede de anúncios, venceu um processo contra uma empresa do Texas chamada de *Auction Experts*, que agia como publicador. Na ação, o Google acusava a *Auction Experts* de pagar pessoas para clicar nos anúncios que apareciam no próprio site, causando um prejuízo de cinquenta mil dólares aos anunciantes. Mesmo com os esforços das redes para parar este tipo de fraude, a verdade é que os publicadores desconfiam dos reais motivos dessas redes, já que elas também lucram com a fraude.

Em julho de 2005, o Yahoo! entrou em acordo em um processo de queixa no qual era acusado de não ter tomado precauções suficientes para prevenir a fraude de clique. O Yahoo! teve que pagar 4,5 milhões de dólares em taxas legais para os queixosos, e concordou em datar os valores do acordo para 2004 (RYAN, 2006). No caso mais conhecido na literatura da área, em março de 2006, o Google acertou, por 90 milhões de dólares (LIEDTKE, 2006), um acordo similar com a *Lane's Gifts & Collectibles*, caso detalhado por Tuzhilin (2006).

Em 2004, um morador da Califórnia, chamado Michael Anthony Bradley, criou o chamado Google Clique (NARAINE, 2004), um programa que, de acordo com o criador, tornaria possível que o Google fosse fraudado em milhões de dólares. De acordo com autoridades, Michael Anthony Bradley foi preso e declarado culpado por chantagem, condenado a pagar cento e cinquenta mil dólares e forçado a entregar o programa à empresa. Acredita-se que essa foi a primeira prisão por fraude de clique.

As acusações foram retiradas em 22 de novembro de 2006. De acordo com Elgin (2006), em artigo publicado na revista Business Week, o Google não quis cooperar com o

processo, uma vez que seria obrigado a expor publicamente suas técnicas de detecção de fraude de clique e a admitir publicamente que lucra com cliques fraudulentos, através de técnicas como essas e talvez como outras estabelecidas.

De acordo com Ana e White Ops Inc (2015), cada vez mais a audiência dos anúncios online não é composta por humanos. Um estado realizado pela *Association of National Advertisers* em 2015 e incorporou bilhões de anúncios com código para determinar que tipos de usuário estava visualizando: humano ou não humano. Cerca de 11% dos anúncios foram acessados por *software*, e não por pessoas. O estudo, denominado *The Bot Baseline: Fraud In Digital Advertising* aponta para um prejuízo de cerca de $ 6,3 bilhões de dólares em tráfego fraudulento em 2015 (ANA; WHITEOPS, 2015).

Provar a fraude de clique pode ser muito complexo, uma vez que é difícil saber a parte interessada por trás de um computador e quais as suas intenções. A estratégia que as redes de anúncios têm adotado até o momento é a de identificar quais cliques são potencialmente fraudulentos e não cobrar os anunciantes por eles. Existem diversos métodos sofisticados de detecção, mas nenhum é livre de falhas. O relatório de Alexander Tuzhilin (2006), produzido como parte do acordo entre *Google* e *The Lane`s Gifts*, possui uma discussão detalhada sobre estes problemas. Em particular, de acordo com Tuzhilin (2006),

> o problema fundamental de cliques inválidos, ou seja, fraudulentos, é que não há uma definição conceitual de cliques inválidos que possa ser operacionalizada, com a exceção de casos óbvios, e que uma definição operacional não pode ser totalmente revelada ao público em geral, uma vez que possibilitará que usuários maliciosos obtenham vantagem através do uso maciço de fraude de clique. Entretanto, se não revelada, anunciantes não podem verificar a razão e, assim, opor-se à cobrança de determinados cliques.

Existe, nos Estados Unidos, um lobby considerável da indústria de PPC para que leis mais rígidas sejam definidas para lidar com esse problema (BUCHDAHL, 2012), (CICILIANO, 2015). A esperança é que tais leis venham descrever casos que não podem ser especificados como contratos clássicos. Um grande número de empresas tem desenvolvendo soluções viáveis para a identificação de fraude de clique através de relações intermediárias com redes de anúncio. Tais soluções subdividem-se em duas categorias:

a) **Análise judicial do histórico dos servidores de anunciantes**: requer uma investigação profunda da fonte do tráfego de dados e do seu comportamento. A ideia é desenvolver arquivos de log, para análise, e comparar os dados disponíveis nos servidores com esses arquivos. O problema com esta abordagem é que ela confia na idoneidade das ferramentas de busca, que têm a responsabilidade de identificar a fraude.

b) **Confirmação de terceiros**: se um *site* externo oferece soluções para "etiquetar" os anúncios através da colocação de imagens, ou de javascript, nas páginas web dos anunciantes para as quais os visitantes são direcionados, o visitante recebe um *cookie* ao visitar tais páginas e a informação do visitante vai ser coletada, armazenada em um banco de dados e disponibilizada para *download*. Dessa forma, ao se analisar as melhores ofertas, é possível identificar um conjunto de cliques suspeitos, e, por causa da "etiqueta", expor as razões para a desconfiança. Essas informações, ao serem associadas com os arquivos de log dos anunciantes, formam um conjunto de evidências mais convincente, que pode ser apresentado à rede de anúncios para verificação. No entanto, as soluções baseadas na colaboração podem enxergar apenas uma parte do tráfego na rede inteira. Consequentemente, elas dificilmente identificarão a amplitude total de padrões de fraude de clique afetando muitos anunciantes ao mesmo tempo. Além disso, devido a limitações na parte do tráfego ao qual têm acesso, os terceiros podem ter dificuldades para julgar um determinado tráfego como fraudulento.

3.4 Considerações do Capítulo

Existem várias abordagens para o combate de fraudes em negócios de anúncios *online*. Neste Capítulo 3, foram analisadas algumas das diversas formas existentes de fraudes e abusos nesse domínio de mercado e um foco especial foi dado à chamada fraude de clique, tendo-se estudado um apanhado acadêmico que incluiu diversas formas para a sua realização e métodos para o seu combate. Essa fraude foi escolhida por apresentar dificuldades técnicas no que diz respeito à sua prevenção. Obviamente, é impossível impedir que alguém realize cliques sem a intenção de compra em um anúncio específico mas acreditamos ser, com a tecnologia atualmente disponível, possível adotar alguns passos para impedir o acontecimento da fraude de clique em larga escala e automatizada.

Além disso, a seção 3.2.1 também apresentou um apanhado de implicações legais que a fraude de clique tem para o direito, especialmente o brasileiro, levando em consideração a Lei No. 12.737, de 30 de novembro de 2012, e a Lei No. 12.965, 23 de abril de 2014 (ARAS, 2009). Foi realizado neste Capítulo uma análise detalhada dos textos das leis e suas implicações às particularidades da fraude de clique, o que nos fez perceber a necessidade de complementar a análise com o entendimento da teoria dos contratos, apresentada na seção 3.2.2.

Para finalizar, apresentamos um número de casos legais envolvendo a fraude de clique na seção 3.3. A maioria desses casos se deu nos Estados Unidos, muito embora Lobo (2011) relate que crimes cibernéticos no Brasil já deem prejuízos registrados da ordem de 15 bilhões de dólares.

4 Completely Automated Public Turing Test to Tell Humans and Computers Apart

Nos últimos anos, um número crescente de serviços na Internet disponíveis para o público em geral tem tentado prevenir o abuso de programas automáticos ao demandarem dos seus usuários a resolução de um desafio de diferenciação entre usuários humanos e máquinas, inspirado no teste proposto por Turing (1950). Esse tipo de teste hoje é conhecido como CAPTCHA, uma sigla para *Completely Automated Public Turing test to tell Computers and Humans Apart* (AHN et al, 2004), e tipicamente precisa ser respondido pelos usuários antes deles começarem a usar o serviço ou de se registrarem no mesmo. Em teoria, os testes são fáceis de serem gerados, e difíceis, para usuários não humanos, de serem resolvidos. Todos os CAPTCHA possuem alguma informação secreta que é conhecida pelo desafiante, mas não pelo usuário ou agente desafiado.

O termo CAPTCHA foi firmado em 2000 por Luis von Ahn, Manuel Blum, Nicholas J. Hopper (todos da universidade Carnegie-Mellon) e por John Langford (IBM) (AHN et al., 2004) e envolve, basicamente, um computador, que age como servidor e que pede que um cliente realize uma interação com o objetivo de responder a um teste. O objetivo do teste é determinar que todo cliente que aponte uma solução correta seja humano.

Um CAPTCHA pode ser visto também como um programa que faz um teste do tipo "desafio-resposta" com o objetivo de determinar se seu usuário é um humano ou um computador. Um tipo comum de CAPTCHA requer que o usuário identifique as letras de uma imagem colorida e distorcida, às vezes com a adição de uma sequência obscurecida das letras ou dos dígitos que aparecem na tela. Desafios CAPTCHA são usados por muitos *websites* como uma forma de prevenir o abuso de *bots* ou de outros programas automáticos normalmente escritos para gerar *spam*.

Na verdade, de acordo com Jakobsson e Razman (2008), *spam* pode ser definido como um artefato que não produz nenhum valor, utilidade ou benefício conforme esperado por um ou mais participantes associados ao artefato. Desta forma, um *e-mail* é considerado *spam* quando o receptor não consegue derivar nenhum valor apreciável, útil ou benéfico do *e-mail*. Sob essa perspectiva, podemos classificar cada uma das fraudes em anúncios online estudadas no Capítulo anterior como formas diferentes de *spam*. CAPTCHA surgem, dessa forma, como uma forma de combate ao *spam*.

De acordo com Ahn et al (2008), um CAPTCHA também pode ser descrito como um "teste reverso de Turing", já que o teste é administrado por um computador, em contraste ao teste padrão de Turing, que é administrado por um ser humano. Em teoria, *bots* não conseguem ler textos distorcidos tão bem quanto humanos, de maneira que os CAPTCHA podem impedir que *bots* naveguem em *sites* protegidos. Na prática, um CAPTCHA introduz um problema cuja solução só pode ser automatizada utilizando técnicas de inteligência artificial. Um sistema de CAPTCHA é uma forma de gerar, de maneira automatizada, novos desafios que:

- Os computadores atuais sejam incapazes de resolver;
- A maioria dos humanos possa resolver;
- Não garantam que o atacante desconhece o desafio. Por exemplo, embora um *checkbox* do tipo "clique aqui se você não é um robô" possa servir para distinguir um humano de computadores, ele sozinho não pode ser caracterizado como um CAPTCHA porque confia no fato que um potencial fraudador não precisou de grandes esforços para acertar aquela pergunta específica.

4.1 Histórico

A dificuldade em distinguir humanos de computadores fingindo ser humanos foi inicialmente discutida em 1950, quando Alan Turing descreveu o teste (TURING, 1950). A primeira discussão sobre usar testes automáticos que realizam essa diferenciação com o objetivo de controlar acesso a serviços *web* que se tem registro é o artigo de Naor (1996) do Instituto de Ciência de Weizmann, entitulado "*Verification of a Human in the Loop, or Identification via the Turing Test*". Os primeiros CAPTCHA que se tem registro foram desenvolvidos em 1997, para o site de busca Altavista, com o objetivo de impedir que *bots* adicionassem páginas web para a ferramenta de busca (ARUNA; KANCHANA, 2015). De maneira a fazer com que as imagens não fossem detectadas por um *software* OCR (*Optical Character Recognition*), a equipe simulou situações que manuais de aparelhos de *scanner* diziam resultar em mal funcionamento para gerar imagens distorcidas, com má qualidade e com ruído, o que resultou na invenção de múltiplos exemplos de CAPTCHA, incluindo os que viriam a ser os primeiros a serem usados publicamente, inicialmente no *Yahoo!* quando esse adquiriu o Altavista.

Os desafios CAPTCHA são usados para prevenir que *softwares* automatizados realizem ações que degradem a qualidade do serviço de um dado sistema por causa do abuso no uso de recursos. Na maioria das vezes, CAPTCHA são usados como uma resposta para invasões por interesses comerciais, mas eles podem ser utilizados em outras aplicações, como a proteção de sistemas vulneráveis a *spam*, como *webmail*, ou de forma a evitar postagem automática em *blogs* ou fóruns, de maneira a impedir os anúncios comerciais não-autorizados, vandalismo ou outras formas de perturbação (MOTOYAMA et al, 2010). Além disso, eles podem ser usados em *sites* de estatísticas, para limitar o uso automatizado de um serviço em detrimento dos usuários humanos quando uma determinada taxa de uso for atingida (ALSUHIBANY, 2011). Em tais casos, os CAPTCHA serão aplicados para garantir políticas de uso de aplicações automáticas, definidas por um administrador, quando métricas de uso

excederem um determinado limiar. Algumas das aplicações de CAPTCHA utilizadas em segurança podem ser encontradas abaixo:

- Proteção de registro em *websites*: muitas empresas (como *Yahoo!* e *Microsoft*) permitem o cadastro de *e-mails* de maneira gratuita. Até alguns anos atrás, a maioria destes serviços sofreram um tipo específico de ataque. A solução para tal problema era o uso de CAPTCHA para garantir que apenas humanos conseguissem contas gratuitas (BURSZTEIN et al, 2011). De maneira geral, praticamente todo tipo de serviço gratuito *online* passou ser protegido por CAPTCHA para prevenir o uso abusivo de programas automáticos.

- Prevenção de abuso de comentários em *blogs*: para impedir comentários aleatórios em *blogs* (como "compre em determinado *site*"), normalmente com o objetivo de aumentar a pontuação em *page ranks* de algum *website*, os chamados *comment spams*, é possível utilizar CAPTCHA, através dos quais apenas humanos conseguirão comentar. Assim, não será necessário requerer que os usuários se registrem para poder comentar e apenas os comentários legítimos são registrados.

- Proteção de pesquisas *online*: em novembro de 1999, no *site* http://www.slashdot.org foi realizada uma pesquisa que perguntava qual era a melhor escola de graduação de ciência da computação nos Estados Unidos (BRIAN, 2002). A segurança era definida da seguinte forma: os endereços IP dos votantes eram armazenados de maneira a prevenir que usuários votassem mais de uma vez. Os alunos da Carnegie Mellon University (CMU) escreveram um script e usaram diferentes *proxies* para votar na sua universidade milhares de vezes, o que fez a pontuação da CMU crescer rapidamente. No dia seguinte, os alunos do Massachusetts Institute of Technology (MIT) escreveram seu programa, e a disputa se tornou uma luta entre *bots*. O MIT terminou com 21.156 votos, e a CMU com 21.032. Todas as outras escolas terminaram com menos de 1.000 votos. Levando isso em consideração,

podemos nos perguntar se, atualmente, o resultado de uma pesquisa *online* é confiável. A resposta é: somente se garantir que apenas humanos votarão (COSTA, 2010).

- Prevenção de ataques de dicionário (LING-ZI; YI-CHUN, 2012): os CAPTCHA podem também ser usados para prevenir ataques de dicionário em sistemas de senha. A ideia consiste em impedir a interação de um computador solicitando que, após uma determinada quantidade de tentativas de acesso sem sucesso, o usuário resolva um CAPTCHA.

- *Bots* de busca: algumas vezes, é interessante manter algumas páginas *web* ocultas, ou seja, não indexadas, para impedir que sejam localizadas de maneira fácil por outros. Existe uma forma de configurar uma *tag* HTML para requisitar que os *bots* não leiam a página[2], mas a mesma não passa de uma requisição informal, ou seja, um "pedido" – é impossível impedir tecnicamente que o *bot* leia, de fato, a página. *Web crawlers* de busca automáticos de grandes empresas normalmente respeitam estas *tags*, mas, para realmente garantir que os mesmos não acessarão um *site* na web, CAPTCHA são necessários (NORBU; BHATTARAKOSOL, 2012).

- *Worms* e *spam*: CAPTCHA normalmente oferecem uma maneira plausível contra *worms* (DOYLE, 2003) de *e-mail* e contra *spam*: a ideia é que o servidor (ou o usuário, por meio de definições de preferência) só aceite *e-mails* de humanos (TAMANG; BHATTARAKOSOL, 2012).

[2] A tag HTML, que precisa ser colocada dentro do cabeçalho (ou seja, da tag <head>), é:

```
<head>...
    <meta name="robots" content="noindex" />
...</head>
```

4.2 Tipologia

De acordo com Elson et al (2007), podem-se definir duas classes distintas de CAPTCHA. Os CAPTCHA de Classe I, nos quais a informação secreta é apenas um número aleatório, que é usado por um algoritmo público para alimentar e gerar um desafio, análogo a um criptossistema de chave pública, que consiste em algoritmos de criptografia que requerem duas chaves, sendo uma delas secreta, privada, e a outra delas pública. E os CAPTCHA de Classe II que, além de um número aleatório, usam um banco de dados secreto com alto grau de entropia, análogo a um criptossistema do tipo *one-time-pad*, ou cifra de uso único, que pode ser provada como sendo segura contra qualquer adversário, devido ao "fator trabalho" (SEDIYONO, E.; SANTOSO, 2013), ou seja, a quantidade de computação necessária para vencê-lo é tão alta que torna o ataque impraticável.

Os CAPTCHA de Classe I possuem muitas vantagens, pois os algoritmos para a sua implementação podem ser desenvolvidos em poucas linhas de código, não tornam necessário o armazenamento de nenhuma informação secreta e podem gerar um conjunto praticamente ilimitado de desafios únicos. Entretanto, a funcionalidade principal consiste em um desafio para reconhecer texto distorcido e a diferença da taxa de sucesso entre humanos e máquinas na resolução desse tipo de desafios é muito pequena (MOTOYAMA et al, 2010). Como algoritmos de OCR conseguem reconhecer caracteres do alfabeto quase tão bem quanto humanos, isso fez com que os pesquisadores e desenvolvedores desse tipo de CAPTCHA se motivassem a aumentar a dificuldade e passassem a segmentar os caracteres de uma imagem em regiões distintas, aumentando distorção e ruído (KORAKAKIS et al, 2014). Esse aumento de dificuldade também afetou humanos e, embora experimentos de laboratório tenham relatado que os usuários reais possam resolver os desafios mais difíceis corretamente, a verdade é que o índice de reclamações dos mesmos aumentou consideravelmente e, em um primeiro momento,

os CAPTCHA em páginas comerciais da Internet continuaram a ser mais simples e segmentados, como a Figura 4.1.

Figura 4.1 – CAPTCHA Simples

Fonte: http://www.register.com (2009)

É importante observar que vários proprietários de páginas *web* atualmente disponíveis decidiram que o sucesso de um usuário ao resolver um CAPTCHA não depende somente do fato de ele ser capaz de resolver o desafio, mas também de sua disposição de se esforçar para tal. Mesmo CAPTCHA simples já são suficientes para afastar um número substancial de clientes (BAECHER, 2010).

Por sua vez, os CAPTCHA de Classe II são capazes de superar as fraquezas descritas anteriormente. Como não são restritos a desafios gerados por algoritmos de baixo nível de entropia, eles podem explorar uma quantidade bem maior de habilidades humanas, tais como o reconhecimento de detalhes em imagens fotográficas capturadas do mundo real. Os CAPTCHA desse tipo evidenciam taxas de sucesso muito distintas entre humanos e não humanos, devido basicamente a dois motivos: além de ser muito mais difícil para um computador interpretar uma imagem com alto grau de entropia do que resolver desafios baseados em texto, é possível elaborar desafios baseados em imagem bem menos tediosas para humanos sem perder sua eficiência no bloqueio de programas. Estudaremos os CAPTCHAs de Classe II, clicáveis, na seção 4.4.

4.3 Optical Character Recognition

Esta seção considera um dos grandes problemas interdisciplinares da computação, a identificação de caracteres em um documento. Essa área é chamada de *Optical Character Recognition* (OCR), ou, em tradução livre, reconhecimento óptico de caracteres (ARUNA; KANCHANA, 2015). A estudaremos explicitamente aplicada à tecnologia de CAPTCHA, sob a luz dos trabalhos de Chen et al (2014) e Korakakis et al (2014).

Na indústria e na história do cinema, especialmente em películas de ficção científica, pode-se encontrar diversos exemplares de robôs com capacidades cognitivas e linguísticas. Os androides do cinema podem servir de inspiração para um número significativo de criações computacionais. Para tal, faz sentido tentarmos entender se a tecnologia da informação, algum dia, poderá satisfazer as previsões mais futuristas. Tais perguntas, no entanto, são respondidas de praxe com mais previsões, já que o conhecimento atual torna possível a subdivisão de tendências nas diversas áreas da computação.

Sistemas para reconhecimento de texto impresso por máquina datam desde o fim dos anos 50 e têm sido largamente utilizados em computadores pessoais desde o fim dos anos 80 (MANTAS, 1986). Uma das motivações para existência de *softwares* dessa natureza é o fato de que uma grande parcela de informações históricas nas organizações ainda está na forma não-digital. Uma vez digitalizado, sistemas de recuperação de informação podem utilizar *software* OCR para a localização do material de interesse, e processadores de texto podem ser utilizados para a edição do mesmo.

Em resumo, de acordo com Aruna et al (2015) OCR pode ser definido como o processo de tradução de imagens de textos que foram escritos à mão, digitados ou impressos (normalmente capturados por meio de um *scanner*) para texto editável em formato de máquina. Entretanto, sistemas de OCR não fazem essa conversão de maneira perfeita, pois cometem erros, de modo que a versão eletrônica de um documento dificilmente será idêntica à versão em

papel. Um erro comum, por exemplo, em determinados tipos de fonte, acontece quando o sistema não consegue distinguir a letra ‘e’ da letra ‘c’. “*Voec podc aeabar eom um tcxto quc sc parcec com isso”*. Em tais casos, é mais eficiente digitar o conteúdo inteiro da página no computador do que sair procurando os erros no texto de saída e corrigi-los (CHANDAVALE; SAPKAL; JALNEKAR, 2009). É importante em OCR definir parâmetros para que a condição de acertos seja observada, pois um sistema desses que comete muitos erros não é útil.

O escopo de pesquisa de OCR não incluía inicialmente o reconhecimento de caracteres digitais (XU, 2013), isso é, usando *scanners* e algoritmos, mas estava concentrado nas técnicas ópticas, como as que fazem uso de espelhos e lentes. Entretanto, devido ao fato de que pouquíssimas aplicações atuais fazerem o uso dessas técnicas, o termo foi estendido para também incluir a área de processamento de imagens digitais (KORAKAKIS et al, 2014).

A tecnologia OCR avançou a ponto de tornar os sistemas atuais confiáveis para processar uma grande variedade de documentos impressos por máquina, por exemplo, o reconhecimento correto do alfabeto latino na sua forma digitada é considerado um problema resolvido, atingindo níveis de precisão de até 99% (SHATNAWI; ABDALLAH, 2016) através do reconhecimento por meio de sistemas para reconhecimento de texto manuscrito digitalizado, com dispositivos de PDA (*personal digital assistant*).

Existe uma espécie de variação de OCR em que diversos fatores de inteligência artificial são considerados, a *Intelligent Character Recognition* (ICR) (ARUNA et al, 2015). O reconhecimento de textos cursivos, de faces, de imagens e de aspectos mais cognitivos ainda é uma área ativa de pesquisa da ICR, com taxas de reconhecimento ainda menores do que de textos manuscritos impressos. De acordo com Kalne e Kolhe (2014), devido aos desafios típicos dessa modalidade de reconhecimento, taxas maiores de reconhecimento e níveis de precisão mais ambiciosos só parecem ser possíveis com o uso de informações contextuais.

4.3.1 OCR e CAPTCHA

Na literatura, é possível encontrar algumas abordagens descritas para derrotar CAPTCHA, que buscam explorar vulnerabilidades diferentes, desde a exploração de *bugs* na implementação ou a disposição da arquitetura, que permitem ao atacante simplesmente ignorar o CAPTCHA (KORAKAKIS et al, 2014), melhorar *softwares* de reconhecimento de caracteres (ARUNA, 2015), ou usar recursos humanos para simplesmente processar os testes (DANCHEV, 2014) e variações de algoritmos de OCR para reconhecer imagens em CAPTCHA baseados em imagem (KALNE; KOLHE, 2014).

Para fraudar CAPTCHA baseados em texto, a maior parte dos programas automatizados segue os passos na Figura 4.2 para obter o código:

Figura 4.2: Pseudocódigo para extrair o código CAPTCHA de uma imagem

```
1.    Extração da imagem da página;

2.    Remoção de padrões de desordem do fundo da imagem (background),
utilizando filtros de cor e detectando padrões de linhas finas, tais como
ruído e distorção;

3.    Segmentação (dividir a imagem em regiões contendo um único caractere –
como "NNNN");

4.    Identificação da letra para cada região (exemplo: respondendo à
pergunta, "que letra o conjunto de NNN aparenta formar?").
```

Fonte: (CHANDAVALE et al, 2009)

De acordo com Chen et al (2014), os passos 1, 2 e 4 não são computacionalmente complexos, sendo a segmentação a única parte onde os humanos ainda são melhores. Por exemplo, se a desordem no *background* consiste de formatos semelhantes a letras, e se as letras estão conectadas à desordem, segmentar a imagem se torna praticamente impossível com os *softwares* atuais (ARUNA; KANCHANA, 2015), mas também é muito difícil para humanos. Um bom CAPTCHA textual deve ter seu foco principal de desenvolvimento em segmentação.

4.4 CAPTCHA Clicáveis

Um problema crítico para se construir um CAPTCHA de Classe II é garantir um banco de dados com um conjunto grande de informações que possua certo grau mínimo de entropia. Para resolver este problema, o CAPTCHA ASIRRA (ELSON et al., 2007), por exemplo, propõe um acordo de alinhamento de interesses utilizando um banco de dados externo, aumenta frequentemente de tamanho e é categorizado manualmente.

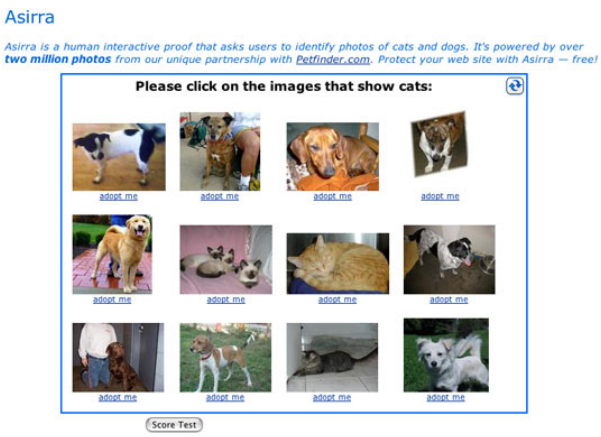
Sob essa perspectiva, as informações que podem ser armazenadas em um banco de dados são fontes potenciais para formarem um CAPTCHA. Analisamos aqui inicialmente dois CAPTCHA clássicos dessa natureza, nas seções 4.4.1 e 4.4.2, que foram estudados mais extensivamente na literatura, um CAPTCHA clicável típico (seção 4.4.1) e um CAPTCHA clicável textual (seção 4.4.2). Em sequência, na seção 4.4.3, apresentamos uma nova tendência no desenvolvimento, uma geração de CAPTCHA que utiliza conceitos adicionais de habilidades e de cognição humana e que poderia ser classificada possivelmente como CAPTCHA de classe III, uma extensão que fazemos do conceito de Elson et al (2007). Mais adiante, no Capítulo 5, estendemos a análise para um outro exemplo de CAPTCHA de classe II apresentaremos o CAPTCHA que adotaremos na nossa abordagem, bem como exploraremos a razão de sua adequação às necessidades da arquitetura que estamos propondo.

4.4.1 ASIRRA – CAPTCHA Clicável Baseado em Imagens

O ASIRRA é um CAPTCHA que solicita aos usuários que classifiquem fotografias em cães ou gatos e a sua força vem de uma parceria com o site Petfinder (ELSON et al, 2007), um serviço de Internet responsável por achar adoções para animais abandonados. O Petfinder tem um banco de dados de cerca de três milhões de imagens de cães e gatos, cada uma categorizada por voluntários humanos que trabalham em abrigos para animais nos Estados

Unidos e Canadá. Um exemplo de desafio ASIRRA pode ser visto na Figura 4.3. O alinhamento de interesses ocorre na medida em que o ASIRRA possui acesso ao banco de dados do Petfinder (que aumenta mensalmente em cerca de 10.000 novas imagens) e utiliza-o para gerar desafios. Em troca, o ASIRRA coloca um *link* com a frase "me adote" próximo a cada foto exibida no CAPTCHA, promovendo a missão do Petfinder de expor animais para possíveis adoções. Tal parceria é mutualmente benéfica e promove um duplo papel social: melhora a segurança em computadores e o de promove o bem-estar animal.

Figura 4.3 – Exemplo de CAPTCHA Clicável (Selecionar os Gatos)



Fonte: http://www.asirra.com (2012)

4.4.2 CAPTCHA clicáveis textuais

Basicamente, a ideia dos CAPTCHA clicáveis textuais é combinar vários desafios em uma grade de desafios clicáveis. A solução é a determinação, através do clique, dos elementos que satisfaçam a um dado requisito. Por exemplo, o usuário pode ser solicitado a identificar as opções que descrevem palavras na língua *inglesa*, adotando naturalmente a premissa que haja palavras de língua inglesa na grade de CAPTCHA clicáveis, quando nem

todas as exibidas fazem sentido, conforme visto na Figura 4.4, no caso, *monster*, *grass* e *nation* (CHOW et al., 2008):

Figura 4.4 – Exemplo de CAPTCHA Clicável Textual

Fonte: Chow et al (2008)

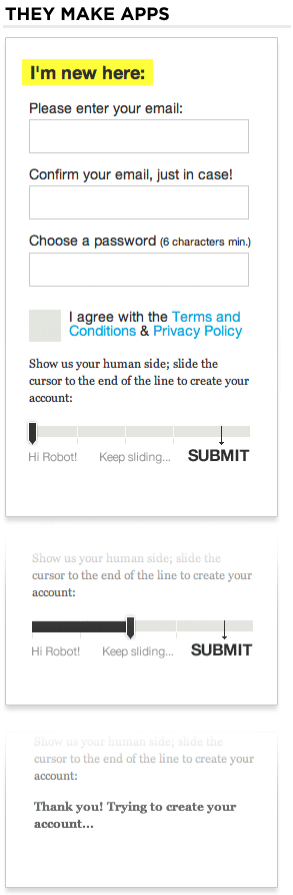
De acordo com Ling-Zi e Yi-Chun (2012), a segurança envolvida por trás de um CAPTCHA clicável textual é baseada na mesma premissa de CAPTCHA textuais tradicionais: a dificuldade de reconhecer caracteres distorcidos. Assim, o desenvolvimento de uma solução confiável de CAPTCHA clicáveis textuais depende da existência de uma forma de gerar desafios textuais fortes. A diferença é que clicáveis textuais podem ser de mais fácil resolução para humanos, pois a tarefa de decidir se uma palavra satisfaz a um determinado requisito normalmente não requer que todas as palavras sejam decifradas.

4.4.3 Outros Exemplos – CAPTCHA de "Classe III"

O senso comum é que os CAPTCHA afastam o uso dos usuários de um site e são vistos pelos mesmos como "chatos" (ALSUHIBANY, 2011). Além disso, pelo que vimos na seção 4.3, os mesmos não são uma garantia que fraudadores estarão definitivamente afastados dos serviços na *web*. Wrobleski (2010) apresenta uma alternativa para os CAPTCHA tradicionais: O formulário do site They Make Apps (http://theymakeapps.com/users/add), que

pode ser encontrado na Figura 4.5, usa um controle deslizante que pede que os novos usuários confirmem que são humanos movendo o controle totalmente para a direita. O usuário do site submete o formulário e aciona os controles de erro, como os botões de submissão fariam.

Figura 4.5 – Controle deslizante para mostrar o "lado humano" dos usuários



Fonte: http://www.theymakeapps.com (2015)

Um estudo aprofundado e detalhado desse tipo de controle, comparando sua eficiência com a de utilização de CAPTCHA, é sugerido como um tópico para estudo futuro no Capítulo 7 desta tese, mas podemos concluir que se trata de um controle seguro pelo seu fator trabalho.

4.5 Considerações do Capítulo

Este Capítulo apresentou um estudo da tecnologia de CAPTCHA, algo que não encontramos em nossos estudos na literatura em língua portuguesa. Foram apresentados conceitos, históricos, aplicações e classificações de diferentes CAPTCHA existentes. Acreditamos que a classificação de CAPTCHA é uma representação satisfatória dos diversos tipos de CAPTCHA que podem ser atualmente encontrados na Internet.

Os CAPTCHA clicáveis são de fácil resolução para usuários humanos. Conforme Bindu (2015), esse tipo de CAPTCHA pode ser respondido por humanos em 99,6% das vezes em um tempo inferior a 10 segundos. De acordo com dados sobre os problemas de visão de máquinas, a probabilidade de os programas acertarem o desafio é de apenas 1/54.000 (ZHU et al, 2014). Apesar de o nível de segurança ser teoricamente o mesmo, os usuários de CAPTCHA clicáveis possuem a tendência de achar a sua experiência de uso bem mais interessante do que os CAPTCHA baseados em texto. Podemos enumerar diversos benefícios adicionais no uso de CAPTCHA clicáveis:

- Humanos podem resolvê-los rapidamente e de maneira correta;

- CAPTCHA de reconhecimento de similaridades como os de Classe II, ao contrário de outros CAPTCHA baseados em imagem, que são abstratos ou subjetivos, são concretos, inofensivos, e não demandam qualquer tipo de conhecimento especializado ou cultural. Na verdade, no máximo, será necessário o conhecimento da língua na qual os desafios são gerados, em casos de CAPTCHA textuais. Isso torna esses CAPTCHA clicáveis menos frustrantes para humanos, havendo até quem os julgue divertidos (AHN et al, 2008);

- No caso de CAPTCHA como o ASIRRA ou o KittenAuth (LEYDEN, 2006), ainda pode promover um benefício social adicional: achar adoção para animais desabrigados.

Podemos listar também algumas desvantagens:

- A maioria dos CAPTCHA é desenvolvido como uma biblioteca de programas integrada a um *site* sem dependências externas. Assim, os CAPTCHA clicáveis devem ser implementados como um serviço *web* centralizado, que gera e verifica desafios sob demanda (O'REILLY, 2015);

- CAPTCHA de classe II têm uma dependência direta da confiabilidade de seu banco de dados. Se um adversário contratar mão-de-obra disponível o suficiente para classificar as entradas em um banco de dados estático, seria muito difícil impedir o ataque;

- Um desafio de um CAPTCHA clicável de classe II requer mais espaço em tela e maior quantidade de dados trafegando na rede que um CAPTCHA tradicional;

- Assim como todos os outros desafios deste tipo, são necessárias adaptações para que os CAPTCHA clicáveis se tornem acessíveis para deficientes visuais. Uma das formas de fazer isso é adicionar uma versão do desafio em áudio.

Pelas razões descritas acima, especialmente no que diz respeito aos números descritos anteriormente e também citados por Zhu et al (2014), o uso de CAPTCHA clicáveis é recomendado como forma de diferenciação entre usuários humanos e programas automatizados na utilização de serviços de redes de anúncios para prevenir a fraude de clique.

Embora não seja uma alternativa sem falhas, neste Capítulo 4 pudemos ver que a utilização de CAPTCHA, mesmo aqueles que não são clicáveis, melhora sensivelmente a segurança dos sites pela diminuição da vulnerabilidade de seus formulários nos programas. A abordagem proposta no Capítulo 5 possui o compromisso com a segurança como principal motivação para prevenir a fraude de clique, pois possui como o objetivo primordial o aumento da credibilidade da indústria de anúncios *online*. Busca também a melhor forma de

disponibilizar um serviço que não seja penoso para o usuário e desestimule o uso. Por causa disso, após o exposto neste Capítulo, a utilização de CAPTCHA clicáveis parece realmente ser a melhor alternativa para satisfazer ambos os objetivos.

5 Abordagem Proposta

Em sua grande maioria, as pesquisas realizadas nesta área investigam a fraude do publicador, já que ela pode ser generalizada para a fraude do anunciante, e discutem a detecção da fraude através de diversos métodos, tais como: a abordagem criptográfica (PEARCE et al, 2014), técnicas de análise de dados (OENTARYO et al, 2014), ferramentas para detecção de fraude (CHO et al, 2015), análise de tráfego (KITTS et al 2013) e algoritmos de força bruta (METWALLY; AGRAWAL; ABBADI, 2007). Entretanto, como já falamos nas Seções 1 e 3, essas abordagens são de detecção, e tratam a fraude depois que ela ocorreu. Como já dito anteriormente, os fraudadores têm desenvolvido redes e coligações de fraudes automatizadas cuja detecção tem se tornado cada vez mais complexas, e a detecção da fraude tem se tornado um problema de resolução cada vez mais difícil.

Por essas razões, a premissa deste trabalho é a proposição de uma abordagem focada na prevenção contra a fraude de clique, de maneira que a detecção se torne uma atividade complementar, isto é, uma checagem realizada esporadicamente pelas redes de anúncios para garantir confiança total nas suas cobranças. A ideia é que a maior parte da segurança já estivesse garantida na prevenção, já que a maior exposição das redes de anúncio hoje é a cliques originários de *bots* de maneira automatizada (SAGIN, 2015).

Dessa forma, uma das ideias iniciais deste trabalho era o preenchimento desta lacuna, através da definição de uma abordagem que seja focada primariamente na prevenção da fraude de clique. Outro objetivo direto é o estudo minucioso da disposição do mercado de anúncios na Internet e a sistemática da fraude de clique, para garantir que a abordagem proposta refletisse as necessidades da indústria. Neste sentido, ambos os objetivos colaboram no sentido de que a proposta de uma abordagem se dará através de uma sólida linha de base técnica, por meio da definição de uma arquitetura que possa ser utilizada no desenvolvimento de sistemas.

5.1 Esquema Proposto

Em um esquema tradicional, exibido na Figura 5.1, quando um usuário clica em um anúncio localizado no *site* de um publicador, o anúncio correspondente é obtido do anunciante e a transação é gravada pela rede de anúncios, para cobrança posterior. Uma rede de anúncios confiável realizará processos de detecção posteriores para garantir que tais cliques foram realizados com a intenção correta, para que cobranças indevidas não venham a ocorrer – só após essa verificação posterior a rede de anúncios cobra ao anunciante e paga ao publicador.

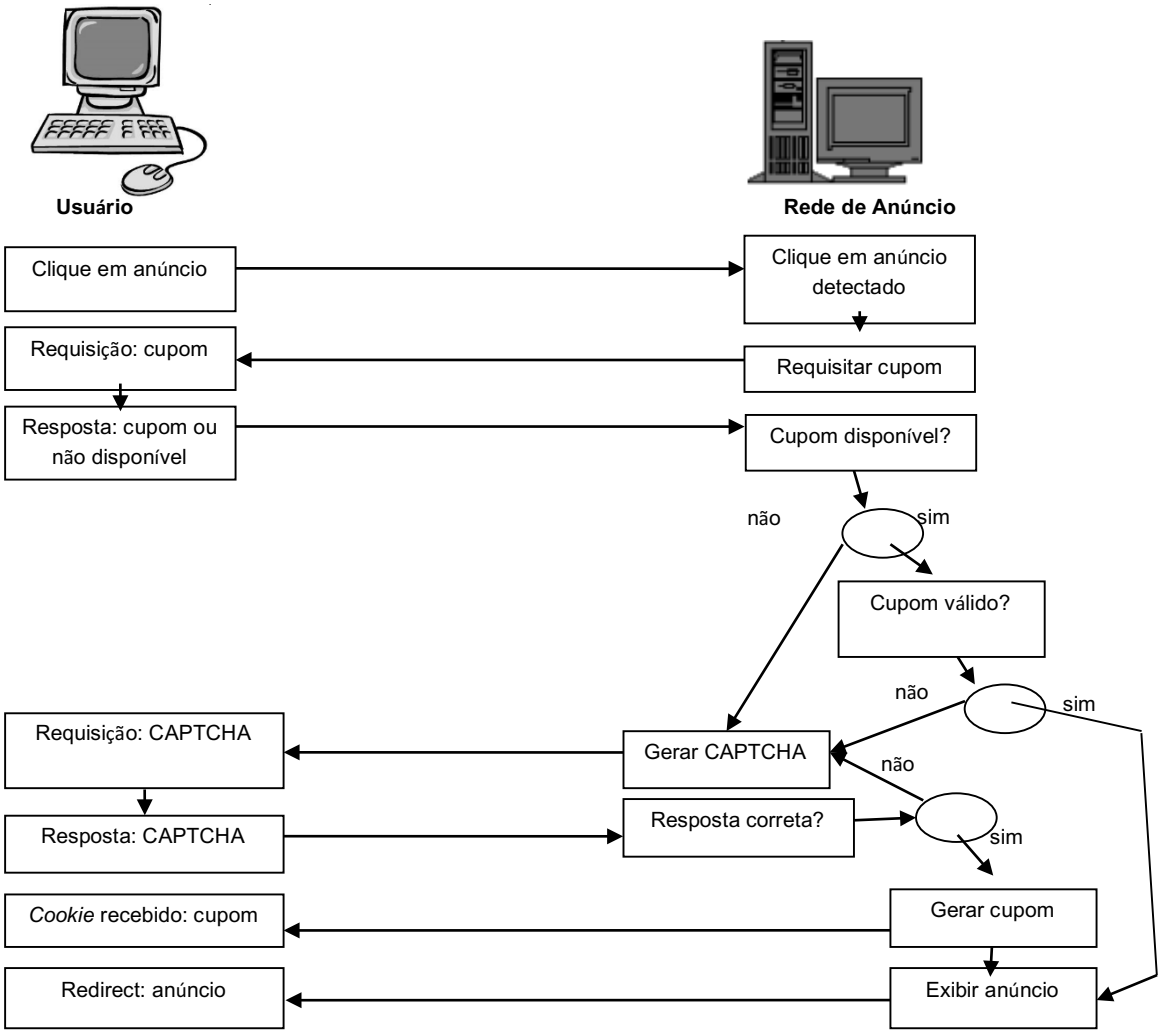
Figura 5.1 – Esquema tradicional de PPC

Fonte: Próprio Autor

No modelo que propomos nesta pesquisa, esquematizado na Figura 5.2, há algumas tarefas adicionais a serem realizadas: quando o usuário clica em um anúncio localizado, por exemplo, em um banner do representante localizado no site de um publicador, recebe do representante uma página web que contém um CAPTCHA clicável. Se o desafio for preenchido erroneamente, um novo desafio é proposto e o anúncio não é exibido. Quando o desafio for preenchido corretamente, um cupom com a comprovação de que se trata de um humano é

enviado, pela rede de anúncios, para o navegador do usuário e, após isso, o anúncio é finalmente exibido.

Figura 5.2 – Esquema proposto para autenticação em cupons.



Fonte: Próprio Autor

Sempre que um usuário previamente autenticado clicar em algum anúncio, esse cupom será disponibilizado para verificação para a rede de anúncios. Essa verificação pode ser iniciada tanto pela rede de anúncios quanto pelo anunciante, e ativa outro processo, agora de validação, na rede de anúncios. Após essa validação e da confirmação que se trata de um cupom válido, o anúncio é exibido. Para evitar a situação onde um usuário roda um programa a partir

de sua estação com o objetivo de realizar a fraude de clique após uma autenticação manual, os cupons são válidos apenas por um certo período de tempo que deve variar de 5 a 10 minutos, após o qual se faz necessária uma nova autenticação (PUTTACHAROEN; BUNYATNOPARAT, 2011).

É importante observar que os métodos atuais para filtragem não podem empregar o uso de *cookies* na detecção de cliques fraudulentos. Isso acontece porque a filtragem é um processo exclusivo: se um *cookie* fosse usado para marcar e excluir um determinado tipo de usuário malicioso, fraudadores poderiam simplesmente remover os *cookies* dos seus navegadores. Em contrapartida, a abordagem desta pesquisa é separatista, pois aceita apenas cliques válidos e pode, assim, fazer uso de *cookies* ou cupons. Os cupons marcam os usuários que responderam o CAPTCHA corretamente e que, portanto, são usuários válidos – essa é exatamente a razão pela qual este método é de prevenção e não de detecção.

O esquema necessário para implementar essa solução é baseado na autenticação de requisições por meio de resposta a CAPTCHA, com o objetivo de distinguir os usuários humanos de usuários automatizados. Uma vez obtido o sucesso no teste, o usuário recebe um cupom, que é armazenado na forma de *cookie*. Por questões de segurança, o cupom possui um tempo de validade, após o qual o usuário deve responder a um novo teste. O tempo de validade do cookie para uma autenticação como essa deve girar em torno de 5 a 10 minutos (PUTTACHAROEN; BUNYATNOPARAT, 2011), razão pela qual sugerimos 10 minutos neste trabalho, sendo esse inclusive o valor configurado inicialmente no cupom, embora tal funcionalidade possa ser alterada. O cupom pode tanto ser armazenado localmente, na sessão dos usuários, quanto pode ser implementado em um mecanismo de autoridade central, sendo essa decisão um tópico para estudo futuro.

O esquema retratado na Figura 5.2 tem dois aspectos distintos:

Autenticação por cupons: esse esquema possibilita a validação de cliques originados por clientes que ainda não produziram um histórico de tráfego de rede. Todo e qualquer cliente que não possua um cupom será considerado "não autenticado", e precisará responder um desafio antes de ter seu clique validado. Uma vez validado, o usuário será redirecionado para o site do anunciante. Após o preenchimento correto do desafio, o usuário receberá um cupom para que não necessite responder novos desafios em requisições adicionais durante o período de validade do cupom, durante o qual o usuário será considerado "autenticado". A razão para a proposta de validade se dá devido à necessidade de se evitar um cenário no qual um potencial fraudador autentique manualmente a primeira requisição e posteriormente rode um script local que realize a fraude de clique. É interessante notar também que esta abordagem facilita o método de detecção da fraude de clique por consultar o histórico de conexão. Sendo o clique considerado válido apenas se vier acompanhado de um cupom, detectar requisições múltiplas da mesma origem se torna fácil, desde que se armazene a apresentação dos cupons. Em abordagens tradicionais, nas quais os cupons não exercem nenhum papel, a detecção de tráfego de origem semelhante é mais difícil, e geralmente depende do mapeamento da origem dos dados, como mapeamento de IP, ou impressões de navegação, como identificadores de sessão.

Comprovador: um elemento arquitetural de nesse esquema é o serviço *web* Comprovador, responsável por identificar e marcar os computadores de usuários humanos, após a resposta correta. A disposição técnica do Comprovador depende da forma como o cupom será armazenado: se a opção de armazenamento local for adotada, uma autenticação simples baseada em *cookie* pode ser utilizada; se o armazenamento em autoridade central for utilizado, será necessária a implementação de uma unidade autenticadora nessa autoridade central, que pode funcionar como um serviço *web* em um dos servidores do representante.

Como já ressaltamos, este trabalho propõe uma abordagem complementar às soluções existentes, que se baseiam em uma abordagem de detecção de cliques fraudulentos. Dessa forma, para possibilitar a detecção de tais cliques, é necessário realizar uma análise no histórico dos cliques que foram realizados e considerados válidos segundo o esquema acima, para possibilitar a cobrança mensal juntamente aos anunciantes e o pagamento juntamente aos publicadores. Vislumbramos, portanto, a existência de uma terceira entidade, o **Detector**, que será responsável por armazenar informações relevantes para a detecção futura de tais cliques em um banco de dados quando o esquema proposto considerar determinado clique como válido, bem como rodar o algoritmo de detecção no histórico de cliques armazenados quando o momento for oportuno. Por questões práticas e como explicitado na seção 3.4 desta tese, decidimos adotar uma abordagem já utilizada com sucesso no passado para esse fim de detecção, a abordagem *Similarity-Seeker*, cujos detalhes técnicos da implementação serão explicitados na seção 5.2.6.

5.2 Arquitetura

Levando em consideração a abordagem a ser proposta, este esquema de PPC foi projetado de maneira a suportar a sua eventual terceirização, ou subcontratação. Ele deve garantir a segurança contra a fraude de clique quando os anúncios são publicados em ferramentas de pesquisa: nesse caso, é necessário tratar a rede de anúncios e os publicadores como uma entidade transparente. Sob esta perspectiva, observa-se os seguintes aspectos:

1. Marcação de cupons: o Comprovador/Detector identifica um visitante como legítimo ou não após a resposta correta de um desafio CAPTCHA, e faz isso através do envio para o histórico de *cookies* no navegador do usuário, de um cupom ‘c’, que passa a funcionar

durante o seu período de validade como um cartão de acesso para tentativas adicionais de uso do sistema, se elas vierem a ocorrer (ou seja, o usuário não precisará responder outro desafio caso o cupom esteja presente);

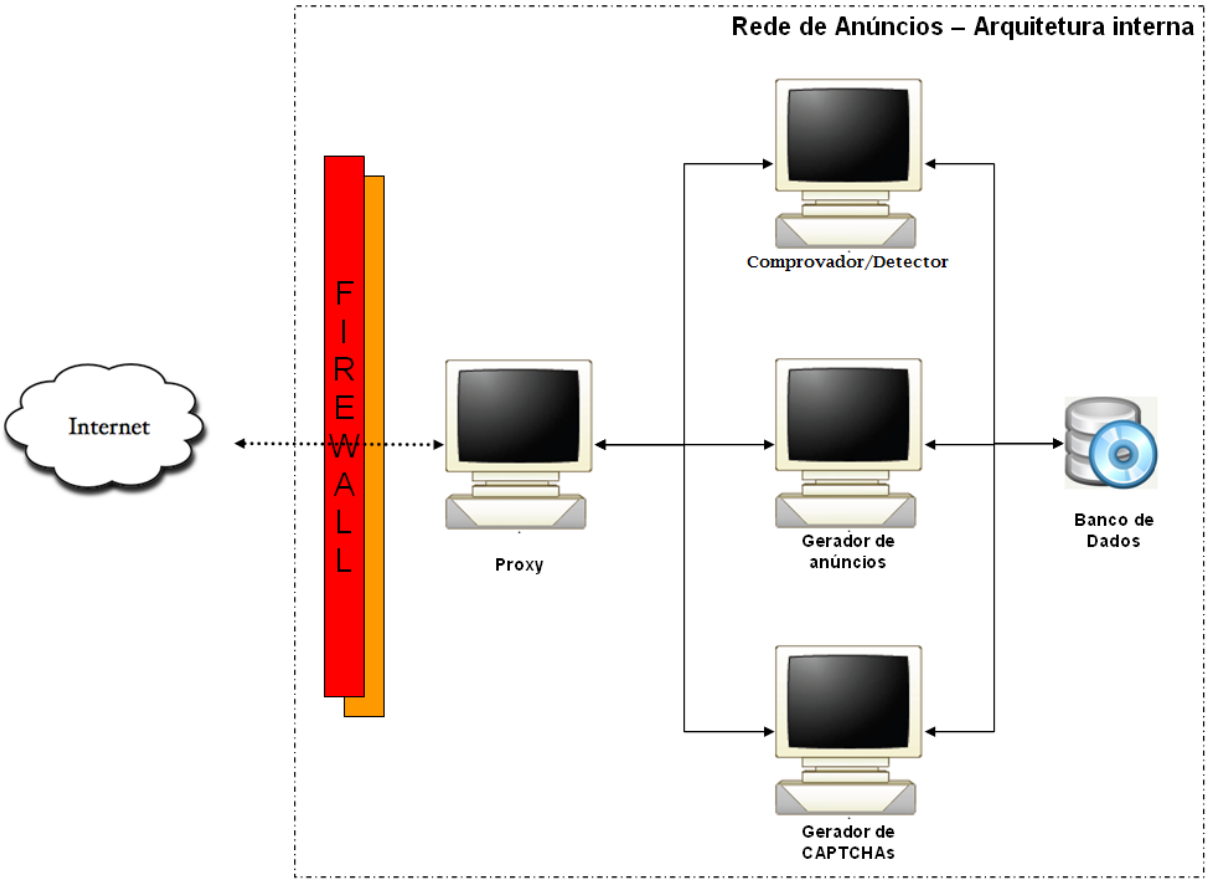
2. Liberação de cupons / geração de desafios CAPTCHA: quando o usuário clica em um anúncio no site de um publicador, o navegador do usuário é direcionado para uma URL no site da rede de anúncios. Essa URL carrega informações tais como a identificação do publicador, ID_{pbc} e a identificação do anúncio que foi clicado, ID_{anc}. A rede de anúncios deve então responder à requisição do usuário, buscando a identificação de um cupom válido existente no seu navegador. Se não existir, o usuário precisará responder um desafio CAPTCHA que será gerado pela entidade "Gerador de CAPTCHA". Ao responder corretamente, o serviço Comprovador/Detector será ativado para liberar um cupom c ao navegador do usuário, simultaneamente com ID_{pbc} e ID_{anc}, além do tempo de validade do cupom. Sendo C = (c, ID_{pbc}, ID_{anc}, v), de agora em diante, C será denominado "cupom", de acordo com o contexto;

3. Verificação de cupons: Ao receber C = (c, ID_{pbc}, ID_{anc}, v), é responsabilidade do Comprovador/Detector verificar que c é um cupom bem formado, como será descrito posteriormente. Além disso, o Comprovador/Detector verificará que C ainda não expirou (caso tenha expirado, será necessária a resolução de um novo desafio CAPTCHA) através do parâmetro v. Como já falamos, o valor pré configurado para 'v' é de 10 minutos, já que, de acordo com pesquisadores como Puttacharoen e Bunyatnoparat (2011), trata-se de um tempo razoável para validade de cookies de autenticação para este tipo de aplicação (com alta probabilidade de ataque). No entanto, o usuário pode alterar esse valor se achar que lhe convém ou por requisição dos clientes da rede de anúncios, pois o mesmo é configurável;

4. Recompensa: se o Comprovador/Detector verificar que C representa um clique válido, então a rede de anúncios irá pagar ao publicador e cobrar ao anunciante adequadamente. É importante observar que uma nova cobrança será disparada cada vez que um cupom for usado.

A partir dessas premissas, podemos propor um diagrama de componentes de alto nível, exibido na Figura 5.3, para implementar esses aspectos. Como se pode perceber, tomamos uma decisão arquitetural de concentrar as entidades Comprovador e Detector em um único elemento para evitar subutilização de elementos de *hardware* e a utilização demasiada de elementos de rede, já que ambas as entidades precisarão manipular os mesmos tipos de dados. Além dessa entidade, pode-se identificar os componentes Gerador de anúncios, que será responsável por acessar o Banco de dados e, a partir do mesmo, criar anúncios para exibição, e Gerador de CAPTCHA, que será responsável pela comunicação com a interface do Google NoCAPTCHA reCAPTCHA, que é externa ao sistema, e pela comunicação com o Comprovador/Detector, caso a resposta ao desafio tenha sido correta. Em caso de resposta errada, o ciclo de comunicação do Gerador de CAPTCHA com o Google NoCAPTCHA reCAPTCHA e o usuário continua. Adicionamos ainda um servidor Proxy (TEIXEIRA, 2004) e um Firewall (TEIXEIRA, 2004) nesse diagrama de componentes, para viabilizar um acesso controlado e seguro ao ambiente, e o próprio servidor de banco de dados, que será responsável por armazenar informações de anúncios, clientes, além de cliques realizados para posterior análise de detecção.

Figura 5.3 – Diagrama de Componentes para a Abordagem



Fonte: Próprio Autor

Obviamente, o publicador possui a opção de incluir informações adicionais em um cupom, tais como endereço MAC da máquina, identificador do hard drive, da sessão, e tipo de navegador (PUTTACHAROEN; BUNYATNOPARAT, 2011). A razão principal para alguém querer detalhar mais um cupom é aumentar a capacidade de se identificar, já que uma mesma rede doméstica pode ser usada por várias pessoas. Se cada um dos usuários acessa a Internet de suas contas pessoais e de suas próprias instâncias de navegadores na máquina do cliente, então essas instâncias carregarão cupons específicos. Nesta seção, serão detalhadas as especificidades técnicas da abordagem proposta.

5.3 Modelo de Dados

A forma mais simples de catalogar anúncios em um banco de dados que satisfaça esta abordagem é armazenar no banco de dados a informação necessária para suportar a execução do serviço Gerador de Anúncios. Para esta abordagem, isso acontecerá usando-se duas tabelas que descreveremos nesta seção.

A primeira tabela, denominada AnunciosBanner (Tabela 5.1), necessita conter informações sobre os anúncios, como a localização das imagens, a URL do site do anunciante e o texto a ser exibido pelo anúncio. Os campos associados a essa tabela são:

- IDAnuncio: a chave primária da tabela;

- ImagePath: armazena a localidade da imagem asssociada a este anúncio (que pode ser local ou externa, como /images/ads/bannerXX.gif);

- URL: link para o qual o anúncio redireciona o usuário;

- AltText: texto que o usuário verá ao mover o ponteiro para cima do anúncio.

Tabela 5.1 – AnunciosBanner

Nome do Campo	Tipo	Tamanho
IDAnuncio	AutoNumber	
ImagePath	Text	100
URL	Text	100
AltText	Text	255

Fonte: Próprio Autor

A segunda relação necessária é a ImpressoesBanner, vista na Tabela 5.2, que contém informações sobre o número de impressões de um determinado anúncio. Sempre que o anúncio é mostrado, um registro é adicionado à tabela. Os campos associados à mesma são:

- IDImpressao: chave primária da tabela.

- IDAnuncio: chave externa que relaciona essa tabela à tabela AnunciosBanner em uma relação “um para muitos”.

- DateClicked: data e hora em que o anúncio apareceu. Este campo pode ser complementado por outros que ordenem os anúncios. Na implementação mais simples possível, a ordenação se dá pela rotação de anúncios.

Tabela 5.2 – ImpressoesBanner

Nome do Campo	Tipo
IDImpressao	AutoNumber
IDAnuncio	Number
DateClicked	DateTime

Fonte: Próprio Autor

5.4 Gerador de CAPTCHA

Conforme explicado na seção 4.4.3, recomendamos a utilização de um CAPTCHA clicável baseado em imagem, para funcionar como gerador de CAPTCHA neste projeto. Conforme discutido na seção 4.5, tais CAPTCHA tendem a ser bem menos vulneráveis a ataques de fraudadores, em termos de automatização da resolução, além de possuírem um bom grau de aceitação de usuários finais.

5.4.1 Google NoCAPTCHA reCAPTCHA

Um exemplo de CAPTCHA de classe II desta natureza é o “Google NoCAPTCHA reCAPTCHA” (SHET, 2014). Para falarmos sobre ele, é necessário dar um pouco de histórico do projeto reCAPTCHA (AHN et al, 2008). Esse foi proposto em 2008 por Ahn et al (2008) como uma forma de aproveitar o tempo e esforço de pessoas que respondem os desafios em

uma espécie de sistema de informação altamente distribuído, com elementos não resolvidos e resolvidos, sendo os últimos constituídos por imagens que não foram reconhecidas satisfatoriamente por um OCR. Quando o usuário responde ambos os elementos, metade do seu trabalho valida o desafio e a outra metade é capturada como trabalho para o sistema distribuído.

Tal sistema é atualmente usado na conversão de trabalhos impressos, ou seja, imagens digitalizadas em textos digitais. Um exemplo da utilização do reCAPTCHA pode ser encontrado na Figura 5.4.

Figura 5.4 - reCAPTCHA

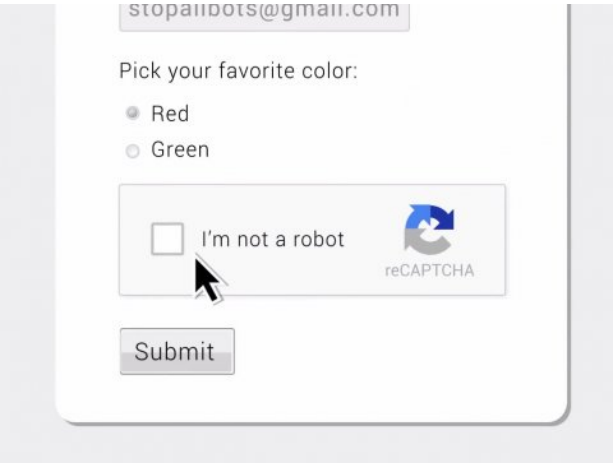
Fonte: http://www.google.com (2015)

Por sua vez, o Google NoCAPTCHA ReCAPTCHA busca simplificar a experiência de confirmação se determinado usuário é humano ou não, por meio da introdução de uma interface que permite que um número significativo de usuários previamente autenticados, ou que disponibilizem evidências suficientes que são humanos, cliquem apenas em um *checkbox*, conforme a Figura 4.6. Nos casos contrários, nos quais os usuários não foram autenticados

anteriormente e em que tais evidências não estejam disponíveis, o desafio terá que ser respondido, conforme a Figura 5.6.

Além do óbvio processo de facilitação para usuários finais na utilização do sistema reCAPTCHA, afinal os usuários vão apenas rapidamente clicar para seguir em frente, o argumento da equipe técnica é que os CAPTCHA tradicionais se confiam na incapacidade de robôs resolverem textos distorcidos. No entanto, uma pesquisa recente do próprio Google mostrou que a tecnologia de OCR é capaz de resolver o tipo mais difícil de texto distorcido em uma taxa de acerto de 99,8% (SHET, 2014).

Figura 5.5 – "No CAPTCHA reCAPTCHA" do Google

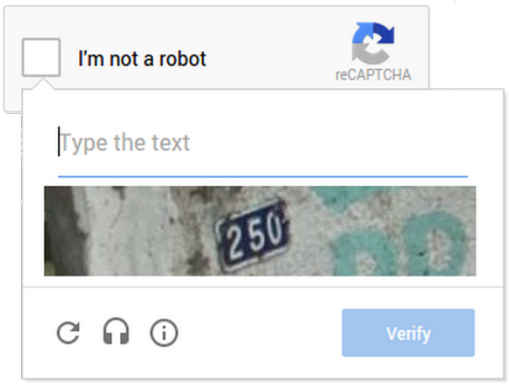


Fonte: http://www.google.com (2015)

A ferramenta de análise de riscos desenvolvida pela equipe do Google passa a considerar diversos fatores de engajamento do usuário para determinar se o mesmo é humano, para confiar cada vez menos no texto distorcido, considerado inseguro. Em contrapartida, uma melhor experiência para os usuários será oferecida, já que os mesmos, na maioria das vezes, poderão apenas clicar e seguir em frente, como demonstrado na Figura 5.5.

No entanto, naqueles casos em que a ferramenta de análise de riscos não consegue prever com um nível de segurança alto o suficiente se determinado usuário é humano, tipicamente, o "Google noCaptcha reCAPTCHA" apresenta um reCAPTCHA clássico para levantar mais pistas, conforme a Figura 5.6 apresenta, aumentando assim o número de pontos de checagem para confirmação. Para usuários em dispositivos móveis, um CAPTCHA clicável será exibido e o usuário pode utilizar o *touchscreen* para selecionar as respostas adequadas, ao invés de um reCAPTCHA, já que estes tornam necessária a digitação de texto. Para usuários em *desktops*, o administrador do *site* invocando a interface pode escolher se deseja a exibição de um reCAPTCHA clássico ou um clicável nessas situações.

Figura 5.6 – reCAPTCHA tradicional exibido no "No CAPTCHA reCAPTCHA"



Fonte: http://www.google.com (2015)

Além da autenticação prévia, semelhante ao processo explicado por Juels, Jakobsson e Jagatic (2006), o sistema utiliza "pistas comportamentais" para determinar se determinado usuário é humano, tais como: padrão de clique, cadência de digitação ou por quantos milissegundos o ponteiro do mouse fica descansando em determinado ponto antes de uma ação. Tantos detalhes, facilidade de uso e confiabilidade levaram empresas como Snapchat, Buzzfeed e Wordpress adotá-lo rapidamente (CONSTANTIN, 2014).

Entretanto, a forma como a ferramenta olha para o comportamento humano é muito parecida com aquela que a solução de *e-mails* do Google, o Gmail, utiliza para busca de *spam* e detecção de robôs, denominada *Botguard* (DANCHEV, 2014). Antes de mais nada, durante a verificação, ela olha se o usuário possui um *cookie* do Google na máquina. Então o Google NoCAPTCHA reCAPTCHA coloca seu próprio *cookie* no navegador, e busca por impressões *pixel a pixel* na janela de navegação do usuário, por informações como (HOMAKOV, 2014): objetos *javascript*, tamanho da tela, resolução, data, língua e *plug-ins*, endereço IP, informação CSS das páginas visitadas e uma impressão sobre o uso do *mouse*. Além disso, o CAPTCHA do Google vai pesquisar e utilizar qualquer *cookie* enviado por outros produtos do Google nos últimos seis meses, a crença é que os produtos do Google são utilizados de determinadas formas particulares e tais formas podem ser detectadas.

Toda essa informação é criptografada e enviada de volta para o Google. Ou seja, o uso do domínio google.com é intencional, e se os usuários permitirem, a empresa pode deixar *cookies* que ficam ativos por muito tempo nos computadores dos usuários, ignorando inclusive bloqueadores de anúncio, desde que o computador tenha utilizado algum serviço do Google anteriormente. Assim, são possíveis, basicamente, três situações (NACHI, 2014):

1. Se os *cookies* do Google estão presentes, o usuário clica e segue em frente;

2. Se os *cookies* do Google foram deletados, será necessário responder o desafio;

3. Se algum *plugin* que impossibilite o acesso à identificação estiver sendo usado, independente da presença dos *cookies*, será necessário responder o desafio.

Devido ao exposto acima, podemos perceber que o Google NoCAPTCHA reCAPTCHA é, portanto, em sua essência, um CAPTCHA de Classe II que habilita um aspecto de autenticação baseada em histórico de uso e utilização dos usuários e se preocupa em incorporar aspectos de usabilidade na experiência com o usuário. Isso é evidenciado no fato de

os engenheiros terem tornado a versão desktop configurável e hoje é possível ignorar o desafio escrito completamente: em nossas configurações, habilitamos apenas a versão clicável do mesmo do reCAPTCHA para o passo 2 descrito acima, tanto para dispositivos móveis como para a versão do protótipo utilizada pelo computador. Dessa forma, por ser essa uma solução que foca nos aspectos de engenharia de *software* descritos neste Capítulo 4 para CAPTCHA de classe II e descreve tais requisitos com foco no usuário, mas sem deixar aspectos de segurança computacional de lado, propomos a adoção desse modelo de CAPTCHA na nossa abordagem.

No entanto, também há pontos a serem questionados neste tipo de CAPTCHA. É possível que o Google não esteja utilizando o novo CAPTCHA para saber apenas se um usuário é ou não humano, mas potencialmente exatamente qual humano ele seja, já que a combinação da autenticação por *cookies* com padrões de navegação pode ser mapeada para um indivíduo específico – e a grande maioria dos usuários clicando em "*I'm not a robot*" nunca entenderá realmente isso. Isso possibilita, indiretamente, ligar uma atividade de navegação realizada muitas vezes fora dos domínios do Google, sob a chancela da segurança computacional, ao conhecimento do Google sobre o indivíduo realizando a atividade, sem dar a tal usuário a possibilidade de se negar a prover tal conhecimento (HOMAKOV, 2014), o que pode possuir impactos em privacidade e confidencialidade.

5.4.2 Método proposto para implementação do Gerador de CAPTCHA

Desta forma, a nossa proposta para este componente consiste em uma interface que invoca o Google NoCAPTCHA ReCAPTCHA como um componente externo ao sistema. O primeiro passo para o Gerador de CAPTCHA, portanto, é implementar um serviço capaz de invocar a API do Google NoCAPTCHA reCAPTCHA. Para isso, o código que invoca o desafio precisa ser semelhante ao demonstrado na Figura 5.3, com a diferença sendo o parâmetro data-sitekey, que precisa ser configurado individualmente (SHET, 2014):

Figura 5.7 – Código HTML Invocando o Desafio

```
1. <html>
2.     <title>Captcha - Clicável</title>
3.     <head>
4.           <script src='https://www.google.com/recaptcha/api.js'></script>
5.     </head>
6.     <body>
7.           Clique para confirmar que é humano, e depois prossiga!
8.           <p></center>
9.           <form action="resultado_captcha.php" method="post" id="mainform">
10.                <div class="g-recaptcha" data-sitekey="6Le3XYZ"></div>
11.                <input type="submit" name="submit" value="Prosseguir">
12.          </form>
13.    </body>
14.</html>
```

Fonte: Próprio Autor

Uma vez que a resposta ao desafio seja positiva, o Google NoCAPTCHA reCAPTCHA retorna um parâmetro POST denominado *g-recaptcha-response* (SHET, 2014), que pode ser verificado por meio do método *javascript grecaptcha.getResponse()* (SHET, 2014). Se o usuário respondeu corretamente ao desafio, o sistema pode entender o usuário como autenticado, e atribuir a ele um cupom, conforme descrito anteriormente na Tabela 5.3, enviando um *cupomauth* para o seu navegador. Nos casos de o usuário não responder corretamente, um novo desafio será exibido – o trecho de código acima é suficiente para cobrir esse caso, já que a interface do desafio permite ao usuário retornar para a página anterior.

É importante ressaltar que, embora a nossa recomendação seja a adoção do Google NoCAPTCHA reCAPTCHA, deixamos a possibilidade dessa proposta ser estendida para utilizar outros tipos de CAPTCHA de classe II como solução para esse problema. Nesses casos, a invocação do CAPTCHA deverá ser realizada em código-fonte, que pode ser compartilhado e integrado na rede de anúncios. Entretanto, os CAPTCHA clicáveis em geral, especialmente os baseados em imagem, necessitam da integração com um banco de dados que deve ser mantido em segurança e para garantir essa proteção, a melhor opção parece ser implementar o serviço Gerador de CAPTCHA como um serviço *web*.

5.5 Autenticadores

A entidade Comprovador/Detector descrita na Figura 5.3 é responsável pelos parâmetros de segurança nessa arquitetura. Conforme já exposto anteriormente, a mesma deve verificar a boa formação de cupons, se os mesmos ainda são válidos, auxiliar na identificação dos anúncios que estão sendo clicados, para possibilitar que o publicador receba, que o anunciante seja cobrado e que a rede de anúncios registre o clique adequadamente, além de ter que decidir se o cupom não foi usado recentemente.

Nesse sentido, um aspecto relevante da abordagem consiste na possibilidade de habilitar ao bom usuário o uso do sistema sem a necessidade de responder repetidamente desafios. A ideia por trás disso é evitar afastar a sobrecarga do uso do sistema de CAPTCHA, evitar demoras na exibição dos anúncios e possibilitar uma maior usabilidade da rede de anúncios online (OU, 2012). Levando em consideração a arquitetura descrita, essa habilitação torna-se possível por meio de uma camada de autenticação, cuja decisão acerca do formato explicamos nesta seção.

5.5.1 Uma discussão sobre autenticação

Uma discussão séria sobre segurança em redes e computadores envolve um conhecimento sólido sobre métodos de autenticação sempre. Esta subseção tenta descrever o papel da autenticação no plano de segurança.

A segurança de computadores e de redes se baseia em dois objetivos básicos:

1. Impedir pessoas não autorizadas de terem acesso a recursos;

2. Garantir que pessoas autorizadas podem acessar os recursos que precisam.

Há um número componentes envolvidos na consecução desses objetivos. Uma forma é garantir permissões a recursos que especificam quais usuários podem ou não os acessar e sob que circunstâncias (por exemplo, um determinado grupo de usuários pode ter acesso apenas se estiver acessando fisicamente no site de trabalho, mas não de uma localidade remota). Permissões de acesso, no entanto, só são relevantes se for possível verificar a identidade do usuário que está tentando acessar os recursos; e é por isso que autenticação é relevante, especialmente se entendida em um contexto geral, levando em consideração a forma como tipicamente funciona e seus protocolos para que, uma vez que se entenda os seus tipos mais comuns em um plano de segurança de rede, seja possível para esta pesquisa tomar uma decisão de uma forma efetiva a respeito de um método de autenticação a ser adotado na abordagem proposta nesta seção.

Autenticação e Segurança vs Autorização

A primeira base a ser estabelecida é a diferenciação entre autenticação e autorização. Autenticação é um elemento absolutamente essencial de um modelo de segurança

típico. Se trata do processo de confirmar a identidade do usuário ou de uma máquina que está tentando se logar ou acessar recursos. Há um número de mecanismos diferentes de autenticação, mas todos servem ao mesmo propósito.

Por sua vez, autorização consiste em verificar que o usuário em questão possui as permissões e credenciais corretas para acessar o recurso requerido. É muito fácil confundir autorização e autenticação, pois ambos são elementos do plano de segurança em redes e computadores – no entanto, autenticação verifica exclusivamente a identidade do usuário. Como se pode perceber, ambas trabalham juntas: a autenticação ocorre primeiro, seguida da autorização.

Em teoria, a autenticação é relativamente simples: um usuário fornece algum tipo de credenciais – uma senha, cartão inteligente, impressão digital, certificado de que digitais identifica o usuário como a pessoa que está autorizada a acessar o sistema. Existem, no entanto, uma multiplicidade de métodos e protocolos que podem ser usados para fazer isso. Independentemente do método, o processo de autenticação de base continua a ser a mesma.

Na maioria dos casos, um usuário deve ter uma conta de usuário válida configurada pelo administrador de rede que especifica as permissões e os direitos. As credenciais do usuário devem estar associadas a esta conta – a senha foi atribuída, ou um certificado de cartão inteligente é emitido, ou uma verificação biométrica é inserida no banco de dados contra a qual leituras futuras serão comparadas. Quando o usuário deseja fazer logon, ele ou ela fornece as credenciais e o sistema verifica a base de dados para a entrada original e faz a comparação. Se as credenciais forem fornecidas pelo usuário e forem verificadas com aquelas contidas no banco de dados, o acesso é concedido.

Em um ambiente de alta segurança, a autenticação de várias camadas adiciona proteção extra. Em outras palavras, é possível exigir que o usuário forneça mais de um tipo de

credencial, como tanto uma impressão digital, como com uma senha de logon. Isso diminui ainda mais as chances de uma pessoa não autorizada contornar o sistema de segurança.

Nesta arquitetura, conforme demonstrado na Figura 5.2, sua identidade como humano é verificada por meio de uma chave de autenticação passada pelo Google NoCAPTCHA reCAPTCHA. Após isso, o usuário recebe um *token* de acesso, que denominamos de cupom, que contém informação sobre os grupos de segurança aos quais o usuário pertence. Ao tentar utilizar um recurso da rede de anúncios, a lista de controle de acesso associada com tal recurso é verificada com respeito ao *token* de acesso. Se a lista de controle mostra que os membros de um determinado grupo podem acessar um determinado site, e o usuário provou ser humano, o usuário obterá acesso, a não ser que houve um banimento explícito de acesso ao recurso. Um modelo semelhante a esse é o das listas de controle de acesso (ACL) dos grupos de usuários do Windows que acesso recursos do sistema operacional.

Neste sentido, existem diversas formas distintas de autenticação. Podemos citar algumas abaixo:

- Autenticação por logon: a maioria dos sistemas operacionais baseados em rede requerem que um usuário seja autenticado para se logar na rede. Tipicamente isso é realizado ao se entrar uma senha, ao se inserir um smartcard e entrar um PIN associado, ao se colocar uma digital, um padrão de voz, um scan de retina, ou ao se usar um outro meio para se provar ao sistema que você é quem diz ser (SEDIYONO; SANTOSO, 2013).

- Autenticação por acesso à rede: verifica a identidade do usuário para cada serviço de rede que o usuário tenta acessar. De acordo com Sediyono e Santoso (2013), esse método se diferencia, pois, este processo de autenticação, na maioria dos casos, é transparente para o usuário, uma vez que ele esteja logado.

- Autenticação IP Security (IPSec): de acordo com Uskov (2012), fornece um meio para os usuários criptografaram ou assinarem mensagens que são enviadas através da rede para garantirem confidencialidade, integridade e autenticidade. As transmissões IPSec podem usar uma variedade de métodos de autenticação, incluindo o protocolo Kerberos (USKOV, 2012), certificados de chave pública emitidos por uma autoridade de certificação confiável (CA), ou uma chave secreta simples pré-compartilhada, como uma cadeia de caracteres conhecidos tanto o remetente como o destinatário. Uma consideração importante é que tanto os computadores que recebem e enviam as mensagens devem ser configurados para usar um método de autenticação comum, ou eles não serão capazes de se envolver em comunicações seguras.

Autenticação remota: Há uma série de métodos de autenticação que podem ser usados para confirmar a identidade de usuários que se conectam à rede através de uma conexão remota, como dial-up ou VPN. De acordo com Pawlowski et al (2014), esses incluem o protocolo de autenticação de senha (PAP), o Shiva PAP (SPAP), o Challenge Handshake Authentication Protocol (CHAP), o Microsoft CHAP (MS-CHAP) e a Extensible Authentication Protocol (EAP) (PAWLOWSKI et al, 2014). Os usuários remotos podem ser autenticados através de um Remote Authentication Dial-In User Service (RADIUS) ou de um serviço de autenticação da Internet (IAS) (ANANTATHANAVIT; MUNLIN, 2013). Discutiremos esses em mais detalhes mais à frente.

- Single Sign-On (SSO): é um recurso que permite ao usuário usar uma senha (ou smart card) para autenticar a vários servidores em uma rede sem precisar reinserir credenciais (REVAR, BHAVSAR, 2011). Esta é uma conveniência óbvia para os usuários, que não têm de se lembrar de várias senhas ou manter o processo de autenticação para acessar recursos diferentes. Há uma série de produtos de SSO no mercado que permitem single sign-on em um ambiente misto (híbrido), que incorpora, por exemplo, servidores Microsoft Windows, Novell NetWare e UNIX (REVAR, BHAVSAR, 2011).

Há também um número considerável de protocolos e métodos de autenticação que podem ser usados, dependendo da aplicação e dos requisitos de segurança envolvidos no sistema. Podemos citar alguns comuns a seguir:

- Kerberos: foi desenvolvido no MIT para fornecer autenticação segura para redes UNIX. Tornou-se um padrão da Internet e é suportado até pelas versões mais recentes de sistema operacional de rede da Microsoft. De acordo com Uskov (2012), o Kerberos usa certificados temporários chamados tickets, que contêm as credenciais que identificam o usuário para os servidores na rede. Na versão atual do Kerberos, v5, os dados constantes dos tickets são criptografados, incluindo a senha do usuário. Um centro de distribuição de chaves (KDC) é um serviço que é executado em um servidor de rede, que emite um ticket chamado de Ticket Granting Ticket (TGT) para os clientes que autentica o tíquete de concessão de Serviço (TGS). O cliente utiliza esta TGT para acessar o TGS (que pode ser executado no mesmo computador que o KDC). A TGS emite um ticket de serviço ou sessão, que é usado para acessar um serviço de rede ou recurso (USKOV, 2012).

- SSL: O protocolo SSL é um outro padrão da Internet, muitas vezes usado para fornecer acesso seguro a sites da Web, usando uma combinação de tecnologia de chave pública e tecnologia chave secreta (SEDIYONO, 2013). A criptografia de chave secreta (também chamada de criptografia simétrica) é mais rápida, mas a criptografia de chave pública assimétrica prevê uma melhor autenticação, de modo SSL, e é projetada para beneficiar as vantagens de ambos. SSL é suportado por todos os principais navegadores, e pela maioria dos softwares de servidor da Web, como o IIS e Apache e opera na camada de aplicação do modelo de rede DoD. Isso significa que os aplicativos devem ser escritos para usá-lo, ao contrário de outros protocolos de segurança (tais como IPSec) que operam em camadas inferiores. O padrão da Internet Transport Layer Security (TLS) é baseada em SSL. A autenticação por meio de SSL é baseada em certificados digitais que permitem que os servidores Web e clientes verifiquem a

identidade de cada um antes de estabelecer uma conexão, a autenticação mútua. Assim, são utilizados dois tipos de certificados: certificados de cliente e certificados de servidor.

- PAP e SPAP: PAP é usado para autenticação de um usuário sobre um controle de acesso remoto. Uma característica importante do PAP é que ele envia senhas de usuários através da rede para o servidor de autenticação em texto simples (ANANTATHANAVIT; MUNLIN, 2013). Isso representa um risco de segurança significativo, como um usuário não autorizado pode capturar os pacotes de dados usando um analisador de protocolo (como um *sniffer*) e obter a senha. A vantagem do PAP é que ele é compatível com muitos tipos de servidores que executam sistemas operacionais diferentes. PAP deve ser utilizado apenas quando necessário para fins de compatibilidade. Por sua vez, SPAP é uma melhoria em relação PAP em termos do nível de segurança, uma vez que utiliza um método de criptografia (utilizado por servidores de acesso remoto Shiva, daí o nome) (PAWLOWSKI et al, 2014). O cliente envia o nome de usuário junto com a senha criptografada, e o servidor remoto decifra a senha. Se o nome de usuário e senha correspondem as informações no banco de dados do servidor, o servidor remoto envia uma mensagem de confirmação (ACK) e permite a conexão. Se não, uma confirmação negativa (NAK) é enviado, e a conexão é recusada.

- CHAP e MS-CHAP: CHAP é outro protocolo de autenticação usado para a segurança de acesso remoto. É um padrão de Internet que usa MD5, um método de criptografia unidirecioal, que realiza uma operação de *hash* na senha e transmite o *hash* resultante, em vez da senha em si, através da rede (PAWLOWSKI et al, 2014). Isso tem vantagens óbvias de segurança sobre o PAP / SPAP, como a senha não vai em toda a rede e não pode ser capturada. O algoritmo de *hash* garante que a operação não pode ser reversa para obter a senha original a partir dos resultados hash. CHAP é, no entanto, vulnerável a representação servidor remoto. MS-CHAP é versão da Microsoft para CHAP. A versão 2 do MS-CHAP usa uma autenticação de duas vias, de modo que a identidade do servidor, assim como o cliente, é verificada. Isso

protege contra a representação servidor. MS-CHAP também aumenta a segurança usando chaves criptográficas separadas para dados transmitidos e recebidos.

- EAP: trata-se de um meio de autenticar uma ligação do protocolo ponto a ponto (PPP) que permite que os computadores se comuniquem para negociar um esquema de autenticação específico, chamado do tipo "EAP". Uma característica chave da EAP é a sua extensibilidade, indicado por seu nome. Módulos podem ser adicionados tanto no cliente quanto no servidor para suportar novos tipos. EAP pode ser usado com TLS (chamado EAP-TLS) para fornecer autenticação mútua através da troca de certificados de usuário e máquina (PAWLOWSKI et al, 2014). EAP-TLS foi definido na RFC 2716 e também pode ser usado com RADIUS (ANANTATHANAVIT; MUNLIN, 2013).

- RADIUS: é frequentemente usado por prestadores de serviços de Internet (ISPs) para autenticar e autorizar dial-up ou usuários de VPN (YANG et al, 2014). De acordo com Yang et al (2014), as normas para RADIUS foram definidas nos RFC 2138 e 2139. Funciona da seguinte forma: um servidor RADIUS recebe as credenciais do usuário e informações de conexão de clientes de uma conexão discada e os autentica na rede. RADIUS também pode executar serviços de contabilidade e mensagens EAP, através da sua passagem para um servidor RADIUS com fins de autenticação. Para isso, basta instalar o suporte à EAP no servidor RADIUS – não é necessário nas máquinas cliente (PAWLOWSKI et al, 2014). A partir do Windows 2000, a Microsoft incluiu um serviço de servidor RADIUS chamado Serviços de Autenticação da Internet (IAS), que implementa as normas RADIUS e permite o uso de PAP, CHAP, MS-CHAP ou EAP.

- Certificados Digitais: consistem em dados que são usados para autenticação e segurança de comunicações, especialmente em redes não protegidas. Certificados associam uma chave pública a um usuário ou a outra entidade (um computador ou serviço) que tem a

chave privada correspondente (DINU; TOGAN, 2014). Os certificados são emitidos por autoridades de certificação (CA), que são entidades confiáveis que "atestam" a identidade do usuário ou computador. A CA assina digitalmente os certificados que emite, usando a sua chave privada. Os certificados são válidos apenas por um período de tempo especificado; quando um certificado expira, uma nova deve ser emitida. A autoridade emissora também pode revogar certificados (DINU; TOGAN, 2014).

5.5.2 Método de autenticação proposto

Para um sistema ou serviço típico complexo de segurança de rede, é importante que os usuários remotos sejam devidamente autenticados, pois eles geralmente representam um risco de segurança maior do que os usuários no local. Na nossa solução, no entanto, temos um nível de requisito de segurança relacionado à autenticação mais permissivo, já que o nível de autenticação aqui presente está relacionado a possibilidade de uma maior usabilidade do serviço, pelo que o nosso critério de decidibilidade vem na verdade ao encontro da percebida segurança do sistema de CAPTCHA, questão que deve ser considerada na tomada dessa decisão de segurança específica.

Levando isso em consideração, uma solução inspirada em um modelo de certificados digitais, semelhante ao de Feng e Li (2013) – que preserva as vantagens de uma certificação implícita embora nenhum esquema de encriptação baseado em certificados seja de fato utilizado – parece ser a que melhor se adapta a necessidade desse modelo, já que que esta abordagem não pode forçar uma autenticação em vários passos ao usuário. É importante ressaltar que, nesta solução, o cenário típico do bom usuário é tal que o mesmo, após uma opção de navegação, clicará em um anúncio, receberá um desafio para confirmar sua identidade como

humano e seguirá com o uso. Caso o mesmo venha a clicar em outro anúncio enquanto autenticado, a necessidade de exibição de anúncios não existirá.

Com respeito às decisões técnicas nessa proposta, é importante ressaltar que a complexidade envolvida em certificados digitais é considerável, e apenas alguns aspectos necessitam ser implementados nesta solução, já que o mecanismo de autenticação se dará pela resposta de um CAPTCHA. Assim, a primeira decisão técnica corresponde a respeito da autenticação com SSL, que propomos a ser um dos aspectos desconsiderados, para evitar problemas de certificados inválidos, sendo essa etapa de segurança substituída por Cache Cookies, que serão explicados a seguir, na seção 5.5.3. Esse foi a mesma decisão adotada por Feng e Li (2013), cuja segurança é garantida pela dificuldade tradicional da paridade bilinear.

Nesse sentido, para a implantação de certificados digitais, é necessária a designação de uma autoridade de certificação. A segunda decisão corresponde a contratação de uma autoridade de certificação externa ou da utilização de um servidor próprio como autoridade. Como todos os servidores são internos e acessados por softwares internos, entendemos que a autoridade de certificação própria é a melhor estratégia, e no nosso caso o próprio serviço comprovador é o mais indicado para essa função.

A terceira decisão diz respeito a simplificar a solução no nível de autoridade de certificação. Uma arquitetura típica de certificados digitais implementa um mecanismo de reconhecimento do certificado root da autoridade, para impedir que partes terceiras confiem em certificados inválidos tentando replicar a autoridade. Na nossa arquitetura, esse nível de segurança é irrelevante, já que não há verificação por terceiros pelos cupons emitidos pelo serviço comprovador, e assim podemos utilizar um certificado de autoridade baseado em cookies, garantindo confiança internamente e evitando a quebra de confiança em navegadores

de usuários externos, enquanto o cupom for válido. Conforme já falamos, o tipo de cookie a ser adotado será explicado na seção a seguir 5.5.3.

Além disso, as vantagens de se simular uma solução com autoridade de certificação interna incluem o fato de a mesma ser gratuita, necessitar de revogação com tempo de expiração e o fato de a autenticação está sendo utilizada por razões internas, inerentes à usabilidade da rede de anúncios online.

Podemos destacar também algumas desvantagens:

- Requisitos adicionais de segurança: a autoridade certificadora será provavelmente a máquina mais importante, em termos de segurança, da solução. Se comprometida, a segurança da arquitetura estará completamente comprometida também – ela precisa ser isolada;

- Backup/alta disponibilidade: é necessária uma solução que garanta que o comprovador, por ser sua autoridade certificadora e responsável por verificar os cupons, esteja sempre disponível e que seus dados sejam frequentemente replicados e salvos em backup;

- Confiança: se, por algum motivo, os cupons forem comprometidos, ou a forma de gerar os cupons for descoberta, a autoridade certificadora estará comprometida e uma nova solução de TI precisará ser adotada;

5.5.3 Cache Cookies

Não é foco desta tese o estudo aprofundado da tecnologia de *cookies*, no sentido da análise de suas vantagens e desvantagens, ou de alternativas ao seu uso. Este tipo de estudo pode ser encontrado em trabalhos como (PUTTACHAROEN; BUNYATNOPARAT, 2011) e (WROBLEWSKI, 2010), que serão utilizados como base da proposta aqui apresentada. Assim,

o primeiro passo para o prosseguimento de nossa proposta é a definição do tipo de *cookie* recomendado, além da apresentação de como eles serão utilizados pelo mecanismo de autenticação. Por essa razão, esta seção será utilizada para apresentar uma rápida conceituação do que sejam *cookies*, e para discutir a disposição da tecnologia de *cookie*s nesta arquitetura.

Um *cookie* é uma *string* (um pedaço de texto) armazenada no navegador web de um usuário de Internet, e pode ser utilizado para diversos fins: autenticação, armazenamento de preferências de navegação, identificação de sessão ou qualquer outra finalidade para se armazenar dados de navegação no formato texto. De acordo com Puttacharoen e Bunyatnoparat (2011), todo e qualquer *cookie* precisa ter cinco parâmetros: "nome do *cookie*, valor, data de validade, URL para a qual o *cookie* é válido e se o *cookie* deve ser enviado apenas através de uma conexão segura por SSL". A maioria dos navegadores modernos dá aos usuários a opção de aceitar ou não *cookies*.

O mecanismo de troca de mensagens cliente-servidor de *cookies* se dá da seguinte forma: inicialmente, o *cookie* é enviado como um cabeçalho HTTP pelo servidor web para o navegador, que, durante o período de validade do *cookie*, o envia de volta ao servidor sempre que ele for acessado. Como são textos, os *cookies* não podem ser executados e, assim, não podem se autorreplicar como vírus. Entretanto, por causa do mecanismo de navegação, que utiliza funções de atribuir e ler *cookies*, eles podem ser utilizados como *spyware* (BINDU, 2015), sendo este o motivo pelo qual alguns softwares *antispyware* alertam os usuários da presença de *cookies*.

Para garantir a associação correta de um *cookie* com o navegador que o criou, o cupom deve ser comunicado na forma de valor armazenado no navegador. Ao mesmo tempo, é importante garantir que os cupons sejam configurados de forma a serem recuperados apenas pela rede de anúncios, não podendo ser coletados por fraudadores.

Cookies de terceiros são a forma mais óbvia de se instanciar cupons. Entretanto, por terem um histórico de uso abusivo, é intuitivo que sejam bloqueados por usuários. Uma alternativa para contornar esse problema é a utilização de *cookies* privados. Uma rede de anúncios pode implantar cupons privados para o seu uso particular, caso o Comprovador use mecanismos de identificação específica por usuários, ou específica por seções. Entretanto, o uso de *cookies* privados pode gerar uma complexidade indesejada se forem necessários redirecionamentos em múltiplos passos, como no caso do fim da validade de um dado cupom.

Para resolver tanto a questão da complexidade quanto contornar o bloqueio de *cookie*s por parte dos usuários, *cache cookies* (JUELS; JAKOBSSON; JAGATIC, 2006) parecem ser uma opção interessante. Nesse caso, um comprovador pode embutir um cupom, que é marcado como de leitura exclusiva pelo servidor proxy da rede de anúncios. Como podem ser configurados para sites de terceiros, *cache cookies* se assemelham a *cookies* de terceiros, mas possuem uma característica diferente: qualquer site pode instanciar a liberação do *cache cookie* para o qual está associado. Além disso, *cache cookies* funcionam em navegadores que bloqueiam o uso de *cookies* comuns, razão pela qual esta abordagem recomenda sua utilização.

Em síntese, um *cache cookie* funciona da seguinte forma: para definir um *cache cookie* associado ao site www.X.com, o *cache cookie* assume a forma de uma página HTML ABC.html que requer um recurso de www.X.com com o seu valor c associado. Por exemplo, a página ABC.html pode requisitar um arquivo texto na forma http://www.X.com/c.ini. Qualquer web site pode criar um ABC.html e implantá-lo no navegador de um visitante. Similarmente, qualquer web site que conheça a página/*cache cookie* ABC.html pode referenciá-la, causando o recebimento, por parte de www.X.com, de uma requisição pelo arquivo texto c.ini. Entretanto, apenas www.X.com pode receber o *cache cookie*, isto é, o valor “c.ini”, associado ao *cache cookie*, quando o mesmo for liberado.

Assim, para realizar cada uma dessas funções, simulando um processo de autenticação por certificado e utilizando cache cookies, durante o processo de desenvolvimento identificou-se a necessidade de se trabalhar com quatro tipos diferentes de *cookies*, que estão dispostos ao longo do código do aplicativo. Os cookies e respectivos comandos de iniciação estão dispostos na Tabela 5.3.

Tabela 5.3 – Cookies necessários para as funcionalidades do Comprovador/Detector

Cookie	**Comando PHP**	**Função primária**
Variavelid	setcookie('variavelid', $id);	Carrega o número do anúncio para registrar cobrança.
Variavellink	setcookie('variavellink', $link);	Carrega o link do anúncio para possibilitar redirecionamento.
Ipusuario	setcookie('ipusuario', $ip);	Carrega o IP do usuário e deve permitir fazer os controles para posterior detecção.
Cupomauth	setcookie('ipusuario', $ip, time()+600000);	Autenticador de fato. Verifica a necessidade de resposta de um novo desafio. Carrega um tempo de validade de 600000 milissegundos por padrão, ou 10 minutos. Esse valor pode ser configurável pelo administrador da rede de anúncios se julgar necessário.

Fonte: Próprio Autor

Conforme explicitado na Tabela 5.3, os cookies ‘variavelid’ e ‘variavellink’ carregam informações inerentes ao anúncio sendo clicado. Ambas as informações são utilizadas por outros elementos arquiteturais para ação posterior junto aos publicadores, anunciantes, já que ‘variavelid’ carrega informações que os identifica, e ao próprio usuário realizando a ação, que deverá ser redirecionado para o link armazenado junto ao cookie ‘variavellink’.

Os elementos associados à segurança do sistema são mesmo ‘ipusuario’ e ‘cupomauth’. O primeiro, na versão atual do protótipo, carrega, conforme o próprio nome diz, o IP do usuário realizando o clique, e deve alimentar os mecanismos de detecção posteriores para pesquisar se a ação corrente não foi realizada com intenções fraudulentas. JavaScript permite identificar um número de itens adicionais, como: endereço MAC, identificadores de disco rígido, como o endereço serial, e ainda identificadores da placa mãe, como o seu endereço serial. Vislumbramos também adicionar identificação do próprio usuário na seção corrente. O ‘cupomauth’, por sua vez, que se trata do cupom de fato, carrega a informação se o usuário se encontra autenticado após a resolução de um CAPTCHA, e serve para carregar aspectos de validade desse cupom como, por exemplo, o seu tempo de validade. O mesmo também carrega a informação do usuário autenticado e pode ser utilizado em conjunto com o substituto de ‘ipusuario’ nas versões futuras do sistema.

5.6 Gerador de Anúncios

Nesta seção apresentaremos os aspectos por trás da entidade Gerador de Anúncios. O desenvolvimento desse serviço é uma parte fundamental na existência de qualquer rede de anúncios online, e a sua implementação tem sido estudada extensivamente ao longo dos últimos anos. Limitar a especificação que se deve seguir para o desenvolvimento do Gerador de Anúncios seria pouco inteligente, já que o desenvolvimento do mesmo precisa ser uma função do modelo de negócios adotado pela rede de anúncios. Por exemplo, há redes de anúncios que leiloam entre os seus anunciantes as melhores posições nos *banners* de anúncio (OU, 2012). Em contrapartida, há redes que preferem ordenar os seus anúncios com base em estatísticas como *page ranks*, para dar melhores oportunidades a anunciantes menos visitados, promovendo

o desenvolvimento de suas marcas (KOTLER; KELLER, 2009). Há ainda as redes que tratam todos os anunciantes igualmente, realizando um rodízio nas exibições (DAVAR, 2013).

A primeira responsabilidade do Gerador de Anúncios é construir o *banner* contendo os anúncios que serão exibidos em um dado momento por um publicador. Essa instância de anúncios poderá ser montada a partir de uma busca, devido a um histórico de navegação, por exemplo, dependendo obviamente da estratégia adotada pela rede de anúncios para montar sua grade de exibição. Por causa disso, é recomendável que cada abordagem defina uma entrada comum, que será utilizada para realização da busca por anúncios. O modelo básico de busca se dá de acordo com palavras-chave. É importante lembrar que o *banner* conterá links para o site da rede de anúncios, contendo identificadores para redirecionamento.

A segunda responsabilidade está associada à seleção dos anúncios que foram retornados por uma requisição, para permitir a busca patrocinada de anúncios e uma estratégia particular para cada rede. Isso caracteriza uma busca por dados relevantes e a classificação destes dados, para exibição. Os anúncios melhor ranqueados serão exibidos, em função da requisição de cada publicador, levando em consideração aspectos como palavras-chave, *page ranks*, geolocalização, leilão às escuras e informações sobre navegação dos usuários. No momento desta tese, implementamos um protótipo do Gerador de Anúncios para possibilitar a visualização dos mesmos e do estado atual da pesquisa, de maneira que neste protótipo os mesmos são exibidos do topo à base de uma página *web*, de acordo com o preço acordado em *slots*[3] em seis preços possíveis, obedecendo o número de menos exibições. Se tudo isso for equivalente, a exibição será aleatória.

[3] Um slot é um local de anúncio no banner de exibição. Anúncios melhor ranqueados conseguem pegar slots melhores.

No entanto, a pesquisa por métodos de busca em anúncios é um tema de pesquisa relevante e complexo (DAVAR, 2013), (OU, 2012) (FAIN; PEDERSEN, 2006), e a elaboração de um método de gerenciamento de campanhas de anunciante seria uma pesquisa por si só, não sendo escopo desta pesquisa de doutorado. Jansen e Mullen (2008) discutem métodos para busca patrocinada.

A terceira responsabilidade do Gerador de Anúncios está associada ao fato de essa entidade ser responsável pelo acesso ao banco de dados de anúncios, especificamente com relação às funcionalidades de interface de consulta, atualização, e cadastramento de novos anúncios. É comum, em implementações de redes de anúncios online, que os anunciantes possuam uma interface de administração para editar os seus anúncios, a qual pode rodar como um serviço web no Gerador de Anúncios, e ser disponibilizado nos moldes de um painel de controle. Na realidade, a implementação desse serviço é muito simples, e uma interface de cadastro na *web* para a construção dos anúncios parece ser o formato ideal, pois permitiria a interação entre anunciantes e redes de anúncios até mesmo remotamente. Para o protótipo há também uma interface similar a um painel de controle para o cadastro, edição e remoção de anúncios.

Para possibilitar a realização das atividades listadas, o banco de dados precisa implementar um conjunto de dados que foi discutido na seção 5.3.

5.7 Detecção de Cliques

A solução descrita até o momento foca no aspecto de prevenção da fraude de clique. Conforme já expomos anteriormente, para uma solução ótima, não se pode desprezar as alternativas de combate atuais à fraude– nesse sentido, o aspecto de prevenção precisa ser

complementado pela detecção de fraudes em cliques registrados. Este aspecto da detecção é importante no sentido de mitigar a ocorrência da fraude, já que adiciona um passo extra de segurança nos cliques previamente identificados como válidos e não fraudulentos. Por exemplo, na abordagem proposta nesta tese, é impossível prevenir a fraude de clique manual por meio de cliques duplos – somente uma abordagem de detecção é capaz de filtrar alguns desses cliques.

Assim, propomos nesta seção uma abordagem complementar, com base em trabalhos prévios, responsável pela realização da detecção de cliques inválidos cuja prevenção por meio de CAPTCHA clicáveis tenha se mostrado insuficiente. Sob essa perspectiva, escolhemos o algoritmo *Similarity-Seeker*, o qual identifica coligações de fraudadores localizados em diversas localidades do globo para inflar cliques em sites de publicadores específicos. Essa solução é ainda generalizada no mesmo trabalho para coligações de tamanhos arbitrários. Essa proposta apresenta a implementação de um protótipo do algoritmo que detectou, com sucesso, coligações de fraudadores de diversos tamanhos – esta foi a nossa contribuição com relação a esse algoritmo.

A ideia da identificação de similaridades é interessante para a nossa abordagem pois, dentro da sistemática de detecção da fraude, apresenta uma proposta de verificação em diversas camadas envolvidas em um sistema de anúncios *online* – desde a investigação nas camadas de interface com usuários, até o armazenamento dos cliques em um histórico. Na medida em que se analisam diferentes camadas de detecção de um histórico de cliques, passa a ser um passo natural o questionamento acerca da prevenção, sendo o algoritmo *Similarity-Seeker* uma verdadeira inspiração para este trabalho.

A abordagem proposta neste trabalho não tem o objetivo de ser exclusivamente um método de detecção que, ao analisar o histórico de cliques, irá eliminar os cliques inválidos, mas precisa utilizar esse passo como um elemento complementar ao aspecto de prevenção que

pode ser obtida devido ao emprego de CAPTCHA clicáveis. Graças a eles, a distinção entre cliques válidos e inválidos acontece no momento em que os mesmos são realizados, utilizando desafios CAPTCHA e autenticação por cupons para validar os cliques. O passo de validação é realizado em tempo de execução. No entanto, como os sistemas CAPTCHA não garantem 100% de inviolabilidade, o emprego do algoritmo *Similarity-Seeker* mostra-se útil na medida em que realiza a análise do histórico de cliques em um período de tempo posterior à execução, através de filtros heurísticos.

Assim, a premissa básica para detecção são esses filtros heurísticos, que consistem em desenhar uma correlação entre as máquinas que realizaram o ataque, identificadas pelos IPs, *cookies* e padrões de comportamento de fraudadores. Por exemplo, para ludibriar as técnicas de análise de dados, publicadores fraudulentos tentam diluir a correlação forte entre seus sites e as máquinas de onde os ataques são enviados, o que faz com o espectro de ataques de aumento de cliques sejam:

a) Para diluir a correlação do lado da máquina, um ataque não coligado é realizado por um fraudador, que esconde ou muda frequentemente a identificação da máquina ou usa um grande número delas;

b) Para diluir a correlação do lado do site, um ataque coligado é realizado por muitos fraudadores onde muitas máquinas podem ser utilizadas para simular tráfego para qualquer site.

A ideia do *Similarity-Seeker* é tornar-se capaz de detectar ambos os tipos de ataque, de maneira que qualquer atacante no espectro seja detectável e o problema de aumento de cliques seja resolvido. Compreensivelmente, a dificuldade de se detectar um ataque não coligado aumenta conforme o número de máquinas de onde o ataque é enviado aumenta mas

veremos mais a seguir que o algoritmo se mostra sensivelmente eficiente inclusive em tais casos.

Por se tratar de uma abordagem de detecção baseada em *cookies* e endereço IP, o questionamento natural inicial pode ser que um fraudador poderia se utilizar de serviços de navegação anônima, como o tor.eff.org. No entanto, tais ferramentas, embora atrativas para fraudadores inexperientes, não são nada efetivas, já que foram desenhadas para proteger a privacidade de usuários navegando na internet. Ao bloquear os *cookies* do usuário, o uso de anonimato de rede pode ser trivialmente detectado ao se monitorar a porcentagem de tráfego de rede sem cookie por publicador e ao investigar aqueles publicadores cujo tráfego desvia da norma.

De certa forma, embora ambos sejam eficientes, os mecanismos de detecção e os de prevenção têm sua utilidade é limitada contra fraudadores sofisticados, e sua performance pode ser gradativamente prejudicada quando esses fraudadores aprendem como ludibriar tais filtros. Sua utilização conjunta é fundamental para aumentar a confiança do sistema de segurança de fraude de clique como um todo. No entanto, ao analisar aspectos como os expostos nessa seção, pode-se perceber a complexidade de se rodar um ataque altamente escalável e de larga escala. Ataques difíceis de detectar de diferentes origens são caros e pouco escaláveis e para conseguir uma taxa normal de tráfego sem cookie e uma porcentagem normal de IPs de provedores de internet não virtuais, fraudadores precisam controlar máquinas de usuários reais através de Trojans ou ainda possuir máquinas reais espalhadas ao longo do mundo. Em outras palavras, iniciar ataques escaláveis consistentemente implica em custos elevadíssimos ou requer uma habilidade significativa em escrever código Trojan, já que esses últimos são facilmente detectados por softwares antivírus, o que os tornam automaticamente não escaláveis. Assim, a correlação entre a dificuldade de rodar um ataque e a dificuldade de detectá-lo é compreensível.

Para detectar ataques de coligação, a rede de anúncios precisa procurar os sites que clicaram links no publicador apresentando tráfego similar. Um indicativo confiável do tráfego do site é o conjunto de endereços IP. Levando em consideração esse conjunto de IPs distintos, a rede de anúncios precisa pesquisar quais sites possuem entradas geradas pelo mesmo conjunto de IPs segundo um parâmetro aproximado.

Já que o tráfego dos diferentes publicadores estão armazenados no arquivo de histórico, basta escanear esse histórico inicialmente para se obter o conjunto de IPs visitando cada site, e armazenar as semelhanças detectadas em um arquivo separado. Diversos coeficientes devem ser utilizados para calcular os sites cuja similaridade excedem um padrão aceitável, a saber, o coeficiente Dice, o cosseno, e o Jaccard (BELLARE; CANETTI; KRAWCZYK, 1996). Para detectar coligação em pares de sites, os endereços IP precisam ser colocados à parte tais que um par de sites A e B cujos endereços IP possuam um conjunto de similaridades excedendo um padrão ‘s’ com alta probabilidade.

5.7.1 O Algoritmo Similarity-Seeker

Uma vez que tenhamos um método eficiente para estimar as similaridades de grandes conjuntos de IPs visitando os publicadores, inclusive considerando, para um número alto de erros de fronteira e confiança, o número correto de amostras a serem recolhidas. Podemos usar esse método de amostragem para descrever o algoritmo *Similarity-Seeker*.

O algoritmo inicia coletando amostras e usando toda permutação para descobrir todos os possíveis pares que compartilham um endereço IP entre suas amostras empregando três estruturas de dados: *Samples, C* e *H*. *Samples* é um array *D x n* cujas linhas representam as

amostras de IPs, ou seja, os sites, colunas representam as permutações e as células representam sites em permutações. *C* é um conjunto de tuplas de três: dois identificadores de sites e o número de *samples* compartilhadas – *C* mapeia pares de sites com amostras compartilhadas ‘sn’, sendo ‘s’ o limiar de similaridade. Já *H* é uma tabela *hash* de tuplas: a ID da amostra e uma lista de sites que a compartilha.

Uma vez que *Samples* esteja populado, o *Similarity-Seeker* o lê coluna a coluna em pré ordem. Para cada coluna sendo permutada, a tabela *hash* H é construída temporariamente por armazenarem D amostras e suas listas correspondentes de sites. A tabela é também preenchida após uma busca na coluna, e a lista de sites que compartilha cada IP é populada. Uma busca é realizada em H e qualquer lista que contenha um número grande, l, de sites compartilhando uma amostra é descartada. Uma amostra típica dessa natureza é um endereço IP de um ISP (ex; aol.com), que é compartilhado por centenas de computadores. De maneira inversa, para cada par na lista compartilhando uma amostra um elemento correspondente é criado com um número de amostras compartilhadas de 1, e é inserida em *C*, se já não pertencer a C, ou é incrementada de modo contrário. A qualquer momento, se o par incrementado satisfizer o limiar de similaridade s, uma saída *Similarity-Seeker* é gerada. O algoritmo para descobrir todos os pares com similaridade maior que ‘s’ com grau de confiança a e erro E é apresentada na Figura 5.4.

Figura 5.8: Algoritmo *Similarity-Seeker*

```
Algorithm: Similarity-Seeker(Double s, ε, α, Integer l)
begin
  Set <SiteID, SiteID, Integer> C = empty set;
  for (Integer j = 1, j ≤ n, j++){// IPs loop
    //Allocating space for H
    HashTable <IP, SiteList> H = empty hash table;
    for (Integer i = 1, i ≤ |Sites|, i++){// Sites Loop
      //Populating H with lists of sites sharing an IP
      let Site_ID be the SiteID at Samples[i]
      let IP be the IP at Samples[i, j]
      Insert Site_ID into H[IP].SiteList;
    }// end for
    //Incrementing similarity of sites sharing the j^th IP
    for each SiteList SL in H{
      if (|SL| < l){// Sites do not share a popular IP
        for each two sites, A and B, in SL{
          if ((A, B) ∈ C){
            C[(A, B)].Integer ++;
            if (sn − 1 ≥ C[(A, B)].Integer > sn){
              Output (A, B) as similar;
            }
          } else{
            Insert (A, B, 1) into C;
          }
        }// end for
      }
    }// end for
  }// end for
end;
```

Fonte: Metwally, Agrawal, Abaddi (2007)

Em termos de complexidade, além das duas buscas no histórico de tráfego, o algoritmo realiza mais uma busca em *Samples*. Como esse *array* é tipicamente menor que o tráfego inteiro, então pode-se considerar que o tráfego é escaneado três vezes. Assim, para calcular o valor de fronteira na complexidade de memória de processar uma coluna pelo *Similarity-Seeker*, temos que considerar dois casos extremos: o primeiro é quando $l - 1$ sites compartilham a mesma amostra e todos as outras $(D - (l - 1))/2$ amostras são compartilhadas por dois sites. O segundo caso é quando todas as amostras D estão agrupadas para formar listas de tamanho $l - 1$. Então a complexidade de processar uma coluna é $O(\max(l^2+D, lD))$. Já a complexidade de qualquer caso não extremo é claramente uma combinação não linear de $O(l^2 + D)$ e $O(lD)$. Já que o fator lD domina e há n colunas, podemos dizer que a complexidade do algoritmos *Similarity-Seeker* é $O(lD / E^2)$, com uma pequena constante. Na verdade, um pouco

menos que isso devido a pequena similaridade entre endereços IPs visitando os sites e, consequentemente, do pouco agrupamento entre os sites.

5.7.2 Modificação para Coligações de Tamanhos Arbitrários

O algoritmo Similarity-Seeker pode detectar eficientemente coligações de pares de fraudadores em sites. No entanto, fraudadores normalmente formam coligações de bots e botnets que vão além de pares de sites em fraude, o que pode ser levado em consideração para alterar esse algoritmo para se detectar coligações maiores, como veremos nessa seção – e que foi o algoritmo que desenvolvemos.

Digamos que um grupo de atacantes, de tamanho 'Q', torne difícil detectar coligações por não compartilhar todos os recursos ao mesmo tempo. Em vez disso, cada publicador de anúncio compartilharia cada recurso que ele controla apenas com 'q' atacantes aleatórios. Assim, como mostrado pelo Teorema 1, o número de semelhanças dos pares cai, enquanto o grupo ainda ganha em coligações

Teorema 1: Um grupo de publicadores atacantes, de tamanho Q, onde cada pulicador compartilha cada recurso que ele controla com apenas q < Q publicadores aleatórios reduz as similaridades entre os pares de 1 a $\frac{q(q+1)}{2(Q-1)+q(2Q-q-3)}$*, e obtém um ganho de 'q'*

Prova. Assuma cada local no grupo de ataque controla R recursos, o coeficiente de Jaccard é utilizado para a semelhança, e que um recurso compartilhado por A com B não é compartilhada por B novamente para uma terceira parte. Para quaisquer dois sites, A e B, a interseção de Sa e Sb é definida pelo conjunto de recursos compartilhados pelos sites, e os recursos compartilhados pelos outros Q – 2 nós tanto com A quanto com B. Isso pode ser

simplificado pela equação: $2r\frac{q}{Q-1} + r(Q-2)\frac{q(q-1)}{(Q-1)(Q-2)} = r(q+1)\frac{q}{Q-1}$. $Sa \cup Sb$ é dado pelos recursos controlados por A, controlados por B, e os recursos compartilhados por todos os outros Q – 2 nós que contenham tanto A quanto B. Isso equivale à equação: $2r + 2r(Q-2)\frac{q}{Q-1} - r(Q-2)\frac{q(q-1)}{(Q-1)(Q-2)}$. Nesse contexto, o coeficiente de Jaccard pode ser dado por $\frac{q(q+1)}{2(Q-1)+q(2Q-q-3)}$. Os recursos direcionando o tráfego para cada site são dados por r(q+1), após formarem a coligação, ao invés de r recursos controlados por cada site.

Corolário 1: Ao aumentar o tamanho de uma coligação, os atacantes podem sustentar a semelhança entre pares em um nível baixo, enquanto continuam a aumentar seus ganhos.

Prova: Para manter similaridade dos pares abaixo de um nível detectável s, o ganho de cada publicador q, onde q é dado por: $\frac{1}{2(s+1)}(2Q-3)s - 1 + \sqrt{((2Qs+1)^2 - (4Qs^2+1) + (1-s)^2)}$. Assim, similaridade entre pares pode ser sustentada em quaisquer níveis, enquanto o aumento do ganho q, através da formação de grupos maiores, isto é, aumentando Q. Ora, do Teorema 1, se $q > 1$ e $q = \sqrt{Q}$, o coeficiente de Jaccard é menor que $\frac{1}{\sqrt{Q}}$. Felizmente, como mostrado na seção 6.3.1, dois sites quaisquer possuem uma similaridade que pode ser ignorada. Assim, mesmo que atacantes formem coligações muito grandes e largas, a similaridade inerente ainda estará acima da norma e será detectável pelo Similarity-Seeker alterado, cujo pseudo-código deve ser estendido pelo método seletor de amostras, demonstrado no pseudo-código na Figura 5.9 abaixo:

Figura 5.9: Algoritmo Samples-Select: Pseudo-código

```
1. Procedimento Samples-Select (Inteiro n)
2. For(int i=1; i <= n; i++)
3.    // construir permutações
4.    For each Traffic Entry 'e' = (ID_Site, Endereço_IP)
5.          Escreva 'e' no Arquivo de Tráfego do ID_Site
6.    // Selecionando amostras
7.    For each ID_Site {
8.          IP Amostra_Site(n) = array_vazio;
9.          For each Traffic Entry 'e' = (Site_ID, IP) {
10.               For (int j = 1; j <= n; j++) {
11.                    If (Permutação(j)(IP) < Permutação(j)Amostra(j)) {
12.                         Amostra(j) = IP;
13.                    } // fim do if
14.               } // fim do for
15.         } // fim do for
16.   } // fim do for
```

Fonte: Próprio Autor

Esse procedimento será responsável por realizar duas buscas preliminares nas amostras dos dados do tráfego – no primeiro, esse método separa o tráfego de cada site em um arquivo individual que se encaixa na memória. Para cada site, o método toma um segundo passo em que ele analisa toda as permutações e, depois de coletar todas as amostras, o arquivo é escrito de volta em cada linha na matriz de amostra *Samples*. Isso tudo fica armazenado na memória e aumenta a escalabilidade do algoritmo proposto e a complexidade do método é $O(D|S|n)$. A ideia da modificação é adicionar esse passo antes da população das amostras na matriz *Samples*, permitindo, assim, a seleção de amostras de para coligações de diversos tamanhos.

Assim, o algoritmo *Similarity-Seeker* alterado pode ser resumido no pseudo-código demonstrado na Figura 5.10 a seguir:

Figura 5.10: Algoritmo Similarity-Seeker modificado para chamada do Samples-Select, com destaques do autor para as modificações em relação ao algoritmo original

```
1. Procedimento Similarity-Seeker-Modificado (Duplo s, E, a, Inteiro l)
2. //1° passo: Calcular o numero de amostras
3. Integer n = (ka/2E)^2;
4. //2° passo: Chamar o Samples-Select para criar um array de Amostras
5. Samples-Select(n);
6. //alocar espaço para C, a partir daqui o algoritmo segue normalmente, levando em consideração, ao invés de pares de sites, uma coligação de fraudadores
7. Malloc C = conjunto vazio;
8. For(int j=1; j <= n; j++)
9.    // alocar espaço para H // construção da tabela Hash
10.   Hashtable <IP, Lista de Sites> H = hash table vazia
11.   For(int i=1; i <= n; i++)
12.   // popular H com lista de sites que compartilham IP a partir da amostra gerada por Samples Select!
13.   // Seja ID_Site e IP em Samples[i]
14.   Insira ID_Site em H[IP].SiteList;
15.   For each SiteList SL em H
16.         If (SL < l) // ou seja, sites não compartilham um IP popular
17.               For each dois sites A e B em SL{//potencialmente fraudadores
18.                     If (A,B pertencem a C) {
19.                           C(A,B).Integer++
20.                           If(sn-1 >= C(A,B).Integer++ > sn)
21.                                 Saida = A e B são similares
22.                           Else
23.                                 C = Insert(A,B,1) //Insira A e B em C
24.                     } // fim do if
25.               } // fim do for
```

Fonte: Próprio Autor

Aumentar o tamanho das coligações para grandes grupos muda o foco de busca de pares de sites com tráfego altamente semelhante para a procura de grupos com similaridade moderada. Por duas razões principais, a solução na seção 5.7.1 tem de ser estendida para coligações de tamanhos arbitrários. Em primeiro lugar, ao dar como saída um conjunto de diversos sites como similares, de tal forma que cada par de sites têm tráfego semelhante, se estabelece a evidência de intenção maliciosa dos fraudadores. É muito improvável, para qualquer grupo aleatório das unidades de tamanho Q, que todos os pares possíveis são similares. Se os sites de quaisquer dois publicadores "podem ser erroneamente julgado para ter o tráfego semelhante com probabilidade P. Então, julgar erroneamente que os sites aleatórios Q estão envolvidos em um ataque da coalizão tem uma probabilidade de $P\frac{Q(Q-1)}{2}$. Por exemplo, se P = 0.1, então julgar erroneamente se um grupo pequeno de 5 sites aleatórios está ou não em uma coligação tem uma probabilidade de 10^{-10}.

Em segundo lugar, a saída de um conjunto de vários sites é mais concisa do que a saída de todos os pares possíveis nesse conjunto. Por exemplo, se 3 coligações de 50 fraudadores são descobertas, é mais conveniente verificar uma lista de 3 entradas, cada uma de tamanho 50, do examinar 3 x 50 x 49 / 2 = 3225 saídas, cada uma com tamanho 2. Além disso, a coesão da saída dá uma visão mais correta e geral da coligação e facilita as investigações manuais, como a verificação de características comuns entre os sites, como data do contrato e taxa de ganho.

Assim, é necessário condensar os pares detectados em grupos de sites. A similaridade pode ser modelada como um grafo não direcionado cujos nós representam sites, e cujas arestas representam pares de sites com tráfego similar. O objetivo do algoritmo passa a ser buscar, ao invés de apenas arestas, todos os cliques máximos nesse grafo crescente.

5.8 Considerações do Capítulo

Em um mundo de perfeita transparência, no qual uma rede de anúncios saberia a identidade real de todos os usuários que clicam em seus anúncios, a fraude de clique seria muito mais fácil de identificar. Nesse cenário, seria muito mais fácil gerenciar o mau comportamento de um dado usuário – por exemplo, a repetição de cliques em um mesmo anúncio – ou os cliques que foram iniciados por usuários maliciosos ou por *bots* seriam de fácil identificação. Uma rede de anúncios poderia ir ainda mais adiante e referenciar bancos de dados contendo perfis de usuários que clicaram em seus anúncios, e talvez criar uma estrutura de preços de cliques altamente refinada, baseada no valor que um determinado usuário potencialmente gastaria, ou ainda criar uma compensação diferenciada, baseada em reputação, para publicadores que consigam gerar cliques de determinados usuários. O esquema proposto nesta seção difere desse mundo ideal em dois sentidos básicos:

1- Conhecimento parcial: dada a natureza fragmentada dos bancos de dados sobre comportamento de usuários, e as políticas de privacidade normalmente difundidas por grupos de regulação, o conhecimento geral dos padrões de cada usuário é modesto, restringindo-se a possibilitar à rede de anúncios o conhecimento de que um determinado clique foi originado por um humano com intenções provavelmente honestas. Não é foco deste trabalho uma diferenciação forte entre usuários, embora os protocolos de comunicação pudessem ser utilizados para suportar este objetivo.

2- O navegador como meio de transporte: ao invés de contar com um repositório central de dados, este esquema confia nos navegadores dos usuários para compartilhar informação entre as entidades participantes. Esta abordagem ajuda eliminar complexidade técnica e protege a privacidade dos usuários.

A princípio, a abordagem aqui proposta poderia suplantar completamente todos os esquemas atuais e tradicionais de segurança de pagamento por clique, já que incorpora os mecanismos atuais de detecção e os estende para incluir um aspecto valioso, o da prevenção

contra cliques automatizados, problema que, de acordo com Ciciliano (2015), constitui até 85% dos cliques fraudulentos atuais em redes de anúncios *online*. Assim, utilizá-la simultaneamente a um esquema tradicional, como uma forma de complementação, parece ser a opção mais indicada. Em todo caso, a abordagem aqui proposta certamente poderia ser lançada de forma experimental por alguma rede de anúncios com impacto mínimo nos negócios atuais e, na medida em que obtivesse garantias de sucesso, ser expandida. Esta abordagem não só é uma nova visão e paradigma para o combate da fraude de clique, mas provê também uma grande oportunidade de controlar essa fraude.

6 Resultados e Validação

Como resultado da proposta de abordagem acima, foi desenvolvido o protótipo de rede de anúncios online que denominamos de *ClickSafe Project*, e pode ser encontrado no site http://www.clicksafe-project.com. Conforme o diagrama de componentes apresentado na seção 5, o mesmo tem o propósito de ser uma ferramenta de exibição de anúncios simples de maneira segura. A parte de segurança segue a arquitetura proposta no Capítulo 5, com autenticação por Google NoCAPTCHA reCAPTCHA e validação de cupons. A simplicidade dos anúncios segue a crença de que links no formato texto são efetivos e muito menos irritantes, é baseada no modelo PPC, em um modelo de leilão às escuras, no qual os anunciantes competem entre si para a exibição de determinadas palavras-chave, cuja ideia baseia-se na percepção de que, na medida em que uma palavra é mais comum, mais valor é necessário para que determinado anúncio apareça após uma busca. Um benefício desse tipo de estrutura de anúncios é que normalmente é bem mais relevante, já que o anúncio exibido será relacionado com um padrão de navegação do usuário, tais como buscas realizadas em ferramentas de pesquisa online, ou ainda devido a sites recentemente visitados.

A porção de *software* desse protótipo foi desenvolvida em linguagem PHP e fez uso dos recursos CSS e Javascript para facilitar o processamento de imagens e a experiência dos usuários no uso dos serviços. A hospedagem se deu no serviço UOL Host (http://www.uol.com.br/uolhost), sob um ambiente de sistema operacional Linux e banco de dados MySQL. O sistema foi testado nos navegadores Google Chrome, Mozilla Firefox e Internet Explorer em suas versões para PC, e para dispositivos móveis nos navegadores Google Chrome e Safari, nos sistemas operacionais iOS e Android kitkat e jellybean.

Até o momento, a solução concentra seu foco em aspectos operacionais para inviabilizar a fraude de clique em cima de um cadastro pré-existente. Em outras palavras, até o

momento dessa proposta de tese, não há interface do tipo “painel de controle”, e o cadastro de clientes e anúncios tem que ser feito manualmente e diretamente no banco de dados. Portanto, não desenvolvemos heurísticas para exibição de anúncios em função de geolocalização, de busca em ferramentas online ou por contexto, como referenciados no Capítulo 2. Todos esses pontos podem ser tópicos para trabalhos futuros e gerariam individualmente uma pesquisa como esta.

Trata-se também de uma interface para testes que, com a exceção do uso do CAPTCHA clicável Google NoCAPTCHA reCAPTCHA, não aprofundou aspectos de interface gráfica, carecendo, até o momento, portanto, de um estudo e um diagnóstico para a escolha de uma interface adequada para o seu uso. O diferencial desta solução está, portanto, no fato de introduzir um novo passo de dificuldade significativo no sentido de combater a fraude de clique.

6.1 Experimento e Validação

Para a porção de experimento, foi utilizada uma configuração semelhante à recomendada por Metwally, Agrawal e Abaddi (2007), que avaliaram a solução usando um *stream* de dados reais coletados de uma empresa da ValueClick. No nosso caso, conseguimos uma *stream* de dados real da empresa Fastclick, Inc. Na fase preliminar dos experimentos, foi utilizada uma combinação de IP do usuário com o ID do anúncio como o ID do clique. Ambos os componentes foram representados como inteiros de 32-bit e, dessa forma, pode-se dizer que o ID do clique foi representado como inteiros de 64-bits. A ideia é que, se os cliques em um mesmo anúncio, em um curto período de tempo, vierem de um mesmo endereço IP mais de uma vez, isso deve revelar alguma forma de fraude no site do publicador. A diferença é que o

foco do experimento dos pesquisadores era na detecção desse tipo de clique, enquanto nosso foco se divide em duas fases: inicialmente elaboramos um programa para tentar quebrar o CAPTCHA, explorando e validando o aspecto da prevenção através da nossa abordagem, e depois elaboramos um *script* para que realiza a fraude de clique com base nos dados reais apresentados anteriormente uma vez que uma autenticação manual tivesse sido realizada, validando assim o aspecto de detecção da abordagem.

O interesse precípuo dos experimentos consistia em testar o aplicativo ao impedir que cliques inflados e automatizados e, portanto, não autenticados, fossem registrados. Os tipos de experimentos conduzidos foram de duas naturezas distintas:

1) Carga automática de cliques, em que *scripts* automatizados tentavam quebrar o sistema utilizando um *software* de reconhecimento de imagens, previamente escritos para busca imagens de CAPTCHA clicáveis similares como forma de resolvê-las. Esses softwares foram utilizados segundo a problemática de reconhecimento de padrões descrita por Xiaoguang (2003) e Xu (2013), que consiste na busca de padrões similares em uma base de dados preexistente de imagens categorizadas e, assim como em Nachi (2014) e Danchev (2015), foram utilizados como parte inerente do código; ou seja, adaptamos as heurísticas existentes de reconhecimento de imagens para a solução para possibilitar o experimento; e

2) Carga automatizada após autenticação manual de cliques, que consiste em testar o cenário em que um usuário humano autentica o Google NoCAPTCHA ReCAPTCHA manualmente e depois rodar um script automatizado para realizar a fraude de clique enquanto o cupom estiver válido. Na verdade, por questões práticas, para validar esse cenário em larga escala foi utilizada uma carga de cliques real de uma organização que não está mais ativa no mercado.

Foi conduzida uma carga de experimentos para avaliar a eficiência e escalabilidade da solução proposta. A principal motivação desta validação é demonstrar que esse modelo é capaz de prevenir, no cenário 1, e detectar, no cenário 2, a fraude nas redes de anúncio online, que representam o homem no meio entre publicadores e anunciantes de Internet. Assim, a configuração dos dois experimentos se deu seguindo essa lógica: no primeiro, a ideia consistia em garantir que a rede de anúncio conseguiria distinguir se os cliques gerados no publicador eram autênticos, ou gerados por um *script* automático (não humanos) no lado do publicador para gerar mais tráfego e, consequentemente, mais receita; no segundo, possibilitamos uma autenticação manual temporária do CAPTCHA clicável e rodamos um script automático para inflar a taxa de clique de determinados anúncios – inicialmente para verificar a escalabilidade da solução de cupons e o algoritmo de detecção numa verificação posterior. Utilizamos um cenário do mundo real para realizar esse experimento, em que o *Similarity-Seeker* foi utilizado para detectar coligações de atacantes de tamanhos arbitrários. Para atestar a validade e confiabilidade do nosso desenvolvimento, comentamos em um conjunto compreensível de experimentos utilizando dados reais de uma operação da empresa Fastclick, Inc, que esteve ativa como rede de anúncio até meados de 2011.

6.2 Caso 1: Testando o Sistema com Software de Reconhecimento de Imagens

Em 3 de dezembro de 2014, o Google apresentou um CAPTCHA para simplificar o processo de verificação humana e que continuaria tendo o objetivo de prevenir *bots* de realizarem atividades tais como se registrar automaticamente em serviços na *web* ou realizar comentários automáticos nos sites disponíveis. Já explicamos o Google NoCAPTCHA reCAPTCHA em detalhes na seção 5.4.1, bem como sua adoção como parte desta solução no início deste Capítulo 6.

A ideia por trás da solução do Google é simples: o usuário vai apertar no *checkbox* e dizer que não é um robô. Se o algoritmo desconfiar que o usuário não diz a verdade, um *popup* adicional vai aparecer e um reCAPTCHA vai pedir que ele responda um desafio para confirmar e provar que ele é um humano. Sabemos que um reCAPTCHA típico é um desafio do tipo texto – mas, para tornar os desafios amigáveis para dispositivos móveis, a plataforma de CAPTCHA clicável também foi disponibilizada, onde a ferramenta pede que imagens semelhantes sejam identificadas umas com as outras, por exemplo, selecione os gatos em um grupo de gatos, cachorros, aves, carros e outros.

Adaptamos o esquema de autenticação e prova acima para a nossa abordagem, levando em consideração a facilidade de configuração do mesmo, de maneira que no nosso protótipo de rede de anúncios, sempre que um usuário acessa um *link* de um anúncio pela primeira vez, um desafio será exibido. Para tornar a taxa de sucesso de resposta para os usuários o mais amigável possível e mais difícil para *bots* (BINDU, 2015), escolhemos a abordagem de CAPTCHA clicáveis ao invés de um reCAPTCHA típico para resposta. Como parte do experimento a seguir, verificamos que todo o tempo de resposta e autenticação não excedia 3 segundos e é pouco oneroso para o usuário humano – testes semelhantes e taxas de sucesso parecidas foram realizados e encontradas por Bindu (2015) e Zhu et al (2014).

Na nossa arquitetura, como já explicado anteriormente, ao responder corretamente, o usuário recebe um cupom de autenticação na forma de *cookie* que tem tempo de validade de 10 segundos – durante esse período, ele pode utilizar o sistema da rede de anúncios sem precisar responder um novo desafio. A ideia do tempo de validade é impedir a ativação de uma fraude de clique do tipo do publicador, que é explorada no exemplo a seguir.

O caso de teste que descrevemos aqui consiste em tentar quebrar o passo resposta do CAPTCHA clicável de maneira automática através de softwares de reconhecimento de padrões (XU, 2013). Se um atacante consegue escrever um software dessa natureza para responder o CAPTCHA de maneira eficiente, a segurança do sistema ficaria comprometida, já que a fraude poderia facilmente ser automatizada sempre que o desafio fosse exibido – bastaria o *software* responder o desafio e continuar realizando a fraude normalmente.

Para isso, pesquisamos algumas ferramentas para resolver CAPTCHA do tipo clicável por meio de reconhecimento de imagens e padrões e as melhores candidatas foram GSA CAPTCHA Breaker (http://scraping.pro/CAPTCHA-breaker-review/), DeCaptcher (http://scraping.pro/decaptcher-review/) e a UBot (http://devmd.com/r/bypassing-no-CAPTCHA-reCAPTCHA-with-ubot). Levando em consideração testes iniciais realizados, a UBot pareceu resolver numa taxa melhor o Google NoCAPTCHA reCAPTCHA, então selecionamos o código dessa ferramenta para adaptar uma porção de software para quebrar o sistema.

O UBot é uma ferramenta automatizada desenvolvida colaborativamente que permite automatizar tarefas como criação de contas, postar em diversos sites ao mesmo tempo e outras atividades que tipicamente levariam muito tempo. É possível criar *scripts* dentro da mesma para automatizar serviços e ele funciona muito bem para quebrar a parte de texto do reCAPTCHA, mas a taxa de sucesso com a parte clicável chega a ser semelhante com força

bruta, tendo resolvido na fase de testes 57 desafios em 31.021 tentativas. O GSA Breaker resolveu apenas 12 desafios em 42.231 tentativas e o DeCaptcher resolveu 31 em 22.576.

Após obter o código-fonte do UBot em http://devmd.com/jump/ubot, foi possível escrever uma porção de código que basicamente acessava o site do *ClickSafe* (www.clicksafe-project.com), e tentava cometer a fraude automatizada através de *link* direto de um dos anúncios. Ao receber um desafio, a função correspondente em PHP ativava o serviço de resolução, e voltava a inflar o clique no anúncio. Como não estávamos testando o algoritmo de detecção, não houve preocupação com ataques de múltiplas fontes ou de esconder o IP, a ideia era apenas ignorar a verificação do CAPTCHA e tentar cometer a fraude. Tratava-se, portanto, de uma porção de *software* bem simples, explicitada no pseudocódigo referenciado na Figura 6.1:

Figura 6.1 – Algoritmo de chamada do teste UBot

```
1. while (true) {
2.       delay(200)
3.       acesse www.clicksafe-project.com/walmart/bicileta1.php
4.       se g-reCAPTCHA-response é necessária {
5.          g-reCAPTCHA-response = Javascript:UbotGetReCAPTCHA()
6.       }
7. }
```

Fonte: Próprio Autor

De maneira geral, a ideia implementada pelo UBot consiste no chamado esquema do modelo de aparência, descrito por Xiaoguang (2003), que utiliza técnicas do tipo reconhecimento análise linear subespacial. Esse esquema gera um modelo a partir de imagens para comparação, e pode utilizar bases de dados de domínio público para isso – há diversos repositórios de imagens disponíveis, tais como a http://www.publicdomainpictures.net, que selecionamos para as imagens de gatos. Para realizar a análise linear subespacial, que é considerado na verdade um método clássico, mas eficiente de reconhecimento de padrões (XU, 2013), são utilizados vetores básicos de classificação de aparência.

De acordo com Xiaoguang (2003), abordagens baseadas em aparência representam um objeto em termos de visão, e uma imagem é considerada um elemento em vetor de múltiplas dimensões, sendo cada dimensão uma imagem, ou seja, um elemento no vetor, ou ainda um ponto em um vetor de espaços altamente dimensional. A abordagem usa, portanto, técnicas estatísticas para analisar a distribuição dos vetores da imagem em questão no chamado espaço de imagens, para derivar então uma representação efetiva e eficiente de acordo com as diferentes imagens anteriormente analisadas. Ou seja, a técnica consiste em projetar vetores sobre as imagens e atribuir notas em função da sua disposição após sua projeção nos chamados coeficientes de projeção, que são usados como a representação das particularidades de cada imagem.

O detalhamento da técnica para o cálculo do reconhecimento de padrões pode ser encontrado em Xiaoguang (2003) e foi validado por Xu (2013), e consiste na manipulação de três classificadores lineares, PCA (*Principal Component Analysis*), ICA (*Independent Component Analysis*) e LDA (*Linear Discriminant Analysis*). Cada classificador tem sua própria representação, denominado "vetor base" e, ao projetar o "vetor face" sobre esses classificadores, obtêm-se, de maneira simples os chamados coeficientes de projeção, que são usados atribuir a pontuação matemática de cada imagem. Todas as três representações podem ser consideradas como uma transformação linear do vetor da imagem original através da "fórmula" genérica $Y = W^T X$, que será detalhada mais a seguir, sendo W^T é a matriz de transformação, que conterá em tempo de execução as imagens a partir de um desafio CAPTCHA que serão preparadas, como um vetor multidimensional, para comparação e X é o vetor com o conjunto de imagens que gera a imagem padrão.

A PCA realiza uma redução de dimensão para achar vetores que melhor realizam a distribuição de imagens dentro do espaço para gerar uma imagem modelo. Em outras palavras,

o primeiro passo do algoritmo é selecionar imagens do repositório de imagens previamente conhecidas e gerar uma imagem base para comparação. Todas as imagens do conjunto são projetadas na imagem alvo para encontrar um conjunto que descreve a contribuição de cada vetor nesse "espaço face". Como, para os testes realizados, selecionamos os CAPTCHA que requisitavam a identificação de imagens de gatos, o algoritmo gerou o rosto padrão de um gato, alimentado a partir da base mencionada anteriormente, e pode ser encontrado na Figura 6.2:

Figura 6.2 – O rosto padrão de um gato

Fonte: Algoritmo UBot, gerado a partir da base http://www.publicdomainpictures.net

O procedimento chave no PCA é um processo estocástico denominado transformação Karhumen-Loeve (XIAOGUANG, 2003), que consiste em reduzir um espaço de dimensão 'n' para vetores em um espaço dimensional.

Por sua vez, o ICA é semelhante ao PCA, mas a distribuição dos componentes é não gaussiana. O trabalho do ICA acontece na entrada, ao separar os elementos de alta ordem, tais como ruído, distorção e outros elementos não necessários na imagem e selecionar os elementos de segunda ordem que foram utilizados no PCA. A ideia do ICA é preparar a entrada para a comparação de vetores base juntamente com aqueles gerados no PCA. Na Figura 6.3

pode-se ver como o ICA prepara as imagens de um CAPTCHA para a comparação posterior com a imagem base, que também será sujeita à transformação, demonstrada na Figura 6.4.

Figura 6.3 – Resultado da aplicação do ICA em parte de um CAPTCHA gerado na aplicação (Parte de uma matrix W^T)

Fonte: Algoritmo UBot, gerado a partir da seleção de um CAPTCHA

Figura 6.4 – Transformação do rosto padrão de um gato depois da aplicação do ICA

Fonte: Algoritmo UBot, gerado após a aplicação do ICA

Finalmente, o objetivo na LDA é representar o espaço de vetores de maneira interessante e realizar as comparações necessárias por meio da pontuação de coeficientes. No nosso caso, de maneira geral, a LDA utiliza os vetores como classificadores lineares para caracterizar as duas classes de objetos. A combinação resultante pode ser usada como um classificador linear adicional, como uma outra forma de redução de dimensão ou, como

utilizado no algoritmo aqui, para análise das similaridades entre as imagens em função de uma imagem padrão (XU, 2013).

A LDA pontua de acordo com o coeficiente na Figura 6.5, denominado W_{LDA} resultante, para cálculo de similaridade:

Figura 6.5 - W_{LDA} resultante, coeficiente para cálculo de similaridade

$$W_{LDA} = \arg\max_{W} \frac{W^T S_B}{W^T S_W}$$

Fonte: Xiaoguang (2003)

Na expressão descrita na Figura 6.5 acima, S_B é denominada de matriz de dispersão entre as duas imagens, e S_W é a matriz de dispersão entre classes de imagens. Elas são definidas por duas expressões na Figura 6.6:

Figura 6.6 – Matrizes de dispersão para o cálculo de similaridades de imagens no LDA

$$S_B = \sum_{i=1}^{c} N_i (\mu_i - \mu)\ (\mu_i - \mu), S_W = \sum_{i=1}^{c} \sum_{x_k \in X_i} (x_k - \mu_i)\ (x_k - \mu_i)$$

Fonte: Xiaoguang (2003)

Na expressão descrita na Figura 6.6 acima, Ni é o número de imagens que geraram a imagem modelo (no nosso exemplo, a Figura 6.2, a face padrão de um gato), c é o número de classes distintas, μ_i é o vetor médio contendo imagens pertencendo à classe i e X_i representa o conjunto de imagens pertencendo à classe i.

Assim, em linhas gerais, podemos resumir o algoritmo de que tentava quebrar os CAPTCHA clicáveis nesta abordagem como um algoritmo de reconhecimento de padrões, conforme explicitado na Figura 6.7:

Figura 6.7 – Pseudocódigo para reconhecimento de padrões de imagens na validação de CAPTCHA clicáveis desta abordagem

```
1.    Para o CAPTCHA, gere a matriz W^T

2.    Em W^T, realize um conjunto de testes rápidos para achar os animais na
figura, tais como reduzir ruído e paisagens

3.    Acesse a base de dados de domínio público, gere o vetor X

4.    Aplique PCA(X)

5.    Aplique ICA(X) e ICA(W^T)

6.    Calcule W_LDA para cada imagem

7.    Retorne as três imagens com melhor resultado como resposta para o
CAPTCHA
```

Fonte: Próprio Autor

Conduzimos experimentos dessa natureza em duas ocasiões, nos dias 20 de dezembro de 2015 e nos dias 13 de janeiro de 2016. Usamos a carga sintética de 1.000.000 tentativas de cliques na primeira vez e de 200.000 tentativas na segunda. As tentativas são todas distintas, ou seja, embora seriam no mesmo anúncio, para evidenciar a fraude, não há duplicação de dados. Em ambas as cargas de experimentos, os resultados foram parecidos, tendo sido registrados uma taxa de sucesso semelhante: 162 cliques foram registrados na primeira carga de experimentos, mostrando uma taxa de falha do sistema de cerca de 0,016% e 38 cliques na segunda, o que deixou a taxa de falha em 0,019%, números não muito distantes àqueles obtidos por Elson (2007) que obteve taxas parecidas há 13 anos atrás quando, em teoria, os *softwares* OCR estariam em um estado muito inferior de desenvolvimento.

São diversos os argumentos (HOMAKOV, 2014), (NACHI, 2014) que explicam que reconhecimento de padrões não é a melhor opção para quebrar a parte um CAPTCHA clicável como o Google noCAPTCHA reCAPTCHA, motivo pela qual decidimos adotá-los em e utilizá-los na nossa arquitetura, devido às taxas de sucesso que encontramos acima. A razão para isso é bem simples: além de não existirem muitos *softwares* especializados no assunto, de maneira geral, computadores ainda não são bons em distinguir imagens "similares" (XU, 2013), e os resultados dos testes acima mostram isso, o que comprova a efetividade da solução apresentada para a fraude de clique automatizada, que era o objetivo principal desta pesquisa.

Não podemos ignorar, no entanto que, para contornar esses casos, os atacantes criarão soluções alternativas, por exemplo através de usuários humanos, como subempregos em países subdesenvolvidos nos quais pessoas respondem desafios em troca de valores irrisórios, ou através da abordagem chamada *click hijacking* (NACHI, 2014), que acontece quando um usuário de um site de larga escala, como um site pornô, tem sua atividade interrompida até que ele resolva um desafio. Nesse tipo de cenário, os aplicativos têm a responsabilidade de meramente recolher o código javascript do desafio, mostrar para outro humano responsável pela resposta, que estará trabalhando no momento, ou realizando uma atividade de alto grau de motivação pessoal, a qual ele tem interesse de continuar realizando o mais rapidamente possível. Esse caso se enquadra como de autenticação manual que validamos na seção 6.3, a seguir.

Concluímos, portanto, que quebrar CAPTCHA clicáveis de maneira automática para realizar a fraude de clique, como era hipótese desta pesquisa desde o princípio é uma opção que deixa muito a desejar, já que apresentou taxas muito baixas de sucesso, como vimos acima.

6.3 Caso 2: Autenticação Manual

Conforme já explicado anteriormente, o esquema geral da autenticação manual consiste em utilizar usuários humanos para responderem o CAPTCHA, se autenticarem e posteriormente habilitarem um *script* para ser realizar a fraude automaticamente durante o tempo de validade do cupom. A ideia é seguir nesse ciclo de autenticação e execução de *script* continuamente enquanto a fraude persistir.

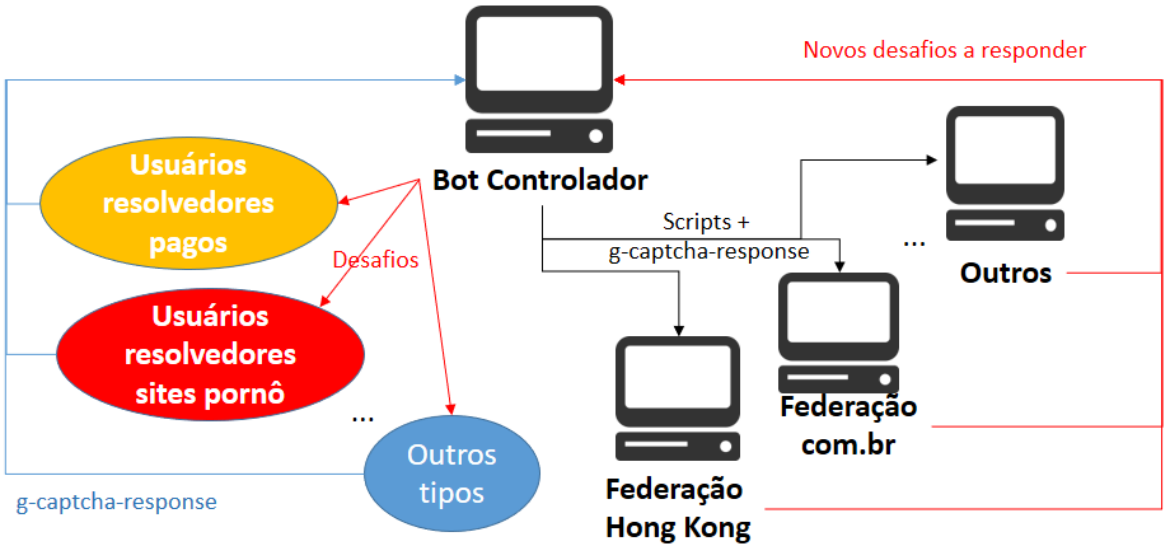
Após as primeiras interações, percebemos que, graças ao tempo de expiração do *cookie*, a interação entre os usuários autenticando os desafios e a taxa em que os *scripts* automatizados são reiniciados precisa ser muito intensa em um sistema de larga escala se a fraude fosse realizada de forma inteiramente manual – isto é, o atacante resolveria diversos CAPTCHA, iniciaria diversos *script* para rodarem em diversas máquinas dispersas geograficamente e quando os cupons fossem expirando, reiniciaria o processo. Imaginando esse processo para uma federação de *scripts*, o processo seria extremamente moroso e impraticável, além de custoso quanto à questão de retrabalho. Vemos isso como um ponto positivo para a solução, ao adicionar uma dificuldade que inviabiliza o que normalmente se pratica em fraudes dessa natureza (NACHI; 2014), (HOMAKOV, 2014) e (TUZHILIN; 2006), que consiste em autenticar e permitir a livre utilização do sistema em questão. No nosso caso, por se tratar de um mecanismo de autenticação de curto prazo, até pela própria natureza da aplicação, uma nova autenticação não é apenas necessária como recomendada.

Quanto à fraude de clique de baixa escala automatizada sob o mesmo esquema, ela até pode ser realizada sem grandes dificuldades. Quando o usuário responde ao CAPTCHA, ele pode rodar na mesma máquina, pelo período de validade do cupom, um *script* para inflar os cliques em determinado grupo de anúncios. No entanto, podemos dizer intuitivamente que essa abordagem da fraude deixa tantas similaridades, tais como endereço IP, MAC, ID de sessão,

cliques duplos e suspeitos, que por si só é inviabilizada pela nossa solução, o que faz sentido levando em consideração todos os aspectos de segurança nela envolvidas.

Para se aproximar então de uma abordagem de validação de larga escala do mundo real, propomos então a modificação do esquema proposto por Nachi (2014), denominada *clickjacking* ou *UI redress attack* (CHO et al, 2015). Pensando nesse caso, digamos que um *bot* queira ignorar o Google NoCAPTCHA reCAPTCHA de um site sem deixar um utilizador da rede de anúncios ficar sabendo que ele está permitindo um *bot* fazer isso. O *bot* poderia então controlar todo o processo e agir como mediador tanto da apresentação do CAPTCHA para o "usuário resolvedor manual", que poderia ser tanto um subempregado em um país subdesenvolvido quanto um usuário acessando um site pornô ou algo semelhante, quanto também seria responsável por ativar o script uma vez que o CAPTCHA estivesse resolvido pelo usuário humano através da resposta do *data-sitekey*. O esquema pode ser visualizado na Figura 6.8 a seguir.

Figura 6.8 – Modificação do *Click hijacking* para fins de validação da solução proposta



Fonte: Próprio Autor

Uma vez que o usuário resolve o CAPTCHA, a resposta, denominada "g-reCAPTCHA-response" pode ser utilizada pelo *bot* que roda no background para submeter um formulário (*form*) para qualquer website. Dessa forma, o *bot* pode enganar o Google e fazê-lo pensar que resolveu o desafio que foi originalmente apresentado pela rede de anúncios *online*. Isso vai fazer com que o Google "autorize" o *ClickSafe* a disponibilizar um cupom válido ao *bot* por 10 minutos, quando, durante esse tempo, ele poderá realizar a fraude de clique automatizada. Isso só funciona porque o Google não valida se o cabeçalho do referenciador (*Referer*) foi desabilitado pelo cliente, ou se está vazio. De certa forma, nesse cenário o usuário é utilizado pelo *bot* como um cartão de acesso.

Para nós, seria interessante obter dados reais para tentar identificar o comportamento de fraudadores reais e inviável gerar uma carga de testes capaz de simular o cenário acima. Por questões práticas, conseguimos um *stream* de cliques reais e históricos junto

à organização Fastclick Inc., que não está mais ativa no mercado e que operava até 2011 como rede de anúncios. Os resultados dessa validação foram rodados em uma carga de testes que aconteceu em 12 de dezembro de 2015. A amostra consistiu em 5.583.301 cliques, dos quais 4.093.573 eram distintos e 1.489.728 eram cliques duplos, e poderiam ser considerados como fraudulentos e gerados como *clickjacking*. Isto é, 27% dos cliques seriam considerados fraudulentos, o que é uma taxa extremamente alta e de maneira interessante, o elemento de maior duplicação, ou seja, o anúncio que foi mais clicado, ocorreu 10.781 vezes pelo mesmo usuário script, no mesmo dia. Para reduzir o ruído, eliminamos todos os IPs que visitaram um grande número de sites, pois os mesmos poderiam fazer parte de endereços do tipo NAT ou ISPs que compartilham IP entre os seus usuários.

Levando em consideração o exposto na seção 5.6, ao se analisar os logs de histórico após a fraude ter ocorrido, na fase de detecção, repetimos a análise do algoritmo *Similarity-Seeker* variando o parâmetro *l*, que considera o número de sites além dos quais os IPs são desconsiderados, de 10 a 1000. Os resultados são discutidos detalhadamente na seção a seguir, 6.3.1.

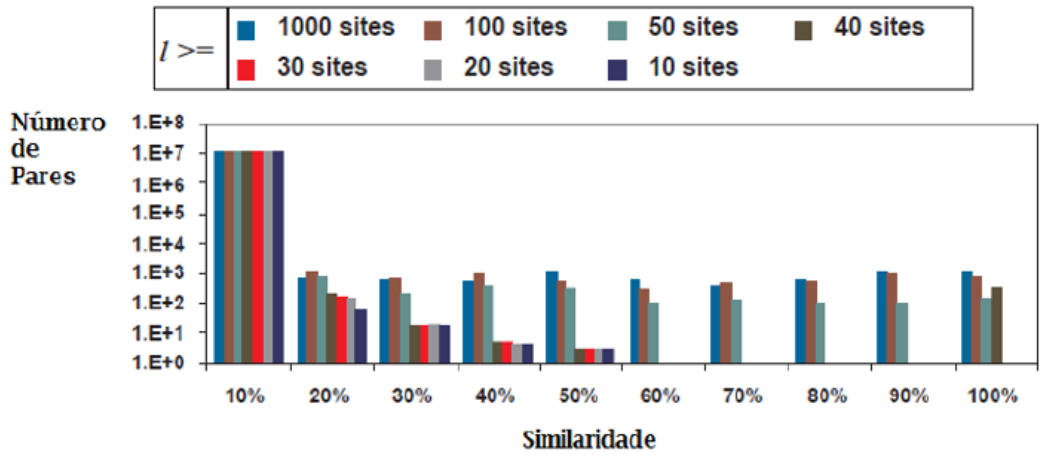
6.3.1 Validação do Similarity-Seeker após Autenticação Manual

Na verdade, a validação do algoritmo *Similarity-Seeker* não era muito o objetivo desta pesquisa, já que o mesmo já havia sido validado e atestado em bancas e pesquisas anteriores (METWALLY, AGRAWAL, ABADDI, 2007), inclusive com cargas de testes bem superiores à realizada por esta pesquisa. No entanto, diante do cenário exposto na seção 6.3.1, decidimos extrapolar a carga de testes anteriormente realizada para validar também o cenário modificado que apresentamos.

Os cliques duplos seguem um ciclo de identificação muito simples do *Similarity-Seeker* e são o grau mais simples de similaridade existente, como já descrevemos. O algoritmo simplesmente considera cliques duplos como erro do usuário e os desconsidera para questões de contabilidade. Ao ajustarmos aspectos de ruído, imediatamente cerca de 25% dos cliques foram identificados como fraudulentos. No entanto, a busca de similaridades não parou por aí.

É interessante notar que quando *l*, ou seja, o número de sites além dos quais os IPs foram desconsiderados, estava configurado para 1.000, 98,84% dos pares tinham menos que 1% de similaridade. Os resultados estão todos demonstrados na Figura 6.3, com uma escala logarítmica no eixo vertical. Quando *l* estava configurado para 10, 99,98% dos pares tinham menos que 1% de similaridade também. Claramente, a similaridade é irrelevante e, para cada rodada, o algoritmo dá como saída de retorno todos os pares de IP com similaridade maior que 10%, os quais são marcados no banco de dados para investigação posterior. A ideia é que, com esses dados em mãos, os fraudadores sejam expostos e coligações possam ser descobertas.

Figura 6.9 – Variação do Número Pares de IPs com Similaridade Específica Excluindo IPs compartilhados por l ou mais sites



Fonte: Próprio Autor

Nesse caso, quando *l* foi configurado para 10, a saída do *Similarity-Seeker* após o experimento apontou 81 pares com similaridade de fator 's' 0,1. Os 81 sites continham 5 coligações de "tamanho" 3, o que significa dizer que consistiam de 5 computadores distintos com algum grau de semelhança (endereço IP, ID do host, ID da máquina) rodando *scripts* automatizados que buscavam inflar os resultados de três publicadores distintos. Na medida em que os valores de *l* aumentavam, mais pares foram sendo descobertos – por exemplo, quando o valor de *l* virou 30, 189 pares de sites foram achados. Na medida em que mais IPs que compartilhavam componentes disjuntos, e o algoritmo foi capaz de entender esses componentes como coligações maiores – muito embora, na verdade, muitos deles fossem simplesmente cliques sobrepostos, outro efeito típico de cliques duplos, de fácil detecção.

Quando o número foi aumentado para 100, 647 pares foram encontrados. Tornou-se claro que os cliques descobertos eram sobrepostos e com uma taxa de ruído muito alta, que acontece quando endereços IP populares são compartilhados por muitos sites legítimos que não foram suficientemente descartados. Na medida em que aumentamos o valor de *l*, mais pares legítimos de cliques foram reportados como similares mesmo sem seres fraudadores, já que sites de IPs populares compartilhados por esses sites passaram a ser considerados. Para resolver esse problema, é necessário aumentar outros parâmetros tais como o tráfego mínimo requerido para reportar similaridade '*s*' na medida em que se aumenta *l*.

Assim, de maneira a reduzir o ruído, aumentamos gradualmente a taxa de similaridade para além de 0,1, até que ela chegou a 0,5 e a saída do *Similarity-Seeker* foi comprimida para 406 pares de IP que foram traduzidos a uma coligação de 29 clientes com 100% de similaridade. Decidimos então analisar o banco de dados de cliente para essa detecção e descobrimos algumas informações interessantes.

A maior parte desses (93%) se inscreveram na rede de anúncios na mesma data. Mesmo tendo o tráfego originário de IPs diferentes, o tráfego tinha um padrão de navegação. Depois de investigações adicionais, o campo "*Referer"* das requisições HTTP tinha origem de páginas que não continham anúncios da rede de anúncios e algumas vezes nenhum anúncio ou anunciante sequer. A suspeita é que os atacantes possuíam os links prontos através de uma rede de *robots* ou *Trojans* e quando a rede de anúncios enviava os e-mails de ativação, os atacantes simplesmente obtinham os segredos de ativação e os utilizavam em máquinas comprometidas nos domínios. Isso significa que as mesmas tinham origem unicamente em computadores de coligação, que acessavam os anúncios por meio dos *scripts* ativados após a autenticação manual ou inscrição no serviço.

Essa foi sem dúvidas uma detecção perfeita de um ataque de coligação, como podemos perceber na descrição do algoritmo na seção 5.7.

Em um mundo real, tal configuração poderia ser semelhante a uma rede de *Trojans* trabalhando em função de diversos domínios diferentes. Ou seja, os usuários eram humanos utilizando redes de anúncios infectados com *Trojans*, que estavam programados para atacar a rede de anúncios em *background*. Na verdade, o código dos mesmos poderia ser desenhado para atacar redes de anúncios diferentes ao mesmo tempo, já que os usuários navegam sem perceber o padrão de cliques em *background*. É interessante notar que a nossa abordagem parece menos exposta a esse tipo de ataque por causa da dupla abordagem: os usuários infectados precisariam responder os desafios esporadicamente para continuar os ataques automatizados em *background* e, como demonstrado nesta seção, o algoritmo *Similarity-Seeker* é um complemento interessante para apontar cliques potencialmente fraudulentos que conseguiram passar na coligação de fraudadores que utilizava usuários humanos.

6.4 Considerações do Capítulo

Nesta seção realizamos a validação da proposta apresentada na seção 5. Ela consistiu de uma carga de testes compreensivo e da elaboração de um protótipo de uma abordagem para rede de anúncios. Tal protótipo atualmente encontra-se mantido no site http://www.clicksafe-project.com e segue implementação de especificação apresentada segundo diagrama de componentes apresentado no Capítulo 5 e o mesmo tem o propósito de ser uma ferramenta de exibição de anúncios simples, mas com foco em segurança, cuja arquitetura se baseia na validação de cupons em cache cookies após autenticação pelo Google NoCAPTCHA reCAPTCHA. O modelo simplificado de segurança é extensível a outras abordagens, bem como a simplicidade de exibição de anúncios, que hoje seguem um modelo de leilão às escuras baseada no modelo PPC, com anunciantes competindo entre si para exibição de determinadas palavras-chave.

Como expomos na introdução desse Capítulo, esse protótipo consiste em uma interface para validação da solução deste trabalho que não aprofundou aspectos de interface gráfica. Carece, portanto, de um estudo e um diagnóstico para a escolha de uma interface adequada para o seu uso. O diferencial desta solução está, portanto, no fato de introduzir um novo passo de dificuldade significativo no sentido de combater a fraude de clique sob um foco de segurança computacional para a internet.

7 Considerações Finais

Prevenir a fraude de clique manualmente preventivamente nunca foi o objetivo precípuo deste projeto de pesquisa, que consistia em combater a fraude de clique automatizada com CAPTCHA clicáveis. Adotamos a abordagem de detecção complementar para tornar essa solução a mais completa possível em termos de segurança. Nesse sentido, muito embora possam ser vistas como plataformas de funcionamento quase independentes, fica claro que os resultados obtidos entre a parte de CAPTCHA, de prevenção e a parte menos proativa, de detecção, se complementam de maneira fundamental.

No que diz respeito a carga de testes realizada no Capítulo anterior, os resultados apresentados pela resolução dos CAPTCHA clicáveis manuais se assemelharam muito com os resultados obtidos por Zhu et al (2014) e Bindu (2015) e, nos testes realizados, embora tenhamos percebido alguns erros pontuais, a taxa de acerto de resposta aos CAPTCHA pelos usuários humanos se aproximou de 98%. O mais interessante é que, como estávamos rodando uma carga de testes semi automatizados, na verdade dos 5.583.301 cliques registrados, apenas 6.404 foram originados da parte humana manual dos testes, ou seja, 0,104%. Talvez a diferença no número de cliques tenha sido exacerbada devido ao fato de os mesmos terem sido rodados em uma rede local, o que possibilitou uma rápida ativação dos scripts automatizados após a autenticação – no entanto, ainda assim, há que se considerar a enorme diferença entre a quantidade de cliques automatizados que uma quantidade muito pequena de autenticações manuais foi capaz de gerar.

No nosso entendimento, esses números mostram que a fraude de clique neste cenário de fato não pode ser ignorada, sendo sobremaneira importante considerar o caso da detecção após a autenticação manual. Os fraudadores naturalmente seguirão esse caminho

enquanto houver um modelo de exploração do usuário humano e é importantíssimo ter algoritmos eficientes na detecção da fraude, como o *Similarity-Seeker* e outros.

No que diz respeito ao *Similarity-Seeker* em si, percebemos que, na medida em que grupos maiores de coligação foram sendo detectadas, as mesmas na verdade deixaram de ser coligações e passaram a ser ruídos de grupos de IP extremamente populares. Só relembrando, utilizamos tráfego de clique simulado de uma rede real e foi necessário verificar tais IPs no *whois* da arin.net (http://www.arin.net/whois). Uma conclusão dessa atividade é que é importante, para a detecção, iniciar a busca por coligação com poucos sites e uma similaridade baixa (5 sites com 0,1 de similaridade, por exemplo) e ir aumentando esse número aos poucos. O ruído será manifestado normalmente em pares isolados, coligações pequenas e fraudes menores com cliques sobrepostos. Segundo essa abordagem, 93% dos ataques realizados após a autenticação manual foram corretamente detectados.

Este trabalho conjunto de prevenção e detecção é uma solução generalizada para inicialmente prevenir ataques de fraude de clique e, posteriormente, detectar se a prevenção não foi capaz de filtrar atacantes que porventura burlaram os diversos mecanismos de prevenção que o sistema já incorpora em si, tais como CAPTCHA clicáveis, autenticação por cupons e tempo de validade dos mesmos. A abordagem de detecção vem aumentar o grau de segurança com base na premissa de que o tráfego dos sites na internet é tipicamente não similar – assim, qualquer grau de similaridade será usualmente suspeita.

Uma outra vantagem dessa validação foi ter aberto portas para outros estudos futuros: por exemplo, identificamos a possibilidade de verificar o problema das chamadas redes de *Trojan*, nas quais muitos publicadores podem se associar para atacar várias redes de anúncios. Cada rede de anúncio tipicamente sabe apenas do seu próprio tráfego e nenhuma rede vai realmente detectar a similaridade do tráfego dos atacantes – uma possível coligação de redes

de anúncios contra a fraude de clique para contra-atacar coligações de fraudadores poderia ser uma solução para esse problema e é possível um tópico para estudos futuros.

7.1 Trabalhos Relacionados

Conforme descrito anteriormente, existe um conflito de interesses entre os anunciantes, publicadores e as redes de anúncio com respeito à sistemática da fraude de clique, pois os três participantes podem ter interesse direto ou indireto com a mesma. Discutiremos mais profundamente acerca desse conflito de interesses no Capítulo 3 mas, de acordo com o relatório de Tuzhilin (2006), elaborado pelo professor Alexander Tuzhilin, da Universidade de Nova Iorque, ele causa uma dificuldade na definição exata do que venha a ser um conceito de "cliques fraudulentos". Uma discussão técnica sobre isso é realizada no relatório (TUZHILIN, 2006), o qual conclui ser impossível descrever de maneira detalhada as definições operacionais por trás da fraude de clique e, assim, definir um conceito universal de cliques inválidos.

Em contrapartida, o relatório torna possível obter uma abstração em um sistema de detecção de fraudes. Tuzhilin (2006) também classifica como "razoável" que a ferramenta de busca seja colocada sob investigação em casos de fraude de clique. Um dos objetivos do relatório é a preservação da privacidade do sistema de detecção de fraudes, de maneira a manter a sua efetividade. Isso acabou por motivar alguns pesquisadores a analisarem como as ferramentas de busca podem combater a fraude de clique. Obviamente, é compreensível que algumas organizações possuam um certo receio acerca da necessidade de se expor sistemas internos de detecção de fraude das redes de anúncios e assim facilitar aos atacantes como burlá-los. Entretanto, se as pesquisas identificarem métodos de controle de fraude para cada técnica de fraude publicada, essa exposição se tornará menos crítica (TUZHILIN, 2006).

Um exemplo de uma dessas pesquisas pode ser encontrada no trabalho de Metwally, Agrawal e Abbadi (2005), que sugere a utilização de um histórico de ações e posterior busca em profundidade para identificação de comportamento suspeito, tornando necessário, assim, o controle de navegação dos usuários de uma rede de anúncios em diversos níveis, desde o acesso ao site do publicador, as ações que foram realizadas nesse site que causaram a exibição, a seleção e o clique no anúncio, e ainda ações posteriores ao redirecionamento. Trabalhos realizados por Majumdar, Kulkarni e Ravishankar (2007) propõem o uso de protocolos de comunicação para a identificação de comportamento fraudulento por parte de intermediários em redes de disseminação de conteúdo.

Juels, Stamm e Jakobsson (2006) apresentam técnicas interessantes para lidar com o problema da fraude de clique em sistemas PPC. A ideia deles é habilitar um compartilhamento de evidências da validade de usuários entre as diversas organizações participantes de uma rede de anúncios, na verdade, formando uma espécie de federação de redes de anúncios online. A validação dos usuários por si só serviria a dois propósitos: distinguir os usuários de *bots* , *scripts* e fraudadores, e identificar quais usuários possuem cliques mais valiosos, o que permitiria também a elaboração de uma espécie de ranking dos mesmos. Embora interessante, essa ideia parece um pouco utópica por contar com o compartilhamento de informações de clientes entre organizações tipicamente concorrentes e pelas óbvias implicações de privacidade que os próprios autores se adiantam a discorrer em seu trabalho (JUELS; STAMM; JAKOBSSON, 2006).

No trabalho DETEcting Coalition hiT Inflation attacks in adVertising nEtworks Streams (DETECTIVES), Metwally, Agrawal e Abaddi (2007) apresentaram o algoritmo *Similarity-Seeker*, que intenciona ser uma ferramenta poderosa na busca por similaridades em históricos de clique na abordagem de detecção. Também inspirados no combate à forma automatizada da fraude de clique em larga escala, os autores buscaram maneiras de detectar a

forma mais sofisticada da mesma, que envolve coligações entre fraudadores. O algoritmo desenvolvido, que também adotamos como parte da nossa arquitetura, para cobrir casos de detecção, conforme descrito no Capítulo 5 deste trabalho, se propõe a descobrir tais coligações e pode ser generalizada para diferentes tamanhos de coligações. No artigo os autores demonstram um conjunto de testes para a identificação de coligações e neste trabalho nós também realizamos uma bateria de validações em escala menor, em um histórico de tráfego fornecido por uma organização real. Tal validação pode ser encontrada no Capítulo 6.

Por sua vez, Haddadi (2010) apresentou o conceito de anúncios falsos (*bluff ads*). Os mesmos seriam uma espécie de "blefe" em meio a anúncios legítimos, cujos objetivos seriam funcionar como espécies de iscas para fraudadores. Os mesmos deveriam funcionar como uma abordagem complementar a outras formas de detecção e deveriam conter ou texto extremamente irrelevante sem informação destino relevante, ou o contrário, contendo informação alvo relevante, mas sem texto relevante. Assim eles agiriam como um teste de legitimidade do indivíduo clicando nos anúncios, e consequentemente ajudando na identificação de similaridades para ferramentas de detecção.

Conforme exposto acima, as abordagens referenciadas e encontrados na academia buscam lidar com a fraude de clique através de análise de históricos. Todas as abordagens descritas nesta seção 7.1 lidam com a fraude numa abordagem de detecção, após a mesma já ter acontecido. Acreditamos ser necessário pelo menos complementar essa abordagem com um método mais proativo, que seja capaz de impedir que a fraude aconteça, pelo menos na sua forma automatizada. Uma das formas de realizar isso é distinguindo programas automatizados de humanos através de testes de Turing automatizados, os chamados CAPTCHA, que estudamos ao longo desta Tese.

Na verdade, CAPTCHA clicáveis também têm sido gradativamente utilizados em outras aplicações de segurança, como se pode verificar no trabalho de Bindu (2015), que propõe a utilização deste mecanismo para combater ataques de *spyware*. O autor verifica, em termos de usabilidade, segurança e performance, que a alternativa aos mecanismos inexistentes de segurança ou aos CAPTCHA tradicionais, que podem ser facilmente quebrados por scripts como keyloggers, são uma forma única de proteção, já que usuários podem facilmente clicar em uma sequência de imagens em uma ordem predefinida para formar um CAPTCHA.

7.2 Trabalhos Futuros

Como trabalho futuro, um dos principais aspectos que a nossa abordagem precisa focar é na usabilidade de um sistema como este em um ambiente de produção real. Mesmo com as preocupações inerentes à autenticação para possibilitar um uso mais razoável do sistema, é inegável que, atualmente, o principal foco é realmente o aspecto de segurança, prevenção e identificação da fraude de clique, o que sacrifica o uso da rede de anúncio online para o bom usuário. Por exemplo, um usuário legitimo que não esteja navegando pela rede de anúncios, mas que clique eventualmente em um dos anúncios exibidos por questões de atratividade de marketing será forçado a responder um CAPTCHA. Uma solução temporária para essa questão pode ser reduzir o número de requisições por CAPTCHA para apenas uma amostra dos usuários (digamos que apenas 50% dos mesmos precisem responder ao desafio), o que pode minimizar o problema.

Além disso, propomos a adoção de um mecanismo que, através da autenticação dos usuários prévia, realizada pela resposta aos CAPTCHA, torne possível identificar e discernir cliques diferenciados, mais valiosos, daqueles usuários que possuem uma probabilidade maior de adquirir os produtos mostrados nos anúncios. Essa abordagem é semelhante àquela

apresentada por Jakobsson et al (2007), com a diferença que a validação dos cliques em tal abordagem é realizada através da troca de evidências entre as entidades participantes da rede de anúncios, o que já serviria também como classificador da qualidade dos cliques. Na nossa proposta, essa validação prévia é realizada pela resposta de um CAPTCHA, e a classificação será realizada com base na atividade registrada de cada usuário autenticado. É importante destacar que essa implementação adicional implica em mudanças de diversos aspectos, incluindo a forma como as redes de anúncios funcionam atualmente em termos de modelo de mercado. Como já destacamos na seção 3.4, esse compartilhamento de evidências requer uma colaboração entre os diversos participantes da rede – incluindo organizações que são concorrentes entre si. É, portanto, nesse ponto, uma proposta que necessita ser estudada de maneira mais profunda, no entendimento de uma oferta que signifique de fato em motivação de tais participantes para a utilização do sistema.

Conforme já exposto anteriormente, o protótipo adota inicialmente a abordagem *Similarity-Seeker* a qual, após testes de funcionalidade, pode ser aprimorada para a necessidade do protótipo, que é complementar a uma abordagem de filtragem de cliques. Hoje, a abordagem de detecção foca na fraude do publicador que visa aumentar suas receitas por meio do clique em anúncios exibidos no seu site, inclusive no caso da formação de coligações de publicadores. Conforme já mencionamos na seção 5.7, há diversas outras formas de fraude de clique manual que podem ser exploradas como abordagem de detecção, sendo essa apenas uma das mais comuns.

Além disso, o software aqui disponibilizado encontra-se obviamente em estágio de protótipo de desenvolvimento, e o mesmo necessita ser aprimorado, de maneira geral, em todas as suas frentes. Por exemplo, conforme exposto no Capítulo 5, os anúncios hoje são exibidos com o mero objetivo de visualização, já que a interface utilizada não seria facilmente exposta em um site de publicador. Tipicamente, para possibilitar tal uso, os mesmos deveriam habilitar

a exibição em um formato de *banner* para a sua disposição em sites de publicadores, para que toda a sistemática já implementada de identificação, seleção de anúncios e posterior cobrança faça sentido.

Finalmente, não foram realizados testes com CAPTCHA de classe III, já que a pesquisa é focada em CAPTCHA clicáveis. Acreditamos que essa categoria precisa ser estudada mais a fundo em pesquisas futuras, especialmente as novas abordagens para o Google NoCAPTCHA reCAPTCHA, cuja adoção é gratuita. Assim, podemos apontar como um trabalho futuro o estudo de CAPTCHA adicionais nesta metodologia, possibilitando a flexibilização da mesma e a sua adoção em larga escala, sem limitações a questões de propriedade, por exemplo.

7.3 Importância e Contribuições desta Pesquisa

Antes de mais nada, sob a percepção dos atores envolvidos na fraude de clique, esta é uma pesquisa extremamente relevante, por causa do alto impacto e do grau de interesse onde quer que tenha sido levada e publicada. Desde as primeiras propostas e descrições, sempre tivemos grande aceitação da comunidade acadêmica e certa facilidade em transitar no meio acadêmico ao falar do tema. Além disso, a competição dos negócios globais têm sido alvos de constantes mudanças nos últimos anos.

Conforme já exposto, o desenvolvimento de *sites* de Internet comerciais acompanhou a popularização da própria Internet e anunciar online é uma das formas mais lucrativas. As empresas de publicidade recomendam a realização de campanhas de marketing na *web*, tanto para pequenas e para grandes empresas, desde que possuam o objetivo de atingir diversos tipos de clientes, em diversas localidades globalizadas. Inclusive, podemos destacar

que um dos grandes benefícios de anunciar desta forma é a possibilidade de publicar informação e conteúdo sem fronteiras geográficas ou de fuso horário.

De acordo com Hargreaves (2014), 90% do lucro do Facebook vem da venda de anúncios, e, enquanto 2014 foi ano da venda dos mesmos associado ao negócio de anúncios em dispositivos móveis, a tendência é que o uso da rede social como uma rede de anúncios online, já que o lucro com o PPC possui uma tendência de 72% de crescimento em 2015. As projeções da Research and Markets (2013) não são diferentes, pois dão conta que a indústria global deste mercado deve valer nada menos que 139,8 bilhões de dólares em 2018. Isso significa que as cinco maiores empresas da indústria acumulam 69% do mercado total de anúncios em 2012, cuja previsão de crescimento beirava 18% em 2010.

Tal crescimento levou a investimentos significativos em *marketing online* e digital por parte de diversas empresas. Em 2007, por exemplo, o Google comprou a DoubleClick por 3,1 bilhões de dólares. Os autores da projeção analisaram os maiores desafios da indústria, e a engenharia social mudou a forma como as redes de anúncio se comportam. Um grande desafio existente é, por exemplo, a falta de habilidade em se comunicar adequadamente, o que introduz necessidades de mudanças na forma como os anúncios precisam ser apresentados em seu conteúdo. Por sua vez, o movimento de neutralidade de rede, e sua consequente padronização e regulação da Internet (WEISER, 2009), cada vez mais crescente, também afetou a lucratividade da mídia como um todo, o que deveria afetar diretamente o mercado, que por sua vez parece imune a essas flutuações naturais de mercado.

Ao se expor a este estudo, as partes interessadas no negócio de redes de anúncio online podem entender os riscos aos quais estão expostos e garantir a confiabilidade do conteúdo que está sendo exibido em suas páginas, incluindo o entendimento dos mesmos ao longo das diferentes redes de anúncios. Assim, nesse sentido, este trabalho apresenta contribuições para os portais que publicam anúncios na medida em que os mesmos passarão a

contar com uma ferramenta que pode auxiliá-los no processo de tomada de decisão acerca dos serviços de anúncio atualmente disponíveis. Também é objetivo deste trabalho que os participantes fundamentais deste negócio de anúncios na Web entendam a necessidade de realizar negócios de maneira ética e socialmente responsável e como atingir altos níveis de confiabilidade influencia positivamente a sustentabilidade desta forma de *marketing*.

Quanto às redes de anúncios atualmente existentes, aos possíveis entrantes e a outras formas de entidades que funcionam como repositório de anúncios para a Internet, acreditamos que este projeto é importante no sentido de ser um diferencial na busca por serviços seguros e de qualidade. Os mesmos são os grandes interessados no assunto em termos organizacionais, e devem buscar sempre estimular os fatores acima entre seus clientes e usuários, para quem o trabalho também será recomendado, já que os mesmos são consumidores em potencial dos serviços disponibilizados pelas redes de anúncios e publicadores. Entendemos que esta pesquisa os auxilia a entender a disposição atual do negócio de pagamento por clique na Internet e os habilita a buscar, de maneira proativa, dentre as diversas opções disponíveis, a utilização de serviços que se apresentam confiáveis e robustos. Esses devem contribuir, de maneira ética, para a evolução contínua da utilização de negócios online, enquanto realizam suas atividades fim e garantem a manutenção de clientes.

Em termos técnicos, finalmente, esta pesquisa produziu diversas contribuições para a área de anúncios *online* e, especificamente para as tecnologias envolvidas no negócio de pagamento por clique. As contribuições se relacionam ao estudo do estado atual da arte de diferentes abordagens da área de *marketing* na Internet e à proposição de uma arquitetura que explora os principais aspectos positivos de cada uma destas abordagens, buscando auxiliar a percepção e a comunicação entre os diversos participantes de uma estrutura de PPC. No final das contas, objetivamos melhorar a aquisição de conhecimento do domínio do negócio de

anúncios na web para apoiar na identificação de requisitos e necessidades de sistemas deste domínio.

Mais especificamente, podemos entender que este trabalho realizou um estudo da sistemática dos anúncios *online* e de sua atual disposição de negócios, garantindo que a arquitetura a proposta se encaixa neste modelo de negócios, aliado a um estudo de CAPTCHA, incluindo seu histórico, classificação, análise de *benchmarks*, OCR, e a introdução de CAPTCHA clicáveis como uma forma de prevenir a automatização da fraude de clique. No entanto, nunca tivemos a pretensão dessa abordagem ser uma solução final para o problema da fraude, e apresentamos como necessária e fundamental a introdução de abordagens já consagradas para a detecção de cliques em históricos de uso com o objetivo garantir a confiabilidade do sistema como um todo.

Com efeito, realizamos a proposta de uma abordagem, que se baseou na utilização de CAPTCHA clicáveis, o que significa, para o combate à fraude de clique, uma evolução de paradigma, já que essa arquitetura se propõe a ser precipuamente de prevenção, algo sem precedentes na academia e para o mercado de anúncios online. No entanto, a abordagem não ignora aspectos valiosos já existentes de detecção, propondo o uso casado de ambos. De acordo com a abordagem proposta, realizamos a implementação de um protótipo que pode servir de referência para desenvolvimentos adicionais futuros.

Tais objetivos da pesquisa só foram alcançados após o estudo realizado quanto ao conjunto de métodos de fraude nesse nicho de negócios, dando especial destaque à fraude de clique. Houve para isso o estudo de casos legais da fraude do clique, em especial do relatório de Tuzhilin (2006), que nos tornou possível entender as motivações envolvidas para esse tipo de fraude, ponto fundamental para elaborar uma estratégia para prevenção e para a proposta de uma rede de anúncios que implementa a abordagem proposta. Nesse sentido, podemos confirmar que já realizamos o estudo extensivo de heurísticas para construção de anúncios a

serem exibidos nos sites de Internet, e possuímos assim, hoje, uma estratégia de exibição de anúncios para a rede. A estratégia final ainda se encontra em desenvolvimento, embora o protótipo apresente a possibilidade de visualizar os anúncios por *slots* de preço e o ajuste de exibição seria uma questão mais cosmética do que de qualquer outra natureza.

Ainda no sentido da exploração do protótipo, realizamos a proposição do tipo de CAPTCHA a ser utilizado por esta abordagem, incluindo a proposta de um esquema de desenvolvimento do fluxo de comunicações associado à resolução dos desafios, bem como o estudo e a escolha de algoritmos para a criação de desafios CAPTCHA. Exploramos opções adicionais de desafios, incluindo os de Classe III e o Google NoCAPTCHA reCAPTCHA, que acabou sendo incorporado na solução final devido a aspectos de confiabilidade, usabilidade, segurança e desenvolvimento constante.

Finalmente, realizamos a proposição da sistemática de comunicação entre os usuários e a rede de anúncios por meio de um fluxograma, que inclui a troca de cookies, respostas aos desafios e fluxo de comunicações em rede. Estudamos ainda um número de diversas outras entidades e serviços envolvidos com o esquema da rede de anúncios, os quais ou já se encontram implementados ou cujo desenvolvimento está considerado para trabalhos futuros. Como efeito desta proposta de tese, podemos subdividir mais especificamente os resultados apresentados deste trabalho em dois grandes pilares mais palpáveis, a produção cientifica em si, que está solidamente apresentada ao longo deste documento de tese e nas publicações realizadas ao longo desta pesquisa de doutoramento, a saber “A Proposal to Prevent Click-Fraud Using Clickable CAPTCHAs” (COSTA et al, 2013), “Uma abordagem para Combater a Click Fraud Usando CAPTCHAs Clicáveis” (COSTA, 2010) e “Um Estudo Sobre a Fraude de Clique: Casos Conhecidos e Impacto no Direito Brasileiro” (DE QUEIROZ; COSTA, 2014), e o desenvolvimento de um protótipo do software, oferecido como uma rede de anúncio online.

Quanto aos resultados apresentados, temos ainda um conjunto de atividades que foram realizadas até a conclusão desta tese de doutorado. Foi necessário realizar uma análise das formas de monetarização das empresas envolvidas nesta sistemática, dando foco especial ao pagamento por clique, e estudar de como esta forma de obtenção de receitas se relaciona com a fraude de clique, de maneira a avaliar se algumas mudanças na disposição do ambiente de negócios de PPC serão necessárias.

Além disso, quanto à proposta de desenvolvimento de uma rede de anúncios que implementa a arquitetura proposta, ainda nos foi necessário introduzir alguns componentes, como o sistema de gerenciamento de banco de dados como um componente da arquitetura através de um alinhamento de interesses externo ao ambiente de produção. Atualmente utilizamos o Google NoCAPTCHA reCAPTCHA para este fim e confiamos na base de dados externa do aplicativo.

Levando em consideração o desenvolvimento do protótipo, embora o mesmo tenha sofrido alguns atrasos devido aos problemas citados anteriormente, o mesmo encontra-se online, embora em desenvolvimento, mas disponível na URL: http://www.clicksafe-project.com/. Precisamos atualmente incrementar os módulos “gerador de anúncios” e “gerador de CAPTCHA”, conforme detalhados na Figura 5.2 de arquitetura, pois os mesmos atualmente estão representados na mesma entidade PHP. Todas as atividades de implementação foram consideradas ao longo do cronograma de desenvolvimento desta tese, e a descrição da proposta de desenvolvimento dos mesmos deve entrou neste documento de tese finalizado.

REFERÊNCIAS

ALSUHIBANY, S. A. Optimising CAPTCHA Generation. In: INTERNATIONAL CONFERENCE ON AVAILABILITY, RELIABILITY AND SECURITY, 6., 2011, Vienna. **Anais**... Vienna: IEEE. 26 ago 2011. p. 740-745.

AHN, L. v. et al. CAPTCHA: Using hard AI problems for security. In: INTERNATIONAL CONFERENCE ON THE THEORY AND APPLICATIONS OF CRYPTOGRAPHIC TECHNIQUES, 4., 2003, Varsóvia. **Anais**... Londres: Springer, 2004. p. 100-143.

_____. reCAPTCHA: Human-Based Character Recognition via Web Security Measures. **Science Express**, Berkeley, v. 321, n. 5895, p. 1456-1468, 28 mar. 2008.

ANA; WHITEOPS. The Bot Baseline: Fraud in Digital Advertising. **Association of National Advertisers, White Ops (Online)**. Nova Iorque, NY, 15 dez. 2015. Disponível em: <http://www.ana.net/getfile/23332>. Acesso em 08 fev. 2016.

ANANTATHANAVIT, M.; MUNLIN, M. Radius Particle Swarm Optimization. In: INTERNATIONAL COMPUTER SCIENCE AND ENGINEERING CONFERENCE, 13., 2013, Nakorn Pathom. **Anais**... Nakorn Pathom: IEEE. 6 Set 2013. p. 126-130.

ANUPAM, V. et al. On the Security of Pay-Per-Click and Other Web Advertising Schemes. In: WWW INTERNATIONAL CONFERENCE ON THE WORLD WIDE WEB, 8., 1999, Toronto. **Anais**... Amsterdã: Elsevier Science, 1999. p. 10-23.

ARAS, V. O projeto de lei dos cibercrimes (PLS 76/2000): crítica ao substitutivo aprovado no Senado. **Revista ANPR Online**. Brasília, v.1, n.8, jan./jun. 2009. Disponível em: <http://www.anpr.org.br/portal/components/com_anpronline/ media/Artigo_Projeto de lei dos cibercrimes - PLS 76 de 2000.doc>. Acesso em: 17 out. 2014.

ARICA, N.; VURAL-YARMAN, F. T. Optical Character Recognition for Cursive Handwriting. **IEEE Transactions on Pattern Analysis and Machine Intelligence,** Washington, DC, v. 24, n. 6, p. 801-813, jun. 2002.

ARRINGTON, M. Alexa says now Youtube is bigger than Google. Alexa is useless. **TechCrunch**. São Francisco, CA, v.5, n.1, 13 ago. 2007. Disponível em <http://techcrunch.com/2007/ 08/13/alexa-says-youtube-is-now-bigger-than-google-theyre-wrong/>. Acesso: 22 mai. 2014.

ARUNA, P.; KANCHANA, R. Face image captcha generation using particle swarm optimization approach. In: IEEE INTERNATIONAL CONFERENCE ON ENGINEERING AND TECHNOLOGY, 7., 2015, Coimbatore. **Anais**... Coimbatore: IEEE. 23 mar. 2015. p. 1-5. 978-1-4799-1853-9

BAECHER, P. et al. CAPTCHA: The Good, the Bad and the Ugly. **Lecture Notes in Informatics**, Berlim, v. 5, n. 2, p 353-365, 2010.

BELLARE, M.; CANETTI, R.; KRAWCZYK, H. Keying hash functions for message authentication. In: ANNUAL INTERNATIONAL CRYPTOLOGY CONFERENCE ON

ADVANCES IN CRYPTOLOGY, 16., 1996, Londres. **Anais**... Londres: Springer, 1996. p. 216-321.

BINDU, C. S. Click based Graphical CAPTCHA to thwart spyware attack. In: IEEE INTERNATIONAL ADVANCE COMPUTING CONFERENCE, 23., 2015, Bangalore, **Anais**... Bangalore: IEEE. 13 jun. 2015. p. 324-328.

BRADY, R. Ecommerce Pay-per-click Ad Trends for 2016. **Pratical E-commerce**. Grand Junction, CO, v.5, n.1, 12 jan. 2016. Disponível em: <http://www.practicalecommerce.com/articles/95845-Ecommerce-Pay-per-click-Ad-Trends-for-2016>. Acesso em 14 fev. 2016.

BRIAN, M. How Web Advertising Works. **How Stuff Works Tech**. Atlanta, v. 31, n. 12, 08 abr. 2002. Disponível em: <http://computer.howstuffworks.com/web-advertising6.htm>. Acesso em 06 fev. 2016.

BROWSERMEDIA. DoubleClick deal means Google controls 69% of the online ad market. **Brower Media,** Essex, v. 6, n. 4, 01 abr. 2008. Disponível em: <http://www.browsermedia.co.uk/2008/04/01/doubleclick-deal-means-google-controls-69-of-the-online-ad-market/>. Acesso em: 23 mai. 2013.

BUCHDAHL, M. Lawyers Are Big Players in SEO, Pay-per-click Game. **Web Marketing Today.** Traverse City, MI, v. 4, n. 1, 23 aug. 2012. Disponível em: <http://webmarketingtoday.com/articles/lawyers-are-big-players-in-seo-pay-per-click-game/>. Acesso em: 08 fev. 2016.

BURSZTEIN, E.; MARTIN, M.; MITCHELL, J. Text-based CAPTCHA strengths and weaknesses. In: ACM CONFERENCE ON COMPUTER AND COMMUNICATIONS SECURITY, 18., 2011, Nova Iorque. **Anais**... Nova Iorque: ACM. 2011. p 125-138. ISBN: 978-1-4503-0948-6

CHAFFEY, D. et al. **Internet Marketing**: Strategy, Implementation and Practice. 3ed. Harlow, England: Prentice Hall. 2006. ISBN 0-273-69405-7.

CHANDAVALE, A.A.; SAPKAL, A.M.; JALNEKAR, R.M. Algorithm to Break Visual CAPTCHA. In: IEEE INTERNATIONAL CONFERENCE ON EMERGING TRENDS IN ENGINEERING AND TECHNOLOGY, 2., 2009, Naqpur. **Anais**… Naqpur, India: IEEE. 18 dec. 2009. p 258 - 262. ISBN: 978-1-4244-5250-7

CHEN, C. J.; WANG, Y. W.; FANG, W. P. A Study on Captcha Recognition. In: INTERNATIONAL CONFERENCE ON INTELLIGENT INFORMATION HIDING AND MULTIMEDIA SIGNAL PROCESSING, 10., 2014, Kitakyushu. **Anais**... Kitakyushu: IEEE. 29 aug 2014. p. 395-398. 978-1-4799-5389-9

CHO, G. et al. An Empirical Study of Click Fraud in Mobile Advertising Networks. In: INTERNATIONAL CONFERENCE ON AVAILABILITY, RELIABILITY AND SECURITY, 10., 2015, Toulouse. **Anais**... Toulouse: IEEE. 27 ago. 2015. p. 382-388.
CHOW et al. Making CAPTCHA Clickable. In: WORKSHOP ON MOBILE COMPUTING SYSTEMS AND APPLICATIONS, 9, 2008. Napa Valley, CA. **Anais**… Nova Iorque: ACM Digital Library. 23 fev. 2008. p. 102-114.

CICILIANO, L. Attorneys Should Avoid The Pay-Per-Click Trap. **SEO For Lawyers (Online)**. Framingham, MA, v. 2, n. 2, 04 jun. 2015. Disponível em: <https://www.seo-for-lawyers.com/attorneys-avoid-the-pay-per-click/>. Acesso em: 05 out. 2015.

CONSTANTIN, L. Snapchat's new image-based human verification system already defeated. **Computerworld (Online)**. Framingham, MA, v. 5, n. 12, 23 jan. 2014. Disponível em: <http://www.computerworld.com/article/2486862/social-media/snapchat-s-new-image-based-human-verification-system-already-defeated.html>. Acesso em: 09 fev. 2016.

COSTA, R. A. **Uma Abordagem para Utilização de CAPTCHA Clicáveis para Combater a Click Fraud**. 27 ago. 2010. 134f. Dissertação (Mestrado em Ciências da Computação) – Centro de Informática, Universidade Federal de Pernambuco, Recife. 2010.

COSTA, R. A.; DE QUEIROZ, R. J. G. B.; CAVALCANTI, E. R. A Proposal to Prevent Click-Fraud Using Clickable CAPTCHA. In: INTERNATIONAL CONFERENCE ON SOFTWARE SECURITY AND RELIABILITY COMPANION*,* 8., 2012, Gaithesburg. **Anais**… Gaithesburg: IEEE, 30 jun 2012. p. 20-22.

DANCHEV, D. Google's reCAPTCHA under automatic fire from a newly launched reCAPTCHA-solving/breaking service. **WebRoot Threat Blog (Online)**. Broomfield, CO, v1, n. 1, 21 jan 2014. Disponível em: <http://www.webroot.com/blog/2014/01/21/googles-reCAPTCHA-automatic-fire-newly-launched-reCAPTCHA-solving-breaking-service/>. Acesso em: 04 de Setembro de 2015.

DAVIS, W. Google Wins $75,000 in Click Fraud Case. **Media Post (Online)**. Nova Iorque, NY, v. 15, n. 10, 05 jul. 2005. Disponível em: <http://www.mediapost.com/publications/article/31772/google-wins-75000-in-click-fraud-case.html?edition= >. Acesso em: 06 fev. 2016.

DAVAR, P. A. Online advertising and its security and privacy concerns. In: INTERNATIONAL CONFERENCE ON ADVANCED COMMUNICATION TECHNOLOGY, 15., 2013, PyeongChang. **Anais**… PyeongChang: IEEE. 30 jan 2013. p. 372-377.

DE JONG, A.; ROSENTHAL, L.; VAN DIJK, M. The Risk and Return of Arbitrage in Dual-Listed Companies. **Review of Finance**, [S.I.]: SSRN, 2009. Disponível em: <http://ssrn.com/abstract=525282>. Acesso em: 17 mai. 2015.

DE QUEIROZ, R. J. G. B; COSTA, R. A. Um Estudo Sobre a Fraude de Clique: Casos Conhecidos e Impacto no Direito Brasileiro. **Novos Estudos Jurídicos**, Itajai, SC: Univali, 2013. v. 18, p. 551-564. Disponível em: <http://siaiap32.univali.br/seer/index.php/nej/article/view/513>. Acesso em: 15 jul. 2013.

DIAS, Jean Carlos. **O direito contratual no ambiente virtual**. Curitiba: Juruá, 2014.

DINIZ, Maria Helena. **Curso de direito civil brasileiro**. São Paulo: Saraiva, 2015.

DINU, D. D.; TOGAN, M. DHCP server authentication using digital certificates. In: INTERNATIONAL CONFERENCE ON COMMUNICATIONS, 10., 2014, Bucharest. **Anais**... Bucharest: IEEE. 31 mai 2014. pp. 1-6.

DIRECT MARKETING ASSOCIATION. **The Power of Direct Marketing**: ROI, Sales, Expenditures and Employment in the U.S. Nova Iorque: DMA, 2016.

DIRECT MARKETING ASSOCIATION. **DMA National Client Email Report**. Nova Iorque: DMA, 2015.

DOYLE, E. Not All Spyware is as Harmless as *Cookie*s: Block it or Your Business Could Pay Dearly. **Computer Weekly (Online)**. 27 nov. 2003. Disponível em: <http://www.computer weekly.com/Articles/2003/11/27/198884/Not-all-spyware-is-as-harmless-as-*cookie*s.htm>. Acesso em: 23 mai. 2010.

ELGIN, B. The Vanishing Click Fraud Case. **Businessweek Technology**. Nova Iorque, Estados Unidos: Businessweek, 2006. Disponível em: <http://www.businessweek .com/technology/content/dec2006/tc20061204_923336.htm?campaign_id=bier_tcc.g3a.rssd1 204f>. Acesso em: 08 jan. 2015.

ELSON, J. et al. ASIRRA: a CAPTCHA that exploits interest-aligned manual image categorization. In: ACM CONFERENCE ON COMPUTER AND COMMUNICATIONS SECURITY, 14., 2007, Alexandria. **Anais**… Nova Iorque: ACM Digital Library, 2007. p. 10-26.

FAIN, D.C.; PEDERSEN, J. O. Sponsored search: a brief history. **Bulletin of the American Society for Information Science and Technology**, Silver Spring, Maryland: ASIS. 2006. vol. 32, pp. 12,13.

FARRIS, P. et al. **Marketing Metrics**: The Definitive Guide to Measuring Marketing Performance. New Jersey: Pearson Education. 2010. ISBN 0-13-705829-2.

FENG, J.; LI, J. A New Certificate-Based Digital Signature Scheme. In: INTERNATIONAL CONFERENCE ON EMERGING INTELLIGENT DATA AND WEB TECHNOLOGIES, 4., 2013, Xi'an. **Anais**... Xi'an: IEEE. 11 set 2013. p 547-549.

GOODMAN, A. **Winning Results with Google AdWords**. Nova Iorque: McGraw-Hill, 2008. 376p.

HADADDI, H. Fighting Online Click-Fraud Using Bluff Ads. **ACM SIGCOMM Computer Communication Review**. London: University of London, 2010. Volume 40, Number 2.

HARGREAVES, Steve. Facebook profit soars. **CNN Money**. Nova Iorque, a. 15, v. 1, n. 2. 30 jan. 2014. Disponível em: <http://money.cnn.com/2014/01/29/technology/facebook-earnings/ >. Acesso em: 11 mar. 2014.

HENDRICKS, Drew. Will 2014 Be An SEO Or PPC Year For Marketers? **Forbes.com.** Nova Iorque, v. 2, n. 1, 2013. Disponível em: <http://www.forbes.com/sites/drewhendricks/2013/11 /22/will-2014-be-an-seo-or-ppc-year-for-marketers/ >. Acesso em: 22 fev. 2014.

HOFFMAN, D; NOVAK, T. P. **How to Acquire Customers on the Web**. [S.I.] Harvard Business Review (online), 2000. Disponível em: <https://hbr.org/2000/05/how-to-acquire-customers-on-the-web>. Acesso em 23 mai. 2014.

HOMAKOV, E. **The No CAPTCHA Problem**. [S.I.] Online Business Security Annual Manifest, 2014. Disponível em <http://homakov.blogspot.in/2014/12/the-no-CAPTCHA-problem.html>. Acesso em 15 jan. 2015.

IAB. IAB internet advertising revenue report: 2015 first six months results. **Pricewaterhouse Coopers Advisory Services LLP**, Nova Iorque, n. 3, v. 1, 2015. Disponível em: <http://www.iab.com/wpcontent/uploads/2015/10/IAB_Internet_Advertising_Revenue_Report _HY_ 2015.pdf>. Acesso em: 06 fev. 2016.

IMMORLICA, N. et al. Click Fraud Resistant Methods for Learning Click-Through Rates. In: CONFERENCE ON WEB AND INTERNET ECONOMICS, 10., 2005, Hong Kong. **Anais**… Heidelberg: Springer Berlin, 2005. p. 412-432.

INTRIERI, Jane. **2014 Trends in Digital Marketing**. [S.I.]: Yahoo! Small Business Advisor, 2014. Disponível em: <https://smallbusiness.yahoo.com/advisor/2014-trends-digital-marketing-032703222.html>. Acesso em: 19 abr. 2014.

JAKOBSSON, M.; MACKENZIE, P.; STERN, J. Secure and Lightweight Advertising on the Web. In: WWW INTERNATIONAL CONFERENCE ON WORLD WIDE WEB, 8., 1999, Toronto. **Anais**… Amsterdã, Holanda: Elsevier Science, 1999. p. 54-75.

JAKOBSSON, M.; RAZMAN, Z. **Crimeware**: Understanding New Attacks and Defenses. Londres: Addison-Wesley Professional, 2008. 608 p.

JANSEN, B. J.; MULLEN, T. Sponsored search: an overview of the concept, history, and technology. **International Journal of Electronic Business**, vol. 6, No. 2, 2008.

JUELS, A., JAKOBSSON, M., JAGATIC, T. Cache cookies for browser authentication. In: IEEE SYMPOSIUM ON PRIVACY AND SECURITY, 12., 2006, Oakland, 2006. **Anais**... Palo Alto: IEEE CS Press, 2006. p. 32-43.

JUELS, A., STAM, S., JAKOBSSON, M. Combating Click Fraud via Premium Clicks. In: USENIX SECURITY SYMPOSIUM, 16., 2007, Berkeley. **Anais**… Berkeley: Usenix Association, 2006. p. 43-76. ISBN: 111-333-5555-77-9

KALNE, P. V.; KOLHE, V. L. A new avatar dynamic image based CAPTCHA service on cloud for Mobile devices. In: INTERNATIONAL CONFERENCE ON ADVANCES IN COMMUNICATION AND COMPUTING TECHNOLOGIES, 3., 2014, Dheli. **Anais**… Mumbai: IEEE. 11 aug 2014. p. 1-6. ISBN: 978-1-4799-7318-7

KANG, S. R.; LEE, E. Investigating Elements on the E-Commerce Homepage: Focus on business to customer websites. In*:* INTERNATIONAL SYMPOSIUM ON DESIGN CONFERENCE, 6., 2003, Tsukuba, Japão. **Anais**... Tsukuba: International Conference Proceedings. 2 mai. 2003. p. 459-473.

KITTS, B. et al. Click Fraud Detection with Bot Signatures. In: IEEE INTERNATIONAL CONFERENCE ON INTELLIGENCE AND SECURITY INFORMATICS, 6., 2013, Seattle. **Anais**... Seattle: IEEE. 7 jun. 2013. p. 146-150.

KHAN, I. et al. **The Rise of Ad Networks: An In-Depth Look at Ad Networks**. [S.I.]: JP Morgan Virtual, 2014. Disponível em: <http://www.mediamath.com/docs/JPMorgan.pdf>. Acesso em 20 mai. 2014.

KORAKAKIS, M.; MAGKOS, E.; MYLONAS, P. Automated CAPTCHA Solving: An Empirical Comparison of Selected Techniques. In: INTERNATIONAL WORKSHOP ON SEMANTIC AND SOCIAL MEDIA ADAPTATION AND PERSONALIZATION, 9., 2014, Corfu. **Anais**... Corfu: IEE. 7 nov 2014. p. 44-47. ISBN 978-1-4799-6813-8

KOTLER, P.; KELLER, K. L.. **Marketing Management**. 13th ed, Upper Saddle River, NJ: Prentice-Hall, 2009. ISBN 978-0131457577.

KRAWCZYK, H.; BELLARE, M.; CANETTI, R. HMAC: Keyed-Hashing for Message Authentication - RFC2104. **Network Working Group. Request for Comments***: 2104 (Online)*, [S.I.]: RFC, 1997. Disponível em: <https://tools.ietf.org/html/rfc2104>. Acesso em: 09 fev. 2016.

LEYDEN, J. **Miaow to kitten-based authentication**. [S.I.]: The Register, 2006. Disponível em: <http://www.theregister.co.uk/2006/04/12/kittenauth>. Acesso em: 11 de Abril de 2015.

LIEDTKE, M. Google to Pay $90M in '*Click Fraud*' Case. **Washington Post Magazine**. Washington, Estados Unidos. 11 mar. 2006.

LING-ZI, X.; YI-CHUN, Z. A Case Study of Text-Based CAPTCHA Attacks. In: INTERNATIONAL CONFERENCE ON CYBER-ENABLED DISTRIBUTED COMPUTING AND KNOWLEDGE DISCOVERY, 3., 2012, Sanya. **Anais**... Sanya: IEEE. 12 out. 2012. p. 121-124.

LOBO, A. P. **Crimes cibernéticos custaram R$ 15,3 bilhões no Brasil**. [S.I.]: Convergencia Digital UOL (Online). São Paulo, v. 15, n. 2, 2011. Disponível em: <http://convergencia digital.uol.com.br/cgi/cgilua.exe/sys/start.htm?infoid=27750&sid=18>. Acesso em: 03 dez. 2011.

MAJUMDAR, S.; KULKARNI, D.; RAVISHANKAR, C. Addressing *Click Fraud* in Content Delivery Systems. In: IEEE CONFERENCE ON COMPUTER COMMUNICATIONS*,* 26., 2007, Anchorage, Alaska. **Anais**… Palo Alto: IEEE CS Press, 2007. p. 626-643.

MANN, C. How Click Fraud Could Swallow the Internet. **Wired Magazine**. São Francisco: Wired, v. 14, n.1, pp 17-20, 2006.

MANTAS, J. An Overview of Character Recognition Methodologies. **ScienceDirect**. Amsterdã: Elsevier, vol. 19, n. 6, pp 425-430, 1986.

METWALLY, A.; AGRAWAL D.; ABBADI, E. Duplicate Detection in Click Stream. In: WWW International Conference of the World Wide Web Conference, 14., Chiba, Japan, 2005. **Anais**... Nova Iorque: ACM Digital Library, 2005. p. 137-166.

METWALLY, A.; AGRAWAL D.; ABBADI, E. DETECTIVES: Detecting Coalition Hit Inflation Attacks in Advertising Networks Streams. In: WWW International Conference of the World Wide Web Conference, 16., Alberta, Canada, 2007. Alberta, Canada, 2007. **Anais**... Nova Iorque: ACM Digital Library, 2007. p. 88-100.

MIRANDA, M. B. **Teoria Geral dos Contratos**. [S.I.]: Revista Virtual Direito Brasil, São Paulo, v.2, n.2, jul./dez. 2009. Disponível em: <http://www.direitobrasil.adv.br/artigos/cont.pdf>. Acesso em: 03 dez. 2014.

MOSCARITOLO, A. FBI arrests six in click-fraud cyber scam that netted $14M. **SC Magazine**. Nova Iorque: 09 nov. 2011. Disponível em: <http://www.scmagazine.com/fbi-arrests-six-in-click-fraud-cyber-scam-that-netted-14m/article/216399/>. Acesso em: 08 fev. 2016.

MOTOYAMA et al. Re:CAPTCHA – Understanding CAPTCHA-Solving Services in an Economic Context. In: USENIX SECURITY SYMPOSIUM, 19., 2010, Washington. **Anais**... Washington, DC. 13 ago. 2010. p. 76-88.

NACHI, A V. **Sorry Google, No CAPTCHA reCAPTCHA doesn't stop bots**. Disponível em <http://www.shieldsquare.com/sorry-google-CAPTCHA-reCAPTCHA-doesnt-stop-bots>. Acesso em: 31 jan. 2015.

NAOR, M. **Verification of a Human in the Loop or Identification via the Turing Test**. [S.I.] Wisdom Journal, 2015. Disponível em <http://www.wisdom.weizmann.ac.il/~naor/PAPERS/human.pdf>. Acesso em: 17 mar. 2015.

NARAINE, R. **Feds Arrest Google Extortionist**. [S.I.]: InternetNews (Online), 2014. Disponível em: <http://www.internetnews.com/bus-news/article.php/3329281>. Acesso em 11 ago. 2014.

NORBU, K.; BHATTARAKOSOL, P. Factors towards the effectiveness of CAPTCHA. In: International Conference on Computing and Convergence Technology, 7., 2012, Seoul. **Anais**... Seoul: IEEE. 5 dez 2012. p. 566-570.

OENTARYO, R. et al. **Detecting Click Fraud in Online Advertising**: A Data Mining Approach. [S.I.]: Journal of Machine Learning Research, Illinois, vol 15, pp. 99-140, 2014. Disponível em: <http://www.jmlr.org/papers/v15/oentaryo14a.htm>. Acesso em: 02 jan. 2015.

O'REILLY, L. **Google's new CAPTCHA security login raises 'legitimate privacy concerns**. [S.I.]: Business Insider, Tech Insider. Nova Iorque, NY, Estados Unidos: 20 fev. 2015. Disponível em: <http://www.businessinsider.com/google-no-captcha-adtruth-privacy-research-2015-2>. Acesso em: 10 mar. 2015.

O'REILLY, T. **What is Web 2.0: Design Patterns and Business Models for the Next Generation of Software**. [S.I.]: SSRN, 2012. Disponível em: <http://papers.ssrn.com /sol3/papers.cfm? abstract_id=1008839&download=yes" >. Acesso em 23 mai. 2014.

O'TOOLE, J. **Pop-up ad creator: 'I'm sorry'**. [S.I]: CNN Money Online, Nova Iorque. 15 ago. 2014. Disponível em: <http://money.cnn.com/2014/08/15/technology/pop-up-ads/index.html >. Acesso em 06 de fevereiro de 2016.

OU, H. Maximization of Online Display Advertising Slots. In: INTERNATIONAL CONFERENCE ON BUSINESS INTELLIGENCE AND FINANCIAL ENGINEERING, 5., 2012, Lanzhou. **Anais**... Lanzhou: IEEE. 21 ago. 2012. pp. 65-68.

PAWLOWSKI, M. P.; JARA, A. J.; OGORZALEK, M. J. Extending Extensible Authentication Protocol over IEEE 802.15.4 Networks. In: INTERNATIONAL CONFERENCE ON INNOVATIVE MOBILE AND INTERNET SERVICES IN UBIQUITOUS COMPUTING, 8., 2014, Birmingam. **Anais**... Birmingham: IEEE. 4 jul. 2014. p. 340-345.

PEARCE, P. et al. Characterizing Large-Scale Click Fraud in ZeroAccess. In: ACM CONFERENCE ON COMPUTER AND COMMUNICATIONS SECURITY, 21., 2014, Scottsdale, Arizona. **Anais**... Arizona: ACM Digital Library. 15 jun. 2014. p. 288-301.

PEREIRA, Maria Neuma. **Processo Digital – A Tecnologia Aplicada na Solução de Conflitos**. São Paulo: Biblioteca 24horas, 2011.

PUTTACHAROEN, R.; BUNYATNOPARAT, P. Protecting cookies from Cross Site Script attacks using Dynamic Cookies Rewriting technique. In: INTERNATIONAL CONFERENCE ON ADVANCED COMMUNICATION TECHNOLOGY, 13., 2011, Seoul. **Anais**.... Seoul: IEEE. 16 fev. 2011. p. 1090-1094.

RAMASUBRAMANIAN, L. Client Behavior and Feed Characteristics of RSS, A PublishSubscribe System for Web Micronews. In*:* INTERNET MEASUREMENT CONFERENCE, 5., Berkeley, California. **Anais**... Berkeley: UNENIX Association, 2005.

RESEARCHANDMARKETS. **Global Online Advertising Industry 2013-2018**: Trends, Profits and Forecast Analysis. [S.I.]: Research and Markets. Nova Iorque, 2013. Disponível em: <http://www.researchandmarkets.com/reports/2558340/global_online_advertising _industry_20132018 >. Acesso em: 19 abr. 2014.

REVAR, A. G.; BHAVSAR, M. D. Securing user authentication using single sign-on in Cloud Computing. In: NIRMA UNIVERSITY INTERNATIONAL CONFERENCE ON ENGINEERING, 3., 2011, Ahmedabad. **Anais**... Ahmedabad, Gujarat: IEEE. 10 dez 2011. p. 1-4.

RYAN, K. M. **Big Yahoo Click Fraud Settlement**. [S.I.]: iMediaConnection.com Online. Londres, UK: iMediaConnection. 2006. Disponível em: <http://www.imediaconnection.com/ content/10294.asp>. Acesso em: 04 set. 2014.

SAGAR, C. **SEO**: A Quick Primer on the Difference Between Ecommerce and Content Sites. [S.I.]: People to Work With (P2W2), Nova Iorque. 15 dez. 2009. Disponível em: <http://

www.p2w2.com/blog/index.php/a-quick-guide-to-seo-our-article-on-open-forum/>. Acesso em: 23 mai. 2014.

SAGIN, E. **Four Powerful Ways to Eliminate Click Fraud in Your Account**. [S.I.]: Wordstream: Online Advertising Made Easy. Boston, MA: 17 ago. 2015. Disponível em: <http://www.word stream.com/blog/ws/2015/08/17/click-fraud>. Acesso em: 08 fev. 2016.

SEDIYONO, E.; SANTOSO, K. I. Secure login by using One-time Password authentication based on MD5 Hash encrypted SMS. In: INTERNATIONAL CONFERENCE ON ADVANCES IN COMPUTING, COMMUNICATIONS AND INFORMATICS, 7., 2013, Mysore. **Anais**... Mysore: IEEE. 25 ago. 2013. p 1604-1608.

SHATNAWI, M.; ABDALLAH, S. Improving Handwritten Arabic Character Recognition by Modeling Human Handwriting Distortions. **ACM Transactions On Asian And Low-Resource Language Information Processing**. v. 15, n. 1. Nova Iorque: ACM Digital Library. 01 jan. 2016.

SHET, Vinay. **Não sou um robô! Conheça o reCAPTCHA sem CAPTCHA**. [S.I.]: Google Blog, Mountain View, CA. 2014. Disponível em: <http://googlebrasilblog.blogspot.com.br /2014/12/nao-sou-um-robo-conheca-o-reCAPTCHA-sem_3.html>. Acesso em: 02 mai. 2015.

SHEKHAR, S. **Online Marketing System**: Affiliate marketing. [S.I.]: FeedMoney.com. Washington. 2009. Disponível em: <http://web.archive.org/web/20110501002424/http://feed money.com/ archives/2009/06/29/online-marketing-system-affiliate-marketing>. Acesso em: 06 fev. 2016.

SMITH, C. S. **Geolocation**: Core to the Local and Key to Click-Fraud Detection. [S.I]: Search Engine Land, Redding, CT. 13 aug. 2007. Disponível em: <http://searchengineland.com/ 070813-082025.php>. Acesso em: 21 mai. 2015.

SOMMERVILLE, I. **Software Engineering**. 10 ed. 816 pp. Nova Jérsei, NY, Estados Unidos: Pearson. 3 abr. 2015. ISBN: 0133943038

STONE, B. **Marketing Direto**, São Paulo: Nobel, 2004. 570pgs.

TAMANG, T.; BHATTARAKOSOL, P. Uncover impact factors of text-based CAPTCHA identification. In: INTERNATIONAL CONFERENCE ON COMPUTING AND CONVERGENCE TECHNOLOGY, 7., 2012, Seoul. **Anais**... Seoul: IEEE. 5 dez. 2012. p. 556-560,

TEIXEIRA, M. A. M. **Suporte a serviços diferenciados em servidores web**: modelos e algoritmos. São Carlos, SP: ICMC, 2004. Originalmente apresentada como tese de doutorado, ICMC, USP, São Carlos, SP, 2004.
TURING, A. M. **Computing Machinery and Intelligence**. Oxford, Inglaterra: Mind, 1950. v. 59.

TUZHILIN, A. **The Lane's Gifts v. Google report**. [S.I.]: Independent evaluators assessment of quality of Google`s click-fraud filtering methods, New York, 22 jul. 2006. Disponível em: <http://googleblog.blogspot.com/pdf/Tuzhilin_Report.pdf>. Acesso em: 17 fev. 2014.

USKOV, A. V. Information Security of IPsec-based Mobile VPN: Authentication and Encryption Algorithms Performance. In: IEEE INTERNATIONAL CONFERENCE ON TRUST, SECURITY AND PRIVACY IN COMPUTING AND COMMUNICATIONS, 11., 2012, Liverpool. **Anais**... Liverpool: IEEE. 27 jun 2012. p. 1042-1048.

VENOSA, Silvio de Salvo. **Direito Civil**: Teoria Geral das Obrigações e Teoria Geral dos Contratos. São Paulo: Atlas, 2014.

WEISER, P. J. **The future of Internet Regulation**. University of California, Davis. L. Rev 509. Boulder, CO: UCLA. 2009.

WEINER, L. It's an Ad, Ad World. Ideas and Resources for Building Business. **CIO Web Business**. vol 11, no 14, Section 2, pp 74-78. Framingham, MA: International Data Group, 1 mai. 1998.

WROBLEWSKI, L. **A sliding alternative to CAPTCHA?** [S.I.]: Luke W Ideation and Design Online, São Francisco. 1 jul 2010. Disponível em: <http://www.lukew.com/ff/entry.asp?1138>. Acesso em: 3 jul. 2015.

XIAOGUANG, L. **Image Analysis for Face Recognition**. [S.I.]: Dept. of Computer Science & Engineering, Michigan State University. Lansing, MI, 2003. Disponível em: <http://www.face-rec.org/interesting-papers/general/imana4facrcg_lu.pdf>. Acesso em 01 dez. 2015.

XU, LU. A facial recognition method based on 3-D images analysis for intuitive human-system interaction. In: INTERNATIONAL JOINT CONFERENCE ON AWARENESS SCIENCE AND TECHNOLOGY AND UBI-MEDIA COMPUTING, 21., 2013, Aizuwakamtsu. **Anais**... Aizuwakamatsu, Japão: IEEE. 4 nov. 2013. p. 371-377.

YANG, G.; LEI, Z.; WANG, H.; DOU, Y. Analysis of the RADIUS Signalling Based on CDMA Mobile Network. In: INTERNATIONAL CONFERENCE ON INTELLIGENT HUMAN-MACHINE SYSTEMS AND CYBERNETICS, 6., 2014, Hangzhou. **Anais**... Hangzhou: IEEE. 27 ago 2014. p 270-274.

ZAMANZADEH, B. et al. **Semantic Advertising**. [S.I.]: CoRR, University of California Irvine, Los Angeles: DataPop Inc. 2013. Disponível em: <http://arxiv.org/ftp/arxiv/papers/1309/1309.5018.pdf>. Acesso em: 06 fev. 2016.

ZELLER JR, T. With Each Technology Advance, a Scourge. **The New York Times**, Nova Iorque, 17 out., 2004.

ZHU, B B. et al. **Captcha as Graphical Passwords—A New Security Primitive Based on Hard AI Problems**. [S.I.]: IEEE Transactions on Information Forensics and Security. Siena, Italia: IEEExplore. Vol 9. No 6. p. 891-903. Disponível em: <http://ieeexplore.ieee.org/xpl/aboutJournal.jsp?punumber=10206#AimsScope>. Acesso em: 15 jun. 2014.