Exploring the Feasibility of a Scalable Internet Payment Architecture

Ed Oakes
oakes@cs.wisc.edu

Keith Funkhouser
wfunkhouser@cs.wisc.edu

Abstract

Online advertising revenue continues to climb higher each quarter. The growth of ad-blockers has been cause for alarm for many of the primary stakeholders in this complex ecosystem. Previous attempts have been made, unsuccessfully, to incorporate micropayments for content producers into the fabric of the internet. Here, we investigate scalable, performant, and easy to use extensions of the internet payment architecture which address the concerns of both internet users and publishers.

1 Overview

Historically, online advertising has served as the primary source of revenue for content creators. This framework has catalyzed the rapid growth of the internet, as consuming quality content for free is extremely attractive to users. However, producing such content is not free for publishers who are currently left with two options: sell space on their site to advertisers, or move to a subscription-based model. Advertisements have largely won this battle, partially due to a few key examples of the paywall architecture failing to retain users [26]. This has led to a modern internet which is filled with publishers entirely reliant on the ads that keep their lights on.

Here, we explore an alternative payment system, Harpocrates, which provides users with a choice to avoid advertisements on the internet and still compensate publishers. In designing the architecture, we focus on two key factors: convenience and scalability. The ultimate goal is to minimize the burden on the user by providing a payment system which integrates into the current internet ecosystem, while designing an architecture flexible enough to scale with the internet.

We also describe proof of concept browser extension modifications, "*Un*acceptable Ads", which allow users to specify the quantity and type of ads that they want to block, offering an immediate compromise between the negatives of ads for users and the need for publishers to be compensated.

Ultimately, the research question explored was:

Can we devise a scalable internet payment architecture that supports ad-free browsing for users and fair compensation for content publishers?

2 Background

2.1 Online Advertising

In fiscal year 2016, internet advertising revenues in the United States totaled $72.5 billion (a 21.8% increase over 2015) [20]. Increasingly, this money is going to mobile advertising; in the first half of 2016, mobile surpassed search for the first time as the leading ad format [36]. In such a massive and diverse industry, the technology surrounding the online ad ecosystem has grown exceedingly complex. Online ads can be sourced via premium campaigns, ad networks, or ad exchanges and involve detailed targeting and accounting mechanisms [23]. The number of players involved is constantly increasing, raising the barrier to disrupt the industry higher than ever [30].

2.2 Ad-blockers

The growth of ad-blockers in recent years has been cause for alarm for many current online advertising stakeholders. In a recent study of 2 million users, Malloy et al. found U.S. ad-blocker penetration to be 15.7-18.6% (95% CI) [31]. An active measurement study by Pujol et al. estimated that 22% of users were likely using ad-blockers [37]. Surveys by various entities in the advertising industry have estimated that 26% of desktop users [19] and 37% of mobile users [14] are blocking ads in the U.S. Another industry survey estimates that 309 million are using mobile ad-blockers globally (16%

of global smartphone users) [17]. A simple back-of-the-envelope calculation reveals the scale of the impact this has on content providers: using the most conservative of the above ad-blocker estimates (15.7%) and the $72.5 billion spent in FY 2016, publishers, marketers, and others are missing out on approximately $11.38 billion in revenue per year due to ad-blocking users. We note that this may even be an aggressively conservative estimate, as a 2015 study by Adobe and PageFair estimated the cost to be $20.3 billion in 2015, and predicted that it would grow to $41.4 billion by 2016 [12]. As both the quantity of money invested into digital advertising and number of ad-blocking users grow, this figure is becoming increasingly significant [36].

In response to this growth in ad-blocking technology, a new breed of partnerships are forming. Eyeo GmbH, which makes Adblock Plus (installed on over 100 million active devices [15]), began its Acceptable Ads program in 2011, which allows large entities to pay in order to have their ads whitelisted for its users. As of 2015, this list was known to include Google, Microsoft, Amazon, and Taboola [13]. Increasingly, the battle over ad-blockers is being fought in the courtroom. Recently, a German court banned Eyeo from collecting money from German publishing house Axel Springer [16]. These battles point to the higher-level rejection of advertisements by users, and suggest that a more sustainable compensation model is necessary for the long-term internet to continue to attract users and reward content producers.

3 Related Work

3.1 Academic

The idea of using "digital cash" to facilitate transactions online experienced substantial popularity in the late 1980's and early 1990's. Chaum was one of the first to introduce the concept of anonymous, untraceable micropayments (then, a "blind signature cryptosystem") [24]. Later, this work was expanded to prevent duplication of coins [25], and eventually spun off into the company DigiCash [40]. Dukach's SNPP (Simple Network Payment Protocol) represents one of the first fully-described protocols for micropayment [27]. Wheeler [43] and Rivest [38] studied "probabilistic" micropayment schemes, in which the "bank" only cashes a small percentage of "checks", but in the long run everyone ends up paying the correct amount of money. This work spawned the startup Peppercoin, which was eventually acquired. Work at DEC on Millicent [32] inspired Rivest and Shamir's work on two schemes, PayWord (credit-based) and MicroMint (less secure but faster) [39], although these same ideas had been proposed shortly before by Anderson et al. [22]. IBM also developed a micropayment-based extension for their *i*KP macropayment system [28].

Although each of these projects had unique approaches to architecture and implementation details, the design parameters primarily revolved around some subset of anonymity, reliability, trust, scalability, usability, and security. Some work has been done to classify and characterize these manifold schemes across various axes [21, 34, 33]. They each foresaw the advent of small transactions on the Web for content like music (e.g., $1 songs on iTunes), however the majority of work was invested in the technology rather than supporting and increasing adoption. In particular, the need for widespread adoption of a micropayment system by both customers and sellers led to the eventual death of each system in its own time.

3.2 Commercial

Interest in online microtransactions has experienced a growing resurgence over the last decade. As the use of ad-blocking software continues to grow worldwide, publishers and users alike realize that we must consider new models for financing the Web as we know it.

One such attempt is a system which participates in ad auctions on behalf of the end user, such as Google's Contributor program [9], which is now discontinued, and Atri [3]. Though this approach will tend to find the market value for space on a given page, it includes network and computational overhead in order to participate in auctions before serving the page. This limits the performance benefit of removing ads from internet browsing, and also requires building and installing software to participate in each ad exchange. Additionally, Google Contributor was limited exclusively to advertisements served via Google AdSense, which resulted in a user experience which was only partially ad-free [8].

Brave, a fork of the Firefox project, is a new browser which allows users to divide up a monthly subscription to the sites that they visit most often via Bitcoin-based micropayments [6]. The project uses the cryptographically secure Anonize protocol [29] to prevent a user's browsing history from being linked to their identity. Systems like these will face scalability concerns with organizing payments when dealing with a large number of content producers and consumers. Additionally, requiring the use of an entirely different browser adds a barrier to entry for new users.

Sourcepoint [11] and Blendle [5] offer consumers a content pass which allows them access to a variety of content providers for a monthly subscription. This approach is very similar to the "paywall" which is currently implemented on many major publisher sites, and will require cooperation from many large publishing players in

order to be feasible at a larger scale.

SatoshiPay [10] and Autotip [4] allow users to send micropayments in Bitcoin. However, Bitcoin transaction fees have increased so much in recent years that micropayments using the currency are almost infeasible.

Flattr Plus [7] (a partnership between Flattr and Adblock Plus) allows consumers to distribute a monthly subscription among selected content provides. Although the intersection of these two user bases is quite large (adblocker users and those who want to contribute to content providers), the implementation currently uses a browser extension to track user behavior, leading to privacy concerns.

4 Harpocrates

In this section, we present our design for a system, Harpocrates (Greek god of silence, secrets, and confidentiality), which allows users to easily compensate publishers at a fair rate for their content without being forced to view ads. Harpocrates is composed of a client-side browser extension for users to configure their preferences and a server for bidding on ad placements and accounting of payments between users and publishers. The main focus of Harpocrates is that it fits into the current ad ecosystem with minimal implementation work required for publishers.

4.1 Design Considerations

In designing Harpocrates, there was a clear set of requirements which, if not accomplished, would yield a system infeasible for wide deployment.

First, the system must address the concerns of current ad-blocker users. This is the number one design principle, as if the system doesn't provide users with at least a subset of the benefits they receive from blocking ads, they won't even consider using it. We define these concerns as three main considerations: intrusiveness, performance, and security/privacy, derived from a 2016 HubSpot survey of 731 Adblock Plus users on why they block ads [18].

The system must also be as unobtrusive to the user as possible. Web users are accustomed to convenience and an uninhibited browsing experience, and our system should follow this pattern to attract users. For example, with our current design, there is a very simple settings page which a user need only visit once at minimum, allowing them to specify their budget. Beyond that everything is handled on the back-end.

The system should fit into the current Web ecosystem. This means that we cannot rely on any new standards or widely-implemented protocols in order to see wide adoption. Correspondingly, the implementation cost for publishers must be minimized, because it will never see wide adoption if it requires a complex or custom implementation for each publisher. This excludes solutions that involve adding significant server-side code, such as nearly all of the services described in Section 3.1. Another key piece to fitting into the Web payment infrastructure is that our system needs to be able to coexist with ad networks as they exist today. Many Web publishers survive exclusively off of ad revenue, and they cannot be expected to switch to an entirely new compensation model which abandons the tried-and-true ad revenue model.

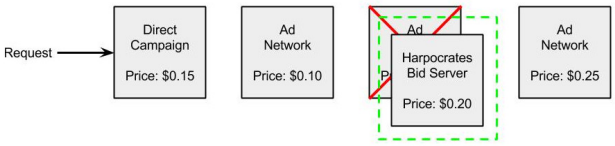

Figure 1: Harpocrates sits inline with the current ad exchange ecosystem.

Finally, the system must provide control over the browsing experience for both users and publishers alike. Currently, publishers are essentially locked in to using a combination of subscription and ad tactics for revenue and users are locked in to paying for these subscriptions and seeing ads across the Web. We want to provide an alternative which can be opted into and out of easily for both publishers and users, which lets them shape their own browsing experience. By enabling publishers to recoup some of the lost revenue due to blockers, and enabling users to opt-in to funding their favorite content, both parties benefit.

4.2 Architecture

Ultimately, Harpocrates can be seen as a Web ad network, which places "blank ads" instead of images or videos and whose advertisers (those paying for the ad placements) are the users themselves. Users specify a monthly "budget," which Harpocrates then uses to bid on each ad that a publisher wants to sell. This design very cleanly fits into the current ecosystem, and requires very minimal effort from the user: they only have to register with one entity, Harpocrates, and they need only set their preferences at least once in order to get off the ground, directly compensating publishers and seeing fewer ads. This solution relies on fact that publishers ad servers and ad exchanges are efficient marketplaces for determining the fair value for a given piece of "real estate" on the internet. In order to take advantage of this fact, Harpocrates sits inline in the ad "waterfall" that currently exists in many publisher ad servers (Figure 1).

From the publisher's point of view, Harpocrates is es-

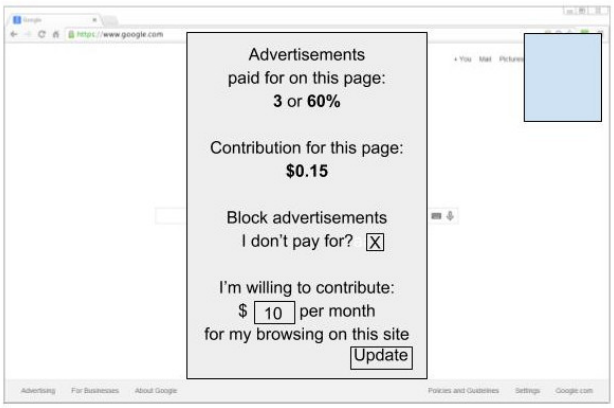

Figure 2: For the user, Harpocrates simply requires installing a browser extension and initializing a few settings.

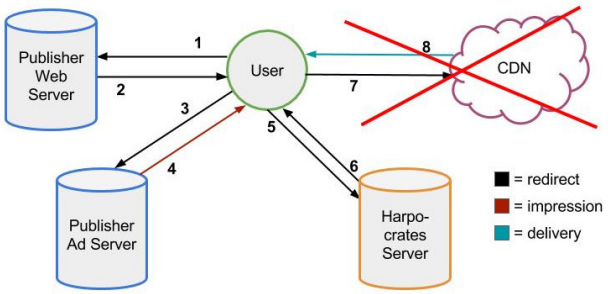

Figure 3: Harpocrates removes the actual ad media element delivery process, cutting out a roundtrip request to the ad server content delivery network and improving performance.

sentially no more than another ad network, which, given an impression, returns a bid for the ad (or none) based on the user's budget and the estimated value of the ad space. This allows the publisher to choose whether or not to use Harpocrates at a per-ad granularity, and its implementation cost is no higher than that of adding another ad network, requiring no additional infrastructure or protocol that isn't already there. The only piece of information which needs to be shared between publisher and Harpocrates is a user ID which associates the user who requested the page and their Harpocrates account, but this can be passed as one of the cookies which are already passed around to identify Web users.

From the user's point of view, Harpocrates is simply a browser extension (Figure 2) which they pay some chosen amount of money to each month, and it in turn removes some of the ads that they see on their favorite sites. They don't need to have any knowledge of the complex ad ecosystem, only that they are directly compensating the content publishers. In addition, users can elect to fall back on blocking ads even when the Harpocrates bid server does not win a given auction.

Finally, Harpocrates also acts as a clearinghouse, settling accounts monthly. For a given set of publishers N and users M, only $O(|N|+|M|)$ payments need to be transacted (i.e. one payment from each user to Harpocrates, and one payment from Harpocrates to each publisher). At scale, we gain a lot from this aggregation of payments when compared to a system such as paywalls, which requires $O(|N|\cdot|M|)$ payments (one for each user/publisher pair) in the worst case.

These aspects of the architecture make Harpocrates a good candidate solution, given the aforementioned design considerations. It addresses the needs of current adblocker users by showing them fewer ads, the user interface makes it practically invisible from a user's point of view, and performance is improved by eliminating the actual delivery of ad media content (Figure 3). Harpocrates also fits into the current complex ecosystem, requiring no new standards or protocols. Finally, the implementation is simply adding another ad network, minimizing implementation costs. The one design point which we do not hit is that of user security/privacy concerns. Unfortunately, because Harpocrates doesn't restrict publishers from serving ads, and needs to be able to make market-rate bids on ad space in order to serve its purpose, user tracking is still necessary. Despite this, however, we believe that Harpocrates' scalability and ease of use for both users and publishers makes it a promising alternative to the ad-based payment model of the internet as we know it today.

5 *Un*acceptable Ads

*Un*acceptable Ads is a proof of concept to illustrate a more immediate path forwards for users who block ads because of the reduced performance and/or visual obtrusiveness. Relying on the idea that users are willing to see some ads in order to support content publishers, our goal is to allow users to specify the number and type of ads that they do not want to see. This idea is similar in motivation to the Acceptable Ads program run by Adblock Plus, but where Acceptable Ads are determined by the Adblock Plus organization and advertisers can pay to be whitelisted, our system will put the control directly in the user's hands. To realize this idea, we have modified two Chrome extensions: uBlock Origin, a request-based blocker, and a "perceptual blocker" which was recently developed by Storey, et al. [41] at Princeton University.

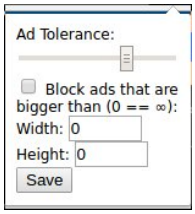

Figure 4: User interface for our modification of the perceptual blocker.

5.1 uBlock Origin

uBlock Origin is a popular open source desktop adblocker with nearly 9 million installs on Google Chrome as of May 2017. This blocker is categorized as a request-based blocker, meaning that it recognizes Web requests which are targeted at ad-related servers and prevents them from ever leaving the client's browser. This strategy offers significant performance benefits, as it reduces network traffic and prevents (often large) media elements from being received and rendered in the client browser. Identifying ad-related network requests is done by matching target URLs against a list of known advertising URLs (e.g., EasyList [35]). The extension uses multiple such community-maintained lists, and matches against them using a combination of simple lookups and regular expressions.

Our modification to the uBlock Origin is a notion of "Ad Tolerance", a configurable setting from 0-100% which determines what proportion of ads will be shown to the user on any given page. If the user has their tolerance set to 0%, the extension will block all ads like uBlock Origin usually does and if the user has their tolerance set to 100% then the extension won't function as an ad-blocker at all. Any setting between 0 and 100% will result in each ad element probabilistically being blocked (Figure 5). The reason for probabilistic blocking is that it is a very difficult problem to predict the number of total ads that will be shown on a page before the page has fully rendered, so blocking a pre-determined number of elements is impractical. While probabilistic dropping means that the proportion of ads blocked might not be exactly as expected on every page, in the long run the number of ads blocked will very closely match the specified proportion.

5.2 Perceptual Blocker

Storey, et al. [41] recently published work on what they dubbed a "perceptual blocker", which blocks ads based on their visual content rather than their associated Web requests or the structure of the page. The primary observation which this work is based on is that there have been

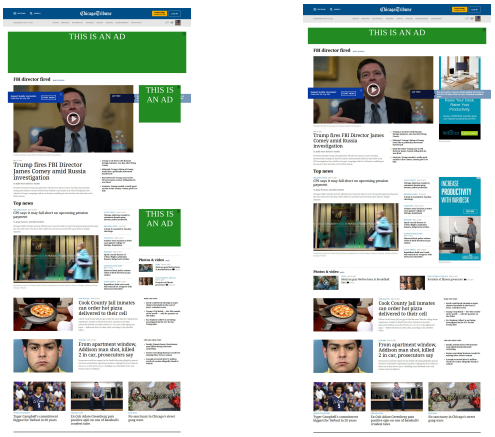

Figure 5: *Un*acceptable Ads allows the user to select their tolerance for advertisements, from 0% (left) upwards (right).

many efforts recently to standardize ad policies on the internet through joint coalitions such as AdChoices [1]. Such programs require ads to identify themselves by including text such as "from our advertisers" or by displaying a common logo, such as the AdChoices logo. Storey, et al. recognized that this offers an opportunity to visually recognize and identify ads as a human would be able to. The blocker that they developed puts this idea into practice, and covers all visually recognized ads with a banner reading "THIS IS AN AD." Covering ads rather than completely hiding them is deliberate, as the researchers wanted to avoid potential backlash from the advertising community, which is discomforted by the idea that their ads could be completely delivered but still not presented to a user - leaving them paying for an impression which didn't actually happen. This raises a broader question about the morality of perceptual blockers, and suggests that if these blockers were to become popular, advertisers would have to make a significant engineering effort to maintain accurate impression accounting in their presence.

Our first modification to the perceptual blocker is the same as that which we made to uBlock Origin, adding a configurable tolerance level which probabilistically blocks the ads it encounters. We also added a second configuration to this blocker, which allows the user to specify a maximum width and height of ad elements which they are willing to see (Figure 4). This highlights a key benefit to the perceptual blocker, because the ads are completely delivered before blocking occurs, the blocker has a more complete set of information about the ad's qualities (e.g., size, content, placement on the page) when the blocking decision is made. We use this additional information to block ads above a certain size, but

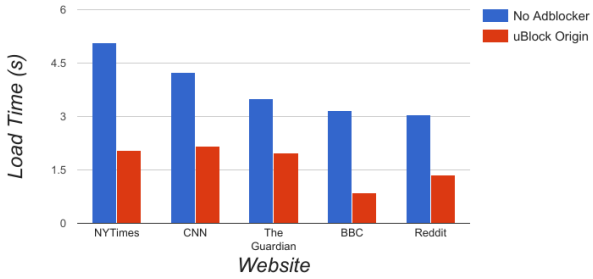

Figure 6: Loading times for five popular Websites with and without the popular ad-blocker uBlock Origin.

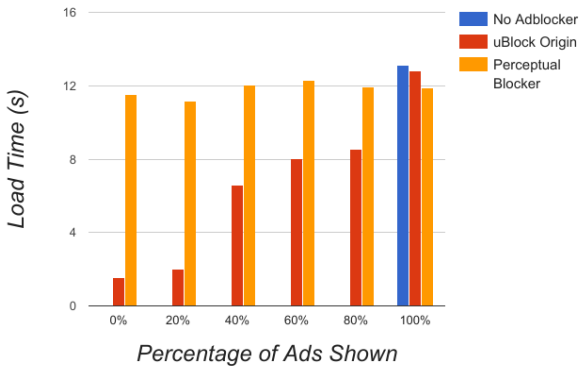

Figure 7: Loading times for Chicago Tribune homepage with varying levels of tolerance for ads using the *Un*acceptable Ads extensions.

this could be extended to other configurations, such as recognizing and blocking inappropriate ads.

5.3 Performance

In order to assess the impact of ad-blockers on performance, we performed an experiment to measure page load times with and without uBlock Origin enabled. The five sites which we chose to measure load times on were the five most popular news sites based on the Alexa ranking [2]. We loaded each site 10 times under each condition and averaged the results, closing the browser between loads to ensure that a full rendering occurred. Page load times were measured using the Chrome extension "Page Load Time" [42], which considers a page fully loaded once all elements have finished rendering. We deliberately chose not to clear the local browser cache between page loads, recognizing that in practice Web content is often cached locally. The results validate survey findings that many users use ad-blockers for performance reasons [18], as there was an average of a 2.33x speedup in load time with uBlock Origin enabled (Figure 6).

To assess the performance of our *Un*acceptable Ads extensions, we performed a similar experiment by measuring page load times for the Chicago Tribune homepage at multiple different levels of ad tolerance. Our findings are consistent with our intuition: since uBlock Origin blocks based on network requests, performance should decrease as tolerance goes up. For the perceptual plocker, however, performance stays relatively constant regardless of tolerance, since ads are detected and hidden after they are loaded (Figure 7). This difference highlights the key benefit to request-based blockers: performance. Preventing Web requests from ever leaving the client browser results in a decrease in network traffic and the costly rendering of media elements. One thing to note is that we consider a page to be done loading when the page is fully rendered, but this doesn't include the additional time required in the case of the perceptual blocker to identify and cover ads. While this occurs quickly, there is some computational overhead associated with visually scanning the Web page which should be considered when comparing the two techniques.

5.4 Summary and Technical Challenges

These extensions serve as two separate proofs of concept for a new generation of ad-blockers which could allow for a more fine-grained tuning of what types of ads are blocked. This approach can provide partial or full compensation for the publisher while still addressing user concerns about the performance and/or visual issues associated with ads. We note that this solution does *not* address the needs of users whose foremost concern is privacy, and leave that concern for future work.

There was a large difference in the technical challenge presented by making modifications to uBlock Origin compare to the perceptual blocker. In the case of the perceptual blocker, the ad recognition and covering sections of the code were completely separate, allowing us to add our tweaks to the covering code (only called when an ad is recognized) in only a few simple lines. Our understanding of the extension was also aided by the fact that the perceptual blocker was well-commented and is more of a proof of concept than a production-ready piece of software, meaning that there were far fewer optimizations and less corner-case handling. This simplicity allowed us to easily fit our modifications in to only one section of the code with minimal debugging pain. On the other hand, uBlock Origin is a more complicated piece of software and includes many optimizations to reduce the memory footprint and computational overhead of its adblocking. Additionally, although the extension is open source, it is almost entirely maintained by a single developer. While there is a fairly complete wiki explaining the design of the extension, the code itself is quite bar-

ren of comments. This led to many hours spent just trying to figure out the end-to-end workflow of blocking a single ad, which involves 3 different identification mechanisms (dynamic URL filtering, dynamic host filtering, and static URL filtering) as well as a caching mechanism to prevent redundant lookups. Our first iteration simply probabilistically let through ads which had been marked by one of the identification mechanisms, but this turned out to block far too many ads. The excessive blocking occurred because a given ad may have upwards of 10 redirects and requests associated with it, and blocking any of these would prevent the ad from being served. This led us to come up with a better solution: adding a second, position-based, cache which identified the position on the page of the iframe associated with an ad-related request and made a single decision to block or allow all requests for that iframe. Although not perfect (multiple ads could have the same position, positions could change, etc.), this additional cache provided far more predictable and accurate results. This development time was definitely well-spent, as we learned not only how ad-blockers function in practice, but had the opportunity to learn some Javascript and the anatomy of browser extensions, both of which are integral parts of the modern Web and aren't something we've worked with in the past.

Looking forward, there are many more directions to explore with this idea of user-configurable ad-blocking. For example, while a request-based blocker doesn't allow for blocking based on a display size threshold, one similar option that could be implemented is blocking based on media element size (i.e., in bytes). The size of an element is often included in the metadata of a redirect from an advertiser server, so blocking based on a simple check could be implemented trivially. One issue with this approach, however, is that we must still go through the chain of redirects which determine what ad will be served and log the impression on the publisher's end. Blocking an ad after these requests have occurred puts some stress on the publisher and marketer servers, and reduce some of the performance benefit realized on the user end. Additionally, a blocker could block specific types of ad media elements which the user deems too intrusive (e.g., video advertisements or images that make up the background of the page).

5.5 Code Repository

We have made our code for the *Un*acceptable Ads extensions available here:

`https://github.com/edoakes/740adblocker`

6 Summary

The modern ad ecosystem continues to grow and evolve at breakneck pace. Although revenues continue to grow, so too does the adoption of ad-blockers, which in the limit threaten the viability of the current model for the internet. Here, we describe a system which fits into the ad ecosystem without any specialized infrastructure or standards. The system looks like an ad exchange to publishers, leading to low implementation costs and high scalability. To the user, it looks like a regular browser extension. Furthermore, it makes a feasible number of macropayments on a monthly basis to compensate publishers by users. We also describe a more immediate compromise between users and publishers, *Un*acceptable Ads. These two browser extensions, which block ads or their requests probabilistically, provide users with greater freedom in choosing to block ads they find unacceptable while still compensating publishers.

7 Conclusion

In such a high-stakes and complex marketplace as the internet, payment solutions must not only fit in, but also must provide *value* to users and publishers alike. Systems such as Harpocrates and the extensions presented here explore the concrete steps that we need to take in order to re-imagine the Web payment scheme in a way that better benefits both users and publishers.

Working on this project was very thought-provoking, and led us to see the Web in a whole new light. In order to reason about the ecosystem at large and how a change to it could feasibly be made, we were forced to go "down the rabbit hole" in order to understand how ads are actually served in practice. In doing so, we gained an appreciation for the technology at use in making the internet as we know it free and accessible, as well as a more complete understanding of an extremely important industry. We also we got a better sense of just how dire the situation is for publishers. For example, our research contextualized why so many internet outlets are turning to "click bait" and other tactics in order to boost traffic. Ultimately, all of the information that we've gathered and learned here points to the fact that there are huge opportunities for innovation in both the research and commercial worlds surrounding internet micropayments, and we hope to see (and be a part of) substantial progress in solving this problem in the near future.

References

[1] Adchoices. `http://youradchoices.com/`.

[2] Alexa top sites. `http://www.alexa.com/topsites`.

[3] Atri. `https://atri.me/`.

[4] Autotip. https://autotip.io/.

[5] Blendle. https://blendle.com.

[6] Brave. https://brave.com/.

[7] Flattr plus. https://flattrplus.com/.

[8] Google adsense. https://www.google.com/adsense/.

[9] Google contributor. https://contributor.google.com/.

[10] Satoshipay. https://satoshipay.io/.

[11] Sourcepoint. https://www.sourcepoint.com/.

[12] The cost of ad blocking: Pagefair and adobe 2015 ad blocking report. https://downloads.pagefair.com/wp-content/uploads/2016/05/2015_report-the_cost_of_ad_blocking.pdf, 2015.

[13] Google, microsoft, and amazon are paying adblock plus huge fees to get their ads unblocked. http://www.businessinsider.com/google-microsoft-amazon-taboola-pay-adblock-plus-to-stop-blocking-their-ads-2015-2, 2015.

[14] 37% of mobile users are blocking ads. https://www.globalwebindex.net/blog/37-of-mobile-users-are-blocking-ads/, 2016.

[15] abp. https://adblockplus.org/, 2016.

[16] Adblock plus' revenue model was just ruled illegal by a german court. http://www.businessinsider.com/r-german-court-hands-springer-partial-victory-in-ad-blocking-case-2016-6, 2016.

[17] Adblocking goes mobile. https://pagefair.com/downloads/2016/05/Adblocking-Goes-Mobile.pdf, 2016.

[18] Hubspot adblock plus research study. https://research.hubspot.com/charts/people-use-ad-blockers-because-ads-are-annoying, 2016.

[19] Iab ad blocking report: Who blocks ads, why, and how to win them back. https://www.iab.com/insights/ad-blocking-blocks-ads-win-back/, 2016.

[20] Iab internet advertising revenue report: 2016 full year results. https://www.iab.com/wp-content/uploads/2016/04/IAB_Internet_Advertising_Revenue_Report_FY_2016.pdf, 2017.

[21] ABRAZHEVICH, D. Classification and characteristics of electronic payment systems. In *International Conference on Electronic Commerce and Web Technologies* (2001), Springer, pp. 81–90.

[22] ANDERSON, R., MANIFAVAS, C., AND SUTHERLAND, C. Netcarda practical electronic-cash system. In *Security Protocols* (1997), Springer, pp. 49–57.

[23] BARFORD, P., CANADI, I., KRUSHEVSKAJA, D., MA, Q., AND MUTHUKRISHNAN, S. Adscape: Harvesting and analyzing online display ads. In *Proceedings of the 23rd international conference on World wide web* (2014), ACM, pp. 597–608.

[24] CHAUM, D. Blind signatures for untraceable payments. In *Advances in cryptology* (1983), Springer, pp. 199–203.

[25] CHAUM, D., FIAT, A., AND NAOR, M. Untraceable electronic cash. In *Proceedings on Advances in cryptology* (1990), Springer-Verlag New York, Inc., pp. 319–327.

[26] COOK, J. E., AND ATTARI, S. Z. Paying for what was free: Lessons from the new york times paywall. *Cyberpsychology, Behavior, and Social Networking 15*, 12 (2012), 682–687.

[27] DUKACH, S. Snpp: A simple network payment protocol. In *Computer Security Applications Conference, 1992. Proceedings., Eighth Annual* (1992), IEEE, pp. 173–179.

[28] HAUSER, R., STEINER, M., AND WAIDNER, M. *Micro-payments based on iKP*. IBM TJ Watson Research Center, 1996.

[29] HOHENBERGER, S., MYERS, S., PASS, R., ET AL. Anonize: A large-scale anonymous survey system. In *Security and Privacy (SP), 2014 IEEE Symposium on* (2014), IEEE, pp. 375–389.

[30] KAWAJA, T. Display LUMAscape. http://www.lumapartners.com/lumascapes/display-ad-tech-lumascape/, 2014.

[31] MALLOY, M., MCNAMARA, M., CAHN, A., AND BARFORD, P. Ad blockers: Global prevalence and impact. In *Proceedings of the 2016 ACM on Internet Measurement Conference* (2016), ACM, pp. 119–125.

[32] MANASSE, M. S. millicent(electronic microcommerce). *Digital Equipment Corp* (1995).

[33] PÁRHONYI, R., NIEUWENHUIS, L., AND PRAS, A. The fall and rise of micropayment systems. *Handbuch E-Money, E-Payment & M-Payment* (2006), 343–362.

[34] PÁRHONYI, R., NIEUWENHUIS, L. J., AND PRAS, A. Second generation micropayment systems: lessons learned. In *Challenges of Expanding Internet: E-Commerce, E-Business, and E-Government*. Springer, 2005, pp. 345–359.

[35] PETNEL, R. Easylist. https://easylist.to/.

[36] PRICEWATERHOUSECOOPERS. IAB internet advertising revenue report: 2016 first six months results. http://www.iab.com/wp-content/uploads/2016/04/IAB_Internet_Advertising_Revenue_Report_HY_2016___.pdf, 2016.

[37] PUJOL, E., HOHLFELD, O., AND FELDMANN, A. Annoyed users: Ads and ad-block usage in the wild. In *Proceedings of the 2015 ACM Conference on Internet Measurement Conference* (2015), ACM, pp. 93–106.

[38] RIVEST, R. Electronic lottery tickets as micropayments. In *Financial Cryptography* (1997), Springer, pp. 307–314.

[39] RIVEST, R., AND SHAMIR, A. Payword and micromint: Two simple micropayment schemes. In *Security protocols* (1997), Springer, pp. 69–87.

[40] SCHOENMAKERS, B. Security aspects of the ecashtm payment system.

[41] STOREY, G., REISMAN, D., MAYER, J., AND NARAYANAN, A. The future of ad blocking: An analytical framework and new techniques.

[42] VYKHODTSEV, A. Page load time. Chrome Web Store.

[43] WHEELER, D. Transactions using bets. In *Security Protocols* (1997), Springer, pp. 89–92.