Web Identity Translator

Behavioral Advertising and Identity Privacy with WIT

Fotios Papaodyssefs[†], Costas Iordanou[‡,*], Jeremy Blackburn[*],
Nikolaos Laoutaris[*], Konstantina Papagiannaki[*]

[†]KTH Royal Institute of Technology, Stockholm, fotiosp@kth.se
[‡]Universidad Carlos III de Madrid, Madrid, kostas.iordanou@telefonica.com
[*]Telefonica Research, Barcelona, {jeremyb, nikos, dina}@tid.es

ABSTRACT

Online Behavioral Advertising (OBA) is an important revenue source for online publishers and content providers. However, the extensive user tracking required to enable OBA raises valid privacy concerns. Existing and proposed solutions either block all tracking, therefore breaking OBA entirely, or require significant changes on the current advertising infrastructure, making adoption hard. We propose Web Identity Translator (WIT), a new privacy service running as a proxy or middlebox. WIT stops the original tracking cookies from being set on the browser of users and instead substitutes them by private cookies it controls. Manipulating the mapping between tracking and private cookies WIT maintains permits transparent OBA to continue while simultaneously protecting the identity of users from attacks based on behavioral analysis of browsing patterns.

Categories and Subject Descriptors

K.4.1 [**Computers and Society**]: Public Policy Issues—*Privacy*; H.3.3 [**Information Storage And Retrieval**]: Systems and Software—*User profiles and alert services*

General Terms

Algorithms; Security

1. INTRODUCTION

The massive growth of the web has been funded almost entirely via advertisements. Web ads have proven superior to traditional advertisements for several reasons, the most prominent being the ability to show personally relevant ads. To serve the most relevant ads, web advertisement agencies rely on mechanisms to *uniquely identify* and *track* user behavior over time. Known as *trackers*, these systems are able to identify a user via a variety of methods and over time build up enough information to show user-targeted ads.

While trackers have enabled the free-to-use model of the web, they also raises privacy concerns. These concerns have led to the creation of client side applications that block trackers and ads, for example AdBlock. While AdBlock has been quite successful in mitigating users' exposure to trackers, *by definition* it prevents the display of ads, and thus hurts web services' revenue streams. A tragedy of the commons around privacy can thus become one of the main threats to the web's sustainability [6].

In this paper, we propose a new *Web Identity Translation* (WIT) service to balance the needs of users for privacy and the needs of advertisers for information to drive OBA. WIT is an active service running on a proxy between users and web-sites that host tracking cookies and OBA code. Unlike proposed solutions [2, 11] WIT is transparent to trackers and does not require any change in the infrastructure of the ad ecosystem. Like Network Address Translation (NAT), it introduces a mapping between *private* and *public* 3rd party tracking cookies. When a user's browsing habits start making him uniquely identifiable, WIT intervenes via the private-to-public cookie mappings using one of several policies aimed at restoring user anonymity within the context of the OBA ecosystem. We evaluate 1) To what degree does browsing history uniquely identify users? 2) To what extent can we intervene to reduce identifiability? 3) Does this intervention allow advertisers to infer user interests? With experiments performed on two large datasets, we show that WIT can effectively protect 70% of identifiable users while only intervening on 10% of their requests. When WIT intervenes in 20%

HotNets '15 November 16–17 2015, Philadelphia, PA USA

DOI: http://dx.doi.org/10.1145/2834050.2834105

of requests, it protects about 90% of identifiable users.

2. PROBLEM STATEMENT

2.1 Background on Online Advertising

Advertising Ecosystem: There are four main entities in the advertisement ecosystem that we consider crucial for WIT. The *user*, the *publisher*, the *advertiser* and the *advertising network*. The user visits web pages provided by the publisher, who in turn is paid by advertisers to display ads. The *advertising network* is the entity that coordinates the whole process. The publishers, advertisers, and advertising networks have a common financial interest to increase the probability that a user actually clicks an ad (click through rate) and makes a purchase. This is where OBA comes into play, having been shown to significantly increase click through rate [1].

User Tracking: For OBA to work, advertising networks need to track the activity of users across the web. This is achieved via tracking beacons placed on publishers' websites. The tracking beacons are usually small images embedded in the web page that trigger a request to the tracker's server. When a new user visits a website that is tracked by an ad network, his browser downloads the image, and the server in turn sets a cookie that is associated to this user. Subsequent requests to any website where the ad network has access will return the same cookie, therefore tracking the user across the web. In this work, we only deal with the traditional web cookie as a tracking mechanism, however WIT is not fundamentally tied to cookies and can operate on any form of unique online identifier.

2.2 The Behavioral Re-identification Problem

The introduction of private cookies and their mapping to external public ones in WIT can protect users from several types of privacy leakage. We now focus on the *Behavioral Re-identification Problem*, described next.

Personally identifiable information (PII) can be leaked through a variety of means, *e.g.*, passing email addresses or real names as HTTP arguments after filling in web forms [6]. When such PII leakage occurs in a page hosting tracking cookies, trackers can associate the real-world identity with online presence at any web sites the cookie is observed. This is a very serious threat to one's privacy. Notice here that online behavioral targeting does not really need the association between the PII and the cookie; it just requires a constant identifier (*e.g.*, a cookie) to be able to say that user X seen at foo.com is the same one now visiting bar.com. Such a constant identifier is effectively anonymous if not connected to PII.

This means that as long as users make sure that PII does not leak, then OBA can be carried out while the users remains anonymous. Eliminating all PII leakage however is quite difficult and in many cases impossible to achieve without breaking much of the web's usability. An alternative to blocking all PII is to just monitor for leakage, and when it occurs clear all cookies to prevent matching the user's PII with past and future browsing. For this to work however, one has to protect against search and re-identification of individuals with leaked PII. The tracker already has a sample of the behavioral pattern of this *named* user. Even if the user clears all his cookies, as soon as he accepts a new cookie from a site that the tracker operates on he risks re-identification and re-association to his real-world identity through a simple comparison of his sampled behavior as a known user and his newly accumulating behavior under the new identifier (cookie). In fact, re-identifying a user by comparing profiles associated with different cookies is a corner stone of the burgeoning cross device identification industry [4].

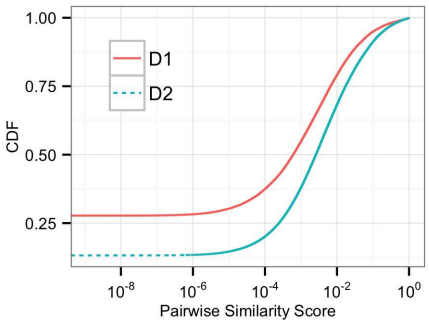

Figure 1: CDF of pairwise similarity scores of user browsing histories for our datasets.

Thus, the overarching threat to identity privacy is the linking of online behavioral profiles to an individual user. As we explain in the next section, WIT addresses this threat model by ensuring that profiles trackers build up are not uniquely identifiable, yet still contain useful behavioral information.

2.3 The Severity of Unique Identification

At the core of the re-identification problem is the surprising "uniqueness" of browsing profiles using *e.g.*, frequency histograms of visits to web-sites hosting tracking cookies. To demonstrate the severity of the problem, Figure 1 plots the distribution of pairwise similarity scores for the browsing behavior of all users across each of our datasets (see Section 4 for dataset details).

From the figure we make several observations. First, both datasets show a median pairwise similarity score less than 0.01. *I.e.*, when comparing the similarity in browsing history of any two randomly selected users, there is a 50% chance it will be less than 0.01. Thus, we conclude that *users are extremely unique when compared to any other given user*, also recently observed

by [9]. However, there is a long tail of user pairs that have an order of magnitude more than the medium similarity. As explained in the next section, WIT can exploit this phenomenon of two users being much more similar than they are to other users to "hide" a user among others.

3. WEB IDENTITY TRANSLATOR

As we just saw, users have distinctively unique browsing patterns and this can be used to re-identify them even after they have flushed old tracking cookies and substituted them with new ones. Still, they are sufficiently similar to permit relatively small alterations to their true browsing pattern to conceal them among other users. WIT leverages this observation by intervening on a user's browsing behavior visible to a tracker, preventing him from becoming uniquely identifiable.

3.1 Terminology

Browsing history: A user's browsing history is a vector of domain and visit frequency tuples over a given time period. This need not be all the domains that a user has visited but those where a certain tracker that is attempting the re-identification attack is present with tracking cookies.

Signature: A user's signature is browsing history collected over some time interval in the past. During this interval the attacker has constructed the behavioral patterns of the user. An attacker can search for this pattern in arbitrary page visiting logs and re-identify the user even if the latter has changed cookies in between. This can also be used to match users between devices.

Similarity: Similarity is a measure of the "closeness" of two users. In this paper, we use $\theta(h_1, h_2)$ to represent the similarity between two browsing histories. We give a detailed explanation of the measure we use in Section 4.

Similarity rank: In this paper we use the ranking of similarity scores between histories as a measure of re-identifiability. When we say $rank(\theta(h_1, h_2)) = \kappa$ we mean that h_2 is the κth most similar history to h_1 (with $\kappa = 1$ meaning it is the closest).

3.2 Architecture

WIT is analogous to NAT in networking. Figure 2 shows a high level view of the WIT architecture. Much like NAT projects private IPs addresses into the public IP address space and manages the mapping between them, WIT manages mappings of private and public *cookies*. The cookie setting/reading mechanism works as usual. A web-site sets a standard (public) cookie on a client visiting for the first time but this cookie is intercepted by WIT and thus never reaches the browser. Instead WIT creates a corresponding private cookie, associates it with the intercepted public one, and sets it on the user's browser. Inversely, when the user returns

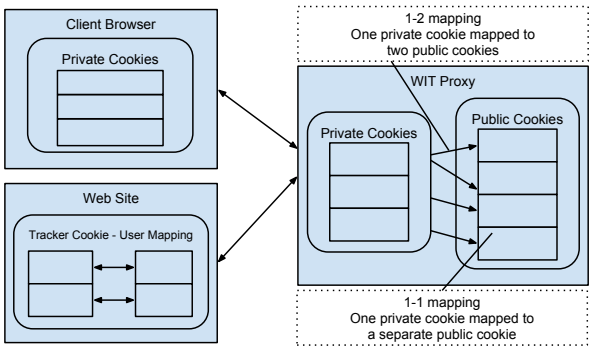

Figure 2: High level overview of the Web Identity Translator architecture.

to a web-site that hosts the same tracker, WIT realizes it and returns the same public cookie that the tracker thought it set on the user's browser.

WIT can be implemented in any middlebox or proxy that sees HTTP traffic of users. This includes CDN nodes or caches, acceleration proxies for wired or wireless networks, or VPN proxies. The latter are increasingly popular and can permit WIT to operate seamlessly even if end to end HTTP encryption is enabled as is the trend lately [8].

When WIT receives a request from a new user, it places him in the *quarantine* phase, where all tracking is blocked until a browsing signature is collected. This signature is stored as a history vector and will be used to make intervention decisions later on, based on the ranking of WIT's private cookies with respect to this signature. How long a user should be kept in quarantine needs to be further evaluated, but for this paper, with respect to identifying user patterns, a week's worth of browsing has proven sufficient and it is likely that even less is required.

Once there is sufficient data, the user enters the *triage* phase. This is an active monitoring phase, where we examine the effect each request has on potentially assisting the de-anonymization of a user. In order to avoid this, we monitor users during the triage phase, and intervene *only* if their current history ranks among the top κ histories with respect to their signature. During the triage phase, multiple history vectors could be associated with a user. These history vectors correspond to a public cookie, and represent what information has been allowed to pass through to the tracker. The triage phase is responsible for guaranteeing that none of these history vectors can be linked with high probability to the signature vector of the user that is associated with them. The general trigger depends on the exact policy used (described in the next section) but revolves around the ranking of users.

3.3 Intervention Algorithms and Policies

The main idea behind WIT is to protect users against

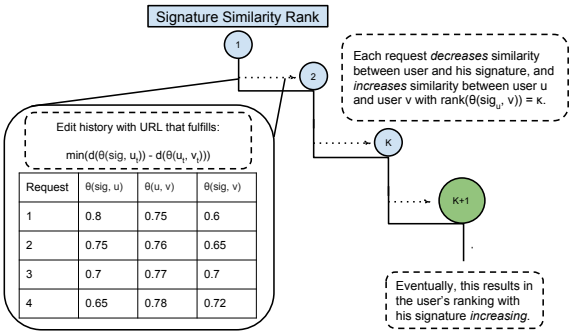

Figure 3: High level sketch of WIT's History Padding algorithm.

the attack presented in Section 2.2 and described in [7]. This attack attempts to rank users based on the similarities of their signatures so that they can match histories to actual users. To mitigate this, WIT maintains the rankings of all users and attempts to stop their histories from ranking very high with their signatures, thus being identifiable. By reducing the rank of a user below a threshold κ, we reduce the probability of an attacker successfully identifying him. The appropriate threshold is a subject for future evaluation.

In this paper we focus on two specific intervention policies: 1) cookie drop and 2) history padding.

Cookie drop: The cookie drop intervention algorithm simply refuses to return the public cookie associated with the user. In essence, this results in a particular request being associated with a "new" user instead of the profile of the user himself. The decision to drop a cookie is made by calculating the ranking of the current history as it has been allowed to leak against the signature WIT collected during the quarantine phase. If a request causes the rank to drop, the request is allowed to pass; if the rank increases, the tracking cookies associated with this request are dropped. A single request rarely affects the ranking, and when it does not $\delta(\theta(sig, u'), \theta(sig, u))$ is calculated. If it is negative, therefore driving the user away from his signature, the cookie is allowed to pass while in the opposite case it is dropped.

History padding: The history padding algorithm is more complex, yet provides superior performance. The advantage with history padding is that WIT can take an *active* approach in obfuscating users, instead of the passive approach of cookie dropped. As requests arrive, if the current history of a user is determined to be ranked below a κ threshold, WIT searches through the history of the κth ranked user for a request that, when added to u's browsing history minimizes the function:

$$\delta(\theta(sig, u'), \theta(sig, u)) - \delta(\theta(u', v), \theta(u, v))$$

, where u' is u's history with the addition of the candidate URL from v's history and δ is the (signed) difference between two similarity scores

Figure 3 sketches a high level view of the history padding algorithm. The core idea behind history padding is to choose the optimal intervention that *increases* the distance of a user from his signature while simultaneously *decreasing* the distance between the user and κth ranked user from the signature. Another way of putting it is that WIT *decreases the similarity* of the user and his signature and at the same time *increases the similarity* of the user and the user that has the κth ranked similarity score with the signature. Eventually, this push and pull mechanism results in the user's re-identification rank increasing, making him exceedingly hard to manually re-identify.

4. EVALUATION

In this section we present an evaluation of WIT. We begin by presenting our datasets, evaluation metrics, and experimental setup before moving on to results.

4.1 Datasets

D_1 from [5] is composed of browsing history donated by Firefox users. It includes all the pages that the users visited, both HTTP and HTTPS. The dataset is made available in obfuscated form where user ids and the URLs they visited are hashed. We make use of one month of data consisting of 6 million total requests from 529 users.

D_2 is derived from traffic logs of a vpn proxy for mobile users operated by a large telecom provider in Europe. This dataset represents mobile traffic over the proxy but does not include HTTPS traffic. Users are identified with an internal proxy identifier that is not linked to any personal information. In total we use approximately 3 weeks of data, consisting of 2.5 million requests from 730 users.

4.2 Metrics

The most commonly used metrics in user similarity studies are Jaccard index and cosine similarity. The problem with Jaccard index is that it operates solely on set membership and therefore cannot capture properties like frequency/popularity. For this reason, we use the *Vector Space Model* as a representation of users' histories and cosine similarity with *tf-idf* weights to calculate similarity. Tf-idf is widely used in information retrieval and it manages to reduce the impact of very popular terms. For two users u and v, their similarity is calculated as follows:

$$\theta(u, v) = \frac{\sum_{i=1}^{n} u_i \times v_i}{\sqrt{\sum_{i=1}^{n} u_i^2} \times \sqrt{\sum_{i=1}^{n} v_i^2}}$$

The values of the attributes u_i, v_i are the corresponding tf-idf values. For example:

$$u_i = tf_i \times (1 + ln(\frac{N}{df_i + 1}))$$

Where tf_i is the number of times user u has visited webpage i, N is the total number of users and df_i is the number of users who have webpage i in their histories.

If two users visit the same set of websites the same number of times in a given time period, their cosine similarity will be 1, and if there is *no* overlap in their browsing history it will be 0. It is important to note that in this work we operate on the domain level, meaning that each attribute corresponds to visits to domain and not a full URL. We do this for two reasons: 1) using only domains highlights similarity and repeated browsing patterns, and 2) from a semantic point of view, it is often times more relevant that a user has visited a given site instead of any particular page on that site. That said, in the future we will examine methods of improving the granularity of the browsing history, *e.g.*, using semantic value of subdomains or content on pages.

4.3 Experimental Setup

We simulate the function of WIT by replaying requests from our two datasets. We split each dataset in two parts, one which will be used for the quarantine phase and one which will be used for the triage phase when the system starts intervening on users' histories. The simulation starts by reading the quarantine part of the logs and storing the corresponding vectors. Once the quarantine is over, the remainder of the dataset is read line by line, simulating the requests as they would arrive in a proxy. For this work, we evaluate the *cookie drop* and *history injection* policies, therefore a one to one mapping is always maintained between public and private cookies. Our experiments are designed to determine if WIT can effectively push users away from their signatures and how much intervention was required.

4.4 Results

Figure 4 plots the distribution of $rank(\theta(sig_u, u))$ as a function of the number of request having passed through WIT during log playback. The x-axis represents the cumulative number of requests handled by WIT and the y-axis is the rankings of the public history vectors with respect to their respective signatures used for re-identification. Each panel represents a different data set/policy combination.

First, we observe that with the control policy, which shows how users naturally rank with respect to their signature without intervention, essentially all users rank lower than 10 for both datasets (median rank of 1 for D_1 and 2 for D_2). This means that if an attacker uses an already known signature of a user to search in a new dataset he will be among the 10 closest matches. Although not plotted, the cookie drop policy was able to improve upon the control with a median rank of 2 in D_1 and 4 in D_2.

Looking at the History Padding results, we see substantial improvement. When allowing WIT to make as many interventions as necessary (panel "HPad"), the median user rank reaches (and stays above) 10 within 50,000 simulated requests. Further, even the 25th percentile is relatively close to 10. Notice here, that for some users, it is close to impossible to hide them among others. This happens if their accumulated profile during the training period is distinctively different from anybody else. In that case even if WIT intervenes in all subsequent requests they can still remain close in rank to the original signature used by an attacker to re-identify them.

Next, we place restrictions on the maximum number of allowed interventions per user (panels "HPad 10%" and "HPad 15%"). When WIT intervenes 10% of the time, we see median ranks of 6 and 8 for D_1 and D_2, respectively. When WIT intervenes on 15% of requests, we see medians of 8 and 10. Placing such constraints of course limits the ability of WIT to hide users (the 25th percentile is still 1 for D_1 and slightly increases to 2 for D_2) but WIT is still quite successful in protecting the majority of users from re-identification.

5. DISCUSSION AND CONCLUSION

Impact on Advertisers: WIT is designed to balance user privacy and utility to advertisers. This means WIT's interventions should not stop the effective targeting of users with relevant ads. We have already shown that with a rather limited number of interventions (10-15%) WIT protects the majority of users which is a sign that damage to ad profiles is small: advertisers still get to see most of a users' original requests. Next we show preliminary validation that WIT does not significantly impact advertisers. We manually played back both the original and WIT intervened histories for three users that WIT appeared to have significant impact on. We then checked the interest tags of the histories via the BlueKai registry, which allows users to see their tags collected by the BlueKai tracking service. Although not a perfect measure, if a user's original and modified browsing histories have similar tags, it indicates a minimal effect on advertising.

Table 1 shows differences in interest tags returned from raw and intervened user histories for three highlighted users. We note that with the exception of one user, WIT intervened histories returned all the tags from the raw histories. While there were some *additional* tags returned, they are related to the original history's tags. As future work, we are developing a large-scale, automated system to evaluate WIT's impact on interest tags.

Deployment optimizations: It might seem that WIT faces scalability challenges due to the computation of cosine similarities and ranks between users and signatures. Although we leave a full implementation to

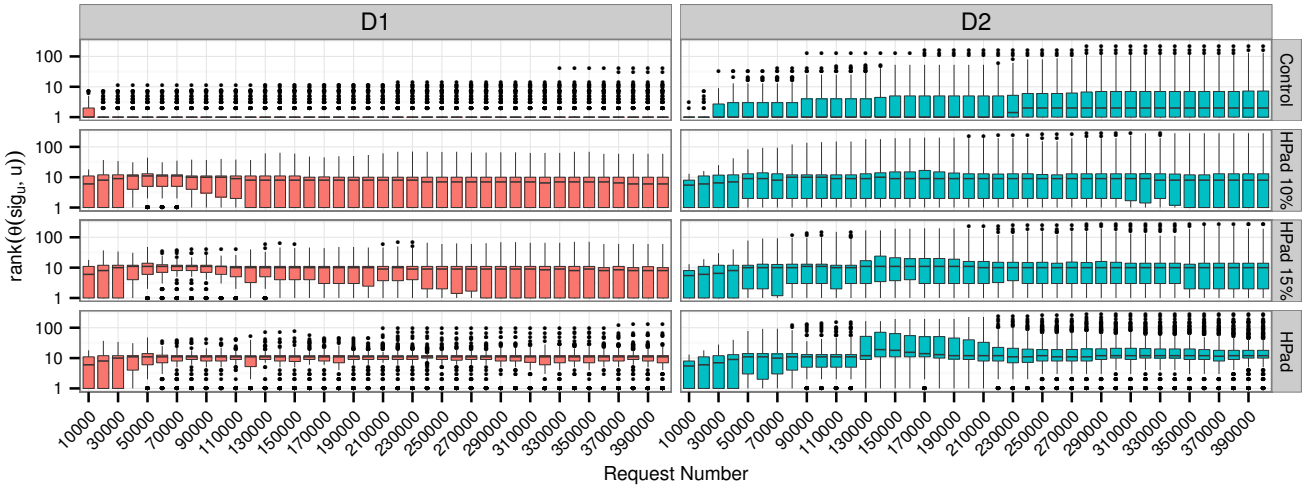

Figure 4: The evolution of re-identification rank over time.

Interest Tag	User 1	User 2	User 3
Broadband		✓	
Cell Phones & Plans	○		
Country	✓	✓	✓
Demographic	✓		○
ISPs		○	
Language Speakers		✓	○
Mobile Phones	✓		
Online Radio		✓	
Technology		✓	

Table 1: Interest tags returned before and after WIT intervenes for three users. ✓ means both histories had the tag and ○ means only the intervened history had it. NB: Country and Language Speakers have been anonymized.

future work, we will sketch some ideas of why it can be scalable in practice. The key to scalability is that we only need to "hide" a given user within a small subset of users, which means that we do not need to compute distances to all other users. If the user is hidden within this small subset (one for which computation is feasible), then the user is effectively hidden in any larger group of users (*e.g.*, those of a telco), including *all* users. While the attacker is assumed to have access to the entire population, since they have no idea of the subset WIT intervenes on, they are left with what amounts to a $\frac{1}{n}$ chance of randomly re-identifying the correct user.

We can further improve the efficiency of WIT in a variety of ways. First, we can *subsample* requests (*i.e.*, WIT only intervenes every $S > 1$ requests). Second, we can do *lazy updating* of the ranking computations (*i.e.*, rankings will only be updated every t time intervals or every S requests). Next, we can cluster users (*i.e.*, computing rankings over groups instead of individual users) or make use of smarter data structures and algorithms for streaming data, *e.g.*, [10].

Balancing advertising and privacy: While WIT as presented performs quite well at balancing privacy and advertising needs, improvements can be made. For example, the current algorithm is greedy and biased towards preserving privacy. To address this, instead of intervening with the URL that most improves privacy, WIT could be augmented to make use of semantic information about the URLs that are requested and choose to intervene with one that maintains the same set of behavioral profiling tags attached to the user. By changing the weight given to improvement of identification privacy vs. accuracy of behavioral tags, WIT can provide an easy to tune privacy preserving mechanism while still providing advertisers with relevant user knowledge.

Additional policies: We plan to explore a policy where an offending request will trigger creating a new **pseudonym** [3] for the user. WIT then balances requests using multiple pseudonyms to ensure privacy is preserved. Another policy **cookie swap**, will transmit the cookie of a *different* but *relatively similar* user, thus obfuscating the original user's behavior. Cookie swap is basically the inverse of history padding: instead of *taking* a request from another user's history, WIT *inserts* a request into another user's history. A combination of policies could maintain advertising utility while offering strong privacy guarantees. We leave the evaluation of these policies to future work.

Acknowledgements

The research leading to these results has received funding from the European Union's Horizon 2020 innovation action program under grant agreement No 653449-TYPES project and grant agreement 653417-ReCRED project.

6. REFERENCES

[1] H. Beales. The value of behavioral targeting. http://www.networkadvertising.org/pdfs/Beales_NAI_Study.pdf, 2010.

[2] S. Guha, B. Cheng, and P. Francis. Privad: Practical privacy in online advertising. In *Proceedings of the 8th USENIX Conference on Networked Systems Design and Implementation*, NSDI '11, 2011.

[3] S. Han, V. Liu, Q. Pu, S. Peter, T. Anderson, A. Krishnamurthy, and D. Wetherall. Expressive privacy control with pseudonyms. *SIGCOMM Comput. Commun. Rev.*, 43(4):291–302, Aug. 2013.

[4] R. Joe. The cross-device question: Krux. http://adexchanger.com/data-exchanges/the-cross-device-question-krux/, 2014.

[5] R. Kawase, G. Papadakis, E. Herder, and W. Nejdl. Beyond the usual suspects: context-aware revisitation support. In *HT'11, Proceedings of the 22nd ACM Conference on Hypertext and Hypermedia, Eindhoven, The Netherlands, June 6-9, 2011*, pages 27–36, 2011.

[6] N. Laoutaris. Cows, privacy, and tragedy of the commons on the web. http://www.thedigitalpost.eu/2015/channel-data/cows-privacy-tragedy-commons-web, 2015.

[7] A. Narayanan and V. Shmatikov. Robust de-anonymization of large sparse datasets. In *Security and Privacy, 2008. SP 2008. IEEE Symposium on*, pages 111–125. IEEE, 2008.

[8] D. Naylor, A. Finamore, I. Leontiadis, Y. Grunenberger, M. Mellia, M. Munafò, K. Papagiannaki, and P. Steenkiste. The cost of the "s" in https. In *Proceedings of the 10th ACM International on Conference on Emerging Networking Experiments and Technologies*, CoNEXT '14, pages 133–140, 2014.

[9] L. Olejnik, C. Castelluccia, and A. Janc. Why Johnny Can't Browse in Peace: On the Uniqueness of Web Browsing History Patterns. In *5th Workshop on Hot Topics in Privacy Enhancing Technologies (HotPETs 2012)*, Vigo, Spain, July 2012.

[10] J. W. Reed, Y. Jiao, T. E. Potok, B. Klump, M. T. Elmore, A. R. Hurson, et al. Tf-icf: A new term weighting scheme for clustering dynamic data streams. In *IEEE 5th International Conference on Machine Learning and Applications*, ICMLA '06, pages 258–263, 2006.

[11] V. Toubiana, A. Narayanan, D. Boneh, H. Nissenbaum, and S. Barocas. Adnostic: Privacy preserving targeted advertising. In *Proceedings of the 17th Annual Network and Distributed System Symposium*, NDSS '10, 2010.