Giorgio Di Tizio* and Fabio Massacci

A Calculus of Tracking: Theory and Practice

Abstract: Online tracking techniques, the interactions among trackers, and the economic and social impact of these procedures in the advertising ecosystem have received increasing attention in the last years. This work proposes a novel formal model that describes the foundations on which the visible process of data sharing behaves in terms of the network configurations of the Internet (included CDNs, shared cookies, etc.). From our model, we define relations that can be used to evaluate the impact of different privacy mitigations and determine if websites should comply with privacy regulations. We show that the calculus, based on a fragment of intuitionistic logic, is tractable and constructive: any formal derivation in the model corresponds to an actual tracking practice that can be implemented given the current configuration of the Internet. We apply our model on a dataset obtained from OpenWPM to evaluate the effectiveness of tracking mitigations up to Alexa Top 100.

Keywords: online tracking, ad-blocker, formal model

DOI Editor to enter DOI
Received ..; revised ..; accepted ...

1 Introduction

Online tracking of users for targeted advertising is the reality of today's Internet, and the extent of such tracking has been the subject of an intense research activity [1]. Research studies span from facts finding studies (e.g. [2, 3]) to technical analysis both for classical techniques (e.g. based on cookies syncing [4]) and novel techniques (e.g. based on browser fingerprinting [5]). Economic analysis are also not uncommon (e.g. [6]). A number of mitigation tools also emerged to (partly) block trackers (e.g. Ghostery, Disconnect, Adblock Plus, etc.) and researchers have also investigated whether they are effective (e.g. [7, 8]) and their side-effects (e.g. [9, 10]).

These robust research activities, which we sample in Tab. 1 and Tab. 2, generated a number of open databases that provide internet snapshots[1] and that can be used to experiment with tracking behavior, the effectiveness of tracking mitigations as well as derive metrics for a tracker market share or trackers concentrations [11], at least for what is measurable from the Internet.[2]

The major privacy concern of the users is with whom and for which purposes their personal information is shared and not by which technology [13, 14]. In other words, users are not worried that an entity collects data when interacted with it but that these data are shared with other entities. An interesting question, so far unanswered, is how can we provide a third party independent verification of empirical tracking claims. A study might claim that tool A is more effective than tool B at mitigating trackers but there is hardly any way for a third-party to check why and how unless one re-runs the entire study and trace all results. Even the claim that a tracker *burstnet.com* can potentially know whoever visited website *amazon.com* is hard to check unless one re-run the entire study.

We answer such question by a formally grounded mechanism: *a calculus of tracking for the internet*. We associate a formal relation between various way to exchange data (access and inclusion of web pages, cookie syncing, redirects, etc.) as they are measurable from the internet (and available in open datasets such as Web Census) and identify formal rules that capture how internet visits can be tracked.

We can formally prove that a tracker can potentially know that a user visited a website. By using a fragment of intuitionistic logic we can extract from the proof the actual web pages configuration that makes such tracking possible. The formal model allows one to determine if a website should comply with privacy laws (e.g. COPPA) or to compare different mitigations and conclude whether a mitigation is strictly better than another or at least quantitatively better along a Pareto frontier. For this approach to be a useful link between theory and practice, such calculus must be tractable and

***Corresponding Author: Giorgio Di Tizio:** University of Trento, E-mail: giorgio.ditizio@unitn.it
Fabio Massacci: University of Trento, Vrije Universiteit Amsterdam, Email: fabio.massacci@unitn.it

[1] E.g. Web Census: http://webtransparency.cs.princeton.edu/webcensus

[2] Obviously, data exchange agreements between owners of seemingly different trackers do affect conclusions based on the internet. Detecting such agreements is hard given the present information asymmetry [12].

the analysis can be performed on the fraction of the Internet visited by a user as available from open datasets (e.g. OpenWPM) and we do so up to Alexa Top 100.

The overview of the key works in this area (§2) summarized in Tab. 1 shows that the majority of the research focused on large scale analysis and technological analysis, while no attempt has been done on formally describing the sharing procedures. We fill this gap with the following contributions:

- we present the *first* formal model (predicates and rules) that describes the passage of tracking information across websites that can be externally measurable (§3 and §4);
- we prove that inferences are tractable and that one can reconstruct the configuration responsible for a concrete tracking practice from the proof (§5);
- we formalize some of the interesting tracking relations that can be captured by our system (§6);
- we extended the model to consider the uncertainty of the Internet interactions (§7);
- we discussed scalability issues and challenges (§8) and we instantiated our model in a state of the art theorem prover to determine tracking practices and websites that should comply with COPPA (§8.2);
- we compare the effectiveness of different mitigations (`Ghostery`, `Disconnect`, `Adblock Plus`, and `Privacy Badger`) on the Top 5, Top 10, Top 50 and Top 100 *visited* domains (§9).

We conclude by discussing the added value and the limitations of our approach (§10) and future work (§11).

Goals: we provide a framework that generates a third-party independent verification of tracking practices for *individual* cases, i.e. for single users that browse a limited number of websites. Large scale studies over millions of domains are unrealistic and inappropriate to help users in determining the best countermeasure to enhance their privacy or determine which websites should comply with COPPA. Furthermore, these studies lack transparency and do not provide concrete evidence of the effectiveness of the mitigations analyzed (as well as provide in some cases contrasting results). Our framework fills the gap by providing *an explanation*, in the form of a formal proof, that can help users to evaluate the effectiveness of different adds-on or provide a proof that shows if a website should comply with COPPA.

Non-goals: we do not consider methods that rely on back-office data exchange for user identification. For example, collecting browser fingerprinting and using it in the back-office data sharing. As such we could assert that certain websites perform fingerprinting but, in absence of a publicly known relation between these websites on how fingerprints (or other personal information) are shared, this would be a meager knowledge. Furthermore, the application of the framework is not intended for large scale analysis of the whole Internet, where formal reasoning hardly scale.

2 Analysis of Tracking Practices

Over the last years, researchers have identified different tracking techniques on the Internet. To make the paper self-contained we present an overview of the techniques and refer the reader to Tab. 8 in the Appendix A.1 for additional information. Tab. 2 reports some research questions addressed by the state-of-the-art. Most papers examined the effectiveness of tracker blocking tools, while others focused on trackers' pervasiveness and the techniques used to track Internet users. We considered works about Online Advertising Ecosystem, privacy policies, and formal modeling of the Internet.

A Summary of User Identification

HTTP Cookies are IDs associated with a user and are set by websites through JavaScript codes or HTTP responses. Cookies are automatically attached by the browser to all subsequent requests to the websites. The major difference compared to browser fingerprinting is that the ID is stored locally on the user's machine [7, 27].

Browser Fingerprinting is used by websites to collect information from the browser to build an unique fingerprint [28]. For example, to personalize the content, a website can request device-specific information like `user-agent`, `HTTP headers`, `plugins`, `fonts`, `screen resolution`, `OS`, `canvas` and `AudioContext` [5, 29, 30] via HTTP headers or JavaScript codes [22]. These attributes can be used to generate a unique fingerprint for tracking purposes. Other approaches exploit O.S. and hardware properties to generate *device fingerprints* that allow cross-browser tracking [31, 32].

Other Browser Storage, for example `HTML5 localStorage`, `Flash LSOs`, and `HTTP headers` (e.g. `ETag`) [33, 34], are used by websites to store IDs and track users even if HTTP cookies are deleted.

Other tracking techniques exploit browsing history [35] and caching process of DNS records [36].

Table 1. Research Topics Addressed by the State of the Art

Research	Research Topic	[15]	[2]	[6]	[16]	[17]	[3]	[18]	[7]	[19]	[4]	[20]	[21]	[8]	[22]	[23]	[24]	[25]	[26]	Our work
Fact finding	Analysis of the tracking ecosystem				✓	✓	✓	✓							✓					✓
	Tracking coverage		✓		✓		✓		✓	✓	✓		✓	✓						✓
	Search context exposure to tracking				✓				✓											
	Detection of hidden flow among trackers					✓				✓	✓		✓							
	Detection of privacy regulation violation															✓				
	Detect duty compliance to privacy regul.																			✓
Economic analysis	Cookie syncing incentives	✓											✓							✓
	Revenue with and without cookies	✓	✓	✓									✓							
Technical analysis	Effectiveness of blocking techniques						✓		✓	✓		✓		✓						✓
	Development of a detection mechanism							✓			✓				✓	✓				
	Classification of Trackers							✓	✓			✓								
	Analysis of Tracking techniques									✓		✓			✓					
Logic	Formal expression of privacy policy																✓	✓	✓	
	Formal analysis of tracking																			✓

Data Sharing Across Websites

Cookie Syncing is an increasingly popular technique [4] employed by trackers to share the IDs associated to a user [17, 19, 30]. A common cookie syncing technique is to pass the IDs as parameters in an HTTP request. This procedure allows the websites to map different IDs to a single user and link information from different trackers.

Analysis of the Online Advertising Ecosystem

Ghosh et al. [15] analyzed the leakage of information in the Real-Time Bidding (RTB) protocol and modeled the revenue of advertisers w/ and w/o syncing. Gill et al. [2] employed HTTP traces to model the revenue earned by different trackers, whereas Marotta et al. [6] empirically estimated the value of targeted advertisement depending on the presence or absence of cookies. We aim to address an orthogonal problem, i.e. how can we formally prove that cookies are employed for tracking users, independently from how trackers utilize them.

Iqbal et al. [37] developed a graph-based ML classifier of ads and tracker. The tool builds a graph representation of the HTML structure, the network requests, and the JavaScript in the web page to determine tracking practices from specific features. Gomer et al. [16] analyzed how the search context exposes users to tracking practices using directed network graphs based on referral header information. Bashir et al. [3, 17] detected flows of information between advertisers based on retargeted ads. They constructed an inclusion graph to model the advertising ecosystem, analyzed the graph properties, and simulated the impacts of tracker blocking tools like `Disconnect` and `Adblock Plus`. Kalavri et al. [18] represented traffic logs through 1-mode and 2-mode graphs to highlight the connected communities of the trackers and proposed an automated tracker detection mechanism based on graphs properties. Similarly, in our paper we (implicitly) employ graphs to represent the interactions among websites that are exploited in the tracking ecosystem. However, we also provide a formal description of the trackers' interactions to prove tracking practices and privacy compliance.

Formal Models for IT-Security

Speicher et al. [38] used a model based on AI planning with grounded predicates in the context of the email infrastructure. Simeonovski et al. [39] proposed a model based on property graphs in the context of Internet core services. We instead focus on tracking practices and present a stronger formal foundation compared to the previous works by proposing a formal Gentzen calculus.

Analysis of Children's Online Privacy Protection Act Compliance

Data collection and web tracking are regulated by data protection laws. For example, the General Data Protection Regulation (GDPR) [40] is currently in force in the EU. Still, it is not uncommon to observe violations [41].

When it comes to collect personal information from children additional laws are applied. In the U.S. the Children's Online Privacy Protection Rule (COPPA) [42] imposes requirements for websites that

Table 2. Research Questions From the State of the Art

Papers belong to some research classes: Fact finding (F), Economic analysis (E), Technical analysis (T), and Logic (L).

Paper	Class	Research Questions
[15]	E	What induce publishers to perform cookie sync?
[2]	F,E	What is the revenue given different information?
[6]	E	How much cookies influence publishers' revenues?
[16]	F	How does search context impact users' privacy? What are the tracking ecosystem characteristics?
[17]	F	Can retargeted ads detect exchange of info?
[3]	F,T	Can we capture interactions among publishers and Ads company based on web inclusions? How effective are tracking mitigations?
[18]	F,T	Can we identify trackers via mode graph?
[7]	F,T	How can we classify web tracking behaviors? How effective are tracking mitigations?
[19]	F,T	How trackers are distributed in the Top 1M? How effective are tracking mitigations? Can we automatically collect tracking behaviors?
[4]	F,T	Which are the characteristics of cookie syncing? Which is the impact for the privacy of the users?
[20]	T	Are invisible pixels used extensively by trackers? Can we classify trackers based on invisible pixels? How effective are mitigations vs invisible pixels?
[21]	F,E	What is the impact of cookie syncing and RTB? Which is the price value of user's private data?
[8]	F,T	How effective are adds-on for desktop and mobile? What are challenges of mobile tracking prot.?
[22]	F,T	How to identify fingerprinting using font-probing? How effective are the tools against fingerprinting?
[23]	F,T	How to detect 3rd-party privacy violations?
[24]	L	How can we formally reasoning about norms of transmission of PII?
[25]	L	How can we formalize and check privacy rules?
[26]	L	How can we formalize privacy rules?

collect personally identifiable information (PII)[3] from children under 13 y/o. COPPA requires to post a privacy policy containing the personal information collected, with who and for which goal this data is shared, to get a verifiable parental consent (for example calling a tool-free number), and to allow parents to review the PII collected and revoke the consent. Determine if a website should comply with COPPA is not easy. There have been several violations in the past, for example by Playdom [43] and Youtube [44], with fines of several millions of dollars. Apart from U.S. FTC reports, some studies developed frameworks to analyze Android apps to determine violations [23, 45, 46]. However, there is not yet a tool for parents to determine such violations due to the complex interactions among websites.

Several papers tried to formally express privacy policy (e.g. [25, 26]). In the context of COPPA, Barth et al. [24] proposed a framework to formally describe this policy based on first-order temporal logic. Although similar to our work, [24] is a theoretical model that focuses on the description of privacy policies. In contrast, our model describes tracking behaviors for advertising and generate proof that websites should comply with COPPA based on data from the Internet. For example, parents can determine which websites should comply based on their children's visited websites. The effective compliance can be manually verified by the parents and the proof can prompt further investigations by the FTC.

3 A Formal Model for Tracking

We define a privacy threat when a website can know that a user visited other websites as a result of a process of data sharing. The major concern for a user is not that a website knows this is a recurring user, but that unrelated websites get knowledge of this activity thanks to the exchange of data. We did not model tracking techniques that are not been observed in the wild (e.g. [47]).

3.1 Predicates

Tab. 3 contains predicates that capture network interactions among websites and users (in our analysis we reduced the websites to their PS[4] and PS+1 of the URL[5]) as well as the type of mitigations considered. $\mathit{IncludeContent}(w, w')$ indicates an inclusion of some content of web site w' in website w. The predicate $\mathit{IncludeContent}_{\mathit{cookie}}(w, w')$ describes, in addition, a transmission of cookies collected by w to w'. $\mathit{Redirect}_{\mathit{cookie}}(w, w')$ ($\mathit{Redirect}(w, w')$) indicates a HTTP redirection from the website w to the website w' with (without) a transfer of cookies collected by w.

$\mathit{Block\_request}(w)$ indicates that the connection to the website w is blocked, for example by an add-on. $\mathit{Block\_tp\_cookie}(w)$ indicates that the website w is not allowed to set cookies. These two mitigations protect users using different techniques. If $\mathit{Block\_request}(w)$

[3] E.g. First and last name, home address, SSN, persistent identifiers (e.g. cookies, fingerprinting), etc.

[4] https://publicsuffix.org/

[5] For example, *https://s.ytimg.com* is reduced to *ytimg.com*.

Table 3. Ground Truth Network Interactions and Mitigations

The predicates are obtained from ground truth data (G) and are *not* derived from rules of the model.

$\mathit{IncludeContent}(w, w')$	G	website w includes 3rd-party content from the website w' (e.g. within an `i-frame` tag).
$\mathit{IncludeContent}_{\mathit{cookie}}(w, w')$	G	website w includes 3rd-party content from the website w' sending its cookies.
$\mathit{Redirect}(w, w')$	G	website w redirects visitors to the website w'. w *does not* append cookies in the redirection.
$\mathit{Redirect}_{\mathit{cookie}}(w, w')$	G	website w redirects visitors to the website w'. w appends its cookies in the URL (or payload).
$\mathit{Visit}(w)$	G	intentional access to website w by a user.
$\mathit{Block\_request}(w)$	G	extension blocks connections directed to the website w based on filter lists (e.g. `Disconnect`).
$\mathit{Block\_tp\_cookie}(w)$	G	3rd-party cookie blocking for website w.

Table 4. Web Tracking Predicates

The predicates are derived (D) from one or more rules of the model.

$\mathit{Link}(w, w')$	D	websites w and w' have a *possible* path to share information w/o exchange of cookies.
$\mathit{Link}_{\mathit{cookie}}(w, w')$	D	websites w and w' have a *possible* path to share information via cookies from w to w'.
$\mathit{Access}(w, w')$	D	website w forces to access a resource in w' via a redirection or an inclusion.
$\mathit{Access}_{\mathit{cookie}}(w, w')$	D	website w forces to access a resource in w' via a redirection (inclusion) attaching w's cookies.
$\mathit{Cookie\_sync}(w, w')$	D	website w synchronizes its cookies with the website w'. The operation is unidirectional.
$\mathit{Knows}(w, w')$	D	*potential* ability of website w to track users on a (possibly different) website w'.
$\mathit{req\_COPPA}(w)$	D	website w should comply to COPPA.

is evaluated as true (e.g. `Disconnect` blocks the website w), then all requests to w are blocked. If $\mathit{Block\_tp\_cookie}(w)$ is evaluated as true (e.g. 3rd-party cookie blocking protection is active), it means that the browser does not allow w to set HTTP cookies, however it does not block HTTP requests directed to w.

Tab. 4 summarizes the predicates that describe a possible exchange of users information between websites. $\mathit{Link}_{\mathit{cookie}}(w, w')$ and $\mathit{Link}(w, w')$ identify a possible path, as a result of an inclusion or a redirection, between w and w' that can be exploited for tracking. $\mathit{Access}(w, w')$ and $\mathit{Access}_{\mathit{cookie}}(w, w')$ capture a successfully redirection or inclusion that forces a user to contact website w' from the website w. $\mathit{Cookie\_sync}(w, w')$ indicates cookie syncing between w and w' to share IDs.

The predicates in Tab. 3 are obtained from the collected data as we will see in §8. The predicates in Tab. 4 are inferred from the rules of the model.

3.2 General Derivation Rules

The model includes the classical Gentzen rules for the quantifier-free fragment of intuitionistic first-order logic (IFOL). We use IFOL since its proofs are constructive and thus there is a pairing between proofs and attacks.

Definition 1 (**Internet Snapshot**). *The symbol $\mathcal{N}$, possibly with subscripts, denotes finite (possibly empty) set of instantiated predicates from Tab. 3 and 4 that captures the interactions (inclusions, redirections, etc.) among websites observed on the Internet.* ■

We denote the pure "status" of the internet with $\mathcal{N}$ and the set of predicates capturing a specific mitigation $\mathcal{X}$ on the internet snapshot $\mathcal{N}$ with $\mathcal{N}^*_{\mathcal{X}}$. For example, $\mathcal{N} = \{\mathit{IncludeContent}(w, w'), \mathit{Redirect}(w', w''), \ldots\}$ and $\mathcal{N}^*_{\mathcal{X}} = \{\mathit{Block\_request}(w'), \mathit{Block\_request}(w''), \ldots\}$.

The turnstyle $\vdash$ separates the assumptions on the left from the propositions on the right. The sequence of formulas on the left of $\vdash$ are in conjunction. The horizontal line separates preconditions from postconditions.

The capital letters A, B and C, possibly with subscripts, denote formulae of the quantifier-free fragment of IFOL, whose predicates are drawn from Tab. 3 and 4. The variable $w \in \mathcal{W}$, possibly with apices, is a variable over websites. Constants (e.g *facebook.com*) denote websites. *WL* and *WR* stand for Weakening Left/Right, *CL* for Contraction Left. We have the following rules for *intuitionistic* logic:

$$\frac{}{A \vdash A}\ (Ax) \qquad \frac{\mathcal{N} \vdash A \qquad A, \mathcal{N} \vdash B}{\mathcal{N} \vdash B}\ (Cut) \qquad \frac{\mathcal{N}, A, A \vdash B}{\mathcal{N}, A \vdash B}\ (CL)$$

$$\frac{\mathcal{N} \vdash B}{\mathcal{N}, A \vdash B}\ (WL) \qquad \frac{\mathcal{N} \vdash}{\mathcal{N} \vdash B}\ (WR) \qquad \frac{\mathcal{N} \vdash A}{\mathcal{N}, \neg A \vdash}\ (\neg L) \qquad \frac{\mathcal{N}, A \vdash}{\mathcal{N} \vdash \neg A}\ (\neg R)$$

$$\frac{\mathcal{N}, A, B \vdash C}{\mathcal{N}, A \land B \vdash C}\ (\land L) \qquad \frac{\mathcal{N} \vdash A \qquad \mathcal{N} \vdash B}{\mathcal{N} \vdash A \land B}\ (\land R)$$

$$\frac{\mathcal{N}, A \vdash C \qquad \mathcal{N}, B \vdash C}{\mathcal{N}, A \lor B \vdash C}\ (\lor L) \qquad \frac{\mathcal{N} \vdash A}{\mathcal{N} \vdash A \lor B}\ (\lor R_1) \qquad \frac{\mathcal{N} \vdash B}{\mathcal{N} \vdash A \lor B}\ (\lor R_1)$$

$$\frac{\mathcal{N} \vdash A \qquad \mathcal{N}, B \vdash C}{\mathcal{N}, A \to B \vdash C}\ (\to L) \qquad \frac{\mathcal{N}, A \vdash B}{\mathcal{N} \vdash A \to B}\ (\to R)$$

In our derivations, we do use neither $\lor R_i$ nor $\lor L$ rules as we are only interested in deriving knowledge predicates.

We can also have domain-specific axioms of the form $A \to B$ that can be added to a derivation with the rule:

$$\frac{\mathcal{N}, A \to B \vdash C \quad A \to B \text{ is Domain Axiom}}{\mathcal{N} \vdash C}\ DomAx$$

We represent a domain axiom $A_1 \land ... \land A_n \to B$ as a rule $\frac{A_1, \ldots, A_n}{B}$ and vice-versa as in IFOL, $\vdash$ and $\to$ are interchangeable. Tracking specific rules in the next section are indeed domain axiom.

4 Tracking Specific Rules

Information Flow

In Fig. 1a the rules `IncludeW` and `RedirectW` show how a link between two website w' and w can be created. Rule `Redirect` (`Include`) in Fig. 1a illustrates how the redirection to (inclusion of) another website can be employed by trackers to pass information, for example cookies. The rule `ImpRed` (`ImpInc`) in Fig. 1b shows that the predicates $Redirect_{cookie}$ ($IncludeContent_{cookie}$) implies the predicates $Redirect$ ($IncludeContent$).

Network Interactions

Rules `AccessToW` and `AccessTo` in Fig. 1a describe access to resources with a possible propagation of information between two websites w and w' (w/ or w/o an exchange of cookies). The rule `PropagateAccess` shows how the access can be propagated through websites.

Third-Party Tracking

The rule `3rdpartyTracking` in Fig. 1a shows that 3rd-parties present on a website w can track users. This rule can be applied recursively to describe complex interactions among websites as shown in the Appendix A.2. This rule does not consider the possibility of browser fingerprinting (not blocked by the *Block_tp_cookie* mitigation). Since we are interested in the data sharing process that brings to track users and no information on how fingerprints are shared is available. Furthermore, as pointed out by several works ([48, 49]), browser fingerprinting is not as accurate as cookies to identify users. Thus, ad transactions carried out without the presence of cookies are not enough to produce targeted advertisements [6]. We further discuss this extension in §11.

Cookie Syncing

The rule `Sync` in Fig. 1c shows the preconditions required to implement cookie syncing between websites w and w'. Cookie syncing requires the exchange of cookies to link the IDs used by the two trackers. This technique is also called *First to Third-party Cookie Syncing* [20]. Rule `PropagateSync` shows how to propagate cookie syncing through a sequence of websites.

Tracking via Cookie Syncing

In Fig. 1c, the rule `SyncTracking` describes how cookie syncing between websites w' and w'' allows to track users even in websites where a tracker is not explicitly present. This is known as *Third to Third-party Cookie Syncing* [20]. We did not define a rule to describe *cookie forwarding* because it is a special case of `3rdpartyTracking` where the tracker passively receives the user's history collected by a 3rd-party. The cookies forwarded could be used for back-office exchange that is outside of the scope of our model. All the rules assume the intention of the websites to track users.

COPPA Compliance

A website w must comply with COPPA if at least one of these conditions hold [42] for w:

1. is directed to children under 13 y/o and collects PII.
2. is directed to children under 13 y/o and allows another website w' to collect PII.
3. has a general audience, but it has actual knowledge that it collects PII from children under 13 y/o.
4. collects PII from users of a website w' directed to children under 13 y/o.

However, websites that fall in conditions (1) and (3) and collect *only* persistent identifiers (e.g. cookies) are not obliged to comply with COPPA if the persistent identifier is used for internal operations *only*. It is important to underline that this exception does not allow behav-

$\dfrac{\mathit{IncludeContent}(w, w')}{\mathit{Link}(w, w')}\ \texttt{IncludeW}\quad \dfrac{\mathit{Redirect}(w, w')}{\mathit{Link}(w, w')}\ \texttt{RedirectW}$	If a website w includes content from (redirects to) site w', then there is a link between w and w' that allows an exchange of information.
$\dfrac{\mathit{Redirect}_{\mathit{cookie}}(w, w')}{\mathit{Link}_{\mathit{cookie}}(w, w')}\ \texttt{Redirect}\quad \dfrac{\mathit{IncludeContent}_{\mathit{cookie}}(w, w')}{\mathit{Link}_{\mathit{cookie}}(w, w')}\ \texttt{Include}$	During a redirection (inclusion) it is possible to append a cookie of w for w'.
$\dfrac{\mathit{Link}(w, w') \quad \neg \mathit{Block\_request}(w')}{\mathit{Access}(w, w')}\ \texttt{AccessToW}$ $\dfrac{\mathit{Link}_{\mathit{cookie}}(w, w') \quad \neg \mathit{Block\_request}(w')}{\mathit{Access}_{\mathit{cookie}}(w, w')}\ \texttt{AccessTo}$	If a website w includes content from (redirects to) a website w' (this case includes connections exploiting social buttons) that is not blocked by any extension, then the user is forced to access the resources of w' from w.
$\dfrac{\mathit{Access}(w, w') \quad \mathit{Access}(w', w'')}{\mathit{Access}(w, w'')}\ \texttt{PropagateAccess}$	If a website w forces to access the resources of w' and w' forces to access the resources of w'', then the user that visits w is forced to access website w''.
$\dfrac{\begin{array}{l}\mathit{Visits}(w)\\ \mathit{Access}(w, w') \quad \neg \mathit{Block\_tp\_cookie}(w')\end{array}}{\mathit{Knows}(w', w)}\ \texttt{3rdpartyTracking}$	If a user visits a website w that forces to access a website w' not blocked by any mitigation, then w' knows that the user visited w.

(a) Tracking Flow

$\dfrac{\mathit{IncludeContent}_{\mathit{cookie}}(w, w')}{\mathit{IncludeContent}(w, w')}\ \texttt{ImpInc}\quad \dfrac{\mathit{Redirect}_{\mathit{cookie}}(w, w')}{\mathit{Redirect}(w, w')}\ \texttt{ImpRed}$	$\mathit{Redirect}_{\mathit{cookie}}$ and $\mathit{IncludeContent}_{\mathit{cookie}}$ are a particular case of *Redirect* and *Include* respectively.

(b) Tracking Implications

$\dfrac{\mathit{Access}_{\mathit{cookie}}(w, w') \quad \neg \mathit{Block\_tp\_cookie}(w')}{\mathit{Cookie\_sync}(w, w')}\ \texttt{Sync}$ $\dfrac{\mathit{Cookie\_sync}(w, w') \quad \mathit{Cookie\_sync}(w', w'')}{\mathit{Cookie\_sync}(w, w'')}\ \texttt{PropagateSync}$	A website w redirects the user to a website w' inserting cookies of w in the request. If the connection to w' is not blocked by any mitigation and the browser allows w' to set its cookies then w' can receive w's cookies and synchronize them with its cookies. Cookie syncing can be propagated.
$\dfrac{\mathit{Knows}(w', w) \quad \mathit{Cookie\_sync}(w', w'')}{\mathit{Knows}(w'', w)}\ \texttt{SyncTracking}$	The presence of cookie syncing with w' allows a website w'' to track users on the website w even if it is not explicitly present.

(c) Tracking With Cookie Sharing

$\dfrac{\mathit{Knows}(w, w') \quad \mathit{Kids}(w') \quad w \neq w'}{\mathit{req\_COPPA}(w')}\ \texttt{COPPAcomplRelease}$	If a website w tracks users on a children related website w', then w' should comply with COPPA. This rule covers case (2).
$\dfrac{\mathit{Knows}(w, w') \quad \mathit{Kids}(w') \quad w \neq w'}{\mathit{req\_COPPA}(w)}\ \texttt{COPPAcomplCollect}$	If a website w tracks users on a children related website w', then w should comply with COPPA. This rule covers case (4).
$\dfrac{\mathit{Kids}(w) \quad \mathit{Knows}(w, w') \quad \mathit{BehavioralAds}(w)}{\mathit{req\_COPPA}(w)}\ \texttt{COPPAcomplBehav}$	If w is a children related website that collects PII on an external website w' then it can perform behavioral advertising. This rule covers the cases (1) and (3).
$\dfrac{\mathit{Kids}(w) \quad \mathit{Cookie\_sync}(w', w)}{\mathit{req\_COPPA}(w)}\ \texttt{COPPAcomplCS}$	It is a special case of `COPPAcomplBehav`. If w is a children related website and performs cookie syncing with w' (i.e. it receives cookies from w') then it can create profiles for its users for behavioral advertising.

(d) COPPA Compliant

Fig. 1. Tracking Derivation Rules

ioral advertising. Fig. 1d shows the rules that describe when a website should comply with COPPA. The predicate *Kids*(w) describes a website directed to children under 13 y/o, *req_COPPA*(w) identifies a website that should comply to COPPA.

`COPPAcomplRelease` and `COPPAcomplCollect` describe conditions (2) and (4): if a children related website w' allows a website w to track its users then both websites must comply with COPPA. We impose $w \neq w'$ to not fall in the conditions (1) and (3) where COPPA is not mandatory if used for internal activities. It is important to underline that in our model the *Knows* predicate implies the employment of a persistent identifiers (e.g. cookies). The scenario described in `COPPAcomplCollect` is not always straightforward to be observed due to exchange of cookies (e.g. cookie syncing) among websites.

`COPPAcomplBehav` captures cases (1) and (3). Our model describes only personal identifiers, thus we need to determine if a certain website uses this information for external operations (e.g. behavioral advertising). *BehavioralAds*(w) could be instantiated using the approach presented by Liu et al. [50] and we leave for future work the integration with our model. Rule `COPPAcomplCS` shows a special case of `COPPAcomplBehav` where a children related website w receives cookies from another website. Cookie syncing is a known technique utilized for behavioral advertising [4]. It is important to underline that the opposite case (*Cookie_sync*(w, w')), in which the children related website w sends cookies to an external website, is already covered by the rule `COPPAcomplCollect` since *Cookie_sync*(w, w') generates a *Knows*(w', w) that triggers the mentioned rule.

Do we really need a formal approach? Consider Youtube and assume *Kids*(*youtube.com*) always holds (as sometimes it might be necessary to treat information according to COPPA). We might be tempted to conclude that any website that includes cookies from *youtube.com* should be COPPA compliant. This informal reasoning seems to imply that any website importing a social button or a video from Youtube should be COPPA compliant. However, by applying the set of rules we previously presented we can instead prove that this is actually incorrect. Suppose *Kids*(*youtube.com*), we have an *IncludeContent*(*abc.com*, *youtube.com*) due to the social gadget, and thus by applying rules `IncludeW`, `AccessToW`, and `3rdpartyTracking` we have *Knows*(*youtube.com*, *abc.com*). At this point, none of the COPPA rules can produce *req_COPPA*(*abc.com*). Instead, it is possible to trigger rule `COPPAcomplBehav` by showing that *youtube.com* is performing behavioral advertising and thus should comply with COPPA.

As previously stated, the predicate *Knows* describes the *potential* ability of a website to track users. Thus, the obtainment of the predicate *req_COPPA* is not by itself a *definitive* proof of the need for compliance. However, it provides an explanation that can trigger further investigations by the FTC on which data are actually sent (for example, due to a complaint of a parent). Websites must then prove that the exchange either did not occur or did not contain children's information.

5 Decidability and Theorem Proving

To show the decidability of our construction we rely on the relation between logic programs and a fragment of intuitionistic logic (in particular Harrop formulae [51]).

Theorem 1 (PTIME Knows Decidability). *It is possible to decide whether the internet snapshot $\mathcal{N}$ allows a website w^* to know about the user's visit to another website w ($\mathcal{N} \vdash Knows(w^*, w)$) in poly time in the size of the snapshot $\mathcal{N}$.* ■

Proof. We rely on embedding both snapshot and rules as a Harrop formulae.

$$\begin{aligned} G \ &::= \ A \mid G_1 \wedge G_2 \mid H \rightarrow G \mid \\ &\qquad \mid G_1 \vee G_2 \mid \forall w G \mid \exists w G \qquad \text{\%Not used here} \end{aligned} \tag{1}$$

$$H \ ::= \ A \mid G \rightarrow A \mid \forall w H \mid H_1 \wedge H_2 \tag{2}$$

where A is a predicate, G is a *goal formula* and H is an *Harrop formula.* An internet snapshot $\mathcal{N}$ is encoded as a (large) conjunction which is a Harrop formula. Each rule from §4 is encoded as a goal formula. For example, `3rdpartyTracking` can be coded as a Harrop formula:

$$\overbrace{\forall w w' \ \overbrace{\overbrace{\overbrace{Visits(w)}^{G::=A} \wedge \overbrace{Access(w, w')}^{G::=A} \wedge \overbrace{Block\_tp\_cookie(w') \rightarrow \bot}^{G::=H\rightarrow A}}^{G::=G_1\wedge G_2\wedge G_3} \rightarrow \overbrace{Knows(w', w)}^{A}}^{H::=G\rightarrow A}}^{H::=\forall w\forall w' H}$$

From Theorem A in [51] the pair of the query and the rules LJ from §3.2 are a logic programming language. As we have no disjunction on the right of $\vdash$ for the query of interest, the rules ($\vee R_i$) responsible for the PSPACE complexity of intuitionistic logic do not apply.

Since $\mathcal{N}$ is finite, there are at most $O(|\mathcal{N}|^2)$ different constants as we have at most two arguments for each predicate. Hence, the instantiation of all quantified formulae embedding the rules generates *at most* $O(|\mathcal{N}|^6)$

ground propositional rules (we have at most three variable per rule), even if no optimization can be done (e.g. distinguishing between content delivery networks and actual websites). Thus, the ground instantiation of the rules is poly in the size of the snapshot and also the query of interest can be decided in poly time. ■

We do *not* claim that the calculus of tracking for *arbitrary formulae* including knowledge predicates is tractable. The presence of disjunction on the right would make decidability jump to PSPACE [52] as one could encode QBF as a decision problem in the formula on the right of $\vdash$. This decision problem could well use $Knows(w^*, w)$ as predicates but they could be replaced with abstract ps and qs and would have no relation with the complexity of inferring visibility relations on the internet. As of now, we do not see a practical need to incorporate disjunction on the right.

From Theorem 1 follows that COPPA compliance rules can be also encoded as Hereditary Harrop formulas using the knowledge relations as basic atoms:

Corollary 1.1 (PTIME COPPA Compliance). *It is possible to decide whether the internet snapshot $\mathcal{N}$ requires a website w to be COPPA Compliant ($\mathcal{N} \vdash req\_COPPA(w)$) in poly time in the size of $\mathcal{N}$.*

Next, we show that from a derivation we can reconstruct the connections responsible for the tracking.

Theorem 2 (Map Proofs to Configurations). *Given a derivation of $Knows(w^*, w)$ from an internet snapshot $\mathcal{N}$ ($\mathcal{N} \vdash Knows(w^*, w)$), one can extract an* essential *subset of the configuration $\mathcal{N}_\omega \subseteq \mathcal{N}$ such that $\mathcal{N} \setminus \mathcal{N}_\omega \not\vdash Knows(w^*, w)$.* ■

Proof. This result follows from the existence of uniform proofs[6] for the fragment of interest [53] and the existence of a feasible interpolation for intuitionistic logic [54, 55]. Given a derivation of $\mathcal{N} \vdash Knows(w^*, w)$ one can construct a uniform proof and the existence of the interpolant guarantees that we have a set of formulae that only includes constants shared from the antecedent (the internet configuration) and the succedent (the knowledge predicate). Hence we can use the proof to reconstruct the tracking steps and data exchanges responsible for $Knows(w^*, w)$ in a subset $\mathcal{N}_1$, as the predicates present in the proof and the interpolant. We can then eliminate $\mathcal{N}_1$ from $\mathcal{N}$ and try to derive $\mathcal{N} \setminus \mathcal{N}_1 \vdash Knows(w^*, w)$. If we succeed, it means there is another way to exchange data, so we extract a new subset $\mathcal{N}_2$ and continue the process until for $\mathcal{N}_i$, $i = 1 \ldots$ no derivation is possible. As deciding a single query is decidable in polynomial time (See Theorem 1) the process terminates after a polynomial number of interactions. The union of all sets $\mathcal{N}_i$ is the desired set $\mathcal{N}_\omega$. ■

As immediate from the proof above, one could also stop the search as soon as the first subset of the internet snapshot, $\mathcal{N}_1$, responsible for the tracking is identified. This is what we do with a theorem prover. There may be more than one proof because a prover can choose to apply one rule before another one according to a suitable heuristic that may lead to a faster proof search (see GAPT [57] for additional information). Different proofs may also come from the existence of different tracking possibilities on the Internet. The important thing is that one can be found in poly time (see Th.1).

6 Using the Calculus for Tracking Relations

We formally define tracking relations that are of practical interest through our formal model. We illustrate some of these relations in the practical case of Alexa Top 5, 10, 50, and 100 websites later in §9.

Flow Propagation

Given the sharing of information through redirections, content inclusions, and cookie syncing and given a sequence of visited web sites, we can study how the knowledge about this sequence is distributed on the Internet. This is possible through a graph where we underline edges with predicate $Knows(w', w)$ to identify the websites that know if a user visited another website. We can map this representation to a Venn diagram where we identify which trackers are directly and indirectly included in the websites visited. We define a mapping between the predicate *Knows* and the set theory:

$$KnowsUser(\mathcal{N}, w) = \{w^* \mid \mathcal{N} \vdash Knows(w^*, w)\} \qquad (3)$$

where $KnowsUser(\mathcal{N}, w)$ represent the set of websites w^* that are able to track a user on the website w in an Internet snapshot $\mathcal{N}$.

[6] A finite constructive process applies uniformly to every formula, either producing an intuitionistic proof of the formula or demonstrating that no such proof can exist.

Lowest Tracking Coverage

Our formal model generates relations between websites through *Knows* predicates for a given $\mathcal{N}$. We compare different mitigations to determine which produces the lowest tracking coverage. A mitigation $\mathcal{X}$ in an Internet snapshot $\mathcal{N}$ ($\mathcal{N}^*_{\mathcal{X}}$), disables some *Knows* predicates.

Definition 2 (**Mitigation Subsumption**). *Let $\mathcal{N}$ be an Internet snapshot and $\mathcal{N}^*_{\mathcal{X}}$,$\mathcal{N}^*_{\mathcal{Y}}$ two mitigations. We say that the mitigation $\mathcal{N}^*_{\mathcal{X}}$ is* more effective *than $\mathcal{N}^*_{\mathcal{Y}}$ iff $\forall$ pairs (w, w'): $\mathcal{N}, \mathcal{N}^*_{\mathcal{X}} \vdash Knows(w', w)$ implies $\mathcal{N}, \mathcal{N}^*_{\mathcal{Y}} \vdash Knows(w', w)$.* ■

Intuitively, any *Knows* predicate obtained from $\mathcal{N}$ applying the mitigation $\mathcal{N}^*_{\mathcal{X}}$ is also obtained from $\mathcal{N}$ applying $\mathcal{N}^*_{\mathcal{Y}}$. We developed a script that automatically generates TPTP input files for `Slakje` to obtain a proof for *Mitigation subsumption.* Unfortunately, this definition can be rarely applied. Indeed, if the mitigations modify different parts of the graph of *Knows* predicates, the results cannot be compared. Thus, we propose a quantitative analysis.

Given an Internet snapshot $\mathcal{N}$, a mitigation $\mathcal{N}^*_{\mathcal{X}}$ is better than a mitigation $\mathcal{N}^*_{\mathcal{Y}}$ if *both* conditions hold:

- *C1:* The mitigation $\mathcal{N}^*_{\mathcal{X}}$ blocks access to a smaller number of websites compared to the mitigation $\mathcal{N}^*_{\mathcal{Y}}$
- *C2:* The trackers obtained with $\mathcal{N}^*_{\mathcal{X}}$ are smaller in number compared to the trackers obtained with $\mathcal{N}^*_{\mathcal{Y}}$

Given an Internet snapshot $\mathcal{N}$, we define its *access size* and *knowledge size* respectively as follows

$$||\mathcal{N}||_A = \sum_{w \in \mathcal{W}} \left| \left\{ (w, w^*) \in \mathcal{W}^2 \mid \mathcal{N} \vdash Access(w, w^*) \right\} \right|$$

$$||\mathcal{N}||_K = \sum_{w \in \mathcal{W}} \left| \left\{ (w, w^*) \in \mathcal{W}^2 \mid \mathcal{N} \vdash Knows(w, w^*) \right\} \right|$$

Site breakage, used to compare mitigations' performance [37], is directly influenced by the *access size.* We can now compare two mitigations $\mathcal{N}^*_{\mathcal{X}}$ and $\mathcal{N}^*_{\mathcal{Y}}$

Definition 3 (**Quantitative Mitig. Subsumption**). *Mitigation $\mathcal{N}^*_{\mathcal{X}}$ is* quantitatively more effective *than mitigation $\mathcal{N}^*_{\mathcal{Y}}$ iff the access size of $\mathcal{X}$ is larger (or equal) than the access size of $\mathcal{Y}$, $||\mathcal{N}^*_{\mathcal{X}}||_A \geq ||\mathcal{N}^*_{\mathcal{Y}}||_A$, and the knowledge size of $\mathcal{X}$ is smaller than the knowledge size of $\mathcal{Y}$, $||\mathcal{N}^*_{\mathcal{X}}||_K < ||\mathcal{N}^*_{\mathcal{Y}}||_K$.* ■

Intuitively the mitigation $\mathcal{X}$ reduces the number of trackers that know the user's visited websites more than $\mathcal{Y}$ does, while still keeping a larger (or equal) number of accessible sites than $\mathcal{Y}$. The ideal performance would be to drop one accessed site per blocked accessed tracker (i.e we lost only the tracker itself).

However, one of the major concerns in terms of privacy is not that a high number of trackers knows about fragments of a user's activity but that few trackers can reconstruct (almost entirely) the activity of a user. For example, *Google* is present in roughly 80% of the Top 1 million domains [19] and thus, has a high tracking coverage. We thus propose an additional definition:

Definition 4 (**Mitigation Subsump. per Tracker**). *Mitigation $\mathcal{N}^*_{\mathcal{X}}$ is* quantitatively more effective *than mitigation $\mathcal{N}^*_{\mathcal{Y}}$ against the tracker w iff the access size of $\mathcal{X}$ is larger (or equal) than the access size of $\mathcal{Y}$, $||\mathcal{N}^*_{\mathcal{X}}||_A \geq ||\mathcal{N}^*_{\mathcal{Y}}||_A$, and the knowledge size of $\mathcal{X}$ projected to w is smaller than the knowledge size of $\mathcal{Y}$ projected to w.* ■

In other words, the mitigation $\mathcal{N}^*_{\mathcal{X}}$ produces a higher reduction of websites where the tracker w can control a user compared to the mitigation $\mathcal{N}^*_{\mathcal{Y}}$ while still keeping a larger (or equal) number of accessible sites than $\mathcal{Y}$.

Unfortunately, it is hard to fulfill both conditions (as we will see in §9.1, Fig. 7): being less tracked means losing more access.

7 Coping With Uncertainty

The model considers the *possibility* of tracking practices given a static description of the Internet (e.g. from OpenWPM). However, interactions among websites are often non-deterministic [56] and the same applies to the tracking behaviors [7]. For example, 3rd-parties embedded in a website can include different 3rd-parties depending on the results of RTB and thus changing the set of connections observed. Furthermore, trackers can have different behaviors on the same site, for example, a tracker can behave both as analytics (and thus cannot track the users over different websites) and as a 3rd-party tracker [7]. To handle this uncertainty, we extended the model by considering the likelihoods of deriving predicates over different snapshots:

- A snapshot $\mathcal{N}$ is a picture of the Internet at some point in time and it is deterministic by construction (either an inclusion is there or it is not, same for mitigations). We intuitionistically derive either the predicate A, $\neg A$, or neither (if we do not have

enough data, e.g. OpenWPM failed to detect an inclusion) but *never* both.
- Uncertainty stems from the fact that snapshots change over time. So yesterday A held, the day before yesterday $\neg A$ held, and three days ago neither was derivable. What can we conclude about today if we do not want to resample it?

The overall semantics for N snapshots is:

$$< \mathcal{N}_1, \dots, \mathcal{N}_N > \vdash A\ (a, b) \iff$$
$$|\mathcal{N}_i : \mathcal{N}_i \vdash A| = a * N \ \wedge \ |\mathcal{N}_i : \mathcal{N}_i \vdash \neg A| = (1-b) * N$$

To each derivation we associate a minimal and a maximal likelihood in the same way Ferson et al. [57] derived a probability-box:

- a: likelihood that A is derivable from the predicates in the Internet snapshots.
- $1-b$: likelihood that $\neg A$ is derivable from the predicates in the Internet snapshots.

In other words, a is the minimum likelihood that A is gonna be true, while b is the maximum.

The p-box captures the evidence across all snapshots so it must potentially include evidence that A holds for sure ($a * N$ snapshots) and it does not hold for sure ($(1-b) * N$ snapshots where $\neg A$ was found to hold e.g. when mitigations were detected). The gap between a and b measures the uncertainty i.e. what we cannot prove because the law of excluded middle does not hold and thus $b \neq (1-a)$. To compute the uncertainty a brute-force solution is just to derive the proof for each snapshot and then aggregate the results.

The likelihood provides information on the Internet as an evolving ecosystem. At any given time of course in the snapshot true at that moment, the probability collapses to 0 or 1, in the same way that a tossed coin is always head or tail.

To illustrate the extension we considered 17 snapshots from January 2016 to October 2017 obtained from WebCensus (see Section 8) and we computed the likelihood that the derivations, responsible for the proof $\mathit{Knows}(\mathit{revsci.net}, \mathit{qq.com})$ in Fig. 15, are obtained from the snapshots. Fig. 15 shows the likelihood derivation for a snapshot $\mathcal{N}_i$. In Appendix A.3 we present the exhaustive set of rules used in the proof. The maximum likelihood b for some derivations is 1 because, as we cannot observe a $\neg \mathit{IncludeContent}$, we cannot exclude that the interaction happened but we failed to observe it.

8 Dataset and Scalability

8.1 Dataset for Internet Snapshot

We evaluate our model with the 10k Site ID Detection(1) 2016 dataset[7] collected using a stateful instance of OpenWPM[8]. We summarize the tables and columns employed to instantiate the predicates of our model in Fig. 2. The table `site_visits` contains the list of the Top 10k Alexa visited domains. The table `urls` contains the set of URLs loaded during the crawling. The table `http_responses` contains the HTTP responses.

From the dataset, we extracted the sequence of redirections and inclusions necessary to instantiate the predicates. From table `http_responses` we employed *visit_id* (ids for the top 10k websites), *url_id* (ids for URLs of the HTTP responses), *response_status* (HTTP Status Codes), *location_id* (in case of a redirection, ids for a new URL to visit. NULL otherwise), and *time_stamp* (timestamps of HTTP responses).

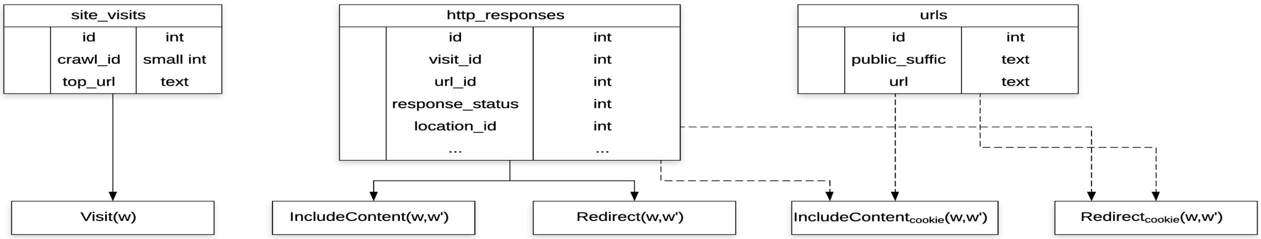
We provide an example of the mapping into the model in the Appendix A.5. Tab. 5 shows the number of HTTP responses received for the Top Alexa. The responses are used to find the sequence of redirections.

In addition, Tab. 5 shows the number of predicates obtained applying our model on the Top 10, 20, 30, 40, and 50 Alexa domains of the dataset without any mitigation. After visiting the Top 50 domains the users contacted 190 different websites with more than 6k connections. The number of redirections remains relatively small compared to the number of inclusions observed.

The number of HTTP responses in Tab. 5 for the Top 30, 40 and 50 domains are slightly different from the values of $\mathit{Link}(w, w')$ because the crawler failed to collect some HTTP responses during a sequence of redirections probably due to network problems. However, we can correctly close the sequence even if a response is missing.

To compute the sequence of redirections, we first extract the sequence of HTTP responses for each *visit_id* considered, then order the responses using the column *time_stamp* (to avoid considering intermediate redirections as the beginning of a new connection) and extract the redirections from the set of HTTP responses. Cookie syncing can be detected by analyzing URLs [20] and payloads in POST requests. For illustrative purposes,

[7] https://webtransparency.cs.princeton.edu/webcensus/
[8] https://github.com/mozilla/OpenWPM



Tables from OpenWPM used to instanciate the predicates *Visits*, *IncludeContent*, and *Redirect* of our model (continuos lines). It is also possible to instanciate the predicates $IncludeContent_{cookie}$ and $Redirect_{cookie}$ from the tables `http_responses` and `urls` (dashed lines) but in Section 9.1 we employed empirically validated pairs from [17].

Fig. 2. Mapping of the Predicates to the Dataset.

Table 5. # of Predicates and HTTP Responses for the Top Alexa

Variables vs Top Domains	10	20	30	40	50
HTTP responses	925	1957	2864	3618	4530
$IncludeContent(w, w')$	824	1803	2681	3391	4272
$Redirect(w, w')$	101	154	184	229	261
$Link(w, w')$	925	1957	2865	3620	4533
$Link_{cookie}(w, w')$	3	3	3	5	6
$Access(w, w')$	925	2272	3636	5024	6382
$Access_{cookie}(w, w')$	3	3	3	5	6
$Cookie\_sync(w, w')$	3	3	3	7	8

we use here 200 empirically validated domain pairs performing cookies syncing from [17].

8.2 Theorem Proving Implementation

We leverage on the GAPT tool [58] to generate proofs for the *Knows* and the *req_COPPA* predicates. We use the intuitionistic prover `Slakje` [59] to produce formal proofs based on the rules in our model. We encode the model and the data using the TPTP syntax. The data used for the axioms is *generated from actual data* obtained using OpenWPM. We instantiated the *Kids*(w) predicate using the Top 50 Alexa In the *Kids and Teens* category. This approach is fully-automated by a script that generates a sequence of axioms from the database, the model, and the conjecture to prove.

Fig. 3 shows a fragment of the TPTP input for the prover, where the model is encoded and the relevant data are inserted as axioms. Fig. 15 and 14 in the Appendix A.4 show an example of the proof generated by `Slakje` for *Knows*(*revsci.net*, *qq.com*) and *req_COPPA*(*flashtalking.com*) respectively. We evaluated the performance of `Slakje` by assuming the Top 5, 10, 50, and 100 as *visited* domains to generate a proof for $Knows(facebook.com, fbcdn.net)$ and the vice versa.

```
%----IncludeW rule
fof(icw,axiom, (! [W,W1] :
(includeContent(W,W1) => link(W,W1)))).
%----RedirectW
fof(rw,axiom,(! [W,W1] :
(redirect(W,W1) => link(W,W1)))).
...
fof(var6,axiom,(includeContent(facebook.com,fbcdn.net)))
fof(var7,axiom,(includeContent(facebook.com,atdmt.com)))
...
fof(var547,conjecture,(knows(fbcdn.net,facebook.com)))
```

Fig. 3. Fragment of the TPTP Input for `Slakje` to Prove *Knows*(*fbcdn.net*, *facebook.com*)

Table 6. Timing for Successful and Failed Proof Attempts

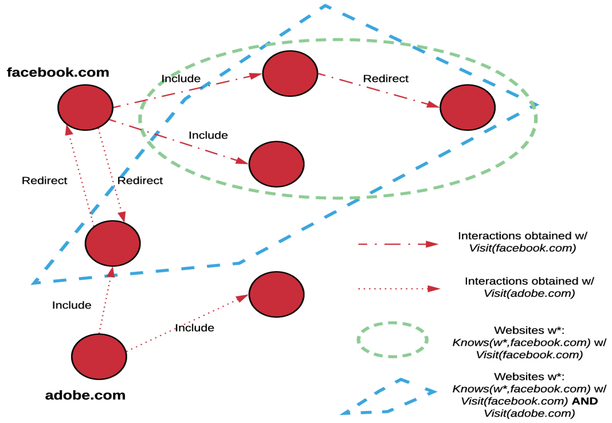
Run `Slakje` to prove *Knows*(*fbcdn.net*, *facebook.com*) and viceversa (not provable) with different visited domains

# visited	TPTP input	Successful proof	Failed proof
domains	[# axioms]	Time [sec]	Time [sec]
5	75	1.4	1.1
10	209	1.8	1.6
50	867	10.5	19.3
100	2,343	1,469.8	>3,600

Tab. 6 shows the performance with an Intel i7-8750H @ 2.20GHz and 2 GB RAM for the Java VM.

8.3 Scalability

Our complexity analysis gives an upper-bound of $O(|\mathcal{N}|^6)$, which is inadequate for the application of the approach beyond very compact domains. Indeed our goal is not to provide Internet-scale analysis but third-party verifiable evidence for *individual cases* where numbers are manageable. For example, users rarely visit


facebook.com
Include
Redirect
Include
Redirect
Redirect
Include
Include
adobe.com
Interactions obtained w/ Visit(facebook.com)
Interactions obtained w/ Visit(adobe.com)
Websites w*: Knows(w*,facebook.com) w/ Visit(facebook.com)
Websites w*: Knows(w*,facebook.com) w/ Visit(facebook.com) AND Visit(adobe.com)

Determine which websites w' knows about the visit of *facebook.com* ($Knows(w', facebook.com)$) by analyzing only on the interactions generated by the *facebook.com* visit misses interactions generated by *adobe.com* (visited by the user) with *facebook.com* and thus potential trackers.

Fig. 4. The Problem of Determine $Knows(w', facebook.com)$

more than 100/120 websites [35, 60][9], the cookie duration is typically short [61] and cliques, important for COPPA, are relatively small [19]. Thus, scalability in this application is not a problem.

As shown in Tab. 6, the time required to generate a proof increases with the number of axioms. This number is dependent on the interactions observed by the user on the *visited* websites. To improve performance we perform DBMS slicing by extracting only the interactions that are obtained from the user's visited websites (e.g. Top5, Top10, etc.) and not the entire Internet interactions and then perform proof reconstruction. Unsound search followed by proof reconstruction is a new trend in Automated Reasoning [62]. This is the minimum set of interactions (and thus axioms) that must be considered to avoid missing possible tracking practices. For example, if we extract only the interactions generated by visiting a website w and not all the other visited websites we can miss interactions generated from other visits that reach w as shown in Fig. 4.

9 Analysis of Mitigations

9.1 Evaluation of Tracking Relations

We evaluated our approach on the dataset previously presented with the filter list of three widely deployed extensions (`Ghostery`, `Disconnect`, `Adblock Plus`). We neither consider the Firefox third-party cookie blocking feature for unvisited websites[10] nor other Firefox configurations that were either too restrictive (e.g. block all cookies) or they overlap (e.g. Firefox uses `Disconnect` blacklist). We used the blacklist of `Ghostery`, `Disconnect`, and `Adblock Plus` from Bashir et al. [3, 17] (the data was *collected* in 2016 too). We then compared the effectiveness of some of the mitigations (`Disconnect` and `Adblock Plus`) in 2016 with their 2019 version. We also extended the comparison with `Privacy Badger` and `Adblock Plus` (enforced with *EasyList&EasyPrivacy*) in the 2019 scenario.

Flow Propagation

Fig. 5 shows the graph of *Access* obtained applying our rules on the Top 5 Alexa domains without any mitigation. While Fig. 6 shows the Venn diagrams obtained computing the *Knows* predicates of the model without any mitigation and with the `Disconnect` mitigation.

Lowest Tracking Coverage

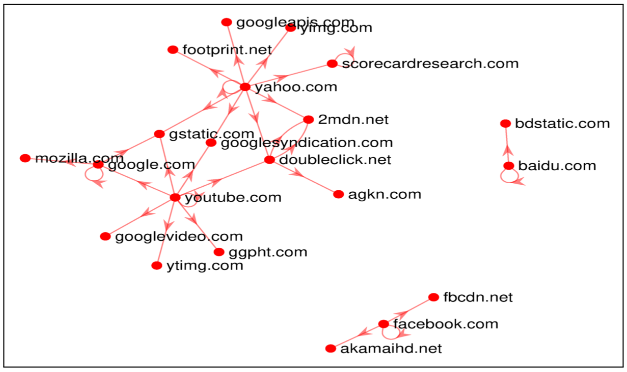
Fig. 6a shows the Venn diagrams for the Top 5 domains without any mitigation ($\mathcal{N}^*_{\mathcal{B}} = \emptyset$) while, Fig. 6b shows the Venn diagram with `Disconnect` mitigation ($\mathcal{N}^*_{\mathcal{A}} =$ `Disconnect`). From Def. 2 we have that $\mathcal{N}^*_{\mathcal{A}}$ is more effective than $\mathcal{N}^*_{\mathcal{B}}$.

Comparing Different Mitigations

We compared the effectiveness of the filter list of `Ghostery`, `Disconnect`, `Adblock Plus` (based on *EasyList*) in 2016 and `Disconnect`, `Adblock Plus` (based on *EasyList*), `Adblock Plus` (enforced

[9] Skewed towards tech-savvy users, thus these values are likely upper bounds.

[10] This feature is unable to block Google in certain situations. Firefox employs by default the Google search engine and, thus, establishes connections with Google domains if the website is not accessed directly through its domain name (we assume non-tech-savvy users behave in this way). As a result, Google domains (and all its subdomains) are whitelisted by Firefox and can bypass the third-party cookies block.

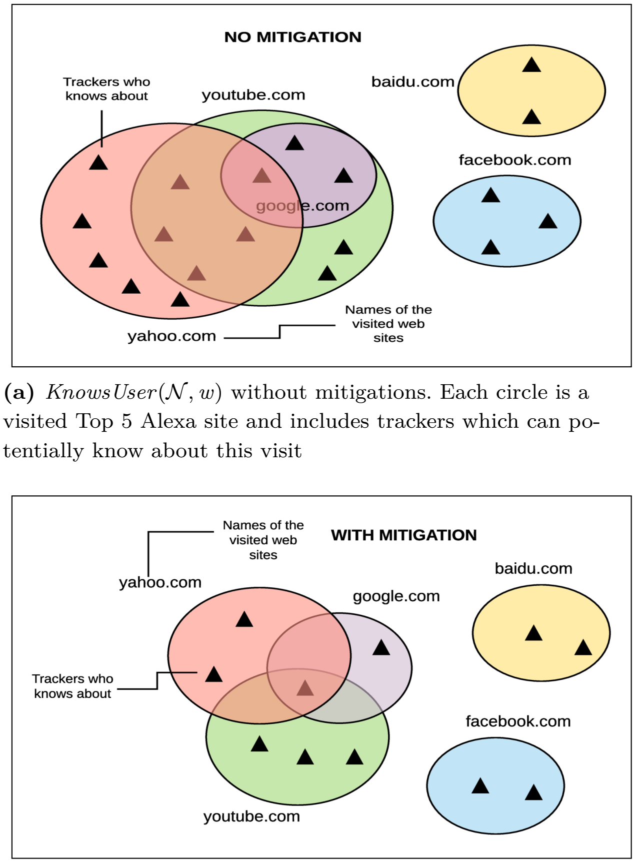


Access predicates obtained without any mitigation in the Top 5 Alexa domains. Several connections are made to different third-party domains. Understanding how many trackers can potentially know about your *youtube.com* visits is far from trivial (even ignoring any back-office sharing agreement).

Fig. 5. *Access* Graph Top 5 Alexa Domains

with *EasyList&EasyPrivacy*), and Privacy Badger ("trained" on the Top 200 Alexa domains in December 2019) in 2019 based on the Def. 3 presented in §6. We extracted the requests from the dataset and we recursively apply the filter lists of the different mitigations to the connections established for the Top 5, 10, 50, 100 Alexa domains. Except for Privacy Badger, that provides the list of domains it "learned" either to block completely ($Block\_request(w)$) or to stop setting the cookies ($Block\_tp\_cookie(w)$), for all the other mitigations we rely on their blacklist of domains, i.e. only $Block\_request(w)$. The results are normalized with respect to $\mathcal{N}$ without any mitigation.

We employed the filter lists from [3, 17] and the database previously presented to analyze the effectiveness of the mitigations in 2016. We then computed the current effectiveness of the filter list of Adblock Plus (with and without the addition of the *EasyPrivacy* list), Disconnect, and Privacy Badger in 2019 with an up-to-date database[11] from June 2019. Fig. 7 shows the comparison of the mitigations. The dashed line and the dash-dotted line correspond to two different efficiency levels. The first is a 1-for-1 drop: for each connection that the mitigation blocks, it blocks one tracker, while the second represents a 1-for-2 drop: for each connection that the mitigation blocks, it blocks two trackers. Fig. 7a shows that, among the filter lists in 2016, Disconnect is the most aggressive mitigation up to the Top 100 domains, where Ghostery behaves similarly. Adblock Plus is the most permissible mitigation in 2016. However, Adblock Plus shows a big increment of efficacy in its 2019 version. For example, in the Top 100, a 26% reduction of the accessed content generates a 66% decrement of trackers. It is worth mention that the filter list of Adblock Plus from [3] is also roughly 46 times smaller than the list in 2019 and that currently there is overlap between EasyList and EasyPrivacy [37]. In contrast, Disconnect does not significantly improve in 2019 with a more restrictive behavior. We found that the combination of EasyList and EasyPrivacy (EasyList&EasyPrivacy) achieves the highest protection at the cost of the most restrictive approach. Finally, Privacy Badger showed a similar level of protection compared to Disconnect and EasyList&EasyPrivacy but with a significantly higher number of connections allowed due to the balancing of blocking connections and cookies.

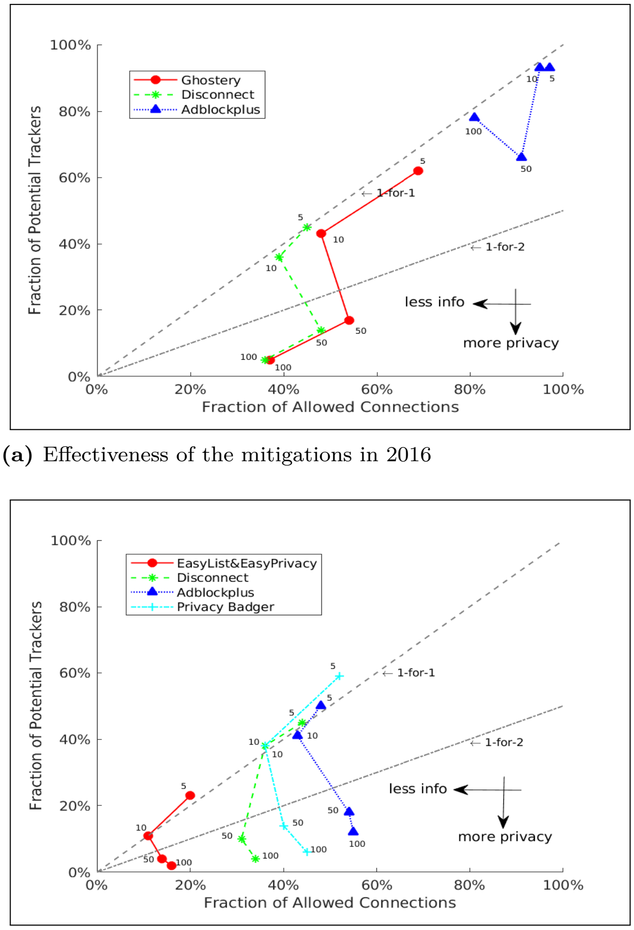


(a) $KnowsUser(\mathcal{N}, w)$ without mitigations. Each circle is a visited Top 5 Alexa site and includes trackers which can potentially know about this visit

(b) $KnowsUser(\mathcal{N}, w)$ with Disconnect mitigation. Disconnect significantly limits potential trackers when visiting *youtube.com* (from 9 to 4) and *yahoo.com* (from 9 to 3)

Fig. 6. Comparing Tracking Knowledge for Alexa Top 5.

[11] The database contains the same information of the 2016 database with small differences in the structure, for example, the 2019 version presents a table for the redirections.



(a) Effectiveness of the mitigations in 2016

(b) Effectiveness of the mitigations in 2019
Comparison between the fraction of potential trackers (counted as # of unique *Knows*) and the allowed connections (counted as # of unique *Access*) on Top 5, 10, 50, 100 domains with different mitigations. The most aggressive mitigations are `Disconnect` in 2016 and `EasyList&EasyPrivacy` in 2019. `Adblock Plus` significantly increased its efficacy in 2019, while `Disconnect` did not improve in terms of protection. `Privacy Badger` shows performance comparable to `Disconnect` in terms of trackers blocked with a less conservative approach.

Fig. 7. From 2016 to 2019 mitigations reduced the amount of access to gain privacy

9.2 Comparison of Effectiveness With Related Works

We compared our findings[12] with the results of [20] and [8]. Albeit our analysis is currently limited to the Top 100 domains, while [20] and [8] analyzed the top 10k and 200k domains respectively. It is still of interest to see if similar results can be obtained from a formal model and thus justifiable to third parties instead of a pure experiment[13]. Tab. 7 shows the comparison of the effectiveness of the mitigations in terms of percentage of allowed connections for different works. Fouad et al. [20] found that `Disconnect` and `EasyList&EasyPrivacy` are roughly comparable in terms of % of allowed requests, while we found that `EasyList&EasyPrivacy` is significantly more aggressive. However, we obtained similar values for `Disconnect` and `Adblock Plus`. `Privacy Badger` differs from [20] but it is similar to [8] probably due to the different size of the "training domains" used by the studies. Finally, as in [20] and [8], we confirmed that adblockers are less effective than trackers blockers.

Table 7. Comparison of % of Allowed Connections by `Adblock Plus` (AdB), `Disconnect` (D), `Privacy Badger` (PB), and `EasyPrivacy&EasyList` (EL&EP) With Previous Works

Paper	AdB	D	PB	EL&EP
Our work	54.9%	33.5%	44.8%	16.4%
[20]	≈ 60%	34.6%	30-35%	39.3%
[8]	65-70%	25-30%	≈ 40%	N.D.

10 Discussions and Limitations

The presented framework fills the gap between graph approaches, that scale but lack transparency of the results, and manual inspection, that is explainable but cannot scale even for few domains (see Tab. 5 where considering only the top 10 domains requires to analyze more than 900 interactions). For a user that wants to determine what is the best adds-on she should install on her browser, large scale studies ([8, 20]) provide contrasting and unverifiable claims (see Tab. 7). Our framework provides a more focused analysis compare to Internet-scale crawling and an *explanation* of why a given mitigation is better than another via a formal proof. The proof could then be traduced to natural language format [63] to hide the complexity to the user.

Our model relies on observable sharing behaviors from the client-side and it misses back-office information flows. While these mechanisms constitute a sizable part of the data-sharing economy they cannot be externally measured until legislators would oblige to divulge to which sites such information is shared. When this would be available the model could be extended by in-

[12] Here we limit our analysis for the 2019 database.

[13] The results of the previous works are obtained crawling the web with OpenWPM and the chosen mitigation.

cluding knows axioms for back-office data sharing. The same applies to cookie syncing, as well as the usage of cookie obtained from cookie forwarding, thus the results on cookie syncing represent a lower bound.

For the first approximation, we only focus on tracking based on cookies and we ignored other techniques (e.g. fingerprinting) for which we do not know how the process of data sharing is done and that it cannot be observed from the client-side.

The database is obtained from previous projects and it could be not representative of the current status of the Internet. However, our model can be applied to any new OpenWPM database. Still, the results apply to what it is possible to collect and observe using a crawler. The probabilistic proofs also strictly depend on the snapshots collected to determine proof of evolution (if snapshots are spaced in time) or of non-determinism [7] (if snapshots describe different requests to a website).

The filter lists of `Ghostery`, `Disconnect`, and `Adblock Plus`, obtained from [3], are domain blacklists. To maintain consistency in the comparison[14] with the 2019 version, we compared only the domain blacklist (for EasyList and EasyPrivacy we used the rules that begin with || and |).

The blacklist for `Disconnect` from [3] does not include any information about the *entity relationships*[15] that allows one to determine if a domain is loaded as first or third-party. We thus decided to not include this feature also in the 2019 evaluation. Future works can extend the analysis to consider the entity relationship. We did not produce a blacklist for `Ghostery` in the 2019 because it is proprietary software and does not provide access to the blacklist and the mechanisms implemented. We plan to compare more mitigations in the future.

The results obtained from the computation of the *Knows* predicate represent an upper bound of the *observable* data sharing scenario, it shows the number of websites that *potentially* can collect information about users' habits due to the presence of interactions among them. We highlight that this situation describes the *worst* possible scenario for the privacy of a user, in which websites intentionally propagate the information collected to other partners. This choice is more conservative than some current approaches that only consider the elements of a blacklist as the all and only trackers.

11 Conclusions and Future Work

We presented the first formal model to characterize tracking procedures based on data sharing. From the model, we extracted sharing relations to determine the tracking ecosystem, the effectiveness of different mitigations, and websites that should be COPPA compliant. We evaluated these properties on a real dataset (Top 100 Alexa sites) extracted from OpenWPM.

A tough question is whether the formal model bought us anything that we could not derive from running an alternative graph algorithm. From a pure computational complexity perspective, the answer is no: being both in PTIME a data-crunching procedure must exist that map the result of one into result of the other.

We argue that the difference is in the representation. A formal model produces *a result that can be independently checked* [64]. For example, one can produce a minimal derivation that shows that a website should be COPPA compliant and such derivation could be transformed (automatically) into a legal argument or a legal document. See [63] for a practical example where a formal logic model based on Datalog (also a PTIME inference framework) is used to reason about privacy practices and the results are presented to final customers (the local health authority) in a table or natural language format that is far easier to consume for them. It is important to underline that such proof is not by itself a *definitive* proof of COPPA violation but can be used by e.g. parents to trigger a first and more accurate investigation by appropriate agencies (e.g. FTC).

Future works can expand in several directions. Updated experimental results can be obtained by crawling again the internet with OpenWPM or improved algorithms to capture additional features. For example, the algorithm proposed by Fouad et al. [20], can be implemented to extract new data about cookie syncing. Another interesting extension would be to consider disjunctions in the mitigations for the rules. For example, in *3rdpartyTracking* one could include a disjunction on the (un)blocking of cookies or (dis)abling of scripts that are used for fingerprinting. We would also need to define fingerprint data-sharing agreements among parties. This might happen at the price of tractability. The model can be extended with mitigations that offer different fingerprints to different websites (assuming that websites know about the fingerprint via back-office agreements).

[14] EasyList and EasyPrivacy present a richer syntax, that allows one to block specific requests of scripts, etc.

[15] https://feeding.cloud.geek.nz/posts/how-tracking-protection-works-in-firefox/

12 Acknowledgement

We would like to express our gratitude to Prof. Nataliia Bielova and the anonymous reviewers for their thoughtful feedback. This research has been partly funded by the EU under the H2020 Programs H2020-EU.2.1.1-CyberSec4Europe (Grant No. 830929).

References

[1] A. Lerner, A. K. Simpson, T. Kohno, and F. Roesner, "Internet jones and the raiders of the lost trackers: An archaeological study of web tracking from 1996 to 2016," in *Proc. of USENIX-16*, 2016.

[2] P. Gill, V. Erramilli, A. Chaintreau, B. Krishnamurthy, K. Papagiannaki, and P. Rodriguez, "Follow the money: understanding economics of online aggregation and advertising," in *Proc. of IMC-13*, 2013.

[3] M. A. Bashir and C. Wilson, "Diffusion of user tracking data in the online advertising ecosystem," in *Proc. of PETS-18*, 2018.

[4] P. Papadopoulos, N. Kourtellis, and E. P. Markatos, "Cookie synchronization: Everything you always wanted to know but were afraid to ask," in *Proc. of WWW-19*, 2019.

[5] P. Eckersley, "How unique is your web browser?," in *Proc. of PETS-10*, 2010.

[6] V. Marotta, V. Abhishek, and A. Acquisti, "Online tracking and publishers' revenues: An empirical analysis," in *Proc. of WEIS-19*, 2019.

[7] F. Roesner, T. Kohno, and D. Wetherall, "Detecting and defending against third-party tracking on the web," in *Proc. of NSDI-12*, 2012.

[8] G. Merzdovnik, M. Huber, D. Buhov, N. Nikiforakis, S. Neuner, M. Schmiedecker, and E. R. Weippl, "Block me if you can: A large-scale study of tracker-blocking tools," in *Proc. of EuroS&P-17*, 2017.

[9] N. Nikiforakis, A. Kapravelos, W. Joosen, C. Kruegel, F. Piessens, and G. Vigna, "Cookieless monster: Exploring the ecosystem of web-based device fingerprinting," in *Proc. of SSP-13*, 2013.

[10] A. Vastel, P. Laperdrix, W. Rudametkin, and R. Rouvoy, "Fp-scanner: The privacy implications of browser fingerprint inconsistencies," in *Proc. of USENIX-18*, 2018.

[11] R. Binns, J. Zhao, M. V. Kleek, and N. Shadbolt, "Measuring third-party tracker power across web and mobile," *ACM-TOIT-18*, vol. 18, 2018.

[12] A. Acquisti, C. Taylor, and L. Wagman, "The economics of privacy," *Journal of Economic Literature*, vol. 54, 2016.

[13] J. S. Olson, J. Grudin, and E. Horvitz, "A study of preferences for sharing and privacy," in *Proc. of CHI-05*, 2005.

[14] A. Hang, E. von Zezschwitz, A. De Luca, and H. Hussmann, "Too much information! user attitudes towards smartphone sharing," in *Proc. of NordiCHI-12*, 2012.

[15] A. Ghosh, M. Mahdian, R. P. McAfee, and S. Vassilvitskii, "To match or not to match: Economics of cookie matching in online advertising," *ACM-TEAC-15*, vol. 3, 2015.

[16] R. Gomer, E. M. Rodrigues, N. Milic-Frayling, and M. Schraefel, "Network analysis of third party tracking: User exposure to tracking cookies through search," in *Proc. of WI-IAT-13*, 2013.

[17] M. A. Bashir, S. Arshad, W. K. Robertson, and C. Wilson, "Tracing information flows between ad exchanges using retargeted ads," in *Proc. of USENIX-16*, 2016.

[18] V. Kalavri, J. Blackburn, M. Varvello, and K. Papagiannaki, "Like a pack of wolves: Community structure of web trackers," in *Proc. of PAM-16*, 2016.

[19] S. Englehardt and A. Narayanan, "Online tracking: A 1-million-site measurement and analysis," in *Proc. of ACM-CCS-16*, 2016.

[20] I. Fouad, N. Bielova, A. Legout, and N. Sarafijanovic-Djukic, "Missed by filter lists: Detecting unknown third-party trackers with invisible pixels," in *Proc. of PETS-20*, 2020.

[21] L. Olejnik, M. Tran, and C. Castelluccia, "Selling off user privacy at auction," in *Proc. of NDSS-14*, 2014.

[22] G. Acar, M. Juárez, N. Nikiforakis, C. Díaz, S. F. Gürses, F. Piessens, and B. Preneel, "Fpdetective: dusting the web for fingerprinters," in *Proc. of ACM-CCS-13*, 2013.

[23] R. Bhoraskar, S. Han, J. Jeon, T. Azim, S. Chen, J. Jung, S. Nath, R. Wang, and D. Wetherall, "Brahmastra: Driving apps to test the security of third-party components," in *Proc. of USENIX-14*, 2014.

[24] A. Barth, A. Datta, J. C. Mitchell, and H. Nissenbaum, "Privacy and contextual integrity: Framework and applications," in *Proc. of SSP-06*, 2006.

[25] M. J. May, C. A. Gunter, and I. Lee, "Privacy apis: Access control techniques to analyze and verify legal privacy policies," in *Proc. of CSFW-19*, 2006.

[26] H. DeYoung, D. Garg, L. Jia, D. K. Kaynar, and A. Datta, "Experiences in the logical specification of the HIPAA and GLBA privacy laws," in *Proc. of WPES-10*, 2010.

[27] A. Cahn, S. Alfeld, P. Barford, and S. Muthukrishnan, "An empirical study of web cookies," in *Proc. of WWW-16*, 2016.

[28] P. Laperdrix, N. Bielova, B. Baudry, and G. Avoine, "Browser fingerprinting: A survey," *CoRR*, 2019.

[29] K. Mowery and H. Shacham, "Pixel perfect: Fingerprinting canvas in html5," in *Proc. of W2SP-12*, 2012.

[30] G. Acar, C. Eubank, S. Englehardt, M. Juárez, A. Narayanan, and C. Díaz, "The web never forgets: Persistent tracking mechanisms in the wild," in *Proc. of ACM-CCS-18*, 2014.

[31] I. Sánchez-Rola, I. Santos, and D. Balzarotti, "Clock around the clock: Time-based device fingerprinting," in *Proc. of ACM-CCS-18*, 2018.

[32] Y. Cao, S. Li, and E. Wijmans, "(cross-)browser fingerprinting via OS and hardware level features," in *Proc. of NDSS-17*, 2017.

[33] A. Soltani, S. Canty, Q. Mayo, L. Thomas, and C. J. Hoofnagle, "Flash cookies and privacy," in *Proc. of AAAI-10*, 2010.

[34] M. Ayenson, D. Wambach, A. Soltani, N. Good, and C. Hoofnagle, "Flash cookies and privacy ii: Now with html5 and etag respawning," *WWW Internet And Web Information Systems*, 2011.

[35] L. Olejnik, C. Castelluccia, and A. Janc, "Why johnny can't browse in peace: On the uniqueness of web browsing history

patterns,” in *Proc. of HotPETs-12*, 2012.

[36] A. Klein and B. Pinkas, “DNS cache-based user tracking,” in *Proc. of NDSS-19*, 2019.

[37] U. Iqbal, Z. Shafiq, P. Snyder, S. Zhu, Z. Qian, and B. Livshits, “Adgraph: A graph-based approach to ad and tracker blocking,” in *Proc. of SSP-20*, 2020.

[38] P. Speicher, M. Steinmetz, R. Künnemann, M. Simeonovski, G. Pellegrino, J. Hoffmann, and M. Backes, “Formally reasoning about the cost and efficacy of securing the email infrastructure,” in *Proc. of EuroS&P-18*, 2018.

[39] M. Simeonovski, G. Pellegrino, C. Rossow, and M. Backes, “Who controls the internet?: Analyzing global threats using property graph traversals,” in *Proc. of WWW-17*, 2017.

[40] “European Parliament Regulation 2016/679: General Data Protection Regulation,” 2016. https://eur-lex.europa.eu/eli/reg/2016/679/oj. Accessed: 2019-17-01.

[41] C. Matte, N. Bielova, and C. Santos, “Do cookie banners respect my choice? measuring legal compliance of banners from IAB europe’s transparency and consent framework,” in *Proc. of SSP-20*, 2020.

[42] “Federal Trade Commission: Children’s Online Privacy Protection Rule ("COPPA"),” 2013. https://www.ftc.gov/enforcement/rules/rulemaking-regulatory-reform-proceedings/childrens-online-privacy-protection-rule. Accessed: 2019-17-01.

[43] “Operators of online "virtual worlds" to pay $ 3 million to settle ftc charges that they illegally collected and disclosed children’s personal information,” 2011. https://www.ftc.gov/news-events/press-releases/2011/05/operators-online-virtual-worlds-pay-3-million-settle-ftc-charges. Accessed: 2019-17-01.

[44] “Google and youtube will pay record $ 170 million for alleged violations of children’s privacy law,” 2019. https://www.ftc.gov/news-events/press-releases/2019/09/google-youtube-will-pay-record-170-million-alleged-violations. Accessed: 2020-30-01.

[45] I. Reyes, P. Wijesekera, J. Reardon, A. E. B. On, A. Razaghpanah, N. Vallina-Rodriguez, and S. Egelman, “"won’t somebody think of the children?" examining COPPA compliance at scale,” in *Proc. of PETS-18*, 2018.

[46] I. Reyes, P. Wiesekera, A. Razaghpanah, J. Reardon, N. Vallina-Rodriguez, S. Egelman, and C. Kreibich, “’is our children’s apps learning?’ automatically detecting coppa violations,” in *Proc. of IEEE ConPro-17*, 2017.

[47] E. Sy, C. Burkert, H. Federrath, and M. Fischer, “A QUIC look at web tracking,” in *Proc. of PETS-19*, 2019.

[48] A. Gómez-Boix, P. Laperdrix, and B. Baudry, “Hiding in the crowd: an analysis of the effectiveness of browser fingerprinting at large scale,” in *Proc. of WWW-18*, 2018.

[49] A. Vastel, P. Laperdrix, W. Rudametkin, and R. Rouvoy, “FP-STALKER: tracking browser fingerprint evolutions,” in *Proc. of SSP-18*, 2018.

[50] B. Liu, A. Sheth, U. Weinsberg, J. Chandrashekar, and R. Govindan, “Adreveal: improving transparency into online targeted advertising,” in *Proc. of HotNets-XII*, 2013.

[51] D. Miller, “Hereditary harrop formulas and logic programming,” in *Proc. of the Eigth CLMPST*, 1987.

[52] R. Statman, “Intuitionistic propositional logic is polynomial-space complete,” *Theoretical Computer Science*, vol. 9, 1979.

[53] S. R. Buss and P. Pudlák, “On the computational content of intuitionistic propositional proofs,” *Annals of Pure and Applied Logic*, vol. 109, 2001.

[54] S. Buss and G. Mints, “The complexity of the disjunction and existential properties in intuitionistic logic,” *Annals of Pure and Applied Logic*, vol. 99, 1999.

[55] A. Goerdt, *Efficient interpolation for the intuitionistic sequent calculus*. Techn. Univ., Fak. für Informatik, 2000.

[56] T. Urban, M. Degeling, T. Holz, and N. Pohlmann, “Beyond the front page: Measuring third party dynamics in the field,” in *Proc. of WWW-20*, 2020.

[57] S. Ferson, V. Kreinovich, L. Grinzburg, D. Myers, and K. Sentz, “Constructing probability boxes and dempster-shafer structures,” tech. rep., Sandia National Lab., 2015.

[58] G. Ebner, S. Hetzl, G. Reis, M. Riener, S. Wolfsteiner, and S. Zivota, “System description: Gapt 2.0,” in *Proc. of IJCAR-16*, 2016.

[59] E. Gabriel, “Herbrand construction for automated intuitionistic theorem proving,” in *Proc. of FISP-18*, 2018.

[60] S. Bird, I. Segall, and M. Lopatka, “Replication: Why we still can’t browse in peace: On the uniqueness and reidentifiability of web browsing histories,” in *Proc. of SOUPS-20*, 2020.

[61] R. Gonzalez, L. Jiang, M. Ahmed, M. Marciel, R. Cuevas, H. Metwalley, and S. Niccolini, “The cookie recipe: Untangling the use of cookies in the wild,” in *Proc. of TMA-17*, 2017.

[62] J. C. Blanchette, S. Böhme, and L. C. Paulson, “Extending sledgehammer with SMT solvers,” *J. Autom. Reasoning*, vol. 51, 2013.

[63] M. Robol, E. Paja, M. Salnitri, and P. Giorgini, “Modeling and reasoning about privacy-consent requirements,” in *Proc. of PoEM*, 2018.

[64] G. C. Necula, “Proof-carrying code,” in *Proc. of POPL-97*, 1997.

[65] U. Iqbal, S. Englehardt, and Z. Shafiq, “Fingerprinting the fingerprinters: Learning to detect browser fingerprinting behaviors,” in *Proc. of SSP-20*, 2021.

[66] B. Krishnamurthy and C. E. Wills, “Privacy diffusion on the web: a longitudinal perspective,” in *Proc. of WWW-09*, 2009.

[67] P. Laperdrix, W. Rudametkin, and B. Baudry, “Beauty and the beast: Diverting modern web browsers to build unique browser fingerprints,” in *Proc. of SSP-16*, 2016.

[68] O. Starov and N. Nikiforakis, “XHOUND: quantifying the fingerprintability of browser extensions,” in *Proc. of SSP-17*, 2017.

[69] G. G. Gulyás, D. F. Somé, N. Bielova, and C. Castelluccia, “To extend or not to extend: On the uniqueness of browser extensions and web logins,” in *Proc. of WPES-18*, 2018.

[70] T. Yen, Y. Xie, F. Yu, R. P. Yu, and M. Abadi, “Host fingerprinting and tracking on the web: Privacy and security implications,” in *Proc. of NDSS-12*, 2012.

[71] G. Franken, T. van Goethem, and W. Joosen, “Who left open the cookie jar? A comprehensive evaluation of third-party cookie policies,” in *Proc. of USENIX-18*, 2018.

[72] J. R. Mayer and J. C. Mitchell, “Third-party web tracking: Policy and technology,” in *Proc. of SSP-12*, 2012.

[73] S. Englehardt, D. Reisman, C. Eubank, P. Zimmerman, J. R. Mayer, A. Narayanan, and E. W. Felten, “Cookies that give

you away: The surveillance implications of web tracking," in *Proc. of WWW-15*, 2015.

A Appendix

A.1 A Survey of the Related Works

Tab. 8 presents a list of papers that cover different aspects of the tracking ecosystem and their techniques.

A.2 Examples of Complex Tracking Interactions

Fig. 9a, 9b, and 10 show different cases of tracking. Fig. 9a describes the use of first-party cookies on a website. Fig. 9b and 10 describe the practice of tracking carried out by third-party websites. In the first case, the tracker is directly present on the website, while in the latter it is included by a third-party website. Fig. 11a describes the leaf of recursion where website w accesses itself because either it uses content from itself or it redirects to its content. Fig. 11b describes how to propagate user's information through websites. The browser is forced to load the resources from w, the resources used by w from w', and the resources used by w' from w''. We can also model particular cases where w shares cookies with w' ($Access_{cookie}(w, w')$), but w' does not propagate its cookies to w'' ($Link(w', w'')$). Thus, we obtain ($Access(w, w'')$). Fig. 12 illustrates how cookie syncing allows an attacker to track users on websites where it is not explicitly present.

A.3 Extended Rules With Uncertainty

Given N different snapshots we can apply the following rules for any Internet snapshot $\mathcal{N}_i$:

$$\frac{}{< \{\},\ldots,\{\},\ldots,\{\} >\Rightarrow A\ (0,1)}\ (NullAx)$$

$$\frac{< \mathcal{N}_1,\ldots,\{\},\ldots,\mathcal{N}_N >\Rightarrow A\ (a,b)}{< \mathcal{N}_1,\ldots,\{A\},\ldots,\mathcal{N}_N >\Rightarrow A\ (a+\frac{1}{N},b)}\ (Axiom)$$

$$\frac{\mathcal{N}_i \vdash A \quad < \mathcal{N}_1,\ldots,\mathcal{N}_i,\ldots,\mathcal{N}_N >\Rightarrow A\ (a,b)}{< \mathcal{N}_1,\ldots,\mathcal{N}_i \cup \{B\},\ldots,\mathcal{N}_N >\Rightarrow A\ (a,b)}\ (WL)$$

$$\frac{\mathcal{N}_i \vdash A \quad < \mathcal{N}_1,\ldots,\mathcal{N}_i \cup \{B\},\ldots,\mathcal{N}_N >\Rightarrow C\ (a,b)}{< \mathcal{N}_1,\ldots,\mathcal{N}_i \cup \{A \rightarrow B\},\ldots,\mathcal{N}_N >\Rightarrow C\ (a,b)}\ (\rightarrow L)$$

$$\frac{\mathcal{N}_i \vdash A \rightarrow B \quad < \mathcal{N}_1,\ldots,\mathcal{N}_i \cup \{A \rightarrow B\},\ldots,\mathcal{N}_N >\Rightarrow C\ (a,b)}{< \mathcal{N}_1,\ldots,\mathcal{N}_i,\ldots,\mathcal{N}_N >\Rightarrow C\ (a,b)}\ (DomAx)$$

where:

- *NullAx*: it is the base of the derivation. From the empty snapshots, we have a minimum and maximum likelihood of 0 and 1 respectively.
- *Axiom*: starting from a minimum likelihood of a and a maximum likelihood of b, if we can add to the empty snapshot $\mathcal{N}_i$ the predicate A, then the minimum likelihood increases by $\frac{1}{N}$.
- *WL*: if $\mathcal{N}_i \vdash A$ and we add a predicate to $\mathcal{N}_i$ then the minimum and maximum likelihood do not change.
- $\rightarrow L$: if $\mathcal{N}_i \vdash A$ and we have a minimum and maximum likelihood for C, we can explicit the predicate A and the likelihood do not change.
- *DomAx*: if $\mathcal{N}_i \vdash A \rightarrow B$ we can replace it and the likelihood do not change.

We represent formulas of the form $A \wedge B \rightarrow C$ as $A \rightarrow B \rightarrow C$, where one first derive A, then derive B and finally C. The proof in Fig. 15 shows the application of these rules to derive *Knows*(*revsci.net*, *qq.com*) for a snapshot $\mathcal{N}_i$ of the 17 Internet snapshots considered.

A.4 Slakje Proof Examples

We employ the *getLKProof* method in `Slakje` to generate a proof as a sequence of sequents. The proof can be visualized using the *prooftool* of GAPT, however, it is extremely verbose. Fig. 13 shows a fragment of the proof for *Knows*(*revsci.net*, *qq.com*) generated by `Slakje`, while Fig. 15 shows a compacted version[16]. Fig. 14 shows the proof for *req_COPPA*(*flashtalking.com*), where PII can be potentially collected.

A.5 Mapping of Data Into the Model

Fig. 8 shows a fragment of the `http_responses` table for the website *yahoo.com*, identified by the ID 5. The IDs 66465, 66468, 66472, 66473, and 28 identify URLs directed to the websites *scorecardresearch.com*,

[16] It was impossible to insert an example in the paper. The compacted proof contains the same information as the original.

Table 8. Works About Tracking

Paper	Description
Iqbal et al. [65]	Developed a ML tool that employs static and dynamic analysis to detect browser fingerprinting
Matte et al. [41]	Crawled 1426 websites with TCF banners to detect violation of GDPR. 54% of them commit violations
Iqbal et al. [37]	Developed a graph-based ML classifier to detect ads and trackers with 95% of accuracy.
Urban et al. [56]	Crawled the 10k domains to generate 3rd-party trees and analyze how cookies are employed
Laperdrix et al. [28]	Presented a survey about browser fingerprinting researches and the fingerprinting techniques
Sy et al. [47]	Described tracking via the QUIC transport protocol, browsers affected and possible countermeasures.
Krishnamurthy et al. [66]	Presented a longitudinal study on the diffusion of trackers in the Web in a period of roughly 3 years
Eckersley et al. [5]	Employ browser fingerprinting using HTTP protocol, JavaScript, and Flash API on privacy-aware users
Laperdrix et al. [67]	Collected 118,934 fingerprints w/ 17 features (e.g. HTML5 canvas) Analyzed also mobile devices
Gomez-Boix et al. [48]	Analyzed 2,067,942 fingerprints w/ 17 features and compared Panopticlick and AmIUnique
Vastel et al. [49]	Analyzed 98,598 browser fingerprints to study the evolution of fingerprints
Starov et al. [68]	Computed fingerprints from browser extensions based on side effects on DOM pages
Gulyás et al. [69]	Analyzed 16,393 users to study the effect of browser adds-on and web logins in fingerprints generation
Acar et al. [22]	Employed a framework to detect device fingerprinting based on font probing on the top 1M Alexa.
Nikiforakis et al. [9]	Analyzed fingerprinting libraries and measured the adoption in the top 10k Alexa domains.
Yen et al. [70]	Analyzed fingerprinting based on user-agent, IPs, and cookies. Studied cookie crunch and IP switching.
Englehardt et al. [19]	Developed OpenWPM to crawl the top 1M Alexa to detect trackers, evaluate mitigs. and cookie sync.
Roesner et al. [7]	Developed ShareMeNot. Evaluated 3rd-party and popus blocking, disable JS, and DNT.
Franken et al. [71]	Crawled Top 10k Alexa to detect bypass of cookie policies and anti-tracking techniques.
Soltani et al. [33]	Crawled the top 100 U.S. websites to study Flash cookies and cookie respawning.
Mayer et al. [72]	Described privacy problems of 3rd-party web tracking, its business models, and the tracking techniques.
Fouad et al. [20]	Classified Web tracking behaviors based on invisible pixels and showed browser extensions limitations.
Acar et al. [30]	Detected canvas fingerprint, analyze cookie respawn, and collected cookie sync.
Olejnik et al. [21]	Developed a plug-in to analyze RTB and cookie syncing and observed the prices paid to collect data.
Englehardt et al. [73]	Studied mass surveillance via passive eavesdroppers and cookies. Implemented a graph URL-Cookie.
Merzdovnik et al. [8]	Analyzed over 100,000 websites and evaluated browser extension in desktop and mobile devices.
Papadopoulos et al. [4]	Implemented a technique to detect encrypted and unencrypted cookie sharing in the mobile ecosystem.
Klein et al. [36]	Presented a DNS-based tracking technique that exploits combinations of A records to generate IDs.

```
id  | visit_id | url_id | method | response_status | location_id
----+----------+--------+--------+-----------------+-------------
250 |        5 |  66465 | GET    |             302 |       66468
259 |        5 |  66472 | GET    |             302 |       66473
...
246 |        5 |     28 | GET    |             200 |
...
254 |        5 |  66468 | GET    |             200 |
...
262 |        5 |  66473 | GET    |             200 |
...
```

(a) Fragment of the `http_responses` table for the websites *yahoo.com* extracted with the SQL query: `SELECT id, visit_id, url_id, method, response_status, location_id FROM http_responses WHERE visit_id=5;`

$\mathit{IncludeContent}(\mathit{yahoo.com}, \mathit{scorecard}\ldots.\mathit{com})$
$\mathit{IncludeContent}(\mathit{yahoo.com}, \mathit{doubleclick.net})$
...
$\mathit{IncludeContent}(\mathit{yahoo.com}, \mathit{yimg.com})$
...
$\mathit{Redirect}(\mathit{scorecard}\ldots.\mathit{com}, \mathit{scorecard}\ldots.\mathit{com})$
...
$\mathit{Redirect}(\mathit{doubleclick.net}, \mathit{agkn.com})$

(b) Predicates instantiated from the fragment of the table

Fig. 8. Mapping of the Table Entries in the Predicates of Our Model.

doubleclick.net, *agkn.com*, and *yimg.com*. The fragment in the table is mapped to the predicates of the model.

$$\dfrac{\mathit{Visits}(w) \qquad \dfrac{\mathit{Link}(w, w) \quad \neg\mathit{Block\_request}(w)}{\mathit{Access}(w, w)} \qquad \neg\mathit{Block\_tp\_cookie}(w)}{\mathit{Knows}(w, w)}$$

(a) If a user visits a website w that is allowed to store cookies, then w can know that the user visited it. This is a special case of `3rdpartyTracking` in Fig.1a. In this case $\neg\mathit{Block\_tp\_cookie}(w)$ is always true because there are not 3rd-party cookies.

$$\dfrac{\dfrac{\mathit{Link}(w, w') \quad \neg\mathit{Block\_request}(w')}{\mathit{Access}(w, w')} \qquad \mathit{Visits}(w) \qquad \neg\mathit{Block\_tp\_cookie}(w')}{\mathit{Knows}(w', w)}$$

(b) If a user visits a website w that forces to access resources from w', then if the website w' is not blocked by any mitigation, it can know that the user visited w.

Fig. 9. Knows Visits Derivations

$$\dfrac{\mathit{Visits}(w) \qquad \dfrac{\dfrac{\mathit{Link}(w, w') \quad \neg\mathit{Block\_request}(w')}{\mathit{Access}(w, w')} \quad \dfrac{\mathit{Link}(w', w'') \quad \neg\mathit{Block\_request}(w'')}{\mathit{Access}(w', w'')}}{\mathit{Access}(w, w'')} \qquad \neg\mathit{Block\_tp\_cookie}(w'')}{\mathit{Knows}(w'', w)}$$

If a user visits a website w that accesses resources from a 3rd-party website w', the website may not only track the user but it can also redirect (include) another website w'' that can set its cookie if no mitigation blocks it. This situation describes both *Third parties that include trackers* and *Basic tracking initiated by a tracker* [20], where w' tracks/does not track w (it can be verified with the rule `3rdpartyTracking`).

Fig. 10. Knows by External Trackers

$$\dfrac{\mathit{Link}(w, w) \quad \neg\mathit{Block\_request}(w)}{\mathit{Access}(w, w)} \qquad \dfrac{\mathit{Link}_{\mathit{cookie}}(w, w) \quad \neg\mathit{Block\_request}(w)}{\mathit{Access}_{\mathit{cookie}}(w, w)}$$

(a) It is the leaf of the derivation corresponding to the access of a sequence of resources.

$$\dfrac{\dfrac{\mathit{Link}(w, w') \quad \neg\mathit{Block\_request}(w')}{\mathit{Access}(w, w')} \qquad \mathit{Link}(w', w'') \quad \neg\mathit{Block\_request}(w'')}{\mathit{Access}(w, w'')}$$

(b) If a website w accesses resources of website w', and w' has a link to content from w'', then there is an access between w and w'' only if w'' is not blocked by any extension. We can also use $\mathit{Access}_{\mathit{cookie}}(w, w')$ and $\mathit{Link}_{\mathit{cookie}}(w', w'')$ to describe a link with exchange of cookies between w and w'' ($\mathit{Link}_{\mathit{cookie}}(w, w'')$).

Fig. 11. Network Interactions Derivations

$$\dfrac{\mathit{Knows}(w', w) \qquad \dfrac{\dfrac{\mathit{Link}_{\mathit{cookie}}(w', w'') \quad \neg\mathit{Block\_request}(w'')}{\mathit{Access}_{\mathit{cookie}}(w', w'')} \qquad \neg\mathit{Block\_tp\_cookie}(w'')}{\mathit{Cookie\_sync}(w', w'')}}{\mathit{Knows}(w'', w)}$$

w' can track a user on w and redirects the user to another 3rd-party website w'' inserting the cookie information. The two 3rd-party websites can share their cookies and w'' can track users on w even if it is not directly embedded. This situation can be mitigated if either w' or w'' are either blocked by an extension or cannot set cookies. This is called *3rd-2-3rd party cookie syncing.*

Fig. 12. Known by External Trackers via Cookie Syncing

```
gapt> val proof = Slakje.getLKProof(problem)
proof: Option(gapt.proofs.lk.LKProof) =
Some((p19) ∀ W ∀ W1
(#c(visit: i > o)(W) ∧ access(W, W1) ∧
¬ block_tp_cookie(W1) → knows(W1, W)),
#c(visit: i > o)('qq.com'),
∀ W ∀ W1 (link(W, W1) ∧
¬ block_requests(W1) → access(W, W1)),
∀ W ∀ W1 (includeContent(W, W1)→
link(W, W1)),
includeContent('qq.com', 'revsci.net'),
¬ block_requests('revsci.net'),
¬ block_tp_cookie('revsci.net')
⊢
knows('revsci.net', 'qq.com')
(ForallLeftRule(p18, Ant(0), ∀ W1
(#c(visit: i>o)(W) ∧ access(W, W1) ∧
¬ block_tp_cookie(W1) → knows(W1, W))...
```

Fig. 13. Fragment of the GAPT Output Using `Slakje` for $\mathit{Knows}(\mathit{revsci.net}, \mathit{qq.com})$

$$
\dfrac{
  \dfrac{
    \dfrac{
      \dfrac{
        \dfrac{
          \dfrac{
            \dfrac{
              \dfrac{
                \dfrac{
                  \dfrac{}{\mathit{includeContent}(\mathit{thes}, \mathit{flt}) \vdash \mathit{includeContent}(\mathit{thes}, \mathit{flt})}\,\mathit{ax}
                  \quad
                  \dfrac{}{\mathit{link}(\mathit{thes}, \mathit{flt}) \vdash \mathit{link}(\mathit{thes}, \mathit{flt})}\,\mathit{ax}
                }{\mathit{includeContent}(\mathit{thes}, \mathit{flt}) \vdash \mathit{link}(\mathit{thes}, \mathit{flt})}\,\text{ß} : l, \forall l, \forall l, \mathit{IncludeW}
                \quad
                \dfrac{}{\neg \mathit{block\_requests}(\mathit{flt}) \vdash \neg \mathit{block\_requests}(\mathit{flt})}\,\mathit{ax}
              }{\mathit{includeContent}(\mathit{thes}, \mathit{flt}), \neg \mathit{block\_requests}(\mathit{flt}) \vdash \mathit{link}(\mathit{thes}, \mathit{flt}) \wedge \neg \mathit{block\_requests}(\mathit{flt})}\,\wedge : R
              \quad
              \dfrac{}{\mathit{access}(\mathit{thes}, \mathit{flt}) \vdash \mathit{access}(\mathit{thes}, \mathit{flt})}\,\mathit{ax}
            }{\mathit{includeContent}(\mathit{thes}, \mathit{flt}), \neg \mathit{block\_requests}(\mathit{flt}) \vdash \mathit{access}(\mathit{thes}, \mathit{flt})}\,\text{ß} : l, \forall l, \forall l, \mathit{AccessToW}
            \quad
            \dfrac{}{\mathit{visit}(\mathit{thes}) \vdash \mathit{visit}(\mathit{thes})}\,\mathit{ax}
          }{\mathit{visit}(\mathit{thes}), \mathit{includeContent}(\mathit{thes}, \mathit{flt}), \neg \mathit{block\_requests}(\mathit{flt}) \vdash \mathit{visit}(\mathit{thes}) \wedge \mathit{access}(\mathit{thes}, \mathit{flt})}\,\wedge : R
          \quad
          \dfrac{}{\neg \mathit{block\_tp\_cookie}(\mathit{flt}) \vdash \neg \mathit{block\_tp\_cookie}(\mathit{flt})}\,\mathit{ax}
        }{\mathit{visit}(\mathit{thes}), \mathit{includeContent}(\mathit{thes}, \mathit{flt}), \neg \mathit{block\_requests}(\mathit{flt}), \neg \mathit{block\_tp\_cookie}(\mathit{flt}) \vdash (\mathit{visit}(\mathit{thes}) \wedge \mathit{access}(\mathit{thes}, \mathit{flt})) \wedge \neg \mathit{block\_tp\_cookie}(\mathit{flt})}\,\wedge : R
        \quad
        \dfrac{}{\mathit{knows}(\mathit{flt}, \mathit{thes}) \vdash \mathit{knows}(\mathit{flt}, \mathit{thes})}\,\mathit{ax}
      }{\mathit{visit}(\mathit{thes}), \mathit{includeContent}(\mathit{thes}, \mathit{flt}), \neg \mathit{block\_requests}(\mathit{flt}), \neg \mathit{block\_tp\_cookie}(\mathit{flt}) \vdash \mathit{knows}(\mathit{flt}, \mathit{thes})}\,\text{ß} : l, \forall : l, \forall : l, \mathit{3rdpartyTracking}
      \quad
      \dfrac{}{\mathit{kids}(\mathit{thes}) \vdash \mathit{kids}(\mathit{thes})}\,\mathit{ax}
    }{\mathit{visit}(\mathit{thes}), \mathit{includeContent}(\mathit{thes}, \mathit{flt}), \neg \mathit{block\_requests}(\mathit{flt}), \neg \mathit{block\_tp\_cookie}(\mathit{flt}) \vdash \mathit{knows}(\mathit{flt}, \mathit{thes}) \wedge \mathit{kids}(\mathit{thes})}\,\wedge : R
    \quad
    \dfrac{}{\mathit{req\_coppa}(\mathit{flt}) \vdash \mathit{req\_coppa}(\mathit{flt})}\,\mathit{ax}
  }{\mathit{visit}(\mathit{thes}), \mathit{includeContent}(\mathit{thes}, \mathit{flt}), \neg \mathit{block\_requests}(\mathit{flt}), \neg \mathit{block\_tp\_cookie}(\mathit{flt}), \mathit{kids}(\mathit{thes}) \vdash \mathit{req\_coppa}(\mathit{flt})}\,\text{ß} : l, \forall : l, \forall : l, \mathit{COPPAcomplColl}
}{}
$$

Fig. 14. Proof of $\mathit{req\_COPPA}(\mathit{flt} = \mathit{flashtalking.com})$, where thes is the children-related website *thesaurus.com*

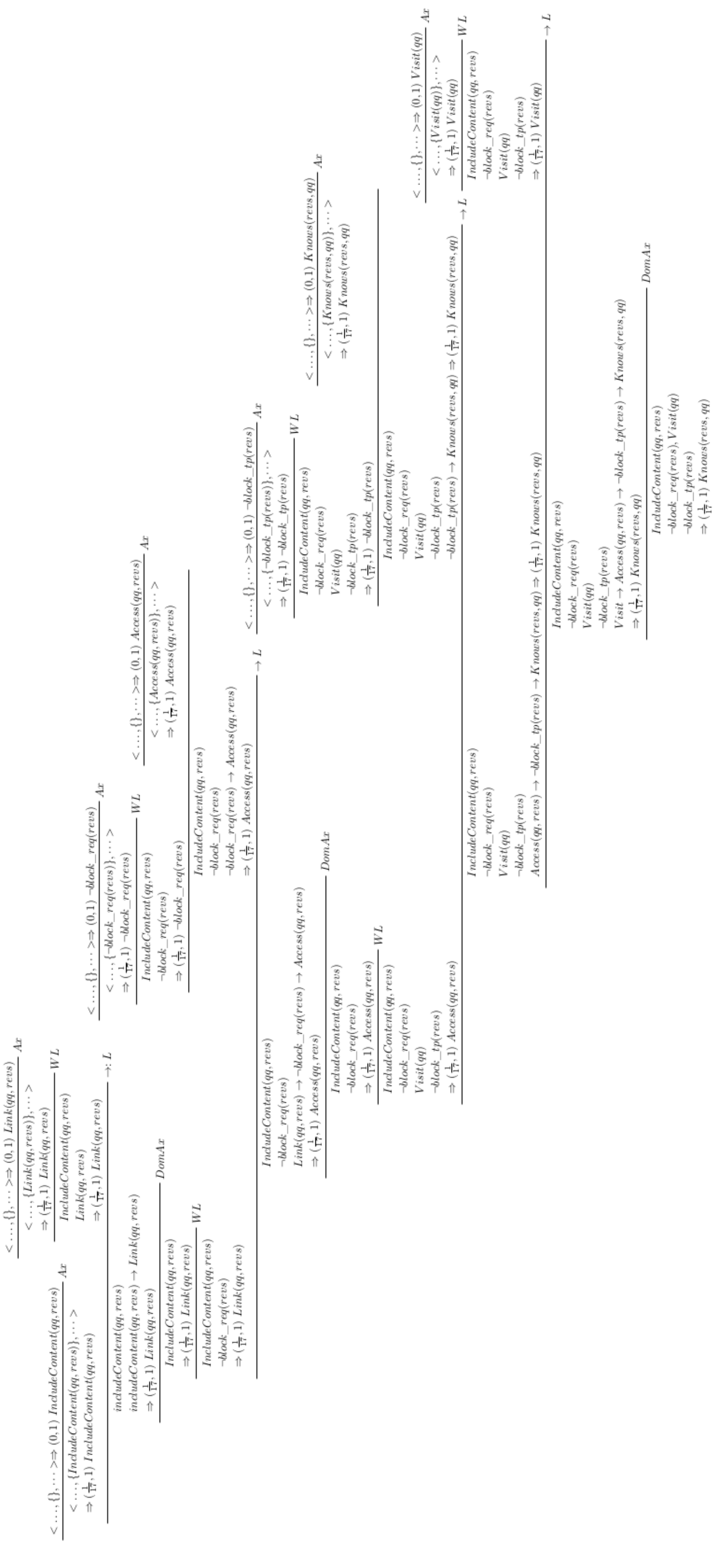

Ax
WL
DomAx
→ L
$< \ldots, \{\}, \ldots > \Rightarrow (0,1)\ IncludeContent(qq, revs)$
$< \ldots, \{\}, \ldots > \Rightarrow (0,1)\ Link(qq, revs)$
$< \ldots, \{\}, \ldots > \Rightarrow (0,1)\ \neg block\_req(revs)$
$< \ldots, \{\}, \ldots > \Rightarrow (0,1)\ Access(qq, revs)$
$< \ldots, \{\}, \ldots > \Rightarrow (0,1)\ \neg block\_tp(revs)$
$< \ldots, \{\}, \ldots > \Rightarrow (0,1)\ Knows(revs, qq)$
$< \ldots, \{\}, \ldots > \Rightarrow (0,1)\ Visit(qq)$
$\Rightarrow (\frac{1}{17}, 1)\ Knows(revs, qq)$

Fig. 15. Proof of $Knows(revs = revsci.net, qq = qq.com)$ via $IncludeContent(qq, revs)$ for a snapshot $\mathcal{N}_i$.