Protecting Privacy of Web Users

Nataliia Bielova

UNIVERSITÉ CÔTE D'AZUR

ÉCOLE DOCTORALE
SCIENCES ET TECHNOLOGIES DE L'INFORMATION ET DE LA COMMUNICATION

Protecting Privacy of Web Users

Mémoire de synthése pour l'obtention d'une

Habilitation à Diriger les Recherches

par

Nataliia Bielova

soutenue le 7 Juin 2021

Rapporteurs:

Prof. Andrei Sabelfeld	Chalmers University of Technology, Sweden
Prof. Simone Fischer-Hübner	Karlstads University, Sweden
Prof. Thorsten Holz	Ruhr-University Bochum, Germany

Examinateurs:

Dr. Gwendal Le Grand	CNIL, France
Dr. Jaap-Henk Hoepman	Radboud University Nijmegen, The Netherlands
Dr. Serge Egelman	University of California, Berkeley, USA
Dr. Walid Dabbous	Inria, France

Contents

Acknowledgements

I would like first to thank Frank Piessens for raising my interest in the Web security at the end of my PhD back in 2011 and making me realise I can do anything if I work hard and when I am truly motivated. Big thank you goes to Frédéric Besson and Thomas Jensen for helping me advance in the new field of Secure Information Flow and sponsor me to attend my first IEEE Security & Privacy in 2012, where I have met the amazing (back thank a PhD student) Franzi Roesner who was presenting her new NDSS'12 paper there in a poster session. I immediately got interested in her work and in Web Tracking Technologies and Ashkan Solani's investigations into cookie respawning only heated up my curiosity.

I would like to thank my colleagues who welcomed me in the first Inria team where I started working as a Tenured Researcher – Tamara Rezk, Ilaria Castellani, Manuel Serrano and Bertrand Serpette – they quickly became not only colleagues but good friends. Big thank you also goes to the ANR for awarding me the ANR JCJC PrivaWeb project funding – this gave me the freedom to explore a completely new area of Web Privacy, combining technical and legal research. Also big thanks to Inria and its ecosystem, where a freedom to explore and start completely new academic fields is highly appreciated.

Thank you goes to Guillene Ribiere, whose questions made me realise how much I'm interested in the legal and practical aspects of privacy. I also thank CPDP 2017 organizers, moderators and speakers that motivated me to start exploring GDPR and ePrivacy directive in depth. Big thank you goes to Gaetan Goldberg, Paul-Olivier Dehaye, Frederik Zuiderveen Borgesius, Anna Buchta and Max Schrems, who made me realise how valuable my research is for the legal community beyond academia. Thank you to Vincent Toubiana, Estelle Hary, Armand Heslot, and Francois Pelligrini from the CNIL for very enlightening discussions.

A special thank you goes to Cedric Lauradoux who helped me believe in the transdicsciplinary research and for explaining the difference between a regulation and a directive with a pizza example! To my colleagues in the Inria Privatics team and other collaborators over the years: Gábor Gulyás, Claude Castelluccia, Arnaud Legout, Vincent Roca, Mathieu Cunche, Daniel Le Metayer, Antoine Boutet. To all the students and postdocs who have worked with me for the fascinating discussions and brilliant ideas we have generated together!

An enormous thank you goes to Cristiana Santos whose curiously has driven our legal and technical transdisciplinary research - Cristiana offered her collabo-

ration with me back in 2018 and have never left me since :-) Big thank you goes to Colin M. Gray who showed me a new research field of Human-Computer Interaction (HCI) and design and with whom, together with Cristiana Santos, new transdisciplinary results across Privacy, HCI and Law saw the light in a difficult year of 2020.

Thank you to my French and International friends Antitza, Tania, Jenya, Julie, Nicolas, Anna, Asun, Nadya, and Yan for their support and friendship over the years that helped me feel happy and motivated in everyday life. To my parents and my sister for their continuous support. To my husband Marco for his infinite love and to our little Christine who grows way too fast and surprises us every day.

Thank you all!

My research path: 2012 - 2020

In this HDR manuscript, I describe a 9-year journey I took in my research since obtaining my PhD in November 2011 until the end of 2020. This research has started during my postdoc at Inria Rennes in 2012, I have then been employed as *Chargée de Recherche* at Inria Sophia Antipolis in 2013, and had one year career break due to maternity. Since April 2020, I joined PRIVATICS[1] team of Inria Grenoble while still conducting my research at Inria Sophia Antipolis.

My research during 2012 - 2020 covers a number of areas in the broad field of security and privacy in Computer Science. Following my PhD, where I proved security properties of runtime monitors [B12], I have switched to the domain of language-based security and information flow control to build monitors for programs and prove their properties. My internship with Frank Piessens (KULeuven) in 2010 made me curious about security of Web browsers and Web applications. However, the crucial point in my career was attending the IEEE Security & Privacy Symposium (S&P) in 2012, where Jonathan Mayer presented his survey paper on Web Tracking technologies [MM12]. At S&P'12, I also discussed with Franzisca Roesner next to her poster about their recent work on detection of cookie-based tracking [RKW12]. Since 2012, I could no longer do only theoretical computer science, but started to work on privacy protection of Web users, and was more and more curious about the legal implications of tracking technologies.

In 2018, I obtained my personal grant via ANR JCJC PrivaWEB project[2] and was extremely lucky to meet Cristiana Santos, a researcher in Law, who decided to work with me. Thanks to these two events, we have started a interdisciplinary research on the legal gaps in the interpretation of law when applied to technical analysis of privacy in Web applications. The research on Web tracking and legal compliance of existing Web applications became the main passion of my research which is my current priority and main interest.

Therefore, in this HDR manuscript first describes my work on Web security and rather theoretical research on information flow control, then more space is dedicated to the interdisciplinary domain of privacy on the Web. I wrote this HDR manuscript to be accessible to a wider audience, so I summarize my contributions here at a very high level – all the technical and legal details can be found in the published papers. The complete list of my publications can be found in the overview chapter 1.

[1] https://team.inria.fr/privatics/
[2] https://project.inria.fr/privaweb/

Chapter 1

Overview

This HDR manuscript is organised as follows. Each of the follow-up chapters introduces the concept or a problem in a studied domain first, followed by the "dream" – that is, the desired state of the world – and the "reality" – underlying the problems in the area at a high level. The contributions in the research area are then presented, and followed by open challenges that still need to be addressed. In the text, citations to my own work since 2012 appear in plain style as in [19] whereas citations to other works appear in alpha style as in [AEE+14b].

Chapter 2: Web Application Security describes early works that started immediately after my PhD in 2012, and which we continued in 2016-2017. This chapter covers the following publications:

> [1] N. Bielova. Survey on JavaScript security policies and their enforcement mechanisms in a web browser. *Special Issue on Automated Specification and Verification of Web Systems of Journal of Logic and Algebraic Programming (JLAP)*, 82(8):243 – 262, 2013. https://doi.org/10.1016/j.jlap.2013.05.001
>
> [2] D. F. Somé, N. Bielova, and T. Rezk. On the content security policy violations due to the same-origin policy. In *Proceedings of the 26th International Conference on World Wide Web, (WWW 2017)*, pages 877–886. ACM, 2017. https://hal.inria.fr/hal-01649526

Chapter 3: Information Flow Monitors describes research we carried out between 2012 and 2017, both during my postdoc at Inria Rennes and in my early career in Inria Sophia Antipolis. This chapter covers the following publications:

> [3] F. Besson, N. Bielova, and T. Jensen. Hybrid information flow monitoring against web tracking. In *IEEE Computer Security Foundations Symposium (CSF 2013)*, pages 240–254. IEEE, 2013. https://hal.inria.fr/hal-00924138
>
> [4] F. Besson, N. Bielova, and T. Jensen. Browser Randomisation against Fingerprinting: A Quantitative Information Flow Approach.

In *Nordic Conference on Secure IT Systems (NordSec 2014)*, pages 181–196, 2014. doi: 10.1007/978-3-319-11599-3_11. https://hal.inria.fr/hal-01081037

[5] N. Bielova and T. Rezk. Spot the difference: Secure multi-execution and multiple facets. In *21st European Symposium on Research in Computer Security, ESORICS*, volume 9878 of *Lecture Notes in Computer Science*, pages 501–519. Springer, 2016. https://hal.inria.fr/hal-01348192

[6] F. Besson, N. Bielova, and T. P. Jensen. Hybrid monitoring of attacker knowledge. In *IEEE 29th Computer Security Foundations Symposium, (CSF'16)*, pages 225–238. IEEE Computer Society, 2016. https://hal.inria.fr/hal-01310572

[7] N. Bielova and T. Rezk. A taxonomy of information flow monitors. In *Principles of Security and Trust - 5th International Conference, POST*, volume 9635 of *Lecture Notes in Computer Science*, pages 46–67. Springer, 2016. https://hal.inria.fr/hal-01348188

[8] N. Bielova. Short paper: Dynamic leakage: A need for a new quantitative information flow measure. In *Proceedings of the 2016 ACM Workshop on Programming Languages and Analysis for Security, PLAS@CCS 2016*, pages 83–88. ACM, 2016. https://hal.inria.fr/hal-01409706

[9] M. Ngo, N. Bielova, C. Flanagan, T. Rezk, A. Russo, and T. Schmitz. A better facet of dynamic information flow control. In *Web Programming, Design, Analysis, And Implementation (WPDAI 2018) alternate track of The Web Conference (WWW 2018)*, pages 731–739. ACM, 2018. https://hal.inria.fr/hal-01723723

Chapter 4: Web Tracking Technologies covers works on measurement, detection and protection of Web users from online tracking carried out since 2014, and actively studied since 2018 until today. This chapter is based on the following publications:

[10] T. Vissers, N. Nikiforakis, N. Bielova, and W. Joosen. Crying Wolf? On the Price Discrimination of Online Airline Tickets. In *7th Workshop on Hot Topics in Privacy Enhancing Technologies (HotPETs 2014)*, 2014. https://hal.inria.fr/hal-01081034

[11] D. F. Somé, N. Bielova, and T. Rezk. Control what you include! server-side protection against third party web tracking. In *International Symposium on Engineering Secure Software and Systems*

(ESSoS), volume 10379 of *Lecture Notes in Computer Science*, pages 115–132. Springer, 2017. https://hal.inria.fr/hal-01649547

[12] 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 *Proceedings of the 2018 Workshop on Privacy in the Electronic Society (WPES) at ACM CCS'18*, pages 14–27. ACM, 2018. https://hal.inria.fr/hal-01921863

[13] P. Laperdrix, N. Bielova, B. Baudry, and G. Avoine. Browser fingerprinting: A survey. *ACM Transactions on the Web (TWEB)*, 14(2):8:1–8:33, 2020. https://dl.acm.org/doi/10.1145/3386040

[14] I. Fouad, N. Bielova, A. Legout, and N. Sarafijanovic-Djukic. Missed by filter lists: Detecting unknown third-party trackers with invisible pixels. *Proceedings on Privacy Enhancing Technologies (PoPETs)*, 2020, 2020. https://doi.org/10.2478/popets-2020-0038

Chapter 5: GDPR and ePrivacy Compliance covers the last 2 years of research, where thanks to multiple fruitful collaborations with researchers in Law and Design, we have made a number of interdisciplinary studies. This chapter covers the following publications:

[15] C. Boniface, I. Fouad, N. Bielova, C. Lauradoux, and C. Santos. Security analysis of subject access request procedures - how to authenticate data subjects safely when they request for their data. In *7th Annual Privacy Forum, APF*, volume 11498 of *Lecture Notes in Computer Science*, pages 182–209. Springer, 2019. https://hal.inria.fr/hal-02072302

[16] 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 *IEEE Symposium on Security and Privacy (IEEE S&P)*, 2020. https://hal.inria.fr/hal-03117294

[17] C. Matte, C. Santos, and N. Bielova. Purposes in IAB Europe's TCF: which legal basis and how are they used by advertisers? In *Annual Privacy Forum, APF*, Lecture Notes in Computer Science, 2020. https://hal.inria.fr/hal-02566891

[18] I. Fouad, C. Santos, F. Al Kassar, N. Bielova, and S. Calzavara. On Compliance of Cookie Purposes with the Purpose Specification Principle. In *2020 International Workshop on Privacy Engineering, IWPE*, 2020. https://hal.inria.fr/hal-02567022

[19] C. Santos, N. Bielova, and C. Matte. Are cookie banners indeed compliant with the law? deciphering EU legal requirements on consent and technical means to verify compliance of cookie banners. *Technology and Regulation (TechReg)*, pages 91–135, 2020. https://doi.org/10.26116/techreg.2020.009

[20]

[21] C. Santos, M. Nouwens, M. Toth, N. Bielova, and V. Roca. Consent management platforms under the gdpr: processors and/or controllers? In *Annual Privacy Forum (APF'21)*, 2021. Accepted for publication, https://hal.inria.fr/hal-03169436

Feedback to DPAs and EDPB. Additionally, we have submitted several opinions and feedback to public consultation based on our research to several EU Data Protection Authorities (DPAs) and European Data Protection Board (EDPB):

[22] M. Toth, N. Bielova, C. Santos, V. Roca, and C. Matte. Contribution to the public consultation on the CNIL's draft recommendation on "cookies and other trackers", 2020. Research report, https://hal.inria.fr/hal-02490531

[23] N. Bielova and C. Santos. Feedback to EDPB regarding Guidelines 07/2020 on the concepts of controller and processor in the GDPR, 2020. Research report, http://www-sop.inria.fr/members/Nataliia.Bielova/opinions/EDPB-contribution-controllers-processors.pdf

[24] N. Bielova and C. Santos. Feedback to the Guidelines on the use of cookies and other tracking tools of the Italian Data Protection Authority, 2020. Research report, https://hal.inria.fr/hal-03079482

Chapter 2

Web Application Security

Web applications today provide a variety of services to its users. The Web browser has become a virtual machine that received and executes various interactive applications from different stakeholders. These Web applications today implement complex interactions between multiple types of code and data, often directly or indirectly obtained from the Web users.

The Dream While Web browsers are a gateway between multiple stakeholders whose code composes a Web application and end users, browsers should provide proper protection mechanisms to ensure that (i) different sources of Web application cannot interfere with each other in non-authorised ways; (ii) the secret data of the Web users are adequately protected from the Web attacks.

The Reality Web applications are complex because they consists of multiple sources of code [NIK+12]. Every source can have its own security policy [RBC+20], and other security mechanisms are enforced directly in the Web browser. Researchers have been proposing solutions to monitor the execution of JavaScript code directly in the browser [HS12, 1] and security policies a browser should implement by default. In reality, proposed research solutions are often partial and don't provide well-defined security guarantees, while browser security policies are inconsistent with each other [2, CUT+21], thus making Web applications vulnerable to Web attacks.

2.1 Security Enforcement of JavaScript applications

JavaScript is one of the most used scripting languages on Web. JavaScript code is most often fetched from a remote server: there were 24.48% new inclusions in 2001, and this number grew to 45.46% in 2010 [NIK+12]. When a remote JavaScript code is included into a web page, it gets the same privileges as any other code inlined in the page. These privileges give JavaScript code the power to perform malicious actions, violating the user's privacy and security. For example, malicious scripts can access the secret user data on the web page and send it to

remote servers, or hijack the user's session and perform requests on behalf of the Web user.

Prior research developed verification techniques for JavaScript programs that answer the question *"Does your program comply with the security policy?"*, while more recent security runtime enforcement techniques have been developed - such techniques do not answer the above question but decide whether a given execution of the program satisfies the policy, and can also *fix* [BLW05, BM11] it in case of non-compliance[1]. The first step in securing JavaScript applications consists in identifying *the sources* of information that must be protected. In case of Web applications, these sources in different research papers have been identified via core DOM objects [SMWL10] and HTML5 APIs [W3C12] accessible to scripts. Two groups of techniques have been developed to protect security-critical objects and APIs: runtime monitoring and information flow control.

The need for systematisation of knowledge. *Dynamic techniques based on runtime monitoring* [HV05, YCIS07, DdSC09, ML10] observe the program execution and check whether this execution satisfies the security policies in question. These techniques are known to enforce a class of security policies that are based on a single program execution. *Secure information flow control techniques* [VNJ+07, CMJL09, LZW10, AF12, HS12] propose program analysis (either static, dynamic or hybrid - see further works in Section 3.1) to find the flows of information inside the program. A particular definition of security policy enforced by these techniques is *non-interference*. It states that no secret inputs to the program can influence publicly observed outputs. Since it is not possible to detect such information flows by observing only one program execution, the definition of non-interference is based on two program executions.

While these groups of techniques often implemented in practice via modifications of JavaScript engine, using code rewriting or with browser extensions, these works have not been compared against each other with respect to the security properties they enforce and formal security guarantees they provide. Each state-of-the art literature provides a new technique without rigorously comparing it to recent works, thus leaving the reader doubtful as to the formal security guarantees each technique provides.

Survey on JavaScript security. In a single-authored survey [1], we analyse and compare the research literature on security enforcement for JavaScript programs by mapping the works into two groups: runtime enforcement and information flow control. Table 2.1 presents a high level overview of the comparison done in the survey.

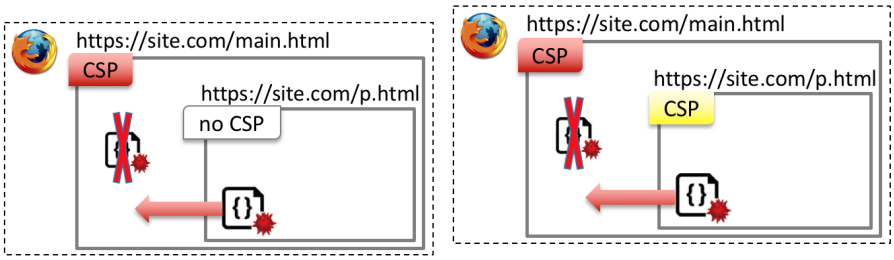
[1] More discussion on the verification vs. enforcement is given in [Fal10].

This survey was the first to cover both theoretical aspects of enforcement and practical considerations of the web browser architecture. The survey has successfully achieved its goal to introduce the field of *Web application security* to new researchers and has been cited in numerous papers and surveys on Web security.

2.2 Web Browser Security

Modern browsers implement different specifications to securely fetch and integrate content. One widely used specification to protect content is the *Same Origin Policy (SOP)* [Sam]. SOP allows developers to isolate untrusted content from a different origin. Intuitively, SOP is a mechanism that governs interactions between resources of web pages and protects undesired access between page sources of different origins.

A more recent specification to protect content in web pages is the *Content Security Policy (CSP)* [SSM10]. CSP is a mechanism designed to mitigate popular web vulnerabilities, such as cross site scripting attacks (XSS), data leaks attacks, and other types of attacks. CSP allows developers to specify, among other features, trusted domain sources from which to fetch content. One of the most important features of CSP is to allow a web application developer to specify trusted JavaScript sources. This kind of restriction is meant to permit execution of only trusted code and prevent untrusted code to access content of the page.



(a) same-origin embedded page does not have a CSP and makes embedding page vulnerable to attacks.

(b) same-origin embedded page has a weaker CSP than the embedding page thus weakening the protection of the embedding page.

Figure 2.1: **Fundamental problem of Content Security Policies (CSP) in embedded pages.** If either embedded (resp. embedding, not shown here) page has no CSP or a weaker CSP than embedding (resp. embedded) page, the overall protection of the Web application is set by the weakest CSP (or no CSP).

In our work [2], we have reported a fundamental problem of CSP described be-

low. CSP[WBV15] defines how to protect content in an isolated page. However, it does not take into consideration the page's context, that is its embedder or embedded content. In particular, CSP is unable to protect the page CSP is settled for if the page embeds other content of the same origin. The problem occurs because the CSP policy of a page will not be applied to an embedded content. However, due to SOP, the embedded content has complete access to the content of its embedder. Because same origin embedded content are transparent due to SOP, this opens loopholes to attackers whenever the CSP policy of an embedded content is not compatible to the one of its embedder page (see Fig. 2.1).

We analysed 1 million pages from the top 10,000 Alexa sites and found that 5.29% of sites contain some pages with CSPs [2]. We have identified that in 94% of cases, CSP may be violated in presence of the `document.domain` API and in 23.5% of cases CSP may be violated without any assumptions. Having found such possible violations on 46 popular websites, including `mail.ru`, `amazon.com`, `twitter.com` and others, we reported this problem to website owners. All our crawling data is publicly available at `http://webstats.inria.fr/?cspviolations` [SRB].

We discussed measures to avoid CSP violations in web applications by installing an origin-wide CSP and embedding content with sandboxed iframes. During our study, we also identified a divergence among browsers implementations in the enforcement of CSP in specific type of embedded content, called `srcdoc` sandboxed iframes – this divergence revealed a problem in Gecko-based browsers CSP implementation and appeared to be a bug in Mozilla Firefox browsers. To ameliorate the problematic conflicts of the security mechanisms, we discussed measures to avoid CSP violations.

Table 2.1: **Comparison of dynamic mechanisms based on runtime monitoring and information flow security analysis for JavaScript from [1].**

(a) Dynamic mechanisms based on runtime monitoring.

	Hallaraker and Vigna [HV05]	Yu et al. [YCIS07]	Phung et al. [DdSC09]	Meyerovich and Livshits [ML10]
Security properties				
safety	✓	✓	✓	✓
renewal		✓		
Formal guarantees				
soundness	×	✓	✓	×
transparency	×	✓	×	×
Implementation strategy	auditing interactions	code rewriting	code rewriting	JS engine modification

(b) Information flow security analysis.

	Vogt et al. [VNJ+07]	Chugh et al. [CMJL09]	Li et al. [LZW10]	Devriese and Piessens [DP10], De Groef et al. [DDNP12]	Austin and Flanagan [AF12]	Hedin and Sabelfeld [HS12]
Information flow						
explicit flow	dyn	stat	stat	dyn	dyn	dyn
implicit flow	stat	stat	×	dyn	dyn	dyn
Formal guarantees						
termination-insensitive noninterference	×	×	×	✓	✓	✓
time-sensitive noninterference	×	×	×	✓	×	×
precision or transparency	×	×	×	✓	×	✓
Implementation strategy						
browser extension		✓	✓			
JavaScript engine	✓			✓	✓	✓

Chapter 3

Information Flow Monitors

Information Flow Control (IFC) is a set of techniques to control how information flows through the execution of the program and is focused on enforcing the formal security guarantee of *noninterference*, ensuring that secret program inputs don't flow into public outputs. The first proposals to enforce noninterference were based on *static analysis* [SM03, HS06, MKS15] applied to the program as a whole, thus ensuring that all executions of the program satisfy noninterference.

Information flow control can also be done *dynamically*, as Fenton proposed back in 1974 [Fen74]. This approach again gained attention in 2000s, when researchers proposed *information flow monitors* [Zda02, LGBJS06, LG08, RS10, AF10, DP10], that are able to make a verdict about *a single execution of an otherwise insecure program* and stop it as soon as a potential violation of noninterference is detected. This approach has become particularly popular for dynamic languages, such as JavaScript [HBS15, HS12], where precise static analysis is practically impossible.

The Dream Information flow monitors must provide a strong security guarantee of *soundness* – that a monitor never allows public program outputs to leak secret inputs. However, this is not enough: a monitor might block all executions of the program, and still be sound. Therefore, the monitor should better accept all noninterferent executions, thus ensuring the formal guarantee of *transparency*[1] [BLW05, Erl03, HMS06, LBW09, B12].

The Reality The notion of noninterference does not apply to one single execution, but to a *a set of executions*. Therefore it is not possible to reason about each execution in isolation, and it is impossible for information flow monitor to be sound and fully transparent at the same time. The research community therefore compares sound monitors against each other in order to formally prove which monitor accepts more noninterferent executions, and defines this as relative *permissiveness* [Gue07, AF12, AF10, HBS15].

[1] In works on reference monitors, transparency requires that the semantics of executions that obey the security property in question is preserved [BLW05].

3.1 Hybrid monitoring of attacker knowledge

Hybrid information flow monitors complement an analysis of one program execution with static analysis of other, non-executed branches of the code to detect possible implicit information flows. Several models of hybrid monitors have been proposed in the early 2000 [LG08, LGBJS06, RS10].

3.1.1 Computing attacker's knowledge with hybrid monitors

While monitoring the execution of a program, an attacker can gain a certain amount of knowledge about the secret inputs to the program. The notion of *attacker's knowledge* has been proposed [AS07, AC12] to define the security conditions of monitors, but have not been used to enforce noninterference.

We have proposed a new model of a hybrid information flow monitor that estimates the *knowledge of the attacker* about the secret program inputs [3]. This new class of monitors, called *hybrid monitors*, are based on a combination of dynamic and static program analysis, and compute *how much information about the secret input has been leaked in the program output.* We prove that four versions of such monitors never underestimate the attacker's knowledge (soundness) and prove which of the four monitors are more precise than others (in modelling the attacker's knowledge more precisely). All theorems are proven within Coq proof assistant.

Example 1. *Program 1 shows a program snippet where* `h1` *and* `h2` *are secret inputs and* `x` *and* `y` *are two public variables. The* `output` *operation makes the program output visible to the attacker whose goal is to learn any information about the secret inputs* `h1` *and* `h2`*.*

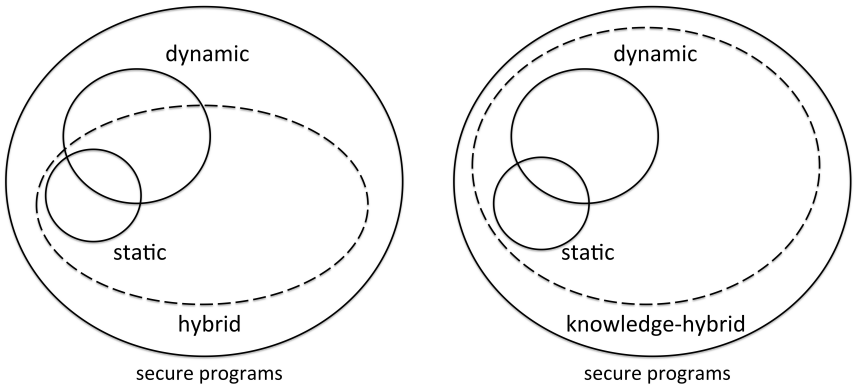
Program 1

```
1 x = 0; y = 0;
2 if (h1) then y = 1;
3 if (h2) then x = 1 else x = y;
4 output x
```

Consider an execution of Program 1 when its secret inputs are `h1` *= false and* `h2` *= true. When Program 1 outputs* 1*, the attacker learns that either* $h1$ *or* $h2$ *was true:*

$$\textit{Actual attacker's knowledge: } h1 \vee h2$$

Given the same initial memory where `h1` *= false and* `h2` *= true, our first hybrid monitor [3] computes the knowledge in the output* `x`*. The first test on* `h1` *fails, thus the value of* `y` *remains unchanged and the knowledge of* `y` *is* $\neg h1$*. Upon the second test, the monitor concludes that the value of* `x` *is* 1 *in the true branch and* 0 *in the false branch. As the values are not the same, the knowledge of the output* `x`=1 *might depend on* $h2$ *and on the knowledge of* `y`*. Therefore, the knowledge*



(a) from Hedin et al. [HBS15]. (b) our result from [6].

Figure 3.1: **Relative permissiveness revisited from [6].** While Hedin et al. [HBS15] proved that dynamic and hybrid monitors are incomparable in their permissiveness, we show that a *knowledge-hybrid* monitor is capable to be provable more permissive than dynamic monitors.

computed by the monitor [3] is:

$$\textit{Approximated knowledge by [3]:} \neg h1 \wedge h2$$

The approximated knowledge is much less precise than the real attacker knowledge. The reason for this gap between estimated and actual knowledge is in the choice of the model of knowledge domain. Intuitively, the knowledge in [3] is limited to a set of environments that can contribute to the *current value of* `x`.

3.1.2 Improving permissiveness of hybrid monitors

It would be expected that hybrid monitors are more *permissive* than their dynamic counterparts: hybrid monitors should recognize more noninterferent executions[2] thanks to the static analysis they rely on for non-executed branches of the program. However, Hedin et al. [HBS15, Theorems 2, 3] have proven that hybrid and dynamic monitors are incomparable in their permissiveness.

We extended our hybrid monitor [3] with a more expressive representation of attacker knowledge, build from the one program execution and enhanced static analysis of non-executed branches. We thus have proposed a new family of monitors that profit from the modeled attacker knowledge, that we refer to here as *knowledge-hybrid monitors* [6]. Such monitors are able to enforce noninterference *more permissively* thus allowing more executions of programs than previous hybrid monitors [HBS15]. Figure 3.1 demonstrates that our knowledge-hybrid

[2]In this section, we only consider termination-insensitive noninterference.

monitor is *provably more permissive than purely dynamic or purely static approaches.*

Moreover, the knowledge-hybrid hybrid monitor is based on a more sophisticated static analysis that much more closely replicates the analysis provided by a dynamic counterpart of the monitor (see a single semantics for static and dynamic analysis in [6, Fig. 3]). As a result, the knowledge-hybrid monitor is able to compute an attacker's knowledge much more precisely.

Example 2. *Consider again an execution of Program 1 from Example 1 when its secret inputs are* `h1` *= false and* `h2` *= true. When Program 1 outputs* 1*, the attacker learns that either* $h1$ *or* $h2$ *was true, and the knowledge-hybrid monitor [6] computes the knowledge precisely:*

Actual attacker's knowledge as computed by [6]: $h1 \vee h2$

3.1.3 Application to browser fingerprinting

A *browser fingerprint* is a set of information related to a user's device from the hardware to the operating system to the browser and its configuration. *Browser fingerprinting* refers to the process of collecting information through a web browser to build a fingerprint of a device [Lap17]. Passive fingerprinting relies on the HTTP headers and other information a server received together with HTTP requests. Active fingerprinting is performed via a JavaScript code executed in a Web browser, from a wide variety of information from public browser APIs. Browser fingerprinting can uniquely identify a Web user and consequently trace her across the Web.

The hybrid information flow monitors that estimate the *knowledge of the attacker* that we proposed [3, 6] could be used to detect active fingerprinting if extended to JavaScript language. Following the standard information theory definition of *self-information* or *surprisal* used by Eckersley in his state-of-the art paper on browser fingerprinting [Eck10a] and other follow-up works, we proposed to adopt it to the attacker knowledge. If the probability of a browser feature f to have a value v is $P(f = v)$, then the self-information is:

$$I(f = v) = -log_2 P(f = v) \tag{3.1}$$

If we label all browser features used for fingerprinting as secret program inputs, then our hybrid monitor can detect at the execution time *when a JavaScript program performs browser fingerprinting.* The monitor observes an execution of a JavaScript program and analyses how much information (in terms of browser and OS features) the program is collecting. The more identifying information the program collects, the more likely it is performing browser fingerprinting.

Example 3. *Consider again Program 1 from Example 1. We assume the following probabilities for fingerprinting features* `h1` *and* `h2`*:* $P(h1) = 0.21$ *(e.g., a test on a "FireFox" browser name) and* $P(h2) = 0.01$ *(e.g., a test on a time zone). In this simple example we assume that the browser features are independent, and therefore* $P(h1 \wedge h2) = P(h1) \cdot P(h2)$.

The first hybrid monitor [3] models the knowledge of the attacker as $\neg h1 \wedge h2$ *and hence,*

$$P(\neg h1 \wedge h2) = P(\neg h1) \cdot \ P(h2) = 0.0079$$

Therefore the information leakage the attacker is modeled to receive is

$$I(\neg h1 \wedge h2) = -log_2 P(\neg h1 \wedge h2) = 6.98 \ bits$$

However, the real knowledge is properly modeled by an advanced monitor [6]:

$$P(h1 \vee h2) = 1 - P(\neg h1) \cdot \ P(\neg h2) = 0.2179$$

Therefore the actual attacker's knowledge contains much less information:

$$I(h1 \vee h2) = -log_2 P(h1 \vee h2) = 2.20 \ bits$$

Browser randomisation against fingerprinting. Several works showed that restricting access to browser fingerprinting APIs or stopping executions of JavaScript programs may not work work in practice [AJN+13b]. Therefore, several approaches directed at *randomisation of browser features* have been proposed [Bod, NJL15]. Nevertheless, none of the proposed approaches have provided privacy guarantees that even with randomisation in place the probability of identifying a user is bounded by a certain threshold. We used information-theoretic channels to model the knowledge of the tracker and the fingerprinting program, and show how to synthesise a randomisation mechanism that defines the distribution of configurations for each user [4]. Our randomisation mechanism provides *a strong guarantee of privacy* (that the probability of identifying the user is bounded by a given threshold) while maximising *usability* (thus minimizing the number of time the user has to switch to other configurations).

3.1.4 Challenge: The need for a new quantitative information flow measure

Differently from dynamic or hybrid monitors that enforce an all-or-nothing property of noninterference [Zda02, AF10, DP10, AF12, HBS15], our hybrid information flow monitors [3, 6] are more general as they rely on the definition of measure for quantifying information leakage.

A number of definitions for such a measure has been proposed in the scientific

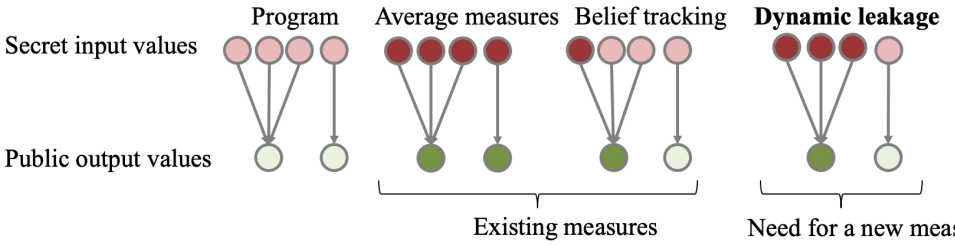


Figure 3.2: **Different measures of information leakage from [8].** Average measures evaluate the amount of leakage for all possible program inputs and outputs (all inputs and outputs are highlighted in vivid colors). Belief tracking evaluates the leakage associated to one input and one output. Dynamic leakage should measure the amount of information leaked when an attacker observes one program output that could have been caused by several possible inputs.

community. The most popular definitions are based on measuring the decrease in attacker's *uncertainty* about the possible values of the secret and use Shannon entropy [CT06], Min entropy [Smi09], Guessing entropy and g-leakage [ACPS12]. An interesting aspect of the proposed definitions is that all of them measure *a program as a whole*. In other words, they evaluate how much information is leaked *on average* by all possible program outputs. We will call these measures "average measures". To analyse the leakage of a program as a whole, average measures were previously used by a number of works on static program analysis [BKR09, KR10] and even approximated by a dynamic analysis [ME08]. While these measures perfectly fit for static program analyses, they cannot be used by dynamic analyses since they do not specify what information an attacker learns through observing one concrete program output. An alternative approach to measure information leakage is to reason about attacker's *accuracy* in his belief about the secret [CMS09]. This belief tracking measure evaluates the leakage *for a concrete secret, and a concrete program output.*

We have studied and compared all existing definitions to find a suitable definition for quantitative information flow analysis [8]. Figure 3.2 graphically shows the difference in the definitions of existing measures. A definition of *dynamic leakage* should evaluate how much information an attacker learns when she observes one program output.

Conclusion We found out that a suitable measure for dynamic leakage exists only for deterministic programs. This work hence opened a new research question in quantitative information flow area that needs to be addressed by the community: *Which definition of dynamic leakage for one program output is suitable for probabilistic programs?*

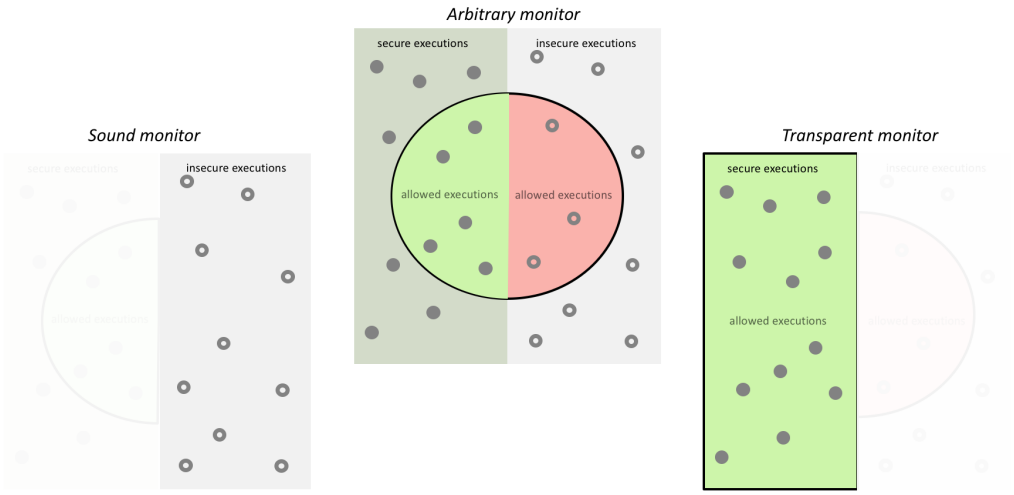


Figure 3.3: **Visualisation of the concepts of soundness and transparency**. Soundness does not allow any insecure executions, but says nothing about secure ones. Transparency requires to allow all secure executions.

3.2 Taxonomy of information flow monitors

In early 2010, dynamic enforcement mechanisms gained attention and became used for controlling information flow in JavaScript programs [HS12, SR14, MSR14, DP10, AF12]. Given the dynamic nature of the proposed solutions, we can apply mathematical definitions for reasoning about the security guarantees derived from the well-studied *theory of runtime monitoring* [BLW05, Erl03, B12]. The two main notions that have been proposed are:

- *soundness* that the monitor always prevents information leaks about secret inputs, and
- *transparency* that the monitor does not block executions that do not leak secret information[3].

Figure 3.3 proposes a very simplified view of soundness and transparency. In simple terms, a monitor is a recognizer of noninterferent (we call them here *secure*) executions of a program, similarly to recognizers in pattern recognition[4]. The goal of the monitor is to block all insecure executions, thus complying with the principle of *soundness*. Orthogonal to soundness, a monitor has a second

[3]Bauer et al. [BLW05] actually provide a more subtle definition, saying a monitor should output a semantically equivalent trace.

[4]Figure 3.3 is inspired by the visualisation of precision and recall for pattern recognition from Wikipedia `https://en.wikipedia.org/wiki/Precision_and_recall` accessed on March 5, 2021.

goal: to accept as many secure executions as possible, and in the best case all of them, thus achieving full *transparency*.

However, since dynamic information flow monitors enforce *noninterference*, which is a property of a *set of executions* [CS10], they are not able to be sound and transparent at the same time. To improve the situation, several hybrid approaches have been proposed, and approaches based on multi-execution[5].

3.2.1 Precision, permissiveness and transparency

A number of works on dynamic information flow monitors try to analyse precision of monitors. Intuitively, precision describes how often a monitor blocks (or modifies) secure programs. Different approaches have been taken to compare precision of monitors, using definitions such as "precision", "permissiveness" and "transparency". We propose a rigorous comparison of these definitions in this section, originally published in [7, Section 6].

Precision (versus well typed programs) Le Guernic et al. [LGBJS06] were among the first to start the discussion on transparency for information flow monitors. The authors have proved that their hybrid monitor accepts all the executions of a program that is well typed under a flow-insensitive type system similar to the one of Volpano et al. [VSI96]. Le Guernic [Gue07] names this result as *partial transparency*. Russo and Sabelfeld [RS10] prove a similar result: they show that a hybrid monitor accepts all the executions of a program that is well typed under the flow-sensitive type system of Hunt and Sands [HS06].

Precision (versus secure programs) Devriese and Piessens [DP10] propose a stronger notion, called *precision*, that requires a monitor to accept all the executions of all secure programs. Notice that this definition is stronger because not only the monitor should recognise the executions of well typed programs, but also of secure programs that are not well typed. Devriese and Piessens have proven that such precision guarantee holds for Secure-Multi Execution (SME) monitor [DP10].

Transparency (versus secure executions) As a follow-up, Zanarini et al. [ZJR13] have proven that another monitor based on SME satisfies *transparency*. This monitor accepts all the secure executions of a program, even if the program itself is insecure.

[5] For the introduction to hybrid monitors, see [LG08, Gue07], while multi-execution has been proposed for the first time in [DP10].

Permissiveness (versus executions accepted by other monitors) In his PhD thesis, Le Guernic [Gue07] compares his hybrid monitor with another hybrid monitor that performs a more precise static analysis, and proves an *improved precision* theorem stating that whenever the first hybrid monitor accepts an execution, the second monitor accepts it as well. Following this result, we investigated other hybrid monitors [3] (see Section 3.1) and have proven their relative precision in the style of Le Guernic, while Austin and Flanagan [AF10, AF12] use the same definition to compare their dynamic monitors. Hedin et al. [HBS15] name the same notion by *permissiveness* and compare the sets of accepted executions: one monitor is more permissive than another one if its set of accepted executions contains a set of accepted executions of the other monitor.

3.2.2 Comparison of information flow monitors

Relative true and false transparency, relative precision To compare precision of different information flow monitors, we proposed to distinguish two notions of transparency [7]. *True transparency* defines the secure executions accepted by a monitor, and *false transparency* defines the insecure executions accepted by a monitor. We found that none of the studied information flow monitors are true transparent for termination-insensitive noninterference. Nevertheless, the notion of *true transparency* allows us to define a *relative true transparency* to better compare the behaviours of information flow monitors when they deal with secure executions.

Intuitively, one monitor A is more *true transparent* than monitor B, if A accepts allows more secure terminating executions than B without modifications. Similarly, a monitor A is more *false transparent* than monitor B, if A accepts allows more insecure terminating executions than B without modifications.

We have discovered that certain monitors are incomparable with respect to true transparency. To compare them, we proposed a more coarse-grained definition that describes the monitors' behaviour on secure programs. Similarly to the notion of permissiveness [HBS15] and precision [DP10], we proposed to reason about secure programs instead of secure executions. To do so, we first define a set of secure programs P, where a monitor A accepts all the executions of P, that we denote by $\mathcal{P}(A)$. Intuitively, *relative precision* is defined as follows. A monitor A is *more precise* than a monitor B, if A accepts all executions of more secure programs than B, that is $\mathcal{P}(A) \supseteq \mathcal{P}(B)$.

We compare previous notions of precision, permissiveness and transparency to our formalisation [7] and summarize our findings in Table 3.1.

Rigorous analysis of five monitors We rigorously analysed [7] five widely explored information flow monitors: no-sensitive-upgrade (NSU) [Zda02], permissive-upgrade (PU) [AF10], hybrid monitor (HM) [GBJS06], secure multi-execution

Table 3.1: Definitions for comparing information flow monitors from the related works and their relation to our formalisation: true and false transparency [7].

Definition from the literature	**Our formalisation [7]**
Permissiveness [HBS15]	Relative true transparency + relative false transparency
Precision [DP10]	True transparency for all the executions of a secure program
Relative Precision (this paper)	Relative true transparency for all the executions of a secure program
Transparency [BLW05]	True transparency

(SME) [DP10], and multiple facets (MF) [AF12], that (at the time of writing) were well-established as state-of-the-art in the foundational security community. We proposed *a fine-grained approach to compare and contrast information flow monitors with respect to their security guarantees* [7].

We have then further analysed two most widely used information flow control mechanisms, based on multi-execution, that were believed to be the equivalent: secure multi-execution (SME) [DP10] and multiple facets (MF) [AF12]. We found *a number of fundamental differences in their design and security guarantees* [5].

Finally, we have then analysed and proposed extensions [9] to the multiple facets (MF) [AF12], that has been proven to be a good fit for implementing information flow security for JavaScript. Our extension can be used for more complex security policies, optimized the number of multi-executions and has stronger security guarantees than MF.

3.2.3 Challenge: The research community needs to use one unified framework to compare information flow monitors

Our work [7] demonstrated that various information flow monitors have been proposed in the latest years [Zda02, AF10, AF12, DP10, 9], yet when new monitor is proposed, it is rarely rigorously evaluated against previously known monitors.

The notion of *soundness* only ensures that insecure executions are never allowed by the monitor (see Fig. 3.3), however what distinguishes the monitors is their ability to allow secure executions (of otherwise insecure programs) without modifications[6].

[6]Interestingly, I've been writing exactly the same conclusion in my PhD thesis on runtime

The notion of *transparency* helps to reason about monitors, while it is impossible to achieve for noninterference. Therefore, notions of relative transparency – such as *relative false transparency* and *relative true transparency*, and *relative precision* defined in [7, Section 6] – can help researchers to compare monitors between themselves.

Conclusion. While comparing effectiveness of various pattern recognition systems (such as, for example, face recognition systems) against each other is a common approach in the corresponding research community, this is not the case yet in *language-based security* research community. The community needs to strive for robust comparisons with respect to soundness and relative transparency and precision when evaluating novel information flow monitors against existing monitors.

monitoring back in 2011 [B12, page 121].

Chapter 4

Web Tracking Technologies

The Web has become an essential part of our lives: billions are using Web applications on a daily basis. While the users browse the web, they are leaving *digital traces* on millions of websites [EN16a, NKJ+13a]. Such traces allow third-party advertising companies and data brokers to continuously profit from collecting a vast amount of data associated to the users. At the same time, the users do not have much control of who is collecting their data and for what purpose. Recent research has shown that such third parties use a wide range of techniques in order to track users across the Web [SCM+10, RKW12, AWS+11, OTC14, EN16a, Eck10a, AJN+13b, NKJ+13a, LSKR16, DPS20]. These techniques can be used to reconstruct browsing sessions and to create profiles of users, inferring, among others, their hobbies, health status, political inclinations, and level of wealth.

The Dream To escape Web tracking, users decide to use privacy-preserving browsers or install browser extensions. To protect users, browser vendors and extensions rely on either *filter lists* of known trackers [EL, EP, Disc] or *detecting trackers by their behaviour* [RKW12, 14, LSKR16]. Users would prefer that such tools block all unneeded (for users) tracking when they visit websites.

The Reality Protection from tracking is often not effective because trackers re-create cookies after deletion [SCM+10, AWS+11] or integrate them into functional third-party content [14]. Moreover, because of a Real-Time-Bidding process [BARW16] in targeted advertisement, *third-parties synchronize cookies between them*, in order to merge users' data collected across websites and via different trackers [AEE+14b, EN16a]. As of today, there is no common methodology that allows to distinguish cookies useful cookies from unneeded tracking cookies. Alternatively, *browser fingerprinting* [13] allows to track users without storing any identifier in the user's browser [Eck10b, LRB16a]. Finally, users can be uniquely identified only by the browser extensions that have installed to protect themselves [12].

4.1 Cookie-based tracking: detection and protection

In the last decade, numerous studies measured prevalence of third-party trackers on the Web [RKW12, AJN+13b, NKJ+13a, EN16a, LSKR16, BARW16, YMMP16, BW18, LN18, LGN18]. Web Tracking is often considered in the context of *targeted behavioral advertising*, but it's not limited to ads. Third-party tracking has become deeply integrated into the Web contents that owners include in their websites.

But how to recognize that a third-party request is performing tracking? To detect trackers, the research community applied a variety of methodologies. The most known Web tracking technique is based on *cookies*, but only some cookies contain *unique identifiers* and hence are capable of tracking the users. Some studies detect trackers by analysing cookie storage, and third-party requests and responses that set or send cookies [RKW12, LSKR16], while other works measured the mere presence of third-party cookies [LN18, LGN18]. To measure *cookie syncing* – a technique that allows two different trackers to match their tracking cookies – researchers applied various heuristics to filter cookies with unique identifiers [AEE+14b, ERE+15, EN16a]. However, this approach has never been applied to detect tracking at large scale. Overall, previous works provide different methods to identify third-party requests that are responsible for tracking [RKW12, YMMP16].

Detection of *unique identifiers* stored in cookies and analysing behaviors of third-party domains is a complex and time-consuming task. Therefore, most of the state-of-the-art works, as well as popular ad- and tracking- blocking browser extensions[1] rely on *filter lists*. In particular, EasyList [EL] and EasyPrivacy [EP] (EL&EP) and Disconnect [Dis] lists became the *de facto* approach to detect third-party tracking requests in privacy and measurement communities [EN16a, BARW16, LCA+17, RNV+18, IAK+17, EHN18, BW18, BAK+18, ISPL18]. We summarize the usage of filter lists in security, privacy and web measurement community in Table 4.1.

Nevertheless, filter lists detect only known tracking and ad-related requests. Therefore, a tracker can avoid this detection by using a different subdomain for tracking, or register a new domain if the filter list blocks the entire domain. Third parties can also incorporate tracking behavior into functional website content, which is never blocked by filter lists because blocking functional content would harm user experience. Therefore, it is important *to evaluate how effective are filter lists at detecting trackers, how many trackers are missed by the research community in their studies*, and whether filter lists should still be used as the *default tools* to detect trackers at scale.

[1] Popular ad-blocking browser extensions AdBlock Plus [adbb] and uBlockOrigin [ubl] also rely on EasyList and EasyPrivacy. Disconnect browser extension [Disc] and in Enhanced Tracking Protection of Firefox browser [fir] rely on Disconnect list.

Table 4.1: **Usage of EL&EP lists in security, privacy, and web measurement communities (venues from 2016-2018) [14, Table 12].** "Detect" column describes how EL&EP were used to detect trackers: whether the filter lists were applied on all requests ("Req"), on requests and follow-up requests that would be blocked ("Req+F") or whether filter lists were further customised before being applied to the dataset ("Custom"). In the column "Depend", by "Rely" we denote that EL&EP are used to build their results, while "Verify" means that EL&EP lists are used to verify the results.

Paper	Venue	EasyList	EasyPrivacy	Detect	Depend
Englehardt & Narayanan [EN16a]	ACM CCS'16	✓	✓	Req	Rely
Bashir et al. [BARW16]	USENIX Security'16	✓		Custom	Rely
Lauinger et al. [LCA+17]	NDSS'17	✓	✓	Req+F	Rely
Razaghpanah et al. [RNV+18]	NDSS'18	✓		Custom.	Rely
Ikram et al. [IAK+17]	PETs'17	✓		Req+F	Verify
Englehardt et al. [EHN18]	PETs'18	✓	✓	Req+F	Verify
Bashir & Wilson [BW18]	PETs'18	✓	✓	Custom	Rely + Verify
Bashir et al. [BAK+18]	IMC'18	✓	✓	Custom	Rely
Iordanou et al. [ISPL18]	IMC'18	✓	✓	Req+F	Rely

4.1.1 Detecting cookie syncing and ID sharing

We propose a new, fine-grained behavior-based tracking detection [14]. Our alternative method to detect trackers is inspired by analyzing behavior (HTTP(S) request and responses) that lead to loading invisible pixels when visiting websites. Since invisible pixels have no particular functionality, we use them as *perfect suspects for tracking* in this research.

By crawling 84,658 webpages from top Alexa 8,744 domains, we detect that third-party invisible pixels are widely deployed: they are present on more than 94.51% of domains and constitute 35.66% of all third-party images. Figure 4.1 shows the distribution of the number of pixels in all collected images.

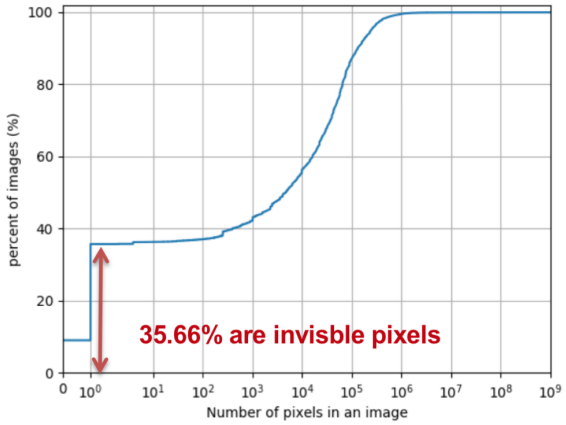


Figure 4.1: **Number of pixels in 2M images collected on 85K pages [14].** Cumulative function of the number of pixels in images with a *content-length* less than 100 KB. Out of 2M images, 36% *are invisible*: 9% have no content and 27% are of size 1×1 pixel.

We propose a fine-grained behavioral classification of tracking based on the analysis of invisible pixels, inspired by Roesner et al. [RKW12]. We use this classification to detect new categories of tracking and uncover new collaborations between domains on the full dataset of 4,216,454 third-party requests. While interested readers can check the classification categories we have detected directly in [14], we here bring particular attention to one category, called *first-to-third party cookie syncing* that has not been studied explicitly prior to this work.

First-to-third party cookie syncing. We have detected that *privacy-friendly first-party analytics cookies get synchronized with third-party tracking cookies.* This technique is particularly worrisome in the light of upcoming deprecation of third-party cookies in the most popular browser, Google Chrome [Goo]. First-to-third party cookie syncing allows third parties to recognize the user within the visited website, and at the same time link them to their partial profile history collected before deprecation. We present this technique in Figure 4.2.

We detected first to third party cookie syncing in 67.96% of visited domains. In total, we found 17,415 different partners involved in such syncing. The top couple of partners is `google-analytics.com` and `doubleclick.net`. We found that `google-analytics.com` first receives the cookie as part of the URL parameters. Then, through a redirection process, `google-analytics.com` transfers the first party cookie to `doubleclick.net` that inserts or receives an identifier in the user's browser. We found out that `google-analytics.com` is triggering such first party cookie syncing on 38.91% of visited websites.

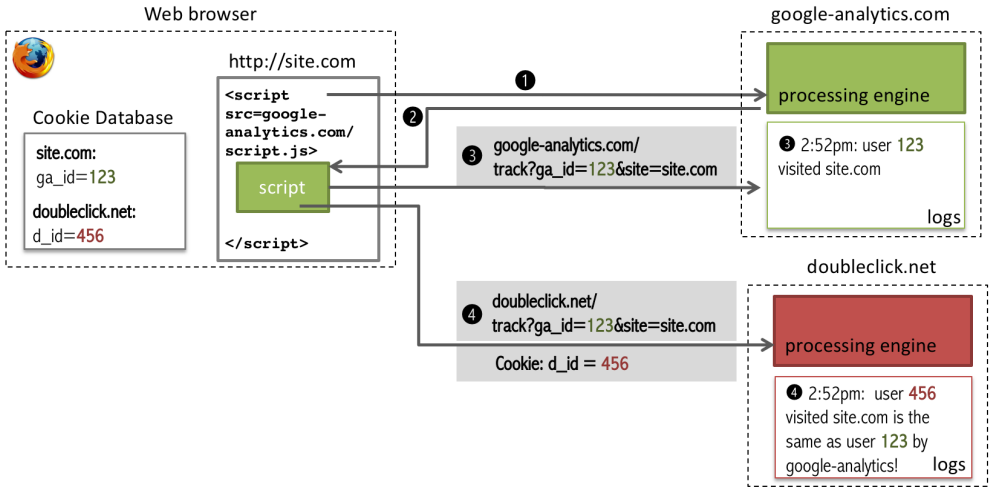


Figure 4.2: **First-to-third party cookie syncing.** In this example, the script loaded from `google-analytics.com` shares the first-party cookie `ga_id` via a redirection in a request parameter to `doubleclick.net`. As a result, `doubleclick.net` receives its own third-party cookie `d_id` and is able to conclude that both cookies belong to the same user.

Why Google Analytics syncs cookies with Doubleclick? We found that the main reason behind such syncronisation activity is the *customization of Google Analytics* allowing each web developer *"Enable the Demographics and Interests reports for the property"*. The settings webpage of Google Analytics states that *"Analytics collects Demographics and Interests data from the following sources:"*, then listing *"Third-party DoubleClick cookie"*, requiring that *"Cookie is present"* and specifying that in this case *"Analytics collects any demographic and interests information available in the [Doubleclick] cookie"* [GAn].

French DPA decision. Interestingly, in November 2020 the French Data Protection Authority (CNIL) sanctioned the website of a popular supermarket chain "Carrefour". In the official text [CNILc], CNIL explains that Carrefour website (1) contains the first-party cookies of Google Analytics, (2) synchronizes Google Analytics cookies with Google Ads (who serve ads with the help of `doubleclick.net`, and (3) user consent is needed for such data merging (for more details on the legal requirements and exceptions of consent, see Chapter 5).

Conclusion. Even though third-party cookies will soon be deprecated from the Web [GSa], the browsing profiles collected by the companies and synced

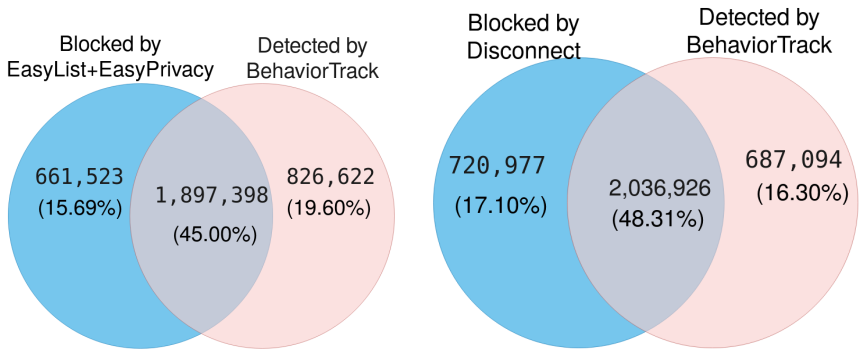
across multiple third parties are still accessible, and techniques such as *first-to-third party cookie syncing* already allow such companies to merge profiles and to continue tracking users via first-party cookies.

4.1.2 How effective are filter lists and browser extensions?

We demonstrate that two popular methods to detect tracking, based on EL&EP and on Disconnect lists respectively miss 25.22% and 30.34% of the trackers that we detect [14]. Moreover, we find that if we combine all three lists, then we still detect 379,245 requests originated from 8,744 domains that still track users on 68.70% of websites.

Figure 4.3 provides an overview of third party requests blocked by filter lists or detected as tracking requests according to our method, that we call BehaviorTrack. Out of all 4,216,454 third party requests collected from 84,658 pages of 8,744 successfully crawled Alexa top domains:

- 2,558,921 (60.7%) requests were blocked by EL&EP,
- 2,757,903 (65.4%) were blocked by Disconnect, and
- 2,724,020 (64.6%) were detected as performing tracking by BehaviorTrack.



(a) EasyList & EasyPrivacy (b) Disconnect

Figure 4.3: **Effectiveness of filter lists at detecting trackers** on 4,216,454 third party requests from 84,658 pages. Further explanations on why filter lists miss trackers can be found in [14, Section 5.1].

We then analyze how effective are the popular privacy protection extensions in blocking the privacy leaks detected by our method [14]. We study the following extensions: Adblock [AdBa], Ghostery [Gho], Disconnect [Disc], and Privacy Badger [PB]. We performed simultaneous stateful Web measurements of the Alexa top 10K websites in November 2019 from servers located in France. Figure 4.4 represents the effectiveness of the extensions in blocking the tracking requests detected by our method. Similarly to Merzdovnik et al. [MHB+17], we show that tracker blockers (Disconnect, Ghostery and Badger) are more efficient

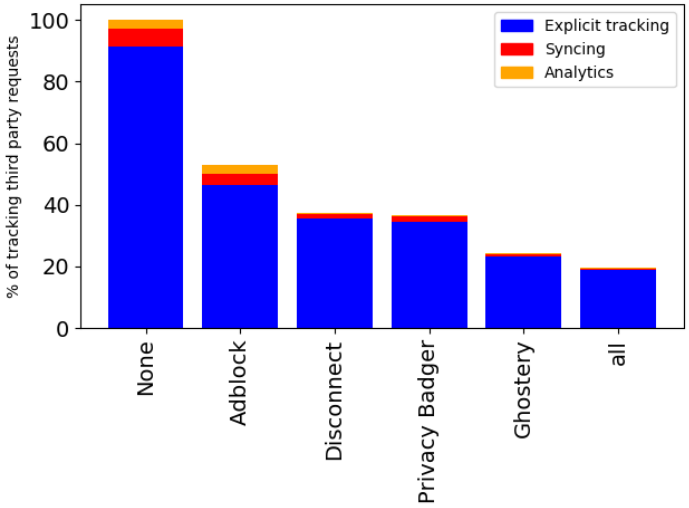


Figure 4.4: **Percentage of third party requests still allowed by ad- and tracking-blocking browser extensions.** 2,924,480 tracking requests detected by our method were collected on Alexa top 10K websites. For further details, check [14, Section 6].

than adblockers (Adblock) in blocking tracking behaviors. However, all studied extensions miss at least 24.38% of the tracking detected by our method. This shows that even though the extensions reduce the amount of tracking performed, they do not solve the problem of protecting users from tracking.

4.1.3 Challenge: How to protect users from Web tracking without breaking Web applications?

In our research [14], we found that both filter lists and browser extensions miss a significant number of trackers that we detect (see Section 4.1.2). However, why do filter lists miss trackers? And is it possible to block all tracking without degrading user experience? In this section we investigate these questions and discuss open challenges.

Blocking vs detection. Filter lists are designed to block outgoing requests that are suspected to contain trackers. We have to be aware that techniques designed to *block* trackers, such as filter lists and browser extensions, are by nature drastically different from techniques for *detection and visualisation* of trackers. We show it with our dataset below.

For simplicity, we say that a request is *FT* is it matches one of the rules of the studied filter list. While comparing requests and responses we detect as tracking

Content type	Missed by EL&EP	Missed by Disconnect
script	33.38%	35.27 %
big images	20.62%	21.73 %
text/html	13.77%	14.73 %
font	8.79%	0.09 %
invisible images	6.68%	12.21 %
stylesheet	6.17%	3.05 %
application/json	4.00%	4.83 %
others	6.59%	8.12%

Table 4.2: **Top content type detected by BehaviorTrack** [14] on the 708,308 requests missed by EL&EP and the 640,809 missed by Disconnect. We refer to images with dimensions larger than 50×50 pixels as "big images".

with BehaviorTrack to *FT* requests, we noticed that a significant number of requests we detect would have been blocked by the filter lists. This phenomena happens because requests detected by BehaviorTrack are not included directly in the website but instead are initiated by other third-party *FT* requests that would have been blocked if a browser extension based on filter lists was installed. By further analyzing the requests only detected as tracking by BehaviorTrack and missed by EL&EP (resp. Disconnect), we found that 14.31% (resp. 6.73%) of the requests detected only by BehaviorTrack are *requests initiated by FT requests*.

What type of content contains trackers? To answer this question, we investigate the type of content that includes cookie-based tracking and sharing of identifiers. Table 4.2 presents the top content types used for tracking and not blocked by the filter lists. We observe that mere *loading of scripts* already includes cookie-based tracking, followed by loading of big images, likely used for advertisement. Other functional content, such as stylesheets and fonts also contain trackers. We selected the top 30 third party services not blocked by neither of the filter lists but detected by BehaviorTrack, and categorized them using Symantec's WebPulse Site Review. Out of 30 such trackers, EL&EP and Disconnect miss 23% of "Content Server" content, followed by "Social Networking", "Search Engines", and "Web Ads/Analytics" content (see [14, Table 7]).

Such content seems to be useful for the *functionality of the website* and hence filter lists do not block it in order to maintain the user experience. But why does the functional content contain tracking?

Our further investigation of content that contains tracking but is missed by popular filter lists allowed us to find two possible answers to this question.

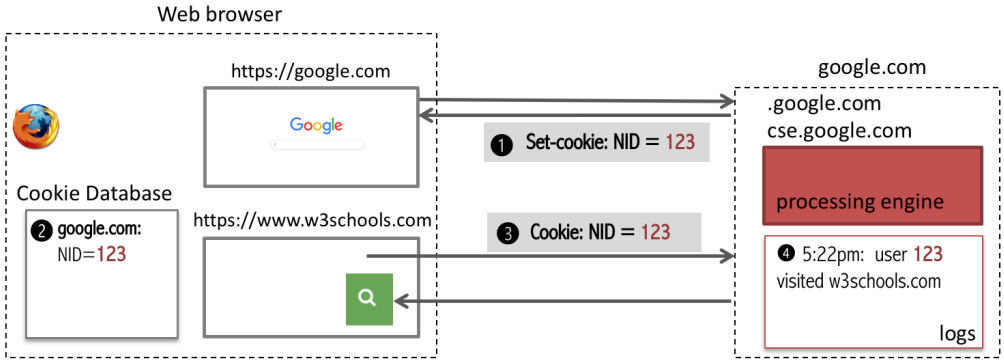


Figure 4.5: **First-party cookie `NID` set in a first party context of `google.com` becomes "third-party" when user visits `w3schools.com`.** First party cookie `NID` is set in the user's browser upon the visit to `google.com`. This cookies is automatically sent to `google.com` when a third-party content from `cse.google.com` is fetched by the browser upon her visit to `w3schools.com`.

First-party cookies become third-party. First, it is still a very common practice that a cookie set as a first-party is subsequently sent to a third-party thus initiating cross-site tracking. We have detected the following example on a website `w3school.com` that we demonstrate in Figure 4.5. When a user visits `google.com`, a first party cookie `NID` is set in the user's browser. Later on, when a user visits `w3school.com`, a request is sent to the service `cse.google.com` (Custom Search Engine by Google). This service seem to provide a search functionality on the `w3school.com` website. Along with the request, Google's `NID` cookie is sent to `cse.google.com`. The filter list cannot block such a request, and is incapable of removing the first party tracking cookies from it because blocking `cse.google.com` breaks the search functionality of the website.

We found that this behavior explains a significant amount of requests missed by filter lists: 45% requests missed by EL&EP and 32% requests missed by Disconnect contain cookies initially set in a first party context.

Large scope cookies. A cookie set with a 2^{nd}-level TLD domain can be accessed by all its subdomains: for example, a cookie set with `tracker.com` as its domain is sent along with all requests to its subdomains, such as `sub.tracker.com`. Such *large scope cookies* are extremely prevalent among requests missed by the filter lists. We found that 77% of third-party cookies in the requests missed by EL&EP and 75% of such cookies in requests missed by Disconnect are set with a 2^{nd}-level TLD domain (such as `tracker.com`) and hence accessible to all subdomains.

Conclusions. We identified that functional content often delivers or transmits cookies that enable cross-site tracking. Such tracking often happens because (i) initially first-party tracking cookies are sent with requests to fetch (functional) third-party content; and (ii) the scope of cookies is not limited to the subdomain that sets it (e.g., cookies set by `cse.google.com` have a large scope of `google.com`). Browser vendors and researchers should work towards more fine-grained approaches than filter lists, identify scripts (instead of domains) responsible for sharing and syncing of cookies, and finally consider techniques to limit first-party cookies that become third-party.

4.1.4 Helping Web developers to remove unwanted tracking

Website developers often need to include third party content in order to provide basic functionality. However, when a developer includes a third party content, they cannot know whether the third party contains tracking mechanisms. If a website developer wants to protect her users from being tracked, the only solution is to exclude any third-party content, thus trading functionality for privacy.

We have developed a new Web application architecture that allows web developers to gain control over certain types of third party content [11]. Our solution is based on the automatic rewriting of the web application in such a way that the third party requests are redirected to a trusted web server, with a different domain than the main site. This trusted web server may be either controlled by a trusted party, or by a main site owner – it is enough that the trusted web server has a different domain. A trusted server is needed so that the user's browser will treat all redirected requests as third party requests, like in the original web application. The trusted server automatically eliminates third-party tracking cookies and other technologies.

4.1.5 Do cookies influence the price you're going to pay?

An alternative monetization strategy to targeted advertising, is *price discrimination*. Price discrimination involves varying the price of any specific product or service, depending on the amount that a customer is willing to pay. The vast amounts of user-data gathered on the modern web are a natural fit for such a pricing strategy [Odl03]. Knowing, for instance, that a user visits webpages which sell high-end goods can be used to infer the income level of that user. This knowledge can, in turn, be used to dynamically increase the price of a product on a, seemingly, unrelated website, simply because that user is likely to have the ability and willingness to pay more for that product.

Motivated by several anecdotal reports [Sam11, Daw11, Mat12], in 2013 we started an investigation on airlines websites in order to detect price discrimination [10] for their tickets. Airline tickets are an attractive target for price

discrimination due to the highly volatile nature of their prices. An airline could, for instance, recognize the same user over time, e.g., through the *use of third-party cookies*, and gradually increase the price of a specific ticket. The user is likely to attribute the price increase to flight congestion (many people buying tickets for that specific flight) and buy the ticket at an increased price, out of fear that the price will increase even more.

In January 2014, we performed [VNBJ14] a three-week long experiment: we queried 25 most popular airlines twice a day, with 66 user profiles, from two different geographical locations simultaneously resulting to a total of over 130,000 queries. The 66 profiles included combination of features of (i) various browsers and operating systems, (ii) 3 types of consumer profiles based on cookies collected from affluent, budget, and flight comparer websites, (iii) different cookie settings (with/without first/third party cookies), and (iv) two geographical locations.

Conclusion. We found that, at the time, airlines seem *not* to be employing any *systematic* price discrimination. At the same time, we show how difficult it is to establish cause and effect for airline prices and we make available the set of prices that we collected so that it can be used on future research on the topic of price discrimination [VNBJ14].

4.2 Browser Fingerprinting

The Web environment today has become very rich and dynamic. While first web browsers supported simple static HTML pages, new Web APIs, HTML5 and CSS3 made it possible to include a large variety of content into web applications. Moreover, the Web today is usable on a diverse variety of devices, that use various operating systems, that run diverse web browsers and mobile applications. Figure 4.6 demonstrates diversity of software and hardware of today's devices[2].

Thanks to such diversity, *browser fingerprinting*[3] became more and more efficient at uniquely identifying and tracking Web users. Browser fingerprinting is rooted into the origin of the Web, and client-side software, such as browsers and apps, and have been sharing device-specific information in its origin to improve user experience, therefore protection from fingerprinting is a very complex problem. A definition of browser fingerprinting that we use in this manuscript have been proposed by Laperdrix in his Phd thesis [Lap17]:

> A **browser fingerprint** is a set of information related to a user's device from the hardware to the operating system to the browser and its configuration.

[2]This figure is a slightly modified version of figure produced by Pierre Laperdrix.
[3]We use the terms "browser fingerprint" and "device fingerprint" interchangeably.

Figure 4.6: **Visualisation of hardware and software diversity**. Numerous types of devices, operating systems, browsers and applications introduce variety in browser fingerprinting.

> **Browser fingerprinting** refers to the process of collecting information through a web browser to build a fingerprint of a device. Via a simple script running inside a browser, a server can collect a wide variety of information from public interfaces called Application Programming Interface (API) and HTTP headers.

An API is an interface that provides an entry point to specific objects and functions. While some APIs require a permission to be accessed like the microphone or the camera, most of them are freely accessible from any JavaScript script rendering the information collection trivial. Contrarily to other identification techniques like cookies that rely on a unique identifier (ID) directly stored inside the browser, browser fingerprinting is qualified as completely *stateless*.

4.2.1 Survey on Browser Fingerprinting

Browser fingerprinting has been a subject of numerous studies in the last decade, starting from now well-known works of Mayer [May09] who noticed that differ-

ences in browsing environments could be exploited by a remote server to identify users, and Eckersley [Eck10c], who found that 83.6% out of 470,161 collected fingerprints were unique. The field has since exploded as researchers kept on finding new fingerprinting attributes and evaluating how robust and unique they are, such as Canvas API [api15, MS12, AEE+14a], WebGL [api18b, MS12, CLW17], Web Audio API [api18a, EN16b], and many others.

To systematize knowledge in the field of browser fingerprinitng, we have written the first and only survey so far [13] on the *research performed in the domain of browser fingerprinting*, while providing an accessible entry point to newcomers in the field. We explain how this technique works and where it stems from. We analyze the related work in detail to understand the composition of modern fingerprints and see how this technique is currently used online. We cover browser fingerprinting literature on the following topics:

- history of browser fingerprinting and all studied fingerprinting attributes,
- measures to evaluate uniqueness of fingerprinting attributes,
- adoption of fingerprinting and techniques used to detect fingerprinting,
- a taxonomy of defense mechanisms by categories, benefits and drawbacks,
- discussion on challenges in science, technology, business and legislation.

While interested reader can find all the technical details in the survey [13], two aspects of it deserve further analysis and critical evaluation, that we further discuss in Section 4.2.3.

4.2.2 Fingerprinting Web users by browser extensions

Since the web browser is the tool people use to navigate through the Web, privacy research community has studied various forms of *browser fingerprinting*. Fingerprinting of users' devices is similar to physical *biometric traits* of people, where only physical characteristics are studied. Similar to previous demonstrations of user uniqueness based on their behavior [OCJ12, AÁC15], *behavioral characteristics* – such as browser settings and the way people use their browsers – can also help to uniquely identify Web users. For example, a user installs web browser extensions she prefers, such as AdBlock [AdBa], LastPass [Las] or Ghostery [Gho] to enrich her Web experience. Also, while browsing the Web, she logs in her preferred social networks, such as Gmail [Gma], Facebook [Fac] or LinkedIn [Lin].

Detecting Web browser extensions. In recent works, Sjösten et al. [SVAS17], Starov and Nikiforakis [SN17] and Sánchez-Rola et al. [SSB17] explored complementary techniques to detect browser extensions. These works were focused on the technical mechanisms to detect extensions, but what was not studied is *how browser extensions contribute to uniqueness of users at large scale.*

Linus [Lin16] showed that some social websites are vulnerable to the "login-leak" attack that allows an arbitrary script to detect whether a user is logged in

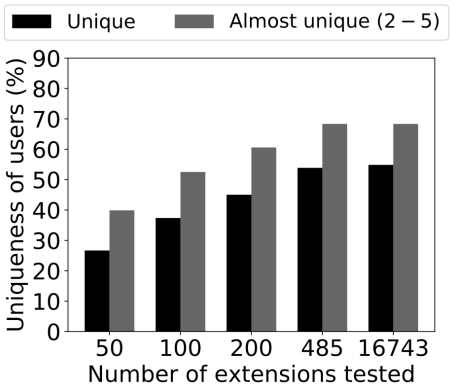


Figure 4.7: **Results of general fingerprinting algorithm [12].** Testing 485 carefully selected extensions provides a very similar uniqueness result to testing all 16,743 extensions.

a vulnerable website. However, it was not studied *whether Web logins can also contribute to users' uniqueness.*

Large-scale study of uniqueness by browser extensions. We performed the first large-scale study of user uniqueness based on browser extensions and Web logins [12], collected from more than 16,000 users who visited our website. Our experimental website identifies installed Google Chrome [Goo] extensions via Web Accessible Resources [SVAS17], and detects websites where the user is logged in by methods that rely on URL redirection and CSP violation reports. Our experimental website was able to detect the presence of 13K Chrome extensions, (approximately 28% of all free Chrome extensions at the time of the experiment). We also detect whether websites where the user is logged in out of 60 websites.

Uniqueness of users based on browser extensions and Web logins. We discovered that 55% of users that have installed at least one detectable extension are unique; 20% of users are unique among those who have logged into one or more detectable websites; and 89% are unique among users with at least one extension and one login. Moreover, we discover that 23% of users could be uniquely identified by web logins, even if they disable JavaScript.

We furthermore show that browser extensions and web logins can be exploited to fingerprint and track users by only checking a limited number of extensions and web logins. We have applied an advanced fingerprinting algorithm [GAC16] that carefully selects a limited number of extensions and logins. Fig. 4.7 shows uniqueness of users when a limited number of extensions is tested. The last column shows that 54.86% of users are unique based on all 16,743 detectable

extensions. However, by testing 485 carefully chosen extensions we can identify more than 53.96% of users. Besides, detecting 485 extensions takes only 625ms.

Evaluating the privacy dilemma. Given our results, one could argue that "the more privacy extensions you install, the more unique you are". In our work [12], we evaluate the trade-off between increase of users' uniqueness with the number of privacy extensions she installs; against the privacy gain (computed as third-party cookies blocked) of the privacy-preserving extensions such as Ghostery [Gho] and Privacy Badger [PB]. We further show that some of these extensions increase user's unicity and can therefore contribute to fingerprinting, which is counter-productive.

Conclusion. Our work [12] illustrates, one more time, that user anonymity is very challenging on the Web. Users are unique in many different ways in the real life and on the Web. This work shows that users are also unique in the way they configure and augment their browser, and by the sites they connect to. Unfortunately, although uniqueness is valuable in society because it increases diversity, it can be misused by malicious websites to fingerprint users and can therefore hurt privacy.

Another important contribution of this work is the definition and the study of the trade-off that exists when a user decides to install a "privacy" extension, for example, an extension that blocks trackers. We argue that these "privacy" extensions are very useful, but they should be included by default in all browsers. "Privacy by default", as advocated by the new EU privacy regulation, should be enforced to improve privacy of all Web users.

4.2.3 Challenge: How to measure uniqueness of browser fingerprints?

Several measures have been proposed to evaluate uniqueness of fingerprints: evaluating percentage of unique users, anonymity sets and entropy-based measures. In this section we take the simplest example: percentage of unique users.

Several studies aimed at *estimating how unique are the users based on their browser fingerprints at large scale*. This evaluation should shed light on the effectiveness of user recognition and tracking by browser fingerprinting. Recently, three such large scale studies were published: Panopticlick [Eck10c], AmIUnique [LRB16b] and Hiding in the Crowd [GBLB18]. Panopticlick [Eck10c] was the first to measure browser fingerprinting at a large scale. This study is based on 470,161 fingerprints in 2010 and they authors found that 83.6% of fingerprints were unique[4]. AmIUnique [LRB16b] is based on 118,934 fingerprints collected

[4]Even without Flash or Java plugins, that at the time were very popular and if enabled,

in 2016, and similarly to the results of Panopticlick, 89.4% of them were unique. AmIUnique study, however, revealed that some of the browser attributes, such as plugins and fonts, that were the most unique features in 2010, were not so useful in 2016. Additionally to the browser attributes from Panopticlick, AmIUnique has tested newly discovered attributes, such as HTML5 canvas [api15]. Hiding in the Crowd [GBLB18] analyzed 2,067,942 fingerprints collected on one of the top 15 French websites. Their findings provide a new layer of understanding to the domain as 33.6% of fingerprints from their dataset were unique.

	Panopticlick [Eck10c] (2010)	**AmIUnique [LRB16b] (2016)**		**Hiding in the Crowd [GBLB18] (2018)**	
	Desktop	Desktop	Mobile	Desktop	Mobile
Number of fingerprints	470,161	105,829	13,105	1,816,776	251,166
Unique fingerprints	94.2%	89.4%	81%	35.7%	18.5%

Table 4.3: **Uniqueness measured by three studies of browser fingerprinting [13].** Each study is based on a different sample of users in size and bias, measuring slightly different fingerprinting features, and over different periods of time in different years. The results differ significantly.

Table 4.3 shows that different studies find rather different results as to the uniqueness, and hence possibility to track online, based on browser fingerprinting. Therefore, several research questions need to be answered: Why the results in these studies are so different? What are the different design choices in these studies that could make an effect on the outcome? How should we, as a community, measure browser fingerprinting and how to compare different studies?

Choice of fingerprinting features. It is rather intuitive that depending on the features studied, the fingerprint may turn out to be more or less unique. For example, since Flash plugins is no longer supported, recent studies don't include it as a fingerprinting feature. As a demonstration, Laperdrix in his PhD thesis [Lap17] compared uniqueness results of fingerprints with and without Flash based on 60,617 fingerprints that contained Flash in 2017 and found that absence of Flask makes users less unique.

increased uniqueness to 94.2%)

Size of the dataset. The first answer is the sample size of the dataset used in the experiments. Indeed, it's rather expected that the bigger is the dataset, the less unique are the users in it. While no studies have analysed this phenomena in details, we have made a preliminary evaluation in our work on uniqueness based on browser extensions [12], in order to compare our results with related works at the time [SSB17, SN17]. Figure 4.8 shows % of unique users in our dataset of 7,643 users while simulating previous studies with 204 [SSB17] and 854 [SN17] users. We clearly see that with bigger dataset comes smaller ratio of unique users. By exploring this comparison, we raise a fundamental question: *What is the "right" size for the dataset?*

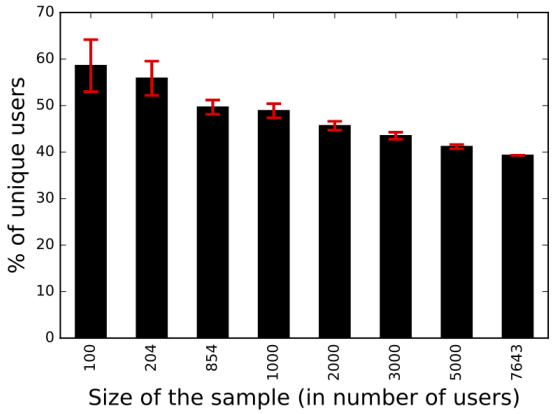


Figure 4.8: **Uniqueness of Chrome users based on their extensions only vs. number of users [12].** 204 is the number of users used in [SSB17] and 854 the number of users considered in [SN17]

Bias in the dataset. The second criteria that impacts the results is bias of the studied audience. It is a fundamental question in such studies that needs to be carefully analysed: how should we, computer science researchers, reach general audience for our measurements? Recent studies go for various options. First, some works [SSB17, LRB16a, 12] address *students, colleagues, and technical forums* – and such sample is biased towards Internet-savvy users. Second option is to use collaborative tools, such as *Amazon Mechanical Turk*. Such tools are known to provide audience that is not representative, since users often Indian and US-based users, who are willing to invest their time into surveys or experiments. They might even use a different system or browser that they use for normal everyday life. Finally, some studies *integrated browser fingerprinting scripts into popular websites* [GBLB18]. Such approach needs to be carefully evaluated from ethical point of view, as well as legal aspects in privacy and data protection in

the country of the researchers (where data processing takes place) and of the users, from whom the data has been collected.

Duration of an experiment. Browser fingerprints change quickly over time: Vastel et al. [VLRR18] showed that out of 1,905 studied browser instances, 50% changed their fingerprints in less than 5 days, 80% in less than 10 days. It means that if researchers collect the users' fingerprints for a long period of time, a big portion of them might appear to be unique just because for the many users the fingerprint has changed. Therefore, to take a "snapshot" of users' fingerprints, researchers need to collect many of them in a short time period. This requirement influences the choice of sampling and bias in the dataset (see previous paragraphs).

Time frame and location of the users. Depending on the software and devices the studied population is using, the results of uniqueness of browser fingerprints may differ significantly. For example, back in 2010, when the first measurement study was conducted by Eckersley [Eck10b], very relatively few people in the US had smartphones, while today almost everyone has one. In the same way, location of tested users impacts the results – fingerprints of users in countries with a more limited number of software and hardware providers are much less diverse than in countries with a big variety of software and devices.

Recognizing returning visitors. Finally, researchers must ensure that they recognize users that come back to their experimental tool. This is particularly important in case of experimental websites such as `https://amiunique.org/`, where the same tech-savvy users come back to check uniqueness of their browser fingerprint. Since fingerprints change significantly over a short period of time [VLRR18], researchers need to employ mechanisms to recognize users that come back several times to their experimental websites. As such, Hiding in the Crowd study [GBLB18] opted for storing only one fingerprint per user (and store a cookie to recognise the user when they come back). Let's analyse this design choice. On one hand, if the user never deletes the cookie, then their old fingerprint will be compared to the fingerprints of other users, which are newer. In practice, their fingerprint may changed, and became even more similar to those of other users (e.g., due to the same system updates that other users received), but the website won't take it into account. Therefore, the diversity this study observes will seem bigger than it is in reality. On the other hand, if the user continuously deletes their cookies and often visits an experimental website, then their fingerprint (that doesn't change much) will be counted many times. This will provide a smaller diversity of fingerprints. As a result, the choice about recognition of returned visitors impacts the outcome of the study and either

under- or overestimates the actual diversity and uniqueness of fingerprints.

Conclusion. To measure the ground truth about diversity and uniqueness of users' browser fingerprints, the community should not use experimental websites, but rather browser extensions, such as in the latest work of Vastel et al. [VLRR18]. The design of the study should be critically analysed with respect to the (non-exhaustive) list of parameters we discuss in this section. As a research community, need to reflect on the design of such studies to be able to compare them (to the new studies as well) in a long term. The community needs to come up with a list of "ingredients" for a solid measurement strategy of uniqueness of browser fingerprints at scale.

4.2.4 Challenge: How to compare prevalence of browser fingerprinting?

Various works studied adoption of browser fingerprinting on the Web. We analysed [13] techniques used to measure adoption between the first work by Nikiforakis et al. in 2013 [NKJ+13b] until the work of Al-Fannah et al. in 2018 [ALM18]. We then show that all these studies were uncomparable due to a number of reasons, explained below. Table 4.4 from [13] provides an overview of the five major studies of browser fingerprinting prevalence.

Cookieless Monster. In 2013, Nikiforakis et al. [NKJ+13b] crawled up to 20 pages for each of the the Alexa top 10,000 sites to look for fingerprinting scripts from the three following companies: BlueCava, Iovation, ThreatMetrix. They discovered 40 sites (0.4%) making use of these companies' fingerprinting code.

FPDetective. The same year, Acar et al. [AJN+13a]. performed a much larger crawl by visiting the homepages of top Alexa 1 million websites and 25 links of 100,000 Alexa websites. FPDetective study was the first to measure adoption of fingerprinting scripts without relying on a known list of tracking scripts as they directly looked for behaviors related to fingerprinting activities. They found 404 sites out of 1 million performing JavaScript-based font probing and 145 sites out of 10,000 performing Flash-based font probing.

The Web Never Forgets In 2014, Acar et al. [AEE+14a] measured adoption of Canvas [api15] fingerprinting on homepages of 100,000 Alexa websites. The authors instrumented the browser to intercept calls and returns to Canvas related methods, and tried to remove false positives by a set of rules [AEE+14a, Section 3.1]). They found 5542 sites out of 100,000 with Canvas fingerprinting.

Table 4.4: **Overview of five major studies measuring adoption of browser fingerprinting on the web [13].** Each study defined "fingerprint" as a set of varied features across studies. The sites crawled and the year varies. More importantly, each detection method was different and tailored to the studied fingerprinting features.

	Fingerprinting techniques detected	**Sites crawled**	**Prevalence**	**Detection method**
Cookieless Monster [NKJ+13b] (2013)	Detection of 3 known fingerprinting libraries	10K sites (up to 20 pages per site)	0.4%	Presence of JS libraries provided by BlueCava, Iovation and ThreatMetrix.
FPDetective [AJN+13a] (2013)	JS-based and Flash-based font probing	1M sites (homepages) 100K sites (25 links per site) for JS 10K (homepages) for Flash	0.04% (404 of 1M) for JS-based 1.45% (145 of 10K) for Flash-based	Logging calls of font probing methods. A script that loads more than 30 fonts or a Flash file that contains font enumeration calls is considered to perform fingerprinting.
The Web Never Forgets [AEE+14a] (2014)	Canvas fingerprinting	100K sites (homepages)	5.5%	Logging calls of canvas fingerprinting related methods. A script is considered to perform fingerprinting if it also checks other FP-related properties.
1M Alexa study with OpenWPM [EN16b] (2016)	Canvas fingerprinting, canvas-based font probing, WebRTC and AudioContext	1M sites (homepages)	1.4% for canvas fingerprinting 0.325% for canvas font probing 0.0715% for WebRTC 0.0067% for AudioContext	Logging calls of advanced FP-related JavaScript functions.
10K Majestic study [ALM18] (2018)	17 attributes (including OS, screen, geolocation, IP address among others)	10K sites (homepages)	68.8%	Data leaving the browser must contain at least one of the 17 monitored attributes.

1 million Alexa study with OpenWPM In 2016, Engelhardt and Narayanan released the OpenWPM platform [rep18] and performed an analysis of the Alexa top 1 million sites to detect and quantify emerging online tracking behaviours [EN16b]. Their findings provide more accurate results than in the past as they instrumented extensively a very high number of JavaScript objects to build a detection criterion for each known fingerprint technique [EN16b, Section 3.2]. Out of 1 million websites, they found 14,371 sites performing canvas fingerprinting, 3,250 sites performing canvas font fingerprinting, 715 sites performing WebRTC-based fingerprinting, and only 67 sites performing AudioContext fin-

gerprinting. These numbers are much higher than what was reported in previous studies.

10K Majestic study Finally, in 2018, Al-Fannah et al. crawled the top Majestic 10,000 websites and recorded what was sent out by the browser [ALM18]. Their definition of fingerprinting is much broader and inclusive than the other studies presented in this section. A website is deemed to be performing fingerprinting if at least one attribute out of a list of 17 is present in the recorded payloads. They identified 6,876 (68.8%) websites as performing fingerprinting which is a much higher number than what was reported in the past. 84.5% of them are third parties and the authors identified in total 284 attributes that can be used for fingerprinting.

Conclusion. Our analysis demonstrates the complexity of detecting fingerprinting, and that various studies relied on different set of websites, used various detection methods, made measurement in different years and obtained different results[5]. Therefore, policy makers and non-technical audience need to be aware that every reported result must be taken into account with all the technical details in mind. As a computer science research community, we need to converge on the methodology to detect fingerprinting in order to be able to evaluate and compare various studies on browser fingerprinting.

[5]More insights into technical challenges in measuring adoption of browser fingerprinting can be found in [13, Section 3.4]

Chapter 5

GDPR & ePrivacy Compliance

Web advertising relies on continuous data collection and tracking that allows advertising companies and data brokers to profit from processing a vast amount of data associated to Web users. The ePrivacy Directive 2002/58/EC [ePD02], amended in 2009 (ePD)[1] [ePD09] made it mandatory to collect user's *consent* before any access or storage of non-mandatory data, such as cookies and other tracking technologies. Since May 2018, the General Data Protection Regulation (GDPR) [GDPR] redefined the rules on what a *legally compliant consent* means. In case of websites, consent is usually presented in the form of *consent pop-ups* that should provide a meaningful choice to accept or reject such data collection. Today, EU users encounter such pop-ups on almost every website.

The Dream Website users would like to be provided with clear and comprehensive information and being able to control [ePD09, Art.5(3)] the usage of tracking technologies by websites and third parties. Users would like to refuse tracking which is unnecessary for the website to function properly, and do so in appropriate settings, which renders their consent *free, unambiguous, informed* and *specific* [GDPR, Art.4(11), Art.7]. Consent should be asked *prior* [GDPR, Art.6] to any data collection, and the user should be able to *revoke* it at any time [GDPR, Art.7(3)]. Moreover, users would like to have access to the data collected by third parties via tracking technologies [GDPR, Art.15].

The Reality Web tracking is inherent to the website design, and appears before any interaction with consent pop-ups [14, LSKR16, LGN18, PKM19]. Though prevalence of consent pop-ups on websites increases over time [DUL+19, HWB20, vAWN19], such pop-ups often only collect consent and do not actually prevent unwanted tracking [TTG+17, TTBM19, SRDK+19, 16]. Moreover, many consent pop-ups integrate manipulative design strategies to coerce the user towards acceptance [DUL+19, LK15, VFG+19, NLV+20, 16, 20]. Additionally, when exercizing their Subject Access Rights on websites, users don't manage to obtain any data collected via third-party cookies [15].

[1] ePrivacy Directive is known among computer scientists and website developers as "cookie law". The upgrade of the ePD into an ePrivacy Regulation is currently under discussion.

5.1 Application of GDPR and ePrivacy Directive to Web tracking technologies

In this section, we introduce the following EU Data Protection laws: GDPR and ePrivacy Directive, and how they are applied to Web tracking technologies. This section mostly includes background knowledge from our legal journal publication [19]. The contributions of this paper are further presented in Section 5.2.

Personal data. Websites, and more generally, web services must be operated in compliance with the data protection principles. *Personal data* means any information relating to an identified or (directly or indirectly) identifiable natural person. In determining whether the information relates to an identifiable individual, website publishers need to consider any means that could reasonably be used by them or any third party to enable the identification of an individual [GDPR, Art. 4(1), Recital 26][WP136]. Examples of personal data include: IP addresses, user identifiers, URLs of the visited pages and other parameters that enable the user to be singled-out. Usage of cookies for storing identifiers are explicitly mentioned in [GDPR, Recital 30]:

> "Natural persons may be associated with online identifiers provided by their devices, applications, tools and protocols, such as internet protocol addresses, cookie identifiers or other identifiers. (...) This may leave traces which, in particular when combined with unique identifiers and other information received by the servers, may be used to create profiles of the natural persons and identify them".

Note that personal data do not consist only in the data originally collected via the web service, but also in any other information collected through other means and that can be linked to personal data collected through the web service. It also means any other information inferred that relates to an individual. The European Data Protection Supervisor (EDPS) declares that the use of browser fingerprinting can lead to a certain percentage of assurance that two different sets of data collected belong to the same individual [EDPS6]. Thus, the GDPR applies to data that can identify users (i.e. when identification of users is likely), whether they are meant or used to track the online activity of such users.

According to the GDPR, processing of personal data can be performed lawfully only if one of the six *legal basis* of processing applies [GDPR, Article 6(1)]:

1. the data subject has given *consent* to the processing of his or her personal data for one or more specific purposes;
2. processing is necessary for the *performance of a contract* to which the data subject is party or in order to take steps at the request of the data subject prior to entering into a contract;

3. processing is necessary for *compliance with a legal obligation* to which the controller is subject;
4. processing is necessary in order to *protect the vital interests* of the data subject or of another natural person;
5. processing is necessary for the *performance of a task carried out in the public interest* or in the exercise of official authority vested in the controller;
6. processing is necessary for the purposes of the *legitimate interests* pursued by the controller or by a third party, except where such interests are overridden by the interests or fundamental rights and freedoms of the data subject which require protection of personal data, in particular where the data subject is a child.

Applicable rules for consent: the ePrivacy Directive and the GDPR. The ePrivacy Directive (ePD) [ePD02, ePD09] prescribes in Article 5(3) that websites are mandated to obtain users' *informed consent* before using any kind of tracking technology. Article 2(f) and Recital 17 of the 2002 ePD [ePD02] define *consent* in reference to the one set forth in Directive 95/46/EC [D95-46], the GDPR predecessor. The GDPR points out the conditions for obtaining valid consent in Articles 4(11) and 7 [GDPR].

Our contributions in this chapter. For websites, consent for cookies and other tracking technologies is usually presented in the form of consent pop-ups. We further analyse conditions for valid consent in detail and present 22 low-level legal and technical requirements for valid consent pop-ups in the context of Web applications in Section 5.2. We then present our work on the automatic analysis of consent pop-ups and further work on consent digitally stored in the browser in Section 5.3. We then found out that the consent pop-ups we studied, implementing IAB Europe Transparency and Consent Framework (TCF), actually contain a possibility for website publishers and advertisers to rely on other legal basis than consent, called *legitimate interest* (legal basis number 6 in the list presented earlier in this section). Hence we study the legality of using legitimate interest in Section 5.4. Since consent pop-ups are interfaces between users and the website, by collaborating with researchers in design and in law, we evaluate four design strategies in consent pop-ups from both legal and UX perspectives and identify manipulative strategies – see Section 5.5. Finally, in Section 5.6 we analyse the GDPR Subject Access Rights [GDPR, Article 15], and detect the problems in user identification and in the exercise of the data subject requests when the user is tracked via third-party cookies.

5.2 Legal requirements for consent pop-ups and means to enforce them

Behind the scenes of almost any website lay pervasive, sophisticated and ubiquitous browsing tracking technologies that collect and share information on the behavior and preferences of users, which is later monetized in the data-driven society. *Consent pop-ups*[2], that ask for user's consent in almost every website a EU user visits, are a result of EU legislation: the ePD amended in 2009 [ePD09] and the GDPR [GDPR] that came into force in May 2018.

In practice, legal compliance of consent pop-ups within websites raises several important and practical questions that we investigated in our research [19]:

- *What makes a consent pop-up compliant with the EU Law?*
- *What kind of consent pop-ups website publishers, consent management providers, and other industrial players should build up to be compliant by design and by default?*
- *Moreover, how regulators can efficiently audit websites' consent pop-ups and tracking technologies to verify compliance?*

To answer these questions, it is no longer sufficient to be a legal expert, nor does it suffice to be a technical expert and detect cookies and other tracking technologies that are used on the audited website. To decipher the ePrivacy and GDPR requirements, we have set up an interdisciplinary team of a legal scholar and two computer scientists experts in Online Tracking. Our joint legal and technical expertise allowed us to deep dive and combine the knowledge of both areas – Data Protection Law and Computer Science. In our work, published in December 2020 in the interdisciplinary international journal of law, technology and society – "Technology and Regulation" [19]– we have provided a comprehensive legal and technical analysis on legal requirements, design and implementation of consent pop-ups.

Tracking technologies exempted of consent. In our analysis, we only refer to the use of Browser-based tracking technologies (BTT) that require consent. According to Article 5(3) of the ePD [ePD09], consent is not required when the purpose of trackers is:

Communication: used for the sole purpose of enabling the communication on the web; and

[2]Previous works give various names to this mechanisms, such as "cookie banner", "consent notice", or "consent banner". We will use the term "consent pop-up" in the rest of this manuscript because invisible data collection applies to various tracking technologies (not only cookies) and are often implemented in various types of pop-up mechanisms (not only banners).

Strict necessity: cookies and BTT strictly necessary to enable the service requested by the user: if BTT is disabled, the service will not work.

The 29 Working Party (29WP), now known as European Data Protection Board (EDPB), issued Guidelines (WP194) [WP212] giving advice on interpretations of these two exceptions from consent in the context of websites (considering browser cookies, but also extending to all BTTs):

- The *communication exemption* applies when the transmission of the communication is impossible without the use of the BTT (e.g. load-balancing cookie). Hence, using BTT to merely "assist" or "facilitate" the communication is insufficient.
- The *strict necessity exemption* involves a narrow interpretation. It means that the use of BTT must be restricted to what is strictly necessary (and hence essential) to provide a service explicitly requested by a user. Thus, using BTT that is *reasonably necessary* or *important* implies that the service provided by the website publisher would not function without the BTT. In this regard, when a website functional content includes BTT (for example, when a Customised Search Engine used on a website includes a tracking third-party cookie - see Section 4.1.3), then it is not enough to justify the strict necessity since the website owner has a different implementation choice that would work without BTT.

In our work, we analyse in details the Data Protection Authorities (DPAs) guidelines, as well as 29WP (now EDPB) and propose an analysis of purposes that need consent and purposes exempted of consent, according to these sources (for details, see [19, Table 5]).

22 fine-grained requirements for a legally valid consent. We first identified 7 high-level requirements for valid consent from the GDPR and the ePrivacy Directive in the context of websites. By analysing them further and applying technical knowledge, the 7 requirements were further refined into 22 fine-grained and operational requirements for rendering a valid consent. Figure 5.1 shows an overview of these 22 low-level requirements. This result stems from an interdisciplinary effort that not only identifies legal requirements but also provides a technical analysis of the possibility to detect violations automatically and at scale. Each of the 22 requirements contains:

- a deep legal analysis of all comprised legal sources, including binding sources (GDPR, ePrivacy, case-law), non-binding sources (EDPB and DPA regulatory guidelines) complemented with our own interpretation based on legal and technical experience gleaned in the team;
- a simple explanation and examples of violations of each requirement on a

Requirements		Assessment	Sources at low-level requirement		
High-Level Requirements	**Low-Level Requirements**	**Manual (M), Technical (T) or User study (U)**	**Binding**	**Non-binding**	**Interpretation: Legal (L) or Computer Science (CS)**
Prior	R1 Prior to storing an identifier	M (partially) or T (partially)	√	√	-
	R2 Prior to sending an identifier	T (partially)	-	-	CS
Free	R3 No merging into a contract	M (fully) or T (partially)	√	√	-
	R4 No tracking walls	M (fully)	-	√	-
Specific	R5 Separate consent per purpose	M (fully)	√	√	-
Informed	R6 Accessibility of information page	M (fully) or T (partially) together with U	-	√	-
	R7 Necessary information on BTT	M (fully) or T (partially)	√	√	-
	R8 Information on consent banner configuration	M (fully) or T (partially)	-	√	-
	R9 Information on the data controller	M (fully) or T (partially)	√	√	-
	R10 Information on rights	M (fully) or T (partially)	√	√	-
Unambiguous	R11 Affirmative action design	Combination of M and T (partially)	√	√	-
	R12 Configurable banner	M or T (partially)	-	√	L
	R13 Balanced choice	M (fully)	-	√	L
	R14 Post-consent registration	T (partially)	-	√	CS
	R15 Correct consent registration	Combination of M and T (partially)	-	√	CS
Readable and accessible	R16 Distinguishable	M (fully) or T (partially)	√	√	-
	R17 Intelligible	U	√	√	-
	R18 Accessible	U	√	√	
	R19 Clear and plain language	U	√	√	-
	R20 No consent wall	M (fully) or T (partially)	-	√	L
Revocable	R21 Possible to change in the future	M (fully)	√	√	-
	R22 Delete "consent cookie" and communicate to third parties	Not possible	-	-	CS

Figure 5.1: **Requirements for a valid consent on consent pop-up [19].** High-level requirements are derived from GDPR [GDPR], ePrivacy Directive [ePD09] and CJEU case law. Low-level requirements are derived additionally from non-binding sources, such as EDPB and DPA guidelines, complemented by Legal (L) or Computer Science (CS) interpretation. Assessment of each requirement can be done manually (M), with CS tools (T), via user studies (U) or via a combination of them.

real world website;
- investigation on how to detect violations of each requirement most efficiently and at scale. We distinguish three ways to detect violations, and for some requirements, a combination thereof is needed:
  1. technically, by an expert using automated tools[3], or
  2. manually, relying only on a human operator, or
  3. performing user studies to evaluate the perception of end users.

Our analysis shows that to verify legal compliance of a consent pop-up on a website, an auditor must not only be familiar with the technical tools to detect every cookie and tracking technology, but also know the data processing purpose of each detected technology. Moreover, the auditor should be able to find out how consent is stored in the browser (see further discussion on problems of storage of consent in the browser in Section 5.3 and on the challenge of consent standardization in Section 5.5.2). Finally, be able to run user studies to validate, for example, whether the information is provided in clear and plain language. Such combination of legal, technical and social studies knowledge is extremely hard to achieve.

Conclusion. Our work [19] is the first to systematize legal and technical knowledge in Online Tracking and End-to-end Consent areas and to propose practically-minded parties, such as DPAs, privacy NGOs, designers[4] or researchers, a clear roadmap on how to verify compliance of a website consent pop-up with the EU legislation. This work also sets a guide to data protection specialists to help website providers achieve the desired compliance with the GDPR and ePrivacy. We argue that many current practices – such as shared consent among websites – are problematic and policy makers should converge on the rules for consent and advance with a unified framework of the ePrivacy Regulation that will be applied similarly to GDPR: uniformly to all EU member states and all organizations providing online services or tracking EU residents.

5.2.1 Challenge: Defining purposes exempted of consent

In our work [19], we analysed purposes for data collection (and hence of tracking technologies) that require consent and those that are exempted of consent [19, Table 5]. This analysis was based on the opinions and official guidelines of various EU Data Protection Authorities (DPAs), European Data Protection Board (EDPB) and European Data Protection Supervisor (EDPS).

[3] In our technical analysis, we assume that the expert has access only to the client-side code accessible in a Web browser. DPAs can have additionally access to the server-side code during audits.

[4] We further discuss the role of design in consent pop-ups in Section 5.5.

We noticed that the question *Which purposes are exempted of consent?* is answered differently by various DPAs. That is, there is no consensus on when tracking technologies can be used without consent, and hence auditing of websites becomes complex and depends on the interpretation of consent exemption rules by each individual DPA. In particular, we observed that:

- the biggest divergence in DPA's opinions lays in the purpose of *analytics*, also called as *audience measurement*;
- DPAs often disagree on which purposes must rely on consent and which ones can be used with a different legal basis, such as *legitimate interest* [GDPR, Art.6(1)(f)];
- DPAs and EDPB often take into account such properties of cookies and BTT as duration (session or persistent) and scope (first or third party), while these properties do not determine the purpose and can be easily forged (see our feedback to the Italian DPA [24, Section 4]).

The ePrivacy Regulation, that is currently under discussion, could unify the rules on consent exemption, but its latest text is still so high level that it leaves a lot of space for DPAs to interpret and set up their own rules on consent exemption, and even to do it on a case-to-case basis, which makes automatic, large-scale and precise auditing impossible.

Conclusion. If Data Protection Authorities, and hence EDPB, provide unified rules on purposes that require consent and the ones exempted of consent, computer scientists could propose automated tools that would speed up the auditing process of consent compliance and help DPAs to efficiently enforce the GDPR and the upcoming ePrivacy Regulation.

5.3 Do consent pop-ups respect user's choice?

Although many research efforts took place after the GDPR to detect and analyze cookie banners and their impact on tracking technologies and on the users, no study has analyzed what actually happens behind the user interface of cookie banners yet. It is unclear how to meaningfully compare the interface of the banners shown to the users with the actual consent that banners store and transmit to the third parties present on the website.

In our work with a computer scientist and a researcher in law [16], that was done at the same time as our analysis on legal requirements for valid consent [19], we investigated new questions regarding the storage of consent in the user's browser. This work has lead to new requirements R14 "Post-consent registration" and R15 "Correct consent registration" of the legal requirements on consent, shown in Figure 5.1 in Section 5.2. In this work [16], we explored the following questions, that correspond to the legal requirements on consent from

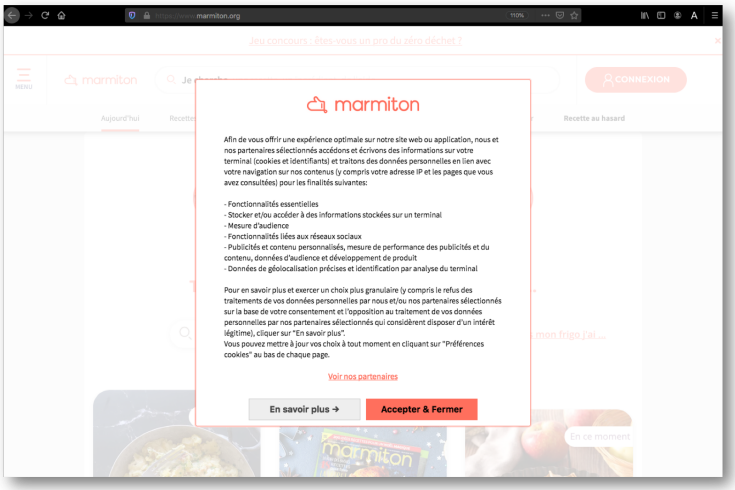


Figure 5.2: **A typical consent pop-up** on a popular cooking website `marmiton.org` provided by a Consent Management Provider "Didomi" that implements IAB Europe's Transparency & Consent Framework (TCF) [II18]. Screenshot made on 10 November, 2020.

Figure 5.1:

- *Do consent pop-ups silently register a positive consent even if the user has not made their choice (R14)?*
- *Do consent pop-ups actually respect user's choice made in the user interface (R15)?*
- *Do they nudge the user to accept everything by pre-choosing a positive consent (R11)?*

Answering such questions, ensuring a proper functionality and legal compliance of a consent pop-up is usually left to the website publisher and is completely obscure for the website visitor.

IAB Europe Trasparency & Consent Framework (TCF) and Consent Management Providers (CMPs). In reaction to the GDPR, the European branch of the Interactive Advertising Bureau (IAB Europe), an advertising business organization, produced the Transparency and Consent Framework (TCF) [II18, IAB20a] to structure the practices of actors of the tracking and advertisement industry regarding consent collection. Notably, they introduced the notion of *Consent Management Providers (CMPs)* – actors in charge of collecting consent from the end-user, and redistributing this consent to advertisers. Figure 5.2 shows a typical example of a consent pop-up implemented by a CMP.

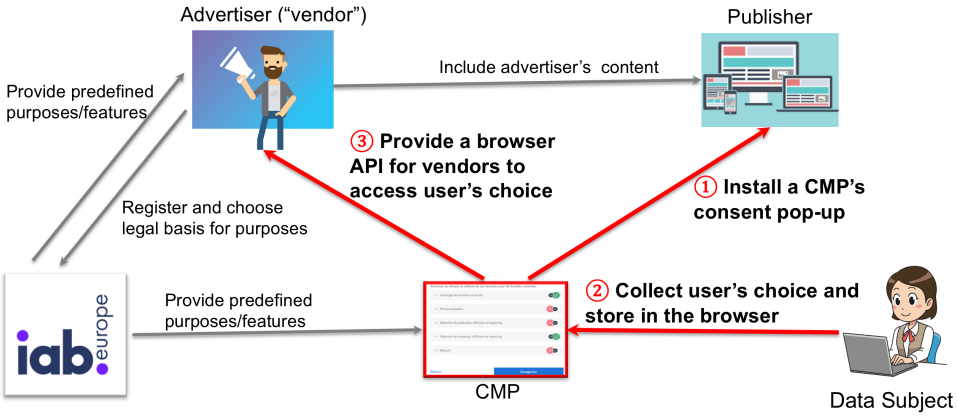


Figure 5.3: **Actors under IAB Europe TCF ecosystem [21].** IAB Europe, Advertisers (called "vendors"), Consent Management Providers (CMPs), Publishers, Data Subjects. The IAB Europe defines the purposes and features that are shown to users. Registered vendors declare purposes and legal basis and the features upon which they rely. CMPs provide consent pop-up, store the user's choice as a browser cookie, and provide an API for advertisers to access this information.

Open specification of consent storage by the IAB Europe TCF. To standardise[5] the technical implementation of these consent pop-ups, the IAB Europe TCF (currently on version 2.0 [IAB20a]) was developed to preserve the exchange of data within the advertising ecosystem, which now requires being able to demonstrate how, when, from whom, and on which legal basis that data is collected. The actors in this ecosystem are IAB Europe, advertisers (called "vendors"), Consent Management Providers (CMPs), publishers, and data subjects (see Figure 5.3).

When a user makes their choice in a IAB Europe TCF consent pop-up, they consent that their data will be invisibly collected and further used with respect to several purposes, pre-defined by IAB Europe TCF (we discuss all the purposes in Section 5.4). Alongside with the purposes, the consent pop-ups include a list of around 700 third party advertisers, called "vendors"[6], that rely on the allowed purposes to collect users' data as they browse the website.

Semi-automated analysis of consent pop-ups. Thanks to the open specification of the TCF, we performed [16] the first systematic comparison of the consent chosen by the users and the consent stored by the CMPs, which is further

[5] Standardization is used within the meaning of streamline at scale consent implementation.
[6] All such advertisers are listed in the IAB TCF Global Vendors' List [IAB20b]. This list contains 717 vendors as of 30 March 2021.

transmitted to third-party advertisers present on a website. Within our analysis of consent, we were able to measure both the GDPR and the ePD compliance of consent pop-ups implemented in the TCF. We note that the responsibility for the suspected violations is shared between the publishers and the CMPs.

First, we designed an automatic method to detect the presence of a consent pop-up developed by a Consent Management Provider (CMP). We automatically detected 1,426 websites with such banners. We also developed and used a methodology to intercept the consent stored in the browser. By analyzing the content of consent, we were able to provide information to Web users about the companies behind CMPs on the website they visited.

Second, by collaborating with a legal scholar, one of the co-authors [16], we thoroughly analyzed the GDPR, the ePrivacy Directive and other legal sources to identify four potential legal violations specific to cookie banners: *Consent stored before choice*, *No way to opt out*, *Pre-selected choices* and *Non-respect of choice*. These violations then helped us improve and extend the list of requirements for valid consent [19].

Detected GDPR and ePD violations on 54% of websites. We have then developed a method to evaluate regulatory compliance of websites. We quantify the identified suspected violations on 1,426 websites by automatic-, semi-automatic crawls and manual detection. By analyzing cookie banners' design and consent stored in the browser on a subset of 560 websites, we have found at least one suspected violation in 304 out of 560 websites (54%), presented in Table 5.1.

Problem of shared consent. We have then measured the problem of escalation of *shared consent* between CMPs. The TCF allows different CMPs and publishers to rely on each other's consent, set in a shared cookie. We observed that 3 websites store a positive consent before any user action in the shared cookie, while 20 websites store a positive consent in a shared cookie even if the user has explicitly opted out. Such invalid consent can be reused by any CMP and publisher and therefore escalates non-compliance to other websites. Finally, we quantified third-party requests that transmit consent and that belong to known third-party tracking services. We observe that various third parties receive consent with third-party requests, where the origin of consent does not necessarily match the CMP present on the website. Such consents are set before user action on 69 websites and despite user refusal on 38 websites. We observe that the number of third-party tracking requests increases both after positive consent and after refusal.

Suspected violation	**Description of a violation**	**# websites analysed**	**# websites with a violation**
Consent stored before choice	The cookie banner stores a positive consent before the user has made their choice in the banner. Therefore, when advertisers request for consent, the cookie banner responds with the positive consent even though the user has not clicked on a banner and has not made their choice yet.	1,426	141 (9.9%)
No way to opt out	The pop-up does not offer a way to refuse consent. The most common case is a pop-up simply informing the users about the site's use of cookies	560	38 (6.8%)
Pre-selected choices	The pop-up gives user a choice between one or more purposes or vendors, but some of the purposes or advertisers are pre-selected: pre-ticked boxes or sliders set to "accept".	508	236 (46.5%)
Non-respect of choice	The consent pop-up stores a positive consent in the browser even though the user has explicitly refused consent.	508	27 (5.3%)

Table 5.1: **Detected GDPR and ePrivacy suspected violations** on websites with IAB Europe TCF consent pop-ups via semi-automatic analysis in September 2019 [16].

Tools for auditing. To measure compliance, we have designed two tools. *Cookinspect* [Mat20] is a Selenium- and Chromium-based crawler which automatically and semi-automatically visits websites, logs stored consent and intercepts transmission of consent to third parties. *Cookie Glasses* [Mat19] is a publicly available browser extension for Google Chrome and Firefox that allows users to detect a CMP that implements a TCF banner and see if their choice is correctly transmitted to advertisers by CMPs.

Impact on society. The European Center for the Defense of Digital Rights called "None Of Your Business" (NOYB) NGO, led by Max Schrems, issued three formal complaints with the French Data Protection Authority (CNIL) in December 2019. NOYB used our "Cookie Glasses" extension that decodes the consent stored in the browser after the user interacts with the cookie banner. These complaints addressed 3 popular French websites: CDiscount, AlloCine and VanityFair and third party ad targeting companies Facebook, AppNexus and PubMatic who received cookies on the 3 French websites unlawfully [fDR].

5.4 Purposes in IAB Europe's TCF: which legal basis and how are they used by advertisers?

An advertiser willing to be involved in the IAB Europe TCF [II18, IAB20a] (see Figure 5.3) and wishing to appear in CMP-based consent pop-ups must register within the TCF. At the moment of registration, an advertiser must select one or more of the predefined purposes for data processing. These purposes are presented to website users in consent pop-ups while collecting their consent (and are often presented in the second layer of the pop-up). For each purpose, advertisers must choose a legal basis for processing: consent or legitimate interest. Note that *legitimate interest* is, just like consent, one of the six legal basis for processing personal data [GDPR, Article 6(1)] that we presented in the Section 5.1.

Figure 5.4 presents a screenshot of the registration form advertisers use to subscribe to IAB Europe TCF. The choice of the purposes and their legal basis hold *strong legal compliance implications* – both on the advertisers, but also on the publishers side, as the latter include third-party resources in their websites.

According to Article 8 of the Charter of Fundamental Rights of the European Union [Eur], personal data must be processed fairly for *specified purposes* and on the basis of a *lawful ground*. Article 5(1)(b) of the GDPR [GDPR] predicates the *"Purpose Limitation"* principle which mandates personal data to be collected for specified, explicit and legitimate purposes. Identifying the appropriate legal basis that corresponds to the purpose of the processing is of essential importance.

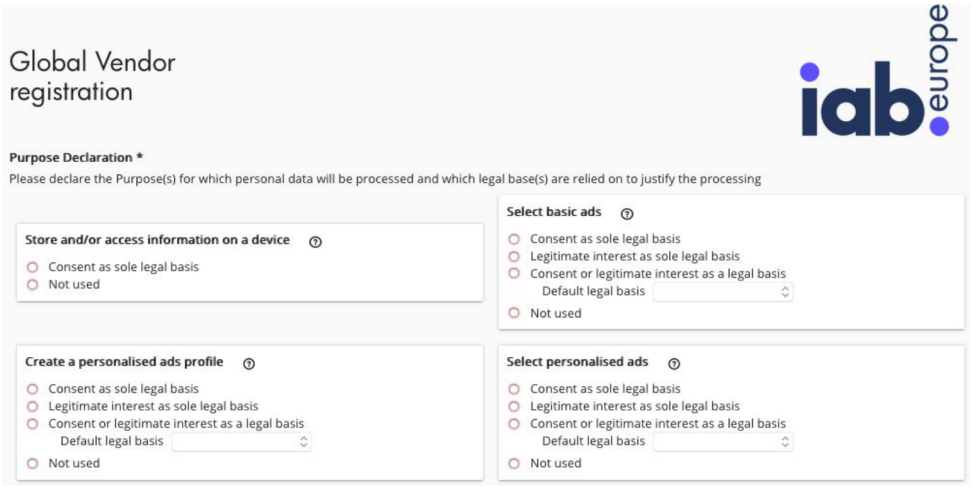


Figure 5.4: **A registration process for an advertiser ("vendor")** under IAB Europe TCF v2.0. Screenshot made on 19 October 2020 at `https://register.consensu.org/`.

Which purposes require consent? In our work [17] we identified the legal requirements for defining purposes based on the GDPR, the 29 Working Party (now EDPB endorsed) and Data Protection Authorities guidance that helped us to answer the following questions: *"Does a purpose satisfy the requirements of the purpose specification principle?"* and *"Which is the legal basis for a specific purpose?"*. We then analyse the purposes defined in IAB Europe's TCF versions 1.1 and 2.0, and in Table 5.2 presents our result on whether such purposes comply with the legal requirements and which purposes should rely on consent. Further legal analysis can be found in our paper [17].

Declaration of purposes and their legal basis by advertisers. We then collected data about all advertisers registered in both versions v1.1 and v2.0 of the IAB Europe TCF in Januray 2020. Our goal was to measure which purposes are used by advertisers and which legal bases advertisers have declared for each purpose.

We identified in Table 5.2 that purposes 3 and 5 of TCF v1.1 ("Ad selection, delivery, reporting" and "Measurement") require consent. Figure 5.5a details the results for each individual purpose in IAB Europe TCF v1.1 [II20a]. Overall, we detect a particularly worrisome number of advertisers: 175 (31%) and 199 (35%) advertisers respectively rely on legitimate interests for purposes 3 and 5 that require consent according to our analysis in Table 5.2.

Figure 5.5 shows that hundreds of advertisers rely on legitimate interest for purposes that instead should rely on consent. Figure 5.5b renders an analysis of

Table 5.2: **Purposes defined in IAB Europe's TCF v1.1 and v2.0 [17].** The "Allowable Lawful Bases by TCF" column indicates the possible legal basis (Consent or Legitimate Interest (LI)) according to the official documentation guidelines of IAB Europe TCF v2.0 [IAB, p .25] In "Our Analysis" we indicate questionable cases with brackets, because the purpose description is not specific enough and does not allow us to conclude with certainty. Cases marked with "?" denote lack of information, where legal basis cannot be derived.

(a) Purposes (TCF v1.1)

Purpose number	**Purpose name**	**Allowable Lawful Bases by TCF**	**Legal basis according to our Analysis [17]**
1	Information storage and access	-	(Consent)
2	Personalisation	-	?
3	Ad selection, delivery, reporting	-	Consent
4	Content selection, delivery, reporting	-	(Consent)
5	Measurement	-	Consent

(b) Purposes (TCF v2.0)

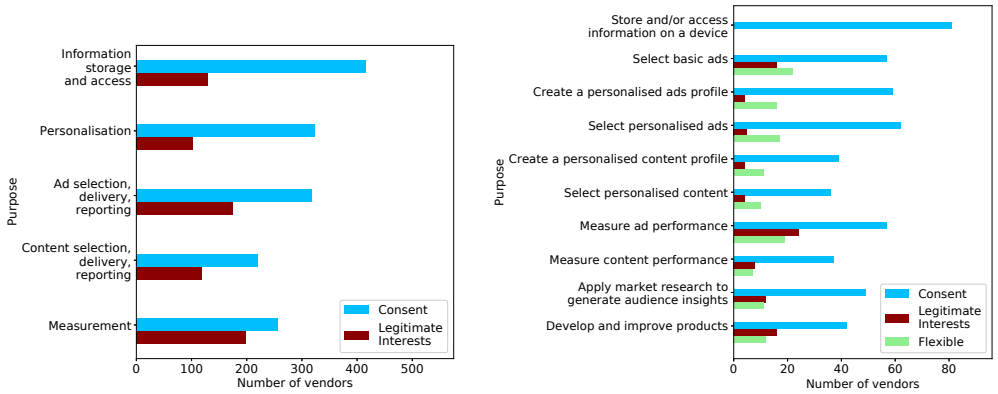
1	Store and/or access information on a device	Consent	(Consent)
2	Select basic ads	Consent, LI	Consent
3	Create a personalised ads profile	Consent, LI	Consent
4	Select personalised ads	Consent, LI	Consent
5	Create a personalised content profile	Consent, LI	(Consent)
6	Select personalised content	Consent, LI	(Consent)
7	Measure ad performance	Consent, LI	Consent
8	Measure content performance	Consent, LI	Consent
9	Apply market research to generate audience insights	Consent, LI	Consent
10	Develop and improve products	Consent, LI	?

(c) Special purposes (TCF v2.0)

1	Ensure security, prevent fraud, and debug	LI	(LI)
2	Technically deliver ads or content	LI	?

purposes for the Global Vendor List of v2.0 [II20b]. The number of advertisers registered in this version was smaller in January 2020, but we still saw that a significant portion of advertisers use legitimate interest for purposes that require consent. For example, 17% advertisers rely on legitimate interest for purpose 2 ("Select basic ads"), and 25% advertisers do it for purpose 7 ("Measure ad performance"), while our legal analysis in Table 5.2 demonstrated that purposes 2 and 7 require consent. It is also notable that 32% of advertisers use "flexible purposes" for at least one purpose thus allowing publishers to change the legal basis for such purposes [II19].

Figure 5.5: **Purposes and legal basis of processing [17]** declared by the registered advertisers in IAB Europe's Transparency and Consent Framework v1.1 and v2.0, measured in January 2020.



(a) Purposes, TCF v1.1 (version 183) [II20a]

(b) Purposes, TCF v2.0 (version 20) [II20b]

Purpose specification for third-party cookies. In our work [18], we investigated the legal compliance of purposes for 20,218 third-party cookies outside of the IAB Europe TCF framework. Surprisingly, only 12.85% of third-party cookies have a corresponding privacy/cookie policy where a cookie is even mentioned. Overall, we find out that purposes declared in cookie policies *do not comply with the purpose specification principle in 95% of cases* in our automatized audit.

Conclusion. Our work [17, 18] demonstrates the persistence of the advertising industry using in non-compliant (with GDPR [GDPR] and ePD [ePD09]) methods for tracking and profiling, bundled in often complex and vague presentation of purposes. The importance of this is further underlined by the extended prior work, guidance, as well as enforcement actions and court decisions in the field.

5.5 Dark patterns, manipulative design strategies and their legality in consent

We have identified several design strategies for consent pop-ups, that are not officially violating the legal requirements of valid consent but yet are questionable as to their manipulative power on Web users [19].

In a joint work with a researcher in design and researchers in law [20], we have further approached four design strategies that correspond to various forms of *dark patterns* and *user manipulation* and yet are not explicitly banned by EU Data Protection Authorities (DPAs).

Data Collection and Framing. We relied upon our previously collected dataset (see Section 5.3) that contains 560 websites of French, Italian or English-speaking EU countries. From this dataset, we focused on locating a range of potentially manipulative design exemplars, using recorded videos or screenshots of the consent experiences to support a manual and collaborative analysis of their design and text. In total, we reviewed recordings from over 50 sites and extensively analyzed the design and users' means of interaction with the consent banners on these websites. While reviewing other recent and relevant literature on ethical issues in the design of consent banners (e.g., [NLV+20, MB20, SNGS20]), we identified four main phases in the consent task flow (Figure 5.6):

1. the initial framing as a user enters the site;
2. the presentation of configuration options to accept or select more precise consent options;
3. the means of accepting the configuration options; and
4. the ability to ultimately revoke consent.

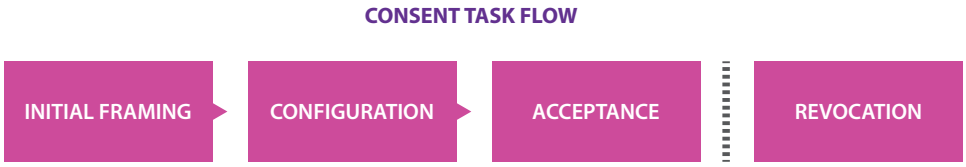


Figure 5.6: The task flow of the consenting process by phase [20].

Within this task flow, we worked as a research team to identify four different combinations of design choices that were represented in the dataset and raised productive ethical dilemmas when viewed from multiple disciplinary perspectives. Because our main goal in this paper is to examine the complexity of these design outcomes and not to identify how common these patterns occur "in the wild," we used the dataset as a source of inspiration and departure rather than as a means

of conducting a content analysis or other inductive form of inquiry.

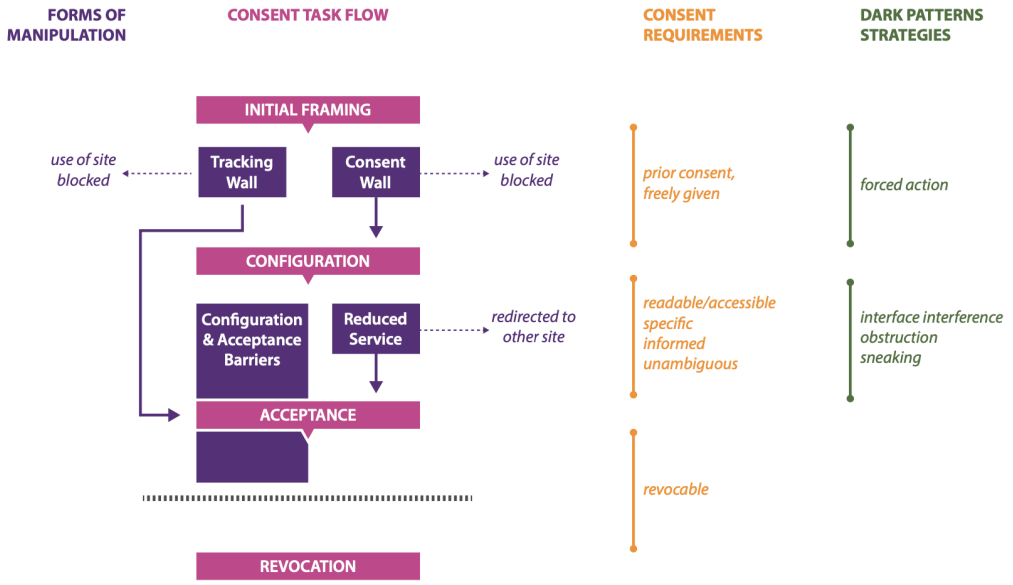


Figure 5.7: **Flowchart describing the forms of manipulation** we observed in our dataset in relation to the consent task flow, compliance requirements, and dark patterns strategies [20].

Interaction criticism approach. We use an *interaction criticism* [Bar11] approach to analyze and reflect upon these design strategies from multiple perspectives: 1) design choices evident in the consenting process artifact itself; 2) the possible experience of the end user; 3) the possible intentions of the designer; and 4) the social milieu and impact of this milieu on the other three perspectives [20]. We consider the four following design strategies that did not receive much attention so far analysed in the literature, especially not from an interdisciplinary approach combining design, law and computer science perspectives. The consent task flow highlighting these four strategies, potentially violated consent requirements and detected dark pattern strategies are shown in Figure 5.7.

Consent wall blocks access to the website until a user expresses their choice regarding consent. The design choice allows a user to select between acceptance and refusal; however, the concrete use of the website is blocked until a choice has been made. In our work on legal requirements for valid consent [19], we explain that this strategy is likely at odds with *freely given* and *readable and accessible* requirement of valid consent (see R20 of Figure 5.1 in Section 5.2).

Tracking wall is an instance of a consent wall, however with more detrimental consequences to the user. In addition to blocking access to the website until the user makes their choice, a tracking wall gives the user only one option: to consent and accept any terms offered by the site, without any possibility to refuse. In the legal domain, a tracking wall is also called a "cookie-wall" or "take it or leave it" choice [BKBH17]. Differently from a *consent wall*, a tracking wall cannot result in a *reduced service* because the only option the user has is merely to accept consent in order to access the website. Tracking wall has been recognized as violating the law by several DPAs (see R4 in Figure 5.1 and [19, Section 5.2.1] on further discussion), yet there is no consensus across all EU countries as to the legality of this practice.

Reduced service refers to the practice of a website offering reduced functionality—for example, allowing a user access to only limited number of pages on a website—based on their consent configuration options. In the scope of consent on websites, reduced service is a result of the user refusing consent in some or all of the proposed privacy configurations. An extreme case of a reduced service occurs when a website fully blocks access because the user refuses some of the privacy configurations. This strategy has not been analysed in previous work, and hence deserves further investigation and guidance from the DPAs and EDPB.

Manipulative design choices and other configuration and acceptance barriers, such as imposition of hierarchies, aesthetic manipulation (known as "interface interference"), toying with emotion, reading order manipulation or hidden information all constitute manipulative design strategies coercing the user towards acceptance. Such strategies is under thorough inspection by several EU DPAs. We explicitly list requirements R11 "Affirmative action design", R12 "Configurable banner", and R13 "Balanced choice" in Figure 5.1 to propose first rules towards protecting Web user from such manipulation.

Conclusion. All such manipulative strategies, often called *dark patterns*, are coercing the user towards acceptance, often under a fear of not accessing the service if the user rejects. We were the first to analyse them in depth and show that most of the design strategies appear in practice due to different intentions of designers, website owners, consent pop-up providers and regulators [20]. Policy makers should provide more guidelines on which strategies indeed violate the EU law and which ones need further investigation.

5.5.1 Challenge: Roles of CMPs under GDPR: controllers or processors?

Although recent work, including ours [16, 17], has started to address the complex technical and legal aspects of the IAB Europe TCF ecosystem [HWB20, NLV+20, PC20, DUL+19], *neither prior work nor court decisions* have so far discussed the role of the CMPs. In our feedback to the European Data Protection Board (EDPB), we have underlined the complexity and raised questions about the legal role of Consent Management Providers (CMPs), whether they are *data controllers* and/or *data processors* [BS20]. However, it is currently unclear what the role of CMPs is under the GDPR, and consequently what their legal requirements and responsibilities are.

In our recently accepted publication [21], with a co-author, expert in EU Data Protection Law, we investigate if and when CMPs can be considered a *data controller* – i.e., an actor responsible for determining the purposes and means of the processing of personal data [GDPR, Art. 4(7)] – or a *data processor* – i.e., an actor which processes personal data on behalf of the controller [GDPR, Art. 4(8)]. Discerning the correct positioning of CMPs is crucial since compliance measures and CMPs liability depend on their accurate characterization [GDPR, Recital 79]. To determine the role of CMPs under the GDPR, we answer the following research questions:

- *When are CMPs processing personal data?*
- *When do CMPs act as data processors?*
- *When do CMPs act as data controllers?*
- *What are the liability implications of CMPs acting as controllers?*

Our argumentation is based on: (1) legal analysis of binding legal sources (GDPR and case-law) and relevant data protection guidelines from the European Data Protection Board and Data Protection Authorities, (2) document analysis of the IAB Europe TCF, and (3) empirical data gathered on our own website by deploying Quantcast and OneTrust – the two most popular CMPs in the EU, found respectively on 38.3% and 16.3% of the websites with a EU or UK TLD analyzed by Hils et al. [HWB20].

Although CMPs are explicitly designated as *data processors* by the IAB Europe TCF specifications [IAB20a], we analysed four functional activities of CMPs that enables their qualification as *data controllers*. In our work [21], we included a technical description of such activities followed by a legal analysis. These activities refer to:

- Including additional processing activities in their tools beyond those specified by the IAB Europe;
- Scanning publisher websites for tracking technologies and sorting them into purpose categories;

- Controlling third-party vendors included by CMPs;
- Deploying manipulative design strategies in the UI of consent pop-ups - we underline when such strategies could violate legal requirements for valid consent (Sections 5.2 and 5.5).

Conclusions. Our work [21] started an investigation into the legal role of Consent Management Providers under GDPR. We provided the first elements that can help DPAs and EDPB to reason about the role and activities of these organisations. However, more in-depth analysis and legal guidance is required for the website publishers who are often unaware of the decision making power and capabilities of CMPs.

5.5.2 Challenge: Standardization of consent

The amount of website audits, complaints and regulatory enforcement actions related to consent - both from DPAs and the Court of Justice - have substantially increased. However, all these actions depend on complex manual analysis of websites that includes detection of a tracking technology; identification of a purpose of each tracking technology; analysis whether consent is required; evaluation of consent compliance requirement. Recent guidelines by the EDPB and DPAs clarify and strengthen the rules for consent requirements and propose best practices, but the implementation of consent on websites still diverges. Several questions hence need to be answered that were object of debate in a panel discussion at Computers, Privacy and Data Protection (CPDP 2021) conference [CPDP], the top-level event for legal privacy experts:

- *What are the next steps for a streamlined, auditable and scalable consent?*
- *How can legal and technical experts help to devise a compliant-by-design consent?*
- *Would standardization of consent help?*
- *What concrete building blocks need to be defined by policy-makers and the legislator?*

As of today, there is no unique standard to represent consent collected on website in a digital form. IAB Europe has its own standard that, as we show in Sections 5.3, 5.4 and 5.5, triggers a number of open legal problems either in the TCF standard or in its implementation.

Foundational elements for standardized consent. For consent to be standardized, it should contain a number of elements that are mandatory to make it streamlined, auditable and scalable. In our contribution to the public consultation to the CNIL [22], we underline four standardized elements: *purposes,*

categories/types of data collected, data controllers, and trackers that facilitate the data collection. These building blocks are needed to be defined by Data Protection Authorities or the EDPB for a standardisation of consent:

- *an exhaustive list of standardized purposes*: the purposes should be human-understood and machine-readable. Additionally, a taxonomy of purposes would allow to reason about them (inclusions, implications and generalisations). Previous works, such as P3P [P3P] have proposed an initial structure for taxonomies of purposes, and proprietary taxonomies have been recently designed by private companies, such as the one of TrustArc [TArc].
- it should be clear *which standardized purposes should rely on which legal basis (consent or other)*: with standardized purposes, DPAs could set stricter rules on which purposes should be used with consent and which are definitely exempted of consent and under which conditions (see discussion in Section 5.2.1).
- *categories (e.g., personal, sensitive) and types (more fine-grained description) of data collected should be standardized.* Each tracker will then be able to contain a complete description of collected or processed data type.
- *standardized presentation of data controllers and data processors*, such as the company name, domain URL, privacy policy link, etc. Some recent attempts have been made by computer scientists to collect this information [Lib18, dOL], but more guided approach is needed to enforce such rules.
- *standardized naming convention and description for trackers* from a predefined name convention and labels that helps determine the nature and owner of the tracker since identifying ownership of trackers has been shown to be a complex task [BARW16].
- some *elements of a standardized design interface* or clear black list of design patterns for a consent pop-up should be defined to avoid user manipulation and dark patterns (see Section 5.5).
- *standardized storage of consent* in a digital form should be provided. Without a standard representation of consent storage, it is very complex, time-consuming and not scalable to assess many of the compliance requirements (see R1, R2, R14, R15, R21, R22 in Figure 5.1 of Section 5.2).

One purpose per tracker. In a standardized consent, in a given consent scope, *each tracker should have only one standard purpose*, categories/types of data collected, list of data controllers and a legal basis applied to it. This would facilitate the auditing process for DPAs. The same standard can be used *to inform the users about the purposes of all the trackers* present on a website independently of the legal basis.

Consent within Web browser interface. If consent is standardized, it can be also set in *a web browser interface*. Web browsers could provide an interface to register a consent pre-defined by the user, using the same standard as other actors. This will allow the browser to exchange the consent with the data controllers automatically. Some of such proposals have been made, such as *Global Privacy Control* [GPC], and require further legal evaluation in the scope of EU Data Protection law.

Conclusion. Standardized consent would provide many benefits to website publishers, consent banner providers, DPAs and other auditing organisations. It would provide legal certainty, save time and resources, and at the same time harmonize approaches across the EU. Standardization of consent would enable automatic and scalable auditing of compliance.

5.6 GDPR access rights and third-party cookies

When Web users are continuously track by third parties on the Web (often without a valid consent - see Sections 5.3 and 5.5), users would like to know what information is collected and processed about them via online trackers and their collaborations with various parties (see Section 4.1). Moreover, users would like to know what data is collected about them when are tracked via stateless tracking techniques, such as browser fingerprinting (Section 4.2)

Chapter 3 of the GDPR [GDPR] defines various rights for the users' (called "data subjects") and aims at protecting their personal data. Every European Data Protection Authority (DPA) provides advises, explanations and recommendations to help data controllers (website owners in the context of this manuscript) implement such rights for the users. However, the GDPR does not provide any prescriptive requirements on *how to authenticate a data subject request*. The lack of guidance and interpretation makes it hard to respect GDPR in practice: data subjects are denied from their access right, the lawfulness of the processing and the derived legal rights are impossible to check in practice. We therefore conducted an interdisciplinary study [15] with two legal scholars and two computer scientists in order to understand the goals and tensions between that data subjects (Web users) and data controllers (website owners).

View of the data subject (user). Every data subject would like to benefit from the rights specified in GDPR, but still wonders: *How do I exercise my access right? How do I prove my identity to the controller?* These questions are critical to build trust between the data subject and the controller. The data subject is concerned with threats like *impersonation* and *abusive identity check*:

Impersonation occurs when a malicious party attempts to abuse the subject access request (SAR) by impersonating a subject to a controller.

Abusive identity check can occur when a data controller is too curious and verifies the identity of a subject by asking irrelevant and unnecessary information like an electricity bill or government issued documents.

View of the data controller (website owner). Symmetrically, every data controller needs to know how to proceed when they receive an access request: *Is the request legitimate? What is necessary to identify the subject's data?* These concerns aggravate when controllers deal with indirectly-linked identifiers, such as IP addresses, or when they have no prior contact with data subjects, as in *Google Spain* [GSp]. Most of all, data controllers want to avoid data breaches, as it can result in legal proceedings and heavy fines. Such consequence occurs in two cases: *(i)* the data controller releases data to an illegitimate subject, or *(ii)* he releases data of a subject A to a legitimate subject B.

Tension between data subject and data controller. All these questions concern the authentication procedure between the data subject and the controller. They both share a common interest in holding a strong authentication procedure to prevent impersonation and data breaches. On one hand, the data subject should not provide too much personal information that could compromise her privacy. On the other hand, the controller needs to ask the appropriate information to identify the subject's data without ambiguity. Therefore, there is clearly a tension during this authentication act between the controller, who tries to get as much information as possible, and the data subject who wants to provide as little as possible. Plausibly, subject access rights can probably increase the incidence of personal records being accidentally or deliberately opened to unauthorised third parties [Cor16].

Evaluating authentication procedures. In our work with two legal scholars [15], we study *the tension during the authentication between the data subject and the data controller*. We first evaluate the threats to the SAR authentication procedure and then we analyze the recommendations of 28 DPAs of European Union countries. We have then evaluated the authentication procedure when exercising the access right of the *50 most popular websites and 30 third-party tracking services*. Several popular websites require to systematically provide a national identity card or government-issued documents to authenticate the data subject. Among third-party tracking services, 9 of them additionally to third-party cookies demand other personal data from the data subjects, like the identity card or the full name. We explain that such demands are not justified because additional information can not prove the ownership of the cookie.

Conclusion. We then provide guidelines to Data Protection Authorities, website owners and third party services on how to authenticate data subjects safely while protecting their identities, and without requesting additional unnecessary information (complying with the data minimization principle). In the conclusion of our work [15] we provide some initial ideas on how digital identifiers can be redesigned in such a way that data subjects can safely exercise their rights and data controllers can safely respond to SAR requests and be compliant with the GDPR.

5.6.1 Challenge: Proof of ownership for cookies is needed

When exercising SAR on third party services, users have to follow complex procedures requiring understanding of legal text [UTD+18]. In our work [15], we have discovered that Subject Access Request (SAR) procedures that implement GDPR access rights, though compliant with the law, are suffering from a number of security vulnerabilities and implementations that can lead to data breaches. The problem is that the same technology that is used to track users (such as cookies) was not initially designed to identify users.

Cookies (as many other digital identifiers) are chosen by third parties on the server side and stored on the client side (in the browser). Cookies can subsequently be changed in the browser or, if not properly secured, even stolen by network attackers. As a result, cookies are not sufficiently secure to be used to reclaim the data associated with a Subject Access Request and to prove data ownership. For this reason, today many third parties refuse SAR and don't provide access to users' data.

Is there a solution? The problem with cookies and similar stateful tracking technologies is that they are at odds with the legal rights provided by the GDPR due the way cookies are generated and used today. To introduce a change: (i) cookies should be generated on the client side. By using cryptographic schemes an *easy and safe mechanism for a user to claim cookie ownership* can be designed; (ii) the current HTTP protocol should be modified to account for the possibility to have cookies and other identifiers generated on the client-side.

Finally, the adoption by the browser vendors, website owners and third parties of such a mechanism is crucial and should be supported by policy makers to ensure that technologies allowing proof of ownership are indeed compliant with the current and future EU Data Protection Laws.

Chapter 6

A short conclusion

Our work covers a broad area of privacy protection of Web users, and contains both theoretical and practical contributions. There are numerous challenges to be resolved in order to advance privacy protection on the Web. In this short conclusion, we first remind the challenges we raised in the manuscript, underline the importance of transdisciplinary collaboration within several disciplines, and then briefly describe upcoming legal and technical challenges in the next few years.

The important *challenges that still need to be resolved* by both the legal and computer science research communities, as well as by policy makers:

§3.1.4 The need for a new quantitative information flow measure.
§3.2.3 The research community needs to use one unified framework to compare information flow monitors.
§4.1.3 How to protect users from Web tracking without breaking Web applications?
§4.2.3 How to measure uniqueness of browser fingerprints?
§4.2.4 How to compare prevalence of browser fingerprinting?
§5.2.1 Defining purposes exempted of consent.
§5.5.1 Roles of CMPs under GDPR: controllers or processors?
§5.5.2 Standardization of consent.
§5.6.1 Proof of ownership for cookies is needed.

The need for transdisciplinary collaborations. Our work demonstrates the need for an *transdisciplinary collaboration between technical and legal researchers* to work hand-in-hand before new laws are written. For example, all contributions in Chapter 5 were accomplished thanks to long-term collaboration with a *legal scholar*, where an transdisciplinary dialog has taken place upon regular visits over the last 3 years. Technical experts need also to be invited to the discussion of new legal proposals or their amendments.

Additionally, since EU Data Protection law contains *consent* as one of the legal basis[1], many data controllers rely on it when processing personal data. The usage

[1] See a full list of legal basis in Section 5.1.

of consent puts an enormous pressure on the user interacting with the system. Therefore, the design of interfaces of systems is of crucial importance since it determines the legality of consent based interactions. Hence, an *transdisciplinary collaboration between computer science, law and researchers in UI and design* is also very important to ensure that the Web user is not manipulated when making important decisions regarding her privacy protection choices.

Technical expertise and research of computer scientists is of crucial importance for EU Data Protection Authorities and other policy making organisations, such as the EU Commission or the Organisation for Economic Co-operation and Development (OECD), who conduct audits and/or provide highly-important guidelines to protect privacy on the Web.

Upcoming legal and technical challenges. The EU will soon finalise the ePrivacy Regulation [ePr21], that is currently under a trialogue discussion between the European Parliament, EU Commission and the Council of the European Union. From a *legal perspective*, the latest proposal of the ePrivacy Regulation has a number of open questions and challenges with respect to the latest technological developments and need a careful evaluation by computer scientists, and also researchers in design. For example, current debate includes:

- the usage of *tracking- and cookie walls* (Section 5.5),
- exception from consent for the purpose of *audience measurement* (Section 5.2.1),
- whether consent can be expressed in the browser settings (Section 5.5.2).

From a *technical perspective*, third-party cookies will be deprecated from the Web in 2022, and Google proposed a new Privacy Sandbox [GSa]. This new proposal has been recently criticized by pro-privacy NGOs and organisations, such as Electronic Frontiers Foundation (EFF) [EFFG]. Therefore, the proposal needs to be carefully evaluated both from a technical perspective, as well as from a legal perspective as to its compliance with the EU Data Protection law.

Bibliography

[10] T. Vissers, N. Nikiforakis, N. Bielova, and W. Joosen. Crying Wolf? On the Price Discrimination of Online Airline Tickets. In *7th Workshop on Hot Topics in Privacy Enhancing Technologies (HotPETs 2014)*, 2014. https://hal.inria.fr/hal-01081034.

[11] D. F. Somé, N. Bielova, and T. Rezk. Control what you include! server-side protection against third party web tracking. In *International Symposium on Engineering Secure Software and Systems (ESSoS)*, volume 10379 of *Lecture Notes in Computer Science*, pages 115–132. Springer, 2017. https://hal.inria.fr/hal-01649547.

[1] N. Bielova. Survey on JavaScript security policies and their enforcement mechanisms in a web browser. *Special Issue on Automated Specification and Verification of Web Systems of Journal of Logic and Algebraic Programming (JLAP)*, 82(8):243 – 262, 2013. https://doi.org/10.1016/j.jlap.2013.05.001.

[12] 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 *Proceedings of the 2018 Workshop on Privacy in the Electronic Society (WPES) at ACM CCS'18*, pages 14–27. ACM, 2018. https://hal.inria.fr/hal-01921863.

[13] P. Laperdrix, N. Bielova, B. Baudry, and G. Avoine. Browser fingerprinting: A survey. *ACM Transactions on the Web (TWEB)*, 14(2):8:1–8:33, 2020. https://dl.acm.org/doi/10.1145/3386040.

[14] I. Fouad, N. Bielova, A. Legout, and N. Sarafijanovic-Djukic. Missed by filter lists: Detecting unknown third-party trackers with invisible pixels. *Proceedings on Privacy Enhancing Technologies (PoPETs)*, 2020, 2020. https://doi.org/10.2478/popets-2020-0038.

[15] C. Boniface, I. Fouad, N. Bielova, C. Lauradoux, and C. Santos. Security analysis of subject access request procedures - how to authenticate data subjects safely when they request for their data. In *7th Annual Privacy Forum, APF*, volume 11498 of *Lecture Notes in Computer Science*, pages 182–209. Springer, 2019. https://hal.inria.fr/hal-02072302.

[16] 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 *IEEE Symposium on Security and Privacy (IEEE S&P)*, 2020. https://hal.inria.fr/hal-03117294.

[17] C. Matte, C. Santos, and N. Bielova. Purposes in IAB Europe's TCF: which legal basis and how are they used by advertisers? In *Annual Privacy Forum, APF*, Lecture Notes in Computer Science, 2020. https://hal.inria.fr/hal-02566891.

[18] I. Fouad, C. Santos, F. Al Kassar, N. Bielova, and S. Calzavara. On Compliance of Cookie Purposes with the Purpose Specification Principle. In *2020 International Workshop on Privacy Engineering, IWPE*, 2020. https://hal.inria.fr/hal-02567022.

[19] C. Santos, N. Bielova, and C. Matte. Are cookie banners indeed compliant with the law? deciphering EU legal requirements on consent and technical means to verify compliance of cookie banners. *Technology and Regulation (TechReg)*, pages 91–135, 2020. https://doi.org/10.26116/techreg.2020.009.

[20] C. Gray, C. Santos, N. Bielova, M. Toth, and D. Clifford. Dark patterns and the legal requirements of consent banners: An interaction criticism perspective. In *ACM CHI Conference on Human Factors in Computing Systems (ACM CHI)*, 2021.

[21] C. Santos, M. Nouwens, M. Toth, N. Bielova, and V. Roca. Consent management platforms under the gdpr: processors and/or controllers? In *Annual Privacy Forum (APF'21)*, 2021. Accepted for publication, https://hal.inria.fr/hal-03169436.

[2] D. F. Somé, N. Bielova, and T. Rezk. On the content security policy violations due to the same-origin policy. In *Proceedings of the 26th International Conference on World Wide Web, (WWW 2017)*, pages 877–886. ACM, 2017. https://hal.inria.fr/hal-01649526.

[22] M. Toth, N. Bielova, C. Santos, V. Roca, and C. Matte. Contribution to the public consultation on the CNIL's draft recommendation on "cookies and other trackers", 2020. Research report, https://hal.inria.fr/hal-02490531.

[23] N. Bielova and C. Santos. Feedback to EDPB regarding Guidelines 07/2020 on the concepts of controller and processor in the GDPR, 2020. Research report, http:

//www-sop.inria.fr/members/Nataliia.Bielova/opinions/EDPB-contribution-controllers-processors.pdf.

[24] N. Bielova and C. Santos. Feedback to the Guidelines on the use of cookies and other tracking tools of the Italian Data Protection Authority, 2020. Research report, https://hal.inria.fr/hal-03079482.

[3] F. Besson, N. Bielova, and T. Jensen. Hybrid information flow monitoring against web tracking. In *IEEE Computer Security Foundations Symposium (CSF 2013)*, pages 240–254. IEEE, 2013. https://hal.inria.fr/hal-00924138.

[4] F. Besson, N. Bielova, and T. Jensen. Browser Randomisation against Fingerprinting: A Quantitative Information Flow Approach. In *Nordic Conference on Secure IT Systems (NordSec 2014)*, pages 181–196, 2014. doi: 10.1007/978-3-319-11599-3_11. https://hal.inria.fr/hal-01081037.

[5] N. Bielova and T. Rezk. Spot the difference: Secure multi-execution and multiple facets. In *21st European Symposium on Research in Computer Security, ESORICS*, volume 9878 of *Lecture Notes in Computer Science*, pages 501–519. Springer, 2016. https://hal.inria.fr/hal-01348192.

[6] F. Besson, N. Bielova, and T. P. Jensen. Hybrid monitoring of attacker knowledge. In *IEEE 29th Computer Security Foundations Symposium, (CSF'16)*, pages 225–238. IEEE Computer Society, 2016. https://hal.inria.fr/hal-01310572.

[7] N. Bielova and T. Rezk. A taxonomy of information flow monitors. In *Principles of Security and Trust - 5th International Conference, POST*, volume 9635 of *Lecture Notes in Computer Science*, pages 46–67. Springer, 2016. https://hal.inria.fr/hal-01348188.

[8] N. Bielova. Short paper: Dynamic leakage: A need for a new quantitative information flow measure. In *Proceedings of the 2016 ACM Workshop on Programming Languages and Analysis for Security, PLAS@CCS 2016*, pages 83–88. ACM, 2016. https://hal.inria.fr/hal-01409706.

[9] M. Ngo, N. Bielova, C. Flanagan, T. Rezk, A. Russo, and T. Schmitz. A better facet of dynamic information flow control. In *Web Programming, Design, Analysis, And Implementation (WPDAI 2018) alter-*

nate track of The Web Conference (WWW 2018), pages 731–739. ACM, 2018. https://hal.inria.fr/hal-01723723.

[AÁC15] J. P. Achara, G. Ács, and C. Castelluccia. On the unicity of smartphone applications. *CoRR*, abs/1507.07851, 2015. URL http://arxiv.org/abs/1507.07851.

[AC12] A. Askarov and S. Chong. Learning is change in knowledge: Knowledge-based security for dynamic policies. In *CSF'12*, pages 308–322. IEEE, 2012.

[ACPS12] M. S. Alvim, K. Chatzikokolakis, C. Palamidessi, and G. Smith. Measuring information leakage using generalized gain functions. In *CSF'12*, pages 265–279, 2012.

[AdBa] AdBlock Official website. https://getadblock.com/.

[adbb] Adblock Plus Official website. https://adblockplus.org/.

[AEE+14a] G. Acar, C. Eubank, S. Englehardt, M. Juarez, A. Narayanan, and C. Diaz. The Web Never Forgets: Persistent Tracking Mechanisms in the Wild. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, CCS '14, pages 674–689, New York, NY, USA, 2014. ACM. doi: 10.1145/2660267.2660347.

[AEE+14b] G. Acar, C. Eubank, S. Englehardt, M. Juárez, A. Narayanan, and C. Díaz. The web never forgets: Persistent tracking mechanisms in the wild. In *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security*, pages 674–689, 2014.

[AF10] T. H. Austin and C. Flanagan. Permissive dynamic information flow analysis. In *Workshop on Programming Language and analysis for security (PLAS'10)*, pages 3:1–3:12, 2010.

[AF12] T. H. Austin and C. Flanagan. Multiple facets for dynamic information flow. In *Proc. of the 39th Symposium of Principles of Programming Languages*. ACM, 2012.

[AJN+13a] G. Acar, M. Juarez, N. Nikiforakis, C. Diaz, S. Gürses, F. Piessens, and B. Preneel. FPDetective: dusting the web for fingerprinters. In *Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Security*, CCS '13, pages 1129–1140, New York, NY, USA, 2013. ACM. doi: 10.1145/2508859.2516674.

[AJN+13b] G. Acar, M. Juárez, N. Nikiforakis, C. Díaz, S. F. Gürses, F. Piessens, and B. Preneel. Fpdetective: dusting the web for fingerprinters. In *2013 ACM SIGSAC Conference on Computer and Communications Security (CCS'13)*, pages 1129–1140, 2013.

[ALM18] N. M. Al-Fannah, W. Li, and C. J. Mitchell. Beyond Cookie Monster Amnesia: Real World Persistent Online Tracking. In *Information Security - 21st International Conference, ISC 2018, Guildford, UK, September 9-12, 2018, Proceedings*, pages 481–501, 2018. doi: 10.1007/978-3-319-99136-8_26.

[api15] HTML Canvas 2D Context - W3C Recommendation 19 November 2015, 2015. https://www.w3.org/TR/2dcontext/.

[api18a] Web Audio API, 2018. https://www.w3.org/TR/webaudio/.

[api18b] WebGL - OpenGL ES for the Web, 2018. https://www.khronos.org/webgl/.

[AS07] A. Askarov and A. Sabelfeld. Gradual release: Unifying declassification, encryption and key release policies. In *S&P'07*, pages 207–221. IEEE, 2007.

[AWS+11] M. D. Ayenson, D. J. Wambach, A. Soltani, N. Good, and C. J. Hoofnagle. Flash cookies and privacy ii: Now with html5 and etag respawning. Technical report, Available at SSRN: https://ssrn.com/abstract=1898390orhttp://dx.doi.org/10.2139/ssrn.1898390, 2011.

[B12] N. Bielova. *A theory of constructive and predictable runtime enforcement mechanisms*. PhD thesis, University of Trento, 2011.

[BAK+18] M. A. Bashir, S. Arshad, E. Kirda, W. K. Robertson, and C. Wilson. How tracking companies circumvented ad blockers using websockets. In *Internet Measurement Conference 2018*, pages 471–477, 2018.

[Bar11] J. Bardzell. Interaction criticism: An introduction to the practice. *Interacting with computers*, 23(6):604–621, 2011. doi: 10.1016/j.intcom.2011.07.001.

[BARW16] M. A. Bashir, S. Arshad, W. Robertson, and C. Wilson. Tracing Information Flows Between Ad Exchanges Using Retargeted Ads. In *Proceedings of the 25th USENIX Security Symposium*, Austin, TX, August 2016.

[BKBH17] F. Borgesius, S. Kruikemeier, S. Boerman, and N. Helberger. Tracking walls, take-it-or-leave-it choices, the gdpr, and the eprivacy regulation. *European Data Protection Law Review*, 3:353–368, 2017.

[BKR09] M. Backes, B. Köpf, and A. Rybalchenko. Automatic discovery and quantification of information leaks. In *IEEE Symposium on Security and Privacy (S& P'09)*, pages 141–153, 2009.

[BLW05] L. Bauer, J. Ligatti, and D. Walker. Edit automata: Enforcement mechanisms for run-time security policies. *International Journal of Information Security*, 4(1-2):2–16, 2005.

[BM11] N. Bielova and F. Massacci. Do you really mean what you actually enforced? *International Journal of Information Security*, pages 239–254, 2011. URL http://dx.doi.org/10.1007/s10207-011-0137-2.

[Bod] K. Boda. Firegloves. http://fingerprint.pet-portal.eu/?menu=6.

[BS20] N. Bielova and C. Santos. Call for Feedback to the EDPB regarding Guidelines 07/2020 on the concepts of controller and processor in the IAB Europe Transparency and Consent Framework, 2020. http://www-sop.inria.fr/members/Nataliia.Bielova/opinions/EDPB-contribution-controllers-processors.pdf.

[BW18] M. A. Bashir and C. Wilson. Diffusion of User Tracking Data in the Online Advertising Ecosystem. In *Proceedings on Privacy Enhancing Technologies (PETS 2018)*, 2018.

[CLW17] Y. Cao, S. Li, and E. Wijmans. (Cross-)Browser Fingerprinting via OS and Hardware Level Features. In *24nd Annual Network and Distributed System Security Symposium, NDSS*, 2017.

[CMJL09] R. Chugh, J. Meister, R. Jhala, and S. Lerner. Staged information flow for Javascript. In *Proc. of the ACM SIGPLAN 2009 Conference on Programming Language Design and Implementation*, 2009.

[CMS09] M. R. Clarkson, A. C. Myers, and F. B. Schneider. Quantifying information flow with beliefs. *Journal of Computer Security*, 17(5):655–701, 2009.

[CNILc] CNIL: Délibération de la formation restreinte n° san-2020-008 du 18 novembre 2020 concernant la société CARREFOUR FRANCE. https://www.legifrance.gouv.fr/cnil/id/CNILTEXT000042563756. Accessed on 19 March, 2021.

[Cor16] A. Cormack. Is the Subject Access Right Now Too Great a Threat to Privacy? *European Data Protection Law Review*, 2(1):15–27, 2016.

[CPDP] Standard for consent: Still a dream or a soon-to-be reality?

[CS10] M. R. Clarkson and F. B. Schneider. Hyperproperties. *J. Comput. Secur.*, 18(6):1157–1210, 2010.

[CT06] T. M. Cover and J. A. Thomas. *Elements of Information Theory (2. ed.)*. Wiley, 2006.

[CUT+21] S. Calzavara, T. Urban, D. Tatang, M. Steffens, and B. Stock. Reining in the web's inconsistencies with site policy. In *Network and Distributed System Security Symposium (NDSS)*, 2021.

[D95-46] Directive 95/46/ec of the european parliament and of the council of 24 october 1995 on the protection of individuals with regard to the processing of personal data and on the free movement of such data. https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=celex%3A31995L0046.

[Daw11] A. Dawson. Save money on ryanair fares by... deleting your cookies?, 2011. URL http://www.bitterwallet.com/save-money-on-ryanair-fares-by-deleting-your-cookies/42133.

[DDNP12] W. De Groef, D. Devriese, N. Nikiforakis, and F. Piessens. Flowfox: a Web Browser with Flexible and Precise Information Flow Control. In *Proc. of the 19th ACM Conference on Communications and Computer Security*, pages 748–759, 2012.

[DdSC09] P. <D-d>Phung, D. Sands, and A. Chudnov. Lightweight self-protecting Javascript. In *Proc. of ACM Symposium on Information, Computer and Communications Security (ASIACCS'09)*, pages 47–60. ACM, 2009.

[Dis] Disconnect. disconnect-tracking-protection. https://github.com/disconnectme/disconnect-tracking-protection, accessed on 2019.07.16.

[Disc] Disconnect Official website. https://disconnect.me/.

[dOL] T. Libert. webxray domain owner list. https://github.com/timlib/webXray_Domain_Owner_List.

[DP10] D. Devriese and F. Piessens. Non-interference through secure multi-execution. In *Proc. of the 2010 Symposium on Security and Privacy*, pages 109–124. IEEE, 2010.

[DPS20] C. Deußer, S. Passmann, and T. Strufe. Browsing unicity: On the limits of anonymizing web tracking data. In *2020 IEEE Symposium on Security and Privacy (SP)*, pages 777–790, 2020. doi: 10.1109/SP40000.2020.00018.

[DUL+19] M. Degeling, C. Utz, C. Lentzsch, H. Hosseini, F. Schaub, and T. Holz. We Value Your Privacy ... Now Take Some Cookies: Measuring the GDPR's Impact on Web Privacy. In *Network and Distributed Systems Security Symposium*, 2019.

[Eck10a] P. Eckersley. How Unique is Your Web Browser? In *Proceedings of the 10th International Conference on Privacy Enhancing Technologies*, PETS'10, pages 1–18. Springer-Verlag, 2010.

[Eck10b] P. Eckersley. How unique is your web browser? In *Privacy Enhancing Technologies (PETs)*, pages 1–18, 2010.

[Eck10c] P. Eckersley. How Unique is Your Web Browser? In *Proceedings of the 10th International Conference on Privacy Enhancing Technologies*, PETS'10, pages 1–18, Berlin, Heidelberg, 2010. Springer-Verlag. URL http://dl.acm.org/citation.cfm?id=1881151.1881152.

[EDPS6] EDPS opinion on the proposal for a regulation on privacy and electronic communications (eprivacy regulation). https://edps.europa.eu/sites/default/files/publication/17-04-24_eprivacy_en.pdf.

[EFFG] Google's floc is a terrible idea. https://www.eff.org/deeplinks/2021/03/googles-floc-terrible-idea. Accessed on April 12, 2021.

[EHN18] S. Englehardt, J. Han, and A. Narayanan. I never signed up for this! privacy implications of email tracking. In *Privacy Enhancing Technologies*, 2018.

[EL] EasyList filter lists. https://easylist.to/.

[EN16a] S. Englehardt and A. Narayanan. Online tracking: A 1-million-site measurement and analysis. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security ACM CCS*, pages 1388–1401, 2016.

[EN16b] S. Englehardt and A. Narayanan. Online Tracking: A 1-million-site Measurement and Analysis. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, CCS '16, pages 1388–1401, New York, NY, USA, 2016. ACM. doi: 10.1145/2976749.2978313.

[EP] EasyPrivacy filter lists. https://easylist.to/easylist/easyprivacy.txt.

[ePD02] The European Parliament and the Council of the European Union. Directive 2002/58/EC of the European Parliament and of the Council of 12 July 2002 concerning the processing of personal data and the protection of privacy in the electronic communications sector (Directive on privacy and electronic communications), 2002. https://eur-lex.europa.eu/legal-content/EN/ALL/?uri=CELEX%3A32002L0058, accessed 29 March 2021.

[ePD09] Directive 2009/136/EC of the European Parliament and of the Council of 25 November 2009. https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=celex%3A32009L0136, accessed 29 March 2021.

[ePr21] Proposal for a regulation of the european parliament and of the council concerning the respect for private life and the protection of personal data in electronic communications and repealing directive 2002/58/ec (regulation on privacy and electronic communications) - mandate for negotiations with ep, 2021. https://data.consilium.europa.eu/doc/document/ST-6087-2021-INIT/en/pdf.

[ERE+15] S. Englehardt, D. Reisman, C. Eubank, P. Zimmerman, J. Mayer, A. Narayanan, and E. W. Felten. Cookies that give you away: The surveillance implications of web tracking. In *Proceedings of WWW 2015*, pages 289–299, 2015.

[Erl03] U. Erlingsson. *The Inlined Reference Monitor Approach to Security Policy Enforcement*. PhD thesis, Cornell University, 2003.

[Eur] European Parliament, the Council and the Commission. Charter of Fundamental Rights of the European Union, Official Journal of the European Communities, 18 December 2000 (2000/C 364/01). https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=OJ%3AC%3A2007%3A303%3ATOC.

[Fac] Faceboook website. https://www.facebook.com/.

[Fal10] Y. Falcone. You should better enforce than verify. In *Proc. of the 9th International Workshop on Runtime Verification (RV'10)*, volume 6418 of *LNCS*, pages 89–105. Springer-Verlag Heidelberg, 2010.

[fDR] N. E. C. for Digital Rights. Say "NO" to cookies – yet see your privacy crumble? https://noyb.eu/en/say-no-cookies-yet-see-your-privacy-crumble. Accessed on March 29, 2021.

[Fen74] J. S. Fenton. Memoryless subsystems. *Comput. J.*, 17(2):143–147, 1974.

[fir] The new Firefox. Fast for good. https://www.mozilla.org/en-US/firefox/new/.

[GAC16] G. G. Gulyás, G. Acs, and C. Castelluccia. Near-optimal fingerprinting with constraints. *Proceedings on Privacy Enhancing Technologies*, 2016(4):470–487, 2016.

[GAn] Google analytics: About demographics and interests analyze users by age, gender, and interest categories. https://support.google.com/analytics/answer/2799357?hl=en. Accessed on March 19, 2021.

[GBJS06] G. L. Guernic, A. Banerjee, T. Jensen, and D. A. Schmidt. Automata-based confidentiality monitoring. In *Asian Computing Science Conference (ASIAN'06)*, volume 4435, pages 75–89, 2006.

[GBLB18] A. Gómez-Boix, P. Laperdrix, and B. Baudry. Hiding in the Crowd: an Analysis of the Effectiveness of Browser Fingerprinting at Large Scale. In *WWW 2018: The 2018 Web Conference*, Lyon, France, Apr. 2018. doi: 10.1145/3178876.3186097.

[GDPR] Regulation (eu) 2016/679 of the european parliament and of the council of 27 april 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing directive 95/46/ec (general data protection regulation) (text with eea relevance). https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=celex:32016R0679.

[Gho] Ghostery Official website. https://www.ghostery.com/.

[Gma] Google's Gmail. https://gmail.com.

[Goo] Google Chrome browser. https://www.google.com/chrome/.

[GPC] Global privacy control (GPC). Unofficial Draft 27 January 2021, https://globalprivacycontrol.github.io/gpc-spec/. Accessed on April 1, 2021.

[GSa] Google. Evaluation of cohort algorithms for the FLoC API. https://github.com/google/ads-privacy/blob/master/proposals/FLoC/FLOC-Whitepaper-Google.pdf, accessed on April 2, 2021.

[GSp] Google spain sl and google inc. v agencia española de protección de datos (aepd) and mario costeja gonzález, case c-131/12. https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:62012CJ0131&from=EN.

[Gue07] G. L. Guernic. *Confidentiality Enforcement Using Dynamic Information Flow Analyses*. PhD thesis, Kansas State University, 2007.

[HBS15] D. Hedin, L. Bello, and A. Sabelfeld. Value-sensitive Hybrid Information Flow Control for a JavaScript-like Language. In *Proc. of the 28th Computer Security Foundations Symposium*. IEEE, 2015.

[HMS06] K. W. Hamlen, G. Morrisett, and F. B. Schneider. Computability classes for enforcement mechanisms. *ACM Trans. Prog. Lang. Syst.*, 28(1):175–205, 2006.

[HS06] S. Hunt and D. Sands. On flow-sensitive security types. In *ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages (POPL'06)*, volume 41:1, pages 79–90, New York, NY, USA, Jan. 2006. ACM.

[HS12] D. Hedin and A. Sabelfeld. Information-flow security for a core of JavaScript. In *Proc. of the 25th Computer Security Foundations Symposium*, pages 3–18. IEEE, 2012.

[HV05] O. Hallaraker and G. Vigna. Detecting malicious Javascript code in Mozilla. In *IEEE International Conference on Engineering of Complex Computer Systems (ICECCS'05)*, pages 85 – 94, 2005.

[HWB20] M. Hils, D. W. Woods, and R. Böhme. Measuring the Emergence of Consent Management on the Web. In *ACM Internet Measurement Conference (IMC'20)*, 2020.

[IAB] IAB Europe. IAB europe transparency & consent framework policies. https://iabeurope.eu/wp-content/uploads/2019/08/TransparencyConsentFramework_PoliciesVersion_TCFv2-0_2019-08-21.3_FINAL-1-1.pdf, accessed on 2020.01.21.

[IAB20a] IAB Europe. IAB Europe Transparency & Consent Framework Policies, 2020. https://iabeurope.eu/wp-content/uploads/2020/11/TCF_v2-0_Policy_version_2020-11-18-3.2a.docx-1.pdf.

[IAB20b] IAB Europe. Vendor List TCF v2.0, 2020. https://iabeurope.eu/vendor-list-tcf-v2-0/.

[IAK+17] M. Ikram, H. J. Asghar, M. A. Kaafar, A. Mahanti, and B. Krishnamurthy. Towards seamless tracking-free web: Improved detection of trackers via one-class learning. In *Privacy Enhancing Technologies*, 2017.

[II18] IAB Europe and IAB Tech Lab. Transparency and consent framework. https://github.com/InteractiveAdvertisingBureau/GDPR-Transparency-and-Consent-Framework, accessed on 2019.05.03, 04 2018.

[II19] IAB Tech Lab and IAB Europe. Transparency and consent string with global vendor & CMP list formats. https://github.com/InteractiveAdvertisingBureau/GDPR-Transparency-and-Consent-Framework/blob/master/TCFv2/IAB%20Tech%20Lab%20-%20Consent%20string%20and%20vendor%20list%20formats%20v2.md#the-core-string, 12 2019.

[II20a] IAB Europe and IAB Tech Lab. Global vendor list (GVL, v1.1, version 183). https://vendorlist.consensu.org/v-183/vendorlist.json, 01 2020.

[II20b] IAB Europe and IAB Tech Lab. Global vendor list (GVL, v2.0, version 20). https://vendorlist.consensu.org/v2/archives/vendor-list-v20.json, 01 2020.

[ISPL18] C. Iordanou, G. Smaragdakis, I. Poese, and N. Laoutaris. Tracing cross border web tracking. In *ACM Internet Measurement Conference (IMC)*, 2018.

[KR10] B. Köpf and A. Rybalchenko. Approximation and randomization for quantitative information-flow analysis. In *IEEE Computer Security Foundations Symposium (CSF'10)*, pages 3–14, 2010.

[Lap17] P. Laperdrix. *Browser Fingerprinting: Exploring Device Diversity to Augment Authentication and Build Client-Side Countermeasures*. PhD thesis, INSA Rennes, 2017. URL https://hal.inria.fr/tel-01621257.

[Las] Lastpass official website. https://www.lastpass.com/business.

[LBW09] J. Ligatti, L. Bauer, and D. Walker. Run-time enforcement of non-safety policies. *ACM Transactions on Information and System Security*, 12(3):1–41, 2009.

[LCA+17] T. Lauinger, A. Chaabane, S. Arshad, W. Robertson, C. Wilson, and E. Kirda. Thou shalt not depend on me: Analysing the use of outdated javascript libraries on the web. In *Network and Distributed System Security Symposium, NDSS*, 2017.

[LG08] G. Le Guernic. Precise Dynamic Verification of Confidentiality. In *International Verification Workshop (VERIFY'08)*, volume 372, pages 82–96, 2008.

[LGBJS06] G. Le Guernic, A. Banerjee, T. Jensen, and D. Schmidt. Automata-based Confidentiality Monitoring. In *Proc. of the Annual Asian Computing Science Conference*, pages 75–89. Springer LNCS vol. 4435, 2006.

[LGN18] T. Libert, L. Graves, and R. K. Nielsen. Changes in third-party content on european news websites after gdpr„ 2018. https://timlibert.me/pdf/Libert_et_al-2018-Changes_in_Third-Party_Content_on_EU_News_After_GDPR.pdf.

[Lib18] T. Libert. An automated approach to auditing disclosure of third-party data collection in website privacy policies. In *Proceedings of the 2018 World Wide Web Conference*, WWW '18, page 207–216. International World Wide Web Conferences Steering Committee, 2018. doi: 10.1145/3178876.3186087.

[Lin] Linkedin website. https://www.linkedin.com/.

[Lin16] R. Linus. Your social media fingerprint. https://robinlinus.github.io/socialmedia-leak/, 2016.

[LK15] R. Leenes and E. Kosta. Taming the cookie monster with Dutch law - a tale of regulatory failure. *Computer Law & Security Review*, 31, 2015.

[LN18] T. Libert and R. K. Nielsen. Third-party web content on eu news sites: Potential challenges and paths to privacy improvement, 2018. https://timlibert.me/pdf/Libert_Nielsen-2018-Third_Party_Content_EU_News_GDPR.pdf.

[LRB16a] P. Laperdrix, W. Rudametkin, and B. Baudry. Beauty and the Beast: Diverting modern web browsers to build unique browser fingerprints. In *37th IEEE Symposium on Security and Privacy (S&P 2016)*, 2016. URL https://hal.inria.fr/hal-01285470.

[LRB16b] P. Laperdrix, W. Rudametkin, and B. Baudry. Beauty and the Beast: Diverting modern web browsers to build unique browser fingerprints. In *37th IEEE Symposium on Security and Privacy (S&P 2016)*, San Jose, United States, May 2016. URL https://hal.inria.fr/hal-01285470.

[LSKR16] 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 *25th USENIX Security Symposium (USENIX Security 16)*. USENIX Association, 2016.

[LZW10] Z. Li, K. Zhang, and X. Wang. Mash-IF : Practical Information-Flow Control within Client-side Mashups. In *Proc. of the 40th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN 2010)*, pages 251–260. IEEE, 2010.

[Mat12] D. Mattioli. On orbitz, mac users steered to pricier hotels, 2012. URL http://online.wsj.com/article/SB10001424052702304458604577488822667325882.html.

[Mat19] C. Matte. Cookie glasses. https://github.com/Perdu/Cookie-Glasses, 2019.

[Mat20] C. Matte. Cookinspect. https://github.com/Perdu/Cookinspect, 2020.

[May09] J. R. Mayer. Any person... a pamphleteer": Internet Anonymity in the Age of Web 2.0. *Undergraduate Senior Thesis, Princeton University*, 2009.

[MB20] D. Machuletz and R. Böhme. Multiple purposes, multiple problems: A user study of consent dialogs after GDPR. *Proceedings on Privacy Enhancing Technologies*, 2020(2):481–498, 2020. doi: 10.2478/popets-2020-0037.

[ME08] S. McCamant and M. D. Ernst. Quantitative information flow as network flow capacity. In *PLDI 2008, Proc.of the ACM SIGPLAN 2008 Conf. on Programming Language Design and Implementation (PLDI 2008)*, pages 193–205, 2008.

[MHB+17] G. Merzdovnik, M. Huber, D. Buhov, N. Nikiforakis, S. Neuner, M. Schmiedecker, and E. Weippl. Block me if you can: A large-scale study of tracker-blocking tools. In *2nd IEEE European Symposium on Security and Privacy*, Paris, France, 2017. To appear.

[MKS15] C. Müller, M. Kovács, and H. Seidl. An analysis of Universal Information Flow based on Self-composition. In *Proc. of the 28th Computer Security Foundations Symposium*. IEEE, 2015.

[ML10] L. Meyerovich and B. Livshits. ConScript: Specifying and enforcing fine-grained security policies for Javascript in the browser. In *Proc. of the 2010 Symposium on Security and Privacy*. IEEE, 2010.

[MM12] J. R. Mayer and J. C. Mitchell. Third-party web tracking: Policy and technology. In *IEEE Symposium on Security and Privacy*, pages 413–427, 2012.

[MS12] K. Mowery and H. Shacham. Pixel Perfect: Fingerprinting Canvas in HTML5. In M. Fredrikson, editor, *Proceedings of W2SP 2012*. IEEE Computer Society, May 2012.

[MSR14] A. G. A. Matos, J. F. Santos, and T. Rezk. An Information Flow Monitor for a Core of DOM - Introducing References and Live Primitives. In *Trustworthy Global Computing - 9th International Symposium, TGC*, 2014.

[NIK+12] N. Nikiforakis, L. Invernizzi, A. Kapravelos, S. V. Acker, W. Joosen, C. Kruegel, F. Piessens, and G. Vigna. You are what you include: Large-scale evaluation of remote javascript inclusions. In *ACM Conference on Communications and Computer Security (ACM CCS'12)*, pages 736–747, 2012.

[NJL15] N. Nikiforakis, W. Joosen, and B. Livshits. Privaricator: Deceiving fingerprinters with little white lies. In *Proceedings of the 24th International Conference on World Wide Web, WWW 2015, Florence, Italy, May 18-22, 2015*, pages 820–830, 2015.

[NKJ+13a] N. Nikiforakis, A. Kapravelos, W. Joosen, C. Kruegel, F. Piessens, and G. Vigna. Cookieless monster: Exploring the ecosystem of web-based device fingerprinting. In *IEEE Symposium on Security and Privacy, SP 2013*, pages 541–555, 2013. doi: 10.1109/SP.2013.43.

[NKJ+13b] N. Nikiforakis, A. Kapravelos, W. Joosen, C. Kruegel, F. Piessens, and G. Vigna. Cookieless Monster: Exploring the Ecosystem of Web-Based Device Fingerprinting. In *Proceedings of the 2013*

IEEE Symposium on Security and Privacy, SP '13, pages 541–555, Washington, DC, USA, 2013. IEEE Computer Society. doi: 10.1109/SP.2013.43.

[NLV+20] M. Nouwens, I. Liccardi, M. Veale, D. Karger, and L. Kagal. Dark Patterns after the GDPR: Scraping Consent Pop-ups and Demonstrating their Influence. In *CHI*, 2020.

[OCJ12] Ł. Olejnik, C. Castelluccia, and A. Janc. Why johnny can't browse in peace: On the uniqueness of web browsing history patterns. In *Hot Topics in Privacy Enhancing Technologies (HotPETs 2012)*, 07 2012.

[Odl03] A. Odlyzko. Privacy, economics, and price discrimination on the internet. In *Proceedings of the 5th International Conference on Electronic Commerce*, ICEC '03, pages 355–366, New York, NY, USA, 2003. ACM. doi: 10.1145/948005.948051.

[OTC14] L. Olejnik, M. Tran, and C. Castelluccia. Selling off user privacy at auction. In *21st Annual Network and Distributed System Security Symposium, NDSS 2014, San Diego, California, USA, February 23-26, 2014*, 2014.

[P3P] The platform for privacy preferences 1.0 (P3P 1.0) specification. https://www.w3.org/TR/P3P/.

[PB] Privacy Badger Official website - Electronic Frontier Foundation. https://www.eff.org/privacybadger.

[PC20] H. Pawlata and G. Caki. The Impact of the Transparency Consent Framework on current Programmatic Advertising Practices. 2020. 4th International Conference on Computer-Human Interaction Research and Applications.

[PKM19] P. Papadopoulos, N. Kourtellis, and E. P. Markatos. Cookie synchronization: Everything you always wanted to know but were afraid to ask. In *The World Wide Web Conference, WWW 2019, San Francisco, CA, USA, May 13-17, 2019*, pages 1432–1442, 2019.

[RBC+20] S. Roth, T. Barron, S. Calzavara, N. Nikiforakis, and B. Stock. Complex security policy? a longitudinal analysis of deployed content security policies. In *Network and Distributed Systems Security Symposium (NDSS)*, 2020.

[rep18] OpenWPM - A web privacy measurement framework, 2018. https://github.com/citp/OpenWPM.

[RKW12] F. Roesner, T. Kohno, and D. Wetherall. Detecting and defending against third-party tracking on the web. In *Proceedings of the 9th USENIX Symposium on Networked Systems Design and Implementation, NSDI 2012*, pages 155–168, 2012.

[RNV+18] A. Razaghpanah, R. Nithyanand, N. Vallina-Rodriguez, S. Sundaresan, M. Allman, C. Kreibich, and P. Gill. Apps, trackers, privacy, and regulators: A global study of the mobile tracking ecosystem. In *Network and Distributed System Security Symposium, NDSS*, 2018.

[RS10] A. Russo and A. Sabelfeld. Dynamic vs. static flow-sensitive security analysis. In *IEEE Computer Security Foundations Symposium (CSF'10)*, pages 186–199. IEEE, 2010.

[Sam] Same Origin Policy. https://www.w3.org/Security/wiki/Same_Origin_Policy.

[Sam11] A. Sampson. Ryanair exhibit a., 2011. URL https://twitter.com/sampsonian/status/50199798099357696.

[SCM+10] A. Soltani, S. Canty, Q. Mayo, L. Thomas, and C. J. Hoofnagle. Flash cookies and privacy. In *AAAI Spring Symposium: Intelligent Information Privacy Management*, 2010.

[SM03] A. Sabelfeld and A. C. Myers. Language-based information-flow security. *IEEE Journal on Selected Areas in Communication*, 21(1):5–19, 2003.

[Smi09] G. Smith. On the foundations of quantitative information flow. In *Foundations of Software Science and Computational Structures*, volume 5504 of *Lecture Notes in Computer Science*, pages 288–302. Springer Berlin / Heidelberg, 2009.

[SMWL10] K. Singh, A. Moshchuk, H. J. Wang, and W. Lee. On the incoherencies in web browser access control policies. In *Proc. of the 2010 Symposium on Security and Privacy*, pages 463–478. IEEE, 2010.

[SN17] O. Starov and N. Nikiforakis. Xhound: Quantifying the fingerprintability of browser extensions. In *Proceedings of the 38th IEEE Symposium on Security and Privacy*, pages 941–956, 2017.

[SNGS20] T. H. Soe, O. E. Nordberg, F. Guribye, and M. Slavkovik. Circumvention by design—dark patterns in cookie consents for online news outlets. 2020, 2006.13985. URL http://arxiv.org/abs/2006.13985.

[SR14] J. F. Santos and T. Rezk. An Information Flow Monitor-Inlining Compiler for Securing a Core of Javascript. In *ICT Systems Security and Privacy Protection - 29th IFIP TC 11 International Conference, SEC 2014*, 2014.

[SRB] D. F. Some, T. Rezk, and N. Bielova. Web stats. http://webstats.inria.fr/.

[SRDK+19] I. Sanchez-Rola, M. Dell'Amico, P. Kotzias, D. Balzarotti, L. Bilge, P.-A. Vervier, and I. Santos. Can I opt out yet?: GDPR and the global illusion of cookie control. In *Asia Conference on Computer and Communications Security (AsiaCCS'19)*, 2019.

[SSB17] I. Sánchez-Rola, I. Santos, and D. Balzarotti. Extension breakdown: Security analysis of browsers extension resources control policies. In *26th USENIX Security Symposium*, pages 679–694, 2017.

[SSM10] S. Stamm, B. Sterne, and G. Markham. Reining in the web with content security policy. In *Proceedings of the 19th International Conference on World Wide Web, WWW 2010, Raleigh, North Carolina, USA, April 26-30, 2010*, pages 921–930, 2010. doi: 10.1145/1772690.1772784.

[SVAS17] A. Sjösten, S. Van Acker, and A. Sabelfeld. Discovering browser extensions via web accessible resources. In *Proceedings of the Seventh ACM on Conference on Data and Application Security and Privacy*, CODASPY '17, pages 329–336. ACM, 2017.

[TArc] Trustarc privacycentral: Your new privacy management experience. https://trustarc.com/. Accessed on April 10, 2021.

[TTBM19] M. Trevisan, S. Traverso, E. Bassi, and M. Mellia. 4 years of EU cookie law: Results and lessons learned. *Proceedings on Privacy Enhancing Technologies Symposium (PETS'19)*, 2019.

[TTG+17] S. Traverso, M. Trevisan, L. Giannantoni, M. Mellia, and H. Metwalley. Benchmark and comparison of tracker-blockers: Should you trust them? In *Network Traffic Measurement and Analysis Conference (TMA'17)*, 2017.

[ubl] uBlock Origin - An efficient blocker for Chromium and Firefox. Fast and lean. https://github.com/gorhill/uBlock.

[UTD+18] T. Urban, D. Tatang, M. Degeling, T. Holz, and N. Pohlmann. The Unwanted Sharing Economy: An Analysis of Cookie Syncing and User Transparency under GDPR. *CoRR*, abs/1811.08660, 2018.

[vAWN19] R. van Eijk, H. Asghari, P. Winter, and A. Narayanan. The impact of user location on cookie notices (inside and outside of the European union). In *Workshop on Technology and Consumer Protection (ConPro'19)*, 2019.

[VFG+19] P. Vallina, Á. Feal, J. Gamba, N. Vallina-Rodriguez, and A. F. Anta. Tales from the porn: A comprehensive privacy analysis of the web porn ecosystem. In *Proceedings of the Internet Measurement Conference (IMC'19)*, 2019.

[VLRR18] A. Vastel, P. Laperdrix, W. Rudametkin, and R. Rouvoy. FP-STALKER: Tracking Browser Fingerprint Evolutions. In *39th IEEE Symposium on Security and Privacy (S&P 2018)*, San Fransisco, United States, May 2018.

[VNBJ14] T. Vissers, N. Nikiforakis, N. Bielova, and W. Joosen. Online airline price discrimination dataset, 2014. URL http://people.cs.kuleuven.be/~thomas.vissers/data/price_discrimination.zip.

[VNJ+07] P. Vogt, F. Nentwich, N. Jovanovic, E. Kirda, C. Kruegel, and G. Vigna. Cross-Site Scripting Prevention with Dynamic Data Tainting and Static Analysis. In *Proc. of the Symposium on Network and Distributed System Security (NDSS'07)*, 2007.

[VSI96] D. Volpano, G. Smith, and C. Irvine. A sound type system for secure flow analysis. 4(2-3):167–187, 1996.

[W3C12] W3C. A vocabulary and associated apis for html and xhtml, 2012. Online: http://dev.w3.org/html5/spec/single-page.html#history.

[WBV15] M. West, A. Barth, and D. Veditz. Content Security Policy Level 2. W3C Candidate Recommendation, 2015.

[WP136] WP29 Opinion 04/2007 on on the concept of personal data (WP 136, 2007. https://ec.europa.eu/justice/article-29/documentation/opinion-recommendation/files/2007/wp136_en.pdf.

[WP212] WP29 Opinion 04/2012 on the Cookie Consent Exemption - ARTICLE 29 DATA PROTECTION WORKING PARTY, 2012. https://ec.europa.eu/justice/article-29/documentation/opinion-recommendation/files/2012/wp194_en.pdf.

[YCIS07] D. Yu, A. Chander, N. Islam, and I. Serikov. Javascript instrumentation for browser security. In *ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages (POPL'07)*, pages 237–249, 2007.

[YMMP16] Z. Yu, S. Macbeth, K. Modi, and J. M. Pujol. Tracking the trackers. In *International Conference on World Wide Web, WWW*, pages 121–132, 2016.

[Zda02] S. A. Zdancewic. *Programming languages for information security.* PhD thesis, Cornell University, 2002.

[ZJR13] D. Zanarini, M. Jaskelioff, and A. Russo. Precise enforcement of confidentiality for reactive systems. In *IEEE 26th Computer Security Foundations Symposium*, pages 18–32, 2013.