Measure Cookie Setting Behavior of Web Pages Showing Cookie Privacy Warnings

Yiwen Cui

Master of Science
Cyber Security, Privacy and Trust
School of Informatics
University of Edinburgh
2021

Abstract

There are different kinds of tracking technologies available to track users' online behaviour. Web cookie is one of tracking technologies applied by most websites. To limit uses of online tracking technologies and better protect users' data, General Data Protection Regulation (GDPR) came into effect in 2018 [17]. Cookie setting behavior is also limited by GDPR. One main limitation about web cookies is called cookie consent dialogs. GDPR also has different requirements about receiving users' cookie consents. This project mainly aims to check cookie compliance of websites according to GDPR and consistency of showing cookie dialogs. In this project, an automatic tool—Cookie Dialog Positioning Assistant (CDPA) was developed to locate cookie dialogs, interactive elements and interact with different kinds of elements to collect cookie related data. CDPA was developed mainly based on the Selenium framework to simulate human operations of browsing websites and interacting with buttons. CDPA was applied to test 1,000 websites to collect needed data. 958 websites were judged as valid websites because they can be connected normally. Final accuracy of locating cookie dialogs with CDPA is 90.8%. In these valid websites, 420 websites were found that they display cookie dialogs. Further data collection about clickable elements was implemented on these websites. First-party cookie numbers and third-party cookie numbers were also collected after interactions with different elements. After whole process of data collection, a final analysis was implemented to judge whether most websites set compliant cookie dialogs and cookies according to the related requirements listed in the GDPR. Findings about cookie setting compliance of websites were also extended to some recommendations for GDPR's improvements in the future.

Acknowledgements

I would like to express my gratitude to my supervisor Kami Vaniea. Thanks to her for the rich insights and support during this difficult period. Her advice can always help me to improve the project and make the project better. I would also like to thank my parents for their timely encouragements even we are in different countries. Finally, thanks everyone that encouraged me to keep going during this year.

Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

(*Yiwen Cui*)

Table of Contents

Chapter 1

Introduction

1.1 Motivation

Tracking technology has been applied on the most of websites to record visitors' browsing history to provide better service for the users. One of popular technologies is called cookie, which is stored on the client's side in the form of piece of textual data [43]. Cookies are commonly set by websites to record users' relevant visiting information. Cookies can provide retained arbitrary like user identifier or any material that the server needs to continue the work [26]. Companies and marketers always take advantage of cookies to analyze customers' interest to achieve their preferences and deliver suitable advertisements to targeted groups [13] [38]. Most users do not really know how their information is used on the Internet and how their privacy is utilized by the host. Moreover, users who focus on their privacy also find that they are limited to have options to control how their private data to be used on the Internet [35]. Meanwhile, one research also concluded that tracking technology, like cookies can provide an efficient way for third-party companies to achieve users' private browsing information [25]. As a result, privacy problems caused by cookies have been paid much more attention by the public now because frequency of revealing users' privacy on the Internet has become higher.

To help ease these kinds of private data related problems, certain regulations were made in the General Data Protection Regulation (GDPR). GDPR was created to limit the behavior of processing users' personal private data by setting strict standards in the European areas [17]. One special regulation is about setting cookie dialog. Cookie dialogs are required by GDPR for those websites which take advantage of unnecessary cookies. Websites need to set visible cookie dialogs or cookie banners on the websites to inform visitors that they are using cookies and how their information will be used with the cookie technology. Location of placing cookie dialog is not strictly specified, the dialogs can be placed at the bottom of the website or in the center of page or anywhere that user can easily notice. Several essential choices should also be set on the cookie dialogs for users to present their attitudes.

Number of cookie dialogs are indeed increased after GDPR [17], while one fact is confirmed that not all the websites have set the cookie dialogs to inform users that they are using cookies [46]. Although some behaviors of using cookies and sharing tracking data have been limited efficiently after GDPR [15] [43], some research also points out that some websites will set cookies no matter users choose which option [44] [29]. Moreover, habitual clicking the choice of 'accept' has become common among users because most of them think their choices are meaningless [44]. These phenomena can lead to several questions:

- Whether all the websites' cookie settings and cookie dialogs settings can satisfy the requirements listed in the GDPR?
- Whether GDPR really can have effects on the websites' behaviors of setting cookies even websites set cookie dialogs?
- Whether websites can display cookie dialogs again when the pages are reloaded after users have given decisions on the first visit time?

The main problem is that a completed dataset with related data is lacked to make that kind of judgement and this project aimed to solve this main problem and do research on these questions. According to the three questions put forward above, several hypothesises were made at the beginning of the project:

- Most websites cannot display compliant cookie dialog and set cookies properly.
- GDPR does not really have effects on the websites' cookie setting behavior and it should further improve cookie related regulations.
- Websites will show cookie dialogs again after users choose to decline cookie use at the first visit time, otherwise, websites will not show dialogs again.

This project can further benefit users and lawyer. Lawyers will be given some recommendations depending on the collected data to modify related regulations in the GDPR to better protect users' data. Users will also be given advice on how should they respond to the cookie dialogs properly depending on the analysis of collected data.

1.2 Main Contribution

Although there has been much relevant research around the cookie settings and cookie dialogs, there is still no one research to completely judge cookie compliance and cookie dialog compliance according to the GDPR. This project realized it by judging cookies and cookie dialogs from a different direction. It connected with GDPR closely to complete judgement of cookie compliance and cookie dialog compliance of 1,000 popular websites. An automatic crawler called Cookie Dialog Positioning Assistant (CDPA) was developed with 90.8% accuracy of locating dialogs. The tool was evaluated from aspects of usability and accuracy. A completed dataset was also collected with help of CDPA to give a completed conclusion about GDPR's effects on the cookie settings and cookie dialog settings. Finally, 1,000 websites were randomly chosen from 'Tranco' list [7], CDPA judged 958 valid websites that can be connected successfully from them. In these 958 websites, there are 420 websites were judged as displaying cookie dialogs and 292 websites were judged as setting compliant cookie type although no cookie dialogs are displayed. In these 420 websites containing cookie dialogs, about 66.2% websites do not correctly display enough interactive elements. All of these websites do not provide a way for users to decline the coookie use. As for the 142 websites that provide 'decline' elements, there are about 35.4% websites' provided 'reject' options are meaningless as cookie number is increased after interaction with 'decline'. As for the consistency, only 10 websites (7.1% in those websites with 'decline' options) can both show cookie dialogs after users reloading the page, no matter users' first visit time choices. Depending on these statistics and results, several useful recommendations were also made for users and GDPR. To summary, the main contributions are listed as following:

- Successfully developed an automatic crawling tool – CDPA to locate cookie dialogs and clickable elements displayed by the websites and to interact with each kind of clickable option on the cookie dialogs to collect cookie numbers.
- Generate a cookie relevant information dataset constructed by cookies relevant information collected from targeted 1,000 popular websites with help of CDPA.
- Measure cookie setting behaviors and cookie dialogs settings of different websites by analyzing collected data to further check websites' cookie compliance.
- Completely judged websites' consistency of displaying cookie dialogs on the

reloaded pages between showing consent and declining cookies on the users' first visit.

1.3 Thesis Structure

The thesis contains five main parts. The following content will be an outline of the thesis. This chapter is the first part of the thesis, the following content will be mainly divided into four chapters. Chapter 2 will illustrate relevant background about the project, background will cover introduction of tracking technology, cookies, GDPR and previous related work. Chapter 3 will discuss the design of the project and detailed implementation process of the project. Chapter 4 will show two experiments tested by CDPA and related data will be collected for analysis use.Result analysis around each research question will also be given with help of the data collected from the second experiment. The analysis will cover the research topics about cookie dialog setting compliance, cookie setting compliance and consistency of showing cookie dialogs. Then, evaluation about the tool's efficiency and accuracy will also be expounded in the chapter 4. A detailed conclusion about the whole project and expected future work will be given in the chapter 5 of the thesis. Recommendations for GDPR about further improvements and advice for users about how to face cookie dialogs shown on the websites properly are also put forward in the chapter 5.

Chapter 2

Background

This chapter discusses the main background of the project and vital concept related with the project. It includes background of web tracking, specific introduction to the web cookies, GDPR and related work. ‘Web tracking’ part introduces main modern web tracking technologies and how they are used on the Internet briefly. Then, section of ‘Web Cookies’ discusses how web cookies work on the Internet in detail and type of web cookies are classified and introduced. The part of ‘GDPR’ introduces GDPR and web cookies related regulations. As for the last section of this part, it discusses related research studies and technical works that in the similar field specifically.

2.1 Web Tracking

To find and retain more targeted customers, most websites try to collect users’ personal data when users are browsing the websites. Essentially, companies take advantage of collected data for targeted advertising for the targeted customers. However, with development of web tracking technologies, customers’ online private data is taken and collected for other reasons. Some examples are listed by [12], collected data is used to personalize research results [21], background research or some other government related usages. Massive spread of personal online data is caused by the first-party company and third-party company. With regard of first-party’s behavior of leaking information to the third-party, one research has shown that 75% websites in the list of 120 Alexa popular websites have leaked users’ personal data to other third parties [24]. Third-party trackers are also set in different websites to monitor customers, one research has shown that about 46% popular websites’ home page listed by Alexa are tracked by one or even more third parties [28]. No matter which kind of information leakage, web tracking technology is the essential role leading different parties to track and collect users' data successfully. Web tracking technology can be divided into stateful tracking and stateless tracking according to the position where the recognition data is stored [30]. Stateful tracking requires to store information on the client side, but trackers can recognize and identify users without storing information on the users' side with stateless tracking technology. Web cookies, web storage and ETags are classified as stateful tracking techniques while stateless tracking techniques include browser and device fingerprinting [33]. Data saved by web storage can be checked by the browser and plugins [33]. Stateless tracking techniques and stateful tracking techniques can store different kinds of information when they are applied. Stateful tracking techniques, such as cookies can store users' visiting history to one certain website. Stateless tracking, like device fingerprinting can store information of installed plugins on the browser, hardware ID and Mac address. As mentioned above, these techniques indeed bring many benefits to the society, the stored information can help promote governments' identification work like identifying theft and network crime and background research [12]. Meanwhile, a lot of privacy related problems are also caused by these tracking techniques. Diffusion of privacy information has caused much relevant research and work which will be introduced in the following section. [37] states that appearance of web cookies is a turning point in the computing field. Thereby, technology of web cookies is a vital tracking technology in the computing area and it will be introduced in the next part.

2.2 Web Cookie

2.2.1 Role of Web Cookies

Firstly, it should mention that Hypertext Transfer Protocol (HTTP) works with stateless requests. It means that each time user clicks a link, the browser will send a request, and the server will send back a response message, however, when the browser receives the response, the connection will be cut and it causes that each time the server will treat the browser as a new one [27]. With appearance of cookies, users' state can be successfully retained. Cookies appeared in the 1994 to record the interaction history between clients and server. Cookies are always referred as a text file containing piece of data related with users' personal data. The personal data usually include users' username, passwords and other preference settings, the website's settings will be retained

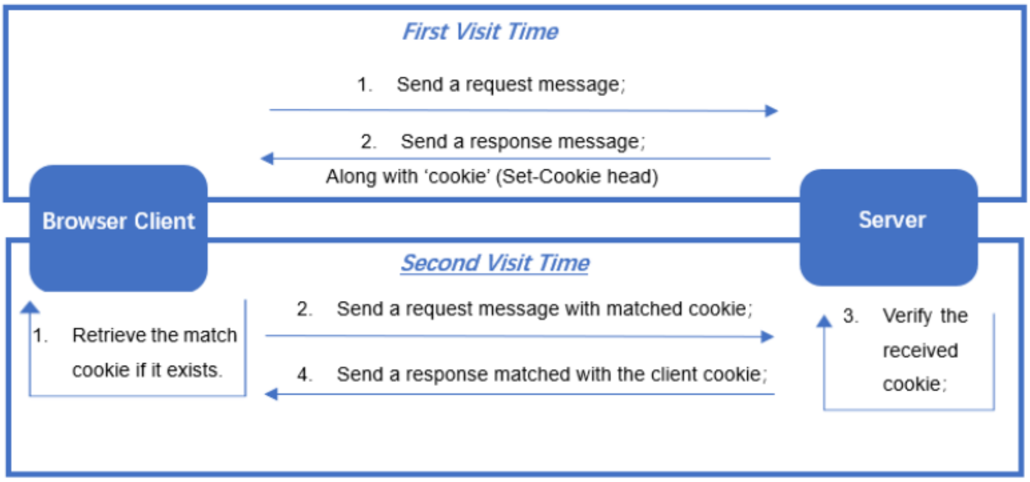

First Visit Time
1. Send a request message;
2. Send a response message;
Along with 'cookie' (Set-Cookie head)
Browser Client
Server
Second Visit Time
1. Retrieve the match cookie if it exists.
2. Send a request message with matched cookie;
4. Send a response matched with the client cookie;
3. Verify the received cookie;

Figure 2.1: Example of how cookies work

as users' choices at the first visit time [45]. With help of cookies, websites are able to retrieve the stored text file so that users do not need to log in again or add their preferred commodities into the cart again when they revisit the same website. Normally, a simple web cookie is consisted of 'Name = Value', host name, attribute of 'HttpOnly' and expiration time [13]. When the server sends back the response messages, it will also pass the special 'cookie' to the client [27]. The cookie contains clients' information so that the server can recognize the user when the cookie is sent back by the client at the next visiting time. The completed cookie relevant process can be simply expressed as 'generate cookie', 'retrieve cookie' and 'verify cookie'. The specific process is shown in the figure 2.1. When the client sends the request message for the first visiting time, server will generate a cookie for the client and send it along with the response message via HTTP 'Set-Cookie head' [13]. Third-party always additionally set up their own cookies when a user visits another website. Third-party will give a cookie with specified ID and store it in the client storage, when the user visits other new websites, that third-party can recognize the user by retrieving the ID [10]. Cookies can easily reveal users' personal information, so it is really important to limit the usage of cookies.

2.2.2 Type of Web Cookies

Web cookies are mainly classified into different types according to different methods. Classification methods include 'duration', 'provenance' and 'purpose' [3]. Cookies are divided into persistent cookies and session cookies depending on the criterion of 'duration'. Persistent cookies will not be removed from the hard drive until it meets the expiration time, or the device owner deletes it manually. Session cookies can be deleted automatically when the browser is closed. According to the classification method called ‘provenance’, there are two groups which called ‘first-party cookies’ and ‘third-party cookies’. This can be judged by exploring who set the cookies, the first-party cookies are set by the websites the users are visiting while the third-party cookies are set by other websites, even users have never visited these third-party websites. With regard of the last classification method of ‘purpose’, cookies are divided into ‘strictly necessary cookies’, ‘preference cookies’, ‘statistics cookies’ and ‘marketing cookies’. These cookies are divided into these specific names. Strictly necessary cookies are those cookies compulsory needed for a website. Preference cookies are the cookies visitors want to set on the website. Statistics cookies are the cookie recording the situation of users browsing the page. Marketing cookies are always seen as persistent cookies to help deliver more targeted advertisements to the users.

2.3 General Data Protection Regulation (GDPR)

2.3.1 Introduction to GDPR

General Data Protection Regulation (GDPR) was published for the purpose of managing the use of online personal data and giving specific standards about processing data. It was enforced on the 5th, May in 2018 in the European Union (EU) [17]. It contains some updates compared with the data privacy regulations that went into effect in 1995 [42]. This regulation can govern the companies that have rights to control EU residents’ personal data over the world [19]. There are 6 principles in the GDPR, which are ‘lawfulness, fairness and transparency’, ‘purpose limitation’, ‘data minimization’, ‘accuracy’, ‘storage limitation’ and ‘integrity and confidentiality’ [41]. To meet these principles, GDPR requires the data processing to be transparent enough and different requirements were put forward. One new point that the GDPR puts forward is about consent. GDPR requires that the users should always be informed of personal data processing and consent should be approved by the users. Explicit consents from users should be verified for further private data processing [19]. Different penalties will also be made if companies violate the GDPR [9]. These changes and requirements were set to ensure that EU residents’ personal data and basic rights can be better protected. Behavior of sharing data between third parties is being limited by the GDPR [43], so the enforcement of GDPR indeed has large effects on the data processing in the EU.

Meanwhile, behavior of setting cookies was also largely affected by the GDPR.

2.3.2 Cookie Related Regulations and Cookie Dialogs

According to the GDPR, users' consents should be considered before processing personal data. With regard of cookies, the GDPR also has relevant requirements which are listed as following [3]:

- Websites should ask for users' consents before setting unnecessary cookies.
- Introduction of how tracked data will be used should be delivered explicitly and specifically before achieving users' consents.
- Users have rights to decline or not give the consents to set cookies and they should still have the rights to access the service normally.
- Users have rights to withdraw their consents made at the first visit time easily.

To meet these cookie related requirements in the GDPR, cookie dialog (also called cookie banner) appeared. Cookie dialog is a dialog to ask for users consents of setting cookies. A compliant cookie dialog should contain enough information and sufficient options for the users to choose. The cookie dialog should firstly explain the purpose of tracking users with cookies in the form of words, and then it should provide options for users to agree or decline. As for the last requirement listed above, the other choice of 'withdraw' should also be provided for users or relevant instructions about withdrawing consents should be specified on the cookie dialogs. These requirements will be the criteria in this project to judge cookie compliance of websites.

2.4 Related Work

Web cookies have received much attention from the society as it is closely related with privacy. Cookies are defined as not only privacy risk [39], yet cookies are still mentioned in much related research like [10] [18] [33]. As mentioned above, cookie dialog can efficiently help users to know about cookies and protect online personal data. However, according to [36], habituation is easily generated if the same action is repeated by one person constantly. One experiment implemented by [11] also pointed that level of habituation is related with dialogs' exposure times. At the present, many websites have set the cookie dialogs at the homepage, frequent dialog exposure time can have large impacts on human's habituation of clicking dialogs. Many users think that it is meaningless to give their choices, so they keep clicking one fixed choice when they see the dialogs. [44] implemented three experiment on the dialog cookies which express that cookie dialogs' position, used language and number of provided choices will affect users a lot. As a result, whether the cookie setting and cookie dialog setting in each website are compliant or not is not paid much attention by the users. Some websites indeed set up cookie dialogs to show that they are asking users' consent, but some of their cookie dialogs and cookies are not set compliantly.

To help users to better manage the cookies, some useful tools or plugins were created. 'TTPCookie' is a tool for third-party management that realized a fine-grained, pre-site protocol proposed by [23]. Meanwhile, different methods to remove and manage cookies are put forward for users to choose. Many projects were implemented to measure and explore cookies setting behavior. Related technical works are discussed in the following words. [35] tried to find out how people can control their privacy under help of GDPR. This project loads the pages and manually identified and classified the cookie dialogs on the targeted website, then each dialog is manually interacted to check cookie settings. One important contribution this project made is that it found that opting out is not implemented properly by most of websites [35]. One related research about classifying cookies was also implemented. The main contribution is a machine learning framework called 'CookieMonster' that can classify cookies in-the-wild into one of four main categories for further research [22]. This project concluded that vast of cookies are not so useful that they can be removed and there will be no impacts on the essential functions, and one extra finding that the project has is that most of users cannot efficiently take advantage of choices made by the GDPR to manage the cookies. One tool called 'Cookie Dialog Analyser' (CDA) was developed to locate the cookie dialogs and clickable buttons [31]. This kind of tool can be useful for measurement of cookie settings in the related project. Two research utilized similar crawler tools to judge website settings' compliance. [29] implemented two tools called 'Cookinspect' and 'Cookie Glasses' to judge cookie banner compliance. 'Cookinspect' is a selenium-based crawler to automatically visit websites to store consents and 'Cookie Glasses' is an extension to check if users' choices can be correctly transmitted. It finally judged that around 54% in 1426 European websites contain at least one of legal violations. [14] developed an automatic tool 'CoolCheck' to check existence of cookie dialogs in 500 most popular Italian websites. [40] developed a tool to locate 1,000 websites' cookie dialogs and she found that about 54% websites contain cookie dialogs, she also judged that cookie number is always increased by evaluating first-party cookie, third-party cookie and ID-like cookie. All related works focus on the exploration about cookies classification and cookie sharing, there are two projects focusing on the cookie compliance. However, these two projects do not completely take all the regulations into account and compared with [35] completed by manually testing, this project aims to implement testing automatically. Compared with [31] and [40], although this project also focused on the cookie dialogs and interactive elements collection, it has an additional and further focus on compliance of cookie settings and cookie dialog settings according to the GDPR.In addition, it also judges the consistency of cookie dialogs appearance after different kinds of interactions.

Chapter 3

Design and Implementation

This chapter discusses the design and implementation of the tool–Cookie dialog Positioning Assistant (CDPA). Design of the tool covers the requirements of the tool and how the tool is expected to work. Implementation part illustrates how CDPA was implemented in detail by presenting relevant codes and figures.

3.1 Design of CDPA

3.1.1 Requirements of CDPA

In this project, there are several functional requirements and non-functional requirements. Functional requirements should cover main functions CDPA needs to implement. Non-function requirements should cover three fields which are reliability, recoverability, and efficiency.

About the functional functions, four basic functions the CDPA needs to realize. These functions can be completed based on the simulation of browsing the websites successfully. CDPA is firstly required to judge whether a website displays a cookie dialog at the homepage. If a website displays a cookie dialog, CDPA needs to locate the cookie dialog and works as a crawler to collect the cookie dialogs accurately. Secondly, CDPA is expected to locate the interactive elements on the cookie dialogs. Clickable elements for users to accept, decline, set cookie preference and check privacy statement should be saved when CDPA locates them. Thirdly, CDPA can simulate clicking buttons to interact with cookie dialogs and store the cookie numbers before and after the simulations. Finally, CDPA is required to reload pages to locate cookie dialogs again to compare the differences and consistency. Additionally, CDPA should provide a convenient way for users to check and verify the scraped results directly.

As for the non-functional requirements, three main requirements will be discussed for the tool. These non-functional requirements are 'reliability', 'recoverability' and 'efficiency'. When it refers to the requirement of 'reliability', the accuracy and speed of responding time should be introduced. The CDPA is mainly designed for the data collection, its accuracy is more vital. The basic goal of accuracy of locating the cookie dialogs by the CDPA is above 80% and accuracy of locating elements is around 50%. Speed is not attached with much attention because accuracy is more important in this project. About the requirement of 'recoverability', the tool should identify as many as possible bugs and errors so that the process of working will not be stopped with unexpected errors. If the tool is applied to test a large number of websites, but working process is constantly stopped with errors, it will consume and waste too much time in the final experiment. As a result, the tool's ability of recovering from errors should be focused a lot. With regard of the last requirement 'efficiency', CDPA is expected to be an efficient crawler tool specifically in the field of scraping cookie dialogs and simulate human operations interacting with the cookie dialogs, so the expected outcome targeting on the 'efficiency' requirement should be that 'CDPA can be efficient enough as a crawler tool to collect needed cookie related information'.

3.1.2 Selenium Framework and Working Flow of CDPA

In the design stage, the working flow and expected working process of CDPA was decided according to the requirements. All the stages were implemented based on the operation of simulating browsing the websites. There are many methods provided to simulate browsing behavior. Selenium is a feasible framework for implementing automation testing. Selenium is an integrated environment for automation testing and selenium WebDriver is one of the automation tools contained in the 'Selenium' [20]. Selenium WebDriver is used in most of tests that need to communicate directly with browser. As a result, Selenium WebDriver is decided as the main framework to complete the development of CDPA.

As shown in the flow chart 3.1, the working flow of CDPA is mainly divided into four stages. The first stage is to locate cookie dialogs automatically on the test websites. In this stage, CDPA needs to judge if the website displays a cookie dialog, if there is a cookie dialog, the CDPA needs to save it in a certain form for future uses. The first stage also contains manually checking and verification. According to the past

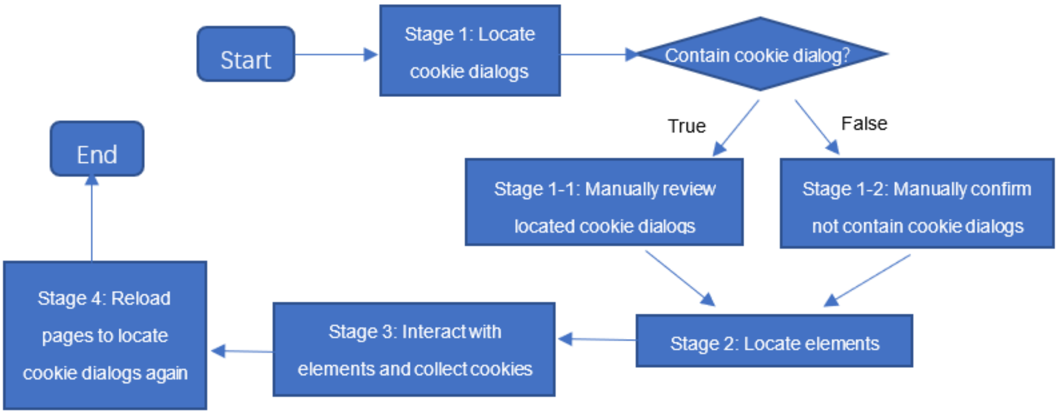

Start
Stage 1: Locate cookie dialogs
Contain cookie dialog?
True
False
Stage 1-1: Manually review located cookie dialogs
Stage 1-2: Manually confirm not contain cookie dialogs
Stage 2: Locate elements
Stage 3: Interact with elements and collect cookies
Stage 4: Reload pages to locate cookie dialogs again
End

Figure 3.1: Working flow of CDPA

research, fully automatic tool cannot locate the cookie dialogs successfully with accuracy of 100% [14] [31]. As result, to increase accuracy of the tool, manually checking will be implemented. Stage of manually checking will be divided into two parts. The first part is set to review and check saved cookie dialogs. The cookie dialogs located by the tool in the first stage will be displayed in the form of figures to the users and users will confirm if the located dialogs are correct or not. The second part in this stage is to confirm that the websites indeed do not contain cookie dialogs. In the first stage, the tool fistly judges that some websites do not contain cookie dialogs, but some judgements may be incorrect, so it is necessary for users to judge manually if these websites contain cookie dialogs or not. The second stage is about locating the elements on the cookie dialogs. The main type of elements the tool aims to look for are divided as ‘accept’, ‘decline’, ‘customize cookies’ and ‘privacy statement’. These elements are the important factors to judge compliance of cookie dialog settings of websites. Meanwhile, these elements will also be saved for future use to collect more useful information. One extra operation of ‘manually check elements’ will also be added in this stage as some elements cannot be correctly identified and classified. The third stage of the design is set as interacting with collected elements and collecting cookie number on the websites. It will judge cookie settings before clicking any elements and after clicking ‘accept’ and ‘decline’. Additionally, one vital step to check if the website provides the way for users to withdraw their choices is also set in this stage. The last stage is to locate cookie dialogs again, it will check whether websites show dialogs again after the first visit and if the result is affected by users’ first visiting time decision. The method to store the collected data is local databases. As the development language is

Python and development tool is ‘PyCharm’, the local database will be set with help of ‘SQLite’. It is convenient to store data and call the data to analyze.

3.2 Implementation of CDPA

3.2.1 Simulate Browsing Website

To meet all the requirements, the first important step for CDPA is to simulate browsing the website. As mentioned above, Selenium WebDriver is taken into account. Selenium WebDriver supports almost all the popular browsers like Firefox, Safari and Chrome [20]. Also, Selenium WebDriver can support certain rare browser like HTMLUnit browser [34]. It provides functions of loading web pages, finding elements on the pages, simulating interaction with elements and operating cookie related simulations, like getting cookies of a certain web page. Moreover, Selenium WebDriver is proved to be compatible with testing of dynamic pages and it is able to simulate realistic human operations easily and accurately. Selenium WebDriver’s basic functions are suitable for this project’s aims. As a result, Selenium WebDriver is decided to be used in the project. Google Chrome is set as the default browser to load the pages in the project. One reason is that the Selenium WebDriver is compatible with Chrome. One official statistic list shows that Google Chrome browser has market share of 64.6% in 2020 and temporary 63.58% market share in 2021 and it can be called as a popular browser [16]. Collect data by browsing the websites with a popular browser can be more reliable. Moreover, Firefox officially claimed that all the cross-site tracking has been banned by default and cross-site tracking means that third-party websites track users with third-party cookies [8]. Compared with Google Chrome, Firefox contains more strict limitations with cookies, to avoid unnecessary settings of web driver, Google Chrome is chosen. As a result, to simulate the automation browsing, Selenium Chrome WebDriver is chosen at last and it takes advantage of Google Chrome (Google Chrome version: 90.0.4430.212) to complete the whole project. After the decision, the correct web driver was installed under the same directory of Chrome according to the version of Chrome.

With help of web driver, the pages can be loaded and visited easily with methods provided by the ‘selenium’ library. When the URL of a website is provided, the web driver can easily visit the website with method of ‘get(URL)’. In this project, the website list is chosen from ‘Tranco’ list [7]. It is a comprehensive list that combines the websites listed on the ‘Alexa’ list, ‘Umbrella’ list, and ‘Majestic’ list. These three lists rank popular websites with their criteria. ‘Tranco’ list aggregates the ranks from them and provide a comprehensive popular website list. In the ‘Tranco’ list, domain of each popular website is provided. However, the web driver needs a complete URL to visit the website successfully, it is important to modify the domains provided by the ‘Tranco’ list firstly so that the web driver can visit each website. Main protocols used on the Internet now are Hypertext Transfer Protocol (HTTP) and Hypertext Transfer Protocol Secure (HTTPS). HTTPS is a more secure protocol compared with HTTP protocol and HTTPS has been adopted in most of websites concerning with data confidentiality [32]. However, depending on the cost, some websites still use the HTTP, so in this project, URL should be defined from two directions. As most of websites have applied HTTPS, HTTPS will be considered firstly. The form of *‘https://www.domain.com/’* is used to run each website, each domain from the ‘Tranco’ list will be firstly added with prefix of ‘https://’ and suffix of ‘/’ (the suffix can be omitted) for the web driver to visit the website normally. If the form is fixed merely with this kind, some websites still cannot be visited successfully. For the websites that cannot be visited successfully, the domain will be recorded in the other database and these domains will be modified in the forms of *‘http://www.domain.com/’, ‘https://domain.com/’ and ‘http://domain.com/’*. Each form will be used only when there are still some errors when visit the websites. For example, after the round of visiting the websites in the form of ‘https://domain.com/’ and locating the websites’ cookie dialogs, if the returned value shows that there is still a website in the database that the driver did not visit successfully, it will continue to modify the domains with the form of ‘http://domain.com/’, otherwise the process of visiting websites in different forms will be stopped. It can also save much time during the experiment. After four kinds of forms are tried, the websites that still have connection errors will be considered as invalid domains and these domains will not be used and tested in the later experiment. After the web driver successfully visit the websites, each website’s URL will be recorded in the database for future use. The URL can be directly called and used from the database, and it can also save much time.

3.2.2 Locate Cookie Dialogs

The figure 3.2 shows an overview of system structure of simulating browsing and locating cookie dialogs. To locate cookie dialogs, basic knowledge of cookie dialogs

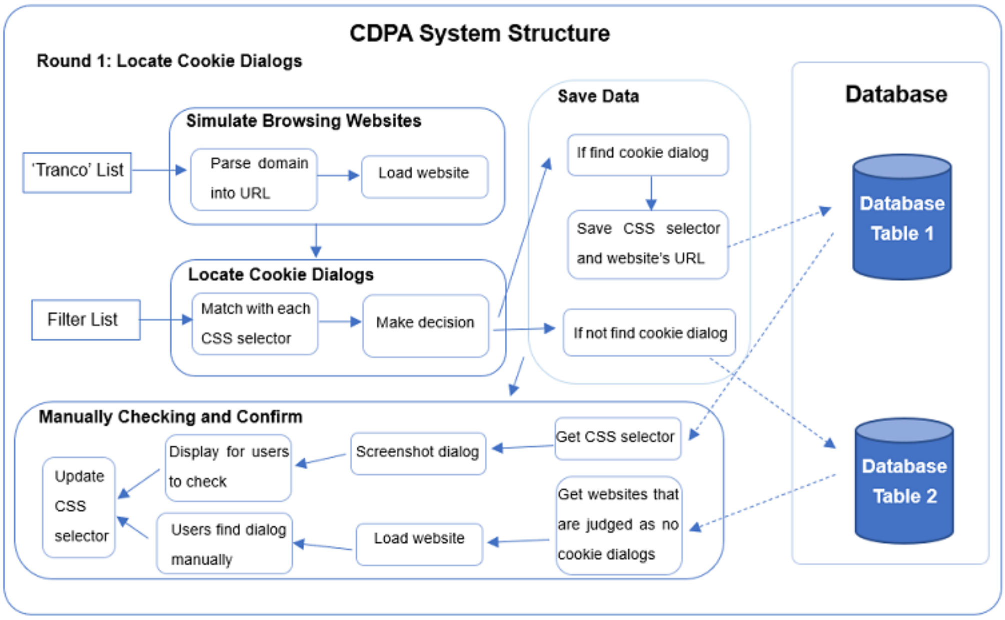
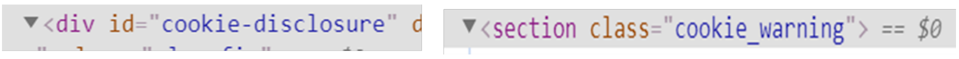

CDPA System Structure
Round 1: Locate Cookie Dialogs
'Tranco' List
Simulate Browsing Websites
Parse domain into URL
Load website
Filter List
Locate Cookie Dialogs
Match with each CSS selector
Make decision
Save Data
If find cookie dialog
Save CSS selector and website's URL
If not find cookie dialog
Database
Database Table 1
Database Table 2
Manually Checking and Confirm
Get CSS selector
Screenshot dialog
Display for users to check
Update CSS selector
Users find dialog manually
Load website
Get websites that are judged as no cookie dialogs

Figure 3.2: System diagram of round 1

should be learnt firstly. Several websites were visited in advance to explore the normal regulations and settings about cookie dialogs. It has been noticed that some websites indeed do not display cookie dialogs, such as 'http://www.baidu.com/' and many websites in the Asia do not display cookie dialogs when users visit them on the first time. However, several websites may display multiple cookie dialogs. It proves that one website can contain several parts of codes for cookie dialogs. With observation of words displayed in the cookie dialogs, one keyword is observed in each cookie dialog, it is 'cookie'. After checking the page source, as shown in the figure 3.3, each cookie dialog is always defined with an ID name or a class name. Position of cookie dialogs is variable, and some cookie dialogs may become visible after scrolling down the page. After basic knowledge of cookie dialogs, there was one attempt. One idea of locating cookie dialogs by searching keywords in the source code was initially came up with. Several keywords were listed, they are 'cookie', 'track' and 'third parties'. However, after several experiments and analysis from Molnar [31], this idea was abandoned. Even one page does not display a cookie dialog, it may still contain these keywords in its source codes, so if this idea was implemented, the accuracy will be very low. Moreover, even the tool can judge whether a website contain a cookie dialog or not with this method, it will not be feasible and usable for the other functions.

Finally, CSS selector is decided as the main method to locate the cookie dialogs. 'AdblockPlus' (available in [1]) is one extension for users to block advertisements



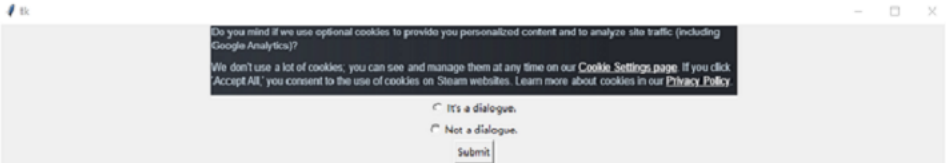
```
▼<div id="cookie-disclosure" d
```

```
▼<section class="cookie_warning"> == $0
```

Figure 3.3: Example of cookie dialog settings in the source code

on the websites and ‘I don’t care about cookies’ (available in [6]) is an extension to remove all the cookie warnings from the websites. Both ‘AdblockPlus’ and ‘I don’t care about cookies’ take advantage of filter list to judge advertisements and cookie warnings. The filter list used by ‘AdblockPlus’ comes from EasyList (available in [4]), it also provides a filter list called ‘EasyList Cookie List’ for blocking cookie banners. The filter list used by ‘I don’t care about cookies’ is classified by the author himself. Moreover, compared with ‘I don’t care about cookies List’, ‘EasyList Cookie List’ is listed with a more specific categories, so ‘EasyList Cookie List’ is applied in this project. ‘EasyList Cookie List’ contains CSS selector for different aims. After the consideration, several important parts were retained for the future use. Firstly, the CSS selector for searching general element and specific element were retained. Moreover, the CSS selector for some special countries’ websites were also saved in the filter list. The filter list actually contains two parts, one contains general element’s CSS selector and the other contains specific websites element’s CSS selector. In the filter list, the form of displaying general element’s selector is ‘##selector’, the selector could be ‘#ID name’ or ‘.class name’. The form of displaying specific element’s selector is ‘domain+## selector’. The selector begins with ‘#’ represents to select all the elements with a certain ID name, and the selector begins with ‘.’ represents to select all the elements with a certain class. Also, there are some CSS selectors like ‘#ID > .class’, it means that the cookie dialog can be located in a class under an element called ‘ID’.

To achieve each page’s HTML code, Beautiful Soup is applied to parse each page’s HTML code. One phenomenon was also noticed that the whole HTML code will not appear immediately, so the best method to achieve completed HTML codes is to wait for seconds. There are three efficient ways to keep the program waiting. One is a basic code ‘time.sleep()’, it is a kind of code that forces the program to wait, no matter in which kind of situation. Other method is called implicitly waiting. It is one of methods from ‘selenium’ library, it should combine with a driver. It can be implemented with code of ‘driver.implicitly_wait()’. This method is more complex, the time set by this line of code is a maximum waiting time. If the browser loads the page within this time, it will continue to complete the next step immediately. The last method is ‘WebDriverWait’, it should combine with ‘until()’, but this method is a little complex, it needs the code to be specific enough. Finally, the first and second methods were applied to achieve completed HTML codes. Some cookie dialogs are set in the iframe and the way to achieve the HTML codes within it is to switch to that iframe firstly. To find the codes within an iframe, the web driver firstly finds the number of iframe in a page with 'find_elements_by_tag_name('iframe')'. Then, the driver can switch to each iframe to obtain the HTML codes wrapped in each iframe. After achieving each website's HTML codes, the driver could search and match the codes with each CSS selector listed in the filter list. If there is one CSS selector matched, this CSS selector will be saved in the database. To increase the accuracy of CDPA, the tool will compare each website's HTML codes with both general elements and specific elements listed in the filter list, although it consumes much time. In addition, several keywords were added to help further judge the cookie dialogs. Some dialogs that are not cookie dialogs can also be found with the CSS selector in the filter list. To decrease this kind false positive errors, several keywords were summarized for a deeper judging. Most of cookie dialogs display the words like 'cookie', 'track', 'experience', 'third parties' and 'personal data' [40]. After locating the corresponding CSS selector for the cookie dialog, CDPA will judge if one of these five phrases is shown on the located dialog.

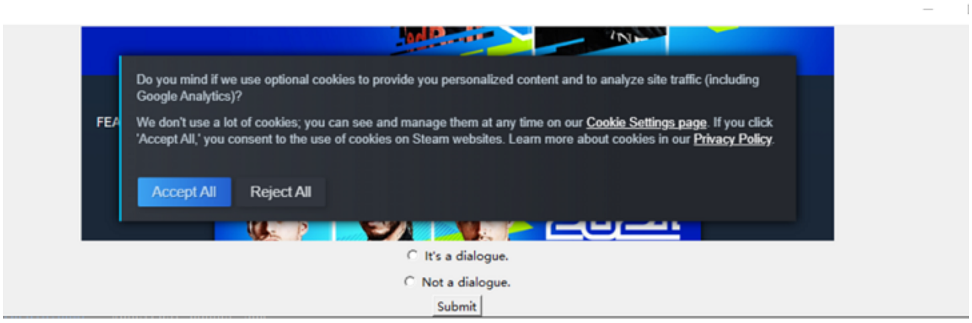


Do you mind if we use optional cookies to provide you personalized content and to analyze site traffic (including Google Analytics)?

We don't use a lot of cookies; you can see and manage them at any time on our Cookie Settings page. If you click 'Accept All,' you consent to the use of cookies on Steam websites. Learn more about cookies in our Privacy Policy.

Figure 3.4: UI for displaying a wrong cookie dialog

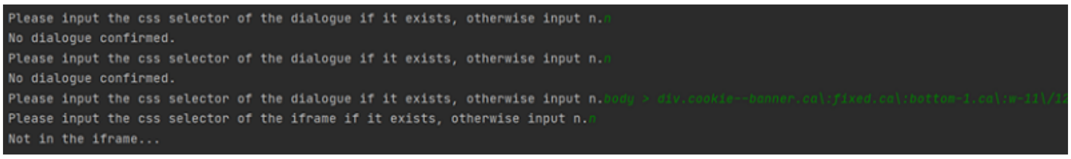
Manually checking is also set in this stage to collect more accurate cookie dialogs. Manually checking is divided into two parts. The first part is to review collected cookie dialogs and the second part is to confirm the pages that judged as not having cookie



Do you mind if we use optional cookies to provide you personalized content and to analyze site traffic (including Google Analytics)?

We don't use a lot of cookies; you can see and manage them at any time on our Cookie Settings page. If you click 'Accept All,' you consent to the use of cookies on Steam websites. Learn more about cookies in our Privacy Policy.

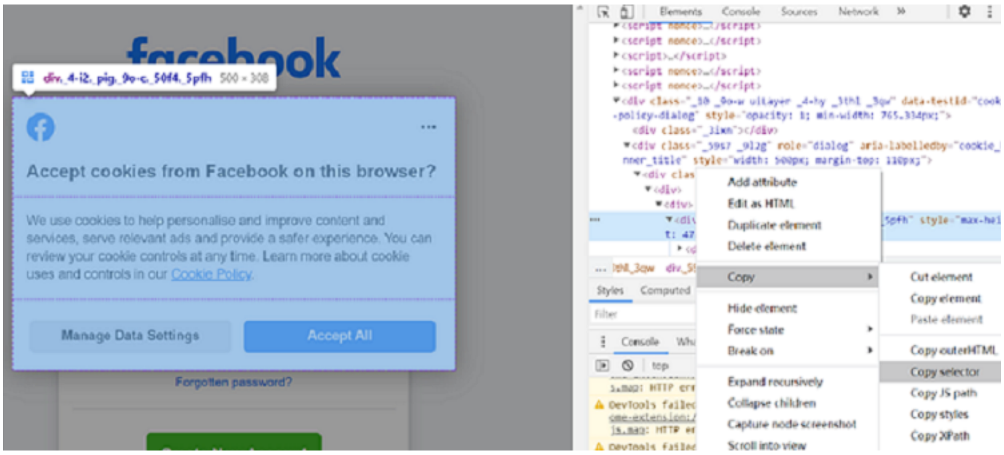
Figure 3.5: UI for displaying a correct cookie dialog



```
Please input the css selector of the dialogue if it exists, otherwise input n.n
No dialogue confirmed.
Please input the css selector of the dialogue if it exists, otherwise input n.n
No dialogue confirmed.
Please input the css selector of the dialogue if it exists, otherwise input n.body > div.cookie--banner.ca\:fixed.ca\:bottom-1.ca\:w-11\/12
Please input the css selector of the iframe if it exists, otherwise input n.n
Not in the iframe...
```

Figure 3.6: Interface for manually confirmation

dialogs. To complete the first part of manually checking, CDPA will firstly screenshot each collected cookie dialog and save it as ‘dialog.png’. This kind of method could ensure that the storage space will not be occupied too much because the next screenshot picture will replace the last figure. Some elements cannot be screenshotted successfully, one reason is that the element is an empty element, the other reason is that the element is a container. As a result, one Exception was set to avoid the errors of not screenshotting correct elements. CDPA will show the screenshot of the cookie dialog to the user and user can choose and decide if the figure is a cookie dialog via a simple UI. As shown in the figure 3.4 and figure 3.5, user will be displayed with pictures, these dialogs are located automatically by the CDPA. However, the first figure is an incorrect one and the second is the correct one. Users can click ‘Not a dialog’ for the first dialog and ‘It’s a dialog’ for the second dialog. Once a dialog is confirmed as not a cookie dialog, CDPA will delete that element immediately in the database.



div._4-i2._pig._9o-c._50f4._5pfh 500 × 308

Accept cookies from Facebook on this browser?

We use cookies to help personalise and improve content and services, serve relevant ads and provide a safer experience. You can review your cookie controls at any time. Learn more about cookie uses and controls in our Cookie Policy.

```
▸<script nonce>…</script>
▸<script nonce>…</script>
▸<script>…</script>
▸<script nonce>…</script>
▸<script nonce>…</script>
▾<div class="_10 _9o-w uiLayer _4-hy _3th1 _3qw" data-testid="cookie-policy-dialog" style="opacity: 1; min-width: 765.334px;">
   <div class="_3ixn"></div>
  ▾<div class="_59s7 _9l2g" role="dialog" aria-labelledby="cookie_banner_title" style="width: 500px; margin-top: 110px;">
```

Figure 3.7: Example of copying CSS selector (*https://www.facebook.com/*)

The second part of manually checking is to manually supplement the database and confirm if CDPA missed some cookie dialogs. CDPA will reopen the websites that it judged as not containing cookie dialogs in the first stage one by one to the users. Users should judge by themselves that if those websites contain cookie dialogs or not. As shown in the figure 3.6, users will be asked to input the CSS selector of the cookie dialog if it exists. Users can get it by inspecting the page and right click to copy the CSS selector of the chosen cookie dialog, like the figure 3.7. Moreover, if the user confirms that the website contains a cookie dialog, CDPA will further confirms if the cookie dialog is within an iframe, if yes, the user will also be required to provide the iframe's CSS selector. Once the user completes the operation, CDPA will store the relevant information in the database.

3.2.3 Locate Clickable Element

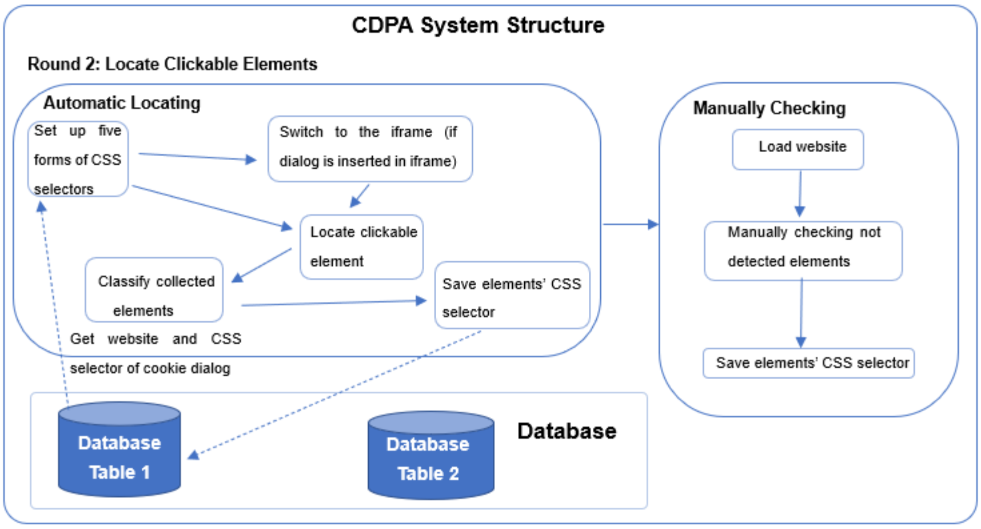

CDPA System Structure
Round 2: Locate Clickable Elements
Automatic Locating
Set up five forms of CSS selectors
Switch to the iframe (if dialog is inserted in iframe)
Locate clickable element
Classify collected elements
Save elements' CSS selector
Get website and CSS selector of cookie dialog
Manually Checking
Load website
Manually checking not detected elements
Save elements' CSS selector
Database Table 1
Database Table 2
Database

Figure 3.8: System diagram of round 2

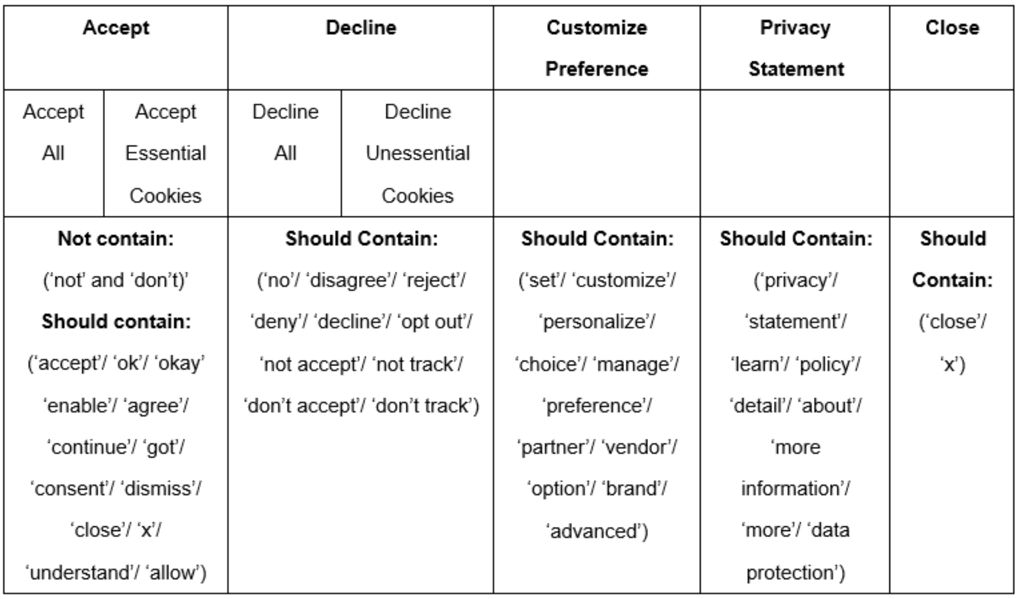
This stage mainly requires the CDPA to locate and classify clickable elements for users to accept, decline or customize their preference.The figure 3.8 provides a system diagram for viewing how CDPA works in stage 2. After the first stage of collecting cookie dialogs, CDPA have saved all the CSS selectors for each website containing cookie dialogs. These data will be called later to locate clickable elements.

Each website will be revisited by the web driver and the CSS selector for that website's cookie dialog will be called. Clickable elements are always defined with the following tags, they are '<button>', '<a>', '<span>', '<input type=submit>' and tags with attribution of 'role = button' [31]. In this project, to collect elements with these five tags, CSS selector is still applied. To collect clickable elements placed within the cookie dialog, the prefix should be defined, otherwise all the clickable elements on the whole web page will be located. As a result, four kinds of possibilities CSS selector are defined as following:

1. #CSS selector for the cookie dialog + blank + button
2. #CSS selector for the cookie dialog + blank + a
3. #CSS selector for the cookie dialog + blank + span
4. #CSS selector for the cookie dialog + blank + input[type=submit]
5. #CSS selector for the cookie dialog + blank + [role=button]

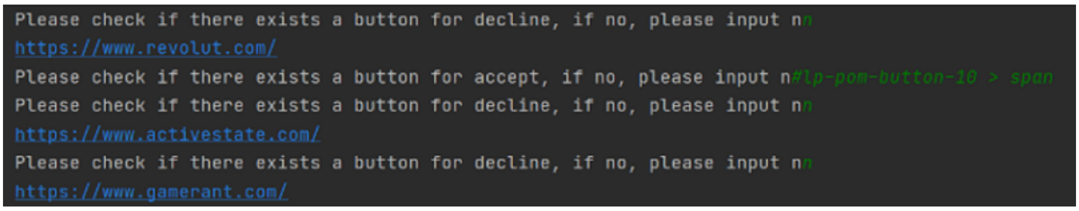
Each cookie dialog will be retrieved with each CSS selector listed above to collect clickable elements. The cookie dialogs set within iframe can also searched. CDPA will firstly switch to the iframe that the cookie dialog locates, then CDPA takes advantage of above CSS selectors to search for the clickable elements. This method can collect most of normally defined clickable elements shown on the cookie dialogs. After collection of clickable elements, the elements should be classified into different groups so that they can be used conveniently by CDPA in the future steps. These clickable elements are classified into five main groups. These five groups can be defined as 'Accept', 'Decline', 'Customize Preference', 'Privacy Statement' and 'Close'. However, there are several sub-groups defined in the 'Accept' and 'Decline'. The detailed information can be found in the figure 3.9. 'Accept' group and 'Decline' group are both divided into two small groups. However, group of 'accept essential cookies' can be seen as the same as group of 'decline unessential cookies'. In the implementation of classifying, these two small groups are merged as one group. It should mention that the project also classifies the clickable elements of 'close' into 'Accept' because 'close' was judged as accept default setting at the beginning. The group of 'Close' set in the following table also represents the operation of 'close' and it is set to record clickable elements to 'close' individually for future use of collecting cookies after clicking 'close' elements. Moreover, some websites display the choice of 'decline' after clicking 'personalize choice'. It is difficult to capture this kind of 'decline' options automatically, so process of collecting this kind of options will be completed manually in the later stage. In the HTML code, one element is always shown as '<button class="class" data-uia="data-uia">Value</button>'. To classify the collected clickable elements, the value that each element shows will be judged. The keywords to judge and classify elements into different groups are also listed in the table shown in the figure 3.9.

Moreover, different countries may use different languages to write down the value of each element. It further needs one translation API to translate different language into English firstly. Google Translate API shown by [5] is imported to complete the translation. It can take any language as root language and destination language can be set as English, then all the language will be translated into English. After translation, the keyword judgements can be completed more accurately. All the clickable elements will be classified and saved in the main database for the future data collection use. After this stage, the number of elements for accepting, declining, customizing preference, privacy statement set in the cookie dialogs can be counted. The number will also be analyzed later to judge cookie dialog setting compliance of each website.



Accept		Decline		Customize Preference	Privacy Statement	Close
Accept All	Accept Essential Cookies	Decline All	Decline Unessential Cookies			
Not contain: (‘not’ and ‘don’t)’ **Should contain:** (‘accept’/ ‘ok’/ ‘okay’ ‘enable’/ ‘agree’/ ‘continue’/ ‘got’/ ‘consent’/ ‘dismiss’/ ‘close’/ ‘x’/ ‘understand’/ ‘allow’)		**Should Contain:** (‘no’/ ‘disagree’/ ‘reject’/ ‘deny’/ ‘decline’/ ‘opt out’/ ‘not accept’/ ‘not track’/ ‘don’t accept’/ ‘don’t track’)		**Should Contain:** (‘set’/ ‘customize’/ ‘personalize’/ ‘choice’/ ‘manage’/ ‘preference’/ ‘partner’/ ‘vendor’/ ‘option’/ ‘brand’/ ‘advanced’)	**Should Contain:** (‘privacy’/ ‘statement’/ ‘learn’/ ‘policy’/ ‘detail’/ ‘about’/ ‘more information’/ ‘more’/ ‘data protection’)	**Should Contain:** (‘close’/ ‘x’)

Figure 3.9: Table of classification groups and judgement keywords



```
Please check if there exists a button for decline, if no, please input nn
https://www.revolut.com/
Please check if there exists a button for accept, if no, please input n#lp-pom-button-10 > span
Please check if there exists a button for decline, if no, please input nn
https://www.activestate.com/
Please check if there exists a button for decline, if no, please input nn
https://www.gamerant.com/
```

Figure 3.10: Manually checking elements

CDPA also has one part of manually checking element in this stage as some of elements cannot be located by CDPA successfully. CDPA will call back the websites that are automatically judged by CDPA as not containing ‘accept’, ‘decline’, ‘personalize cookies’ or ‘close’. CDPA will judge which kind of elements it did not identify and then it will confirm each kind of element with user one by one. As shown in the figure 3.10, it gives an example of how CDPA interacts with users to check elements manually. If users discover some elements were left out by CDPA, user could copy CSS selector of that element into the console, like copying CSS selector of cookie dialogs at the first stage. Then, CDPA will save the corresponding information in the database.

3.2.4 Interact with Elements and Collect Cookies

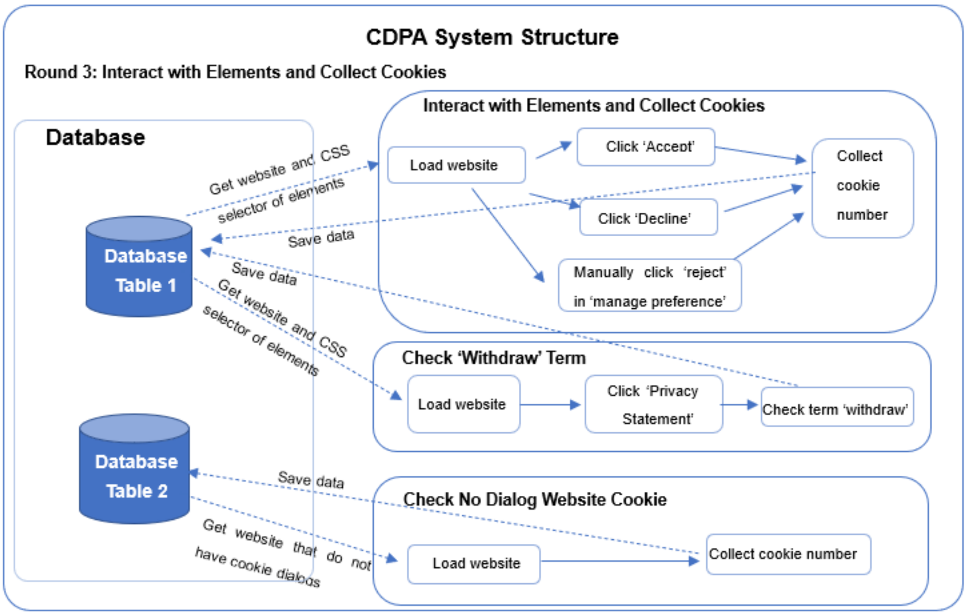

CDPA System Structure
Round 3: Interact with Elements and Collect Cookies
Database
Database Table 1
Database Table 2
Get website and CSS selector of elements
Save data
Save data
Get website and CSS selector of elements
Save data
Get website that do not have cookie dialogs
Interact with Elements and Collect Cookies
Load website
Click 'Accept'
Click 'Decline'
Manually click 'reject' in 'manage preference'
Collect cookie number
Check 'Withdraw' Term
Load website
Click 'Privacy Statement'
Check term 'withdraw'
Check No Dialog Website Cookie
Load website
Collect cookie number

Figure 3.11: System diagram of round 3

After collection of clickable elements, CDPA will attempt to interact with each kind of element to collect cookies and other needed data, the whole process is shown in the figure 3.11. To interact each element, the web driver will reopen each website and interact with founded elements. One repeated operation is that if the cookie dialog is set within an iframe, the web driver needs firstly to switch to that iframe to click the elements. One kind of exception is that the target element may be covered by other elements, and it will not be clicked successfully. Then, it will need user to manually complete clicking.

To collect cookies, one method called 'get_cookies()' from 'selenium' is firstly considered. However, this method was found that it cannot collect all kinds of cookies [40]. All the third-party cookies will be missed and not collected if this method is applied. Chrome web driver can apply 'execute_cdp_cmd('Network.getAllCookies', {})' from Chrome DevTools Protocol (avaliable in [2]) to obtain all kinds of cookies. To collect correct number of cookies, this command will be executed after command of 'time.sleep(30)', when the program waits for a period of time, the number of cookies can be more accurate. After collection of cookies, a judgement of first-party cookies and third-party cookies will be implemented by CDPA. CDPA judges the cookies mainly by analyzing the domain. It filters the collected cookie list and then extracts the domain set in the cookie, CDPA will compare the domain with the URL domain the driver is visiting. If they are the same, CDPA will judge that cookie as a first-party cookie, otherwise it is a third-party cookie.

This stage mainly aims to collect cookies after interactions with different kinds of elements, it also has the goal of counting appearance time of term ‘withdraw’ in the cookie dialog or in the privacy statement. To count the appearance time of ‘withdraw’, each cookie dialog’s HTML code will be parsed with help of ‘Beautiful Soup’. The website will be recorded if the term of ‘withdraw’ is found in the cookie dialog. If not found in the cookie dialog, CDPA will automatically interact with the ‘privacy statement’ element if this element exists. After loading the page of privacy statement, the HTML code will be parsed again to search for the keyword of ‘withdraw’. If the term is found, the system will identify that this website can provide an efficient method for users to withdraw their consents.

To collect cookies after different interactions with the cookie dialogs, CDPA will automatically visit each website again. When the page is loaded, it will wait for several seconds, then it will firstly collect default cookies set by the website before interacting with elements. When the number of cookies is collected, CDPA will interact with ‘accept’ elements and it will also collect the corresponding number of cookies after this operation. CDPA will repeat this operation until all the websites are visited. After this round, CDPA will then begin the second round to visit all the websites. It will open each website again and it will automatically interact with ‘decline’ elements. After seconds’ waiting, it will collect the number of cookies after interaction of ‘decline’. As for the interaction with ‘decline’ elements set within ‘manage preference’ page, manually operation is needed. Users will be asked to interact with the cookie dialogs containing choices of ‘manage preference’ and click the ‘decline’ option manually if it exists. Also, CDPA will record the cookie number after manually clicking.

3.2.5 Reload Pages to Locate Cookie Dialogs

The last round is to satisfy the goal of comparing the consistency of whether showing cookie dialogs when reloading the page after interacting with ‘accept’ element and ‘decline’ element. This round’s system diagram is shown in the figure 3.12. However,

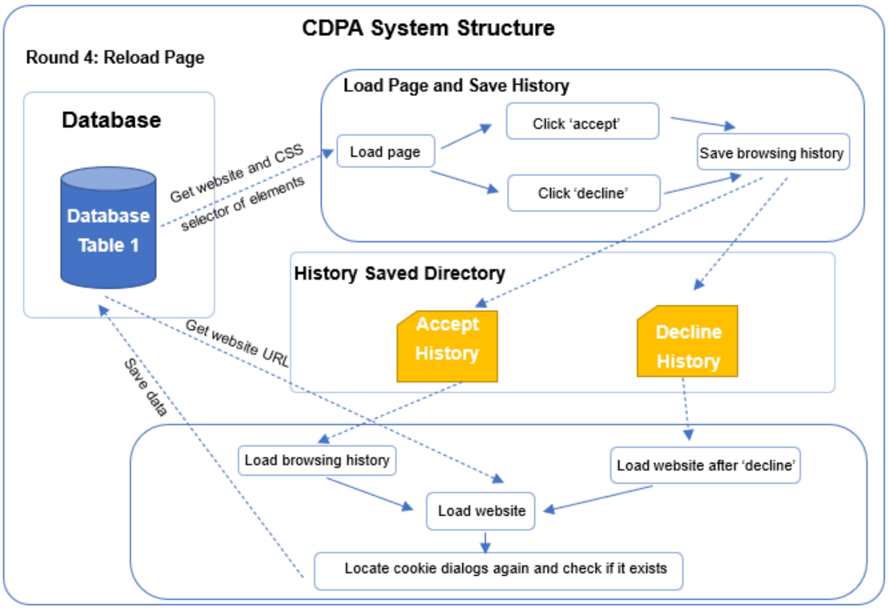


Figure 3.12: System diagram of round 4

Selenium cannot retain browsing history as it will clear all the history and data after each round of starting up the Chrome web driver. To retain the history of clicking 'accept' or 'decline', one directory for saving users browsing data should be created in advance. Directories are separately created for clicking 'accept' and 'decline' to save the data and visiting record separately. Then, when the web driver begins to interact with pages, the browser history will be added with 'add_argument("user-data-dir=Directory ")' when the web driver is initialized. The web driver will interact with 'accept' choice and 'decline' choice again after over 12 hours. Interaction with 'decline' choice set in the 'manage preference' will also be manually completed. After two rounds of interactions, the web driver will start another two rounds to check if these pages still display the cookie dialogs. To locate cookie dialogs, it is completed with help of previously collected cookie dialogs' CSS selectors. Final results will also be saved in the database.

Chapter 4

Experiment and Evaluation

This part introduces the experiments implemented by CDPA. The first experiment was more like a small test to find some possible improvements and tiny bugs with CDPA. The second experiment was an official test to collect all kinds of data for the later data analysis. Final result in the second experiment will be analyzed with help of several graphs showing detailed data. Moreover, evaluation of CDPA's behavior in the second experiment is also introduced specifically in this part.

4.1 Experiment Process

After implementation of CDPA system, CDPA was used to complete two rounds of experiments. The first experiment was with testing 200 websites randomly selected from the top 10,000 popular websites from 'Tranco' list. The first round of experiment aimed to calculate the basic accuracy of CDPA and attempted to discover some points to improve with CDPA system. After the testing with 200 websites, the accuracy of locating cookie dialogs is around 85%. After the first round of experiment, one improvement was noticed. There are 115 in 183 valid websites judged as not containing cookie dialogs, but there are 22 websites in these 115 websites indeed containing cookie dialogs. In these 22 websites, several websites cookie dialogs' CSS selector are marked as '# qc-cmp2-ui', '#notice' and '#CXQnmb' after manually checking. After confirming with the 'EasyList Cookie List', these three commonly used CSS selector are left out in this filter list. To improve the ability of CDPA crawling cookie dialogs, these three elements were added in the filter list at last. Three omitted CSS selectors are marked in the form of '###qc-cmp2-ui', '###notice' and '###CXQnmb'.

After modifying the filter list downloaded on the 7th of July by adding three extra

CSS selectors, the final experiment was implemented. 1,000 websites were chosen randomly from 20,000 top popular websites listed in the ‘Tranco’ list (downloaded on the 7th of July). As ‘Tranco’ list only provides domain name of website, it is possible to include same sub-websites that belong to one main website. To ensure the randomness, this kind of possibility was negligible. CDPA ran each stage mentioned in the ‘Implementation’ part. It firstly located the cookies dialogs of 1,000 websites and it recognized part of valid websites with or without cookie dialogs and invalid websites. Then manually checking with cookie dialogs was completed. The process of locating cookie dialogs and clickable elements were both finished between 7th of July and 10th of July. After collection of clickable elements, CDPA interacted with each collected element to collect number of cookies and information after reloading the pages. These operations were finished and modified between 13th of July and 20th of July.

4.2 Experiment Result and Analysis

4.2.1 Cookie Dialog Setting Situation

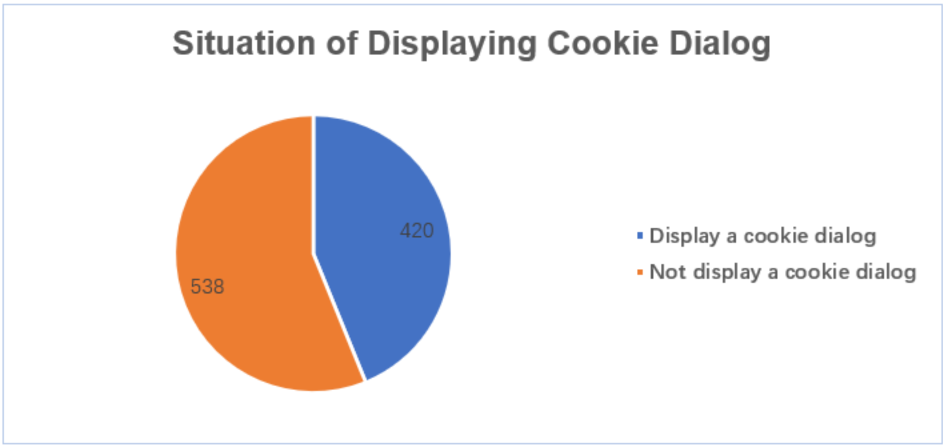


Figure 4.1: Setting of cookie dialog

958 websites in 1,000 websites were judged as valid websites. There are 420 websites in 958 websites judged as displaying cookies dialogs while the rest of 538 websites are judged as not displaying cookies dialogs. In other words, there are about 43.8% websites containing cookie dialogs on the website homepages. The statistics are directly shown in the form of pie chart. The websites not displaying cookie dialogs are mainly Asia websites, especially in China. Several Chinese companies’ websites were picked up. One URL called ‘https://www.tencent.com/’ which gives access to

Tencent international website, one cookie dialog was found in it. However, several websites like ‘https://www.focus.cn/’ and ‘https://www.chinacdc.cn/’ were found not containing cookie dialogs. From domain point of view, these websites all contain ‘cn’ in the domain which is a special Chinese domain. China has a strict Internet censorship and Internet filtering regulations. Some international websites are proved to be blocked by China with Internet located in China [47]. One assumption is that the special domains like ‘cn’ is specially set for the China Internet and no cookie consent is needed in China, so most inner Chinese websites do not display cookie dialogs.

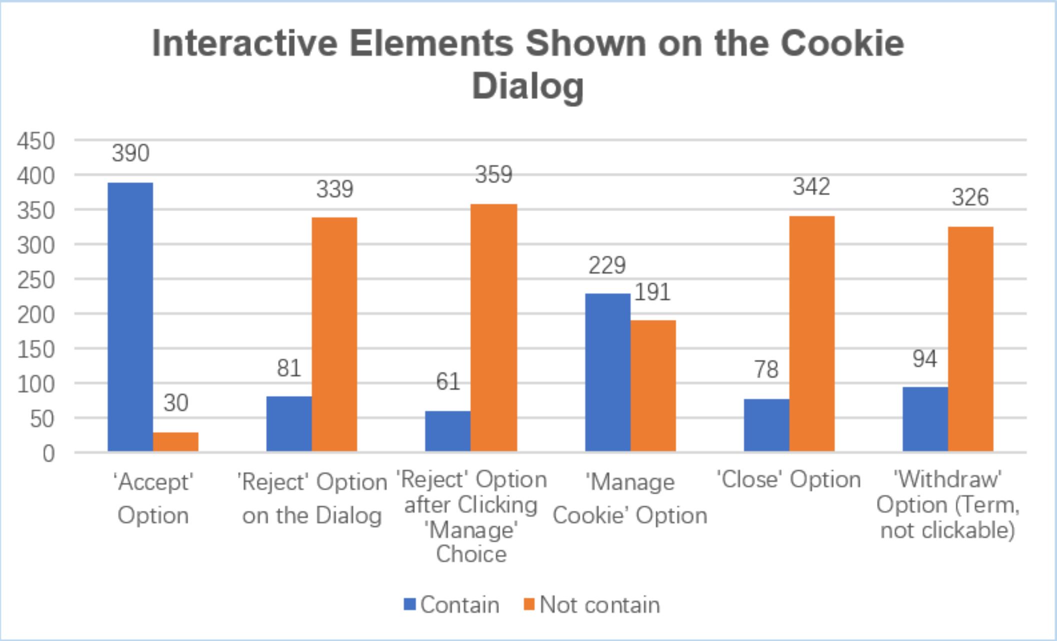


Figure 4.2: Interactive elements on the cookie diaogs

The result of interactive elements shown on the cookie dialogs is also displayed. It is shown in the form of bar chart (figure 4.2). Most of cookie dialogs provide clickable options for users to interact, except four websites which only show cookie dialogs without clickable elements. These four websites only display the cookie dialogs to inform people that they are using cookies and users cannot do anything with dialogs, even to close dialogs. There are 78 websites showing choice of ‘close’, but 26 in these 78 websites only contain one option of ‘close’ and it also limits users to show their attitudes. Users can only close dialogs instead of expressing their consents. Option of ‘reject’ is also divided into two types, one type is the ‘reject’ option directly displayed on the cookie dialog and the other type is that the ‘reject’ option appears after the option of ‘manage cookie’ is clicked. Two kinds of ‘reject’ options are separately shown with two bars. In addition, the last bar shows the number of websites that display the term of ‘withdraw’. Almost all of websites do not directly display clickable option ‘withdraw’ for users and they only mention the term of ‘withdraw’ on the dialogs or in the privacy statement. It is also listed as one criterion to judge websites’ cookie compliance.

4.2.2 Cookie Setting Result and Analysis

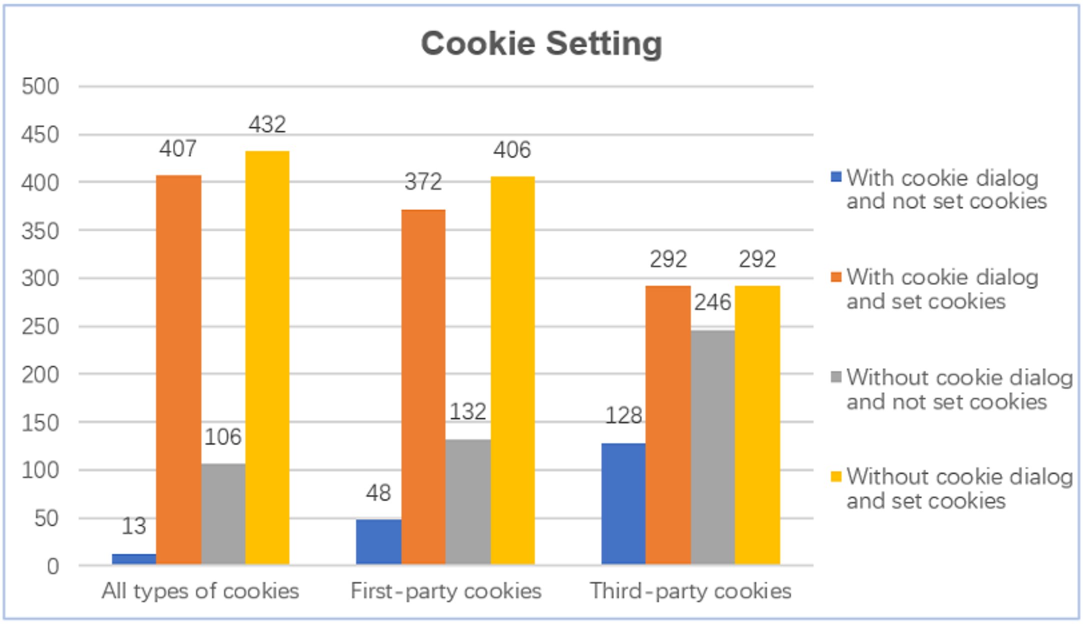


Figure 4.3: Default cookie setting in websites

Cookie setting behavior is also analyzed after collection of cookie number with different kinds of interactions. Firstly, as shown in the figure 4.3, default cookie setting is presented in the form of bar chart. This chart’s description objects are websites with cookie dialogs and websites without cookie dialogs. Most of websites set cookies on the user’s first visiting time, no matter they contain cookie dialogs or not. However, it is reasonable for websites to set first-party cookies before achieving users’ consents. When it focuses on the third-party cookies setting, 69.5% websites with cookie dialogs set third-party cookies before achieving users’ consents and 54% websites without cookie dialogs set third-party cookies when users visit them for the first time.

After analysis of default cookie settings on the websites, cookie setting changes after different kinds of interactions are also analyzed. Distribution of cookie number changes after different operations is represented in the form of box and whisker plot. This kind of plot mainly displays distribution of number by calculating median, max-

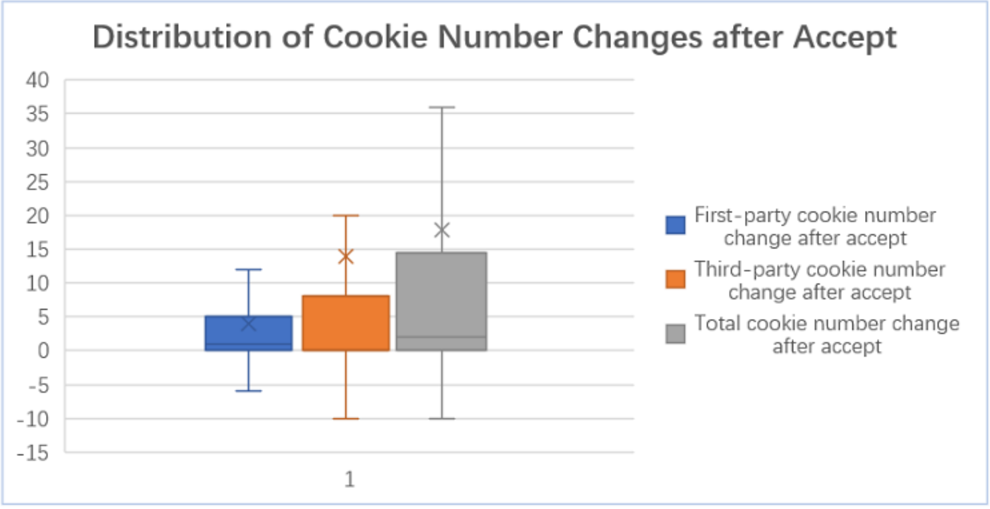


Figure 4.4: Cookie number changes distribution after accepting

imum, minimum, first quartile (Q1) and third quartile (Q3) of numbers. To show the distribution more clearly, the outlier points are removed from the plots. These box and whisker plots are constructed by numbers that minus with each other. For example, figure 4.4 shows the distribution of cookie number changes after clicking ’accept’. If the distribution covers the field that is greater than 0, it means that there is a growth in the cookie numbers and most of websites set more cookies after operation. Otherwise, if the box is below 0, it means that most cookie numbers are decreased and most of websites set fewer cookies after operation. It can conclude that there is an obvious growth in all types of cookie numbers after clicking ‘accept’ option.

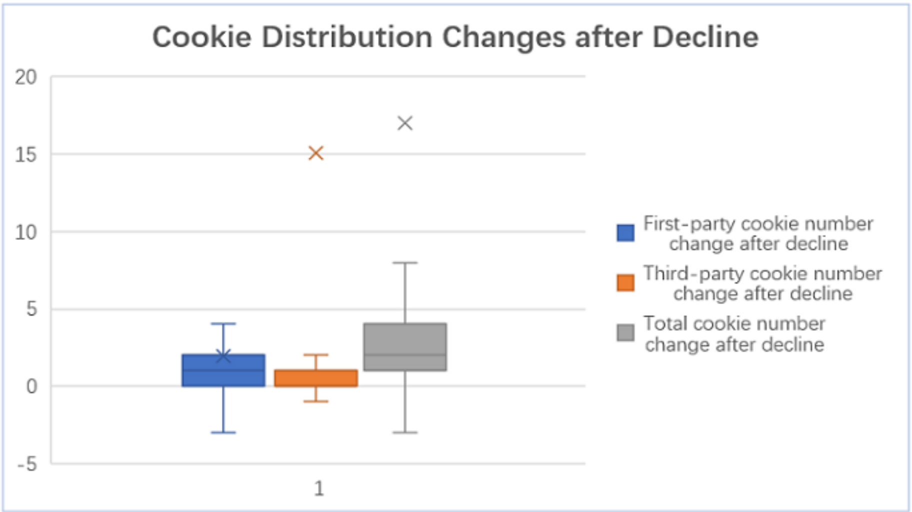


Figure 4.5: Cookie number changes distribution after declining

Figure 4.5 shows the distribution of cookie number changes after clicking ‘decline’ option. The box shows that most of websites’ cookies still increase after clicking ‘decline’, but the size of box is much smaller than the box in figure 4.4. The result also shows that only 11 websites set fewer cookies (all types of cookies) after choosing ‘decline’ option and 17 websites set fewer third-party cookies after ‘decline’ choice is chosen. Figure 4.6 shows the distribution of cookie changes between clicking accept and clicking decline. It calculates cookie number changes by ‘cookie number after decline minus cookie number after accept’. It can be observed that large parts of boxes are placed under 0. It means that fewer cookies are set after clicking ‘decline’ comparing with cookie numbers after clicking ‘accept’, although option of ‘decline’ can not avoid growth of cookie number, this kind of option can indeed reject some unnecessary cookies.

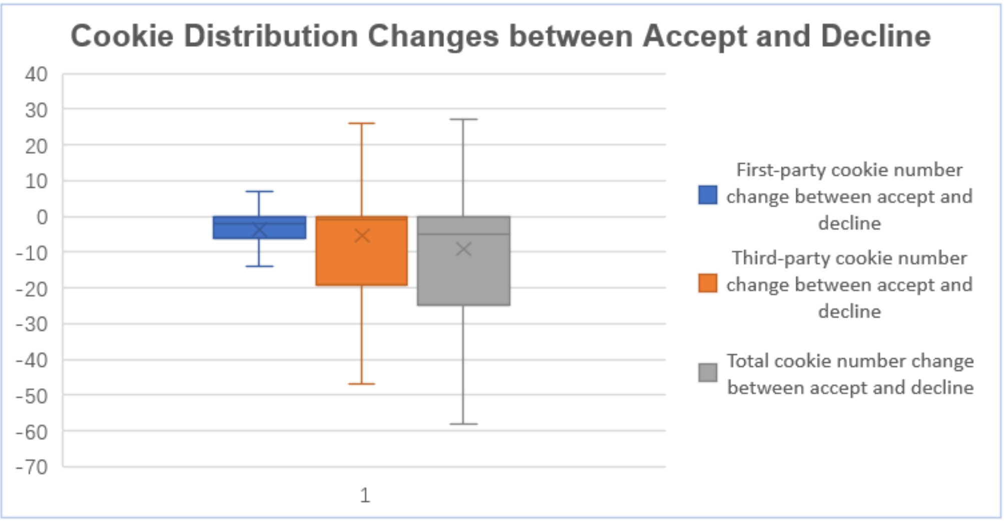


Figure 4.6: Cookie number changes distribution between accepting and declining

Distribution of cookie number change after clicking ‘close’ is also analyzed. The box plot is shown in the figure 4.7. Compared with distribution of clicking ‘accept’ and distribution of clicking ‘decline’, first-party cookie number change and all types of cookie number change are more similar with that of clicking ‘accept’ while third-party cookie number change is more like that of clicking ‘decline’. From this point of view, option of ‘close’ also has effects on the cookie numbers and its affects is similar to the option of ‘accept’. When users click option of ‘close’, it can be equal to clicking option of ‘accept’ in some ways. However, compared with clicking ‘accept’, choice of ‘close’ will generate fewer third-party cookies. From third-party cookies of view, clicking ‘close’ is equal to clicking ‘accept’.

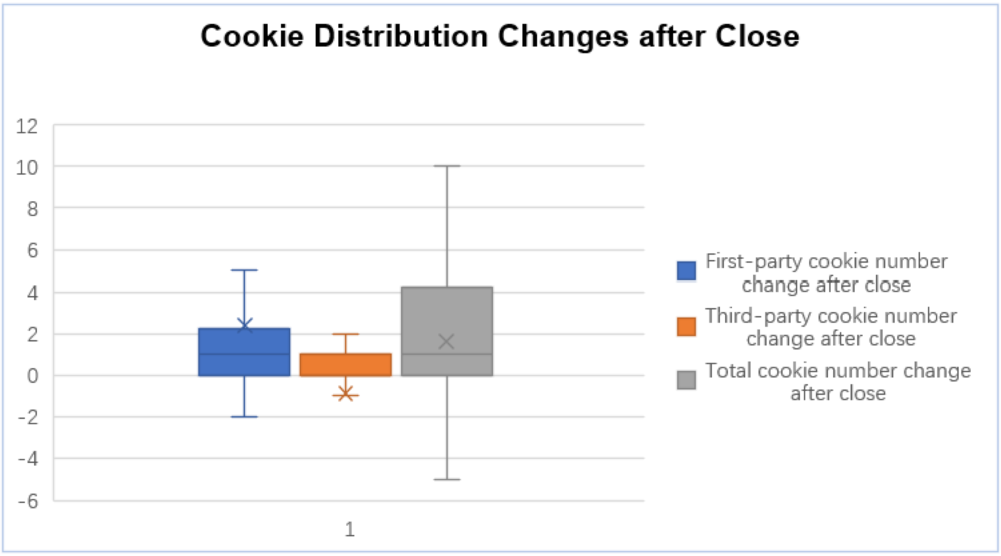


Figure 4.7: Cookie number changes distribution after close

4.2.3 Cookie Compliance Result and Analysis

Figure 4.8 shows a summary kind of bar chart about cookie compliance of tested websites according to GDPR. The criteria are mentioned in the section 2.3.2. For the first and second criteria, they are about asking for users' consents before setting unnecessary cookies with words of introducing cookies. Whether meeting these two requirements can be judged by if websites display cookie dialogues when they attempt to set unnecessary cookies (third-party cookies). As for the third criterion, it is about users should be given choices to accept or decline, it can be judged from two aspects, one is that if there are interactive elements (excluding option of 'close') set on the cookie dialogs and the other is that whether websites indeed follow users' choices to decrease cookie numbers after clicking 'decline'. The last criterion is about if users can withdraw their consents, as mentioned above, it will be judged by checking if the term of 'withdraw' is mentioned in the privacy statement. By analyzing the bar chart (figure 4.8), it will be easy to know how many websites set cookies compliantly.

The result shows that there are 712 websites (74.3%) satisfying cookie dialog setting regulations. Cookie dialog setting can be analyzed from two aspects: the websites which set up unnecessary cookies (like third-party cookies) should display cookie dialogs to users to obtain their consents and other websites that do not set up unnecessary cookies can choose not to display cookies dialogs. In these 712 compliant websites, there are 420 websites showing cookie dialogs. Although other 298 websites do not contain cookie dialogs, they do not set third-party cookies, so they can also be judged

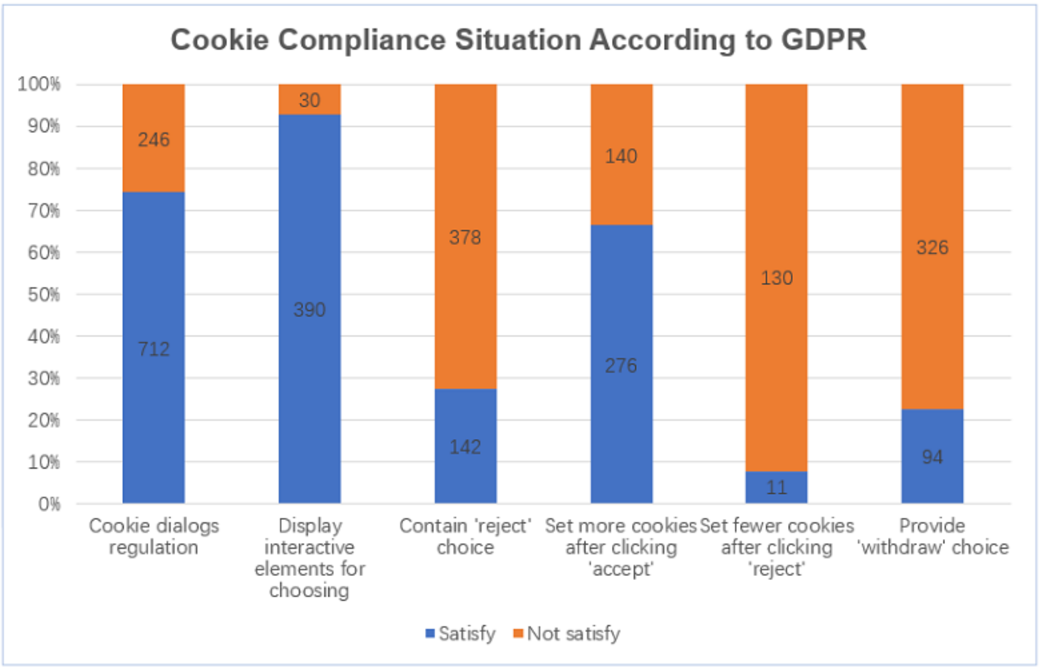


Figure 4.8: Cookie compliance situation

as compliant websites according to cookie dialog regulation. As for the interactive elements, the analysis is targeted on the websites containing cookie dialogs. There are 4 websites show cookie dialogs in form of a small window without providing any interactive elements although they display cookie dialogs. There are 26 websites only show ‘close’ option for users instead of providing more specific interactive elements. In these websites displaying efficient interactive elements, there are only 142 websites providing ‘reject’ options (including ‘reject’ option shown in the ‘manage preference’), so only 33.8% of 420 websites meet the requirement of setting enough interactive elements on the cookie dialogs for users to choose. With regard of right of withdrawing consents, as the measurement method is limited by only tracking term of ‘withdraw’, the result is that 94 websites (22.4% in 420 websites) are identified as compliant websites to provide the approaches for users to withdraw consents.

Cookie setting compliance can also be concluded by analyzing the data shown in the bar chart. Specific cookie setting behavior is measured and analyzed in the section 4.2.2. This part’s analysis mainly judges the meaning of setting different kinds of elements. As shown in the bar chart, for the websites that provide ‘reject’ options, there are 91 websites indeed decrease or not change the third-party cookie number after declining cookies use. In contrast, other 50 websites do not implement any essential changes to third-party cookies as the number of third-party cookies is increased. To summarize, about 35.4% websites that provide ‘reject’ options do not meet the requirement of setting suitable cookies after users expressing their attitudes of rejection.

4.2.4 Cookie Dialog Setting after Reloading Page

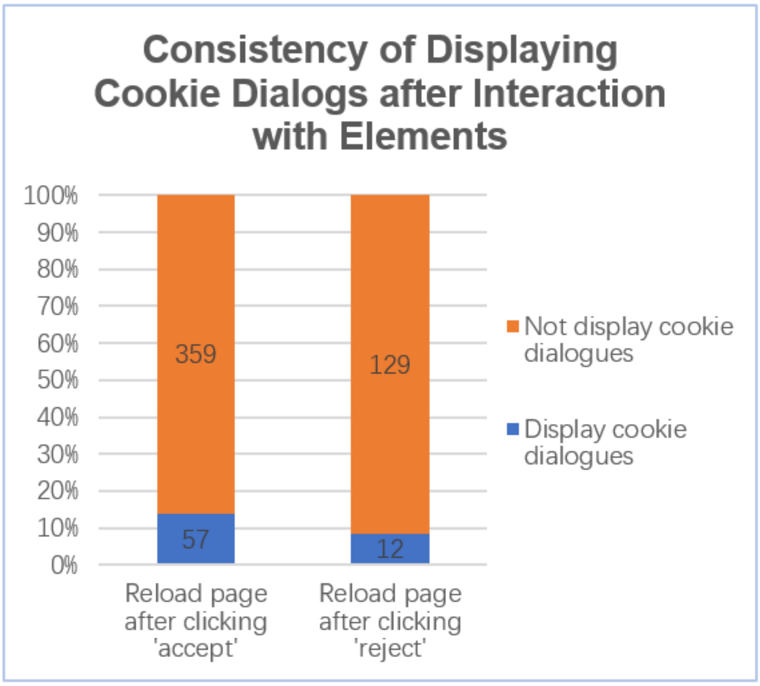


Figure 4.9: Situation of showing cookie dialog after reloading pages

One extra research about if websites will show cookie dialogs again after users submit their choices. Figure 4.9 shows that there are 57 in 416 (about 13.7%) websites show cookie dialogs again when users submit choice of ‘accept’ on the first visiting while 12 in 142 (around 8.5%) websites show cookie dialogs again when users submit choice of ‘decline’ on the first visiting. Clicking on the ‘reject’ option also include clicking ‘reject’ in the ‘manage preference’ part. The result is achieved by reloading pages after 12 hours since the first decision was made. To confirm the result’s correctness, after one week, the websites were tested again, and the result is still the same as before. To be more specific, 2 in 12 websites that show cookie dialogs again after choosing ‘decline’ do not show cookie dialogs again after choosing ‘accept’. Moreover, one assumption about not all the websites show cookie dialogs again is that some websites only show a small button instead of cookie dialogs when the page is reloaded. One example of small button (marked with an orange box) is shown in the figure 4.10, some websites will not show cookie dialogs again no matter users’ choices, they will

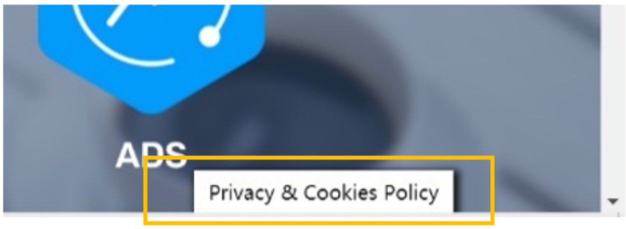


Figure 4.10: Example of button shown by the website (*https://www.worldnow.com/*)

show one clickable element for users to click to chow the cookie dialog again. Moreover, this measurement is based on choosing all kinds of ‘decline’, it includes the ‘reject’ option directly shown on the cookie dialogs and that inserted in the ‘manage preference’ part. The other assumption is that most websites have thoughts that users usually have a fixed decision once they have clicked the option of ‘manage preference’, then they will not show cookie dialogs again to ask users repeatedly decide.

4.3 Evaluation

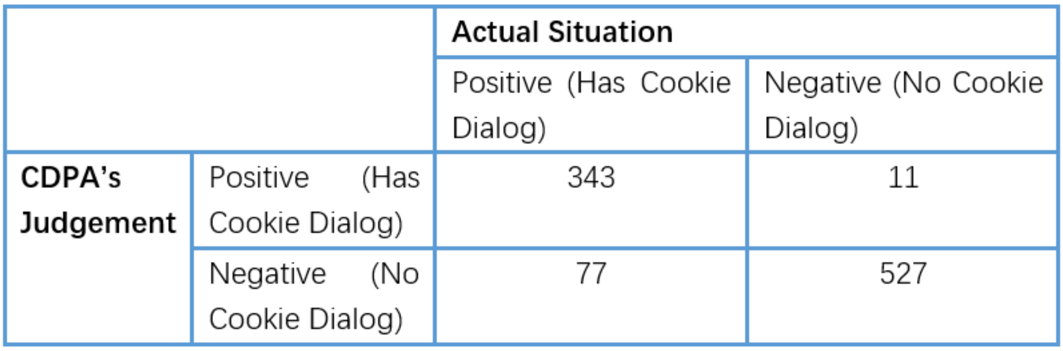
4.3.1 Evaluation about CDPA’s Efficiency

Efficiency of one system always decides the time spent on the system working process and if one system is efficient enough, it can be more reliable. Efficiency can be judged from how many times the system is forced to stop, how many issues happen during working process and if it is efficient for humans to interact with UI in the system.

In the experiment with 1,000 websites, the system was stopped for one time during the time of locating cookie dialogs. The interruption was because the update of the laptop and it closed the program automatically. Also, quality of Internet also affected CDPA, it took much time to response when the quality of Internet was bad. CDPA was restarted manually for several times because of unstable Internet and Internet problems. Meanwhile, UI of judging cookie dialogs was found too simple. Some cookie dialogs that CDPA captured are difficult to identify by users and users are not able to have correct judgments. The choices for users to choose are too few and users cannot choose the most suitable options. As a result, efficiency for manually checking module is not high. In total, CDPA can run without technical problems during several crawling stages, and it can collect most of needed data automatically. From ‘efficiency’ point of view, CDPA is efficient enough to complete whole work. It is also reliable to crawl needed data and it can recover from small issues immediately to continue the work.

4.3.2 Evaluation about CDPA's Accuracy

CDPA tested 1,000 websites and there are 42 invalid websites. Invalid websites mean that those websites have connection errors, and they cannot be visited normally. After automatically crawling cookie dialogs and manually checking with cookie dialogs, CDPA's accuracy of locating cookie dialogs is calculated. CDPA judged 354 websites containing cookie dialogs but 11 in them were confirmed as not containing cookie dialogs. In the left 604 websites that judged as no cookie dialogs by CDPA, there are 77 websites containing cookie dialogs that judged incorrectly by CDPA. The accuracy of CDPA to locate cookie dialogs automatically is around 90.8%. The specific situation of judging is shown in the following confusing matrix (figure 4.11).



		Actual Situation	
		Positive (Has Cookie Dialog)	Negative (No Cookie Dialog)
CDPA's Judgement	Positive (Has Cookie Dialog)	343	11
	Negative (No Cookie Dialog)	77	527

Figure 4.11: CDPA's accuracy of judging cookie dialogs

Accuracy can also be calculated by judging from the function of locating clickable elements. CDPA located needed clickable elements from the websites containing cookie dialogs. There are 420 websites that displaying cookie dialogs. CDPA located 278 'accept' elements, 47 'decline' elements (not including 'decline' elements after clicking 'manage preference') from the 420 websites. The actual situation is that there are 390 websites containing 'accept' elements and there are 81 websites containing 'decline' elements. The other two kinds of elements 'manage preference' and 'privacy statement' were not confirmed manually. From the view of locating 'accept' and 'decline' elements, the accuracy of locating 'accept' elements is around 73.33% and accuracy of locating 'decline' elements is around 91.9%.

All the accuracy number listed above is based on the automatically locating by CDPA. If the operation of manually checking was completed carefully, the accuracy of locating cookie dialogs and clickable elements should be up to 100%.

Chapter 5

Conclusion

This chapter firstly summarizes total work and findings of the project. Then, some recommendations for GDPR future imrpovements and for users are put forward. At last, related future work is discussed to take a deeper research about cookie settings and cookie dialog settings.

5.1 Summary of Project

In conclusion, this project developed an automatic crawler (CDPA) successfully to collect cookie related data and generated a dataset to measure cookie setting's compliance according to the GDPR. With help of filter list and CSS selector, CDPA can run without any technical problems to locate cookie dialogs. Finally, CDPA can automatically locate cookie dialogs with accuracy of 90.8% while accuracy of automatically locating elements for accepting is 73.3% and 91.9% for declining. Manually checking is also added during the experiment to make the result to be more accurate. CDPA is also evaluated from aspect of efficiency, it costs too much time to collect different kinds of data by iterating the process of searching CSS selectors in the filter list for many times although its accuracy is not low. Although CDPA still contains some limitations, it is still can be judged as an efficient tool to locate cookie dialogs and interactive elements, and to simulate humans to interact with elements to collect cookies. CDPA is listed as one main contribution in this project because both data collection process and measurement process depend on it.

To answer the research questions, this project took advantage of the developed tool to test 1,000 websites. The tool specifically collected interactive elements' type and cookie number after different interactions. Two kinds of cookies were given insights, they are first-party cookies and third-party cookies. These cookies' numbers and changes were mainly focused. Finally, conclusions were given according to different questions. For the question about cookie compliance of websites, in the 958 valid websites, there are around 74.3% websites comply with the regulation of setting cookie dialogs, but there are about 66.2% websites in those 420 websites containing cookie dialogs do not provide enough interactive elements on the cookie dialogs. Among these cookie dialogs with enough interactive options, there are about 35.4% in 142 websites do not set meaningful options. Cookie number is not decreased as expected when users want to decline cookie settings. In addition, in those 420 websites that contain cookie dialogs, there are about 22.4% websites mention the term of 'withdraw' and they are temporarily judged as complying with that regulation. As for the question about consistency of displaying cookie dialogs on the reloaded pages after different interactions, there are only 7.1% in 142 websites meet consistency requirement, they both show cookie dialogs no matter users' choices.

5.2 Recommendations for GDPR and Users

According to the result of project, some recommendations are listed for GDPR to better protect users' data. As mentioned above in the section 5.1, many websites are found not to decrease tracking cookies after users showing 'decline' attitudes although they indeed display cookie dialogs and 'decline' elements to users. One GDPR modification recommendation about cookies is that websites should be required to respect users' options and number of unnecessary cookies should be indeed decreased. Meanwhile, most of websites do not directly show clickable option for users to withdraw consents about cookies and some websites only mention this policy in the 'statement privacy' or on the cookie dialog. To be strict, this does not satisfy the regulation of 'user should withdraw their consents as easy as giving consents' listed in the GDPR [3]. The other recommendation for GDPR modification is about requiring websites to specifically show the option of 'withdraw consents' in the form of clickable elements or other convenient methods. The last recommendation is about whether to show cookie dialogs again after users revisiting pages. If GDPR requires websites to show cookie dialogs every time when users visit them, users may also be confused about it. However, if cookie dialogs are not shown again after users giving consents, it will be difficult for them to find ways to withdraw their consents. As a result, the last recommendation is that GDPR can require websites to set up small buttons in the corner of each page, as shown in the figure 4.10. When users want to review the cookie dialog or withdraw their consents, they can click that button to view the cookie dialog again.

Users always think that their choices are meaningless, so they have habits to click ‘accept’ when they see the cookie dialogs [44]. However, some websites indeed respect their choices, and they will decrease number of third-party cookies when users show the attitude of ‘decline’. As a result, there is one recommendation provided for users that users can spend a little time to take insight of websites’ provided options, and if they care much about their privacy, they can choose to decline the cookie use and their choices are meaningful. Only when users attempt to pay attention to their privacy, their privacy will be better protected.

5.3 Limitation and Future Work

As this project only applied one kind of filter list modified by one community, some commonly used CSS selectors for cookie dialogs are omitted. It causes that some cookie dialogs are left out by CDPA, and accuracy could be increased by integrating with a more completed filter list in the future work. Moreover, this project also found that some websites display ‘decline’ options only when users click ‘manage preference’. This project collected this kind of elements with help of manually checking. Future work can focus on the development of tool’s automatic function of locating this kind of interactive elements shown by the website. This can help to save much time and the collected elements and dataset will be more completed. Meanwhile, this project searched for the option of ’withdraw’ by searching for the term of ’withdraw’ because almost all websites do not show clickable elements of ’withdraw’, it can be improved in the future work by analyzing more accurate term or phrases with n-gram to achieve a more accurate result. This project only took insight of first-party cookie number and third-party cookie number to judge cookie compliance, researched cookie type and content can be extended to a wider range to get a more specific conclusion about cookie compliance of each website. Meanwhile, other kinds of tracking technologies can also be researched to compare with web cookies to find out more about privacy problems.

Bibliography

[1] Adblockplus: Surf the web with no annoying ads. Available online: https://adblockplus.org/.

[2] Chrome devtools protocol. Available online: https://chromedevtools.github.io/devtools-protocol/.

[3] Cookies, the gdpr, and the eprivacy directive. Available online: https://gdpr.eu/cookies.

[4] Easylist. Available online: https://easylist.to/.

[5] googletrans 3.0.0. Available online: https://pypi.org/project/googletrans/.

[6] I don't care about cookies 3.3.1. Available online: https://www.i-dont-care-about-cookies.eu/.

[7] Information on the tranco list with id 52xn. Available online: https://tranco-list.eu/list/52XN/full.

[8] Third-party cookies and firefox tracking protection. Available online: https://support.mozilla.org/en-US/kb/third-party-cookies-firefox-tracking-protection.

[9] What is gdpr, the eu's new data protection law? Available online: https://gdpr.eu/what-is-gdpr/?cn-reloaded=1.

[10] Nataliia Bielova. Web tracking technologies and protection mechanisms. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pages 2607–2609, 2017.

[11] Cristian Bravo-Lillo, Lorrie Cranor, Saranga Komanduri, Stuart Schechter, and Manya Sleeper. Harder to ignore? revisiting pop-up fatigue and approaches to prevent it. In *10th Symposium On Usable Privacy and Security ({SOUPS} 2014)*, pages 105–111, 2014.

[12] Tomasz Bujlow, Valentín Carela-Español, Josep Sole-Pareta, and Pere Barlet-Ros. A survey on web tracking: Mechanisms, implications, and defenses. *Proceedings of the IEEE*, 105(8):1476–1510, 2017.

[13] Aaron Cahn, Scott Alfeld, Paul Barford, and Shanmugavelayutham Muthukrishnan. An empirical study of web cookies. In *Proceedings of the 25th international conference on world wide web*, pages 891–901, 2016.

[14] Caudio Carpineto, Davide Lo Re, and Giovanni Romano. Automatic assessment of website compliance to the european cookie law with coolcheck. In *Proceedings of the 2016 ACM on Workshop on Privacy in the Electronic Society*, pages 135–138, 2016.

[15] Adrian Dabrowski, Georg Merzdovnik, Johanna Ullrich, Gerald Sendera, and Edgar Weippl. Measuring cookies and web privacy in a post-gdpr world. In *International Conference on Passive and Active Network Measurement*, pages 258–270. Springer, 2019.

[16] Brian Dean. Google chrome statistics for 2021. Available online: `https://backlinko.com/chrome-users`.

[17] Martin Degeling, Christine Utz, Christopher Lentzsch, Henry Hosseini, Florian Schaub, and Thorsten Holz. We value your privacy... now take some cookies: Measuring the gdpr's impact on web privacy. *arXiv preprint arXiv:1808.05096*, 2018.

[18] Steven Englehardt, Dillon Reisman, Christian Eubank, Peter Zimmerman, Jonathan Mayer, Arvind Narayanan, and Edward W Felten. Cookies that give you away: The surveillance implications of web tracking. In *Proceedings of the 24th International Conference on World Wide Web*, pages 289–299, 2015.

[19] Michelle Goddard. The eu general data protection regulation (gdpr): European regulation that has a global impact. *International Journal of Market Research*, 59(6):703–705, 2017.

[20] Satish Gojare, Rahul Joshi, and Dhanashree Gaigaware. Analysis and design of selenium webdriver automation testing framework. *Procedia Computer Science*, 50:341–346, 2015.

[21] Aniko Hannak, Gary Soeller, David Lazer, Alan Mislove, and Christo Wilson. Measuring price discrimination and steering on e-commerce web sites. In *Proceedings of the 2014 conference on internet measurement conference*, pages 305–318, 2014.

[22] Xuehui Hu, Nishanth Sastry, and Mainack Mondal. Cccc: Corralling cookies into categories with cookiemonster. In *13th ACM Web Science Conference 2021*, pages 234–242, 2021.

[23] Ashar Javed, Christian Merz, and Joerg Schwenk. Ttpcookie: Flexible third-party cookie management for increasing online privacy. In *2014 IEEE 13th International Conference on Trust, Security and Privacy in Computing and Communications*, pages 37–44. IEEE, 2014.

[24] Balachander Krishnamurthy, Konstantin Naryshkin, and Craig Wills. Privacy leakage vs. protection measures: the growing disconnect. In *Proceedings of the Web*, volume 2, pages 1–10, 2011.

[25] Balachander Krishnamurthy and Craig Wills. Privacy diffusion on the web: a longitudinal perspective. In *Proceedings of the 18th international conference on World wide web*, pages 541–550, 2009.

[26] David M Kristol. Http cookies: Standards, privacy, and politics. *ACM Transactions on Internet Technology (TOIT)*, 1(2):151–198, 2001.

[27] David M Kristol. Http cookies: Standards, privacy, and politics. *ACM Transactions on Internet Technology (TOIT)*, 1(2):151–198, 2001.

[28] Tai-Ching Li, Huy Hang, Michalis Faloutsos, and Petros Efstathopoulos. Trackadvisor: Taking back browsing privacy from third-party trackers. In *International Conference on Passive and Active Network Measurement*, pages 277–289. Springer, 2015.

[29] Célestin Matte, Nataliia Bielova, and Cristiana Santos. Do cookie banners respect my choice?: Measuring legal compliance of banners from iab europe's

transparency and consent framework. In *2020 IEEE Symposium on Security and Privacy (SP)*, pages 791–809. IEEE, 2020.

[30] Jonathan R Mayer and John C Mitchell. Third-party web tracking: Policy and technology. In *2012 IEEE symposium on security and privacy*, pages 413–427. IEEE, 2012.

[31] Barnabas Molnar. Measuring the cookie-setting behaviour of web pages showing privacy warnings.

[32] David Naylor, Alessandro Finamore, Ilias Leontiadis, Yan Grunenberger, Marco Mellia, Maurizio Munafò, Konstantina Papagiannaki, and Peter Steenkiste. The cost of the" s" in https. In *Proceedings of the 10th ACM International on Conference on emerging Networking Experiments and Technologies*, pages 133–140, 2014.

[33] Gaston Pugliese. Web tracking: Overview and applicability in digital investigations. *it-Information Technology*, 57(6):366–375, 2015.

[34] Paruchuri Ramya, Vemuri Sindhura, and P Vidya Sagar. Testing using selenium web driver. In *2017 Second International Conference on Electrical, Computer and Communication Technologies (ICECCT)*, pages 1–7. IEEE, 2017.

[35] Iskander Sanchez-Rola, Matteo Dell'Amico, Platon Kotzias, Davide Balzarotti, Leyla Bilge, Pierre-Antoine Vervier, and Igor Santos. Can i opt out yet? gdpr and the global illusion of cookie control. In *Proceedings of the 2019 ACM Asia conference on computer and communications security*, pages 340–351, 2019.

[36] DL Schacter, DT Gilbert, and DM Wegner. Self esteem. *Psychology, 2nd Edition, Barnes & Noble, Worth Publishers, New York*, 2009.

[37] John Schwartz. Giving the web a memory cost its users privacy. *New York Times*, 4(01), 2001.

[38] Janice C Sipior, Burke T Ward, and Ruben A Mendoza. Online privacy concerns associated with cookies, flash cookies, and web beacons. *Journal of internet commerce*, 10(1):1–16, 2011.

[39] Emil Sit and Kevin Fu. Inside risks: Web cookies: not just a privacy risk. *Communications of the ACM*, 44(9):120, 2001.

[40] Petronyte Smilte. Measure the cookie setting behavior of web pages showing cookie privacy warnings, 2021.

[41] IT Governance Privacy Team. *Eu general data protection regulation (gdpr)–an implementation and compliance guide*. IT Governance Ltd, 2020.

[42] Nguyen Binh Truong, Kai Sun, Gyu Myoung Lee, and Yike Guo. Gdpr-compliant personal data management: A blockchain-based solution. *IEEE Transactions on Information Forensics and Security*, 15:1746–1761, 2019.

[43] Tobias Urban, Dennis Tatang, Martin Degeling, Thorsten Holz, and Norbert Pohlmann. Measuring the impact of the gdpr on data sharing in ad networks. In *Proceedings of the 15th ACM Asia Conference on Computer and Communications Security*, pages 222–235, 2020.

[44] Christine Utz, Martin Degeling, Sascha Fahl, Florian Schaub, and Thorsten Holz. (un) informed consent: Studying gdpr consent notices in the field. In *Proceedings of the 2019 acm sigsac conference on computer and communications security*, pages 973–990, 2019.

[45] René van Bavel, Nuria Rodríguez-Priego, et al. Testing the effect of the cookie banners on behaviour. *JRC Technical Reports*, 2016.

[46] Rob Van Eijk, Hadi Asghari, Philipp Winter, and Arvind 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). IEEE*, 2019.

[47] Jonathan Zittrain and Benjamin Edelman. Internet filtering in china. *IEEE Internet Computing*, 7(2):70–77, 2003.