ISSN 2347 - 3983
Volume 9. No. 2, February 2021
International Journal of Emerging Trends in Engineering Research
Available Online at http://www.warse.org/IJETER/static/pdf/file/ijeter11922021.pdf
https://doi.org/10.30534/ijeter/2021/11922021

# Securing the Web Browser Local Data Storage

**Osama Salim[1] Thamer Al-Rousan[2]**
Faculty of Information Technology, Isra University, Amman, Jordan
[1]Iragian, Jordan,os1995ama@yahoo.com,
[2]Jordanian,Jordan,thamer.rousan@iu.edu.jo

**ABSTRACT**

The security browsers represent the level of trust in system security. This is has led to the urgent need for an encryption process for these personal accounts on the browser page, because intruders are working to steal these accounts, in addition to trying to penetrate accounts, either for material or personal purpose. What has been worked on in this paper is made an extension layer on chrome browser used the user's key encryption. The RSA algorithm was used as it is one of the strong encryption algorithms in the current era. Where a 2048-byte encryption key was used, the aim of increasing is the strength of the encryption.Finally, this paper tested the experiments by applying the model on the experimental application and WhatsApp application.

**Key words:** RSA algorithm; security; local storage

## 1. INTRODUCTION

In last few years, the utilization of the Internet and its services has ceaselessly magnified because it has become the most supply of information people, home, work and education. People spend so much time.

Lately, the numeral of hackers is more and more aiming Web applications. The world is facing a speedy growth in the number of attacks on the Web applications due to given that providers have become greater adept at writing steady code and growing and dispensing patches to counter conventional types of attack (e.g., buffer overflows)[1].

The Hypertext Transfer Protocol (HTTP) cookie was the primary and the most known technique which provides the privilege for users to store information on local storages. Nevertheless, the cookies do not match the particular demands of modern web applications, where a code section of the web application is transferred from the server-side to the user-side. So, new techniques were required for user-side storage. Alongside these advantages, the user requires to be confirmed concerning his information security and privacy [2].

In spite of several efforts and volumes of literature selling such security procedures, vulnerabilities are continuously being determined and exploited. Therefore, the researchers' is paintings on growing answers to deal with those problems. Web application safety may be more suitable via the multiplied applications of steady improvement practices. The browser providers can assist to boom their customers secure relies upon on 3 factors [3]:

- The riskiness of vulnerabilities: the browsers can lessen the cruelty of gaps rely on sandboxing their rendering engine. This is applied via sandboxes to restrict the harm that may be resulting from an intruder who utilizes gaps with inside the rendering engine.
- The window of gaps: the browsers can lessen this window via enhancing the consumer practice for putting in browser updates, hence decreasing the variety of customers working on previous versions that need safety patches.
- The frequency of exposure: the browsers can lessen the frequency wherein customers engage with malicious content material through caution customers earlier than they go to acknowledged malicious sites.

In this paper, lessen the threat in their local storage safe through encrypting the browser.

## 2. BACKGROUND OF PROBLEM

The Local storage of the browser has enormous factors of interest while contrasted and a server-side database, containing brief reaction, disconnected utilization, and dwindle network latency. Nevertheless, this time, local browser storage isn't always secure. The primary security difficulty is that data spared domestically is decoded. The associated affirmed that the decoded data was now no longer through any method the only problem confronting local browser storage.

Nowadays, there are serious privacy implications, along with the information of a user's bank account or credit cards. Despite all of the benefits, the process of client-side storage did not happen without a cost. The data stored through client-side storage is unencrypted, so customers can also additionally enjoy privacy violations, along with a provider.

The goal of this paper is to improve the security and privacy level for the data which is stored in the browser's storage. There are 3 primaries scientific of this study.

1. To investigate security technical connected with local browser storage and popular security resolutions.
2. To attain a security model for browser-based storage.
3. To implement and apply the security model.

## 3. SECURITY OF WEB

The Web is considered one of the fundamental ways for individuals to communicate with devices, so that individuals are associated with the broadening of substance, administrations and applications. Clients can undoubtedly discover fascinating new substance on the Web, but this represents a security challenge: pernicious site administrators can assault clients through their Web programs. Browsers face the challenge of providing a rich stage to web applications while guaranteeing client safety.

Browsers are ideal for assailants since they have an enormous and complex confided in figuring base and a wide scope of organization noticeable interfaces. Generally, every Browser contained weakness eventually that permitted pernicious site administrators to bypass the Browser's security strategies and bargain the client's device. Regardless of whether these weaknesses are fixed, numerous clients keep on running the prior, weak form. At the point when these clients visit malevolent sites, they hazard putting their devices in danger[4].

Web application security weaknesses usually stem from programming errors using web application programming languages (such as Java, .NET, PHP, Python, Perl, and Ruby), code bases, design patterns, or restructure. These weaknesses can be complex and can occur in many situations. Using a web application firewall may control the impact of certain weaknesses, but cannot resolve potential weaknesses[4].

### 3.1 Types of Authentication Mechanisms of Web Security

In this section, explain the mechanisms approved in web security as follows:

1- **User authentication.** The fundamental substance of authentication in the user see is the secret phrase, which might be the principal opportunities for an assailant to enter the organization. The preventive measures or measures for any gadget to diminish network weaknesses are to produce or make solid passwords and solid validation [5].

2- **Biometrics authentication.** Biometrics innovation utilizes individuals' characteristic behavior and hereditary attributes, (for example, hands, fingerprints, eyes, retina, hands, and so forth) to give clients strong identity verification for user identity verification [6].

3- **Token based authentication.**A token is an authentication mechanism that gives an encoded token to prove the identity of the client of the authentication worker to get entrance. The client needs to get to enter the confided covertly key between the authentication server and the application. Yet, this mechanism is not the same as biometric gadgets. A genuine illustration of a token is a smart card [5].

4- **Out of band.** Out-of-band authentication is utilized to help two communication methods utilizing two authentication methods (factors). There is a regular flow of authentication messages between the computer and the server, and out-of-band utilize implies that some of them utilize other communication techniques it may be mobile communications. Part of the information is sent by the network, and the other part is sent by the mobile device [7].

5- **Certificates.** There are numerous approaches to pick up trust in people. These can be given by voice, face or penmanship. This is simple for individuals who know it previously. For the rest, it needs more innovation to trust. Believing the other party by posing explicit inquiries requires singular upgrades to each actualized innovation. This is designated "Trust Threshold"[6].

6- **Public Key Infrastructure (PKI) structure.** PKI is utilized as a confided in correspondence among e-commerce contacts on the Internet as a trusted third party. PKI gives identification and access control administrations[7].

7- **Network authentication.** It gives validation archives dependent on the confirmation and approval credits of Web clients, including portrayals of authentication occasions for Web clients between the application and the endeavor security framework [5].

## 4. WEB STORAGE

Web storage is a term that permits web applications to build determined key-value storage in the browser, and the content is saved till the end of the session (session storage) or the end (local storage). Compared to HTTP cookies, Web applications can save large amounts of data using this technology. Depending on the browser, the storage provides storage capacity ranging from 5MB to 25MB.
The term suggests that browser merchants handle web storage content in the identical method as HTTP cookies. Especially, manufacturers are urged to design client interfaces to clear data so as to allow clients to remove all various kinds of persistent data at the same time. Although Web Storage is less well known than HTTP cookies, its use is not subject to regulations regarding personal client data information [8].

### 3-1 Session Storage

Session storage used in a similar way to session cookies, but with some performance improvements. Since the data in the user session storage area is stored through the window rather than through the browser, data "leakage" can be prevented. For example, if a user has opened two windows for the same shopping cart transaction, a purchase on one window may be processed in two windows because cookies are shared between the two windows [8].

### 3-2 Local Storage

Local storage used in the same way you use persistent cookies, but like session storage, they have some obvious advantages over using cookies. First of all, the storage capacity is huge. The "almost arbitrary limit" of each source is 5MB, and the domain can store a large amount of key-value data; the realization is rich, and the data limit can be

flexibly performed according to the user agent. However, if you store sensitive data, it is best to encrypt the data because it is transparent to the user. Because any user who has access to the browser can see the web storage data, this may bring security risks[9].

Limitations of local storage: Let us take a daily transaction mobile website as an example to understand the limitations. In the daily transaction website/application, the seller uploads a discounted product with a coupon code; the user can purchase the card code from the regular transaction website and receive a large commission on the goods. On the regular transaction website, clients can store personal transactions in their wallets for later review. For example, to store some transactions in "My Wallet", it is best to store such knowledge on the client's local system. Local storage can be used, but there are some restrictions because local storage cannot directly access objects from server-side languages. In addition, local storage is not very good for storing data for various clients. In local storage, data can only be stored in key-value pairs. Since it does not have any formal arrangement (table structure), it is hard to save structured data of various clients. Therefore, the following limitations were found in local storage [10]:

- The local storage cannot be accessed directly from the server.
- It does not have any regular structure (table structure).
- It does not support concurrent environments.

### 3-3 Web Storage security

Storing data with the client instead of the server also has inherent advantages. One is that security vulnerabilities in web applications do not necessarily mean that user data is threatened. Since the data is saved with the customer, the unique spot of arriving at the data is within the customer's local computer. In addition, although cookies need to be continuously sent back to the server to maintain the correct state, local storage is not required. Client-side JavaScript can be used to continuously change the user's key/value pairs without the need to communicate with the server. As the need for continuous HTTP requests is reduced, the possibility of packet sniffing attacks is reduced [10].

Privacy: Due to privacy issues related to the use of persistent cookies and session cookies, the Web Storage specification provides a client-side alternative. Both session storage and local storage attributes can be used as alternatives to cookies to provide users with control over their data [8].

Performance and accessibility: The Web Storage Specification also focuses on improving the performance of Web applications. Web applications running mainly on the client instead of the server can greatly reduce the workload of the server, thereby reducing the need for expensive server equipment [8].

## 4. RELATED WORK

Previous studies with respect to making sure about web browser storage and its adequacy are as yet in the beginning phases and are restricted. [11]Studied portable browsers to decide whether information actually existed after the browser meeting halted. They examined the RAM in the browser meetings, including authorizations, history, pictures, and recordings. [12]Conducted one of the principal studies to analyze browsing security weaknesses. The outcomes uncovered that there was a lacking execution of the security system in various internet browsers, which highlighted client exercises. Furthermore, security control for Firefox was proposed, which ensured clients after private mode was empowered. [13] Investigated the log records produced by an internet browser, focusing on inquiry history, timetable examination, and URL encoding. A Classification of Web Browser Log (CWB) instrument was proposed to demonstrate the investigation. Shockingly, in the analyses, old adaptations of browsers were utilized that are as of now obsolete.

[14]The authors introduced a survey of the exploration done on the browser augmentation weaknesses. Contrasted with other web security issues, the quantity of distributions for augmentation security is low in number. The majority of the exploration works utilized program investigation strategies. They established that Artificial Indulgent (AI) procedures were not really utilized. One of the conceivable future exploration bearings in this field could be to attempt distinctive AI methods to distinguish noxious/weak augmentations. Additionally, the instruments to test the security of the given augmentation are not promptly accessible for download. [15] Broadened the work by analyzing the document framework and the organization, which uncovered huge inconsistencies in the browsing execution that abused a customer's security. They built up a security access control to ensure private information that could be gotten to and utilized by assailants, zeroing in on JavaScript expansions in Chrome.

[16] Studied JavaScript expansions in Firefox from alternate points of view, for example, social, safety, and debugging, to determine which might be malignant. They focused on recuperation methods made during perusing. Experiments within four individual levels indicated how the program could be halted, in particular closure, shut down, freeze, and execute measures. The results revealed that all levels included client protection infringement regarding acquiring perusing information. [17]

The authors proposed system provides a suitable certification model for the Iraqi academic supervision system. It provides user access control, well-designed authentication mechanism and high security to save user data. It is also necessary to increase the information security knowledge system of appropriate authentication control to reduce the threat of impression in the application. Therefore, it provides appropriate authentication controls to reduce simulated threats in the system and verify the correct level of authentication strength. [18] Conducted a survey that focused on mindfulness from a client's perspective. The authors studied in excess of 200 clients with respect to the security and protection instruments gave by the most well-known internet browsers on cell phone and work area stages. The

outcomes uncovered that better security ensures were required concerning a client's protection.[19], [12]the author applied a security model for local browser storage. The model added an extra layer among local browser storage and the internet browser itself. It contained an algorithmic structure for improving protection from weaknesses recognized in this study.

Javascript Crypto Library (JSCL) was utilized in the proposed calculation and was applied to the internet browser as an expansion. The Stanford JavaScript Crypto Library (SJCL) library was picked in light of the fact that SJCL is a safe, little, quick, multi-stage and ground-breaking library for cryptography in JavaScript.

The examination consolidated the Rivest–Shamir–Adleman (RSA) standard with a SJCL library, which brought about a hybrid encryption algorithm to guarantee data integrity.[20], [21]reviewed the properties, limitations, and related tools of web browsers in general, private, and portable browsing modes. In the private browsing session of the browser, the recovered artifacts are not as important as the public browsing session, which verifies several claims made by the creators of these programs. The TOR browser bundle does a great job of minimizing the amount of information. Therefore, effective methods and appropriate tools are required so that attacks can be easily noticed and changes in memory content can be minimized.

## 5. PROPOSED MODEL

In this problem, the proposed extends version of local storage and name (My extension). My extension has a proposed model of encryption and decryption and it has the ability to synchronize with the webserver. It is added functionality and implementation in chrome.

The proposed model includes an algorithmic which was made to enhance the security against threats. The RSA encryption was implemented to the web browser as an extension while the model utilized it. The mentioned extension was stored on the top of the web browser storage, so that whenever reading or writing information, it will be encrypted, as the next algorithm.

**The step of algorithm:**

1. The Encrypt Web: Users register on the website and have the username and password.
2. The Extension on his browser “chrome: User installs the add-on and login using his username and password.
3. The User navigate to any website “ex: abc.com”: the user then should approve that the add-on encrypts data (used RSA algorithm (2048 byte key size)) fetched through the website and stored in the local storage.
4. The add-on will communicate with encrypt web and generate 2 pair of keys and encrypt/decrypt the data for abc.com before storing them and decrypt before restoring to use.
5. Once the uses close abc.com, the add-on will encrypt the abc.com data and remove any no encrypted data.
6. Each login user account with press encrypt update his secret key on encrypt web to make sure his password didn’t compromise.

**Creating User Algorithm**

This part used to create a new account for accessing to extension layer that applied encrypts / decrypts process, as the next algorithm.

**The step of algorithm:**

**Step1:** Start algorithm

**Step2:** Register the (first name) of the user

**Step3:** Register the (last name) of the user

**Step4:** Register the (Email) of the user

**Step5:** Register the (password) of the user

**Step6:** press on the (Register button) to save information in the user database

**Step7:** End algorithm

**Apply model Algorithm**

In this part present the algorithm to apply model, as the following steps:

**Step1:** Start model

**Step2:** The user opens the browser page on which to search the Internet

**Step3:** The user presses the (extension icon) at the top of the browser page

**Step4:** The applied model shows the user three options, including (activation, encryption, and decryption)

**Step5:** If the user is running the proposed program for the first time since entering the browser, he will click on (the activation icon) to run the program

**Step6:** If the user wants to encrypt the data that he is searching for hiding it from intruders, he will press the (encryption icon) to encrypt all the data that the user is searching for it

**Step7**: If the user wants to decrypt the data that he encrypted in the previous step, he will press (the decryption icon) to return the data to its original form before encryption

**Step8**: End model.

At the final the part, in this chapter presented the proposed model and applying model. In the next chapter explained the implementation model with the result.

### 5.1 Methodology

The goal is to present a security model to discuss problems produced by storing data in a dangerous method. In the proposed model which was mentioned before, an additional layer was added between the web browser itself and the web browser storage.

The model includes a new algorithm enhancing security upon attacks. The model utilizes the RSA encryption library, which was implemented to the web browser as an extension. The extension was established on the head of the web

browser storage so that whenever writing or reading a piece of information, it is encrypted.

In this paper, extension layer has been added on Google Chrome. When entering any web page, this Extension Layer is activated, in addition to the encryption process for all types of keys. In each login to the Extension Layer then press the encrypt process started dynamically for encrypting the keys, not performing one-time encryption and storing it, and this increases the strength of the proposed model, as shown in the figure (1).

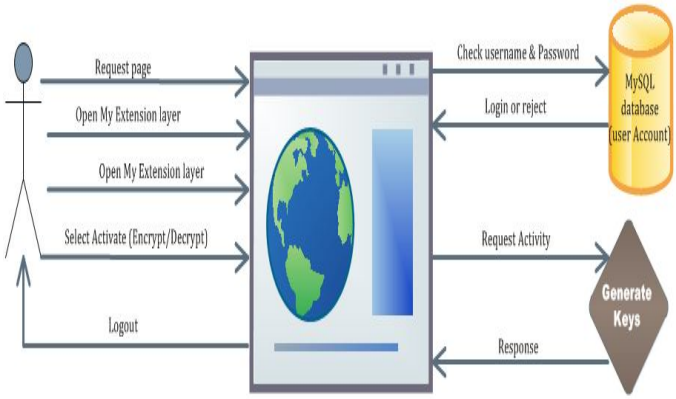


**Figure 1:** Procedure model

## 5.2 Testing proposed model

In this part tested the proposed model, he following page declares the number of keys can be added to test the system, and declares the keys that used in encryption for this page can also be deleted. The following example shows the addition and deletion and also making sure that the key is correct if it is displayed more than once before encryption, as shown in the figure (2).

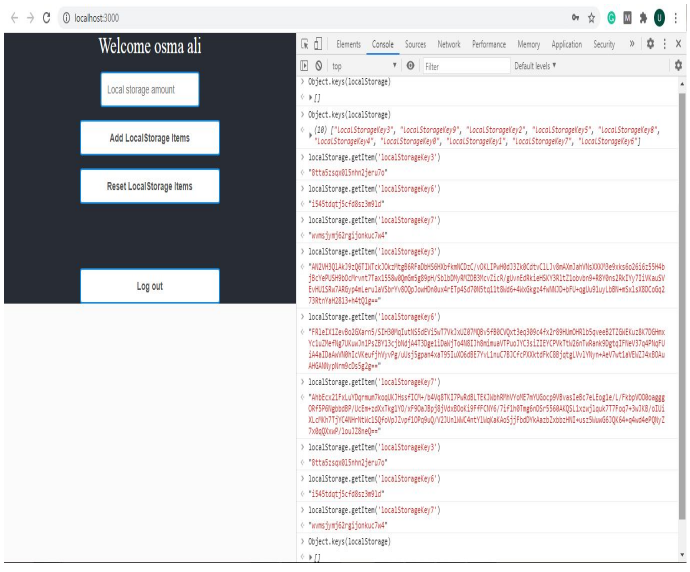


**Figure 2:** Example to apply model in testing program

## 5.3 Applied the proposed model

The main program is an interface before it is starting to add sites with entering them to perform encryption operations, as shown in Figure (3).

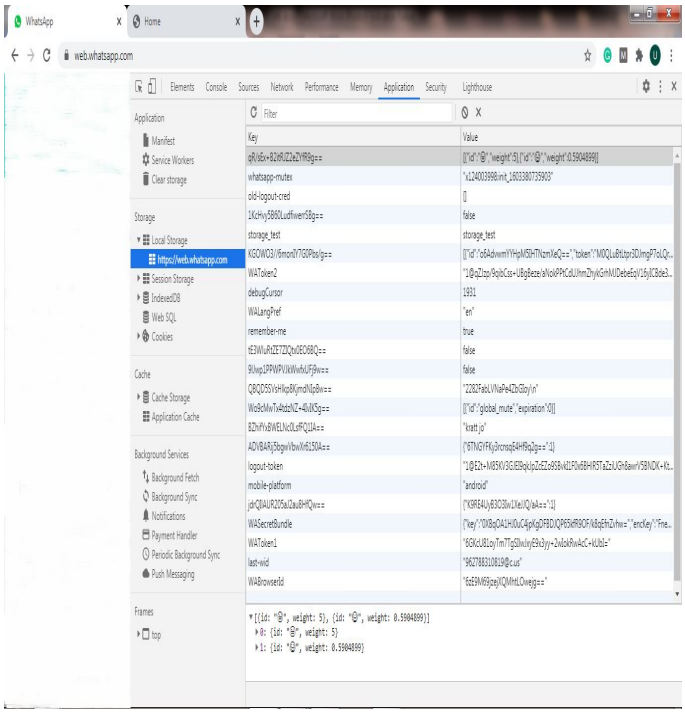

**Figure 3:**Whatsapp key in local storage

The following figure (4) shows the interface (Extension layer) after entering and adding to perform operations (Encryption/decryption).

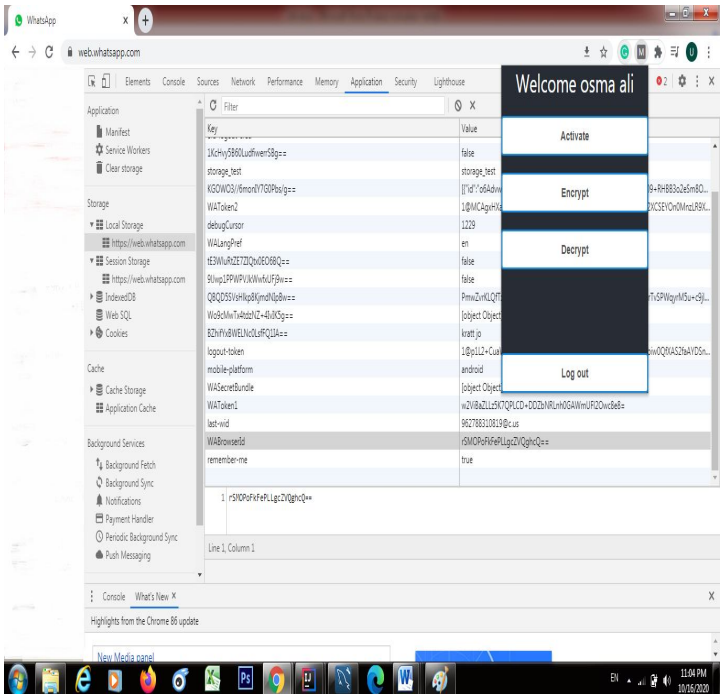


**Figure 4**: Open My extension layer home page

The following figure (5) shows the main interface after performing the encryption process.

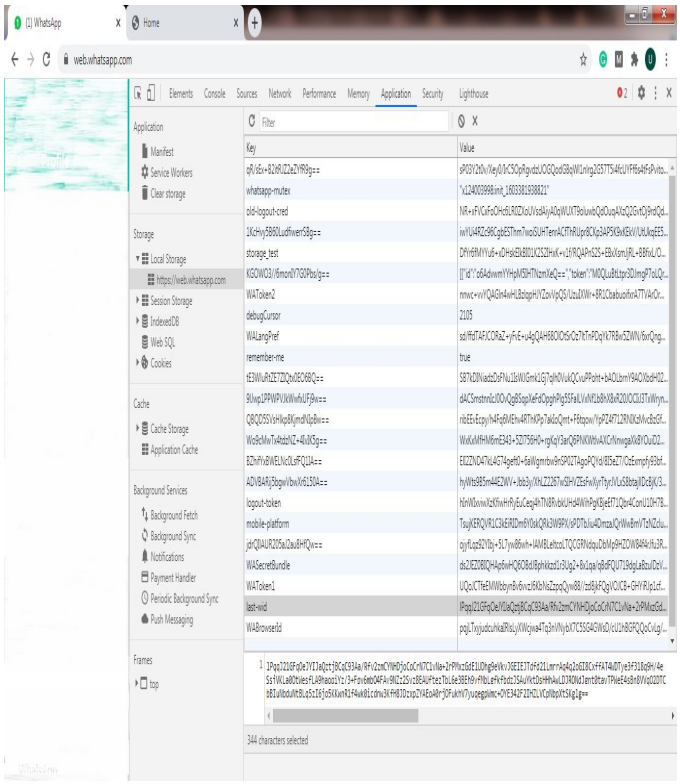

**Figure 5:** Encrypt keys for Whatsapp

The following figure (6) shows the number of keys present in one of the stored sites, choosing specific models and performing operations on them in terms of encryption and decryption process.

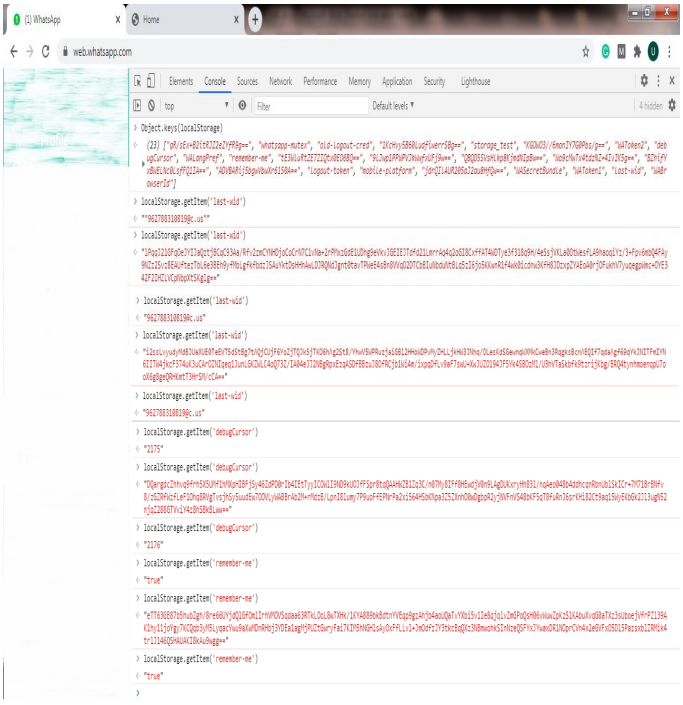

**Figure 6:** Shown Encrypt/ Decrypt by inspect Browser

### 5.4 Calculation time and Encryption and Decryption processes

The following figure (7) shows the schematic diagram of time the encrypting and decrypting 10 keys.

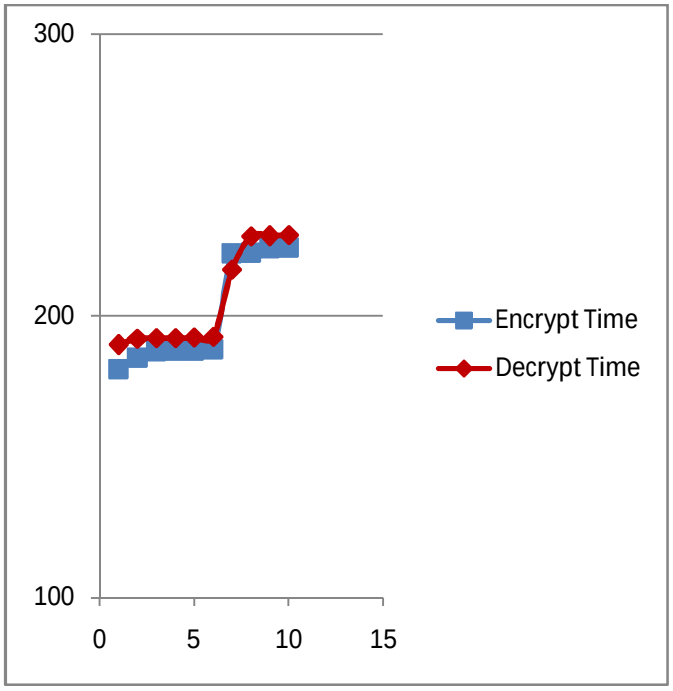


**Figure 7:** Time chart for encrypt/decrypt for 10 keys.

The following figure (8) shows the time diagram for the encrypting and decrypting 100 Keys.

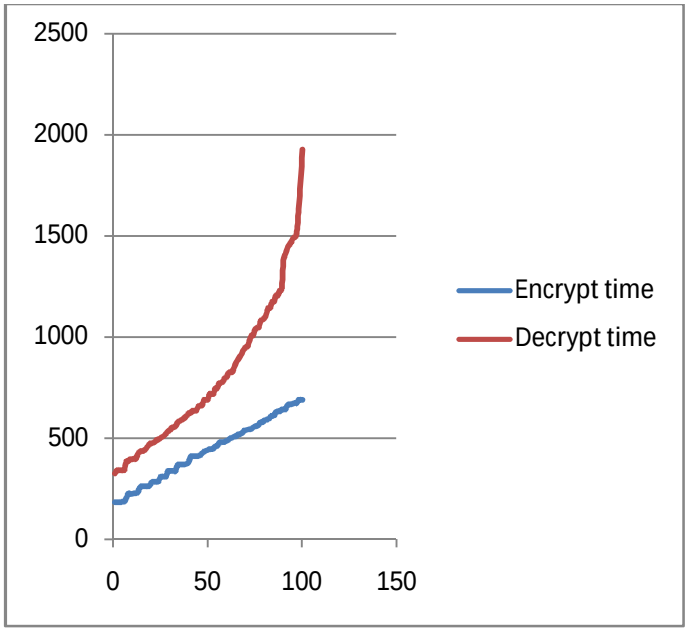


**Figure 8:** Timing for encrypt/decrypt process for 100 keys

The following figure (9) shows the diagram of the time taken for the encrypting and decrypting process in Whatsapp application.

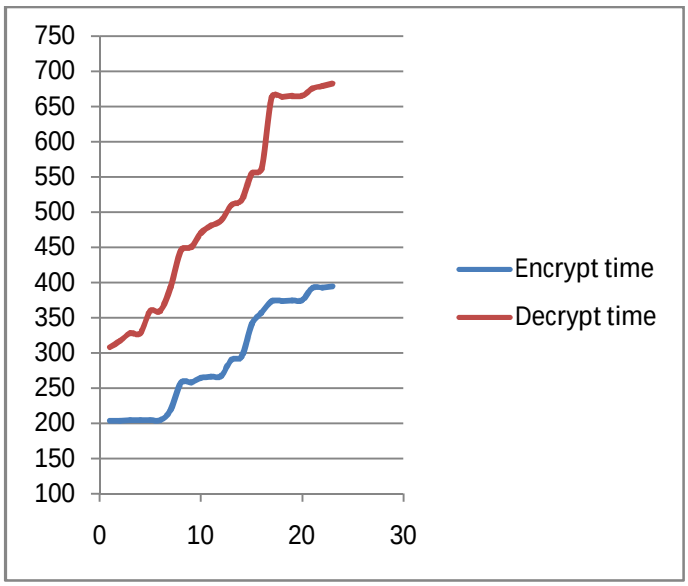


**Figure 9:**Time chart for encrypt/ decrypt for Whatsapp

Finally, the following was observed after implementing these experiments, where notes that several experiments were performed for the encryption algorithm and the experiments were executed on a user's page. Additional to note that time the process of encoding and re-encoding the keys may increase as the key size increases, and decrease as the key size decreases. Generates two keys from each key (public key, private key) stored inside the server. The benefit of this is that the code sent between the user and the server is not known by the intruder in the event that the connection could be compromised.

It was possible to take a larger key, but since the encryption process is related to time, a key of size 2048 was used due to the time required for encryption. At the same time, the researcher worked to store all the keys that are generated dynamically and not statically inside the server, and this storage process was for several purposes, which are: First: The server on the Internet is more efficient than the users' devices, and if the intruder attacks on the connection path, it can only benefit from the public key that is sent between the server and the user. In addition to the fact that the server works on changing these keys in a dynamic and continuous manner, and they are not stored statically. In every login, to the browser, the user's private keys are re-encrypted. Second: The time factor for storage inside the server is used because the encryption process inside the server and the server being more efficient than the device is faster.

## 6. CONCLUSION AND FUTURE WORK

This paper tests the experiments by applying the model on the WhatsApp program. The model encrypted all the keys and stored them with the server, and all these keys are encrypted according to the proposed model. It takes advantage of encryption inside the server to take advantage of time; in addition to that the server encrypts the keys in every Extension Layer entry. An Extension Layer has been added to a specific browser, in this thesis, it has been added on Google Chrome. When entering any web page, this Extension Layer is activated, in addition to the encryption process for all types of keys. In each entry, the Extension Layer performs the process of dynamically encrypting the keys, not performing one-time encryption and storing it and this increases the strength of the proposed model.

After what information has been provided in this thesis, the author suggests the idea for future work to produce more trust between the client and web application, this idea is reviewed in:-

1. The proposed model can be applied to browsers in mobile phones, due to the large number of phones used at the present time, especially in the business world.
2. Apply an algorithm other than (RSA) and test the results on it and compare it with the proposed algorithm in this thesis.
3. Suggest procedure to secure account in database.

## 7. ACKNOWLEDGEMENTS

I would like to thank the School of Information Technology, Isra University, Jordan, for providing a conducive environment during the course of my research.

## REFERENCE

1. S. Divya,andS.Malathi.**Preventing web Application to avoid Illegal Entry of Hackers-a Review**. In 2018 3rd International Conference on Communication and Electronics Systems (ICCES) (pp. 183-187). IEEE. 2018.
2. V. Bajpai, andJ. Schönwälder. A **survey on internet performance measurement platforms and related standardization efforts**. *IEEE Communications Surveys and Tutorials*, vol. 17, no3, 1313-1341. 2015.
3. T. Al-Rousan, S.Sulaimin, S. Rosalina.**A risk identification architecture pattern based on bayesian network.***International Symposium on Information Technology*.IEEE, 2008
4. T. Bujlow, V. Carela-Español, J. Sole-Pareta, andP. Barlet-Ros. **A survey on web tracking: Mechanisms, implications, and defenses.** *IEEE Transaction*, vol. 105, no. 8, 1476-1510. 2017.
5. M. Bugliesi, S. Calzavara, and F. Focardi. **Formal methods for web security**. *Journal of Logical and Algebraic Methods in Programming*, vo. 87, 110-126. 2017
6. J. Bonneau, C.Herley, P. Van Oorschot,andF.Stajano. **Passwords and the evolution of imperfect authentication**. *Communications of the ACM*, vol. 58, no. 7, 78-87. 2015
7. P. Ruiu, G.Caragnano, L. Masala, andE. Grosso. **Accessing cloud services through biometrics authentication**. *In 2016 10th International Conference on Complex, Intelligent, and Software Intensive Systems (CISIS)*, (pp. 38-43). IEEE. 2016.
8. T. Al-Rousan, S.Sulaimin, S. Rosalina.**Supporting architectural design decision through risk identification architecturepattern (RIAP) model.**

*WSEAS transactions on information science and applications*, vol. 6, no. 4, 2009.

9. K. Li, P. Liu, Q. Tan, andX. Wang. **Out-of-band discovery and evaluation for tor hidden services**. *In Proceedings of the 31st Annual ACM Symposium on Applied Computing* (pp. 2057-2062). 2016.
10. J. Ni. **Web based security system**. *U.S. Patent No. 10,694,149. Washington, DC: U.S.* Patent and Trademark Office. 2020.
11. A. Nochimowski, M. Rave, I.Pomerantz,andE. Mardiks. **Data usage profiling by local storage device**. *U.S. Patent No. 10,289,349. Washington, DC:* U.S. Patent and Trademark Office. 2019.
12. T. Al-Rousan. **An Investigation of User Privacy and Data Protection on User-Side Storage**. *International Journal of Online and Biomedical Engineering (iJOE)*, vol. 15, no. 09, 17-30. 2019.
13. D. Ohana, andN. Shashidhar. **Do private and portable web browsers leave incriminating evidence?A forensic analysis of residual artifacts from private and portable web browsing sessions**. *EURASIP Journal on Information Security*, vol. 13, no. 1, 2013.
14. T. Al-Rousan, and H. Abualese**. Impact of cloud computing on educational institutions: A case study.***Recent Patents on Computer Science*, vol.8, no. 2.2015.
15. C. Aggarwal, Y. Li, S. Yu, andR. Jin.**On dense pattern mining in graph streams**. *VLDB Endowment*, vol. 3, no.2, 975-984. 2014.
16. J. Oh, N. Son, and S. Lee. **A Study for Classification of Web Browser Log and Timeline Visualization**. *Lecture Notes in Computer Science*. 2016.
17. J. Arunagiri, S. Rakhi, andK. Jevitha. **A systematic review of security measures for web browser extension vulnerabilities**. *In Proceedings of the International Conference on Soft Computing Systems (pp. 99-112)*. Springer, New Delhi.2016.
18. T. Al-Rousan, S.Sulaimin, S. Rosalina.**Risk analysis and Web project management.***Journal of software*, vol. 4, no. 6, 2009.
19. K. Satvat, M. Forshaw, F.Hao, and E. Toreini. **On the privacy of private browsing –A forensic approach**. *Journal of Information Securityand Applications,* vol. 19, no. 1, pp. 88-100. 2017.
20. R. Ruiz, F.Amatte, and K. Park. **Overconfidence: Personal Behaviors Regarding Privacy that Allows the Leakage of Information in Private Browsing Mode**. *International Journal of Cyber-Security and Digital Forensics*, vol. 4, no. 36, pp. 104-416. 2018.
21. A. Hadi, S. Shaker, andA. El-Ameer. **Design and Implementation of authentication model for the Iraqi AMS Web-based Management System**.*International Journal of Scientific and Engineering Research (IJSER,)*vol. 9, no. 7, 2018.
22. X. Gao, Y. Yang, H. Fu, and J. Lindqvist. **Private Browsing: an Inquiry on Usability and Privacy Protection**, *in Proceedings of the 15th Conference on Privacy in the Electronic Society*, pp. 48-56. 2020.
23. T. Al-Rousan, H.Abualese, andB. Al-Shargabi. **A New Security Model for Web Browser Local Storag**e. *International Journal of Advanced Computer Science and Applications (IJACSA)*, vol. 10, no. 8, 2019.
24. A. Rasool,andZ. Jalil. **A review of web browser forensic analysis tools and techniques**. *Researchpedia Journal of Computing*. Vol. 8, no. 5, 2020.