Privacy Preserving Over Encrypted Data And Urls Detection

1Dhivya V, 2M.Goudhaman
1Master of Engineering, 2Assistant Professor
Jeppiaar Engineering College

Abstract - **Detect Malicious or Fake URLs which consider as social engineering sites (phishing and deceptive sites) that host malware or unwanted software. Safe Browsing(SB) is an important feature in browser world to prevent user from unsafe URL's /malicious. Provide secure encryption method to encrypt the user searching details and URL's before stored on the server. User's browsing data are used to extract valuable information about users interest. These data are under the risk of being exposed to third parties and user browsing might be unauthorized leakage to service provided. Proposed a model which encrypts the user's search data in local history storage and same details stored in server as hashing(SHA512) .Safe browser actually its checked URL as it associated hashes or hash prefix, never leave browser as empty text. Prevents privacy of data from both outside analysts and the intermediate server. It also supports unsafe URL detection, to prevent users from accessing malicious URL. Third party blacklist provider who provide the unsafe URL's with guarantee of privacy and act as blacklist provider. AES algorithm is used for encrypting and decrypting user's browsing data. User history will be stored in local as encrypted format and kept within our platform, if Authorized user want to see history details need to request the Admin(server side) to provided key and then you will receive the secure key to your registered Mobile number or Email to decrypt the stored local history details. Encrypted user history provide more Security and prevent from unauthorized user(Hacker).¬¬¬**

keywords - **Privacy preserving, safe browsing, Encrypted data, hashing, web browser,malware, phishing.**

I. INTRODUCTION

In Real world ,Users are more interested to search to get interesting data through browser and also all users are using them personal accounts such as banking, work related thing, here the change to leakage of our information to third party. we proposed model of SAFE BROWSING to prevent valuable data from phishing. Used Encryption to save our user local history(cipher text) as secret code that's help the hide true meaning. Safe browsing highly helping in real time user to use freely and protect from malicious. In our model, User browsing local history are stored as Encrypted(cipher text) no one can easily see your history and other personal details all could be stored as Cipher text. If authorized user want to see them own browsing data ,they should initiate request to server to get OTP or else Keyword through registered email or phone number. After provided valuable key to server then user encrypted data enable to decryption(plain text).Also we are using as detect Unsafe URLs, third part blacklist who provide the unsafe URLs with guarantee from malicious. This enabled feature prevent user from unsafe URLs and not allow user to get through into further. if suppose user feel like some of information relevant keywords or URLs are unsafe they can send feedback to server to move those into unsafe. we have blacklist to store keywords to prevent user from unsafe words.

II. LITRATURE SURVEY

In this paper[1],The user will be able to access the encrypted data by the CP and generate encrypted queries for secure NDD with her own key. Stage Two - Secure Detection: In order to locate near duplicate data items from the encrypted in-network storage, the user needs to generate an encrypted query tq from the interested data with her own key and send it to the nearest IS. If there exists at least one authorization token Δ, tq would be transformed into the form that can be tested with the ciphertext metadata {c} prepared by the corresponding CP. Propose the fingerprint techniques and locality-sensitive hashing to convert the problem of NDD into the keyword search. We then adopt an efficient multi-key searchable encryption scheme, which requires only one encrypted query from the user even the data are from multiple content providers encrypted with different keys.eg.AES

In[2], To detect and remove malware in app stores, phone vendors could cooperate with SPs and leverage their malware signatures. However, a dilemma is that vendors do not want to share the apps and SPs do not want to share the malware signatures. In this section, we will demonstrate that two commonly used static detection techniques could be used in a privacy preserving way. The first detection technique is inspired by Droid Ranger. To illustrate, we detect malware using requested permissions and the extracted semantic footprints, which are semantic based facts about each app. Second, we compare the similarity of the apps to be scanned with malware samples also on the web.It detects malicious apps in app stores and on users' phones without sharing apps, app runtime behavior,or malware signatures with 3rd parties.

In[3], Consider a system with n users, each of which owns a private data pair (xi , yi) derived from different modalities such as images and texts. We assume that all the data have been locally pre-processed and are represented with

feature vectors. The server, on the other hands, is logically split into two parts: the trusted SGX environment (i.e., the isolated enclave), that uses private information, and the untrusted environment (e.g., operating system, hypervisor), does not. The encrypted images and texts are stored in the untrusted memory outside the enclave. Our security design goal is that the private information about each user's data should not be revealed to other parties, including other users and untrusted environment of the remote server.

In[4], In our system design, the crypto service provider CSP is responsible for initializing the system, i.e., generating and issuing encryption/decryption keys for all other parties. At certain points, the remote server is also required to run secure interactive protocols during the training process. The word vectors learned through our privacy-preserving methods should reach the same accuracy level as that obtained in the plaintext domain. The computation and communication costs at both the remote server side and the crypto service provider side should be practically acceptable.

In[5], Membership test is a query with an outcome of True or False, determining whether an item is in a given set or not. A server with such a set of values and a client one wants to query this set for a specific value are all engaged in the process. Users can perform membership tests without revealing their search values to database holders using Private Membership Test (PMT) protocols.Example is measure the value of PMT: A database is stored on computer. The contents of the file may be sensitive of prevensing. However, the distribution of the query set X is disclosed. To improve the security strength, the second scheme maintains a local store at the client to cache the intermediate results.

III. PROBLEM STATEMENT

In Safe browsing as encryption format stored in local and server SHA512(Hashing) we are overcome below

- When the server receives countless prefixes for a URL, it fails to define metadata. Metadata can results in the leakage of unexpected details.
- To find a matching hash, the attacker can always try adding random values to common passwords.
- One-way functionality that converts plain text into a specific message digests
- The complexity of URL re-identification can be minimized by using multiple prefix matching.

IV. EXISTING SYSTEM

The service providers or adversaries could directly obtain users' browsing history, or indirectly infer the visited URLs by leveraging the shared information. In server there is blacklist to store all the information browsed by user and stored in Hash value formats to prevent data and secure from phishing.

Hashing, however, is a one-way function that plain text to produce a unique message digest(##%$66%^&**).There is no way to reverse the hashing process to reveal the original and can't see whatever hashed details stored and don't know related to which type and also couldn't retrieve datas from server.The blacklist provider provide extra metadata corresponding to each unsafe URL.It helps distinguish between threat types, e.g., malware or phishing.Here Blacklist has some list of unsafe Urls proceed next level with some advanced feature and allow user to get some unwanted and restricted information with certain terms and conditions local setups. Metadata usually contains Hash generation process.

V. PROPOSED SYSTEM

The major detection process is checking the URL to be visited by a user with the records in an encrypted blacklist. Encryption is a two-way function what is encrypted can be decrypted with the proper key.

Encryption data is stored or transferred, making unique solution no matter how data is being used. Usually, data is most vulnerable to attack when being moved from one place to another, therefore encryption ensures protection during this process. Can't edit anything from receiver side only have a read access. provided key we can use to view the contents and secured data for any verification.

Support Vector Machine: Used for Classifying safe and unsafe URLs. Once a match is found (i.e., the URL is unsafe), the corresponding web page will not be loaded. Users will be warned with the threat information for further action. Users' search URL data are encrypted using AES encryption in local and then stored as hashing(SHA512) in server. This also supports keywords based malicious detection process. When user find unsafe URL during search, they will suggest that URL to admin. we proposed encryption to stored local history as encrypted format in both local and server .User will permit to view their history using OTP verification process with proper authentication using Registered phone number or Email of owner.

if you enter wrong key ,user will get notification like someone crack your Account.

Example: User using banking transaction in browser user will get notification "would you like save your password" in respective site. If we saved in local anyone can see your details and steal them. And still banking system directly allow to login with saved details in local in there only you need to enter provided captcha for verify you are Robot or human. you can enter showing captcha then site will allow to enter dashboard and start transaction .so we stored all details in encrypted format only respective user can see details with proper key to prevent sensitive data's and ensure our date will not leak or steal third party.

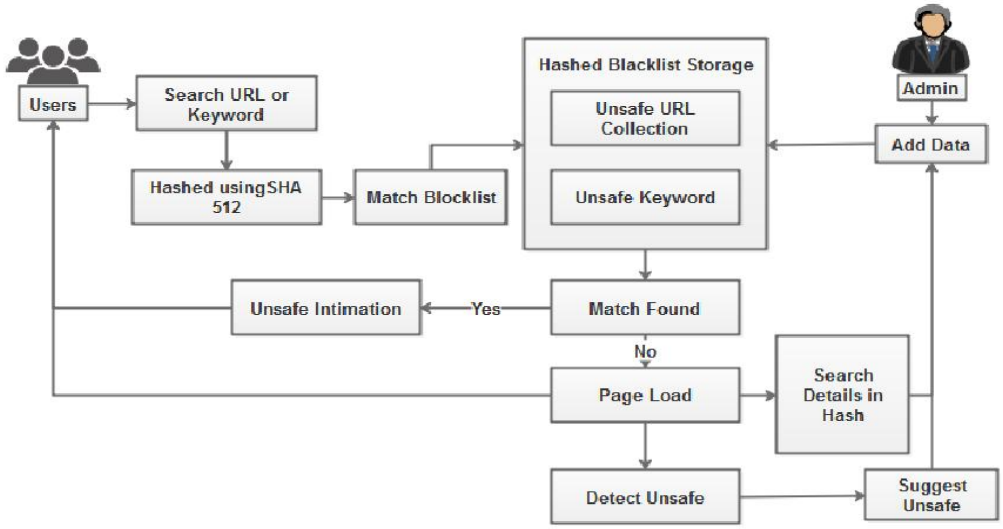
ADVANTAGES

- Urls should blocked with proper notification and doesn't proceed next step.
- Using AES encryption makes sure that this model is strong secured.
- Encryption is a two-way function we can retrieve data with proper key.
- Encrypted User local history ensure and prevent your personal details like banking and any private sites
- There is no clue for the server or malicious user to predict the users' online usage of websites.

- DataBase in server all will be stored as Hashing it's a oneway function so no one can retrieve the details.
- Prevent users from accessing malicious website

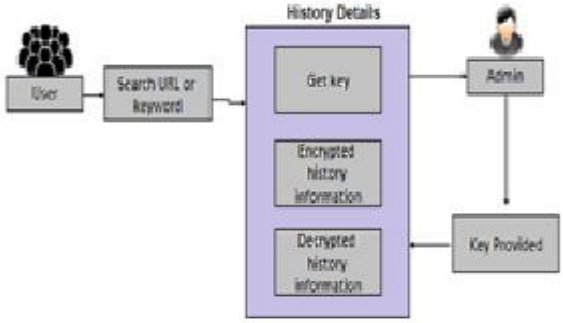
VI. ARCHITECTURE

User browsing data as user interest in safe browsing all the search URLs or Keywords are will be stored in Blacklist storage .The third party black list provide unsafe URLs who can prevent details from malicious with guarantee of preserving data's. In Blacklist all the information are stored as encryption format in cipher text ,there if any of them match with user browsing details it will intimate user stating user are using unsafe URLs we are not allow you to further. if not server allow user to get more information as interest and also user can send feedback as well to server like user can feel some of information as unsafe. In Server side review the feedback and act as their interest .



LOCAL HISTORY:

User searching URLs or data's every information are stored as encryption format in user local. if authorized user want to see as plain text, user need to initiate request to server for security key ot OTP through registered mobiles or Emails. After provided valuable information to key field and then field validation success user will permit to see their history in Decryption as plain text.



VII.ALGORITHM

The block cypher is another name for the AES cypher. On AES, no successful attacks have been recorded. AES has a range of advantages, including being simple to implement on 8-bit architecture processors and being efficient on 32-bit architecture processors. Furthermore, all operations are basic (e.g., XOR, substitution and permutation). Multiple rounds of AES encryption are used. There are four key steps in each round including sub-byte, shift row, mix column and add round key.

Algorithm Procedure

The algorithm starts with a main stage called Add, which is followed by nine rounds of four stages and a tenth round of three stages. This applies for both encryption and decryption, with the exception that of round's decryption algorithm is the inverse of its encryption algorithm counterpart.The four stages are as follows:

1. Substitute bytes
2. Shift rows
3. Mix Columns
4. Add Round Key

The tenth round will leave as Mix Columns stage. The first nine rounds algorithm of ecryption consist of the following:

1. Inverse Shift rows
2. Inverse Substitute bytes
3. Inverse Add Round Key
4. Inverse Mix Columns

Again, the tenth round will leave as the Inverse Mix Columns stage.

In Intermediate Server:

SHA512 we are using hashing the user history to make more secure instead of SHA216.Hashing is a one way functionality.

VIII. CONCLUSION

In this proposed work, implement a Malicious URL Detection process using machine learning techniques. This focuses on detecting unsafe website URLs and keywords with the help of Hashing blacklist storage. According to few selected features can be used to differentiate between legitimate and malicious web pages. These selected features are many such as URLs and Keywords. In proposed work a service provider that owns a high-quality blacklist, which may be more frequently updated or simply contains more items. User also allowed to directly sharing blacklists with servers in an uncontrollable way could make these dataset be obtained by every user. With the help of efficient classification approach will detect the fake websites accurately and prevent the users from accessing that websites. User will be blocked while hit unsafe urls. This also provides the secure encryption approach avoid the unknown access of search history. The security is provided to the search data which has been stored in the database. Encrypted history local prevent unauthorized user and if access you will get notification .

User will get secure key to decrypt the history detail from admin.

IX. REFERENCES:

[1] Keelveedhi, Sriram, Mihir Bellare, and Thomas Ristenpart. "DupLESS: server-aided encryption for deduplicated storage." In Presented as part of the 22nd {USENIX} Security Symposium ({USENIX} Security 13), pp. 179-194. 2013.

[2]Yuan, Xingliang, Xinyu Wang, Cong Wang, Chenyun Yu, and Sarana Nutanong. "Privacy-preserving similarity joins over encrypted data." IEEE Transactions on Information Forensics and Security 12, no. 11 (2017): 2763-2775.

[3] Hu, Shengshan, Leo Yu Zhang, Qian Wang, Zhan Qin, and Cong Wang. "Towards private and scalable cross-media retrieval." IEEE Transactions on Dependable and Secure Computing (2019).

[4] Wang, Qian, Minxin Du, Xiuying Chen, Yanjiao Chen, Pan Zhou, Xiaofeng Chen, and Xinyi Huang. "Privacy-preserving collaborative model learning: The case of word vector training." IEEE Transactions on Knowledge and Data Engineering 30, no. 12 (2018): 2381-2393.

[5] Cui, Helei, Yajin Zhou, Cong Wang, Qi Li, and Kui Ren. "Towards Privacy-Preserving Malware Detection Systems for Android." In 2018 IEEE 24th International Conference on Parallel and Distributed Systems (ICPADS), pp. 545-552. IEEE, 2018.

[6] Ramezanian, Sara, Tommi Meskanen, Masoud Naderpour, Ville Junnila, and Valtteri Niemi. "Private membership test protocol with low communication complexity." Digital Communications and Networks (2019).

[7] Cui, Helei, Xingliang Yuan, Yifeng Zheng, and Cong Wang. "Towards Encrypted In-Network Storage Services with Secure Near-Duplicate Detection." IEEE Transactions on Services Computing (2018).

[8] Armknecht, Frederik, Jens-Matthias Bohli, Ghassan O. Karame, and Franck Youssef. "Transparent data deduplication in the cloud." In Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, pp. 886-900. ACM, 2015.

[9] Demir, Levent, Amrit Kumar, Mathieu Cunche, and Cédric Lauradoux. "The pitfalls of hashing for privacy." IEEE Communications Surveys & Tutorials 20, no. 1 (2017): 551-565.

[10] Gerbet, Thomas, Amrit Kumar, and Cédric Lauradoux. "A privacy analysis of Google and Yandex safe browsing." In 2016 46th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN), pp. 347-358. IEEE, 2016.