Unclaimed Web Browsing Opposing Traffic Analysis Intervention Using Conjecture Packet

V.Bamadevi, Dr N.Rajendran

Abstract:- Anonymous communication has become a hot research topic in order to meet the increasing demand for web privacy protection. Previously the dummy packet padding strategy was used and it was vulnerable to security attacks. This method inherits huge delay and bandwidth waste, however, there are few such systems which can provide high level anonymity for web browsing. A predicted packet padding strategy is proposed to replace the dummy packet padding method for anonymous web browsing systems. The proposed strategy mitigates delay and bandwidth waste significantly on average. The traffic analysis attack and defense problem are defined and also a metric, cost coefficient of anonymization (CCA) is defined to measure the performance of anonymization. Authorization of access to data in a network, which is controlled by the network administrator. Users choose an ID

Index Terms—Anonymity, predicted packet padding, web browsing.

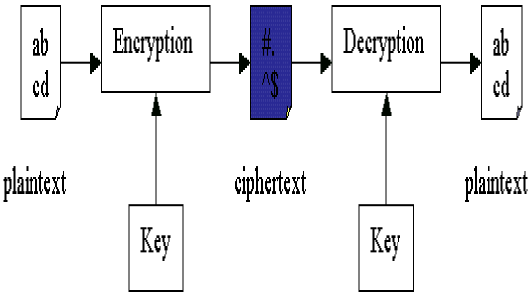
1. INTRODUCTION

Advanced Networking consists of the provisions and policies adopted by a network administrator to prevent and monitor unauthorized access, misuse, modification, or denial of a computer network and network-accessible resources. It involves the authorization of access to data in a network, which is controlled by the network administrator. Users choose an ID and password or other authenticating information that allows them access to information and programs within their authority. It covers a variety of computer networks, both public and private, that are used in everyday jobs conducting transactions and communications among businesses, government agencies and individuals. Networks can be private, such as within a company, and others which might be open to public access.

One common method of attack involves saturating the target machine with external communications requests, so that it cannot respond to legitimate traffic or responds so slowly as to be rendered essentially unavailable. Such attacks usually lead to a server overload. In general terms, DoS attacks are implemented by either forcing the targeted computer(s) to reset, or consuming its resources so that it can no longer provide its intended service or obstructing the communication media between the intended users and the victim so that they can no longer communicate adequately horses are broken down in classification based on .

1.1. Solutions to Security Attacks

- Symmetric Encryption: Invented in 1975 by Diffie and Hellman. A type of encryption where the same key is used to encrypt and decrypt the message. The keys, in practice, represent a shared secret between two or more parties that can be used to maintain a private information link.



- Types of Symmetric-Key Algorithms: Symmetric-key encryption can use either stream ciphers or block ciphers. Stream ciphers encrypt the digits (typically bytes) of a message one at a time. Block ciphers take a number of bits and encrypt them as a single unit, padding the plaintext so that it is a multiple of the block size. Blocks of 64 bits have been commonly used. The Advanced Encryption Standard (AES) algorithm approved by NIST in December 2001 uses 128-bit blocks.

- Hash Function: A Hash function is an algorithm or subroutine that maps large data sets of variable length called keys into smaller data sets of a fixed length. The values returned by a hash function are called hash values, hash codes, hash sums, checksums or simply hashes. Hash functions are primarily used to generate fixed-length output data that acts as a shortened reference to the original data. This is useful when the output data is too cumbersome to use in its entirety. One practical use is a data structure called a hash table where the data is stored associatively. Searching for a person's name in a list is slow, but the hashed value can be used to store a reference to the original data and retrieve constant time (barring collisions). Another use is in cryptography, the science of encoding and safe guarding data. It is easy to generate hash values from input data and easy to verify that the data matches the hash, but hard to 'fake' a hash value to hide malicious data. This is the principle behind the Pretty Good Privacy algorithm for data validation.

- Message Authentication Code: A message authentication code (often MAC) is a short piece of information used to authenticate a message and to provide integrity and authenticity assurances on the message. A MAC algorithm, sometimes called a keyed (cryptographic) hash function (however, cryptographic hash function is only one of the possible ways to generate MACs), accepts as input a secret key and an arbitrary-length message to be authenticated, and outputs a MAC (sometimes known as a tag). MAC algorithms can be constructed from other cryptographic primitives, such as cryptographic hash functions (as in the case of HMAC) or from block cipher algorithms (OMAC, CBC-MAC and PMAC). However many of the fastest MAC algorithms such as UMAC and VMAC are constructed based on universal hashing.

2. RELATED WORK

2.1Analysis of Related Work

Self-Similarity In World Wide Web Traffic: Evidence And Possible Causes Recently the notion of self-similarity has been shown to apply to wide-area and local-area network trace show evidence that the subset of network trace that is due to World Wide Web transfers can show characteristics that are consistent with self-similarity, and the hypothesized explanation for that self-similarity are presented. Using a set of traces of actual user executions of NCSA Mosaic and the dependence structure of WWW trace are examined. First evidence show that WWW trace exhibits behavior that is consistent with self-similar trace models. Then the self-similarity in such cases can be explained based on the underlying distributions of WWW document sizes, the effects of caching and user preference in the transfer, the effect of user think time," and the superimposition of many such transfers in a local area network. To do this it is relied on empirically measured distributions both from client traces and from data independently collected at WWW servers.

Onion Routing is an infrastructure for private communication over a public network. It provides anonymous connections that are strongly resistant to both eavesdropping and traffic analysis. Onion routing's anonymous connections are bidirectional and near real-time, and can be used anywhere a socket connection can be used. Any identifying information must be in the data stream carried over an anonymous connection. An onion is a data structure that is treated as the destination address by onion routers; thus, it is used to establish an anonymous connection. Onions themselves appear differently to each onion router as well as to network observers. The same goes for data carried over the connections they establish.

This paper introduces an information theoretic model that allows quantifying the degree of anonymity provided by schemes for anonymous connections. It considers attackers that obtain - information about users. The degree is based on the probabilities an attacker, after observing the system, assigns to the different users of the system as being the originators of a message. As a proof of concept, the model is applied to some existing systems.

Power Laws, Pareto Distributions and Zipf's Law

Many man-made and naturally occurring phenomena, including city sizes, incomes, word frequencies, and earthquake magnitudes, are distributed according to a power-law distribution. A power-law implies that small occurrences are extremely common, whereas large instances are extremely rare. This regularity or 'law' is sometimes also referred to as Zipf and sometimes Pareto. To add to the confusion, the laws alternately refer to ranked and unranked distributions. Here we show that all three terms, Zipf, power-law, and Pareto, can refer to the same thing, and how to easily move from the ranked to the unranked distributions and relate their exponents.

Statistical Identification of Encrypted Web Browsing Traffic

Encryption is often proposed as a tool for protecting the privacy of World Wide Web browsing. However, encryption particularly implemented in, or in concert with popular Web browsers does not hide all information about the encrypted plaintext. Specification HTTP objects count and sizes are often revealed (or at least incompletely concealed). For this investigation is done on the identity of World Wide Web traffic, based on this unconcealed information in a large sample of Web pages, and it is proved that it identifies a significant fraction of them quite reliably. Some possible counter measures against the exposure of this kind of information and experimentally evaluates their effectiveness.

Web Caching and Zipf-Like Distributions: Evidence And Implications

This paper addresses two unresolved issues about web caching. The First issue is whether web requests from a fixed user community are distributed according to Zipf's law. Several early studies have supported this claim, while other recent studies have suggested otherwise. The second issue relates to a number of recent studies on the characteristics of web proxy traces, which have shown that the hit-ratios and temporal locality of the traces exhibit certain asymptotic properties that are uniform across the deferent sets of the traces. In particular, the question is whether these properties are inherent to web accesses or whether they are simply an artifact of the traces. Answers to these unresolved issues will facilitate both web cache resource planning and cache hierarchy design. The answers to the two questions are related. Firstly the page request distribution seen by web proxy caches using traces from a variety of sources is investigated. Secondly the distribution does not follow Zipf's law precisely, but instead follows a Zipf-like distribution with the exponent varying from trace to trace is investigated. Furthermore, there is only a weak correlation between the access frequency of a web page and its size and a weak correlation between access frequency and its rate of change.

Consider a simple model where the web accesses are independent and the reference probability of the documents follows a Zipf-like distribution. The model ends yielding asymptotic behaviors that are consistent with the experimental observations, suggesting that the various observed properties of hit-ratios and temporal locality are indeed inherent to web accesses observed by proxies. Finally the web cache replacement algorithms are revisited and it is shown that the algorithm that is suggested by this simple model performs best on real trace data. The results indicate that while page requests do indeed reveal short-term correlations and other structures, a simple model for an independent request stream following a Zipf-like distribution is sufficient to capture certain asymptotic properties observed at web proxies.

Traffic Morphing: An Efficient Defense against Statistical Traffic Analysis

Recent work has shown that properties of network traffic that remain observable after encryption, namely packet sizes and timing, can reveal surprising information about the traffic's contents (e.g., the language of a VoIP call passwords in secure shell logins or even web browsing habits . A common tactic for mitigating such threats is to pad packets to uniform sizes or to send packets at fixed timing intervals; however, this approach is often inefficient. In this paper, a novel method is proposed for thwarting statistical traffic analysis algorithms by optimally morphing one class of traffic to look like another class. Through the use of convex optimization techniques modify packets in real-time to reduce the accuracy of a variety of traffic classifiers while incurring much less overhead than padding. That morphing works well on a wide range of network data providing better privacy and lower overhead than native defenses.

Modeling Online Browsing And Path Analysis Using Click stream Data

Click stream data provide information about the sequence of pages or the path viewed by users as they navigate a website. It is also showed how path information can be categorized and modeled using a dynamic multinomial probate model of Web browsing. Estimation of this model is done using data from a major online bookseller. Our results show that the memory component of the model is crucial in accurately predicting a path. In comparison, traditional multinomial probate and first-order Markov models predict paths poorly. These results suggest that paths may reflect a user's goals, which could be helpful in predicting future movements at a website. One potential application of our model is to predict purchase conversion. It was found that after only six viewings purchasers can be predicted with more than 40% accuracy, which is much better than the benchmark 7% purchase conversion prediction rate made without path information. This technique could be used to personalize Web designs and product offerings based upon a user's path.

3. METHODOLOGY

Predicted Packet Padding Method

Packet padding can achieve anonymity for web browsing even though it is not practical because there are huge delays and extreme bandwidth cost caused by the dummy packet padding mechanism. Therefore data encryption and predicted packet padding are combined, rather than dummy packet padding, at the server side to achieve feasible anonymous web browsing. In the practice of traffic analysis, eavesdroppers employ various features to break anonymity, such as packet arrival time interval, number of packets of different web objects. The aim is to present the novelty and effectiveness of the proposed predicted packet padding strategy and therefore confine our discussion within the following reasonable conditions and algorithm.

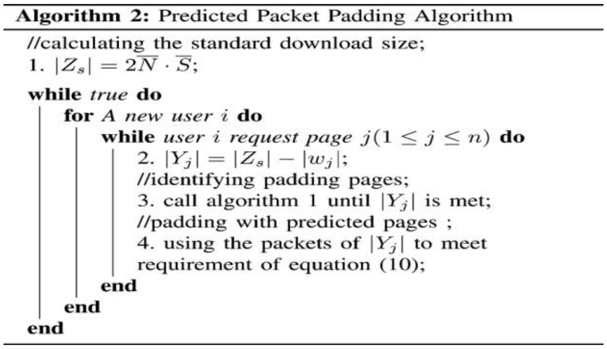
1) Here only the traffic analysis attack on the fingerprint of web object size is discussed (number of packets). Note that the proposed method is also effective against other types of attacks that use other kinds of page fingerprint, such as packet arrival time interval.

2) Here the cache is in place at client computers. The memory of major computing devices, such as PC, notebook, are sufficient to offer caching. Moreover, the major web browsers, e.g., IE and Firefox, use cache browsing as the default setting.

For a given dynamic web site, it is assumed that there are $n\,(n > 0)$ web pages, denoted as$\{w_{1,} w_2, \ldots, w_n\}$, where $\mathcal{W}_i$ is the ith$(1 \le i \le n)$ popular web page. For a given web page $w_i(1 \le i \le n)$of the web site, it possesses m (m > 0)web objects, $\{w_i^1, w_i^2, .., w_i^m\}$. In general, we denote $w_i^k(1 \le i \le n, 1 \le i \le n)$as the kth web object of the ith web page; and we denote the size (in terms of packets) of web object as $w_i^k(1 \le i \le n, 1 \le i \le n)$ as$|w_i^k|$.The fingerprint of a web page as a set, $\{t_i^1, t_i^2, .., t_i^m\}$.Each element of the set, $t_i^k(1 \le i \le n, 1 \le k \le m)$, is defined as follows

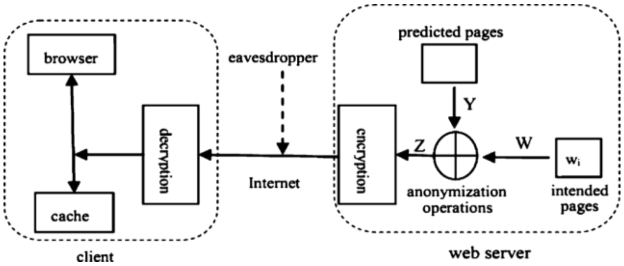
$$t_i^k = \frac{|w_i^k|}{\sum_{j=1}^{m} |w_i^j|}$$

Predicted Packet Padding Algorithm:



```
Algorithm 2: Predicted Packet Padding Algorithm
//calculating the standard download size;
1. $|Z_s| = 2\overline{N} \cdot \overline{S}$;
while true do
    for A new user i do
        while user i request page $j(1 \le j \le n)$ do
            2. $|Y_j| = |Z_s| - |w_j|$;
            //identifying padding pages;
            3. call algorithm 1 until $|Y_j|$ is met;
            //padding with predicted pages ;
            4. using the packets of $|Y_j|$ to meet
            requirement of equation (10);
        end
    end
end
```

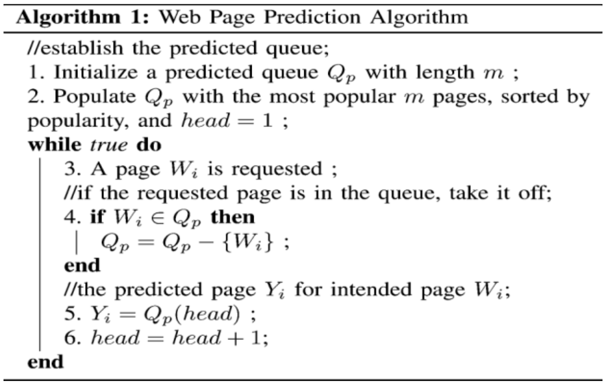
The proposed predicted packet padding mechanism is shown in Fig. 4.1.1. At the server side, the predicted web pages Y will be used as cover traffic for packet padding for the intended page W. The output of the anonymization, Z, will be encrypted and transported to the client via the Internet or related anonymous networks. At the client side, once is decrypted, the intended page will be displayed by a web browser, and the predicted pages will be deposited at the local cache. We use symbol $\oplus$ to represent the packet padding operation.



Anonymous Web Browsing System Model with Packet padding using Predicted pages

The details are depicted as follows: The Client submits an encrypted HTTP request for web page $w_i(1 \le i \le n)$ to the web server. The web server will return the intended traffic $w_i = \{w_i^1, w_i^2, .., w_i^m\}$, where $w_i^k(1 \le k \le m)$ represents the kth web object of web page w_i, and the size of the object w_i^k is $|w_i^k|$ in terms of packets. A common method is to inject cover traffic $y_i = \{y_i^1, y_i^2, .., y_i^m\}$, where y_i^k $(1 \le k \le m', m \le m')$ represents the cover traffic for $w_i^k(1 \le k \le m)$ into the intended traffic w_i to obtain a covered output $z_i = \{z_i^1, z_i^2, .., z_i^{m'}\}$.

The predicted web pages will be used as cover traffic for packet padding for the intended page. The output of the anonymization will be encrypted and transported to the client via the Internet or related anonymous networks. At the client side, once is decrypted, the intended page will be displayed by a web browser, and the predicted pages will be deposited at the local cache. The following page prediction algorithm is used.



```
Algorithm 1: Web Page Prediction Algorithm
//establish the predicted queue;
1. Initialize a predicted queue $Q_p$ with length $m$ ;
2. Populate $Q_p$ with the most popular $m$ pages, sorted by
popularity, and $head = 1$ ;
while true do
    3. A page $W_i$ is requested ;
    //if the requested page is in the queue, take it off;
    4. if $W_i \in Q_p$ then
        $Q_p = Q_p - \{W_i\}$ ;
    end
    //the predicted page $Y_i$ for intended page $W_i$;
    5. $Y_i = Q_p(head)$ ;
    6. $head = head + 1$;
end
```

Different from the traditional dummy packet padding strategy, we use the predicted web pages that users are going to download in the near future as the cover traffic. At Client site, after the decryption, the intended data W goes to the web browser, and the cover traffic Y goes to the cache. Following the web browser cache mechanism, client's following requests will be checked with the cache first, if the requested web page is in the cache, then there is no need to download it from the server again, and there is no request for the page from Client to the server. As a result, the goal of anonymity in web browsing is served and the delay issue is addressed from a long term perspective. It is true that we cannot predict the expected web pages 100% accurately for clients, therefore, part of the pre fetched pages may never be requested, and such pages cause bandwidth waste and delay as dummy packets do. In order to measure the efficiency of anonymization operation, a metric is defined as follows.

3.1.COST COEFFICIENT OF ANONYMIZATION (CCA)

Let function C(X) represents the cost for network traffic. For an intended network traffic W, we inject a cover traffic Y to achieve anonymity, then the cost coefficient of the anonymization operation is defined as

$$\boldsymbol{\beta} = \frac{\mathbf{C(Y|W)+C(W)}}{\mathbf{C(W)}}$$

Where $C(Y|W)$ denotes the cost of traffic, which is used to cover an intended traffic W.

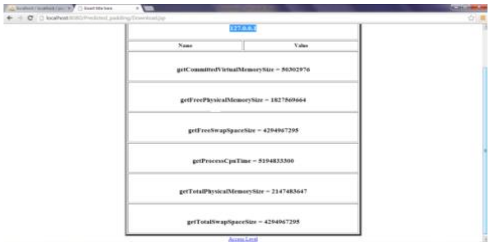
The cost function $C(X)$ is defined as the number of packets for a given network traffic. Suppose a user has browsed $k(k = 1,2,\ldots)$ pages, then the cost coefficient of anonymization is defined as

$$\boldsymbol{\beta} = \frac{\sum_{i=1}^{k} (|y_i|+|w_i|)}{\sum_{i=1}^{k} |w_i|} = \frac{\sum_{i=1}^{k} |z_i|}{\sum_{i=1}^{k} |w_i|},$$

In an ideal situation, the intended page possesses $\overline{\mathcal{N}}$ objects and each object's size is $\overline{\mathcal{S}}$, and the same for the cover traffic. Therefore, each session of downloading possesses $2\overline{\mathcal{N}}$ web objects and each web object's size is $\bar{S}$ packages. As a result, the standard size of one data downloading session is as follows:

$$\left|\mathbf{Z}\right| = \left|\mathbf{W}\right| + \left|\mathbf{Y}\right| = 2\overline{\mathcal{N}}.\overline{\mathcal{S}}.$$

4. EXPERIMENTS AND RESULTS



127.0.0.1

Name | Value

getCommittedVirtualMemorySize = 50302976

getFreePhysicalMemorySize = 1827569664

getFreeSwapSpaceSize = 4294967295

getProcessCpuTime = 5194833300

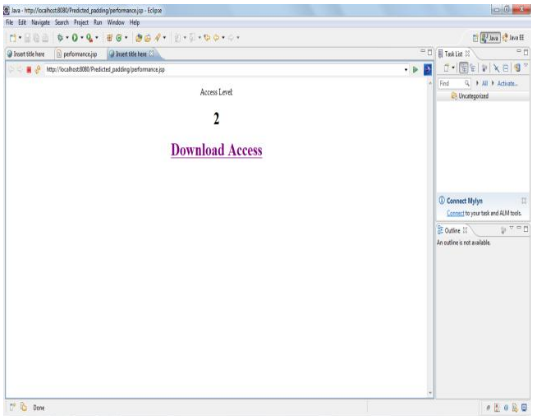
getTotalPhysicalMemorySize = 2147483647

getTotalSwapSpaceSize = 4294967295

Access Level

Cache Memory Calculation

In this step IP address, CPU time, system CPU load, virtual memory size, physical memory size, swap space size is calculated and then the priority is provided. By this method the time delay is certainly reduced.


Java - http://localhost:8080/Predicted_padding/performance.jsp - Eclipse
File Edit Navigate Search Project Run Window Help
Java
Java EE
Insert title here
performance.jsp
Insert title here
http://localhost:8080/Predicted_padding/performance.jsp
Task List
Find
All
Activate...
Uncategorized
Connect Mylyn
Connect to your task and ALM tools.
Outline
Done

Access Level

2

Download Access

An outline is not available.

The Access level is provided based on the cache memory available which is calculated using predicted packet padding algorithm. By this process the time delay is reduced and therefore the bandwidth waste is also reduced.

The CCA of predicted packet padding is less when compared to that of Dummy packet padding method. Thus the predicted packet padding method to make anonymous web browsing feasible in practice. My goal was to show the effectiveness of the proposed method. However, it is hard and expensive to offer anonymous communication in reality as any vulnerable component of a system can be used by attackers to break the system anonymity.

5. CONCLUSION

The main focus is on reducing the delay and bandwidth waste of anonymous web browsing systems in order to make anonymous web browsing applicable for web viewers. The predicted packet padding strategy helps to achieve this goal. A simple mathematical model for the packet adding mechanism was established, followed by a thorough analysis and comparison between the proposed strategy and the traditional dummy packet padding method. Moreover, a metric, CCA, was defined to measure the performance of different packet padding strategies. The CCA of the proposed padding strategy decreases when browsing length increases, which confirms the advantages of the proposed method. However, the CCA increases for website when the browsing length is more than 20. This is caused by the decrease of the prediction accuracy. However, it is always less than 2, which means the proposed method beats the traditional dummy packet padding strategy when the browsing length is sufficient.

REFERENCES

1. M. Edman and B. Yener, "On anonymity in an electronic society: A survey of anonymous communication systems," ACM Computing Survey, vol. 42, no. 1, 2009.
2. D. Chaum, "Untraceable electronic mail, return addresses, and digital pseudonyms," Commun. ACM, vol. 24, no. 2, pp. 84–88, 1981.
3. D. M. Goldschlag, M. G. Reed, and P. F. Syverson, "Hiding routing information," in Information Hiding, 1996, pp. 137–150.
4. M. G. Reed, P. F. Syverson, and D. M. Goldschlag, "Anonymous connections and onion routing," IEEE J. Select. Areas Commun. vol. 16, no. 2, pp. 482–494, Feb. 1998.
5. [Online]. Available: http://www.torproject.org
6. R. Dingledine, N. Mathewson, and P. F. Syverson, "Tor: The secondgeneration onion router," in Proc. USENIX Security Symp., 2004, pp. 303–320.
7. M. K. Reiter and A. D. Rubin, "Crowds: Anonymity for web transactions," ACM Trans. Inform. Syst. Security, vol. 1, no. 1, pp. 66–92, 1998.
8. C. Diaz, S. Seys, J. Claessens, and B. Preneel, R. Dingledine and P. Syverson, Eds., "Towards measuring anonymity," in Proc. Privacy Enhancing Technologies Workshop (PET 2002), Apr. 2002, Springer-Verlag, LNCS 2482.
9. A. Serjantov and G. Danezis, R. Dingledine and P. Syverson, Eds., "Towards an information theoretic metric for anonymity," in Proc. Privacy Enhancing Technologies Workshop (PET 2002), LNCS 2482, Apr.
2002, Springer-Verlag.
10. Q. Sun, D. R. Simon, Y.-M. Wang, W. Russell, V. N. Padmanabhan, and L. Qiu, "Statistical identification of encrypted web browsing traffic," in Proc. IEEE Symp. Security and Privacy, 2002.

V.Bamadevi, currently pursuing M.Phil Computer Science under one of the college affiliated to Periyar University. I also received my B.Sc and M.Sc degrees from the affiliated Colleges under Periyar University and Anna University.

I have done two projects during my Post Graduate.

Dr N.Rajendran, the Head of the Department of Computer Science in Vivekanandha College for Women. He has done his Ph.D degree in Data Mining.