CSE4001

PARALLEL AND DISTRIBUTED COMPUTING

PROJECT REPORT

INTERNET TRAFFIC PRIVACY ENHANCEMENT WITH MASKING: OPTIMIZATION AND TRADEOFFS

PRESENTED BY:

VARTIKA (18BCE2084)

ANSHUMAAN SINGH (18BCE2193)

RAJ KUSHWAHA (18BCE2489)

PRESENTED TO:

PROF. DEEBAK B D

School of Computing Science and Engineering, VIT University,

Vellore 632014, Tamil Nadu, India

CONTENTS

PROBLEM ADDRESSED

An increasing number of recent experimental works have been demonstrated that the supposedly secure channels in the Internet are prone to privacy breaking under many respects, due to packet traffic features leaking information on the user activity and traffic content. We aim at understanding if and how complex it is to becloud the information leaked by packet traffic features, namely packet lengths, directions, and times: we call this technique traffic masking. A number of works over the last few years have given extensive experimental evidence that even within a secure channel, a packetized flow leaks information to an adversary through observation of features of traffic flows. Thus with the growning domain of internet, these masking technques are necessary for its safe and secure usage . Therefore, here we aim to analyze different masking techniques used for internet traffic privacy , their current accuracy and flaws and try to analyse the best methods that can be used for traffic privacy in future .

PRIOR RESEARCH

PACKET LENGTH AND DIRECTION MASKING CAN EXPLOIT TWO BASIC MECHANISMS: PADDING AND FRAGMENTING

PADDING - Padding amounts to the insertion of a number of extra bit in the packet, that can be stripped off by the recipient so that information is not corrupted, but such that the adversary measures an altered value of packet length of the ciphered packet. Padding can come in two forms: adding bits to a packet provided by a host or creating a fake packet (dummy packet, that will be discarded altogether at destination).

FRAGMENTING- Fragmenting is another form of packet length modification: from a single original packet with payload L, n packets spring out, with payload lengths Lj , j = 1, . . . , n, such that L = L1 +· · ·+Ln. Overhead must be added to the newly generated packets to allow correct reassembly of the original packet at destination1. New packets are added to the original traffic flow (byte overhead), and a delay is imposed to the original packet flow (time overhead).

EXISTING ARCHITECTURES AND PROTOCOL

SSH: RFC 4253 [24] specifies that padding can be of arbitrary length, such that the total length of packet is a multiple of the cipher block size or 8, whichever is larger. There MUST be at least four bytes of padding. The padding SHOULD consist of random bytes. The maximum amount of padding is 255 bytes. In the basic OpenSSH implementation, the padding length will depend on the payload and the cipher block size. So although

the padding itself is random, the final packet size will be just a step function of the payload size.

SSL: Also the specifications of TLS protocol (versions 1.1 and 1.2) described in RFCs 4346 and 5246, offer the possibility to add some padding in order to alterate the packet lengths. Even here, padding that is added to force the length of the plaintext must be an integral multiple of the block cipher's block length. The padding may be any length up to 255 bytes. The choice of possible techniques for adding padding is left to the discretion of the individual implementations. GnuTLS is one of the most famous secure communications library implementing the SSL, TLS protocols. If properly enabled, it allows to add a random padding with uniform distribution.

IPSec: IPSec is a standards of network level that provides various cryptographic algorithms in order to provide security services. Even IPSec allows to add a random padding with uniform distribution. In 2007, Kiraly et al. [25, 26] have proposed a new framework based on IPSec, which allows to integrate techniques of Traffic Flow Confidentiality (TFC). The authors have designed a new architecture that can be considered as a specific substrate maintaining backward compatibility with traditional IPsec implementations. The developed architecture is structured into two main modules: one deals with the logical control of TFC procedures and algorithms

SIGNIFICANCE

The threats to one's privacy on the Internet are two-fold: your online actions could be monitored by unauthorized parties and logged and preserved for future access many years later. You might not realize that your personal information has been monitored, logged, and subsequently disclosed; those who would compromise your privacy have no incentive to warn you. The threat of long-term storage and eventual disclosure of personal information is especially acute on the Internet. It is technically quite easy to collect information (such as a compendium of all posts you have made to electronic newsgroups) and store it for years or decades, indexed by your name for easy retrieval.

INTRODUCTION

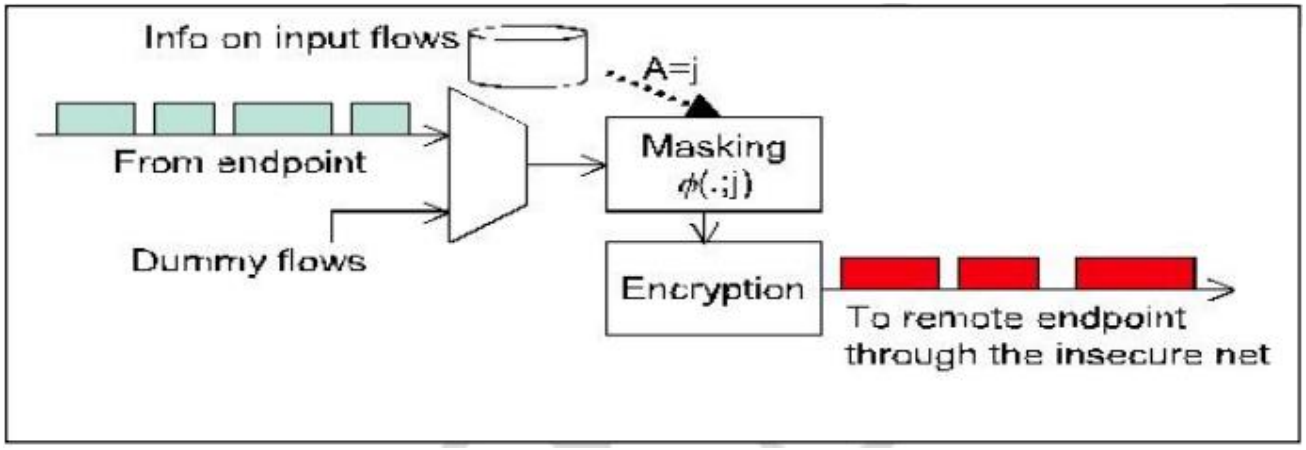
The increased use of the Internet for everyday activities is bringing new threats to personal privacy. An increasing number of recent experimental works have demonstrated that the supposedly secure channels in the Internet are prone to privacy breaking under many respects, due to packet traffic features leaking information on the user activity and traffic content. From the last few years an extensive experimental evidence within a secure channel, a packetized flow leaks information through observation of features of traffic flows, for example, the ordered sequence of packet lengths, packet inter arrival times and packet directions.When the flow is carried within a secure channel (e.g., SSL/TLS or SSH connections), to identify the type of service or application protocol run among a given set of alternatives to classify the traffic. Other privacy breaking attacks based on analysis of packet flow features have been demonstrated, for example, to profile web access, to infer language of phone calls or even conversation transcripts .This communication privacy break is a positive proof that ciphering does not conceal all relevant information of a packetized application flow; hence, we aim at investigating protection of privacy against traffic analysis. Besides being a privacy issue, traffic analysis tools can be useful to network administrators and operators for enforcement of security policies and traffic filtering, or to support quality of service mechanisms. We term this as traffic masking.

Our objective is to understand the complexity of the information leakage on the client activity and traffic. We collected the ongoing research papers and observed different techniques used to obtain traffic privacy at an optimizing level. We will understanding if and how complex it is to obfuscate the information leaked by packet traffic features, namely packet lengths, directions.Traffic masking is definitely better than the older

method used for protecting the privacy of information like encryption. Hence , seeing the scope of technologies used and that can be used to resolve the privacy issues.

LITERATURE SURVEY

The traffic masking operation includes introducing dummy flows, to modify the a priority probabilities as Pj into new values as Qj , and transforming each flow sent through the network and the flow transformation between the different hosts implies message padding, fragmenting, insertion of dummy messages, and message delaying.



Dummy flows are added to modify the priority of generating applications. From the input for the flow transformation can be considered as a A=j and from the end point this flow can be masked as ω(j).We assume that when an eavesdropping adversary, aiming at flow classification. The adversary can be observing ciphered and masked flows (including dummy flows) .It can detect the feature vector y for each observed flow. In other words, the

adversary can collect samples y of the random variables Y. The reverse operations (deciphering and de masking) take place at receiver side. Probability of correctly classified ob-served flows to get its theoretical minimum 1=M. Given the model above, we mean removing any information leakage that could be exploited by the adversary to classify observed flows.

EXISTING TECHNIQUES FOR TRAFFIC ANALYSIS

Over the years various techniques have been implemented by people and used ,here are those countermeasures developed .

1. Wright et al. [8] make use of convex optimization techniques to modify the source packet lengths distribution to look like a target distribution (morphing), with minimum overhead. No explicit solution is provided and protocol complexity of morphing a given flow into a different one is not accounted for.

2. Shui Yu et al.[9-11] implemented a new strategy of packet padding aiming at offering perfect anonymity on web browsing. Their proposal comes from the fact that users generally access a number of web pages at one web site according to their own habits or interests. This has been confirmed by applications of web caching and web page prefetching technologies. The proposed solution allows to disguise the fingerprints of web sites at the server side by injecting predicted web pages that users are going to download as cover traffic, rather than using dummy packets as cover traffic. So Authors conclude that from a long term viewpoint, this novel strategy wastes limited bandwidth and causes limited delay.

3.Zhang at al. [12] contrast traffic analysis by means of traffic reshaping technique. By exploiting multiple virtual MAC interfaces, an application flow is dynamically subdivided in a set of new flows and then dispatched

among these interfaces, and different traffic features are reshaped on each virtual interface to hide those of the original traffic.

4.Furthermore, in [13] Dyer et al. show that it is still possible to classify traffic flows after masking. They consider nine masking countermeasures applied to web pages, and adopt some machine learning algorithms (Na¨ıve bayes, multinomial Na¨ıve bayes and support vector machine) to identify which of two web pages was downloaded. Accuracy of 98% is obtained, and they conclude that more investigation is necessary to effectively conceal the whole information leakage.

METHODOLOGY

GENERAL MODEL OF AN APPLICATION FLOW

Any application traffic flow between an initiator entity A and the responder entity B (e.g., client and server for the given flow, respectively) can be cast into a sequence of N _ 1 message bursts.2 Each burst consists of one or more messages in one direction (A ! B or B ! A). Bursts in the two opposite directions alternate, starting from the initial burst sent by the initiator A to the responder B.

A full description of the flow is obtained with:

1. the vector K - K1 ... KN of the numbers of messages in each burst.
2. message lengths in each burst, denoted as Li =[Li(1)……Li(Ki)] where i=1,…….N
3. message epochs,3 denoted as Ti =[Ti(1)……Ti(Ki)] where i=1,…….N

The following figure depicts an example of two flows, where A and B sides are represented by vertical lines and time increases downward. The flow labeled (a) has N = 4, with K = [1, 3, 1, 1]. For flow (b), it is N =6 and K = [1, 1,1, 1, 1, 1].

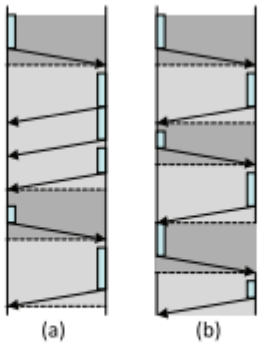


Fig. 1. Examples of message exchanges of two application flows: (a) one way data transfer, like http; (b) alternate messages, like most signaling and control protocols.

DIFFERENT TYPES OF MASKING TECHNIQUES USED

OPTIMAL FULL MASKING

In optimal masking ,we focus on the case of two applications (M = 2) and state an optimization problem that yields a constructive solution for a full masking algorithm that achieves minimum overhead in the set of ideal masking algorithms. This optimal full masking algorithm serves as a term of comparison for practical masking, while it is unfeasible to realize, both because of computational complexity and because, in principle, it requires the entire flow to be available to the masking device to decide upon each message transformation. In optimal masking we take a flow belonging to A1, with features xh . Next draw a random index in the set [1,w2] of value k with probability ch(k). And then transform the input flow into the output masked flow with features yh(k).

PRACTICAL AND PARTIAL MASKING

A key limitation of ideal masking algorithms is the requirement that the entire flow be available to decide on masking, before sending any message out. This cannot work for transactional, interactive applications, where a message burst is produced by the application based on previously received bursts from the remote entity.

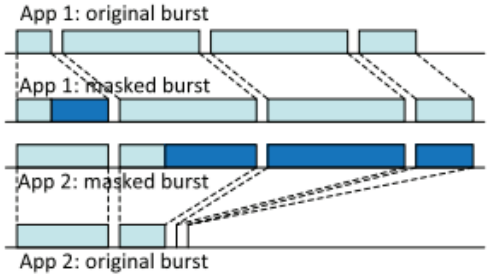


Fig. 3. Examples of message length masking of bursts of two applications.

Thus here we aim at practical masking algorithm, i.e., applicable to a message/burst only based on features of previous messages/bursts. First,

we consider statistical masking all messages of the input traffic then we introduce an algorithm to mask input traffic only partially, so as to be able to tradeoff between overhead and degree of masking , Neither case leads to full masking, because at least correlations among features of different bursts cannot be fully masked. In both cases, we define masking so as to minimize overhead. The key idea applied is to apply the optimized full masking of burst by burst, so that the decision on the masking flow can be taken at each endpoint as the traffic flow runs. Hence, the feature subvector of message lengths and gap time for each burst separately. The complexity of the ideal, full masking within a burst is limited, because typical bursts comprise one or few messages.

PRACTICAL MASKING

The key idea is to apply the optimized full masking is of burst by burst, so that the decision on the masking flow can be taken at each endpoint as the traffic flow runs. Hence, the feature subvector of message lengths and gap time for each burst separately. The complexity of the ideal, full masking within a burst is limited, because typical bursts comprise one or few messages. The overall flow masking is no more full, because correlations across bursts are not taken care of. In general, padding, fragmentation, and dummy messages can be used. If only padding is used, we define the Burst-by-Burst Padding Only (BbBPO) masking algorithm and minimum byte and time overhead is obtained as follows for each burst

FIXED PATTERN MASKING

Fixed Pattern Masking Full or practical flow masking algorithms in previous sections are statistical in nature. They imply knowledge of pdfs of the features to be masked. As a counterpart, they promise some kind of optimization of the introduced overhead and delay. A much simpler and supposedly far from optimal approach is fixed pattern masking. In general,

it means that the input flow, whatever its originating application, be forced to be framed into a predefined pattern with features. Enforcement of these features is obtained practically as follows: Upon emission of a given burst, the sending application entity shapes the message(s) making up the burst according to the desired fixed pattern, by using fragmenting, padding, and delaying. If at any given time, an output message must be issued according to the fixed masking pattern, while there are no input bytes to be sent, a dummy message is emitted. Imposing such a fixed pattern to input flows generated by whatever application is certainly possible and it is guaranteed to raze any possible information useful to the classification adversary. Moreover, it can be applied message by message. So fixed pattern masking is a practical, full masking algorithm. The choice of the values of the fixed pattern features y0 influences the resulting overhead and delay. In general, the choice of y0 leading to minimal overhead is a multidimensional optimization problem, given the overall mix of traffic that it is expected at the masking device. Special cases of the fixed pattern, that simplify its implementation, are obtained by, for example, setting a common value for features of all bursts in a same direction.

CONTRIBUTIONS

The Authors of this report, i.e, Vartika, Anshumaan Singh, Raj Kushwaha, distributed work in terms of research, and shared the ideas presented in the various papers, materials and information about technologies available online. We were constantly supported, guided and motivated by our faculty, Prof. Deebak B D, and he has helped us thoroughly clearing our concepts about parallel and Distributed computing in computer systems. The report is also prepared with a joint effort by Vartika, Anshumaan Singh and Raj Kushwaha.

RESULTS AND DISCUSSION

In this paper we tried to assess the performance of several masking algorithms in terms of overhead and complexity of a traffic flow masking device . We characterize that the optimal masking algorithm is under the constraint of perfect masking. To overcome implementation difficulties arising in case conversational application are masked, practical masking algorithms are found to be more optimal , working burst by burst (or message by message) or even masking only a fraction of the overall flow. Practical masking lets correlation over features of different bursts leak, yet it offers practical realizability of the masking device and reduced overhead, though leakage might be critical in case of adversaries that are not time constrained .

There is a basic distinction must be done between real-time and offline adversary, observing features of an extended segment of the flow. In the former case, even if some information useful to the opponent leaks when relaxing to practical masking from full masking, still it appears that classification is essentially impaired. This is due to the limited number of features used by the adversary . Things turn out to be completely different if the adversary can take time to classify the flow and observe its features over several bursts, in the order of tens. In that case practical masking, though canceling any information useful for classification inside each burst, cannot remove entirely correlations across features of different bursts. So, statistical, practical masking, even though it minimized overhead burst by burst, still introduces a considerable amount of overhead while failing to protect privacy against an offline adversary.

Another question concerns the amount of data requested in advance by the masking device, to carry out an accurate process of obfuscationAs the duration of a flow grows up, the amount of data required becomes very high and quite hard to estimate reliably. Thus simpler masking approaches, that give up to global or local optimization leveraging on statistical masking and resort instead to rigid, fixed pattern masking. While increasing overhead with respect to statistical masking, as expected because no optimization is attempted with fixed masking, yet that approach removes any information that could be exploited by the classification adversary, preserves implementation simplicity, and overhead price is not terribly greater than that entailed by optimized solutions .

FURTHER RESEARCH

A different and potentially promising approach we can pursue is aggregate masking, i.e., application of feature masking at IP flow level to the aggregate flow carried over an IP tunnel, for example, an IPSec connection. Route aggregation lets you take several specific routes and combine them into one inclusive route. Route aggregation can reduce the number of routes a given protocol advertises. The aggregates are activated by contributing routes. For example, if a router has many interface routes subnetted from class C and is running Routing Information Protocol (RIP) 2 on another interface, the interface routes can be used to create an aggregate route (of the class C) that can then be redistributed into RIP. Creating an aggregate route reduces the number of routes advertised using RIP.

OUR REFERENCES

BASE PAPER

https://www.researchgate.net/publication/260721483_Internet_Traffic_Privacy_Enhancement_with_Masking_Optimization_and_Tradeoffs

OTHER REFERENCE PAPERS

1. https://ieeexplore.ieee.org/document/6464256
2. https://www.sciencedirect.com/science/article/abs/pii/S138912861400437X
3. https://ieeexplore.ieee.org/document/4738466
4. https://www.researchgate.net/publication/224184897_Optimum_packet_length_masking
5. https://ieeexplore.ieee.org/document/6193430
6. https://ieeexplore.ieee.org/abstract/document/584680
7. http://www.rroij.com/open-access/enhancing-the-traffic-privacy-with-leakagedetection-and-optimization.pdf
8. Thomas Karagiannis, Konstantina Papagiannaki, and Michalis Faloutsos. BLINC: multilevel traffic classification in the dark. In SIGCOMM, pages 229–240, 2005.
9. Shui Yu, Theerasak Thapngam, Hou In Tse, and Jilong Wang. Anonymous web browsing through predicted pages. In IEEE Globecom Workshops, pages 1581–1585, 2010.
10. Shui Yu, Theerasak Thapngam, Su Wei, and Wanlei Zhou. Efficient Web Browsing with Perfect Anonymity Using Page Prefetching. In ICA3PP (1), pages 1–12, 2010.
11. Shui Yu, Guofeng Zhao, Wanchun Dou, and Simon James. Predicted Packet Padding for Anonymous Web Browsing Against Traffic Analysis Attacks. pages 1381–1393, 2012.

12. Fan Zhang, Wenbo He, and Xue Liu. Defending Against Traffic Analysis in Wireless Networks through Traffic Reshaping. In ICDCS, pages 593– 602, 2011.
13. Kevin P. Dyer, Scott E. Coull, Thomas Ristenpart, and Thomas Shrimpton. Peek-a-Boo, I Still See You: Why Efficient Traffic Analysis Countermeasures Fail. In IEEE Symposium on Security and Privacy, pages 332–346, 2012
14. T. Nguyen and G. Armitage, "A Survey of Techniques for Internet Traffic Classification Using Machine Learning," IEEE Comm. Surveys Tutorials, vol. 10, no. 4, pp. 56-76, Fourth Quarter 2008.
15. A. Dainotti, A. Pescape, and K. Claffy, "Issues and Future Directions in Traffic Classification," IEEE Network, vol. 26, no. 1, pp. 35-40, Jan./Feb. 2012.
16. S. Yu, G. Zhao, W. Dou, and S. James, "Predicted Packet Padding for Anonymous Web Browsing Against Traffic Analysis Attacks," IEEE Trans. Information Forensics and Security, vol. 7, no. 4, pp. 1381-1393, Aug. 2012
17. A. Iacovazzi and A. Baiocchi, "Optimum Packet Length Masking," Proc. Int'l Teletraffic Congress, 2010.