Rattle Adversary In IP Address Race Of Puzzler Networks

1Mr. R.Ramesh Kannan, 2Saranya J, 3Sorna Sukanya G
1AP/CSE, Dhanalakshmi College of Engineering, 2Student, 3Student
1kannanrrk@gmail.com, 2saranjai11@gmail.com, 3sagarika.g19@gmail.com

Department of Computer Science and Engineering, Anna University, Chennai, India

Abstract— Large datasets of existent network streams procured from the mutual network are indispensable resource for the exploration group. An application encompasses network modeling and simulation, recognition of privacy assaults, and formalization of research results. Unfortunately, network flows transfer extremely delicate data, and this demoralizes the emergence of those information sets. Indeed, existing techniques for network flow sanitization are vulnerable to different kinds of assaults, and results suggested for micro data anonymity cannot be directly applied to network traces. In this paper we provide an extremely secure transfer of delicate data through network flows by obfuscating the IP address. We discover the risks presented by the incremental release of network flows, we suggest a modern defense algorithm, and we officially certify the attained confidentiality guarantees. We group hosts based on the fingerprint and obfuscated address and secure the IP address during the release of incremental network flows. We demonstrate it by sending text and image through the flows and showcase how the quality of the data sent remain unchanged and completely secured. An ample speculative assessment of the algorithm for incremental obfuscation, performed with billions of existent Internet flows, reveals that our obfuscation technique conserves the utility of flows for network traffic analysis.

Keywords— DataSharing, network flow analysis, privacy, security, IP networks, Uncertainty

I. INTRODUCTION

Network flows carry extremely confidential information that should not be released for privacy reasons. For the sake of this work, we assume that the payload is removed from all packets. However, even in this case, an adversary observing the source and destination IP addresses and may easily attack the data. On the other hand, those datasets may also help an adversary to perform security attacks. For instance, observing the traffic of a target network, an adversary could identify possible bottlenecks to be exploited for denial-of-service (DoS) attacks. Crypto technique, which is currently, incorporated within several network flows collector tools. Crypto is a sanitization tool for network flows that encrypts IP addresses in a prefix-preserving manner. Crypto-Pan [1] has the following properties: 1) it perform a one to one mapping from original IP addresses to anonymized IP addresses [4], based on AES cryptographic encryption; 2) IP address encryption is prefix-preserving and 3) it is consistent across the whole set release traces. In this paper we propose an obfuscation technique which involves 1) Fingerprint based group creation. 2) Creating Group Identity and Group Intimation. 3) Obfuscation of sensitive data for network flows.

II. METHODOLOGY

In this paper we perform the obfuscation for incremental network flows. We illustrate the confidentiality threats and data quality issues involved in the incremental release of network flows. Furthermore, we show how our defense algorithm- MD5 can be extended to overcome these issues. We also prove the confidentiality guarantees of our defense, as well as the computational complexity of the extended algorithm.

A. Fingerprint based Group creation

Fingerprint creation is based on OS, RAM, Processor, Username and IP address on each node. Creating fingerprint for each nodes and mapping the nodes. Nodes having similar fingerprint values are grouped together. The goal of our fingerprint-based IP-groups creation method is to enforce property obfuscation while preserving the quality of obfuscated data. In order to reach this goal, IP-groups are
created by grouping together IPs whose hosts have a similar fingerprint (i.e., they originate similar flows). Group creation is done by mapping fingerprint vector in an integer value by exploiting the Hilbert space filling curves [3].

B. Creating Group Identity and Group Intimation

To identify the group, we create a group ID for each group based on the count and relevant score values obtained by the obfuscated IP address and the 32-bit fingerprint values. These fingerprint values are in the form of vectors. Group information is sent to all the nodes and the node which matches the group information sent responds to that host and the data is sent to that corresponding nodes. Markov models are used to create groups of hosts having similar network behavior [6]. In order to enforce anonymity, the real IP address of each network flow is substituted by its group ID being released. This group ID is a unique value and hence it eliminates the redundant or duplicate value of real IP addresses. However, there is neither experimental evidence nor a formal guarantee that, with this statistically driven approach, an adversary applying available domain knowledge cannot re-identify hosts by their fingerprint.

C. Obfuscation of sensitive data in network flows

In network flow a sensitive data is transmitted from source and destination, here we obfuscate the source and destination IP address as a fingerprint. Using fingerprint we transmit a data from source to router. Router sends the data to the entire host IDs. In every host, the host ID is obtained by many-to-many mapping the fingerprint in respective group. Data is sent to the nodes having similar fingerprint. The quality of the data is not compromised. We send text as packets [5] and [7]. We even send image via the flows and the quality of the image remains the same.

III. RATTLE ADVERSARY IN IP ADDRESS RACE

In general, the fact that two specific hosts A and B exchanged some message may be considered confidential information [2]. Since IP address uniquely identifies its host, we assume that confidential information in network flows as source and destination address along with the obfuscated address and fingerprint values. Even if we remove this confidentiality information from the network flows there is no violation of privacy and security of the data that is been sent through the flows. Removal of only the IP address may disrupt the utility of the data, but when real IP address is mapped with the obfuscated IP address and the fingerprint generated there will never be any disruption in the transfer and utility of the data. Hence the performance in incremental network flows is higher than the performance of single release of network flows.

A. Network Flow Obfuscation

Since every network flow involves IP address of the sender and receiver nodes, we provide a secure transmission by obfuscating the IP address. This obfuscation is performed by mapping the fingerprint value, the real IP address and configurations of every host in the router. The result of this obfuscation is a 32 bit value which is a combination of text and numerical values. Parameters of this obfuscation involves OS, RAM, Processor, Username and real IP address of each node so the values of each node varies and results in unique obfuscated value which will be very difficult for the intruder to hack and get the data from the incremental release of network flows.

B. Fingerprinting

Fingerprinting is matching the flow field’s values to the characteristics of the target environment (OS, RAM, Processor, distance, range and architecture). The typical values of network flows are types of service, number of bytes, and number of packets per flow. The initial fingerprint obtained is a 128 bit alpha-numeric value which is then compressed to 32 bit value using the MD5 algorithm. Our algorithm is extended from many-to-one mapping to many-to-many mapping [3] which enforces a further level of protection. Our extended algorithm also guarantees formal confidentiality.

IV. INCREMENTAL RELEASE

The incremental release has an impact on confidentiality and data quality. Some hosts may disappear when it is dismissed from the network. To avoid this IP address is distributed to all the hosts and is also saved in the router. The same may happen when a host is inactive but this problem is overcome by mapping among the real IP address and group ID, restricting the cardinality of candidate IP addresses for an IP group. Using the proposed adversary model the sets of hosts that has disappeared is found by performing different probabilistic reasoning attacks to associate flows to their hosts. In worst-case if all but one host disappeared the adversary can reconstruct the source and destination of an obfuscated network flow.

The incremental release must also concentrate on the quality of the fingerprint. The fingerprint of some hosts may significantly change with time. The disruption in the fingerprint may happen only when the administrator of the hosts change the values used in the creation of fingerprint. Moreover, when dynamic addressing is used, the mapping between the IP address and the host machine changes with time. Hence, hosts with different fingerprints may be associated under the router, this eliminates the decreasing data quality.
A. Extended Defense Algorithm

The extended algorithm is used to create and update the IP groups such that each group occurs in any of the host. This algorithm is extended for assigning IP address that never appeared in the previous releases to an IP-group and also enforced for incremental network flows. This algorithm takes as input the original set of network flows at the current release. The fingerprint and the IP-group of the current release are calculated in the previous release. This generates the updated set of IP-groups. IP address of current release and the previous release are checked for unique values using the Hilbert index for each new IP address and we sort the IP addresses based on that Hilbert index. We eliminate the IP address that is same as the previous release or establish a new address by calculating the average of the values. In this we merge the group of IP addresses that have the closest Hilbert index, such that each group satisfies the property of unique group creation. These groups are then merged to the hosts. The other IP addresses are simply substituted by their group-ID. This algorithm returns the obfuscated flows, as well as updated IP-groups.

B. Correctness and Computational Complexity

We prove that, at each release the extended algorithm correctly computes a obfuscation function and fingerprint. It follows that the extended algorithm provides the same confidentiality guarantees of single-release network flow obfuscation algorithm. In order to prove the correctness of the extended algorithm, we shall demonstrate that, for each original flow at the release, the algorithm returns a set of obfuscated flows that satisfies the property of the obfuscation function. We first execute the incremental group creation to update the set of IP-groups calculated at the previous release. As explained in IV A, existing IP-groups are updated by inserting IP addresses that appear in the original set but not in the previous release of network flows, in order to ensure that each group contains at least minimum group size IP addresses appearing in original set of network flows at the current release. These groups are merged with ones that satisfies the property of obfuscation, such that their union satisfies the closure property.

As for the single-release obfuscation algorithm, the computational complexity of incremental algorithm is \( O(n \log n) \) in the worst case since the most complex operation is sorting. Note that the incremental algorithm does not consider the network flows of previous releases, it takes as input only the network flows of current release and the IP groups of the previous release.

V. EXPERIMENTAL EVALUATION

The goal of our experiments is to evaluate the role of incremental obfuscation parameters on the utility of obfuscated network flows. Higher the values of indistinguishable values of IP addresses determine stronger confidentiality protection. In order to perform the experiments, we have used a scalable and efficient implementation of the obfuscation technique [7]. We evaluate the data utility of the original and obfuscated flows using a very large set of virtual networks connected to the networks.

- **Traffic diversity:** In order to evaluate the impact of parameter (minimum size of IP address groups), we considered the entropy of source and destination IP addresses distribution in both the original and the obfuscated datasets; indeed, the entropy of network traffic well represents the traffic diversity. We also considered the entropy of multidimensional fields since that metric is largely used in security and networking applications, e.g., anomaly detection [9] and traffic classification [8], [11].

- **Statistical analysis of network flow fields:** In order to evaluate the impact of parameter (minimum size of indistinguishable sets of flows) on the quantitative flows fields, we issued a large amount of statistical queries and measured the error introduced by obfuscation of indistinguishable flows and minimum group size.

- **Network flow analysis:** We evaluated the actual utility of obfuscated flows using the stepping stone detection [10] algorithm, which is widely adopted in real-world network analysis applications but here we apply it in a virtual way.

For the sake of these experiments, we consider a model in which the adversary may have an in-depth knowledge of the network host’s fingerprint. In particular, we assume that the fields of flows are types of service, protocol type, number of packets, number of exchanged bytes, OS, RAM, Processor and Architecture.

A. IP Address Grouping-Fingerprint Grouping

In order to partition hosts in homogeneous groups by fingerprint-based group creation we used the parameters stated in I A: OS, RAM, Processor, Port number, User, Architecture and the interior parameters. For each host, we built the fingerprint vector by computing, on the whole set of flows generated by that host, the mean and the standard deviation of each considered feature. Effectiveness of our grouping method, we measured the homogeneity of hosts of the same group according to their fingerprint vectors by considering the distance and range of the nodes in the hosts.
We also define a intra group distance $d_g$ of a group. The intra group distance values of a group may assume values from 0 (optimal homogeneity) to 1 (maximum diversity).

Figure 1 shows the cumulative distribution function (CDF) of the intra group distance for a single release of 24h. Results show that more than 90% of groups have an intra group distance lower than 0.03, i.e., distance $>0.9$. Thus we can conclude that our grouping method is effective since it produces clusters of nodes and hosts being very similar in terms of traffic behavior.

![CDF of intra group distance](image)

Figure 1. Cumulative distribution functions of intra group distance.

### B. Traffic Diversity

At first, we have studied the impact of our defense on the quality of obfuscated flows using an information-theoretic perspective.

1) **Entropy of IP addresses:**

The first set of experiments was aimed at evaluating the role of parameter minimum group size of incremental obfuscation, i.e., the minimum dimension of IP address groups. This value determines the level of obfuscation of IP addresses. In order to study the impact of such parameter in isolation, we applied the algorithm for fingerprint-based group creation to the original set of flows partitioning the IP addresses in groups of dimension greater than or equal to the group size. Then, we substituted each real IP address in original flows with its corresponding group-ID. An indistinguishable flow has no impact on the creation of IP-groups. We measured how the application of obfuscation affects the entropy of network flows. Since high values of entropy are correlated to high diversity of the IP address distribution, network flows entropy evaluation is widespread in traffic analysis—for instance, for traffic anomaly detection [12], and traffic classification [11]. For example, a distributed DoS attack would determine low entropy on destination IP addresses (many flows are directed to the same host), and high entropy on source IP addresses (many different hosts are performing the attack). We modeled the distribution of IP addresses in flows collected during 1-min-long time-windows, both in the original and in the obfuscated dataset. This is a common setting: for instance, in [12], authors detect traffic anomalies by comparing the entropy in a fine-grained time-window.

Initially, we evaluated how the minimum group size influences the traffic diversity with a single release of the dataset, by applying Algorithm Fingerprint-based group creation. Fig. 2(a) shows the values of entropy, setting the parameter $k$ to 5, 10, and 20. As it can be observed, the values of entropy are inversely correlated to the value of minimum group size. The more IP addresses are grouped together, the less variegated traffic and, consequently, the lower the entropy. However, algorithms for traffic survey rely on fluctuations of the entropy value, not on its absolute value [12]. As it can be seen in Fig. 2(a), the trends and temporal patterns are preserved; hence, obfuscation preserves the utility of network flows for entropy based network flow analysis. We obtained analogous results considering entropy of destination IP address distributions.

Then, we evaluated how the traffic diversity is affected by the incremental release of obfuscated network flows by applying Group creation for incremental releases algorithm. We set the minimum group size value to 10 and used different release frequencies: datasets released every 3, 6, 12, or 24 h. Fig. 2(b) shows the values of source IP address entropy using these release frequencies. Single release refers to the non incremental release of the whole dataset; original flows refer to the non obfuscated network flows. As it can be observed, the temporal trend of entropy is preserved in the obfuscated datasets, independently from the chosen release frequency. For the sake of readability, in Fig. 2(c) we plot the results during a representative peak hour for Internet traffic.

2) **Entropy of Multidimensional Fields:**

We also measured the entropy value of the quantitative fields, as well as the entropy value of the combination of three fields: source IP address, number of bytes, and number of packets. Such values are widely used in many network and security tasks [8], [11]. For the sake of this experiment, we fixed group size to 10 and minimum number of indistinguishable flows to 4 since, as explained, these values provide solid protection against the considered attacks.

We considered two long datasets: 1) the original dataset of network flows, and 2) the correspondent dataset incrementally obfuscated with 3-hour frequency; the latter is the worst case, in terms of data quality (i.e., the case with the highest number of suppressed flows), that we have
considered in our experiments. Results are shown in Figure 2(d)–(f). It can be observed that the entropy values of the incrementally obfuscated dataset essentially keep the same behavior of the corresponding values of the original dataset for both the number of bytes and number of packets fields. Moreover, as shown in Fig. 2(f), the entropy of the multidimensional fields maintains the same trend.

These results indicate that our technique for IP address grouping preserves data utility for algorithms based on information theoretic measures, even when combinations of multiple flow fields are considered. Moreover, the incremental release of obfuscated network flows does not negatively impact the data utility.

\[ k \text{- minimum group size} \]
\[ j \text{- indistinguishable flow fields} \]
\[ dg \text{- intra group distance} \]

The following graphs define the various process carried out during the experimental evaluation of incremental release of network flows that involves the entropy values and the obfuscated IP addresses, group IDs and fingerprint values.

![Graphs of data entropy](image)

**Figure. 2. Evaluation of single and incremental release.** (a) Entropy of source IP address for single release. (b) Entropy of source IP address for incremental release across (group size \( k = 10 \)). (c) Entropy of source IP address for incremental release (\( k = 20 \)). (d) Entropy of byte field across 24 h. (e) Entropy of packet field across 24 h (\( r = 5 \)). (f) Entropy of combined fields (source IP, byte, and packets) across 24 h (\( r = 5 \)).

**C. Statistical Analysis of Flow Fields**

In order to evaluate the impact of indistinguishable flows, we measured the data utility in terms of the precision in answering aggregate queries on Net Flow fields’ values. For the fields having numerical domains (number of bytes and number of packets), we executed the queries considering ranges of different suitability: e.g., “tally the number of flows at minute \( t \) whose number of packets is between 200 and 300.” We contrived these queries appraising each interval of dimension 100 (for the number of bytes) and 5 (for the number of packets), starting from 0 until the maximum dimension of bytes and packets in our dataset. For the fields having non-numerical domains we executed queries about their specific values, whose protocol is TCP.” We accomplished queries for each possible value/range, and for each minute in a 1-h time-window, for a total of about 120,000 queries.

The error rate is calculated and the hosts are grouped and data is sent from one host to the other through the router.
The count and the relevant score values are calculated by the configurations of each system which will be a percentage value. The nodes are grouped based on the range of percentage values. The statistical values are shown in the graphs for single release and error correction rate.

At first, we considered the single release of the dataset by applying Single-release network flow obfuscation algorithm to the whole dataset. Fig. 3 shows the average error rate on the considered network fields based on different values of indistinguishable flows and fixing the value of group size to 10. Results show that the error rate is very low with values of \( k \) less than or equal to 4. On the contrary, values of \( k \) higher than 4 determine a steep rise in the average error. Since, values of \( \Delta = 4 \) ensure strong privacy guarantees, we use these values in the following experiments.

Then, we evaluated the impact of incremental obfuscation on the data quality by applying Incremental network flow obfuscation algorithm. Fig. 4(a)–(c) reports the query error rate based on different frequencies of releases: every 3, 6, 12, or 24 h. Sing. Rel refers to the non-incremental release; an original flow refers to the non-obfuscated network flows.

For the sake of readability, we plot only a portion of the graphs related to the queries on byte and packet fields and the average errors for the queries on TOS, flag, and protocol fields. Fig. 4(a) shows a detail of the error rate on the byte field limited to the range. The error is very low (less than 2%) and is essentially independent from the frequency of incremental releases.

We obtained similar results considering queries on the packet field [Fig. 4(b)]. In order to evaluate the data quality on TOS, flag and protocol fields, in Fig. 4(c) we plot the average errors obtained by the queries on all the possible values that each field may assume. Similarly to the queries on byte and packet fields, the error rates are very small (less than 2%) and are only slightly affected by the release frequency. These results confirm that: 1) obfuscation preserves the data quality for statistical network data analysis; and 2) the incremental obfuscation of network flows does not negatively impact the data quality.

Figure 4. Query evaluations over bytes, packets, TOS, flags, and protocol for the incremental releases.
D. Network flow analysis

We have evaluated the utility of obfuscated network flows for real-world network analysis. We have chosen the stepping stone detection algorithm [10] since it is a representative and widely adopted network flows analysis tool. Moreover, it is based on the combined and temporally correlated analysis of multiple flows, rather than on the aggregate analysis of single flows. By definition, a stepping stone occurs when a host A uses a chain of intermediate hosts to communicate with another host B. In many cases, stepping stones are used by adversaries to hide their identity by using compromised hosts to perpetrate their attacks.

We have implemented the algorithm for stepping stone detection proposed in [10]. The algorithm is based on the observation that couples of flows involved in a stepping stone path often pass simultaneously from an idle state (i.e., absence of exchanged messages for a relatively long period of time) to an active state. A timeout threshold is used to determine the idle state of a couple of flows, and a time-window interval is used to determine whether idle-to-active transitions of a couple of flows are temporally correlated. The algorithm considers as stepping stones those couples of flows whose number of temporally correlated idle-to-active transitions exceeds a given support threshold.

We compared the results when using as input the original flows and the incrementally obfuscated flows, respectively. We used 3-h release frequency of obfuscation an 6hour datasets of flows. The size of that dataset is comparable to the one of those used in [10]. We used the same thresholds used in [13] and [10] for idle state timeout (0.5 s) and idle-to-active transition (20 ms).

The used parameters of obfuscation are $k=4$ and $\ell=4$. Results are shown in Fig. 5. The number of stepping stones recognized using obfuscated flows is slightly lower than using original flows. This can be explained by the fact that, in some cases, the idle-to-active state transitions of some obfuscated flows are not recognized due to IP address grouping. However, the number of stepping stones based on support threshold values follows the same trend with both obfuscated and original flows; hence, we claim that these results show the utility of obfuscated data for real-world network flow analysis. In order to obtain more accurate results, slightly lower values of support should be used with obfuscated flows; this will counter balance the effect of IP grouping on the detection of idle-to-active state transitions.

VI. CONCLUSION AND FUTURE SCOPE

In this paper, we addressed the challenging research issue of network flow obfuscation. We have identified the threats posed by the incremental release of network flow, we have proposed a novel algorithm to enforce obfuscation to incremental releases, and we have formally proved the confidentiality guarantees provided by the new extended algorithm. We have experimentally evaluated our technique with a very large set of virtual incremental network flows. Results showed that our technique preserves the utility of network flows for different network analysis tasks.

Future research directions include the extension of our formal model and defense technique to different adversary models. In particular, we aim at addressing the case in which an adversary has external knowledge about the temporal communication pattern of specific hosts and may use this knowledge to re-identify IP addresses in the observed history of obfuscated flows.
REFERENCES


