2018 IEEE 38th International Conference on Distributed Computing Systems

Chaff Allocation and Performance for Network Traffic Obfuscation

Ertugrul N. Ciftcioglu, Rommie L. Hardy, Kevin S. Chan, Lisa M. Scott, Diego F. M. Oliveira and Gunjan Verma
U.S. Army Research Lab, Adelphi, MD 20783
Email: {enciftcioglu, diegofregolente}@gmail.com, {rommie.l.hardy, kevin.s.chan, lisa.m.scott92, gunjan.verma}.civ@mail.mil

***Abstract*—This work considers performance analysis of chaff-based traffic obfuscation against a passive adversary aiming to obtain contextual information, e.g. such as the protocol being used. The obfuscation could be either in terms of *chaff bytes* which are dummy bytes appended to packets of the intended traffic stream, or *chaff packets* which are dummy packets again inserted in specific intervals of the original packet stream. Despite consisting of dummy bytes, chaff deployment still results in additional resource consumption and potential drawbacks, and hence has to be deployed in a controlled manner. We first define notions of *vulnerability* of traffic patterns in terms of contextual privacy. Next, we fix the adversary and focus on optimal allocation of the chaff resources among the traffic to be obfuscated. For adversaries which perform statistical characterization based on packet sizes and interarrival times, we derive chaff placement algorithms based on the waterfilling algorithm commonly used in the field of information theory. We apply our derived algorithms to representative real-world scenarios to obfuscate certain applications vulnerable to contextual privacy leakage.**

***Index Terms*—Contextual privacy, Resource allocation, Traffic analysis, Obfuscation**

I. Introduction

Despite advances of cryptography in securing information content, sophisticated adversaries might still obtain important contextual information such as the protocol being used. In this paper, we consider chaff-based traffic obfuscation as a means to thwart passive adversary eavesdropping on communications. We consider two main obfuscation approaches: *chaff bytes* where dummy bytes are appended to packets in the intended traffic stream, or *chaff packets* where dummy packets are inserted in specific intervals of the original packet stream.

The goal of network obfuscation is to decrease the amount of information that can be gathered by performing traffic analysis on an information flow by an adversary that is eavesdropping on a network link.

In terms of related work, network traffic can be observed from various levels of the network stack, each providing different detail about activity on networked communications. One can observe network traffic on the physical layer, capturing network packet traffic, observing types of network protocol activity and also user behavior activity. [1] uses convex optimization techniques to show how to optimally modify VoIP and web packets in real-time to reduce the accuracy of a variety of traffic classifiers. [2] also focus on privacy through padding, but their focus is solely on packet sizes. We propose this form of traffic morphing to change the network chaff to provide anonymity of network traffic. Our work focuses on inserting chaff, but is not limited to adversarial analysis based on packet lengths. [3] provides a survey on machine learning techniques that classify various types of IP traffic. Recently, we considered chaff-based network obfuscation when an operator and adversary might both adapt their limited resources with a game-theoretic treatment [4].

Research was sponsored by the Army Research Laboratory and was accomplished under Cooperative Agreement Number W911NF-16-2-0014 (ARL-ORAU Research Associate Program). The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation here on.

II. System Model

1) Network and Traffic Model: We assume a general scenario where communication parties are interested in missions or tasks, which can be either two-way communication between parties, or query responses between users of different hierarchy, as command-and-control centers in a military wireless setting [5]. Depending on the specific task, parties may choose different applications or modalities (include but not limited to VoIP, http, video and/or text). We index the possible communication modalities $i \in \{1, \ldots, M\}$, and assume that each communication modality i is associated with a traffic pattern τ_i. Each τ_i is characterized by either a time series, such as ON-OFF traffic activity profiles, or statistics of its features as packet size distribution or inter-arrival time distributions.

We assume that communication is encrypted to secure information content. Moreover, in order to provide confidentiality regarding the task, the network operator wants to provide traffic obfuscation.

2) Adversarial Model: Network adversaries are present in tactical wireless networks for multiple purposes, and may be active (jammers) or passive (eavesdroppers), which ranges from eavesdropping physical layer to deep packet inspection.

Here, we assume a passive adversary which performs physical layer eavesdropping where the adversary can listen on a link and determine link activity, but is not able to decrypt the information content. On the other hand, the adversary is still able to observe presence of activity, and might be further able to operate on the network layer and extract information regarding the traffic as packet sizes and/or interarrival time. We assume that the adversary possesses knowledge on the possible traffic types that the communication parties may use. With the adversary possessing a database corresponding to statistical features of each traffic type and pattern, it may be able to identify the specific traffic type used by comparing the network outputs to the database of traffic types $\tau_1, \tau_2, \ldots$.

2575-8411/18/$31.00 ©2018 IEEE
DOI 10.1109/ICDCS.2018.00170

Nevertheless, in our scenario the adversary is aware that the network operator may perform network obfuscation, particularly *chaff based* as discussed next.

3) Obfuscation/Chaff Approach: We assume that the network operator knows the classification principles of the adversary which it is tackled with, but may not know when or how frequently the adversary is active, which still poses a significant level of uncertainty in defense.

Although the network operator may opt to obfuscate traffic through delaying packets, such methods are *intrusive* in the sense that they affect quality of service from the users' perspective, particularly for delay-sensitive applications and communication modalities. Hence, we consider the scenario when the network operator is *non-intrusive* and only adds redundancy in terms of *chaff*.

For different types of adversaries, different chaff mechanisms will be deployed through either chaff bytes or chaff packets. Note that even though chaff bytes or packets are dummy packets, their transmission and processing still incurs *costs* for the network operator. Accordingly, the amount of chaff resources may be limited, and optimal resource allocation of the chaff resources is also important.

III. Chaff Allocation Strategies

First, we note that chaff-aided obfuscation may only *densify* the traffic pattern. Specifically, it may increase the duration of link activity (due to chaff bytes and/or packets: we envision these techniques to be applicable for cases when eavesdropping occurs on the physical layer) and ultimate packet sizes(due to chaff bytes), or reduce interarrival time (due to chaff packets: we assume these techniques are applicable when network traffic classification is gathering packet statistics). Noting that the adversary knows the possibility of chaff deployment, it is also aware of potential artifacts chaff deployment may cause in the actual *post-chaff* traffic patterns, which might also reflect in empirical time samples or distributions. Consequently, we note that this implies that the necessity of traffic obfuscation at all depends on the intended communication modality, and the traffic pattern τ_i in comparison with other traffic patterns. Depending on the type of adversary decision rule, traffic pattern τ_i may or may not be *vulnerable* for contextual privacy leakage.

Definition 3.1: A traffic pattern τ_i is **vulnerable in time** if it possess at least one time slot t_i' which is only absent in τ_i, i.e. $s_i(t_i') = 0$ and $s_j(t_i') > 0, \forall j \neq i$, or at least one time slot T_i' where the number of packets are less than the other patterns, i.e. $K_i(T_i') \leq K_j(T_i') > 0, \forall j \neq i$.

On the other hand a traffic pattern τ_i is **vulnerable in distribution**, specifically in *packet-size distribution* if the minimum packet size it possess is smaller than the minimum packet sizes of the other traffic patterns (Fig. 1a). Alternatively, τ_i is vulnerable in *inter-arrival time distribution* if the maximum inter-arrival time of τ_i is greater than the maximum inter-arrival times of the traffic patterns (Fig. 1b).

In words, a traffic pattern which is vulnerable has a easily identifiable and distinct signature. Next, let us assume that the intended communication modality results in a vulnerable traffic pattern in at least one dimension as discussed above. To that end, we next focus on the optimal allocation of a given budget of chaff resources.

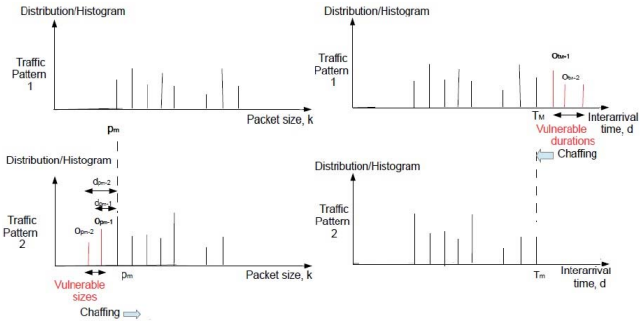

Fig. 1. Examples of a)vulnerable packet sizes b) vulnerable inter-arrival times; Arrow indicates chaff goal.

A. Time-Series Adversary: Packet Count-based Classification

1) Hard decision: If the traffic pattern τ_i has a packet number profile lower than all other traffic profiles at a given time slot t, that is $K_i(t) < K_j(t), \forall j \neq i$, then the adversary may distinguish the presence of traffic pattern i if it samples at slot t. Assuming the adversary takes uniform samples across time regardless of the traffic profiles, against such an adversary, the best strategy is to allocate chaff resources successively to time slots to t^* where $t^* := \arg\min_t(min_{j:j\neq i} K_j(t) - K_i(t))$ to cover the deficient time slots.

2) MSE-based decision: The adversary may acknowledge additional fluctuations among a specific traffic pattern, hence may try to avoid making hard decisions based on deficient time slots. Rather, reminiscent of classification approaches, it may decide presence of a traffic pattern only if a measure of deficiency throughout all samples exceeds a threshold. A widely used metric is based on mean-square-error [6][1]. Against such an adversary, the best strategy for the operator is to follow the *waterfilling* algorithm. In essence, the algorithm starts reducing the deficiency of the time slots with largest deficiency in number of packets, and some time slots may not be aided with chaff depending on the total chaff budget.

B. Statistical Adversary: Vulnerable Outstanding Features

In this section, we assume an adversary that is performing statistical traffic analysis. Given that we have a limited budget for chaff resources, we address the problem of allocation of the available chaff for each type of adversary.

1) Packet-size distribution: We assume that the adversary, among its empirical distribution $\tilde{f}_K(k)$, focuses on vulnerable packet sizes that can occur in case a vulnerable traffic pattern was used. If any such packet sizes, which we term as *outstanding packet sizes* (Fig. 1a) are observed, rather than making a hard decision, it focuses on the MSE of these observations and compares if they exceed a decision threshold K_T.

Against such an adversary, assuming the intended communication modality is vulnerable in packet size distribution, we

[1]We note that several other metrics such as Kullback-Leiber(KL) divergence or Cross-Entropy may also be used, and we leave this study for future work.

focus on how to allocate *chaff bytes* to append to raw packets to minimize the MSE of outstanding distribution. This results in the following problem formulation:

$$\min_{\mathbf{c}} \sum_{k=1}^{K} (o_k - c_k)^2 \quad \text{s.t.} \sum_{k=1}^{K} c_k = C_P, \tag{1}$$

where k is the packet size, o_k is the histogtram/distribution of outstanding packets of size k, c_k is the number of size k packets enlargened to size p_m, and the budget C_P reflects the number of packets that can be aided by chaff.

Theorem 3.1: The optimal chaff byte allocations c_k to packet size k are given by the (reverse) waterfilling algorithm $c_k^* = (o_k - \frac{\lambda}{2})_+$, where λ is determined by the requirement that $\sum_{k=1}^{K} c_k^* = C_P$.

Proof: Omitted due to space limitations. □

The above formulation was based on the assumption that the chaff byte allocation resources were constant regardless of raw packet sizes. In reality, to enlengthen outstanding packets to eliminate their vulnerability, more resources may be needed to obfuscate smaller packets, specifically given by $l_k = p_m - k$. In this case, the solution is given by a waterfilling algorithm which is modified and depends on *outstanding packet sizes in addition to the distribution*, $c_k^* = (o_k - l_k \frac{\lambda}{2})_+ = (o_k - (p_m - k)\frac{\lambda}{2})_+$, where λ such that $\sum_{k=1}^{K} c_k^* l_k = C_B$, with C_B now reflecting chaff budget in terms of bytes.

2) Interarrival-time based distribution: We re-iterate that interarrival times may only decrease with chaff. Hence, if traffic pattern τ_i possesses packet inter-arrival times so large that other traffic patterns do not contain them, the operator must use *chaff packets* to reduce those inter-arrival times. The most efficient way to reduce those times would be to evenly place the chaff packets within the large packet interarrivals.

If the adversary calculates the mean square of outstanding interarrival times and performs decisions based on thresholds, the chaff packet insertion carried out depends on the outstanding statistics. If the traffic pattern includes interarrival times greater than $2T_M$, more than one packet may be required to "hide" the vulnerable interarrival time t_d, specifically $\left\lceil \frac{t_d}{T_M} \right\rceil$. In this case, the formulation becomes $\min_{\mathbf{c}} \sum_{d=1}^{D} (o_d - c_d)^2$ s.t. $\sum_{d=1}^{D} c_d n_d = C$, with the solution following the waterfilling algorithm, $c_d^* = (o_d - n_d \frac{\lambda}{2})_+ = (o_d - \left\lceil \frac{t_d}{T_M} \right\rceil \frac{\lambda}{2})_+$, where λ such that $\sum_{d=1}^{D} c_d^* \left\lceil \frac{t_d}{T_M} \right\rceil = C$.

C. Statistical Adversary: Outstanding and Deficit Features

Next, rather than solely focusing on the strictly vulnerable sizes/intervals as above, we consider an adversary which operates on the whole spectrum of statistics, and evaluates distance metrics of the empirical distribution, and the database of distributions. Against such an adversary the operator also takes into account the deficit sizes/interarrival times (Fig. 2). We consider chaff algorithms which aim to minimize both outstanding and deficit features in the distributions.

Assuming the adversary focuses on packet size distribution, allocation of chaff bytes to append to raw packets to minimize the sum square error of outstanding and deficit distributions results in the following problem formulation:

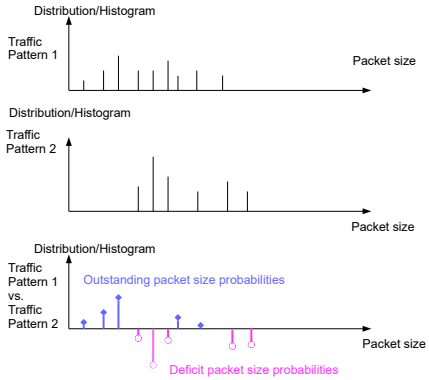

Fig. 2. Outstanding and deficit packet size distributions

$$\min_{\mathbf{c_{i,j}:i<j}} \sum_{k_o \in \mathcal{O}} (o_{k_o} - \sum_{j} c_{k_o,j})^2 + \sum_{k_d \in \mathcal{D}} (d_{k_d} - c_{i,k_d})^2 \tag{2}$$

$$\text{s.t.} \sum_{k_o \in \mathcal{O}} \sum_{k_d \in \mathcal{D}, k_d > k_o} c_{k_o,k_d} = C_P, \tag{3}$$

where k is the packet size, o_{k_o} is the distribution of outstanding packets of size k_o, d_{k_d} is the distribution of deficit packets of size k_d, c_{k_o,k_d} is the number of size k_o packets enlargened to size k_d, and the budget C_P reflects the number of packets that can be aided by chaff.

We note that this problem is significantly more challenging compared with the one in Sec. III-B1 due to various reasons. First, the dimensionality of the problem is increased since even if a source packet is selected to be padded, we have to also determine resultant packet size in contrast to padding to a fixed packet size. Next, the source and target distributions may be such that (i) there are not enough smaller outstanding size packets to fill the deficit size distributions (ii) there are not sufficient large packet size distributions to pad large size outstanding packet sizes to. Consequently, we first study some basic case studies in view to develop algorithms generalizing to larger scenarios.

Case Study 1: Let us consider the simple case with $\mathcal{O} = \{1\}$, $\mathcal{D} = \{2, 3\}$ hence we only have $c_{1,2}$ and $c_{1,3}$ to allocate, with $c_{1,2} + c_{1,3} = C$. The objective function becomes $S =: (o_1 - c_{1,2} - c_{1,3})^2 + (d_2 - c_{1,2})^2 + (d_3 - c_{1,3})^2$.

The Hessian matrix analysis implies convexity, and Lagrangian analysis reveals the solutions to $c_{1,2} = \frac{2o_1 + 2d_2 - 2c_{1,3} - \lambda}{4}$, and $c_{1,3} = \frac{2o_1 + 2d_3 - 2c_{1,2} - \lambda}{4}$. Solving these expressions, we obtain $c_{1,2} = \frac{2o_1 + 4d_2 - 2d_3 - \lambda}{6}$, and $c_{1,3} = \frac{2o_1 + 4d_3 - 2d_2 - \lambda}{6}$. Accordingly, we can define a reverse waterfilling-type solution where $c_{i,j}$ increases with $o_i + d_j$.

Case Study 2: Here, we expand the previous case with $\mathcal{O} = \{1, 3\}$, $\mathcal{D} = \{2, 4\}$ hence we also have $c_{3,4}$ in addition to $c_{1,2}$ and $c_{1,4}$ to allocate, with $c_{1,2} + c_{1,4} + c_{3,4} = C$. The objective function becomes $S =: (o_1 - c_{1,2} - c_{1,4})^2 + (o_3 - c_{3,4})^2 + (d_2 - c_{1,2})^2 + (d_4 - c_{1,4} - c_{3,4})^2$.

Confirming convexity through the Hessian matrix, Lagrangian analysis reveals the solutions to $c_{1,2} = \frac{2o_1 + 2d_2 - 2c_{1,4} - \lambda}{4}$, $c_{1,4} = \frac{2o_1 + 2d_4 - 2c_{1,2} - 2c_{3,4} - \lambda}{4}$, and $c_{3,4} = \frac{2o_3 + 2d_4 - 2c_{1,4} - \lambda}{4}$. Accordingly, we can again approximate the solution with a reverse waterfilling-type solution where $c_{i,j}$ increases with $o_i + d_j$.

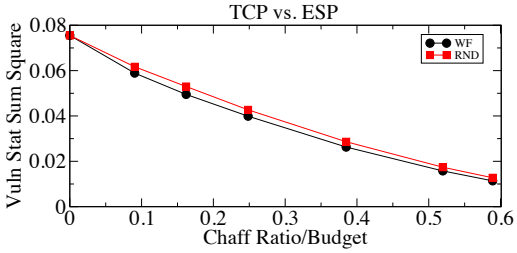

Fig. 3. Chaff allocation of vulnerable sizes for TCP against ESP.

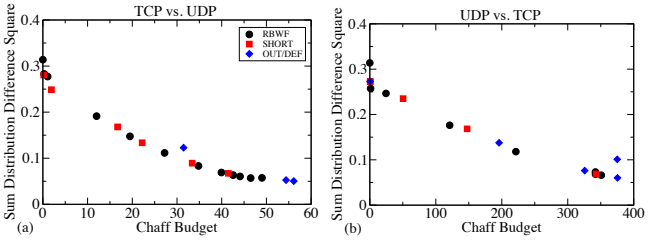

Fig. 4. Budgeted chaff performance for a) TCP vs UDP b) UDP vs TCP

Based on these two case studies, we can derive an algorithm which prioritizes the chaffing from source packet size i to resultant size j ($j > i$) in the form of $c_{i,j} =: (o_i + d_j - \lambda)_+$ with λ such that all $c_{i,j}$ sum to C_P, or $c_{i,j} =: (o_i + d_j - (j - i)\lambda)_+$ with λ such that all $(j - i)c_{i,j}$ sum to the chaff byte budget C_B when padding amounts are taken into account.

IV. Numerical Results

In this section, we provide chaff allocation solutions to a real-world scenario with given applications and traffic pattern statistics characterized in the literature. We have obtained packet size distributions for a variety of protocols as TCP, UDP and ESP [7]. While we omit the plots of these distributions in this paper due to size limitations, we point out that both TCP and UDP have packet sizes starting from size 46, whereas ESP has a minimum packet size of 84. Hence, (according to definitions in Sec. III) TCP and UDP are strictly vulnerable in packet size compared with ESP. In Fig. 3, we observe the improvement in the sum square of vulnerable outstanding distributions metric depending on chaff budget. We observe that even a small fraction of the total chaff budget needed for full obfuscation provides significant reductions in the metric compared with employing no chaff. We also study performance of a random chaff algorithm which probabilistically pads any source packet which is identified to require obfuscation. We note that our waterfilling based approach outperforms the random algorithm.

Note that both TCP and UDP have the same minimum packet size. Yet, we observe the rest of the packet size distributions still have significant deviations, hence apply algorithms studied in Section III-C to obfuscate against an adversary which takes into account the whole packet size spectrum. We next vary the chaff budget and apply our reverse waterfilling approach in Fig. 4a. Moreover, we also study performance of a few other algorithms: (i) An algorithm which prioritizes short paddings for efficiency(agnostic of distributions) (ii) An algorithm which prioritizes padding to reduce outstanding or deficit distributions (agnostic of padding required) (iii) Random padding algorithms which equally distribute among possible padding options. Our numerical results clearly demonstrate that while the reverse waterfilling algorithm can provide a significant improvement in the metric while trading off with budget constraints, along with a more reliable performance, the other algorithms have inherent limitations. The short padding algorithm may be efficient in terms of resources, but can be limited (as this case) by not addressing deficit distributions enough and rather prioritizing the small padding amounts. On the other hand, while the greedy outstanding/deficit distribution addressing algorithm may provide a low metric, it risks spending too much on chaff budget since it does not take resources into account. All in all, our algorithm aims to balance among these different factors.

Finally, we present corresponding results for obfuscating UDP against TCP in Fig. 4b. While we observe a similar trade-off provided by the reverse waterfilling algorithm, one striking difference between Fig. 4a and Fig. 4b is the chaff budget required to achieve similar levels of obfuscation. Due to the relative forms of distributions, lower amount of chaff was sufficient for obfuscating TCP to a desired degree, highlighting the dependency on datasets on obfuscation performance.

V. Conclusion

In this work, we have considered resource allocations for chaff-aided traffic obfuscation. First, we consider an adversary which performs decisions based on time-series of the activity measurements. Next, we focused on adversaries which aim to obtain contextual privacy through statistical traffic analysis. We considered the cases where the adversary may uncover contextual information through packet size and/or interarrival time distributions of the traffic patterns, and derived several resource allocation algorithms to maximize traffic obfuscation given a chaff budget. For a packet size based classifier adversary, we demonstrate that the chaff bytes are allocated to small packets based the waterfilling algorithm, based on outstanding packet size distributions, and the packet size. Alternatively, against eavesdroppers performing analysis based on interarrival times, we derive chaff packet insertion algorithms based on the waterfilling algorithm, which depend on the outstanding distribution of large interarrival times. Our algorithms are flexible and propose a theoretical-based solution to important data-based scenarios for contextual privacy with limited resources.

References

[1] C. V. Wright, S. E. Coull, and F. Monrose, "Traffic morphing: An efficient defense against statistical traffic analysis.," in *NDSS*.

[2] A. Iacovazzi and A. Baiocchi, "Optimum packet length masking," in *Teletraffic Congress (ITC), International*, pp. 1–8, IEEE, 2010.

[3] T. T. Nguyen and G. Armitage, "A survey of techniques for internet traffic classification using machine learning," *Commun. Surveys Tuts.*, vol. 10, pp. 56–76, Oct. 2008.

[4] E. N. Ciftcioglu, R. L. Hardy, L. M. Scott, and K. S. Chan, "Efficient chaff-aided obfuscation in resource constrained environments," in *IEEE MILCOM*, IEEE, 2017.

[5] D. S. Alberts, "Agility, focus, and convergence: The future of command and control," tech. rep., Office of The Assistant Secretary of Defense for Networks and Information Integration Washington DC, 2007.

[6] E. L. Lehmann and G. Casella, *Theory of point estimation*. Springer Science & Business Media, 2006.

[7] "The caida anonymized internet traces 2016 dataset." http://www.caida.org/data/passive/passive_2016_dataset.xml. Accessed: 2018-03-14.