Linking Correlated Network Flows through Packet Timing: a Game-Theoretic Approach

Juan A. Elices
University of New Mexico
Email: jelices@ece.unm.edu

Fernando Pérez-González
Universidad de Vigo
Email: fperez@gts.uvigo.es

***Abstract*—Deciding that two network flows are essentially the same is an important problem in intrusion detection or in tracing anonymous connections. A stepping stone or an anonymity network may try to prevent flow correlation by delaying the packets, introducing chaff traffic, or even splitting the flow in several subflows.**

We introduce a game-theoretic framework for this problem. The framework is used to derive the Nash equilibrium under two different adversary models: the first one, when the adversary is limited to delaying packets, and the second, when the adversary also adds dummy packets and removes packets from the flow. As the optimal decoder is not computationally feasible, we restrict the possible decoder to one that estimates and compensates the attack. Our analysis can be used for understanding the limits of flow correlation based on packet timings under an active attacker.

***Index Terms*—traffic analysis, game theory, flow watermark, network security**

I. Introduction

Network attackers intentionally hide their identity to avoid prosecution. A broadly-used way of achieving this anonymity is relaying the traffic through a chain of compromised hosts called stepping stones [1]. Intrusion detection and tracing back an attack require deciding that two flows are correlated. Linking network flows can also be used to compromise low-latency anonymous networks, such as Tor.

Two approaches exist for finding correlated flows: passive analysis and active watermarks. They differ in whether the flow is modified or not. In general, an active watermark needs shorter sequences but at the expense of being detectable [2].

An adversary (AD), for instance a stepping stone or an anonymous network, may take countermeasures to prevent the correlation such as introducing delays to packets or adding dummy packets to the flow, or even more drastic measures, e.g. dividing the flow into different subflows each taking different paths. To the best of our knowledge, in previous works, only [3] and [4] consider an AD, and in both the AD is limited to delaying packets.

This paper studies the limits of traffic analysis, passive or active, in an adversarial environment. The most natural solution to avoid the loop of proposing an attack and creating an *ad-hoc* solution is to cast the problem into a game-theoretic framework and look for the optimum strategies that the players, traffic analyst (TA) and AD, should adopt. A game-theoretic framework similar to the proposed one has been used in other contexts such as Information Hiding [5] or Source Identification [6].

The rest of the paper is organized as follows: in Section II we introduce the notation, together with some basic concepts of game theory. Section III presents a rigorous denition of the traffic analysis game. Section IV solves the problem under an AD model that only introduces delays. Section V deals with a stronger AD problem that also adds chaff traffic and divides the flow. Conclusions are presented in Section VI.

II. Notation and Basic Concepts

We use the following notation. Random variables are denoted by capital letters (e.g., X), and their individual realizations by lower case letters (e.g., x). The domains over which random variables are defined are denoted by script letters (e.g., $\mathcal{X}$). Sequences of n random variables are denoted with X^n if they have random nature or by x^n if they are deterministic. X_i or x_i indicate the ith element of X^n or x^n, respectively. The probability distribution function (pdf) of a random variable X is denoted by $f_X(x)$, $x \in \mathcal{X}$. We use the same notation to refer to pdf of sequences, i.e. $f_{X^n}(x^n)$, $x^n \in \mathcal{X}^n$. When no confusion is possible, we drop the subscript in order to simplify the notation. We denote with Δ the difference operation of a sequence, i.e $\Delta x^n = \{x_2 - x_1, \ldots, x_n - x_{n-1}\}$ and with $\Delta x_i = x_{i+1} - x_i$ the ith element of this sequence.

A. Game Theory

Game theory is the mathematical study of interaction among independent, self-interested agents. Formally, a two-player game is defined as a quadruple $G(A_1, A_2, u_1, u_2)$, where $A_i = \{a_{i,1}, \ldots a_{i,n_i}\}$ are the actions available to the ith player, and $u_i : A_1 \times A_2 \mapsto \mathbb{R}$, $i = 1, 2$ is the utility function or payoff of the game for player i. An action profile is the double $a \in A_1 \times A_2$. We are interested in zero-sum games, where $u_1(a) + u_2(a) = 0, \forall a \in A_1 \times A_2$, which means that the gain (or loss) of utility of player 1 is exactly balanced by the losses (or gains) of the utility of player 2. In this case, we can simplify the game notation to a triplet $G(A_1, A_2, u)$, where $u = u_1 = -u_2$.

We say that an action profile $(a_{1,i^*}; a_{2,j^*})$ represents a Nash equilibrium (NE) if $u(a_{1,i^*}; a_{2,j^*}) \geq u(a_{1,i}; a_{2,j^*}) \quad \forall a_{1,i} \in A_1$ and $u(a_{1,i^*}; a_{2,j^*}) \leq u(a_{1,i^*}; a_{2,j}) \quad \forall a_{2,j} \in A_2$, intuitively, this means that none of the players can improve its

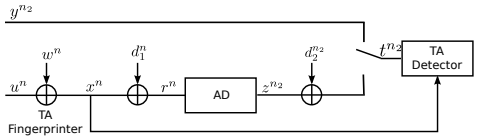


Fig. 1: Model of the TAG game

utility by modifying his strategy assuming the other player does not change his own.

Games can be classified in simultaneous games, where both players move unaware of the other player action, and sequential games, where later players have some knowledge about earlier actions. In sequential games an action profile is a subgame perfect equilibrium (SPE) if it represents a NE of every subgame of the original game. Therefore, a SPE is a refinement of the NE that eliminates non-credible threats.

III. Traffic Analysis Game

The Traffic Analysis game (TAG) is represented in Figure 1. In this game, there are two players: TA and AD.

The task of the TA is to create a test to accept or reject the hypothesis that a flow w^m is indeed the same flow as a known one, x^n, that can be the original flow (passive traffic analysis) or the output of a watermarker. In this paper we do not consider the problem of how to create the watermark. Furthermore, all the simulations are done in a passive analysis situation.

On the other side, the goal of the AD is to modify the flow in such a way that the TA decides that this sequence is not a modified version of x^n. In order to do this, AD can delay any packet at most A_{max} seconds. We denote by a^n the sequence of delays added by AD. In Section V, we propose a more powerful attacker model that can add up to $P_A \cdot n$ dummy packets and remove $P_L \cdot n$ packets, assuming they are sent through a different path, hence P_A is the maximum ratio between chaff traffic and original traffic and P_L is the maximum probability of a packet being removed.

As shown in Figure 1 the flow suffers two additional delays D_1^n and D_2^m that are due to the network between where x^n is measured (TA-1) and AD, and between AD and where w^m is measured (TA-2), respectively. We represent by D the delay that suffers a packet from TA-1 to TA-2, i.e. $D = D_1 + D_2$. Note that ΔD is the packet delay variation (PDV), also called jitter.

Let Y^m represent flows without any relation to x^n, but from the same application. We assume that $f_{\Delta Y}(\Delta y)$ is known by both players, and define the hypotheses: H_0: w^m is not a modified version of x^n, and H_1: w^m is a modified version of x^n.

We define the TAG game as follows:

Definition 1. The $TAG(A_{TA}; A_{AD}; u)$ is a simultaneous, zero-sum game played by the FA and the AD, where

- The set of actions the FA can choose from, i.e. A_{TA}, is the set of acceptance regions for H_1 for which the probability of false positive (i.e., accepting H_1 when H_0 is true) is below a certain threshold η. Note that the acceptance region can be different for each sequence. Therefore, $A_{TA} = \{\Lambda_1(x^n) : Pr(y^m \in \Lambda_1(x^n)) < \eta\}$.
- The set of possible attacks that the AD can choose from, i.e. A_{AD}. It depends on the assumed adversary model.
- The utility function is the probability of detection, (i.e., accepting H_1 when H_1 is true), namely: $u(\Lambda_1(x^n), A_{AD}) = Pr(w^m | H_1 \in \Lambda_1(x^n))$.

As the TA can choose its action $\Lambda_1(x^n)$ after knowing w^n, the solution to the game is

$$u = \min_{A_{AD}} \max_{A_{TA}} u(A_{TA}, A_{AD}). \tag{1}$$

IV. Delaying Adversary model

This section derives the detector under an AD that is limited to delaying the packets constrained to A_{max} seconds. Under this condition, there exists a one-to-one correspondence between x^n and w^m, consequently $m = n$.

We confine the detector to those based on first-order statistics of the inter-packet delay (IPD), i.e., Δx and Δw, for feasibility reasons. Among those, the FA constructs the optimal detector which, according to Neyman-Pearson Lemma, is the likelihood ratio test

$$\Lambda_{AD}(w^n, x^n, f_{\hat{A}^n|X^n}) = \frac{\ell(H_1|\Delta w^n, \Delta x^n)}{\ell(H_0|\Delta w^n, \Delta x^n)} = \sum_{i=1}^{n-1} \log\left(\iint_{\hat{A}^2} \frac{f_{\Delta D}(\Delta r_i - \Delta \hat{a}_i) f_{\hat{A}_{i,i+1}|X_{i,i+1}}(\hat{a}_i, \hat{a}_{i+1}|x_{i,i+1})}{f_{\Delta Y}(\Delta w_i)} d\hat{a}_i d\hat{a}_{i+1} \right), \tag{2}$$

where r^n is $w^n - x^n$, ℓ represents the log-likelihood function, and $f_{\hat{A}^n}$ the estimated joint pdf for A^n. Hence, the test chooses H_1 when $\Lambda_{AD}(w^n, x^n, f_{\hat{A}^n|X^n}) \geq \epsilon$ where ϵ is a threshold that we fix to achieve a certain probability of false positive. In this case, the TA actions are all the possible joint pdfs for $\hat{A}^n$ given x^n, i.e. $A_{TA} = f_{\hat{A}^n|X^n}(a^n|x^n)$. Unfortunately, calculating the joint pdf that maximizes Λ_{AD} is a computationally intractable problem.

Therefore, we restrict the detector, at the expense of losing the optimality, to one that estimates the sequence $\hat{a}^n$ instead of its joint pdf. In this case, the likelihood ratio test is

$$\Lambda_{AD}(w^n, x^n, \hat{a}^n) = \sum_{i=1}^{n-1} \log\left(\frac{f_{\Delta D}(\Delta_i(r^n - \hat{a}^n))}{f_{\Delta Y}(\Delta_i w^n)} \right), \tag{3}$$

and the TAG game is modified as follows:

$$A_{TA} = \{\hat{a}^n : \forall i \in [0, n],\ 0 \leq \hat{a}_i \leq A_{max}\ \} \tag{4}$$

$$A_{AD} = \{a^n : \forall i \in [0, n],\ 0 \leq a_i \leq A_{max}\} \tag{5}$$

$$u(\hat{a}^n, a^n) = Pr(\Lambda_{AD}(x^n + a^n + D^n, x^n, \hat{a}^n) > \epsilon), \tag{6}$$

$$\text{where } Pr(\Lambda_{AD}(Y^n, x^n, \hat{a}^n) \leq \epsilon) = \eta. \tag{7}$$

From (1), we obtain the solution to the game

$$A_{TA} = \arg\max_{\hat{a}} \Lambda_{AD}(w^n, x^n, \hat{a}^n) \tag{8}$$

$$A_{AD} = \arg\min_{a^n} Pr(\max_{\hat{a}^n} \Lambda_{AD}(x^n + a^n + D^n, x^n, \hat{a}^n) > \epsilon). \tag{9}$$

As the AD must decide its action in real time and (9) is computationally expensive, we approximate it by

$$A_{AD} \approx \arg\min_{a^n} \sum_{i=1}^{n-1} \log f_{\Delta Y}(\Delta x_i + \Delta a_i). \qquad (10)$$

which is a good approximation when ΔD has zero-mean and its variance is much smaller than ΔY (as it is the case in practice). Note that under this approximation AD is maximizing the likelihood of $x^n + a^n$ coming from y^n, i.e., making the sequence as typical as possible.

A. Performance

In this section we construct a simulator and present the scenarios we use in the remaining of the paper. Afterwards, we compare the detector performance between the AD choosing its action optimally and an AD that selects its attack randomly.

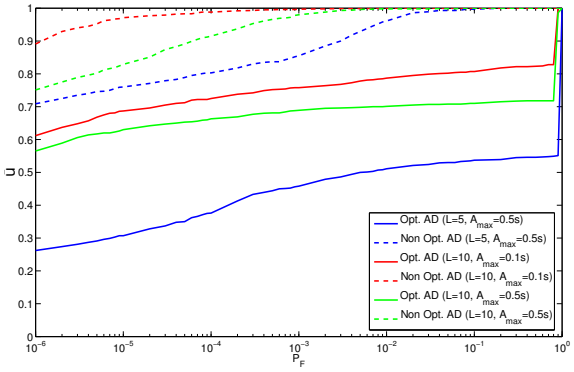
1) Simulator and Scenarios: Simulations are carried out in the following way. First, we randomly generate one sequence, X^n. Then we generate 10^6 sequences Y^n and we calculate ϵ such that (7) holds. To deal with the probabilistic nature of the delay, we repeat the following 50 times: we apply D_1^n, next, we implement the adversary action according to (10). Subsequently, we introduce another delay D_2^n. Finally, we calculate the utility for this X^n using (6). We repeat the whole process for 1000 different X^n, the plots show the average utility for the 1000 sequences, $\bar{u}$. Recall that the utility function is the probability of detection. In order to show that AD is acting rationally we compare it with an AD that introduces delays randomly, i.e., a^n is an independent and identically distributed (i.i.d.) sequence uniformly distributed between 0 and A_{max}.

Note that (3) needs an estimation of $f_{\Delta D}$ and $f_{\Delta Y}$, to this end we apply kernel smoothing techniques [7]. As it is customary, we separate the data into two subsets: training, to obtain the pdfs, and test, used in the simulator, using 50% of the samples for each.

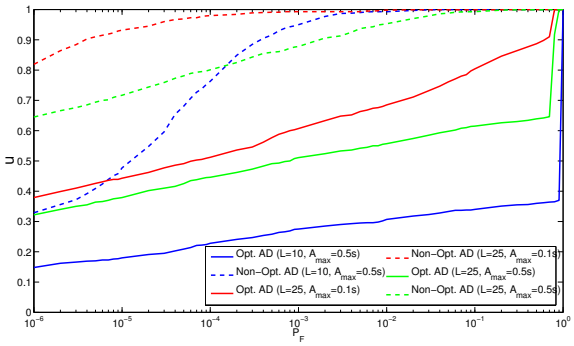
Scenario A represents a stepping stone that forwards SSH traffic inside the Amazon Web Services network. TA-1, AD and TA-2 (cf. Figure 1) are EC2 instances located in Virginia, Oregon and California, respectively. We use the IPDs from 8746 replayed SSH connection captures with 64 million packets from [8] and [9]. The simulated delays correspond to Scenario 10 from [10].

Scenario B simulates a web page accessed from Tor network whose real origin is to be found. In it, TA-1 corresponds to the web server, AD to the Tor entry relay, and TA-2 to the client. For instance, this case can correspond to a company in whose forum an anonymous insulting post has been placed using Tor and it is to be known whether the source comes from an employee within the company. We use the IPDs of 113690 replayed HTTP connections that sum around 139 million packets taken from the same repositories. The delays correspond to the measurements of Scenario 11 from [10].

Results are depicted in Figures 2a and 2b for Scenario A and B, respectively. We see that if AD chooses the delays optimally, the impact of the attack is much larger than using random delays.



(a) Scenario 1



(b) Scenario 2

Fig. 2: Performance under a delaying AD model.

V. Chaff traffic and flow splitting

Hitherto, we have assumed that there exists a one-to-one relation between the flows at the creator and detector; i.e., no packets are added or removed. In this section, we make a robust test to repacketization and study the game when, in addition to random delays, AD can also insert chaff traffic and divide the flow, and TA-2 observes only one subflow.

To deal with packet addition and removal, TA matches each packet from x^n with the most likely from w^m, $m \leq (1 + P_A)n$, denoting this as $i \to j$, as follows: $|x_i - (w_j - \rho)| < |x_i - (w_k - \rho)|$, $\forall k \neq j$, and to avoid considering it lost $|x_i - (w_j - \rho)| > \gamma$, where ρ is a synchronization constant obtained through an exhaustive search (auto-synchronization property), and γ is a threshold for which a packet is considered lost. To prevent that two packets i_1 and i_2 are matched to the same j packet, the set of possible matching candidates for the later packet is reduced to the non-matched ones. Also γ should be big enough so that the probability that a packet is considered lost when it actually is not is very small, for instance our simulation uses $\gamma = Q_{|\Delta D|,0.999} + A_{max}$ where $Q_{|\Delta D|,0.999}$ means the 0.999-quantile of the absolute value of ΔD. This procedure outputs two subsequences, w^{n_2} and x^{n_2}, where $n_2 \leq n$.

After this matching we apply a modification to Λ_{AD} in (3)

that takes into account the new sequence length as follows:

$$\Lambda_{AD}(w^{n_2}, x^{n_2}, \hat{a}^{n_2}) = (n - n_2)\log(P_L) + \sum_{i=1}^{n_2-1} \log\left(P_L + (1 - P_L)\frac{f_{\Delta D}(\Delta w_i - \Delta x_i - \Delta \hat{a}_i)}{f_{\Delta Y}(\Delta w_i)}\right), \quad (11)$$

We propose a new AD model that apart from adding delays as previously, it can remove up to $P_L \cdot n$ packets and add at most $P_A \cdot n$ dummy packets. Formally, the available actions for the adversary are:

$$A_{AD} = \{a^n \times l^n \times c^{n_A} : 0 \leq a_i \leq A_{max}, l_i = \{0, 1\} \sum_{i=1}^{n} \frac{l_i}{n} \leq P_L,\ \frac{n_A}{n} \leq P_A,\ m = n + n_A - \sum_{i=1}^{n} l_i\}, \quad (12)$$

where l^n is a binary sequence that represents which packets are removed from the flow, and c^{n_A} is the sequence with the timing of the n_A introduced dummy packets.

A. Results

To evaluate the proposed robust algorithm, we modify the simulator using the matching process as follows: longer sequences, i.e. $Y^{n+\lfloor P_A n \rfloor}$, are needed to calculate ϵ. Afterwards, AD chooses its action such that (11) is minimized. We compare the optimal AD with two non-optimal ADs: a) NO1 AD chooses its attack randomly, i.e a^n is an i.i.d. sequence uniformly distributed between 0 and A_{max}, l^n is a binary random sequence that contains $\lfloor nP_L \rfloor$ '1's, and c^{n_A} is an i.i.d. sequence uniformly distributed between the timing of the first packet and the timing of the last packet. The b) NO2 AD selects a^n to minimize (11) (optimally) but chooses l^n and c^{n_A} identically as the previous AD.

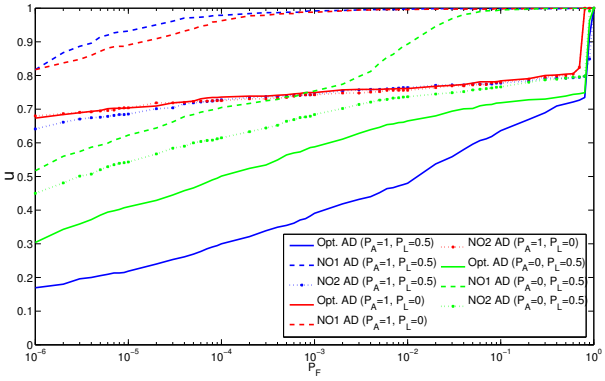
Results are depicted in Figures 3a and 3b. We see that the chaff traffic location when there are no losses does not make a big impact (the Opt-AD and NO-2 AD lines difference is very small). The rest of the data confirm that AD can harm the TA considerably more when deciding its actions logically than acting randomly.

VI. Conclusions

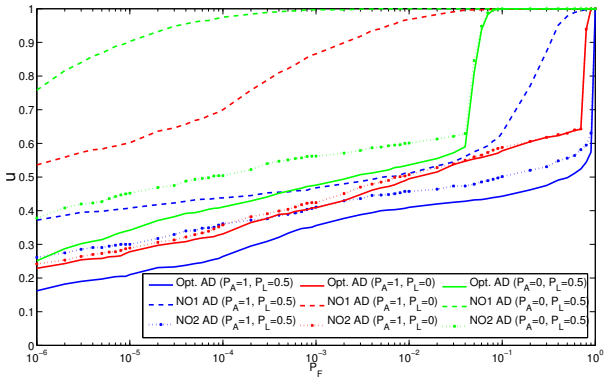
Using a game theoretic framework we have analyzed the flow linking problem. This problem consists in deciding if two flows are linked, having an adversary in the middle that tries to impair the correlation. Using this framework we obtain the optimal choices if both players act rationally. However, in a real implementation finding these optimal choices is not computationally feasible. This made us restrict the decoder to those that estimate the attack. We apply it to two different adversaries, one that only delays packets and another that can also add and drop packets from the flow. While in the present work the original flow is given, a future work will allow this flow to be modified through a watermark.

Acknowledgment

Research supported by Iberdrola Foundation through the Prince of Asturias Endowed Chair in Information Science and Related Technologies.



(a) Scenario 1, L=20, $A_{max} = 500$ms



(b) Scenario 2, L=30, $A_{max} = 500$ms

Fig. 3: Performance under chaff packets and removed packets.

References

[1] S. Staniford-Chen and L. Heberlein, "Holding intruders accountable on the internet," in *Security and Privacy, 1995. Proceedings., 1995 IEEE Symposium on*, may 1995, pp. 39 –49.

[2] X. Luo, P. Zhou, J. Zhang, R. Perdisci, W. Lee, and R. K. C. Chang, "Exposing invisible timing-based traffic watermarks with BACKLIT," in *Proceedings of the 27th Annual Computer Security Applications Conference*. ACM, 2011, pp. 197–206.

[3] D. L. Donoho, A. G. Flesia, U. Shankar, V. Paxson, J. Coit, and S. Staniford, "Multiscale stepping-stone detection: detecting pairs of jittered interactive streams by exploiting maximum tolerable delay," in *Proceedings of the 5th international conference on Recent advances in intrusion detection*, ser. RAID'02. Berlin, Heidelberg: Springer-Verlag, 2002, pp. 17–35.

[4] A. Blum, D. Song, and S. Venkataraman, "Detection of interactive stepping stones: Algorithms and confidence bounds," in *Recent Advances in Intrusion Detection*, ser. Lecture Notes in Computer Science. Springer Berlin / Heidelberg, 2004, vol. 3224, pp. 258–277.

[5] P. Moulin and J. O'Sullivan, "Information-theoretic analysis of information hiding," *Information Theory, IEEE Transactions on*, vol. 49, no. 3, pp. 563–593, 2003.

[6] M. Barni and B. Tondi, "The source identification game: An information-theoretic perspective," *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 3, pp. 450–463, 2013.

[7] A. W. Bowman and A. Azzalini, *Applied Smoothing Techniques for Data Analysis: The Kernel Approach with S-Plus Illustrations (Oxford Statistical Science Series)*. Oxford University Press, USA, Nov. 1997.

[8] D. Kotz, T. Henderson, I. Abyzov, and J. Yeo, "CRAWDAD trace set dartmouth/campus/tcpdump (v. 2004-11-09)," http://crawdad.cs.dartmouth.edu/dartmouth/campus/tcpdump, Nov. 2004.

[9] R. R. R. Barbosa, R. Sadre, A. Pras, and R. van de Meent, "Simpleweb/university of twente traffic traces data repository," Centre for Telematics and Information Technology University of Twente, Enschede, Technical Report, 2010.

[10] J. A. Elices and F. Pérez-González, "Measures to model delays on internet," http://www.unm.edu/~elices/captures.html, Jan. 2013.