Slotted Packet Counting Attacks on Anonymity Protocols

Volker Fusenig **Eugen Staab** **Uli Sorger** **Thomas Engel**

University of Luxembourg
Faculty of Sciences, Technology and Communication
6, rue Richard Coudenhove-Kalergi
L-1359 Luxembourg
Email: {Volker.Fusenig, Eugen.Staab, Ulrich.Sorger, Thomas.Engel}@uni.lu

Abstract

In this paper we present a slotted packet counting attack against anonymity protocols. Common packet counting attacks make strong assumptions on the setup and can easily lead to wrong conclusions, as we will show in our work. To overcome these limitations, we account for the variation of traffic load over time. We use correlation to express the relation between sender and receiver nodes. Our attack is applicable to many anonymity protocols. It assumes a passive attacker and works with partial knowledge of the network traffic.

Keywords: Anonymity, Unlinkability, Traffic Analysis, Packet Counting.

1 Introduction

Privacy protection and anonymity in the Internet attract more and more attention because of the public discussion on the storage of personal information by many service providers. At the moment there exist several protocols for anonymous communication in the Internet that are ready to use, such as Tor (Dingledine et al. 2004) and JAP (Berthold et al. 2000). These protocols share a common idea: The traffic is sent over a cascade of so called "onion routers" (or "mixes" respectively), which obfuscate the routes the packets will take and so make it hard for an attacker to trace the route of a certain packet. But although an attacker both cannot easily trace a packet and cannot extract any information about the communication partners from the content of the packet, he still can perform attacks on these protocols, e.g. by means of *traffic analysis*.

In this paper we present a traffic analysis attack which is a further development of the idea of *packet counting*. The first packet counting attacks were proposed in (Back et al. 2001, Raymond 2001). In their attacks, a passive attacker counts the packets entering and leaving a mix on different links; the attacker then tries to find coherences in the number of packets entering on an input link and a similar number of packets leaving on an output link. To some extent, this enables an attacker to trace packets. However, as Serjantov and Sewell (Serjantov & Sewell 2003) pointed out, this attack is restricted to scenarios where it can be assumed that connections that enter and leave an anonymity system are *lone*, i.e. two connections going into an anonymity system never leave the system on a shared link and vice versa. Because of this strong assumption the attack is not applicable in most settings.

In our work, we present a packet counting attack that does not require this assumption. In our scenario, an attacker first counts packets sent and received by the users of an anonymity system in different time slots. Then the attacker identifies for sender/receiver-pairs the *correlation* of changes in traffic load over time; in this manner, he can probabilistically reveal communication links. The only assumption we need to make is that the attacker can count the respective numbers of sent and received packets that enter and leave an anonymity system (as is also required for common packet counting).

An application of such an attack might be crime prevention and detection. Using this attack it is possible to link the IP addresses of the communication partners. With a directive of the European Union (EU Commission 2006), that demands the logging of electronic communication within the EU member states as from 2009, it also becomes possible to map the IP addresses to the identities of the users.

The remainder of the paper is organized as follows. In Section 3 we give an overview of existing anonymity protocols and known attacks on these protocols. In Section 4 we show that packet counting as described in (Back et al. 2001) and (Raymond 2001) is not very effective and explain why this is the case. We present an improved slotted packet counting attack in Section 5 and apply it to real network traffic traces in Section 6. Section 7 gives an overview on related work and Section 8 concludes the work and points out future work.

2 Attacker Model

In this paper we assume that the attacker passively observes the network communication. This means that the attacker knows who sends packets and who receives packets at any time. This is a widely used attacker model for the evaluation of traffic analysis attacks (see e.g. (Serjantov & Sewell 2003, Danezis et al. 2007)). For the analysis of the attacks we are only interested in the packets entering and leaving an anonymity system. In Section 6.3 we show the impact of a weaker attacker model on our attack, where the attacker only has partial knowledge of the traffic entering and leaving an anonymity system.

Furthermore we assume that the attacker cannot break the anonymity systems and all used cryptographic functions. This includes that the attacker cannot take over the anonymity system or any part of it. Because we do not concentrate on one anonymity system, the attacker does not use any vulnerability of one specific anonymity system. We only assume that the attacker observes an anonymity system that offers unlinkability without sender and receiver anonymity.

By this he tries to link the sender and receiver nodes. We give details on protocols that match these assumptions in Section 3.1.

3 Background

We use this section to give a short introduction on related work in the area of anonymity. First, we introduce the concept of anonymity and show how it can be implemented for network communication. Secondly we give an overview on traffic analysis attacks that assume a passive attacker as introduced in Section 2.

3.1 Anonymity Protocols

There are many protocols that try to offer anonymity to their users. Before going into these protocols, we need to define what anonymity means. To this end, we give the widely accepted definition of anonymity by Pfitzmann (see (Pfitzmann & Köhntopp 2000)):

> Anonymity of a subject means that the subject is not identifiable within a set of subjects, the anonymity set.

Out of this definition, the main functionality of anonymity protocols should be: First, try to make the anonymity set as big as possible and second, try to make the subjects as unidentifiable as possible. In the best case the probability for having performed an action, from the attackers perspective, is the same for all subjects.In the case of network communication there is a specialization in *sender anonymity*, where the sender belongs to the *sender anonymity set*, and *receiver anonymity*, where the receiver belongs to the *receiver anonymity set*. *Unlinkability* means that the attacker cannot link the sender and the receiver of a communication. This is the case when a protocol offers either sender anonymity, or receiver anonymity, or both, sender and receiver anonymity, or it just makes the mapping of sent and received packets impossible. More details on the definitions of terms in the context of anonymity can be found in (Pfitzmann & Köhntopp 2000).

The first category of anonymity protocols bases on the only protocol that offers provable anonymity, the *dining cryptographers network* (Chaum 1981). The protocol offers sender anonymity, receiver anonymity and therefore also unlinkability of sender and receiver. The main disadvantage of this approach is the communication overhead produced. The idea of this protocol is used in *CliqueNet* (see (Sirer et al. 2001)), *Herbivore* (see (Goel et al. 2003)) and *Acimn* (see (Fusenig et al. 2008)).

The second category of anonymity protocols only offers unlinkability without sender and receiver anonymity. This means that the attacker can observe when a node is sending packets and also when a node is receiving packets. To this class of protocols belong *mixes* (see (Chaum 1981)) and its advancements by (Danezis et al. 2003, Reiter & Rubin 1998, Shields & Levine 2000, Berthold et al. 2000), as well as *Tor,* the *Second Generation Onion Router* which bases on *Onion Routing* (see (Goldschlag et al. 1996, Syverson et al. 2000, Dingledine et al. 2004)), and also the following protocols for anonymous communication in ad hoc networks: ANODR (Kong & Hong 2003), Mask (Zhang et al. 2006), Odar (Sy et al. 2006), and ARM (Seys & Preneel 2006). All the protocols of this second category are vulnerable to our attack that is presented in Section 5.

In the rest of the paper we abstract from the concrete protocol. Instead we use the term *anonymity system*, which refers to an anonymity protocol of the second category.

3.2 Traffic Analysis Attacks

The most powerful attacks on anonymity systems are traffic analysis attacks. In this kind of attack the attacker observes the network traffic of the anonymity system and its users in order to detect communication partners. An attacker can do this passively by just observing, but can also actively induce situations where traffic analysis is much easier.

In Section 2, we restrict ourselves to attackers that passively observe the network traffic. For this reason, we give an overview on traffic analysis attacks for passive attackers.

Message Coding Attack By looking at the coding or content of a message, the attacker might get information on the sender or receiver. For example, in some anonymity protocols the messages do not change the appearance when entering and leaving the anonymity system (i.e. (Kong & Hong 2003)). In this case the attacker can just map the identical packets.

Timing Attack The attacker measures the latency of the reply of a packet. By knowing this, he tries to find the communication partner with the help of a list with known latencies. This technique can also be used for geolocating nodes in the internet (see (Huffmann & Reifer 2005)).

Communication Pattern Attack Because network communication is not random, the attacker can try to identify communication patterns at sender and receiver side.

Intersection Attack For this attack it is assumed that the users of a network usually communicate with the same communication partners more frequently. By observing the network for a longer time period, the attacker can intersect the set of active users at the different points in time.

Packet Volume Attack If the anonymity system allows packets of different size, the attacker can downsize the set of possible receivers of a packet to these nodes which received a packet with the same size.

Packet Counting Attack While counting packets sent and received by the users of an anonymity system, the attacker might link communication partners.

The above attacks show that an attacker has a chance to find communication partners even if they use an anonymity system. Because our attack bases on the concept of packet counting attacks, we give a deeper analysis of this attack in Section 4.

4 Packet Counting Attacks

In this section we show that common packet counting attacks, as introduced in recent literature, are not efficient when a passive attacker as described in Section 2 is assumed.

The idea of packet counting attacks, as in the literature (see (Back et al. 2001, Raymond 2001, Serjantov & Sewell 2003)), is that the attacker can observe the sending and receiving of packets. The attacker counts these sending and receiving events and tries to find similarities in order to derive plausible pairs of communication partners. These approaches assume

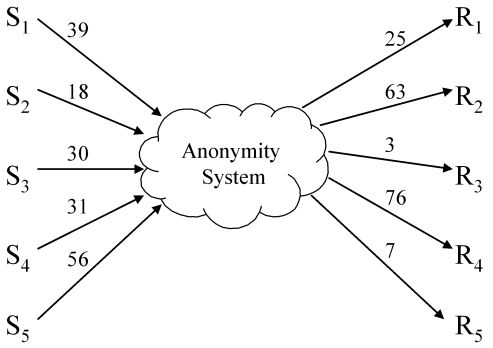

Figure 1: Example: Packet Counting

Table 1: Communication partners

	S_1	S_2	S_3	S_4	S_5
R_1	0	0	25	0	0
R_2	32	0	0	31	0
R_3	0	3	0	0	0
R_4	5	15	0	0	56
R_5	2	0	5	0	0

that a relation between the similarity of the number of sending and receiving events and the probability for the associated communication link exists.

Assume that the attacker has no further information except the number of packets sent by some sender nodes $S_1, \cdots, S_n$ and received by receiver nodes $R_1, \cdots, R_m$. Then, the probability that a node S_i sent a packet to node R_j actually does not depend on how similar the number of sent packets by S_i and received packets at R_j are. The only thing one can say is that the probability becomes higher the more packets S_i sends and the more packets R_j receives while at the same time the number of sender and receiver nodes stays constant. This becomes clear when looking at the following example. Figure 1 shows an anonymity system with sender nodes $S_1, \cdots, S_5$ and receiver nodes $R_1, \cdots, R_5$ as it is seen by an attacker. For evaluation of attacks we show the same setting in Table 1 where additionally the communication partners with the corresponding number of packets are given. Now, assuming that we have no further information except the given number of packets sent and received as in Fig. 1, we can compute the probabilities for every node S_i communicating with R_j as relative frequencies. We write $p(S_i)$ to denote the number of packets that node S_i sends, and $p(R_j)$ for the number of packets received by node R_j respectively. We denote the number of all receiver nodes by n. The nodes $R_k, 1 \leq k \leq n$ receive together $\sum_{k=1}^{n} p(R_k)$ packets, and $R_k, 1 \leq k \leq n, k \neq j$ receive together $\sum_{\substack{k=1 \\ k \neq j}}^{n} p(R_k)$ packets. With this information we can calculate the probability of S_i having sent a packet to R_j by:

$$P(S_i \text{ sent to } R_j) = 1 - \frac{\binom{\sum_{\substack{k=1 \\ k \neq j}}^{n} p(R_k)}{p(S_i)}}{\binom{\sum_{k=1}^{n} p(R_k)}{p(S_i)}} . \quad (1)$$

The resulting probabilities when using the values of Fig. 1 are shown in Table 2. The table shows that the more packets are sent by a node S_i and the more packets are received by a node R_j the higher is the probability that at least one packet was send by S_i to R_j. Hence, a correlation of sender nodes and receiver nodes by only counting packets is not very effective.

This negative effect is also experienced in existing literature. Raymond (Raymond 2001) says that only if one node sends an unusual number of packets, the attacker can find the receiver of this unusual number of packets. Serjantov and Sewell (Serjantov & Sewell 2003) stress that packet counting only works if the whole connection is *lone*. This means that several connections using an anonymity system at the same time, neither share incoming links nor outgoing links of the system, i.e. the number of packets counted for an incoming or outgoing link can be related to one unique connection. Aside from the fact that this is a strong assumption, it is not always possible for the attacker to find out whether a connection is lone or not.

Table 2: Likelihood for communication

	S_1	S_2	S_3	S_4	S_5
R_1	0.999	0.948	0.994	0.995	1.000
R_2	1.000	1.000	1.000	1.000	1.000
R_3	0.535	0.281	0.435	0.447	0.691
R_4	1.000	1.000	1.000	1.000	1.000
R_5	0.837	0.541	0.741	0.753	0.938

5 Slotted Packet Counting

As seen in the previous Section 4 we do not obtain enough information to correlate sender nodes to receiver nodes only by knowing the number of packets sent and received by these nodes. In this section we show how to improve this kind of attack by measuring the alteration of sent and received packets over time.

5.1 Attack Details

We define time slots $t_1, \cdots, t_m$ and count the sent and received packets at every node per time slot. For every sender node S_i we build a random variable $\mathcal{P}_{S_i}$ with values $p_k(S_i)(1 \leq k \leq m)$, giving the number of packets sent in time slot t_k. In the same way we have a random variable $\mathcal{P}_{R_j}$ with values $p_k(R_j)$ $(1 \leq k \leq m)$, representing the number of packets received in time slot t_k at node R_j.

To map the packets in time slot t_k at the sender side to packets at receiver side the attacker has to be aware of timing. Because packets need some time to pass the anonymity system he must shift the time slot t_k at the receiver side by this delay. How to determine the size of the delay is not part of this work. More information on this topic can be found in (Levine et al. 2004).

Now we can calculate the correlation coefficient of the random variables of the sender nodes to the random variables of the receiver nodes in order to map them together (for details on probability theory see i.e. (Bain & Engelhardt 2000)). The correlation coefficient of two random variables gives the strength and direction of their linear relationship. Its value is in the interval $[-1, 1]$, where 0 stands for not correlating variables, 1 and -1 for correlating variables. We are only interested in the cases where the correlation coefficient is near to 1 what means that the random variables have a positive linear relationship. These are the cases where it is highly probable that the corresponding nodes communicated with each other. The correlation coefficient of two random variables X and Y is calculated as follows:

$$\rho_{X,Y} = \frac{cov(X,Y)}{\sigma_x \sigma_y} = \frac{E((X - \mu_X)(Y - \mu_Y))}{\sigma_x \sigma_y} . \quad (2)$$

In order to calculate the correlation coefficient of two random variables $\mathcal{P}_{S_i}$ and $\mathcal{P}_{R_j}$, we need the related expectation values μ_{S_i} and μ_{R_j} as well as the

Table 3: Packet count for sending per time slot

	t_1	t_2	t_3	t_4	t_5
S_1	7	3	14	6	9
S_2	3	3	7	3	2
S_3	7	11	3	7	2
S_4	3	8	6	2	12
S_5	8	12	4	20	12

Table 4: Packet count for receiving per time slot

	t_1	t_2	t_3	t_4	t_5
R_1	6	9	1	7	2
R_2	8	11	18	5	21
R_3	3	0	0	0	0
R_4	8	15	13	26	14
R_5	3	2	2	0	0

standard deviations σ_{S_i} and σ_{R_j}. We assume that the random variables $\mathcal{P}_{S_i}$ and $\mathcal{P}_{R_j}$ are normal distributed and justify this with the central limit theorem (see (Bain & Engelhardt 2000)). With this we can calculate the expectation value μ_{S_i} as follows:

$$\mu_{S_i} = \frac{1}{m} \sum_{k=1}^{m} p_k(S_i) \ . \tag{3}$$

Just as well we can compute the standard deviation σ_{S_i} by:

$$\sigma_{S_i} = \sqrt{\frac{1}{m} \sum_{k=1}^{m} \left(\mu_{S_i} - p_k(S_i)\right)^2} \ . \tag{4}$$

Having calculated the expected values and standard deviations for all sender and receiver nodes we can calculate the correlation between every pair of sender S_i and receiver R_j. We give an example for the calculation in the following Section 5.2.

5.2 Example

We use the network and traffic of Fig. 1 (see Section 4) but split the observations of the traffic in five time slots $t_1, \cdots, t_5$. Table 3 shows the slotted packet count for sent packets by $S_1, \cdots, S_5$; Table 4, respectively, for received packets by $R_1, \cdots, R_5$.

With these values we can calculate expectation values and standard deviations of all combinations of sender and receiver nodes using the formulas (3) and (4). Table 5 shows the correlation coefficients calculated with formula (2) of sender and receiver nodes. Out of these values one can conclude that the nodes S_2 and R_1, S_1 and R_3, S_2 and R_4, S_4 and R_5 communicated with high probability (bold entries in Table 5), which also agrees with the assumed communication pattern given in Table 1. The comparison with the results of a normal packet counting in Table 2 clearly shows that taking the correlation into account offers much more precise results.

Table 5: Correlation coefficients of sender and receiver nodes

	S_1	S_2	S_3	S_4	S_5
R_1	-0.938	-0.492	**0.961**	-0.385	0.547
R_2	**0.623**	0.252	-0.721	**0.843**	-0.536
R_3	-0.109	-0.172	0.155	-0.444	-0.302
R_4	-0.257	-0.147	0.147	-0.274	**0.846**
R_5	0.018	0.363	0.362	-0.296	-0.704

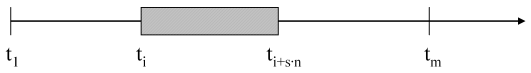

Figure 2: Locating communication slots

6 Attack in Practice

We have already shown the basic concept of the slotted packet counting attack. This section covers the application of the attack in practice.

If an attacker wants to determine whether node S communicated with node R, he has to apply the slotted packed counting attack. The problem is that the attacker has to find the points in time when S communicates with R. For that the attacker measures the packets sent at node S and received at node R during the time period where he guesses a communication. He divides the observations in $m > n$ time slots of length s and scans these values. Therefore he applies the slotted packet counting attack for all intervals with the time slots t_i to $t_{i+n}, 1 \leq i \leq m-n$ (see Fig. 2). A correlation coefficient near to one for any of the intervals indicates that there was a communication with high probability. If there exists more than one such interval, the probability that S sent packets to R becomes higher.

The attacker has to specify the values for size of time slots s and the number of time slots n first. The optimal values for s and n depend mainly on the communication characteristics. Using longer time slots or a higher number of time slots does not necessarily mean that it is easier for an attacker to find a correlation. In the optimal case, the duration $s \cdot n$ of the observation is exactly as long as the communication period of the two target nodes. If the observation time becomes longer, the chance increases that the sender node sends packets to different receiver nodes or that another sender node interferes at the receiver side (more false negatives). A short observation interval on the other hand makes it more likely that correlations between sender and receiver nodes occur even if there is no communication (more false positives). Details on the parameter will be specified in the next section, when applying it to real network traffic.

6.1 Setting

For defining the parameters for the attack we apply it to captured network traffic traces. We use the LBL-PKT-5 traffic traces (Paxson & Floyd 1994) of TCP traffic. The traffic was captured at the Lawrence Berkely Laboratory from 14:00 to 15:00 on January 28, 1994. We assume two scenarios where we want to measure the correlation coefficient: In the first scenario we assume a *mix* with different numbers of input and output links, where one slot refers to one communication round. In the second scenario we assume a low latency anonymity protocol, such as *Tor*, where we define communication slots by time intervals.

We execute the attack with different parameters:

Threshold The threshold for the correlation coefficient. Above this threshold the algorithm assumes a communication link. We vary this parameter in the interval from 0.90 to 0.99.

Number of slots We vary the number of slots which are used for the measurement of the correlation coefficient from 3 up to 500.

Size of slots For the mix scenario we use slots of size 5 up to 455 and for the Tor scenario we use slots of size from 1 sec to 5 sec.

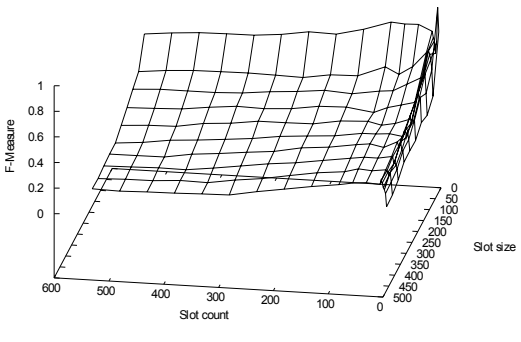

Figure 3: F-Measure of Mix scenario

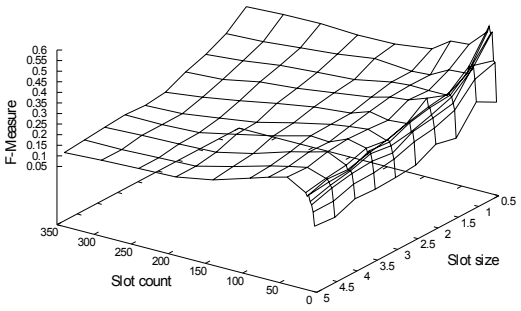

Figure 4: F-Measure of Tor scenario

For evaluation we start the algorithm with different input traces and every possible combination of parameters and use the mean of the different runs. The results of the measurement are discussed in the following section.

6.2 Identifying parameters

We use the F-Measure (Rijsbergen 1979) to identify optimal values for the size and the number of slots used for the measurement. The value of the F-Measure is in the intervall $(0, 1]$, wherein higher values are better. With the F-Measure we want to find values for the parameters that offer a trade-off between the false positives that occur with higher probability when only few slots are used for the measurements and the false negatives. The reason for false negatives is mainly interferences with other communications, i.e. change in communication links or nodes communicating with more than one node, which can be minimized by using a shorter period of time for the measurement.

Figure 3 and 4 show that the value for the F-Measure depends on the size and the number of the communication slots. In both examples we fixed the threshold for the correlation coefficient to 0.95. In the Mix scenario the F-Measure value reaches its maximum for 90 slots for a Mix with 5 incoming and outgoing links and around 10 slots for a Mix with links between 50 and 500. We reached the best results for the Tor scenario at a slot size of 0.5 seconds and 13 slots. The F-Measure value becomes smaller when we increase the size of the slots, where the optimum for the number of measured slots stays in the region of 13 slots.

Figure 5 gives an example where more communication slots used for measuring the correlation coefficient does not lead to a better correlation. In the example a fixed node S_i sends packets to a fixed node R_j. At some point node S_i changes its communication partner to node R_k which results in a decreasing correlation coefficient of the sending and receiving events of S_i and R_j.

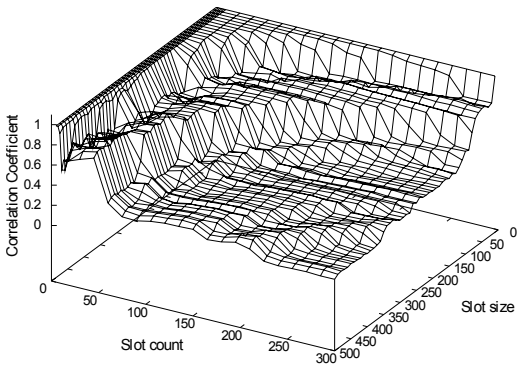

Figure 5: Correlation coefficient of nodes S_i and R_j

6.3 Discussion

The application of the slotted packet counting attack to real network traffic traces shows the practicability of the attack. We showed how to adjust the parameters of the attack in order to improve the output of the attack. Further improvements in the parameters can be made: It might be better to use different parameters for different communication settings. For example visiting a web site in the internet produces a different number of packets than a VoIP phone call. For communication settings where a lot of traffic is produced the attack will be more precise when using more slots. The same number of slots offers no useful results for settings where only few packets over a short period of time are sent.

The attack can fail in two ways as we have seen in the previous section: In the first case there is a strong correlation between a sender node S_i and a receiver node R_j while there was no communication between these nodes (false positive). One reason can be that the measurement uses not enough information, i.e. the covered period of time for the measurement is too short. Another reason might be that the sets of sender and receiver nodes are too big, which results in a higher chance of correlations of sender and receiver nodes, where no communication takes place. Hence, one mission to make this attack effective is to minimize the set of potential sender and receiver nodes. By additionally using the timing of sending and receiving events the set of potential sender and receiver nodes can be reduced. By combining these techniques we might gain better results for slotted packet counting attacks.

The second way of failure is when there is no correlation between nodes S_i and R_j while these nodes are communicating (false negative). As seen in the previous section this happens if there are interferences by other nodes, i.e. there is a node S_k also sending packets to R_j, so that the number of received packets at R_j not only relates to the sending of S_i. In this case, the attacker can calculate the correlation of the sum of packets counted at the sender nodes S_i and S_k with the packets counted at the receiver node R_j. For performance reasons this grouping of sender and receiver nodes should be limited to small groups.

Note that the problems addressed above also exist in the packet counting attacks described in Section 4. But slotted packet counting has the advantage that false positives are less probable, because the packet count has to be similar in every slot. Good results can also be obtained by just observing the nodes of interest because the correlation coefficient only needs the knowledge of the sending and receiving of these nodes. Only for more complex attacks, as mentioned before, more information is needed.

In general this attack is applicable to all

anonymity protocols where the attacker can detect the sending and the receiving of packets. In more specific settings where there is only a limited group of users of the anonymity protocol (i.e. anonymity protocols for ad hoc networks such as (Kong & Hong 2003), (Zhang et al. 2006) and (Seys & Preneel 2006)) the slotted packet counting attack is even more effective. The only way to prevent any kind of packet counting for this class of protocols is to use *constant link padding*, which is not practicable in most settings (see also (Raymond 2001)).

7 Related Work

Packet counting attacks were introduced in (Back et al. 2001, Raymond 2001, Serjantov & Sewell 2003). The attack described in these papers only take the number of packets into account, which are measured at the incoming and outgoing links of an anonymity system. By comparing these numbers they try to detect communication links. If a number of an incoming link and one of an outgoing link are similar, then they assume that these two links belong together. The problem of this approach is, as seen before in Section 4, that they assume lone connections, e.g. both the links for incoming and outgoing packets are only used for one communication. Even if this assumption can be fulfilled it might be possible, that there exist similarities between incoming and outgoing links, which do not belong together. Our slotted packet counting does not need the strong constraint of lone connections. Similarities between incoming and outgoing links are less probable, because they have to be similar for several time slots in order that there is a correlation.

A similar approach to slotted packet counting is proposed by (Levine et al. 2004). They use cross correlation to map input and output traffic of mixes while we concentrate on the mapping of sender and receiver devices. They show effects of network parameters on the attack but do no optimization on the attack parameters. We also give rules how to find parameters for the attack in different network settings.

Watermarking is used to ease traffic analysis. In this case the attacker modifies a traffic flow so that it contains a specific pattern. If this pattern is detected at another position in the network, the two traffic flows are considered to be linked. Watermarking for traffic analysis is used in (Wang & Reeves 2003, Wang et al. 2007, Pyun et al. 2007). A defense to such an attack is proposed in (Kiyavash et al. 2008), where a method is introduced to detect and remove a watermark of a traffic flow. The watermarking technique might improve slotted packet counting in scenarios with active attackers.

8 Conclusion and future work

We have presented a new traffic analysis attack which is applicable to many anonymity systems. It constitutes an improvement of the packet counting attack because it does not require the strong assumption of lone connections as stated in (Serjantov & Sewell 2003). We have shown by way of example and tests on real network traffic traces that our slotted packet counting attack can offer useful results in cases where the standard packet counting fails. The attack is applicable to all anonymity protocols which offer unlinkability but no sender or receiver anonymity.

We have shown different ideas of how to improve slotted packet counting to gain better results in realistic settings. Reducing the size of the anonymity sets helps to improve the results. This can be done by combining the slotted packet counting with timing attacks.

Further improvements of the choice of the parameters can be made: By choosing different parameters for different communication scenarios, e.g. VoIP communication and Web surfing, the results of slotted packet counting attacks can be optimized.

References

Back, A., Möller, U. & Stiglic, A. (2001), Traffic analysis attacks and trade-offs in anonymity providing systems, *in* 'IHW '01: Proceedings of the 4th International Workshop on Information Hiding', Springer-Verlag, London, UK, pp. 245–257.

Bain, L. J. & Engelhardt, M. (2000), *Introduction to Probability and Mathematical Statistics*, 2nd, edn, Duxbury Press.

Berthold, O., Federrath, H. & Köpsell, S. (2000), Web MIXes: A system for anonymous and unobservable Internet access, *in* H. Federrath, ed., 'Proceedings of Designing Privacy Enhancing Technologies: Workshop on Design Issues in Anonymity and Unobservability', Springer-Verlag, LNCS 2009, pp. 115–129.

Chaum, D. (1981), 'Untraceable electronic mail, return addresses, and digital pseudonyms', *Communications of the ACM* **4**(2).

Danezis, G., Díaz, C. & Troncoso, C. (2007), Two-sided statistical disclosure attack, *in* N. Borisov & P. Golle, eds, 'Proceedings of the Seventh Workshop on Privacy Enhancing Technologies (PET 2007)', Springer, Ottawa, Canada.

Danezis, G., Dingledine, R. & Mathewson, N. (2003), Mixminion: Design of a Type III Anonymous Remailer Protocol, *in* 'Proceedings of the 2003 IEEE Symposium on Security and Privacy'.

Dingledine, R., Mathewson, N. & Syverson, P. (2004), Tor: the second-generation onion router, *in* 'SSYM'04: Proceedings of the 13th conference on USENIX Security Symposium', USENIX Association, Berkeley, CA, USA, pp. 21–21.

EU Commission (2006), 'Directive 2006/24/ec of the european parliament and of the council of 15 march 2006', *Office Journal of the European Union* **L 105/54**.

Fusenig, V., Spiewak, D. & Engel, T. (2008), Acimn: A protocol for anonymous communication in multi hop wireless networks, *in* L. Brankovic & M. Miller, eds, 'Sixth Australasian Information Security Conference (AISC 2008)', Vol. 81 of *CRPIT*, ACS, Wollongong, NSW, Australia, pp. 107–114.

Goel, S., Robson, M., Polte, M. & Sirer, E. G. (2003), Herbivore: A Scalable and Efficient Protocol for Anonymous Communication, Technical Report 2003-1890, Cornell University, Ithaca, NY.

Goldschlag, D. M., Reed, M. G. & Syverson, P. F. (1996), Hiding Routing Information, *in* R. Anderson, ed., 'Proceedings of Information Hiding: First International Workshop', Springer-Verlag, LNCS 1174, pp. 137–150.

Huffmann, S. M. & Reifer, M. H. (2005), 'Method for geolocating logical network addresses', Patent of the United States of America as represented by the Directo, National Security Agency. Patent Number: 6,947,978.

Kiyavash, N., Houmansadr, A. & Borisov, N. (2008), Multi-flow attacks against network flow watermarking schemes, *in* 'USENIX Security Symposium', USENIX, p. 307Ű320.

Kong, J. & Hong, X. (2003), Anodr: anonymous on demand routing with untraceable routes for mobile ad-hoc networks, *in* 'MobiHoc '03: Proceedings of the 4th ACM international symposium on Mobile ad hoc networking & computing', ACM Press, New York, NY, USA, pp. 291–302.

Levine, B. N., Reiter, M. K., Wang, C. & Wright, M. K. (2004), Timing attacks in low-latency mix-based systems, *in* A. Juels, ed., 'Proceedings of Financial Cryptography (FC '04)', Springer-Verlag, LNCS 3110, pp. 251–265.

Paxson, V. & Floyd, S. (1994), 'Wide-area traffic: the failure of poisson modeling', *SIGCOMM Comput. Commun. Rev.* **24**(4), 257–268.

Pfitzmann, A. & Köhntopp, M. (2000), Anonymity, unobservability, and pseudonymity - a proposal for terminology, *in* H. Federrath, ed., 'Workshop on Design Issues in Anonymity and Unobservability', Vol. 2009 of *Lecture Notes in Computer Science*, Springer, pp. 1–9.

Pyun, Y. J., Park, Y. H., Wang, X., Reeves, D. S. & Ning, P. (2007), Tracing traffic through intermediate hosts that repacketize flows, *in* 'INFOCOM', IEEE, pp. 634–642.

Raymond, J.-F. (2001), Traffic analysis: protocols, attacks, design issues, and open problems, *in* H. Federrath, ed., 'Designing Privacy Enhancing Technologies: Proceedings of International Workshop on Design Issues in Anonymity and Unobservability', Vol. 2009 of *LNCS*, Springer-Verlag New York, Inc., New York, NY, USA, pp. 10–29.

Reiter, M. & Rubin, A. (1998), 'Crowds: Anonymity for web transactions', *ACM Transactions on Information and System Security* **1**(1).

Rijsbergen, C. J. V. (1979), *Information Retrieval*, Butterworth-Heinemann, Newton, MA, USA.

Serjantov, A. & Sewell, P. (2003), Passive attack analysis for connection-based anonymity systems, *in* 'Proceedings of ESORICS 2003'.

Seys, S. & Preneel, B. (2006), Arm: Anonymous routing protocol for mobile ad hoc networks, *in* 'Proceedings of the 20th IEEE International Conference on Advanced Information Networking and Applications - Workshops (AINA 2006 Workshops)', IEEE, Vienna,AU, pp. 133–137.

Shields, C. & Levine, B. N. (2000), A protocol for anonymous communication over the internet, *in* 'CCS '00: Proceedings of the 7th ACM conference on Computer and communications security', pp. 33–42.

Sirer, E. G., Polte, M. & Robson, M. (2001), Cliquenet: A self-organizing, scalable, peer-to-peer anonymous communication substrate, Technical Report TR2001, Cornell University, Computing and Information Science.

Sy, D., Chen, R. & Bao, L. (2006), Odar: On-demand anonymous routing in ad hoc networks, *in* 'The Third IEEE International Conference on Mobile Ad-hoc and Sensor Systems (MASS)'.

Syverson, P., Reed, M. & Goldschlag, D. (2000), Onion Routing access configurations, *in* 'DARPA Information Survivability Conference and Exposition (DISCEX 2000)', Vol. 1, IEEE CS Press, pp. 34–40.

Wang, X., Chen, S. & Jajodia, S. (2007), Network flow watermarking attack on low-latency anonymous communication systems, *in* 'IEEE Symposium on Security and Privacy', pp. 116–130.

Wang, X. & Reeves, D. S. (2003), Robust correlation of encrypted attack traffic through stepping stones by manipulation of interpacket delays, *in* 'ACM Conference on Computer and Communications Security', pp. 20–29.

Zhang, Y., Liu, W., Lou, W. & Fang, Y. (2006), Mask: Anonymous on-demand routing in mobile ad hoc networks, *in* 'Transactions on Wireless Communications', Vol. 21, IEEE, pp. 2376–2385.