Crowds Revisited: Practically Effective Predecessor Attack

Andriy Panchenko and Lexi Pimenidis*

RWTH Aachen University,
Computer Science Department - Informatik IV,
Ahornstr. 55, D-52074 Aachen, Germany
{panchenko,lexi}@i4.informatik.rwth-aachen.de

Abstract. Crowds is a peer-to-peer system for protecting users' anonymity for web transactions. One of the more serious disadvantages of it is the degree of anonymity provided with respect to the colluding system members: the one who forwards a message to a colluding node is more likely to be the originator of the message than any other member in the system. Furthermore, with the system size growth, the probability that the request came from the initiator of the communication becomes more likely.
In this paper we want to assess to which degree Crowds is applicable despite these weaknesses. To this end, we calculate the needed number of observations for colluding members in order to determine with arbitrary precision how often some users communicate with an external service. An additional question that will be addressed is the possibility to hamper this degradation of the provided anonymity level by a method for adaptive behavior of honest members.

1 Introduction

Anonymization networks are designed and developed in order to achieve anonymity for senders, recipients, or both at the same time. Here, the term *anonymity* is defined as "the state of not being identifiable within a set of subjects, the anonymity set" [9], i.e. the users are protected by means of a set of other users in order to avoid their identification.

Providing an anonymous communication service for the Internet is a demanding task. While cryptography can be used to protect integrity and confidentiality of the data part of the packets, everyone along a route of packets can observe the addresses of the communication partners. To achieve anonymity against third parties a packet's source and destination addresses must be hidden and its appearance should vary from hop to hop. Moreover, timing correlations should be thwarted in order to provide protection against attackers that are able to observe large portions of networks and therewith have a good overview of the traffic within.

There is a number of proposals and practical implementations of anonymization networks (see e.g. [10]). Most of them are based on mixing [2], onion routing [6], or on DC-networks [3]. A number of attacks exist, especially on the low latency implementations (c.f. [8]) that are not trivial to defend against. Those, based on the DC-networks, can be used to provide perfect anonymity under some rather demanding assumptions [3]. The protocol has serious drawbacks causing questionable

* The authors are funded by the European Commission's 6th Framework Program.

practical implementation, i.e. it requires secure and reliable broadcast channels, is prone to channel jamming, inefficient in large networks, etc. [10].

Crowds is a system for anonymous web browsing that was introduced in 1998 by Reiter and Rubin [11] and provides an alternative to the above mentioned popularly deployed and analyzed techniques. It is a peer-to-peer network with a simple randomized routing protocol, where all participants forward messages on behalf of other users as well as their own. It provides security by means of increased path length and was meant to be a tradeoff between performance and security. The main idea of Crowds is to hide each user's communications by routing them randomly within a group of similar users ("blending into a crowd"). When a user requests a web page, he sends the request to another (randomly chosen) crowd member. This member forwards the message to its final destination with probability $1 - p_f$, or to some other random participant with probability p_f. The participating peers in the network are called *jondos* and the central entity for membership management is called *blender* (responsible for new user registration, notification of changes in network, key distribution). Communication between jondos is encrypted, however each of them sees the content of passing messages, including the address of the final destination. The final request to the server is usually sent in plain.

Although the originally proposed network is no longer in use, there are recent implementations and proposals using the Crowds algorithm for network layer anonymization [1, 7, 13]. With this work we aim to make the next step in the security analysis of Crowds. We want to answer the question to which extend it is possible to use Crowds, while still keeping up a certain degree of anonymity in the presence of collusion. Therefore we provide the following two contributions to this topic:

1. A calculation to determine the amount of observations that have to be made by colluding nodes in order to determine a user's peer partners and communication frequencies with a given accuracy. This calculation can then be used to determine a set of system parameters and user behavior, where an attacker will not be able to draw significant advantage from his observations.
2. A proposal for honest nodes' behavior to hamper this traffic analysis.

2 Related Works

The authors of Crowds provide a fairly detailed security analysis of the system [11]. They introduce a degree of anonymity that ranges from absolute privacy to provably exposed and calculate the provided anonymity against local eavesdropper, c collaborating jondos, and an end server. The most interesting and difficult property to achieve is the degree of anonymity with respect to colluding crowd members: a set of malicious jondos that collaborate (or belong to the same entity). Rubin and Reiter have already pointed out that the last node forwarding a message to a colluding jondo, is the originator with a higher probability than any other given node in the network. The achieved degree of the sender anonymity in this case is "probable innocence" (from an attacker's point of view, the sender appears no more likely to be the originator than not to be the originator) given that the quotient of colluding nodes is limited by some threshold[1] that depends on the forwarding probability p_f. However, "probable innocence" is provided only

[1] The interested reader can find more information in [11].

under the assumption of static paths: once a path is established, it does not change for a longer period of time.

Shmatikov [12] applied formal methods to analyze probabilistic anonymity systems on the example of Crowds. The results showed that the degree of anonymity degrades with a larger crowd. That means, if the number of users in a crowd grows and a colluding member receives a request, e.g. for a web page, the probability that the request came from the initiator of the communication increases (it is more likely that the predecessor is the initiator of the communication). Note that this is contrary to a desired property in the area of anonymous communication where the provided degree of protection should rise with the number of its users.

Wright et al. [15] have provided investigations on the predecessor attack on anonymization networks. Crowds was analyzed among other systems. Their approach is applicable in the case if after some rounds of observations only a single member of the crowd has reached the threshold on the number of path initiations. Then he is considered as the communication initiator with a high probability. However, this work does not cover the situation, where more that one user communicates with some external party, although in web surfing scenarios it is very often the case that multiple users have common peers. Our approach has additional significant improvements over this work, as our calculation will not only be able to tell which users communicated with a certain peer, but also how frequently this communication took place. Frequency consideration is important because sending a few messages to some third party is not necessary a sign for having it as a confirmed communication partner: consider for example the case where a user just receives a link per e-mail and is tricked to enter it in the browser. This would lead to false positive categorization, which can be easily filtered out with minimum rates of the communication. Therewith an attacker can define its own threshold in order to distinguish whether the communication takes place or not. Furthermore, an honest user can use our analysis to adapt his behavior and improve the degree of anonymity.

Andersson et al. [1] have empirically compared the correlation between the forward probability and the quotient of the colluding nodes needed in order to achieve "probable innocence", based on the results of Reiter and Rubin [11]. The outcome was that in order to have an acceptable path length, and thus a limited forwarding probability p_f, the adversary should not be in control of more that 40% of nodes in the network.

Díaz [5] has used information-theoretical metrics (effective anonymity set size and degree of anonymity) that are based on the entropy in order to evaluate the security features of Crowds. She has also analytically shown that the degree of anonymity slightly decreases with size of the crowd. An adversary is also able to extract more information from a crowd with more members because the distribution of probabilities is less uniform.

3 Model and Calculus

In this section we will shortly display our model of Crowds and then develop our calculation for the security analysis.

Let $n + c$ be the size of the crowd (please note the difference to the original notation, where n was the total number of members in the system), c is the number of colluding jondos and p_f is the probability of forwarding in the system as defined

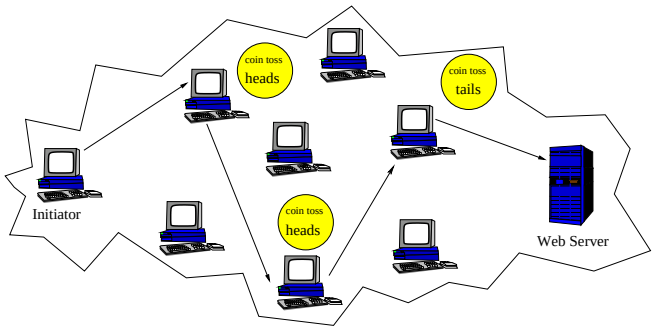


Fig. 1. A crowd having 8 nodes, performing a website access

in [11]. $n + c$ and p_f are system parameters known to everyone, while c is known only to the adversary. A simple example is depicted in Figure 1.

3.1 Static Paths

In this subsection we will shortly recapitulate the security the Crowds provides for a single path. This evaluation is also valid for static paths, where the path does not change once it is initiated.

In [11] Rubin and Reiter calculate p_h (in the original notation $P(I)$), the probability that a colluding jondo receives a message directly from the path initiator. Let p_l denote the probability for each other honest jondo that the collaborators on the path receives the message from him, instead of the originating jondo. For convenience of the reader we provide the formulas here:

$$p_h = \frac{c(n + c - p_f(n-1))}{(n+c)(n+c-np_f)} \tag{1}$$

$$\Rightarrow p_l = \frac{1-p_h}{n-1} = \frac{n^2(1-p_f) + c(n-p_f)}{(n+c)(n+c-np_f)(n-1)} \tag{2}$$

In order to negate the colluding nodes from gaining any information, i.e. to provide *perfect security*, the following must hold: $p_h = \frac{1}{n} = p_l$. This means that the one who forwards the message to colluding jondo must be not more suspicious than any other honest jondo. Taking into account that $p_f \in (0, 1)$ and $n > 1$, the equation does not have a solution for $c > 0$. Thus it is possible to declare that the legacy Crowds can not provide perfect security against colluding jondos.

The probabilities p_h and p_l provide a measure for the security under the static paths assumption, i.e. if the same path is reused for successive requests. However, since $p_h \neq p_l$, some information is leaked. This knowledge can be accumulated by an attacker to make more precise statements about the users' peers. In the following we will generalize the model and include dynamic paths that change over time. This means that a new route is constructed either per fixed time period (e.g. as in original Crowds approach between joins of new nodes), or even per each outgoing request.

3.2 Dynamic Paths

In the original approach the participating jondo sets up one path for all its users' communications. This path is altered only under two circumstances: when failures are detected in the path, or when new jondos join the crowd (in order to protect the joiners). Note that both are subject to manipulation by an attacker, even if the system is designed in the way that such an action is only allowed in discrete time intervals. Thus, even if a crowd uses static paths in order to protect the privacy of its users, there will be nevertheless changes in the paths under the above mentioned circumstances. Additionally, Crowds could be extended in order to provide support for dynamic paths for each request to achieve better performance through load balancing among the system members. In this case sending a single message can be seen as a new path creation. Without loss of generality and for the sake of simplicity we will consider the latter to be the case.

We will assume that the attacker participates in the crowd with c jondos ($c \geq 1$) and, without loss of generality, only stores the passing communication to a single external entity, namely Bob. In this section we will calculate the number of observations t that are needed in order to estimate the amount of communication that each honest user initiated with Bob with arbitrary precision.

From the adversary's point of view, there are n jondos that are sending messages to Bob, each with its own average rate λ_i per time interval. We model these as Poisson processes $A_i = \mathcal{P}_{\lambda_i}$ for $i \in 1 \ldots n$. This kind of arrival distribution is in our opinion a fair tradeoff between the analytical complexity and realism. Herewith we want to provide a first approximation, which can be further refined modelling arrivals in a more sophisticated manner. The colluding entities observe on average from the i-th system member the following number of messages to Bob per time interval:

$$E[\text{msg to Bob from } i] = p_h \cdot \lambda_i + \frac{(1-p_h)}{n-1} \cdot \sum_{j=1, j\neq i}^{n} \lambda_j \tag{3}$$

With the help of the following matrix M, we can model the number of messages O_i arriving at the attacker from jondo i to Bob:

$$M = \begin{pmatrix} p_h & p_l & \cdots & p_l \\ p_l & p_h & \cdots & p_l \\ \vdots & \vdots & & \vdots \\ p_l & p_l & \cdots & p_h \end{pmatrix} \tag{4}$$

$$O_i = \sum_{j=1}^{n} \mathcal{P}_{m_{i,j}\lambda_j} \tag{5}$$

The observations can be seen as a vector of observed messages (each element is the number of messages received from corresponding jondo by colluding members and addressed to Bob), which is a product of the vector of actually sent messages with the matrix M (see Figure 3). Since O_i is a sum of the Poisson processes, it is also a Poisson process with the following properties:

$$O_i = \mathcal{P}_{\omega_i} \quad \text{for} \quad \omega_i = \sum_{j=1}^{n} m_{i,j}\lambda_j \tag{6}$$

$$E[O_i] = V[O_i] = \omega_i \tag{7}$$

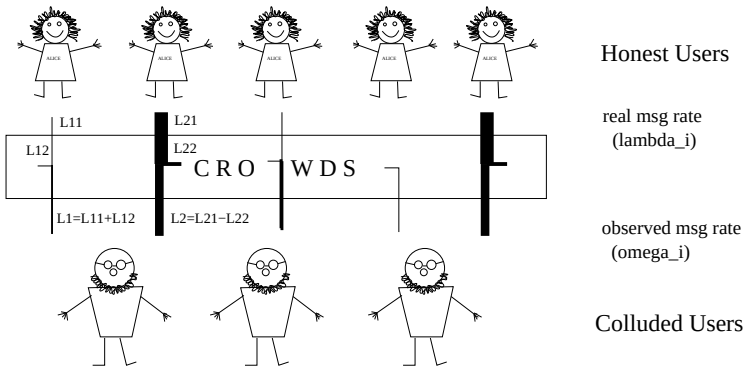


Fig. 2. Information flow for the attacker: ω_i depends on λ_i and vice versa

We depict the information flow of the O_i in Figure 2 (the thickness of the lines denotes the sending rate of the corresponding jondos). While the upper lines denote the rate of own messages to Bob, the lower ones stand for the number of messages to Bob that are forwarded to the colluding nodes by the corresponding jondo. There it can be seen that the cardinality of ω_i depends on the value of the λ_i. Thus, from the observations of O_i it is possible to draw conclusions about the λ_i (for $t \to \infty$):

$$(\omega_1, \ldots, \omega_n) = M(\lambda_1, \ldots, \lambda_n) \tag{8}$$

$$\iff (\lambda_1, \ldots, \lambda_n) = M^{-1}(\omega_1, \ldots, \omega_n) \tag{9}$$

Indeed, the matrix M is invertible to M^{-1} because Crowds is unable to provide perfect security ($p_h \neq p_l$), thus:

$$M^{-1} = \begin{pmatrix} \tilde{p_h} & \tilde{p_l} & \cdots & \tilde{p_l} \\ \tilde{p_l} & \tilde{p_h} & \cdots & \tilde{p_l} \\ \vdots & \vdots & & \vdots \\ \tilde{p_l} & \tilde{p_l} & \cdots & \tilde{p_h} \end{pmatrix} \tag{10}$$

$$\tilde{p_h} = \frac{n+p_h-2}{np_h-1} \tag{11}$$

$$\tilde{p_l} = \frac{p_h-1}{np_h-1} \tag{12}$$

Under the common values for p_f, n, and c ($0 < p_f < 1$, $n \gg c$) the following inequalities hold: $\tilde{p_h} > 0$ and $\tilde{p_l} < 0$.

Having made the observations of the ω_i, it is thus possible to calculate an estimation for the λ_i, namely $\tilde{\lambda_i}$ as follows:

$$\tilde{\lambda_i} = \sum_{j=1}^{n} {m^{-1}}_{i,j} \omega_j \tag{13}$$

In the next sections, we will discuss the number of observations that an adversary has to make in order to estimate λ_i precisely enough.

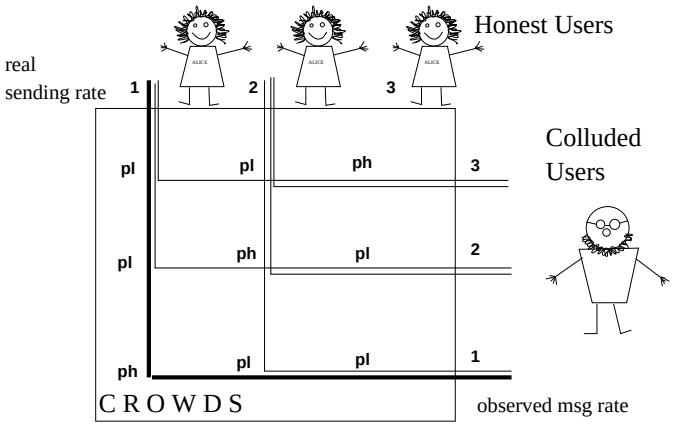


Fig. 3. Information flow for the attacker: Matrix M

3.3 Expected Error of ω_i

While calculating an estimated mean value for λ_i is quite simple, we will now calculate the expected error of this estimation. The attacker can use this measure to decide whether he already collected enough evidence or needs to observe the honest nodes for a longer period. To this end, we use a Chernoff style bound for the probability that a Poisson process X deviates from its mean $\mu = E[X]$ with a factor of $\delta \geq 0$. This is (lower and upper bound)

$$p(X < (1-\delta)\mu) < \left(\frac{e^{-\delta}}{(1-\delta)^{1-\delta}}\right)^{\mu} \tag{14}$$

$$p(X > (1+\delta)\mu) < \left(\frac{e^{\delta}}{(1+\delta)^{1+\delta}}\right)^{\mu} \tag{15}$$

Now we want to asses which bound is stronger. To this end we show that:

$$p(X > (1+\delta)\mu) - p(X < (1-\delta)\mu) > 0 \tag{16}$$

The latter is due to the fact that for $\delta \in (0, 0.5]$ the following holds:

$$e^{2\delta}(1-\delta)^{1-\delta} - (1+\delta)^{1+\delta} > 0 \tag{17}$$

So if we can guarantee the upper bound with the probability p and the deviation factor δ, the lower bound is at least as good as the upper one. Let us set $k = 1+\delta$ ($\Rightarrow k > 1$), then we have

$$p(X > k\mu) < e^{-(k\log k - k + 1)\mu} \tag{18}$$

Performing t observations and summing the number of messages that were received from the i-th jondo it is now possible to estimate the probability p of the deviation of his sum from $k \cdot t \cdot \mu$. This formula can be resolved to t: for a

desired deviation factor of k and a chosen probability p, the formula determines the minimum number of observations needed in order to estimate the expected value (mean) with the given probability p (to be in the interval $(\mu(2-k), \mu k)$ with the probability $1-p$):

$$p \leq e^{-(k \log k - k + 1)t\mu} \tag{19}$$

$$\iff \; t \geq \frac{-\log p}{\mu(k \log(k) - k + 1)} \tag{20}$$

To calculate t for each O_i set $\mu = \omega_i$ (average, so far observed) and choose desired probability $p = err_p(\omega_i)$ and deviation $k = err_k(\omega_i)$:

$$t(\omega_i, err_p(\omega_i), err_k(\omega_i)) = \frac{-\log err_p(\omega_i)}{\omega_i(err_k(\omega_i)\log(err_k(\omega_i)) - err_k(\omega_i) + 1)} \tag{21}$$

Thus, an attacker can easily determine the quality of his observations. As the next we will use this to calculate the quality of the attacker's estimated $\tilde{\lambda_i}$.

3.4 Expected Error of $\tilde{\lambda_i}$

It should be noted that due to the fact that Formula 13 has negative coefficients, the $\tilde{\lambda_i}$ are not Poisson distributed with the parameter $\tilde{\lambda_i}$. But we can still estimate the expected deviation $err_k(\tilde{\lambda_i})$ from the real value (that is arised through the linear recombination of ω_i) and the probability that the real value is within the bounds of the deviation $err_p(\tilde{\lambda_i})$.

$$err_p(\tilde{\lambda_i}) = 1 - (1 - err_p(\omega_i))^n \tag{22}$$

$$err_k(\tilde{\lambda_i}) = \quad err_k(\omega_i) \tag{23}$$

From these, for an arbitrary chosen $err_k(\tilde{\lambda_i})$ and $err_p(\tilde{\lambda_i})$ it follows that an attacker needs to have observations of this quality:

$$err_p(\omega_i) = 1 - e^{\frac{\log(1 - err_p(\tilde{\lambda_i}))}{n}} \tag{24}$$

$$err_k(\omega_i) = \quad err_k(\tilde{\lambda_i}) \tag{25}$$

Finally, the minimum number of observations t_{total} that is required in order to estimate all λ_i within the chosen deviation factor of $err_k(\tilde{\lambda_i})$ with arbitrary chosen probability (that the estimation is wrong) $err_p(\tilde{\lambda_i})$ then can be derived as follows:

$$t_{total} = \max_{i=1..n} t(\omega_i, 1 - e^{\frac{\log(1 - err_p(\tilde{\lambda_i}))}{n}}, err_k(\tilde{\lambda_i})) \tag{26}$$

$$\iff$$

$$t_{total} = t(\min_{i=1..n} \omega_i, 1 - e^{\frac{\log(1 - err_p(\tilde{\lambda_i}))}{n}}, err_k(\tilde{\lambda_i})) \tag{27}$$

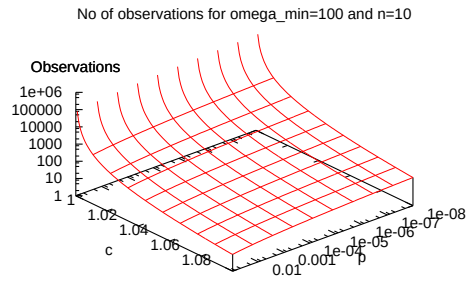
$$\iff$$

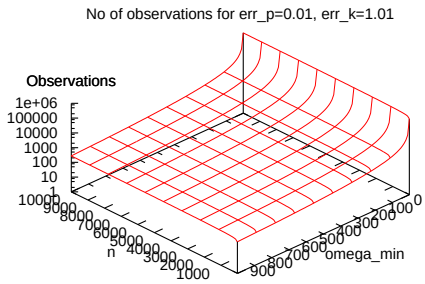
$$t_{total} = \frac{-\log(1 - e^{\frac{\log(1 - err_p(\tilde{\lambda_i}))}{n}})}{\omega_{min}(err_k(\tilde{\lambda_i})\log(err_k(\tilde{\lambda_i})) - err_k(\tilde{\lambda_i}) + 1)} \tag{28}$$

Using Formula 28, we can calculate the number of observations needed in order to estimate sending rates $\tilde{\lambda_i}$ with any desired precision.

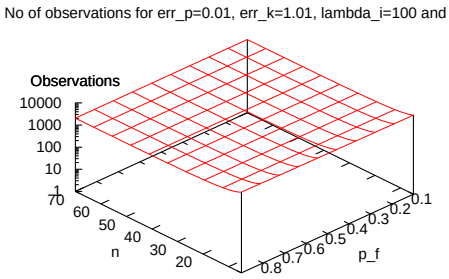
4 Analysis

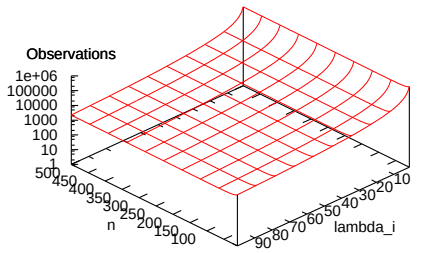
In order to see how Formula 28 can be used, we have applied it for different parameter ranges, i.e. size of the honest crowd users n, desired arbitrary small probability that the estimation is wrong $err_p(\tilde{\lambda}_i)$, deviation factor $err_k(\tilde{\lambda}_i)$ and the minimum rate of observed messages to Bob ω_{min}. Therewith we compute the needed number of observations for estimate of the sending rates of all users with arbitrary precision. The results are illustrated in the form of the plots in Figure 4. It should be also noted that, for example, on the plot where ω_{min} is mapped against n, the required number of observations for a small ω_{min} is quite large. This is true, but it has only a minor practical implication due to the fact, that even one doesn't send own messages to some third party, the observed by colluding users message rate going to this party from the user will be reasonably high (see an example in the next section). Therefore, usually in the practice ω_{min} will not have small values.





t_{total} for $err_k = 1 - 1.1$ and $err_p = 0 - 0.1$ t_{total} for $\omega_{min} = 1 - 1000$ and $n = 2 - 100$





t_{total} for $\lambda_0 = 0$, $\lambda_i = 100$ and $p_f = 0.1 - 0.9$ t_{total} for $\lambda_i = 1 - 1000$ and $\lambda_0 = 0$

Fig. 4. Number of observations to be made in order to break Crowds with forwarding probability (p_f), number of honest (n) and colluding (c) jondos, required error in estimations (err_p) and precision factor (err_k), min real sending rate of one jondo (λ_0) and of all others (λ_i), as well as min observed sending rate (ω_{min})

Note that the value of err_p and the size of the group of honest jondos n do not have much impact on the result for the considered ranges of parameters. Therewith it is possible to say that the provided degree of anonymity does not degrade with a larger crowd. However, neither it does not increase with the larger user base as usually expected in the area of anonymous communication.

4.1 Practical Application

We will provide an example on the needed number of observations in a system, where one user does not communicate with Bob, while the others frequently send messages. Let the number of honest jondos be $n = 500$, the number of colluding jondos $c = 10$, and the forwarding probability is $p_f = 0.9$. Assume that one jondo does not send own messages to Bob $\lambda_{min} = 0$, the required probability of error in the estimation is $err_p(\tilde{\lambda}_i) = 0.15$ and the precision factor $err_k(\tilde{\lambda}_i) = 1.1$ (thus, the real sending rate is $\pm 10\%$ from the estimated value with the probability 85%). If all jondos except the *min*-th send on average with the rate $\lambda_i = 10$ messages per time interval, $\omega_{min} = 9.8$, and we have:

$$t_{total} = \frac{-\log(1-e^{\frac{\log(1-0.15)}{500}})}{9.8(1.1\log(1.1)-1.1+1)} \approx 169 \tag{29}$$

Under the given setting the adversary has to monitor network only 169 time intervals in order to estimate rates of all jondos with the precision $\pm 10\%$ and the error probability of 15%. Even if the *min*-th system member doesn't send own messages to Bob, the average number of messages to Bob he forwards to the colluding jondos per time interval is more than nine.

Note that the granularity (length) of the chosen time interval for rate observation does not have an impact on the overall time needed in order to estimate the rates. This is due to the proportional dependency between ω_{min} and t_{total} in Formula 28. If the length of the observations' time interval is doubled, so are the values of ω_i. This halves the number of needed observation intervals in return, keeping the absolute observation time fix as a total.

5 Discussion

As we have seen in the previous section, the number of observations needed in order to determine the sending rate of the jondos in the crowd with a high precision is relatively small. This rises a question on the possibility of using the system for users with a high demand on the level of provided anonymity.

In order to overcome the described traffic analysis it is possible to introduce an adaptive behavior of the honest system members: each honest jondo has to passively monitor the network in the same way as an adversary, but having only himself at the disposal ($c = 1$). Doing so, it is possible to estimate the sending rate and peers of the other jondos. Thus, a node is able to estimate the total flow of messages to a potential peer and calculate the maximum frequency it may send messages to this peer without being detected by an attacker. This number depends on a maximum estimated number of colluding nodes, the existing traffic to the peer, and the allowed error probability. Nevertheless a user should take into account that a set of colluding nodes will be able to make more precise estimations than a single node, thus the honest user should take into account additional discounts on the calculated rate in order to be sure to leak as less information as possible.

A second way to circumvent this attack is to increase the own rate to the maximum observed rate (by all honest jondos) with the help of dummy messages in order to protect the privacy of the other members and to form a uniform anonymity set. Adapting the sending rate to the highest rate of the honest jondos will cause the attacker to observe an equal rate from all of the honest system members. Provided

good quality dummy messages, he will not be able to determine the original sending rate of each user.

Please note that an adversary would not benefit from injection of own fake messages. In case of having the maximum rate he would only cause a global increase of dummy traffic of all other members in the system, thus making his own life harder. On the other hand, if the colluding jondos do not obey the rules and do not adapt to the highest rate in the system, their behavior becomes detectable (honest jondos receive less messages from them than from any other members) and they can be excluded from the crowd.

As opposed to DC-networks [3], having adaptive behavior in a crowd it is provable that each jondo sends messages to the third party, even if they were only dummy ones. In this case system members cannot deny their communication with the external party, but rather have to claim that all that was a dummy traffic.

A third way to eliminate the threat of the described traffic analysis relies on *helper nodes* [14] to prevent the offender from gaining such information. This is because the discussed attack on Crowds relies on the fact that a colluding entity receives a message from the communication initiator with a higher probability. The basic idea behind the helper node is to always choose a single node (or one from a small subset) as the first one to relay traffic to. If the chosen node is an honest one, the described attack is not successful. On the other hand, the problem of finding helper nodes is dual to the problem of the trust establishment inside of the crowd.

Making observations and using above provided computations, honest jondos can give users a feedback on the provided level of anonymity (based on the probability and precision factor) and, if needed, warn about the danger of consecutive requests to the same peer. Alternatively, the communication rate can be decreased in order to achieve the desired anonymity properties.

It should be noted that the model presented is simplified and outgoing messages are modelled as Poisson processes which is, from our point of view, a fair tradeoff between the analytical complexity and realism. There will definitely be fluctuations in the rate according to the time, date, user etc. However, this is one commonly made assumption of many researchers in the field [4].

6 Conclusion

We have provided a calculation for the needed number of observations by a set of collaborating colluding jondos in order to determine the rates of all honest jondos that send messages to some third party with arbitrary precision. Frequency consideration is important because it eliminates false positive categorization of peer partners, depending on the own threshold value for rate.

It was shown that Crowds can not provide perfect security for its members under any admissible parameter values. The number of observations needed in order to determine the sending rate of the jondos in the crowd precisely enough can be relatively small. This rises a reasonable doubt on the possibility of using the system for strong anonymity in an open environment.

Furthermore, we proposed an adaptive behavior for the jondos to improve the provided degree of anonymity.

References

1. C. Andersson, R. Lundin, and S. Fischer-Hübner. Privacy-enhanced WAP Browsing with mCrowds - Anonymity Properties and Performance Evaluation of the mCrowds System. In *Proceedings of the Fourth annual ISSA 2004 IT Security Conference*, Johannesburg, South Africa, July 2004.
2. D. L. Chaum. Untraceable Electronic Mail, Return Addresses, and Digital Pseudonyms. *Communications of the ACM*, 24(2):84 – 88, Feb 1981.
3. D. L. Chaum. The Dining Cryptographers Problem: Unconditional Sender and Recipient Untraceability. *Journal of Cryptology*, (1):65 – 75, 1988.
4. G. Danezis, C. Diaz, and C. Troncoso. Two-sided Statistical Disclosure Attack. In N. Borisov and P. Golle, editors, *Proceedings of the Seventh Workshop on Privacy Enhancing Technologies (PET 2007)*, Ottawa, Canada, July 2007.
5. C. Díaz, S. Seys, J. Claessens, and B. Preneel. Towards Measuring Anonymity. In R. Dingledine and P. Syverson, editors, *Proceedings of Privacy Enhancing Technologies Workshop (PET 2002)*. Springer-Verlag, LNCS 2482, April 2002.
6. R. Dingledine, N. Mathewson, and P. Syverson. Tor: The Second-Generation Onion Router. In *Proceedings of the 13th USENIX Security Symposium*, 2004.
7. L. A. Martucci, C. Andersson, and S. Fischer-Hübner. Chameleon and the identity-anonymity paradox: Anonymity in mobile ad hoc networks. In *Proceedings of the 1^{st} International Workshop on Security (IWSEC 2006)*, Kyoto, Japan, 23–24 Oct 2006.
8. S. J. Murdoch and G. Danezis. Low-Cost Traffic Analysis of Tor. In *Proceedings of the 2005 IEEE Symposium on Security and Privacy*, Oakland, California, USA, May 2005. IEEE Computer Society Press.
9. A. Pfitzmann and M. Hansen. Anonymity, Unlinkability, Unobservability, Pseudonymity, and Identity Management - A Consolidated Proposal for Terminology. Draft, version 0.28, May 2006.
10. J.-F. Raymond. Traffic Analysis: Protocols, Attacks, Design Issues, and Open Problems. In H. Federrath, editor, *Proceedings of Designing Privacy Enhancing Technologies: Workshop on Design Issues in Anonymity and Unobservability*, pages 10–29. Springer-Verlag, LNCS 2009, July 2000.
11. M. K. Reiter and A. D. Rubin. Crowds: Anonymity for Web Transactions. *ACM Transactions on Information and System Security*, pages 66 – 92, April 1998.
12. V. Shmatikov. Probabilistic model checking of an anonymity system. *Journal of Computer Security*, 12(3-4):355–377, 2004.
13. A. Vaha-Sipila and T. Virtanen. BT-Crowds: Crowds-Style Anonymity with Bluetooth and Java. In *HICSS '05: Proceedings of the 38th Annual Hawaii International Conference on System Sciences (HICSS'05) - Track 9*, page 320.1, Washington, DC, USA, 2005. IEEE Computer Society.
14. M. Wright, M. Adler, B. N. Levine, and C. Shields. Defending anonymous communication against passive logging attacks. In *Proceedings of the 2003 IEEE Symposium on Security and Privacy*. IEEE, May 2003.
15. M. Wright, M. Adler, B. N. Levine, and C. Shields. The predecessor attack: An analysis of a threat to anonymous communications systems. In *ACM Transactions on Information and System Security TISSEC'04*, volume 7 (4), pages 489 – 522. ACM Press, November 2004.