The wisdom of Crowds: attacks and optimal constructions

George Danezis[1], Claudia Diaz[2], Emilia Käsper[2], and Carmela Troncoso[2]

[1] Microsoft Research Cambridge
gdane@microsoft.com
[2] K.U. Leuven/IBBT, ESAT/SCD-COSIC
firstname.lastname@esat.kuleuven.be

Abstract. We present a traffic analysis of the ADU anonymity scheme presented at ESORICS 2008, and the related RADU scheme. We show that optimal attacks are able to de-anonymize messages more effectively than believed before. Our analysis applies to single messages as well as long term observations using multiple messages. The search of a "better" scheme is bound to fail, since we prove that the original Crowds anonymity system provides the best security for any given mean messaging latency. Finally we present D-Crowds, a scheme that supports any path length distribution, while leaking the least possible information, and quantify the optimal attacks against it.

1 Introduction

Muñoz-Gea *et al.* [4] presented at ESORICS 2008 a variant of Crowds [5] to anonymously route packets in a peer-to-peer network. The *always–down-or-up* algorithm (ADU) they propose is similar to Crowds in that when a node receives a message, it decides probabilistically whether to forward it to its final destination or to another node in the crowd. The difference with Crowds is in the decision procedure. Instead of forwarding messages with a fixed probability $\bar{p}$, nodes in ADU forward messages with a probability that depends on their position in the message path. This probability is computed using a variable u decided locally by each node and forwarded to its successor in the path. The ADU algorithm results in path lengths with smaller variance than those of Crowds.

In this work, we study the anonymity given by both algorithms and show how an attacker who controls a fraction of the crowd can exploit the value of the parameter u to better identify the initiator of a communication. Further we show that, contrary to Crowds, the ADU algorithm is vulnerable to predecessor attacks [7] performed by the destination server – because it allows the initiator to send the message directly to the server.

We also prove that Crowds' decision procedure provides optimal anonymity for a given mean path length, and that changing the path length distribution necessarily results in weaker anonymity. For the cases where the geometric path length distribution of Crowds is not adequate we propose D-Crowds, an algorithm that supports arbitrary path length distributions while leaking the least possible amount of information. Finally, we evaluate the resistance of D-Crowds against optimal attacks.

The rest of the paper is organized as follows. We first recall Crowds in Sect. 2. The ADU algorithm, and a variant of it, are presented in Sect. 3. We evaluate the performance of the three algorithms in terms of path length and anonymity in Sect. 4. In Sect. 5 we prove the optimality of the Crowds' decision procedure and describe the D-Crowds algorithm. Finally we offer our conclusions in Sect. 6.

2 Crowds

Crowds [5] was proposed as a system for communicating anonymously, using a peer-to-peer network (a crowd) to pass messages. The message-passing algorithm for Crowds is simple: a user wishing to send a message to a destination first passes it to a random node in the crowd. Each subsequent recipient then flips a (biased) coin to decide whether to send the message to the destination or to pass it to another crowd member. We say that Crowds has parameter $\bar{p}$ if the probability of sending the message to the end destination is $p = 1 - \bar{p}$. The average number of hops a message travels in the crowd before reaching the final destination is then $1 + \bar{p}/p = 1/p$.

The key feature that enables anonymity in Crowds is that upon receiving a message from a crowd member, we do not know whether this is the initiator of the message, or an intermediary who is just forwarding it. We can however, compute the probability that each member in the crowd is the initiator of the message, and quantify anonymity [2, 6] as the entropy of this probability distribution.

Crowds provides the initiator with perfect anonymity with respect to the end destination, since the destination is equally likely to receive the message from any crowd member. Collaborating dishonest crowd members, on the other hand, can infer some information about the initiator. More specifically, the anonymity of the initiator with respect to the crowd is a function of two parameters, the fraction of dishonest nodes f and the Crowds parameter $\bar{p}$.

Hence, it is natural to ask whether there exist other Crowds-like message passing algorithms that provide better security guarantees for a given message delivery latency. We proceed to show that the always–down–or–up algorithm is less secure compared to Crowds, and furthermore, that the message passing algorithm of Crowds is in fact optimal, and thus *all* attempts to improve upon Crowds are bound to fail.

3 The Always–Down-or-Up Algorithm

The advantage of the always–down-or-up algorithm (ADU) [4] decision procedure with respect to Crowds [5] is that it results in a smaller variance of the path length. Hence, the length of a path does not differ substantially from the mean length determined by the system parameters. The ADU decision procedure is a mix of two algorithms: the *always–down* (AD) and the *always–up* (AU) algorithms. In the AD scheme, the initiator n^0 of a message chooses a random

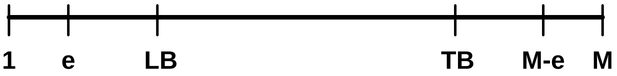

Fig. 1. Parameters for the ADU algorithm.

integer u^0 in the interval $[1, M]$ (being M a parameter of the system.) We denote n^i the i-th node in the path, and u^i the value it generates. If $u^0 = 1$ the message is sent to its final destination; otherwise it is forwarded to the next node, n^1, along with u^0. n^1 selects a new value u^1, but using u^0 as upper bound of the interval. This process is repeated, with $u^{i+1} \in [1, u^i)$, until the message exits the network. The AU algorithm operates similarly, substituting the lower bound by the previous u at each hop (i.e., $u^{i+1} \in (u^i, M]$.)

Already in [4], it is noted that both AD and AU reduce the variance of the path length at the cost of anonymity, as the value u transmitted from a node to its successor leaks information about its position in the path. The ADU algorithm tries to alleviate this problem by choosing the mode of operation (AD or AU) at random. For this purpose the algorithm has four integer numbers as system parameters: M, e, LB and TB, represented in Fig. 1. In ADU, the initiator of a request chooses a random number u between 1 and M. When this number belongs to the intervals $[1, e]$ or $[M - e, M]$, the message is sent directly to its destination. If the message stays in the network, the initiator chooses between AD and AU depending on u: the chosen mode is AD if $u \in (e, LB]$, AU if $u \in [TB, M - e)$ and it is decided at random otherwise ($u \in (LB, TB)$.)

Even though the initiator selects the mode of operation at random, the choice is communicated to subsequent nodes on the path when forwarding the message along with the u. Any corrupt node in the path observes the selected mode of operation, and in that sense ADU is no better than the AU or AD algorithms, contrary to the security analysis in [4].

An alternative algorithm, that we call “Random Always Down-or-Up” algorithm (RADU,) does not forward the mode of operation, and nodes choose independently between AD and AU. The algorithm would work as follows: the initiator n^0 chooses $u^0 \in [1, M]$ and sends the message to the destination if $u^0 \in [1, e]$ or $u^0 \in [M - e, M]$. If the message remains in the network, it is forwarded to a new node n^1 along with u^0. Upon receiving u^0, n^1 decides which mode to use: it chooses AD if $u^0 \in (e, LB]$, AU if $u^0 \in [TB, M - e)$ or at random otherwise ($u^0 \in (LB, TB)$.) Once the mode is selected, the node picks u^1 from $[1, u^0)$ (respectively $(u^0, M]$) and restarts the process. We note that contrary to the ADU algorithm, a node does not transmit to its successor the mode of operation it has chosen. Thus, n^{i+1} cannot make inferences about its position in the path assuming that u^i has been generated according to a concrete mode of operation.

The next sections compare ADU and RADU to Crowds in terms of path length variance and anonymity.

Table 1. Comparison between ADU, RADU and Crowds algorithms.

	(M, e, LB, TB)	$\bar{l}$	$var(l)$	$\bar{p}$	$var_{\text{Crowds}}(l)$
ADU	(100,21,30,70)	0.91	1.02	-	-
	(100,8,20,80)	1.91	2.10	0.53	1.73
	(100,3,20,80)	2.27	2.79	0.44	2.88
	(150,2,20,130)	3.52	3.62	0.29	8.87
	(350,2,20,330)	4.55	4.65	0.22	16.15
RADU	(100,21,30,70)	0.94	1.19	-	-
	(100,8,20,80)	2.08	3.13	0.48	2.25
	(100,3,20,80)	2.78	3.80	0.36	4.95
	(150,2,20,130)	3.98	6.86	0.25	11.86
	(350,2,20,330)	6.27	19.72	0.16	33.04

4 Evaluation

4.1 Path length variance

Muñoz-Gea *et al.* [4] demonstrate that the ADU algorithm leads to paths with smaller variance than Crowds. In this section we confirm this result and compare the variance of ADU, RADU and Crowds. We note that our results differ from those presented by Muñoz-Gea *et al.* : in [4], the "minimum path" for ADU is one hop, when the initiator sends the request directly to the end destination; while for Crowds a path length of one corresponds to the request passing by an intermediate node before reaching its destination – i.e., the definition of "path length" is different for Crowds than for ADU, rendering the comparison in [4] unfair.

We implemented simulators for the ADU and RADU algorithms and computed the mean and the variance of the path length denoted, respectively, as $\bar{l}$ and var(l). In the case of Crowds these values can be computed analytically as the mean and variance of a geometric distribution with parameter $\bar{p}$:

$$\bar{l}_{\text{Crowds}} = 1 + \frac{1-\bar{p}}{\bar{p}} = \frac{1}{\bar{p}} \qquad \text{var}_{\text{Crowds}}(l) = \frac{1-\bar{p}}{\bar{p}^2}$$

In all three algorithms, we consider that path length l corresponds to l intermediate hops between initiator and destination, with $l = 0$ indicating the case when the initiator sends the request directly to the destination.

In our experiments we use sets of values proposed in [4] for M, e, LB, and TB. The results are summarized in Table 1. The fourth column expresses the value of $\bar{p}$ necessary in Crowds to obtain the same mean path length as in ADU or RADU, respectively. The symbol '-' in the first row of the table indicates that there is no possible $\bar{p}$ in Crowds that achieves a mean path length smaller than one.

Table 1 shows how for the same parameters, the path length in RADU has a larger mean and variance than in ADU. This is because in ADU the mode of operation (AU or AD) is fixed, and successive nodes choose u from decreasing size intervals; while in RADU the size of the interval may increase. To illustrate this effect let us consider a scenario with parameters (M=100,e=8,LB=20,TB=80) in which the initiator n^0 selects $u^0 = 47$. As $47 \notin [1, 8] \cup [92, 100]$ the message and u^0 are forwarded to node n^1. When n^1 receives u^0 it selects an operation mode. Let us assume that the selected mode is AD, and $u^1 = 35$ is chosen from $[1, 47)$. Thus, the message is forwarded again to node n^2. This node, however, selects AU as mode of operation and chooses u^2 from the interval $(35, 100]$. In this case the third node in the path is less likely to send the message to the destination than its predecessor. If the ADU algorithm was used, u^2 would be chosen from $[1, 35)$, and the probability of a shorter path would be higher. This effect also explains the larger path length variance of RADU.

Although the performance of RADU in terms of variance is worse than ADU, it is still better than Crowds (significantly better as the mean path length increases.) As we explain in the next section, the penalty in performance comes in exchange for better anonymity.

4.2 Anonymity with respect to corrupt nodes

We consider a threat model in which the attacker controls C out of the N nodes in the network. When a corrupt node receives a message, it tries to infer whether its predecessor is the initiator or not. We denote by $\Pr[n_i|u, n_x]$ the probability that node n_i is the initiator of a message given all the information available to the attacker – i.e., the node n_x from which the message was received and the ADU/RADU routing parameter u associated with the message. This probability can be decomposed as:

$$\Pr[n_i|u, n_x] = \frac{\Pr[u|n_x, n_i] \cdot \Pr[n_x|n_i] \cdot \Pr[n_i]}{\sum_{\forall j} \Pr[u|n_x, n_j] \cdot \Pr[n_x|n_j] \cdot \Pr[n_j]} .$$

Where $\Pr[n_i]$ is the *a priori* probability of a node n_i being the initiator; $\Pr[n_i|n_x]$ is the probability that node n_i is the initiator of the message when n_x is the predecessor of the first corrupt node in the path (not taking into account u); and $\Pr[u|n_x, n_i]$ denotes the probability that a value u is received from predecessor n_x when n_i is the initiator.

We assume the adversary has no prior information on who is likely to be the initiator, and thus $\Pr[n_j] = \Pr[n_i] \, \forall i, j$. We estimate the distribution of $\Pr[n_i|n_x]$ and of $\Pr[u|n_x, n_i]$ experimentally. For this, we have implemented simulations of the ADU, RADU, and Crowds routing algorithms.

For each of the algorithms, we simulate $C_T = 100\,000$ experiments and count the number C_i of times that the predecessor n_x of a corrupt node is the same node as the initiator n_i. We compute the probability that $n_x = n_i$ as $\Pr[n_i|n_i] = \frac{C_i}{C_T}$. Similarly to Crowds, all other honest nodes are equally likely to be the initiator with probability $\Pr[n_x|n_i] = \frac{1-\Pr[n_i|n_i]}{N-C-1}$, $\forall x \neq i$.

We proceed similarly to estimate $\Pr[u|n_x, n_i]$: we simulate a large number of ADU and RADU experiments and collect values of u received when $n_x = n_i$ and when $n_x \neq n_i$. Figure 2 shows the distribution of u when the initiator and predecessor coincide (i.e., $\Pr[u|n_i, n_i]$) and when they do not coincide (i.e., $\Pr[u|n_x, n_i]$.) The experiments were conducted in a network formed by $N = 100$ nodes of which $C = 10$ are corrupt (i.e., $f = 0.1$,) when considering ADU and RADU with parameters (M=100,e=8,LB=20,TB=80), and Crowds with parameter $\bar{p} = 0.53$ (for comparison with ADU) and $\bar{p} = 0.48$ (for comparison with RADU.)

We observe that in both ADU and RADU initiators forward values of u that are uniformly distributed between $e + 1$ and $M - e - 1$ (values of $u \in [1, e] \cup [M - e, M]$ never appear in forwarded requests, as the node generating that u would send the request to the end server.) In ADU, the distribution of u when the node that relays message is other than the initiator (i.e., $n_x \neq n_i$) is skewed towards large or small u's depending on the chosen mode (AD or AU) – given that, as a message is forwarded, nodes choose u from decreasing intervals. For RADU, the distribution behaves roughly as a combination of AD and AU.

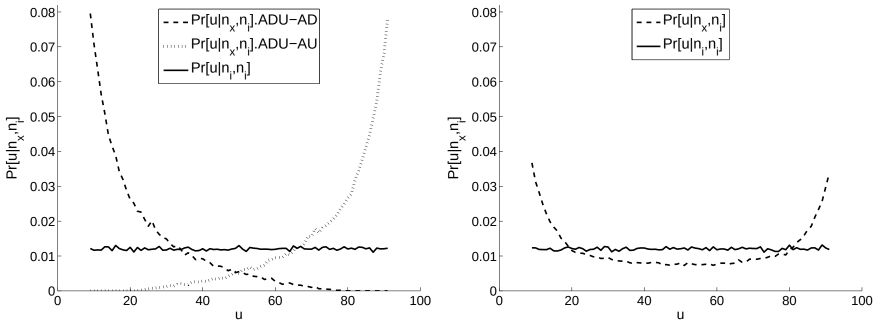

Fig. 2. $\Pr[u|n_i, n_i]$ and $\Pr[u|n_x, n_i]$ for ADU (left) and RADU (right) with (M=100,e=8,LB=20,TB=80).

Figure 3, left, shows $\Pr[n_i|u, n_x]$ for all considered algorithms. In Crowds there is no u parameter, and thus $\Pr[n_i|u, n_x] = \Pr[n_i|n_x]$ is constant in u. We observe that, for ADU in AD mode it is not possible to have u's larger than $TB = 80$ (or AU would have been chosen,) and the same holds for AU and u's lower than $LB = 20$. Secondly, we can see in the figure how any of the operation modes severely diminishes the uncertainty of the attacker with respect to the initiator. For example, in AD mode large u's indicate that the predecessor is likely to be the initiator. This uncertainty is even non-existent if for example $u = TB - 1$ and mode AD is chosen, as only the initiator could have generated this value (subsequent nodes choose from $[1, u), u < TB - 1$.)

In Fig. 3, right, we show the entropy of the probability distribution $\Pr[n_i|u, n_x]$, which expresses the initiator anonymity [2, 6]. As expected, ADU provides the worst anonymity in most of the cases. RADU improves considerably this result, but still it leaks more information than simple Crowds. It is worth noting that in some cases (e.g., a very low u when operating in ADU-AD) anonymity is higher for ADU than for Crowds, even though the adversary has gained knowledge from the u. In these cases the adversary is more uncertain about the initiator because it is probably *not* its predecessor – i.e., the adversary gains the knowledge that it is probably *not* succeeding the initiator in the path. The fact that additional information may increase anonymity was explained in [3].

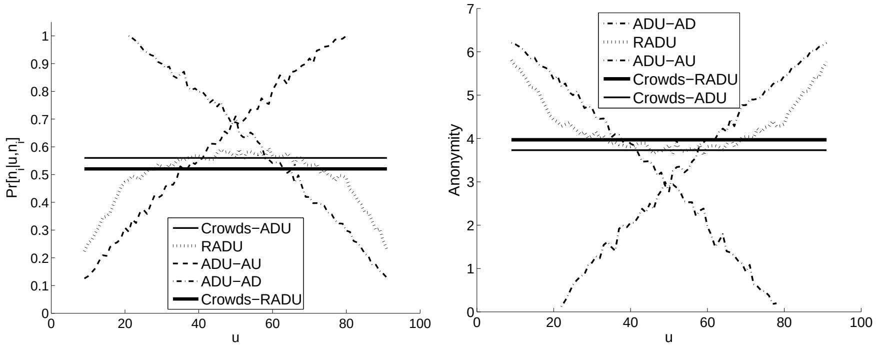

Fig. 3. The probability $\Pr[n_i|u, n_i]$ (left;) and the entropy the distribution $\Pr[n_i|u, n_x]$ (right.) The ADU and RADU parameters are (M=100,e=8,LB=20,TB=80). Crowds-ADU has parameter $\bar{p} = 0.53$ (i.e., same $\bar{l}$ as ADU in the figure), and Crowds-RADU has parameter $\bar{p} = 0.48$ (i.e., same $\bar{l}$ as RADU in the figure).

4.3 Anonymity with respect to the end server

One of the adversaries considered in Crowds [5] corresponds to the end server to which the initiator is connecting; i.e., the recipient of the communication. As explained in Sect. 2, the initiator in Crowds first selects a crowd member (possibly itself) uniformly at random, and forwards the request to it. When this node receives the request, it flips a biased coin to determine whether or not to forward the request to another node (with probability $\bar{p}$) or to the end server (with probability $p = 1 - \bar{p}$.) In Crowds, any member of the crowd is equally likely to be the initiator of a request from the point of view of the end server (i.e., with probability $\frac{1}{N}$,) regardless of the identity of the exit Crowds node. For this reason, Crowds provides maximum anonymity [2, 6] towards this adversary, which corresponds to $\log_2(N)$ for a crowd of N members.

In the ADU scheme [4] on the other hand, the initiator sends the request *directly* to the end server with probability $\frac{2e}{M}$ (whenever $u \leq e$ or $u \geq M - e$,) and it forwards the request to a crowd member with probability $1 - \frac{2e}{M}$. Given this algorithm[1], the initiator is more likely to be the exit node of its own request than any other node. Let e and M be the parameters of the ADU routing algorithm,

[1] Note that RADU operates in the same way.

and let N be the number of nodes in a crowd. Let $\Pr[n_x|n_i]$ denote the probability that node n_x ($x = 1, \ldots, N$) is the exit node for a request made by initiator n_i ($i = 1, \ldots, N$.) In ADU, the probability $\Pr[n_x|n_i]$ is higher when $x = i$ than when $x \neq i$:

$$\Pr[n_x|n_i] = \begin{cases} \frac{2e}{M} + (1 - \frac{2e}{M})\frac{1}{N} & x = i \\ (1 - \frac{2e}{M})\frac{1}{N} & x \neq i \end{cases} \tag{1}$$

As a result, the initiator anonymity provided by ADU with respect to the end server is lower than that provided by Crowds. Note that we assume that no prior information is available to the adversary, and thus $\Pr[n_j] = \Pr[n_i]\ \forall j, i$. Therefore,

$$\Pr[n_i|n_x] = \frac{\Pr[n_x|n_i]\Pr[n_i]}{\sum_{j=1}^{N}\Pr[n_x|n_j]\Pr[n_j]} = \Pr[n_x|n_i]$$

expresses the probability that n_i is the initiator of a request, given that n_x sends the request to the end server (i.e., n_x is the exit node.)

Figure 4 compares the anonymity provided by ADU and Crowds against this adversary model, and shows its variation with respect to the the crowd size N and the ADU parameter e. We can see in the figure of the left that both Crowds and ADU provide better anonymity when the N grows, but that for any given N the anonymity of Crowds is substantially higher than that of ADU. For a crowd size of 500, Crowds provides 9 bits of anonymity, while ADU provides little more than 6 bits – this corresponds to the anonymity that Crowds provides to a crowd smaller than 80.

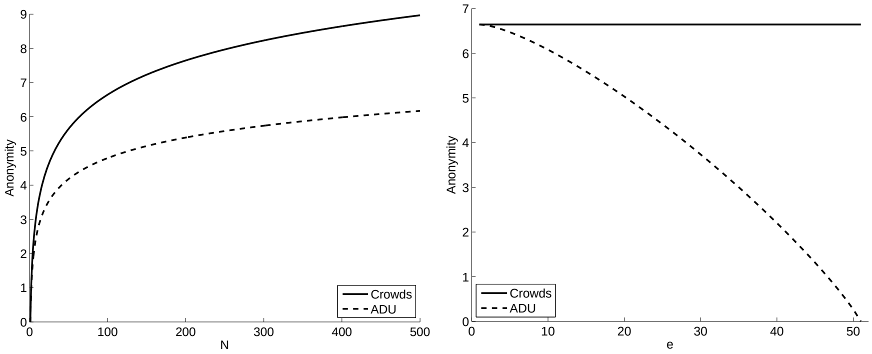

Fig. 4. Initiator anonymity for one request with respect to the end server; i.e., entropy of the distribution $\Pr[n_i|n_x]$, $1 \leq i \leq N$. Variation with respect to the crowd size N (left) with $M = 100$ and $e = 21$; and with respect to e (right) with $N = 100$, $M = 100$

The figure on the right shows the variation with respect to e. When e grows, the initiator sends the request directly to the server with a higher probability. A large e parameter increases efficiency by reducing the path length, but the penalty in anonymity is rather severe. At $e = 15$, the anonymity loss of ADU with respect to Crowds is one bit, which has the same effect as cutting the crowd size by half. When $e = 50$, the initiator always sends the requests directly to the end server, and thus ADU provides no anonymity.

4.4 Multiple requests by the same initiator to the same server

If we consider multiple requests from the same initiator to the same end server over time, the anonymity provided by the ADU algorithm further degrades with the number of requests. This section extends the Predecessor attack [7] to evaluate the anonymity degradation of ADU towards the end server. The key idea behind the Predecessor attack is that the true initiator of an anonymous request will always appear in the path. If independent requests by the same initiator can be linked together (e.g., someone frequently visiting the same unpopular web page,) and the adversary has a chance of being the immediate successor of the initiator in the anonymous path, then the adversary is able to identify the initiator with high probability after a number of requests.

The attack in [7] examined an adversary model that consists of a subset of corrupted nodes – a more complex case than that of the end server, since the adversary only sees some of the requests – and provides bounds on the number of requests beyond which anonymity degrades to unacceptable levels. The end server on the other hand, is always on the path of the request (at the end of it,) and in ADU it receives the request directly from the initiator with higher probability than a corrupt node for the sets of parameters suggested in [4].

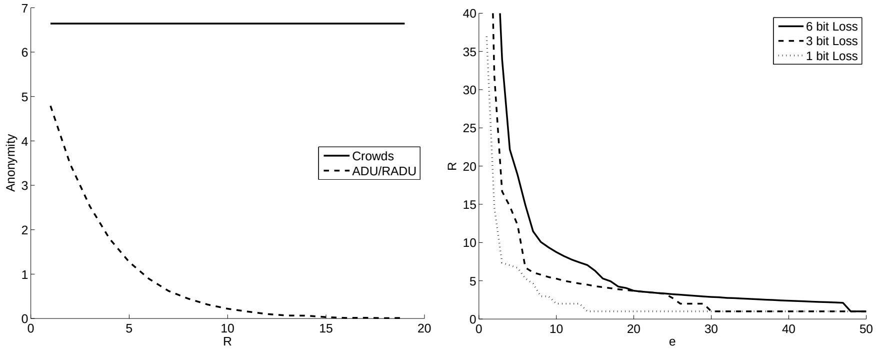

Fig. 5. Anonymity with respect to the end server relative to the number R of requests with $e = 21$ (left); and number R of requests after which anonymity is degraded by 1, 3, and 6 bits (right). Average over ten thousand tests with $M = 100$, $N = 100$.

In a worst-case scenario, consider that node n_i is the only node in a stable crowd of N nodes that is sending requests to an end server S. Let R be the number of requests sent by n_i to S, and $\Phi = \{\phi_x; 1 \leq x \leq N\}$ be the observed vector of frequencies, where ϕ_x is the number of times that n_x appears as the exit node for the requests of n_i – i.e., $\sum_{x=1}^{N} \phi_x = R$.

The probability $\Pr[\varPhi|n_i]$ of observing a vector of frequencies $\varPhi$ when n_i is the initiator of R requests, is given by the probability mass function of the multinomial distribution $f(n_1 \ldots n_N; R, \Pr[n_1|n_i] \ldots \Pr[n_N|n_i])$, with $\Pr[n_x|n_i]$ computed with formula (1). Let q_0 denote $\Pr[n_i|n_i]$, and q_1 denote $\Pr[n_x|n_i, x \neq i]$, and note that $q_0 + (N-1)q_1 = 1$. The probability $\Pr[\varPhi|n_i]$ is given by:

$$\Pr[\varPhi|n_i] = \frac{R!}{\prod_{j=1}^{N} \phi_j!} q_0^{\phi_i} \prod_{k=1, k\neq i}^{N} q_1^{\phi_k} = \frac{R!}{\prod_{j=1}^{N} \phi_j!} q_0^{\phi_i} q_1^{R-\phi_i}$$

Given an observed vector of frequencies $\varPhi$, we can compute the posterior probability $\Pr[n_i|\varPhi]$ applying Bayes' theorem:

$$\Pr[n_i|\varPhi] = \frac{\Pr[\varPhi|n_i] \Pr[n_i]}{\sum_{j=1}^{N} \Pr[\varPhi|n_j] \Pr[n_j]}$$

Considering that a priori $\Pr[n_i] = \Pr[n_j]\ \forall i, j$, we obtain:

$$\Pr[n_i|\varPhi] = \frac{q_0^{\phi_i} q_1^{R-\phi_i}}{\sum_{j=1}^{N} q_0^{\phi_j} q_1^{R-\phi_j}}$$

We have simulated the ADU algorithm and experimentally generated observation vectors $\varPhi$. Given these vectors, we compute initiator anonymity as the entropy of the distribution $\Pr[n_i|\varPhi], 1 \leq i \leq N$. As we can see in Fig. 5, left, the anonymity provided by ADU quickly degrades when several requests are made – after ten requests, the end server is able to identify the initiator with overwhelming probability – while the anonymity provided by Crowds remains stable.[2] Figure 5, right, shows the number of ADU/RADU requests after which anonymity has decreased from its maximum by 1, 3 and 6 bits, as a function of the parameter e.

5 Optimal decision procedures

We have seen that the ADU mechanism, as well as its RADU variant are less secure than Crowds. In this section we prove a key result: the decision criterion used by Crowds, that leads to a geometric distribution of path length, is in fact optimal for passing messages anonymously through a crowd.

In order to model message passing through a crowd, we first propose D-Crowds, a variant of Crowds that only leaks the time-to-live of a message—the number of remaining hops in the crowd—to the attacker, while allowing an arbitrary path length distribution D. We then argue that all crowds-based systems can be reduced to D-Crowds without loss in security. Finally, we prove

[2] In Crowds, $q_0 = q_1 = \frac{1}{N}$, thus $\Pr[n_i|\varPhi] = \frac{q_1^R}{\sum_{j=1}^{N} q_1^R} = \frac{1}{N}$, and initiator anonymity is $\log_2(N)$.

that D-Crowds provides optimal security when D is a geometric distribution. More specifically, we show that any other distribution of path lengths D would require a longer mean path length to achieve the same level of anonymity.

5.1 D-Crowds: A generic TTL-based Crowds

The original Crowds, as well as ADU, RADU and other algorithms for passing messages through a crowd can all be captured via the following general model: the initiator of the connection passes her message, along with its destination and some routing information we denote by r_0, to a randomly chosen node in the crowd. The routing information may or may not be updated as the message passes through the crowd. The nodes in the path apply some arbitrary decision procedure based on the routing information r_i they have received, to decide whether to forward the message to another node, along with some routing information r_{i+1}. If the message is not forwarded within the crowd it is relayed to its final destination.

In the case of Crowds, the routing information is simply the static forwarding probability $\bar{p}$; in the case of ADU/RADU, it is the dynamically updated random value $u^i \in [1, M]$ (and the direction AD or AU for ADU). We call any system that follows this model a crowds-based system, and we eventually prove that the original Crowds is an optimal crowds-based system with respect to anonymity in the crowd.[3]

First, we note that each crowds-based routing procedure results in path lengths that are overall distributed according to some fixed distribution $D(l)$ for $l \geq 0$. The following key observation allows to abstract away from details of the decision procedure, or the routing information: every crowds-based system necessarily leaks the time-to-live of a message—the number of remaining hops in the crowd—to the adversary. Namely, the adversary, after observing a message, can "simulate" its trajectory by forwarding it to other corrupt nodes or simply to itself until the message exits the crowd. Since all nodes, including corrupt ones, must be able to decide whether to pass the message to the destination, it is necessary to leak such information, and our traffic analysis is based on the adversary observing a message and its time-to-live.

On the other hand, the time-to-live is also sufficient to decide whether to forward the message or keep it in the crowd, and any other additional auxiliary information can only decrease the security of the system. Thus, we can restrict our security analysis to the case where the auxiliary information consists of only the time-to-live of the message, More formally, we define D-Crowds in the following way:

[3] Strictly speaking, ADU and RADU as proposed do not fully satisfy this definition, as they pass a small fraction of messages directly to the destination. Obviously, a system where all messages are passed directly to the destination provides best crowd anonymity, while being trivially insecure against the end server. In order to guarantee security against the end server, we thus require that the initiator *always* passes the message through the crowd.

Definition 1. *In D-*Crowds*, the initiator draws a path length l_0 from an arbitrary distribution of paths $l_0 \sim D$, and explicitly forwards it as a time-to-live value with the message to a randomly chosen node within the D-Crowds network. Upon receiving a message, a node checks the TTL value l_i: if it is zero, it outputs the message to its ultimate destination, if not, it forwards the message to a random node within the crowd with a TTL value $l_{i+1} = l_i - 1$.*

When D is a geometric distribution, we refer to the system simply as Crowds.

The TTL value is both necessary and sufficient to perform the routing. There is no need to include any other information for routing at all, since the TTL allows nodes to make a decision on whether to forward the message or keep it within the crowd. Nevertheless, for simplicity of analysis, we assume that the distribution D is also public. Contrary to the original Crowds which leaks its path length distribution via the parameter $\bar{p}$, D-Crowds does not require the initiator to publish D. However, the adversary may be able to infer information about D from traffic patterns, so to be on the safe side, we assume the strongest adversary that knows the whole distribution D.

5.2 The optimality of Crowds

We model D-Crowds as having two components: a distribution D of non-negative[4] integer path lengths $l \geq 0$, and a probability any node is dishonest f.

Denote the probability the h^{th} node on a path is the first dishonest node by $\Pr[H = h]$; $H = 0$ corresponds to the event that the initiator forwards the message to a dishonest node. We note that some messages are never observed by a dishonest participant; this corresponds to the event $l < h$.

In case the adversary observes a message, the traffic analysis of D-Crowds boils down to the following question: given the distribution D and a message with its observed time-to-live value, what is the probability that the predecessor is the initiator of the connection?

Since a single time-to-live value is available to an adversary seeing the message, the best possible analysis is to calculate the probability $\Pr[H = 0|\text{TTL} = \text{ttl}]$, where $\text{TTL} = \text{ttl}$ is the current time-to-live value observed by the adversary. Since no additional routing information r_i is passed along the message, aside the TTL, no additional information can leak though the routing strategy of D-Crowds, and this probability indeed captures the full traffic analysis capabilities of the adversary.

For any fraction f of corrupt nodes, we define the advantage of the D-Crowds adversary to be

$$\mathsf{Adv}^f(D) = \max_{\text{ttl}} \Pr[H = 0|\text{TTL} = \text{ttl}].$$

In order to say that some general D-crowds provides better security than original Crowds, the following needs to hold: for all possible values of f ($0 < f < 1$), the advantage of the adversary must be smaller for D-Crowds.

[4] Each message always passes at least one node in the crowd, but as the first hop is deterministic, we ignore it in our analysis.

A key result we prove is that: if the condition above holds, thus the security provided by a length distribution D is better than what is provided by a geometric distribution Geom_p, then it must follow that the mean of distribution D is larger, namely $\mathbb{E}(D) \geq \mathbb{E}(\mathrm{Geom}_p)$. We formalize this as the following theorem (The detailed proof is shown in Appendix A):

Theorem A1 *For an arbitrary distribution $D(l)$ over path lengths, if for all f, $0 < f < 1$,*

$$\mathsf{Adv}^f(D) \leq \mathsf{Adv}^f(Geom_p),$$

then

$$\mathbb{E}(D) \geq \mathbb{E}(Geom_p).$$

Note that we consider worst-case rather than average-case security. We argue that it is of no use if a system is better only for some values of the observed TTL, or for the expected TTL. First of all, providing average case guarantees is not appropriate for a security system, since it is unknown to us what the cost of a single compromise would be. What's worse in the case of Crowds, messages are not necessarily independent, and compromising one message may lead to the deanonymization of others. Second, each sender cares about their own message, and has no incentive to forward a message with a TTL that is *a priori* known to be particularly vulnerable.

In order to prove the theorem, we express the advantage of the adversary via the distribution D. Recall that we are interested in the probability $\Pr[H = 0|\mathrm{TTL} = \mathrm{ttl}]$ that the message with an observed time-to-live value ttl came from the initiator. The probability $\Pr[H = h|\mathrm{TTL} = \mathrm{ttl}]$ is easy to relate, using Bayes theorem, with the probability $\Pr[\mathrm{TTL} = \mathrm{ttl}, D = h + \mathrm{ttl}|H = h]$ that a message travels a further ttl hops, while it has already travelled h hops. The latter can be expressed as

$$\Pr[\mathrm{TTL} = \mathrm{ttl}|H = h] = \frac{D(\mathrm{ttl} + h)}{\sum_{\mathrm{ttl} \geq 0} D(\mathrm{ttl} + h)} = \frac{D(\mathrm{ttl} + h)}{F(h)}, \tag{2}$$

where $F(h)$ is a cumulative value defined as $F(h) = \sum_{l \geq h} D(l)$.

We also need the probability $\Pr[H = h]$ that the h^{th} node on a path is the first dishonest node. The number of hops a message will transit until it is observed by the adversary is distributed geometrically according to the fraction of dishonest members of the crowd, and the desired probability can be expressed as:

$$\Pr[H = h] = \bar{f}^h f \sum_{l \geq h} D(l) = \bar{f}^h f F(h), \tag{3}$$

Assuming that H, the distribution of first compromised node, and D the distribution of lengths are independent, we can now provide the following expression:

$$\begin{aligned}\Pr[H=h|\mathrm{TTL}=\mathrm{ttl}]_D = & \quad \frac{\Pr[\mathrm{TTL}=\mathrm{ttl}|H=h]\cdot\Pr[H=h]}{\sum_{h\geq 0}\Pr[\mathrm{TTL}=\mathrm{ttl}|H=h]\cdot\Pr[H=h]} \\ = & \quad \frac{D(h+\mathrm{ttl})\cdot\bar{f}^h fF(h)}{\sum_{h\geq 0}D(h+\mathrm{ttl})\cdot\bar{f}^h fF(h)} \qquad (4)\end{aligned}$$

In the special case of Crowds where D is a geometric distribution ($D(l) = \mathrm{Geom}_p(l) = \bar{p}^l p$,) we have that:

$$\Pr[H=h|\mathrm{TTL}=\mathrm{ttl}]_{\mathrm{Geom}_p} = (\bar{p}\bar{f})^h(1-\bar{p}\bar{f}) \qquad (5)$$

Note that, due to the memoryless property of the geometric distribution of paths, the above probability distribution is independent from the time-to-live (ttl,) and the adversary gains no additional information from observing it. In the general case this is not true (eq. 4,) and the probability of inferring the initiator ($\Pr[H=0|\mathrm{TTL}]$) varies according to the observed time-to-live of the message.

In order for D-Crowds to provide better security than Crowds, we must thus have

$$\forall 0<f<1.\ \max_{\mathrm{ttl}}\Pr[H=0|\mathrm{TTL}=\mathrm{ttl}]_D \leq \max_{\mathrm{ttl}}\Pr[H=0|\mathrm{TTL}=\mathrm{ttl}]_{\mathrm{Geom}_p}.$$

which, from eq. 4 and eq. 5, implies that,

$$\forall 0<f<1, \mathrm{ttl}\geq 0. \quad \frac{D(\mathrm{ttl})}{\sum_{h\geq 0}D(h+\mathrm{ttl})\bar{f}^h} \leq 1-\bar{p}\bar{f}. \qquad (6)$$

Finally, we prove Theorem A1 by showing that if condition 6 holds for some distribution D, then its mean is larger than that of the geometric distribution with parameter p (see app. A for details).

We can conclude that for any decision procedure to be uniformly better than Crowds (i.e., for all f and ttl), it must lead to longer paths. Conversely, for a fixed mean path length, Crowds provides the best security. Thus, from traffic analysis and security perspective, there is little reason to look beyond Crowds.

5.3 *D*-Crowds for other distributions

Recall that Crowds with exit probability p has mean path length $\bar{l} = 1/p$, variance $(1-p)/p^2$, and deanonymization probability $\Pr[H=0|\mathrm{TTL}=\mathrm{ttl}] = 1-\bar{p}\bar{f}$ for any observed time-to-live in a Crowd with a fraction f corrupt nodes. We have already shown that any D-Crowds with the same mean provides suboptimal anonymity guarantees. Nevertheless, we next consider different distributions D to illustrate the trade-off between path length variance and anonymity.

In our examples, we fix the fraction of corrupt nodes to $f = 0.1$ and take Crowds with probability $p = 0.25$, mean path length $\bar{l} = 4$, variance $\sigma^2 = 12$, and uniform deanonymization probability $Pr[H=0|\mathrm{TTL}=\mathrm{ttl}] = 0.325$ as our

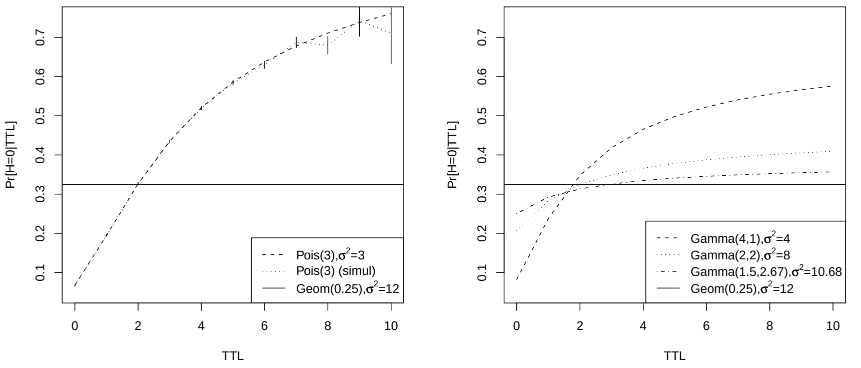

Fig. 6. Deanonymization probabilities $\Pr[H = 0|\mathrm{TTL} = \mathrm{ttl}]$ for Poisson-Crowds (left) and Gamma-Crowds (right) with fixed mean $\bar{l} = 4$.

benchmark. First, we sample path lengths from a Poisson distribution $Pois(\lambda)$; $\lambda = 3$ yields the desired mean $\bar{l} = \lambda + 1 = 4$. Namely, we sample path lengths from $[0, \infty)$ and add 1 to the length, as each message has to travel at least one hop, from the initiator to the first Crowd node.

Fig. 6 (left) plots the theoretical probability curve, as well as the results of 1000000 simulations; vertical bars indicate the 90% confidence interval. We see that the Poisson distribution $Pois(3)$ turns out to be a poor choice for this parameter set: when the adversary observes a time-to-live $TTL \geq 4$, there is at least 50% confidence that the sender of the message is indeed the initiator.

Next, we consider the discrete quantized version of the gamma-distribution. Fig. 6 plots the deanonymization probabilities for three distributions $\Gamma(4, 1)$, $\Gamma(2, 2)$ and $\Gamma(4/3, 3)$ with mean $\bar{l} = 4$ and variances $\sigma^2 = 4$, $\sigma^2 = 8$ and $\sigma^2 = 10.67$, respectively. We observe a clear trade-off: when keeping the mean fixed, decreased variance yields decreased anonymity guarantees. In particular, while $\Gamma(1.5, 2.67)$-Crowds indeed provides rather good anonymity, it also has little performance advantage over Crowds, as its variance approaches that of Crowds.

Finally, we also simulated a TTL-based variant of the RADU(150,2,30,130) algorithm, yielding $\bar{l} = 3.97$ and $\sigma^2 = 6.86$. Fig. 7 compares RADU-Crowds against other D-Crowds. The anonymity curve of RADU-Crowds closely follows $\Gamma(2.32, 1.72)$-Crowds with equal variance $\sigma^2 = 6.86$, once again confirming that anonymity is a function of path length variance.

6 Conclusions

The original Crowds is one of the most simple and elegant schemes proposed to provide anonymity, and over the years it has received significant attention from

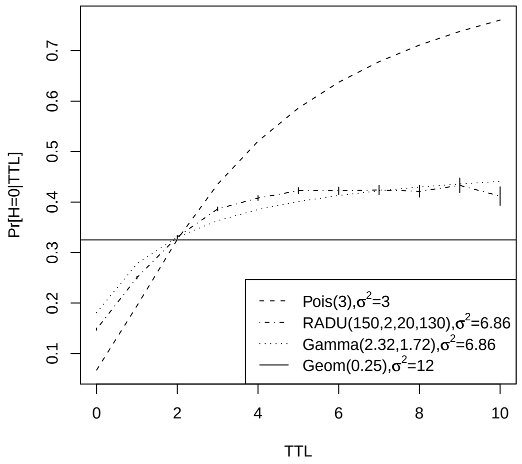

Fig. 7. Deanonymization probabilities $\Pr[H = 0|\text{TTL} = \text{ttl}]$ for different D-Crowds with fixed mean $\bar{l} = 4$.

the anonymity community. We conclusively show for the first time that its path lengths, and associated latency, is also optimal in providing anonymity within its system constraints. To provide better guarantees, more robust source routing is required to limit the adversary from learning the remaining time-to-live of intercepted messages. This advantage would be provided though cryptography, which would turn Crowds closer to a mix-network scheme [1].

Our analysis of the ADU and RADU schemes demonstrate practically that proposals with different path length distributions will provide weaker guarantees. Previous analysis of these schemes did not take into account all information leaked, and overlooked the fact that anonymity systems have to protect against a corrupt end server, and thus drew mistaken conclusions about their safety. Once more it becomes clear that even small modifications to anonymity systems need to be accompanied by thorough traffic analysis, to demonstrate their security. We have to be very suspicious of proposals that go against the simple rule of thumb: the less latency and variance in latency, the less anonymity a system is likely to provide.

Furthermore, we show that the simple D-Crowds TTL based scheme, can be adapted to accommodate any path length distribution, while leaking the minimal amount of information. Our probabilistic model of D-Crowds, and the Bayesian analysis to describe the probability of success of the adversary guarantees that.

Acknowledgements: Emilia Käsper thanks the Computer Laboratory of the University of Cambridge for hosting her. The authors would like to thank J.P. Muñoz-Gea for his clarifications in the functioning of the ADU algorithm. This work was supported in part by the European Commission through the ICT Programme under Contract ICT-2007-216646 ECRYPT II, the FWO Flanders project nr. G.0317.06 Linear Codes and Cryptography, the IWT SBO ADAPID project, the Concerted Research Action (GOA) Ambiorics 2005/11 of the Flemish Government and the IAP Programme P6/26 BCRYPT. C. Troncoso is a research assistant of the Fund for Scientific Research in Flanders (FWO).

References

1. David Chaum. Untraceable electronic mail, return addresses, and digital pseudonyms. *Communications of the ACM*, 4(2), February 1981.
2. Claudia Diaz, Stefaan Seys, Joris Claessens, and Bart Preneel. Towards measuring anonymity. In Roger Dingledine and Paul F. Syverson, editors, *Proceedings of Privacy Enhancing Technologies, 2nd International Workshop, PET 2002*, volume 2482 of *Lecture Notes in Computer Science*, pages 54–68, San Francisco,CA,USA, 2002. Springer-Verlag.
3. Claudia Diaz, Carmela Troncoso, and George Danezis. Does additional information always reduce anonymity? In Ting Yu, editor, *Proceedings of the 6th ACM workshop on Privacy in the electronic society (WPES 2007)*, pages 72–75, Alexandria,VA,USA, 2007. ACM.
4. J. P. Muñoz Gea, J. Malgosa-Sanahuja, P. Manzanares-Lopez, J. C. Sanchez-Aarnoutse, and J. Garcia-Haro. A low-variance random-walk procedure to provide anonymity in overlay networks. In *ESORICS '08: Proceedings of the 13th European Symposium on Research in Computer Security*, volume 5283 of *Lecture Notes in Computer Science*, pages 238–250. Springer-Verlag, 2008.
5. Michael Reiter and Aviel Rubin. Crowds: Anonymity for web transactions. *ACM Transactions on Information and System Security*, 1(1):66–92, 1998.
6. Andrei Serjantov and George Danezis. Towards an information theoretic metric for anonymity. In Roger Dingledine and Paul F. Syverson, editors, *Proceedings of Privacy Enhancing Technologies, 2nd International Workshop, PET 2002*, volume 2482 of *Lecture Notes in Computer Science*, pages 41–53, San Francisco,CA,USA, 2002. Springer-Verlag.
7. Matthew K. Wright, Micah Adler, Brian Neil Levine, and Clay Shields. The predecessor attack: An analysis of a threat to anonymous communications systems. *ACM Transactions on Information and System Security (TISSEC)*, 7(4):489–522, 2004.

A Optimality proof for Crowds

Theorem A1 *For an arbitrary distribution $D(l)$ over path lengths, if for all f, $0 < f < 1$,*

$$\mathsf{Adv}^f(D) \leq \mathsf{Adv}^f(Geom_p),$$

then

$$\mathbb{E}(D) \geq \mathbb{E}(Geom_p).$$

Proof. The fact that the advantage of the adversary for Crowds with an arbitrary distribution $D(l)$ is smaller than for Crowds with a specific geometric distribution

$\mathrm{Geom}_p(l) = \bar{f}^h f$ means, from eq. 6, that:

$$\forall \mathrm{ttl}. \qquad (1 - \bar{p}\bar{f}) \geq \frac{D(\mathrm{ttl})}{\sum_{h \geq 0} D(\mathrm{ttl} + h)\bar{f}^h}. \tag{7}$$

By Lemma A2 we know that the condition above implies that:

$$\forall \mathrm{ttl}. \qquad D(\mathrm{ttl}) \leq pF(\mathrm{ttl}), \tag{8}$$

where $F(l)$ is related to the cumulative distribution of $D(l)$, by $F(l) = \sum_{k \geq l} D(k)$. We express the expectation of $D(l)$ as a sum of cumulative distributions and use the inequality from Lemma A2 twice to prove our theorem.

$$\begin{aligned}
\mathbb{E}(D(l)) = \sum_{l \geq 0} lD(l) = \sum_{l \geq 0} \sum_{k \leq l} D(l) = \sum_{k \geq 0} \sum_{k \leq l} D(l) = \sum_{k > 0} F(k) = \sum_{l > 0} F(l) \\
\geq \sum_{l > 0} \frac{D(l)}{p} = \frac{1 - D(0)}{p} \geq \frac{1 - p}{p} = \mathbb{E}(\mathrm{Geom}_p(l))
\end{aligned}$$

and therefore $\mathbb{E}(D(l)) \geq \mathbb{E}(\mathrm{Geom}_p(l))$. QED.

Lemma A2 *We show that,*

$$\forall ttl. (1 - \bar{p}\bar{f}) \geq \frac{D(ttl)}{\sum_{h \geq 0} D(ttl + h)\bar{f}^h} \Rightarrow \forall ttl. D(ttl) \leq pF(ttl).$$

Proof. We start from the left hand side of the implication, and rearrange terms:

$$\begin{aligned}
D(\mathrm{ttl}) &\leq (1 - \bar{p}\bar{f}) \sum_{h \geq 0} D(h + \mathrm{ttl})\bar{f}^h \qquad (9) \\
\sum_{k \geq \mathrm{ttl}} D(\mathrm{ttl}) &\leq (1 - \bar{p}\bar{f}) \sum_{h \geq 0} \bar{f}^h \sum_{k \geq \mathrm{ttl}} D(h + \mathrm{ttl}) \\
F(\mathrm{ttl}) &\leq (1 - \bar{p}\bar{f}) \sum_{h \geq 0} \bar{f}^h F(h + \mathrm{ttl}) \\
F(\mathrm{ttl}) &\leq (1 - \bar{p}\bar{f}) \left[F(\mathrm{ttl}) + F(\mathrm{ttl} + 1)\bar{f} + F(\mathrm{ttl} + 2)\bar{f}^2 + \ldots\right] \\
F(\mathrm{ttl}) &\leq (1 - \bar{p}\bar{f}) \left[F(\mathrm{ttl}) + (F(\mathrm{ttl}) - D(\mathrm{ttl}))\,\bar{f} + \right. \\
&\qquad \left. + \left(F(\mathrm{ttl}) - \sum_{k < 2} D(k + \mathrm{ttl})\right)\bar{f}^2 + \ldots\right] \\
F(\mathrm{ttl}) &\leq (1 - \bar{p}\bar{f}) \left[F(\mathrm{ttl}) \left(\sum_{l \geq 0} \bar{f}^l\right) - \left(\sum_{l \geq 0} \sum_{k < l} D(k + \mathrm{ttl})\bar{f}^l\right)\right].
\end{aligned}$$

We now change the indexes of the double summation, to their equivalent conditions,

$$\begin{aligned}
F(\mathrm{ttl}) &\leq (1-\bar{p}\bar{f})\left[F(\mathrm{ttl})\left(\sum_{l\geq 0}\bar{f}^{l}\right)-\left(\sum_{k\geq 0}\sum_{l\geq k+1}D(k+\mathrm{ttl})\bar{f}^{l}\right)\right]\\
F(\mathrm{ttl}) &\leq (1-\bar{p}\bar{f})\left[F(\mathrm{ttl})\left(\sum_{l\geq 0}\bar{f}^{l}\right)-\left(\sum_{k\geq 0}D(k+\mathrm{ttl})\bar{f}^{k+1}\sum_{l\geq k+1}\bar{f}^{l-k-1}\right)\right]\\
F(\mathrm{ttl}) &\leq (1-\bar{p}\bar{f})\left[\frac{1}{1-\bar{f}}F(\mathrm{ttl})-\left(\frac{\bar{f}}{1-\bar{f}}\sum_{k\geq 0}D(k+\mathrm{ttl})\bar{f}^{k}\right)\right]\\
\frac{\bar{f}(1-\bar{p}\bar{f})}{1-\bar{f}}\sum_{k\geq 0}D(k+\mathrm{ttl})\bar{f}^{k} &\leq \left[\frac{1-\bar{f}\bar{p}}{1-\bar{f}}-1\right]F(\mathrm{ttl})=\frac{\bar{f}-\bar{f}\bar{p}}{1-\bar{f}}F(\mathrm{ttl})\\
(1-\bar{p}\bar{f})\sum_{k\geq 0}D(k+\mathrm{ttl})\bar{f}^{k} &\leq pF(\mathrm{ttl}).
\end{aligned}$$

Note that the last derivation is a bound on $(1-\bar{p}\bar{f})\sum_{k\geq 0}D(k+\mathrm{ttl})$. From eq. 9 we derive

$$D(\mathrm{ttl})\leq(1-\bar{p}\bar{f})\sum_{k\geq 0}D(k+\mathrm{ttl})\leq pF(\mathrm{ttl}),$$

which concludes the proof of the lemma.