An analysis of trust in anonymity networks in the presence of adaptive attackers

Sardaouna Hamadou, Vladimiro Sassone, Mu Yang

Under consideration for publication in Math. Struct. in Comp. Science

An analysis of trust in anonymity networks in the presence of adaptive attackers

SARDAOUNA HAMADOU[1†], VLADIMIRO SASSONE[2] and MU YANG[3]

[1] *INRIA, LIX, Ecole Polytechnique.*
[2] *Cybersecurity Centre, University of Southampton.*
[3] *ECS, University of Southampton.*

Received July 2012

Anonymity is a security property of paramount importance, as we move steadily towards a wired, online community. Its import touches upon subjects as different as eGovernance, eBusiness and eLeisure, as well as personal freedom of speech in authoritarian societies. Trust metrics are used in anonymity networks to support and enhance reliability in the absence of verifiable identities, and a variety of security attacks currently focus on degrading a user's trustworthiness in the eyes of the other users. In this paper, we analyse the privacy guarantees of the Crowds anonymity protocol, with and without onion forwarding, for standard and adaptive attacks against the trust level of honest users.

1. Introduction

Protecting online privacy is an essential part of today's society and its importance is increasingly recognised as crucial in many fields of computer-aided human activity, such as eVoting, eAuctions, bill payments, online betting and electronic communication. One of the most common mechanisms for privacy is *anonymity*, which generally refers to the condition of being unidentifiable within a given set of subjects, known as the *anonymity set*.

Many schemes have been proposed to enforce privacy through anonymity networks (e.g. (Chaum, 1981; Jakobsson, 1999; Neff, 2001; Freedman and Morris, 2002; Nambiar and Wright, 2006)). Yet, the open nature of such networks and the unaccountability which results from the very idea of anonymity, make the existing systems prone to various attacks (e.g. (Hopper et al., 2010; McLachlan et al., 2009; Murdoch and Danezis, 2005; Dingledine et al., 2004)). An honest user may have to suffer repeated misbehaviour (e.g., receiving infected files) without being able to identify the malicious perpetrator. Keeping users anonymous also conceals their trustworthiness, which in turn makes the information exchanged through system transactions untrustworthy as well. Consequently, a considerable amount of research has recently been focussing on the development of trust-and-reputation-based metrics aimed at enhancing the reliability of anonymity networks (Damiani et al., 2003; Damiani et al., 2002; Dingledine et al., 2001; Dingledine and Syverson, 2002; Singh and Liu, 2003; Wang and Vassileva, 2003).

[†] Partly supported by the project ANR-09-BLAN-016901 PANDA: PArallel aNd Distributed Analysis

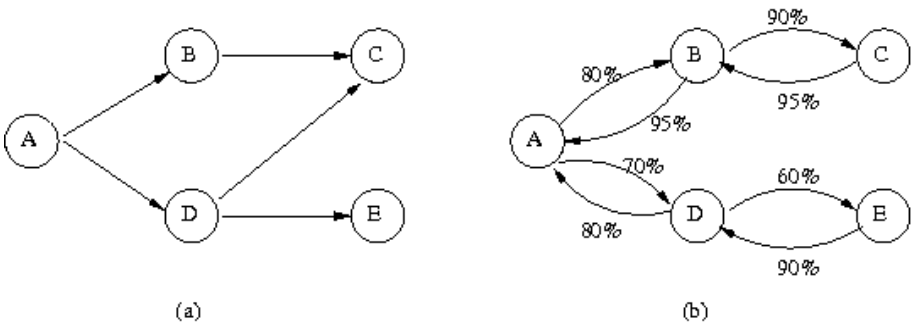


Fig. 1. Trust networks (Backes et al., 2010)

Developing an appropriate trust metric for anonymity is very challenging, due to the fact that trust and anonymity are seemingly conflicting notions. Consider for instance the trust networks of Figure 1. In (a) peer A trusts B and D, who both trust C. Assume now that C wants to request a service from A *anonymously*, by proving her trustworthiness to A (i.e., the existence of a trust link to it). If C can prove that she is trusted by D without revealing her identity (using e.g. a *zero-knowledge* proof (Backes et al., 2010)), then A cannot distinguish whether the request originated from C or E. Yet, A's trust in D could be insufficient to obtain that specific service from A. Therefore, C could strengthen her request by proving that she is trusted by both D and B. This increases the trust guarantee. Unfortunately, it also decreases C's anonymity, as A can compute the intersection of peers trusted by both D and B, and therefore restrict the range of possible identities for the request's originator, or even identify C uniquely. Indeed, consider Figure 1(b). Here the trust level between two principals is weighted, and trust between two non-adjacent principals is computed by multiplying the values over link sequences in the obvious way. Assume that the reliability constraint is that principal X can send (resp. receive) a message to (from) principal Y if and only if her trust in Y is not lower than 60%. Principal E can therefore only communicate through principal D. So, assuming that trust values are publicly known, E cannot possibly keep her identity from D as soon as she tries to interact at all. These examples document the existence of an inherent trade-off between anonymity and trust. The fundamental challenge is to achieve an appropriate balance between practical privacy, and acceptable network performance.

Community-based reputation systems are becoming increasingly popular both in the research literature and in practical applications. They are systems designed to estimate the trustworthiness of principals participating in some activity, as well as predict their future behaviour. Metrics for trustworthiness are primarily based on *peer-review*, where peers can rate each other according to the quality they experienced in their past mutual interactions (Krukow et al., 2008; ElSalamouny et al., 2009a; ElSalamouny et al., 2009b). A good reputation indicates a peer's good past behaviour, and is reflected in a high trust value. Recent research in this domain has raised fundamental issues in the design of reputation management systems for anonymous networks. In particular,

1 what metrics are suitable for computing trust for a given application field?

2 how to ensure the integrity of the peers' trust values, i.e., how to securely store and access trust values against malicious peers?
3 how to ensure that honest users accurately rate other members?

The latter issue requires a mechanism to distinguish a user's bad behaviour resulting from her being under attack, from a deliberately malicious behaviour. This is a challenging and fundamental problem. Indeed, if we cannot accurately tell these two situations apart, malicious users will target honest members in order to deteriorate their performance, and hence reduce other members' trust in them, while maintaining their apparent good behaviour. Thus, honest users may in the long term end up enjoying very low trust levels, while attackers might see their reputation increased, and so they increase their probability of being trusted by others. Over time this will, of course, severely affect the system's anonymity performance. Nevertheless, although a considerable effort has recently been devoted to tackle the first two issues (Damiani et al., 2002; Damiani et al., 2003; Singh and Liu, 2003), to the best of our knowledge the latter has been so far relatively ignored.

In this paper we investigate the effect of attacks to the trust level of honest users on the security of existing anonymity networks, such as the Reiter and Rubin's Crowds protocol (Reiter and Rubin, 1998) and onion routing networks (Dingledine et al., 2004).

The Crowds protocol allows Internet users to perform anonymous web transactions by sending their messages through a random chain of users participating in the protocol. Each user in the '*crowd*' must establish a path between her and a set of servers by selecting randomly some users to act as routers (or forwarders). The formation of such routing paths is performed so as to guarantee that users do not know whether their predecessors are message originators or just forwarders. Each user only has access to messages routed through her. It is well known that Crowds cannot ensure strong anonymity in presence of corrupt participants (Reiter and Rubin, 1998; Chatzikokolakis and Palamidessi, 2006), yet when the number of corrupt users is sufficiently small, it provides a weaker notion of anonymity known as *probable innocence*. Informally, a sender is probably innocent if to an attacker she is no more likely to be the message originator than not to be.

Networks based on Onion Routing are distributed anonymising networks that use *onion routing* (Syverson et al., 1997) to provide anonymity to their users. Similarly to Crowds, users choose randomly a path through the network in which each node knows its predecessor and successor, but no other node. The main difference with respect to Crowds is that traffic flows through the path in cells, which are created by the initiator by successively encrypting the message with the session keys of the nodes in the path, in reverse order. Each node in the act of receiving the message peels the topmost layer, discovers who the next node is, and then relays it forward. In particular, only the last node can see the message in clear and learn its final destination.

In the paper we propose two variants of the congestion attacks in the literature, aimed at deteriorating the trust level of target users in different extension of the Crowds protocol. More specifically, we first extend the protocol so that trust is used to inform the selection of forwarding users. Our analysis of this extension shows that a DoS type attack targeting a user who initially enjoys satisfactory anonymity protection, may threaten her privacy, as her trust level quickly decreases over the time. We then extend the protocol further with a more advanced message forwarding technique, namely onion routing. While this extension offers much better protection

than the previous one, our analysis ultimately shows that it suffers from similar DoS attacks as the others.

A second major contribution of this paper is the study of '*adaptive*' attackers. Adaptive attacks are carried out by malicious users who, rather than immediately reporting detected forwarders, attempt to travel back along the anonymity chain (e.g., by successive brute-force attacks on its nodes) so as to increase the probability that the node they eventually report (i.e., the first node they fail to compromise) actually is the message originator.

Related work. Anonymity networks date back thirty years, to when Chaum introduced the concept of *Mix-net* (Chaum, 1981) for anonymous communications, where different sources send encrypted messages to a mix which forwards them to their respective destinations. Various designs (Syverson et al., 1997; Reiter and Rubin, 1998; Abe, 1998; Dingledine et al., 2004; Neff, 2001; Ohkubo and Abe, 2000; Freedman and Morris, 2002; Nambiar and Wright, 2006; Rennhard and Plattner, 2002) have since been proposed to improve Chaum's mixes, e.g., by combinations of artificial delays, variation in message ordering, encrypted message formats, message batching, and random chaining of multiple mixes.

A variety of attacks (Back et al., 2001; Borisov et al., 2007; Hopper et al., 2010; Evans et al., 2009; McLachlan and Hopper, 2008; McLachlan et al., 2009; Murdoch and Danezis, 2005; Pappas et al., 2008; Dingledine et al., 2004) have since been discovered against such anonymity systems. Those most related to the present work are the so-called *congestion* or *clogging* attacks. In an congestion attack, the adversary monitors the flow through a node, builds paths through other nodes, and tries to use all of their available capacity (Back et al., 2001). The idea is that if the congested node belongs to the monitored path, the variation in the messages' arrival times will reflect at the monitored node. In (Murdoch and Danezis, 2005), Murdoch and Danezis describe a congestion attack that may allow them to reveal all Tor's routers (cf. (Dingledine et al., 2004)) involved in a path. However, although their attack works well against a Tor network of a relatively small size, it fails against networks of typical sizes, counting nodes in the thousands. More recently, Evans *et al.* (Evans et al., 2009) improved Murdoch and Danezis's attack so as to practically de-anonymise Tor's users in currently deployed system. A similar attack against MorphMix (Rennhard and Plattner, 2002) was recently described by Mclachlan and Hopper (McLachlan and Hopper, 2008), proving wrong the previously held view that MorphMix is robust against such attacks (Wiangsripanawan et al., 2007). Finally, a congestion attack is used by Hopper *et al.* (Hopper et al., 2010) to estimate the latency between the source of a message and its first relay in Tor. In *loc. cit.* the authors first use a congestion attack to identify the path, and then create a parallel circuit throughout the same path to make their measurements.

Numerous denial of service (DoS) attacks have been reported in the literature. In particular, the *'packet spinning'* attack of (Pappas et al., 2008) tries to lure users into selecting malicious relays by targeting honest users by DoS attacks. The attacker creates long circular paths involving honest users and sends large amount of data through the paths, forcing the users to employ all their bandwidth and then timing out. These attacks motivate the demand for mechanisms to enhance the reliability of anonymity networks. In recent years, a considerable amount of research has been focusing on defining such mechanisms. In particular, trust-and-reputation-based metrics are quite popular in this domain (Backes et al., 2010; Damiani et al., 2003; Damiani et al., 2002; Dingledine et al., 2001; Dingledine and Syverson, 2002; Singh and Liu, 2003; Wang and

Vassileva, 2003). Enhancing the reliability by trust, not only does improve the system's usability, but may also increase its anonymity guarantee. Indeed, a trust-based selection of relays improves both the reliability and the anonymity of the network, by delivering messages through 'trusted' routers. Moreover, the more reliable the system, the more it may attract users and hence improve the anonymity guarantee by growing the anonymity set. Introducing trust in anonymity networks does however open the flank to novel security attacks, as we prove in this paper.

In a recent paper of ours (Sassone et al., 2010a) we have analysed the anonymity provided by Crowds extended with some trust information, yet against a completely different threat model. The two papers differ in several ways. Firstly, (Sassone et al., 2010a) considers a global and 'credential-based' trust notion, unlike the individual-and-reputation-based trust considered here. Secondly, in (Sassone et al., 2010a) we considered an attack scenario where all protocol members are honest but vulnerable to being corrupted by an external attacker. The global and fixed trust in a user contrasts with the local and dynamic trust of this paper, as is meant to reflect the user's degree of resistance against corruption, that is the probability that the external attacker will fail to corrupt her. The paper derives necessary and sufficient conditions to define a 'social' policy of selecting relays nodes in order to achieve a given level of anonymity protection to all members against such attackers, as well as a 'rational' policy maximise one's own privacy.

Structure of the paper. The paper is organised as follows: in §2 we fix some basic notations and recall the fundamental ideas of the Crowds protocol and its properties, including the notion of probable innocence. In §3 we present our first contribution: the Crowds protocol extended with trust information in the form of a forwarding policy of its participating members, and the privacy properties of the resulting protocol are studied; §4 repeats the analysis for an extension of the protocol with a more advanced forwarding technique inspired by onion routing. Finally, §5 introduces a new 'adaptive' attack scenario, and presents some preliminary results on its analysis, both for the protocol with and without onion forwarding.

This paper is a full and extended version of (Sassone et al., 2010b), where the bulk of the present results were first reported in succinct form. The analysis of adaptive attacks in *loc. cit.* is however incomplete, in that it assumes that attackers who travel back over a path towards its originator, need to corrupt each honest node each time they meet her. Arguably, this is not so. Typically a node j will act according to a routing table, say T_j. This will contain for each path's *id* a translation *id* and a forwarding address (either another user, or the destination server) and, in the case of onion forwarding, the relevant encryption key. (Observe that since path's *id* are translated at each step, j may not be able to tell whether or not two entries in T_j actually correspond to a same path and, therefore, may not know how many times she occurs on each path.) It is reasonable to assume that upon corruption an attacker c will seize T_j, so that if she ever reaches j again, c will find all the information to continue the attack just by inspecting T_j.

This full exposition significantly improves the treatment of adaptive attackers in the general case. A substantial amount of new work was devoted to §5, which indeed was extensively rewritten. More precisely, using the formal framework of (Sassone et al., 2010b), some ingenuity and a lot of combinatorics, one can write an infinite series to compute the probability of success for an adaptive attack, containing a term for each possible occurrence pattern of honest users and attackers in the path. The reason why this is sufficient, is that the only relevant factor in the com-

putation is how many times each honest user appears in between the attacker at the end of the path and the detected node. Nothing in that, however, indicates how to simplify that series so as to distill a usable formula. This is indeed the main step forward we make here with respect to (Sassone et al., 2010b): by indexing our calculations on the set of users compromised by the adaptive attack, we reach a final presentation for our analysis which we believe is sufficiently simple and elegant. A further novelty with respect to *loc. cit.* is the quantitative comparison between the probability of success of adaptive attacks versus standard ones at the end of §5.

2. Crowds

In this section, we briefly revise the Crowds protocol and the notion of probable innocence.

2.1. *The protocol*

Crowds is a protocol proposed by Reiter and Rubin in (Reiter and Rubin, 1998) to allow Internet users to perform anonymous web transactions by protecting their identities as originators of messages. The central idea to ensure anonymity is that the originator forwards the message to another, randomly-selected user, which in turn forwards the message to a third user, and so on until the message reaches its destination (the end server). This routing process ensures that, even when a user is detected sending a message, there is a substantial probability that she is simply forwarding it on behalf of somebody else.

More specifically, a crowd consists of a *fixed* number of users participating in the protocol. Some members (users) of the crowd may be corrupt (the *attackers*), and they collaborate in order to discover the originator's identity. The purpose of the protocol is to protect the identity of the message originator from the attackers. When an *originator* –also known as *initiator*– wants to communicate with a server, she creates a random *path* between herself and the server through the crowd by the following process.

— *Initial step:* the initiator selects randomly a member of the crowd (possibly herself) and forwards the request to her. We refer to the latter user as the *forwarder*.
— *Forwarding steps:* a forwarder, upon receiving a request, flips a *biased* coin. With probability $1 - p_f$ she delivers the request to the end server. With probability p_f she selects randomly a new forwarder (possibly herself) and forwards the request to her. The new forwarder repeats the same forwarding process.

The response from the server to the originator follows the same path in the opposite direction. Users (including corrupt users) are assumed to only have access to messages routed through them, so that each user only knows the identities of her immediate predecessor and successor in the path, as well as the server.

2.2. *Probable innocence*

Reiter and Rubin have proposed in (Reiter and Rubin, 1998) a hierarchy of anonymity notions in the context of Crowds. These range from '*absolute privacy*,' where the attacker cannot perceive the presence of an actual communication, to '*provably exposed*,' where the attacker can

prove a sender-and-receiver relationship. Clearly, as most protocols used in practice, Crowds cannot ensure absolute privacy in presence of attackers or corrupted users, but can only provide weaker notions of anonymity. In particular, in (Reiter and Rubin, 1998) the authors propose an anonymity notion called *probable innocence* and prove that, under some conditions on the protocol parameters, Crowds ensures the probable innocence property to the originator. Informally, they define it as follows:

> *A sender is probably innocent if, from the attacker's point of view, she appears no more likely to be the originator than to not be the originator.* (1)

In other words, the attacker may have reason to suspect the sender of being more likely than any other potential sender to be the originator, but it still appears at least as likely that she is not.

We use capital letters A, B to denote discrete random variables and the corresponding small letters a, b and calligraphic letters $\mathcal{A}$, $\mathcal{B}$ for their values and set of values respectively. We denote by $P(a), P(b)$ the probabilities of a and b respectively and by $P(a, b)$ their *joint probability*. The *conditional probability* of a given b is defined as

$$P(a \,|\, b) = \frac{P(a, b)}{P(b)} \,.$$

Bayes Theorem relates the conditional probabilities $P(a \,|\, b)$ and $P(a \,|\, b)$ as follows

$$P(a \,|\, b) = \frac{P(b \,|\, a)\, P(a)}{P(b)} \,. \tag{2}$$

Let n be the number of users participating in the protocol and let c and $n - c$ be the number of the corrupt and honest members, respectively. Since anonymity makes only sense for honest users, we define the set of anonymous events as $\mathcal{A} = \{a_1, a_2, \ldots, a_{n-c}\}$, where a_i indicates that user i is the initiator of the message.

As it is usually the case in the analysis of Crowds, We assume that attackers will always deliver a request to forward immediately to the end server, since forwarding it any further cannot help them learn anything more about the identity of the originator. Thus in any given path, there is at most one detected user: the first honest member to forward the message to a corrupt member. Therefore we define the set of observable events as $\mathcal{O} = \{o_1, o_2, \ldots, o_{n-c}\}$, where o_j indicates that user j forwarded a message to a corrupted user. In this case we also say that user j *is detected* by the attacker.

Reiter and Rubin (Reiter and Rubin, 1998) formalise their notion of probable innocence via the conditional probability that the initiator is detected given that any user is detected at all. This property can be written in our setting as the probability that user i is detected given that she is the initiator, that is the conditional probability $P(o_i \,|\, a_i)$.[†] Probable innocence holds if

$$\forall i.\ P(o_i \,|\, a_i) \leq \frac{1}{2} \tag{3}$$

[†] We are only interested in the case in which a user is detected, although for the sake of simplicity we shall not note that condition explicitly.

Reiter and Rubin proved in (Reiter and Rubin, 1998) that, in Crowds, the following holds:

$$P(o_j \mid a_i) \;=\; \begin{cases} 1 - \dfrac{n-c-1}{n} p_f & i = j \\ \dfrac{1}{n} p_f & i \neq j \end{cases} \tag{4}$$

Therefore, probable innocence (3) holds if and only if

$$n \geq \frac{p_f}{p_f - 1/2}\,(c+1) \quad \textit{and} \quad p_f \geq \frac{1}{2}$$

As previously noticed in several papers (e.g., (Chatzikokolakis and Palamidessi, 2006)), there is a mismatch between the idea of probable innocence expressed informally by (1), and the property actually proved by Reiter and Rubin, viz. (3). The former seems indeed to correspond to the following interpretation given by Halpern and O'Neill (Halpern and O'Neill, 2005):

$$\forall i, j.\; P(a_i \mid o_j) \leq \frac{1}{2}\,. \tag{5}$$

In turn, this has been criticised for relying on the probability of users' actions, which the protocol is not really in control of, and for being too strong. However, both (3) and (5) work satisfactorily for Crowds, thanks to its high symmetry: in fact, they coincide under its standard assumption that the *a priori* distribution is uniform, i.e., that each honest user has equal probability of being the initiator, which we follow in this paper too.

We remark that the concept of probable innocence was recently generalised in (Hamadou et al., 2009). Instead of just comparing the probability of being innocent with the probability of being guilty, the paper focusses on the degree of innocence. Formally, given a real number $\alpha \in [0, 1]$, a protocol satisfies α-probable innocence if and only if

$$\forall i, j.\; P(a_i \mid o_j) \leq \alpha \tag{6}$$

Clearly α-probable innocence coincides with standard probable innocence for $\alpha = 1/2$.

3. Trust in Crowds

In the previous section, we have revised the fundamental ideas of the Crowds protocol and its properties under the assumption that all members are deemed equal. However, as observed in §1, this is clearly not a realistic assumption for today's open and dynamic systems. Indeed, as shown by the so-called 'packet spinning' attack (Pappas et al., 2008), malicious users can attempt to make honest users select bogus routers by causing legitimate routers time out. The use attributes relating to some level of *trust* is therefore pivotal to enhance the reliability of the system. In this section, we firstly reformulate the Crowds protocol under a novel scenario where the interaction between participating users is governed by their level of mutual trust; we then evaluate its privacy guarantees using property (6). We then focus on the analysis of attacks to the trust level of honest users and their impact on the anonymity of the extended protocol. Finally, we investigate the effect of a congestion attack (Evans et al., 2009) to the trust level of honest users.

3.1. Crowds *extended*

We now extend the Crowds protocol to factor in a notion of trust for its participating members. To this end, we associate a trust level t_{ij} to each pair of users i and j, which represents user i's trust in user j. Accordingly, each user i defines her *policy of forwarding* to other members (including herself) based on her trust in each of them. A policy of forwarding for user i is a discrete probability distribution $\{q_{i1}, q_{i2}, \cdots, q_{in}\}$, where q_{ij} denotes the probability that i chooses j as the forwarder, once she has decided to forward the message.

A natural extension of Crowds would obviously allow the initiator to select her first forwarder according to her own policy, and then leave it to the forwarder to pick the next relay, according to the forwarder's policy. This would however have the counter-intuitive property that users may take part in the path which are not trusted by the initiator, just because they are trusted by a subsequent forwarder. We rather take the same view as most current systems, that the initiator is in charge of selecting the entire path which will carry her transactions. In fact, this allows the initiator to enhance both performance and privacy by routing messages through trusted peers, which is the primary goal of adding a trust mechanism to anonymity protocols. When an initiator wants to communicate with a server, she selects a random *path* through the crowd between herself and the server by the following process.

— *First forwarder:* with probability q_{ij} the initiator i selects a member j of the crowd (possibly herself) according to her policy of forwarding $\{q_{i1}, q_{i2}, \cdots, q_{in}\}$.
— *Subsequent forwarders:* the initiator flips a *biased* coin; with probability $1 - p_f$ the current forwarder will be the last on the path, referred to as the *path's exit user*. Otherwise, with probability $p_f \times q_{ik}$, she selects k (possibly herself) as the next forwarder in the path; and so on until a path's exit user is reached.

The original Crowds does not reveal to a(n adversary) router any information about a path, apart from the previous and the next router; in particular, an adversary router does not learn how far it sits from the message destination. Of course we need our extension to preserve this property, as it makes traffic analysis substantially harder for the adversary. For this reason we resort to a mechanism proposed by Camenisch and Lysyanskaya (Camenisch and Lysyanskaya, 2005) which was formally proved to enjoy this property. The idea is as follows. Once the initiator has selected her random forwarders, she generates a session key for each of them. She then creates a so-called *onion*. This is a data structure consisting of as many 'layers' as forwarders. The ith layer contains the remaining part, say O_i, of the onion to be forwarded and the session key K_i of the i^{th} forwarder, say F_i, encrypted with its public key. Once decrypted, the session key K_i allows F_i to peels the topmost layer and discover the identity of next forwarder (if any). Importantly, F_i will pad the resulting onion with as many bits as required to make O_{i+1} of the same size as O_i. The padding will be such that any deviation from these rules will be noticed at subsequent routers' integrity check. This ensures that each intermediate forwarder F_i cannot assess its distance from the exit router and from the initiator. For suitably long chains, this guarantees that intermediary routers only know her immediate predecessor and successor. We refer the reader to (Camenisch and Lysyanskaya, 2005) for a full description of the mechanism. To notify the initiator that the path is fully created, the exit node (i.e., the last router on the path) sends an encrypted message with her session key travelling back the path to the initiator.

Once the path is formed, messages from the initiator to the server are sent in the same way as

in the normal Crowds. Thus, all the nodes in the path have access to the contain of the message and, obviously, to the end server. In particular, this means that the notion of detection remains the same in the extended protocol as in the original one.

Adversary model. We assume here that users' trust values are personal and private.[‡] Hence, the attackers have no knowledge of honest users' forwarding policies. This implies that they have no reasonable way to compute the probability $P(o_j \mid a_i)$ of a specific user j being detected given that user i initiates a transaction, or conversely the probability $P(a_i \mid o_j)$ of a user i being the initiator given that j is detected. These quantities are required by the probable innocence metrics (3) and (5). Since building paths does reveals no additional information, the process gives attackers no clear strategy to determine the most likely initiator. We therefore focus here on the typical (lazy) adversaries which bet on what they have in hand, i.e. that the most likely initiator is the detected user. We then evaluate the anonymity guaranteed to a user i by the probability that the attacker's guess is correct, that is the probability $P(a_i \mid o_i)$.

Now we use our probabilistic framework to evaluate Crowds extended protocol. We start by evaluating the conditional probability $P(o_j \mid a_i)$. Let η_i (resp. $\zeta_i = 1-\eta_i$) be the overall probability that user i chooses a honest (resp. corrupt) member as a forwarder. Then we have the following result.

Proposition 1.

$$P\left(o_j \mid a_i\right) \;=\; \zeta_i \epsilon_{ij} + \frac{q_{ij}\zeta_i p_f}{1-\eta_i p_f}\,,$$

where $\eta_i = \sum_{k\leq(n-c)} q_{ik}$, $\zeta_i = \sum_{k\leq c} q_{ik}$ and $\epsilon_{ij} \;=\; \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases}$

Proof. Let k denote the position occupied by the first honest user preceding an attacker on the path, with the initiator occupying position zero. Let $P(o_j \mid a_i)_{(k)}$ denote the probability that user j is detected exactly at position k. Only the initiator can be detected at position zero, and the probability that this happens is equal to the overall probability that the initiator chooses a corrupt member as a forwarder. Therefore

$$P\left(o_j \mid a_i\right)_{(0)} \;=\; \begin{cases} \zeta_i & i = j \\ 0 & i \neq j \end{cases}$$

Now the probability that j is detected at position $k > 0$ is given by

— the probability that she decides to forward k times and picks $k-1$ honest users, i.e.,$p_f^{k-1}\eta_i^{k-1}$ (recall that at the initial step she does not flip the coin),
— times the probability of choosing j as the kth forwarder, i.e., q_{ij},
— times the probability that she picks any attacker at stage $k+1$, i.e., $\zeta_i p_f$.

[‡] As this is one of the first papers (Johnson and Syverson, 2009; Sassone et al., 2010b; Johnson et al., 2011) on trust in anonymity networks, we feel that such simplifying assumption is justified. The more general case where attackers can attempt to infer honest users' trust values will be investigated in future work. Note that compared with (Sassone et al., 2010b) and our work, (Johnson and Syverson, 2009; Johnson et al., 2011) rely on a different notion of trust based on 'difficulty-of-compromise' rather than users' performance, although still personal and private.

Therefore

$$\forall k \geq 1, \;\; P\left(o_j \mid a_i\right)_{(k)} \;=\; \eta_i^{k-1} p_f^k q_{ij} \zeta_i$$

and hence

$$\begin{aligned} P\left(o_j \mid a_i\right) &= \sum_{k=0}^{\infty} P\left(o_j \mid a_i\right)_{(k)} \\ &= \zeta_i \epsilon_{ij} + \sum_{k=1}^{\infty} \eta_i^{k-1} p_f^k q_{ij} \zeta_i \\ &= \zeta_i \epsilon_{ij} + \sum_{k=0}^{\infty} \eta_i^{k} p_f^{k+1} q_{ij} \zeta_i \\ &= \zeta_i \epsilon_{ij} + p_f q_{ij} \zeta_i \sum_{k=0}^{\infty} \eta_i^k p_f^k \\ &= \zeta_i \epsilon_{ij} + \frac{q_{ij} \zeta_i p_f}{1 - \eta_i p_f} \,. \end{aligned}$$

□

An immediate consequence is that when user i initiates a transaction, user j is not detectable if and only if the initiator's policy of forwarding never chooses an attacker or j as forwarder.

Corollary 1. $P(o_j \mid a_i) = 0$ if and only if one of the following holds:

1 $\zeta_i = 0$;
2 $q_{ij} = 0$ and $i \neq j$.

Now, let us compute the probability of detecting a user $P(o_j)$. We assume a uniform distribution for anonymous events.

Proposition 2. If the honest members are equally likely to initiate a transaction, then

$$P(o_j) = \frac{1}{n-c}\left(\zeta_j + \sum_{i \leq (n-c)} \frac{q_{ij} \zeta_i p_f}{1 - \eta_i p_f} \right),$$

where ζ_j and η_i are defined as in Proposition 1.

Proof. Since the anonymous events are uniformly distributed then $P(a_i) = 1/(n-c)$ for all i.

Thus

$$\begin{aligned} P(o_j) &= \sum_{i\leq(n-c)} P\left(o_j \mid a_i\right) P(a_i) \\ &= \sum_{i\leq(n-c)} P\left(o_j \mid a_i\right) \frac{1}{n-c} \\ &= \frac{1}{n-c} \sum_{i\leq(n-c)} P\left(o_j \mid a_i\right) \\ &= \frac{1}{n-c} \sum_{i\leq(n-c)} \left(\zeta_i \epsilon_{ij} + \frac{q_{ij}\zeta_i p_f}{1-\eta_i p_f}\right) \\ &= \frac{1}{n-c}\left(\zeta_j + \sum_{i\leq(n-c)} \frac{q_{ij}\zeta_i p_f}{1-\eta_i p_f}\right). \end{aligned}$$

□

As one could expect, a user j is not detectable if both herself and any user i that might include j in her path never choose a corrupted member as a forwarder. Formally:

Corollary 2. $P(o_j) = 0$ if and only if

$$\zeta_j = 0 \quad \text{and} \quad \forall i.\, (\, q_{ij} = 0 \text{ or } \zeta_i = 0\,)\,.$$

Now from Propositions 1 and 2 and Bayes Theorem (2), we have the following expression for the degree of anonymity provided by the extended protocol, which holds when $P(o_j) \neq 0$.

Proposition 3. If the honest members are equally likely to initiate a transaction, then

$$P\left(a_i \mid o_j\right) = \frac{\zeta_i \epsilon_{ij} + \dfrac{q_{ij}\zeta_i p_f}{1-\eta_i p_f}}{\zeta_j + \displaystyle\sum_{k\leq(n-c)} \frac{q_{kj}\zeta_k p_f}{1-\eta_k p_f}}\,,$$

where ζ_i and η_j are defined as above.

It is now easy to see that if all honest users have uniform probability distributions as forwarding policies, the extended protocol reduces to the original Crowds protocol.

Corollary 3. If for all i and j, $q_{ij} = 1/n$, then $\eta_i = (n-c)/n$ and $\zeta_i = c/n$. Therefore

$$P\left(a_i \mid o_j\right) = \begin{cases} 1 - \dfrac{n-c-1}{n} p_f & i = j \\ \dfrac{1}{n} p_f & i \neq j \end{cases}$$

3.2. *On the security of extended* Crowds

Here we show that the absence of a uniform forwarding policy makes it very hard to achieve adequate anonymity protection both in the context of our "bet on the detected user" adversary

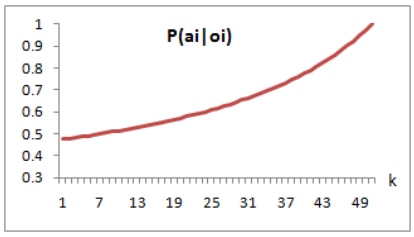
model and the probable innocence as defined by Halpern and O'Neill (5). Indeed consider the following instance of the protocol, where three honest users {1, 2, 3 } face a single attacker {4}. Assume that the honest users are aware of the malicious behaviour of 4, and choose their forwarding policies as follows: $p_f = 2/3$, and $q_{1j} = q_{2j} = 1/3$, and $q_{3j} = 0.33$ for all $j \leq 3$. In other words, the first two choose uniformly any honest users as a forwarder and never pick the attacker, whilst the third one may choose the attacker, though with a small probability $q_{34} = 0.01$. Thus, $\zeta_1 = \zeta_2 = q_{14} = q_{24} = 0$ and $\zeta_3 = q_{34} = 0.01$. It follows that $P(a_3 \mid o_3) = 1$, and the instance does not ensure anonymity to user 3, even though her policy is after all very similar to those of the other honest users. This is because if someone is detected, then user 3 is necessarily the initiator, as she is the only one who might possibly pick the attacker in her path.

Observe however that this instance of the protocol ensures probable innocence in Reiter and Rubin's formulation: indeed, $P(o_i \mid a_i) < 0.0165$ for all honest user i. The key difference at play here is that Halpern and O'Neill's definition is stronger, as it focuses on the probability that a specific user is the initiator once somebody has been detected, regardless of the probability of the detection event. On the other hand, Reiter and Rubin's formula measures exactly (the conditional probability of) the latter. This means that if the probability of detection is small, as in this case, systems may be classified as statistically secure even when one such detection event may lead to complete exposure for some initiators, as in this case.

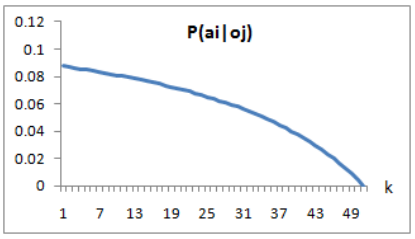
On the other hand, Reiter and Rubin's formulation, together with several other anonymity measures (Syverson et al., 2001; Feigenbaum et al., 2007; Smith, 2009; Hamadou et al., 2010) which take into account the probabilities of the observable events, show that the use of trust hugely improves the anonymity of this simple instance of the Crowds protocol, as the likelihood of someone being detected is almost zero. To determine which metric is most appropriate is out of the scope of this paper. Indeed, as we show below, attacks on trust levels could in fact (severely) impact the anonymity of the protocol regardless of the metric used.

Attackings trust. As already observed by its authors, Crowds is vulnerable to denial of service (DoS) attacks: it is enough that a single malicious router delays her forwarding action to severely hinder the viability of an entire path. This kind of attack is in fact hard for the initiator to respond to. Just because the creation of multiple paths by any single user substantially increases their security risk, the initiator has a strong incentive to keep using the degraded path. Indeed, it is advisable in Crowds to modify a path only when it has collapsed irremediably, e.g. due to a system crash of a router, or their quitting the crowd. In this case the path is re-routed from the node preceding the failed router. As a consequence, recent research has been devoted to developing '*trust metrics*' meant enhance the reliability of anonymity systems (Damiani et al., 2002; Damiani et al., 2003; Singh and Liu, 2003).

Although the primary goal of incorporating trust in anonymity networks is to 'enhance' the privacy guarantees by routing messages through *trusted* relays, preventing the presence of attackers in forwarding paths is in itself not sufficient. External attackers may in fact target honest users with DoS attacks independent of the protocol, to make them look unreliable and/or unstable. In this way, the target users will gradually loose others members' trust, whilst internal attackers may keep accruing good reputations. Thus, over the time the trust mechanisms may become counterproductive.



(a) $i = j = 7$



(b) $i \neq j = 7$

Fig. 2. Crowds extended

Let us illustrate an attack of this kind. Consider an instance of the protocol where seven honest users $\{1, 2, \cdots, 7\}$ face a single attacker $\{8\}$, assume that 7 is the honest user targeted by the attack, and that all users are equally likely to initiate a transaction. Recall that a path in Crowds remains fixed for a certain amount of time –typically one day– known as a *session*. In practice, *all* transactions initiated by a given user follow the same path, regardless of their destination servers. At the end of the session then, all existing paths are destroyed, new members can join the crowd, and each member willing to initiate anonymous transactions creates a new path. Trust level updates play therefore their role at the beginning of each session. For the purpose of this example, we assume that the protocol is equipped with mechanisms to detect unstable routers (e.g., by monitoring loss of messages, timeouts, variations in response time and so on); upon realising that her path is unstable, an initiator will notify all members of the identity of the unstable node (in this case 7).[§] When a node is reported as unstable, all other honest nodes decrease their trust in her at the beginning of the following session. For simplicity, we assume that all users start with the same trust level τ, and that the target user remains fixed over time. The following policies of forwarding are therefore in place for each session, with $n = 8$, $c = 1$ and $\tau = 50$.

$$q_{ij}^{(k)} = \begin{cases} \dfrac{1}{n} & i = 7 \\ \dfrac{\tau - k}{n \times \tau - k} & i \neq 7 \text{ and } j = 7 \\ \dfrac{\tau}{n \times \tau - k} & i \neq 7 \text{ and } j \neq 7\,. \end{cases}$$

In words, honest users other that the target decrease their trust in her by one and redistributed it uniformly to the remaining users. On the other hand, the target has no reason to change her trust, as there is no evidence to suspect anybody as the source of the external attack. Thus, her policy remains the same over the time. Hence, we have

$$\zeta_i^{(k)} = \begin{cases} \dfrac{c}{n} & i = 7 \\ \dfrac{\tau}{n \times \tau - k} & \text{otherwise.} \end{cases}$$

[§] This contrasts with the approach of (Dingledine and Syverson, 2002), where the initiator would directly decrease her trust in all users in the path.

Assuming that the forwarding probability is $p_f = 0.7$, Figure 2 shows the probability that the target will be identified over time. Clearly, the target's privacy deteriorates quickly, as it becomes increasingly unlikely that users other than herself pick her. In particular, after seven sessions the protocol can no longer ensure adequate anonymity to user 7 as the probability $P(a_7 \mid o_7)$ that the attacker guess is correct becomes greater than 0.5.

4. Onion forwarding in Crowds

In the previous section we analysed the privacy protection afforded by Crowds extended with a notion of trust. Following a similar pattern, in this section we focus on the privacy guarantees offered by our protocol when equipped with '*onion forwarding*,' a superior forwarding technique used in systems actually deployed, such as Tor (Dingledine et al., 2004).

In Crowds, any user participating in a path has access to the cleartext messages routed through it. In particular, as all relay requests expose the message's final destination, a team of attackers will soon build up a host of observations suitable to classify the behaviour of honest participants. We recently proved in (Hamadou et al., 2009) that such extra attackers' knowledge makes it very difficult to achieve anonymity in Crowds. The most effective technique available against such a risk is onion forwarding, originally used in the 'Onion Routing' protocol (Syverson et al., 1997), and currently implemented widely in real-world systems. The idea is roughly as follows. The session *encryption keys* established when forming a path (See Section 3.1) by the initiator , one for each user in it, are used to repeatedly encrypt each message she routes through, starting with the last node on the path, and ending with the first. Each intermediate user, in the act of receiving the message decrypts it with her key. Doing so, she 'peels' away the outmost layer of encryption, discovers who the next forwarder is, and relays the message as required. In particular, only the last node sees the message in clear and learns its actual destination. Thus, a transaction is detected only if the last user in the path, also known as the '*exit node*,' is an attacker, and the last honest user in the path is then detected.

4.1. *Privacy level of the onion forwarding*

Next we study the privacy ensured to each member participating in the protocol under the onion forwarding scheme. As we did earlier, we begin with computing the conditional probability $P(o_j \mid a_i)$.

Proposition 4.

$$P\left(o_j \mid a_i\right) = \frac{(1-p_f)\,\zeta_i \epsilon_{ij}}{1-\zeta_i p_f} + \frac{q_{ij}\,\zeta_i p_f}{1-\zeta_i p_f}\,.$$

Proof. Let k denote the last position occupied by an honest user preceding an attacker on the path, i.e., the position of the detected user. We denote by $P(o_j \mid a_i)_{(k)}$ the probability that user j is detected exactly at position k. Again, only the initiator can be detected at position zero, and the probability that this happens is equal to the overall probability that the initiator chooses a/some corrupt members as forwarders, multiplied by the probability that the last corrupt member is the

last node in the path. Therefore

$$P\left(o_j \mid a_i\right)_{(0)} = \begin{cases} \sum_{m=1}^{\infty} \zeta_i^m p_f^{m-1}(1-p_f) & i = j \\ 0 & i \neq j \end{cases} \quad (7)$$

$$= \begin{cases} \frac{(1-p_f)\zeta_i}{1-\zeta_i p_f} & i = j \\ 0 & i \neq j \end{cases}$$

Now the probability that j is detected at position $k > 0$ is given by

— the probability that she decides to forward k times and picks $k-1$ users (does not matter whether honest or not, as non-exit attackers cannot see the messages), i.e., p_f^{k-1} (recall that at the initial step she does not flip the coin),
— times the probability of choosing j as the kth forwarder, i.e. q_{ij},
— times the probability that she picks any number k' of attackers at the end of the path, i.e. $\sum_{k'=1}^{\infty} p_f^{k'} \zeta_i^{k'} (1-p_f)$.

Therefore

$$\forall k \geq 1, \;\; P\left(o_j \mid a_i\right)_{(k)} = \sum_{k=1}^{\infty}\Big(p_f^{k-1} q_{ij} \sum_{k'=1}^{\infty} p_f^{k'} \zeta_i^{k'} (1-p_f)\Big),$$

and hence

$$\begin{aligned} P\left(o_j \mid a_i\right) &= \sum_{k=0}^{\infty} P\left(o_j \mid a_i\right)_{(k)} \\ &= \frac{(1-p_f)\zeta_i}{1-\zeta_i p_f}\epsilon_{ij} + \sum_{k=1}^{\infty}\Big(p_f^{k-1} q_{ij} \sum_{k'=1}^{\infty} p_f^{k'} \zeta_i^{k'} (1-p_f)\Big) \\ &= \frac{(1-p_f)\zeta_i}{1-\zeta_i p_f}\epsilon_{ij} + q_{ij}(1-p_f) \sum_{k=1}^{\infty}\Big(p_f^{k-1} \sum_{k'=1}^{\infty} p_f^{k'} \zeta_i^{k'}\Big) \\ &= \frac{(1-p_f)\zeta_i}{1-\zeta_i p_f}\epsilon_{ij} + q_{ij}(1-p_f) \sum_{k=1}^{\infty} p_f^{k-1} \frac{\zeta_i p_f}{1-\zeta_i p_f} \\ &= \frac{(1-p_f)\zeta_i}{1-\zeta_i p_f}\epsilon_{ij} + \frac{q_{ij}(1-p_f)\zeta_i p_f}{1-\zeta_i p_f} \frac{1}{1-p_f} \\ &= \frac{(1-p_f)\zeta_i}{1-\zeta_i p_f}\epsilon_{ij} + \frac{q_{ij}\zeta_i p_f}{1-\zeta_i p_f} \;. \end{aligned}$$

□

Corollary 4. $P(o_j \mid a_i) = 0$ if and only if one of the following holds:

1 $\zeta_i = 0$;
2 $q_{ij} = 0$ and $i \neq j$.

Now on the probability of detecting a user $P(o_j)$. Assuming uniform distribution of anonymous events we have the following result.

Proposition 5. If the honest member are equally likely to initiate a transaction then.

$$P(o_j) = \frac{1}{n-c}\left(\frac{(1-p_f)}{1-\zeta_j p_f}\,\zeta_j + \sum_{i\leq(n-c)} \frac{q_{ij}\,\zeta_i p_f}{1-\zeta_i p_f}\right).$$

Proof. Since the anonymous events are uniformly distributed then $P(a_i) = 1/(n-c)$ for all i. Thus

$$\begin{aligned}
P(o_j) &= \sum_{i\leq(n-c)} P\left(o_j \mid a_i\right) P(a_i) \\
&= \sum_{i\leq(n-c)} P\left(o_j \mid a_i\right) \frac{1}{n-c} \\
&= \frac{1}{n-c} \sum_{i\leq(n-c)} P\left(o_j \mid a_i\right) \\
&= \frac{1}{n-c} \sum_{i\leq(n-c)} \left(\frac{(1-p_f)}{1-\zeta_i p_f}\zeta_i \epsilon_{ij} + \frac{q_{ij}\zeta_i p_f}{1-\zeta_i p_f}\right) \\
&= \frac{1}{n-c}\left(\frac{(1-p_f)}{1-\zeta_j p_f}\zeta_j + \sum_{i\leq(n-c)} \frac{q_{ij}\zeta_i p_f}{1-\zeta_i p_f}\right).
\end{aligned}$$

□

We then have the same conditions of non-detectability as in the previous section; that is, the following result holds.

Corollary 5. $P(o_j) = 0$ if and only if

$$\zeta_j = 0 \quad \text{and} \quad \forall i.\,(\,q_{ij} = 0 \text{ or } \zeta_i = 0\,)\,.$$

Now from Proposition 4 and 5 and the Bayes theorem, we have the following result.

Proposition 6. If the honest members are equally likely to initiate a transaction, then

$$P\left(a_i \mid o_j\right) = \frac{\dfrac{\zeta_i}{1-\zeta_i p_f}\epsilon_{ij} + \dfrac{q_{ij}\,\zeta_i\,p_f}{(1-p_f)(1-\zeta_i\,p_f)}}{\dfrac{\zeta_j}{1-\zeta_j p_f} + \displaystyle\sum_{k\leq(n-c)} \frac{q_{kj}\,\zeta_k\,p_f}{(1-p_f)(1-\zeta_k\,p_f)}}\,.$$

Now from Propositions 3 and 6, we can prove effectively that the privacy level ensured by the onion version is better than those offered by the versions where messages are forwarded in cleartext. More formally, let $\left[P(a_i \mid o_j)\right]_{CR}$ and $\left[P(a_i \mid o_j)\right]_{OR}$ denote the probability that i is the initiator given that j is detected under cleartext routing and onion routing, respectively. Then the following holds, whose prove is simple and therefore omitted.

Theorem 1. $\left[\,P(a_i \mid o_i)\,\right]_{OR} \leq \left[\,P(a_i \mid o_i)\,\right]_{CR}$, for all i.

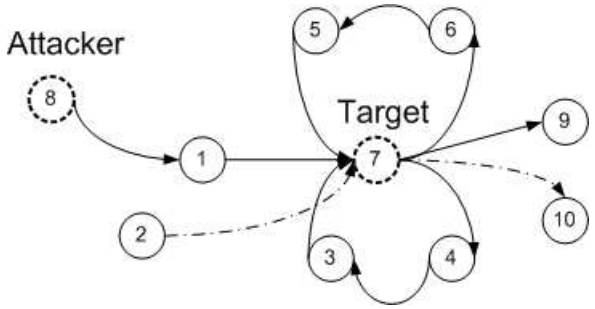


Fig. 3. Congestion attack

4.2. *On the security of the onion forwarding version*

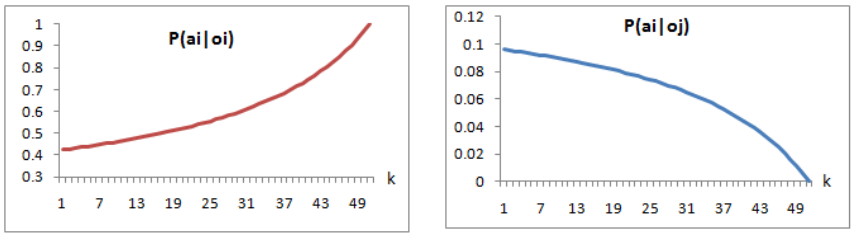
As mentioned before, onion forwarding is the forwarding technique of choice in several real-world systems. Recent work (Hopper et al., 2010; McLachlan and Hopper, 2008; McLachlan et al., 2009; Murdoch and Danezis, 2005; Evans et al., 2009) shows that such systems are vulnerable to so-called *congestion attacks*, which intuitively work as follows. Assume that the initiator selects a path which contains a corrupt user as the exit node. The attacker can then observe the pattern of arrival times of the initiator's requests, and tries to identify the entire path by selectively congesting the nodes she suspect to belong to it. Precisely, to determine whether or not a specific node occurs in the path, she asks a collaborating attacker to build a long path looping on the target node and ending with a corrupt node. Using this, the attacker perturbs the flow through the target node, so that if the latter belongs also to the path under observation, the perturbation will reflect at its exit node.

Here we use a variant of the congestion attack which, similarly to the previous section, allows internal attackers to deteriorate the reputation of a targeted honest user, and does not require the attacker to belong to a path. Figure 3 illustrates the attack, where a long path is built looping as many times as possible over the target, preferably using different loops involving different users. Thank to such properties, the target user will be significantly busy handling the same message again and again, whilst no other member of the path will be congested.

Figure 4 illustrates the effect of this attack using the same example as in the cleartext forwarding version in §3. The results are completely in tune with those presented by Figure 2: even though the target node initially enjoys a better anonymity protection, her anonymity will unequivocally fall, although more smoothly than in §3. In particular, after twenty sessions, the protocol no longer ensures adequate anonymity, as the probability of correctly gessing transactions of the target node becomes greater than 0.5.

5. Adaptive attackers

We have worked so far under the assumption that protocol participants either behave always honestly or always maliciously. Arguably, this is a rather unrealistic hypothesis in open and dynamic systems, where honest nodes can become malicious upon being successfully attacked. In this section we take the more realistic view that nodes may become corrupt, and study a new kind of attackers, which we dub 'adaptive,' and the relative attacks.



(a) $i = j = 7$ (b) $i \neq j = 7$

Fig. 4. Onion forwarding

Adaptive attackers differ from those we considered so far in the paper –and indeed from those considered so far in the literature on Crowds– in that when they intercept a message, rather than just reporting its sender as the initiator, they attempt to travel the path back in order to improve their chance to catch the actual originator. They do so by trying to *corrupt* the sender of the message, say j_1. If the attack succeeds, then the attacker effectively learns from j_1 all she needs to identify j_1's predecessor on the path, say j_2, and repeat the adaptive attack on j_2, having moved a step closer to the initiator. The process is repeated iteratively until the attacker either fails to corrupt the current node (or timeouts whilst trying to) or reaches the beginning of the path. When that happens, the attacker reports the current node, say j_k, which is obviously a better candidate than j_1 to have originated the transaction. Note that since a single node will typically appear several times in a path, an adaptive attacker in her attempt to travel the path backwards towards the initiator will in general meet each node several times. Thus, the attacker has no need to corrupt the node again, and no new knowledge may be acquired by doing so.

We regard this as a significant and realistic kind of attack, as there clearly are a multitude of ways in which the adaptive attacker may attempt to corrupt a node. These range from brute force attacks via virus and worms which gains the attacker complete control over the node, to milder approaches based on luring the target to give away some bit of information in exchange for some form of benefit, and in general are entirely independent of the Crowds protocol. We therefore do not postulate here about the means which may be available to the attacker to carry out her task, make no assumptions whatsoever about her power, and take the simplified view that each node has at all time the same probability π to become corrupted.

In the rest of the section we re-evaluate the privacy guarantees afforded by Crowds extended –with and without onion forwarding– under this new adaptive attack scenario.

5.1. Crowds *extended*

Our technical development proceeds *mutatis mutandis* as in §3 and §4. In particular, as before we first evaluate the conditional probability $P(o_j \mid a_i)$, then under the hypothesis that all honest users are equally likely to initiate a transaction, we compute $P(o_j)$, and finally, using Bayes Theorem, we obtain $P(a_i \mid o_j)$.

Before computing these probabilities, we note that once a clever attacker successfully corrupts a user, she will try to determine the first position of the victim in the path in order to jump to

the later position and just avoid the (eventually) useless task of corrupting intermediary users between positions of the same user.

Without onion routing, the attacker can send a message on the behalf of her victim to one of the victim's successors. If the message, which she can recognise because is sent in clear, is again received by the victim, then she knows that both positions of the victim are on the same path. By repeating the process with each successor, the attacker will determine the first position of the victim on the targeted path.

Now, let $H = \{1, 2 \ldots, n - c\}$ denote the set of honest users, naturally ordered, and $R \subseteq H$ be the subset of honest users that the attacker has successfully corrupted in her backward attack. Let $\overline{R} = H \setminus R$ be the complement of R and let $\eta_i(J) = \sum_{j \in J} q_{ij}$ be the overall probability that user i chooses a member of J, for J a subset of H. We denote $\mathbf{Perm}(R)$ the set of permutations of elements of R. Let σ be a permutation in $\mathbf{Perm}(R)$, and ρ ($1 \leq \rho \leq |R|$) a positive number. We denote $\overline{R}_\sigma \oplus \rho = \overline{R} \cup \{r_{\sigma(1)}, r_{\sigma(2)} \ldots, r_{\sigma(\rho)}\}$ the set of non-corrupted users augmented by the first ρ members of the σ permutation, in increasing order ($\sigma(i) \leq \sigma(j)$ iff $i \leq j$).

We observe that when the attackers successfully corrupted the initiator then there is only one possible detection/observable as the initiator will be detected with probability one. The analysis is therefore trivial. However, since our metric is conditioned by the observable events, we will exclude this limit case and only consider the case of detections when the initiator is not corrupted. Thus, we will say that *a user is detected if she is the first user preceding an attacker or a corrupted user and the attackers fail to corrupt her*.

Let $P(o_j, R \,|\, a_i)_{(k)}$ denote the probability that j is detected at position k and that the attacker successfully corrupted the honest users R in the path, given that i is the initiator.

For $k \geq 1$, the initiator cannot be corrupt as we assume the attacker knows the first position of her victim, and the initiator first position is $k = 0$. In this way, the attacker reports (i.e., detects) a user at a position greater than zero if and only if she fails to corrupt her. The result is as follows.

Proposition 7. For all (non empty) strict subset R of H, for all i and j in $\overline{R}$, and for all $k \geq 1$, the following holds.

$$P(o_j, R \,|\, a_i)_{(k)} = \eta_i^{k-1}(\overline{R}) p_f^k q_{ij} (1 - \pi) \sum_{\sigma \in \mathbf{Perm}(R)} \Big(\prod_{\rho=1}^{|R|} \phi_i(R, \sigma, \rho) \Big) \zeta_i$$

Where

$$\phi_i(R, \sigma, \rho) = q_{ir_\rho} p_f \pi \Big(1 + \frac{q_{ir_\rho} p_f}{1 - \eta_i(\overline{R}_\sigma \oplus \rho) p_f} \Big)$$

Proof. Since j is detected at the position k, then all users occupying positions from zero to k are not corrupt, i.e., they belong to the set $\overline{R}$. Then

$$\eta_i^{k-1}(\overline{R}) p_f^{k-1} q_{ij} (1 - \pi)$$

is the probability that the initiator i picks $k - 1$ users in $\overline{R}$ and then chooses the detected user j, times the probability that the attacker fails to corrupt j. Also, $\phi_i(R, \sigma, \rho)$ represents the probability that $r_{\sigma(\rho)}$, the currently under attack node, is a predecessor of a corrupted user, whose first position on the path and the position where she is detected are separated by a finite number of users who are so far not corrupted, i.e., those who belong to the set $\overline{R}_\sigma \oplus \rho$. The permutations are required

since the corrupted elements R could be chosen by the initiator in any order. Therefore

$$\begin{aligned}\phi_i(R,\sigma,\rho) &= q_{ir_{\sigma(\rho)}} p_f \pi\Big(1 + \sum_{k=0}^{\infty} \eta_i^k (\overline{R}_\sigma \oplus \rho) p_f^{k+1} q_{ir_{\sigma(\rho)}}\Big) \\ &= q_{ir_{\sigma(\rho)}} p_f \pi\Big(1 + \frac{q_{ir_{\sigma(\rho)}} p_f}{1 - \eta_i(\overline{R}_\sigma \oplus \rho) p_f}\Big)\end{aligned}$$

Finally, ζ_i is the probability that an attacker belongs to the path. □

We proceed now with the case when $k = 0$. If user j is detected at the first position, $k = 0$, then j is actually the initiator. Since we exclude the case the initiator is corrupted then her immediate successor is corrupted at any position $k \geq 1$ and the attacker is not successful in corrupting the initiator. Hence

$$P(o_i, R \,|\, a_i)_{(0)} = \sum_{\sigma \in \mathbf{Perm}(R)} \Big(\prod_{\rho=1}^{|R|} \phi_i(R,\sigma,\rho)\Big)(1-\pi)\zeta_i$$

Thus we have the following.

Proposition 8. For all (non empty) strict subset R of H and for all i not in R, the following holds.

$$P(o_j, R \,|\, a_i)_{(0)} = \begin{cases} \sum_{\sigma \in \mathbf{Perm}(R)} \prod_{\rho=1}^{|R|} \phi_i(R,\sigma,\rho)(1-\pi)\zeta_i & i = j \text{ and } i \notin R \\ 0 & i \neq j \end{cases}$$

Note that the computations above are based on the assumption that the set R is not empty. However, it might be the case that the first corruption attempt fails. In this case, the probability of detecting j when i initiates, is the same as without adaptive attackers, only weighted by the probability of the first corruption attempt fails.

Proposition 9.

$$P(o_j, \emptyset \,|\, a_i) = (1-\pi)\Big(\zeta_i \epsilon_{ij} + \frac{q_{ij}\zeta_i p_f}{1 - \eta_i p_f}\Big)$$

Now , from the results of Propositions 7, 8 and 9, we have:

Proposition 10.

$$P(o_j \,|\, a_i) = (1-\pi) \sum_{R \subseteq H \setminus \{i,j\}} \Big(\epsilon_{ij} + \frac{q_{ij} p_f}{1 - \eta_i p_f}\Big)\Phi_i(R,\sigma,\rho)\zeta_i$$

where

$$\Phi_i(R,\sigma,\rho) = \begin{cases} 1 & \text{if } R = \emptyset \\ \sum_{\sigma \in \mathbf{Perm}(R)} \prod_{\rho=1}^{|R|} \phi_i(R,\sigma,\rho) & \text{otherwise} \end{cases}$$

Now it can be easily shown that the result above extend the non adaptive attacker since when $\pi = 0$ we obtain the same result as in Proposition 1. In fact we have:

Corollary 6. If $\pi = 0$, then $P(o_j \,|\, a_i) = \zeta_i \epsilon_{ij} + \frac{q_{ij}\zeta_i p_f}{1 - \eta_i p_f}$,

The probabilities $P(o_j)$ and $P(a_i \mid o_j)$ can be derived from the above result, as in the previous sections, by computing $P(o_j) = \sum_{k \leq (n-c)} P(o_j \mid a_k)$ and $P(a_i \mid o_j) = \frac{P(o_j|a_i)P(a_i)}{P(o_j)}$, assuming a uniform a priori distribution, i.e $P(a_k) = \frac{1}{n-c}$ for all k. To keep the exposition simple, we do not give here the formal expressions of these probabilities, but simply show how the result compare to the case of non adaptive attackers. In particular the followings hold.

If the attacker is not adaptive then we obtain the same result as in Proposition 3.

Corollary 7. If $\pi = 0$, then

$$P\big(a_i \mid o_j\big) = \frac{\zeta_i \epsilon_{ij} + \dfrac{q_{ij}\zeta_i p_f}{1-\eta_i p_f}}{\zeta_j + \displaystyle\sum_{k \leq (n-c)} \frac{q_{kj}\zeta_k p_f}{1-\eta_k p_f}}$$

If the attacker is too strong, e.g. a government forcing by law people involved in a suspicious transaction to reveal their data, then the protocol cannot ensure any degree of anonymity. In fact we have:

Corollary 8. If $\pi = 1$, then

$$P\big(a_i \mid o_j\big) \quad = \quad \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases}$$

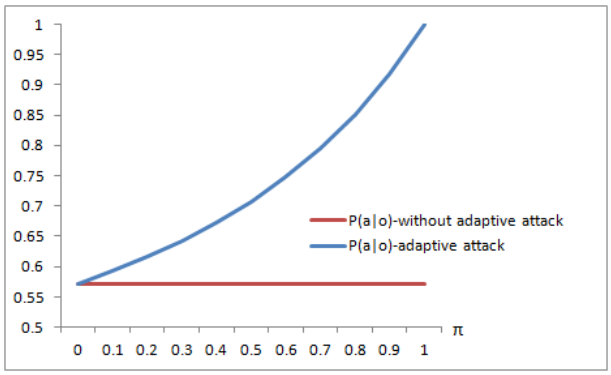


Fig. 5. Privacy level $P(a_i|o_i)$ under varying adaptive attack power π, for $n = 7$, $c = 2$, $p_f = 0.75$.

To conclude this section, we illustrate how the protocol behaves in the presence of adaptive attackers vs non adaptive attackers. Fig. 5 shows the privacy level $P(a_i \mid o_i)$ of extended Crowds as π varies. Here, we set $n = 7$, $c = 2$, $p_f = 0.75$ and let π range from 0 to 1. The probability $P(a_i \mid o_i)$ of adaptive attack increased from 0.5714 to 1. When the π equals 1, that is the attacker can corrupt every node in the path, then the protocol is no secure. The probability $P(a_i \mid o_i)$ for non-adaptive attacks is constant at 0.5714, which is always smaller than in the case of adaptive attack. Therefore, the protocol gets less secure as π increases.

5.2. *Onion forwarding*

For onion forwarding, the attack in the previous section to identify the first position of a corrupted user will clearly not work, as the attacker will not be able to recognise her own message, due to encryption. However the attacker can still vary the delay of forwarding legitimate packets so as to identify the victim's positions which exhibit the same pattern of delay: the so called *timing attack* (Hopper et al., 2010; Murdoch and Danezis, 2005; Wang et al., 2007). Repeating this attack as much as possible will allow her to ultimately determine the first position of the victim in the targeted path. Thus, we again assume that when an attacker successfully corrupts a user, she knows the first position occupied by the victim. Another difference with the privious case is that the attackers must appear as the last node on the path and there may be several attackers on the path. Under this adaptive attack scenario, the attackers will firstly determine the first position in the path containing an attacker, rather than start to corrupt nodes. They will then start the adaptive attack with the predecessor of the malicious user closest to the initiator.

Now under these assumptions, the analysis is quite the same as in the previous section. The only difference is that ζ_i, the overall probability of choosing an attacker, is replaced by T_i, the probability that either choosing one attacker and then the path selection ends; or firstly choosing an attacker node, then a finite number of users (honest users and attakers) and finally an attacker as exit node. Hence

$$\begin{aligned} T_i &= \Big[\sum_{m=0}^{\infty} p_f\zeta_i\Big(\sum_{j\leq n} q_{ij}^m p_f^m\Big) + 1\Big] p_f\zeta_i(1-p_f) \\ &= \Big(\sum_{m=0}^{\infty} p_f\zeta_i 1^m p_f^m + 1\Big) p_f\zeta_i(1-p_f) \\ &= p_f\zeta_i(1 + p_f\zeta_i - p_f). \end{aligned}$$

Using the notations introduced in the previous section, we have

Proposition 11.

$$P(o_j \,|\, a_i) = (1-\pi) \sum_{R\subseteq H\setminus\{i,j\}} \Big(\epsilon_{ij} + \frac{q_{ij}p_f}{1-\eta_i p_f}\Big)\Phi_i(R,\sigma,\rho)T_i$$

Unlike the results in the previous section, the case where $\pi = 0$ does not lead here to the same results as for onion forwarding in the presence of non adaptive attackers. The reason is that the adaptive attackers are stronger than their non-adaptive counterpart, even when $\pi = 0$. This is because they will always proceed to determine the position of the first attacker on the path before "betting on the detected user." In particular the following holds.

Corollary 9. If $\pi = 0$, then $P(o_j \,|\, a_i) = \Big(\epsilon_{ij} + \frac{q_{ij}p_f}{1-\eta_i p_f}\Big)T_i$.

Corollary 10. If $\pi = 0$, then

$$P\Big(a_i \mid o_j\Big) = \frac{\Big(\epsilon_{ij} + \dfrac{q_{ij}p_f}{1-\eta_i p_f}\Big)T_i}{T_j + \displaystyle\sum_{k\leq(n-c)} \frac{q_{kj}p_f}{1-\eta_k p_f}T_k}.$$

Observe that for a very strong attacker, we have zero level of anonymity.

Corollary 11. If $\pi = 1$, then

$$P\Big(a_i \mid o_j\Big) \quad = \quad \begin{cases} 1 & i = j \\ 0 & i \neq j\,. \end{cases}$$

To conclude this section, we observe that the gain, in term of security, obtained by onion forwarding in the case of non-adaptive adversaries (see Theorem 1), might be offset by the vulnerability of the protocol to the timing attack in the presence of adaptive attackers. In particular, when $\pi = 0$, that is the attackers cannot corrupt honest users, and each honest user has the same value $T_i = T$, which holds if and only if each of them has the same overall probability of picking an adversary, i.e. $\zeta_i = \zeta_j$ for all i and j, then we have the following result.

Corollary 12. If $\pi = 0$ and $\zeta_i = \zeta_j$, for all i and j, then

$$\Big[\, P(a_i \mid o_i)\,\Big]_{OR} \leq \Big[\, P(a_i \mid o_i)\,\Big]_{CR} = \Big[\, P(a_i \mid o_i)\,\Big]_{CR\_Adapt} = \Big[\, P(a_i \mid o_i)\,\Big]_{OR\_Adapt}$$

for all i.

Where CR_Adapt and OR_Adapt stand for the context of cleartext and onion routing in the presence of adaptive attackers respectively and CR and OR as in Theorem 1. However had we expressed the security of the protocol via the conditional probability $P\Big(o_i \mid a_i\Big)$ a la Reiter and Rubin probable innocence (Reiter and Rubin, 1998) then from Proposition 10 and 11, it is easy to observe that onion routing is more secure than its cleartext routing counterpart since $T_i \leq 1$.

Corollary 13. $P\Big[(o_i \mid a_i)\Big]_{OR\_Adapt} \leq P\Big[(o_i \mid a_i)\Big]_{CR\_Adapt}$, for all i.

6. Conclusion

In this paper we have presented an enhancement of the Crowds anonymity protocol via a notion of trust which allows crowd members to route their traffic according to their perceived degree of trustworthiness of each other member of the crowd. Such trust relations are not simply meant to reflect an immutable web of trust; rather, they represent mutable values meant to quantify the individual level of expectation that crowd members have in obtaining a satisfactory service from each other. In particular, they express a measure of an individual's belief that another user may become compromised by an attacker, either by a direct attempt to corrupt or by a denial-of-service attack.

We formalised our ideas of trust-driven routing quite simply by means of (variable) forwarding policies, with and without onion forwarding techniques. Our protocol variation has the potential

of improving the overall trustworthiness of data exchanges in anonymity networks, which cannot normally be taken for granted in a context where users are actively trying to conceal their identities.

Using such formalisation, in the paeper we then analysed quantitatively the privacy properties of the protocol, both for Crowds and onion forwarding, under standard and adaptive attacks.

References

Abe, M. (1998). Universally verifiable Mix-net with verification work indendent of the number of Mix-servers. In *Advances in Cryptology, EUROCRYPT*, volume 1403 of *LNCS*, pages 437–447.

Back, A., Möller, U., and Stiglic, A. (2001). Traffic analysis attacks and trade-offs in anonymity providing systems. In Moskowitz, I. S., editor, *Information Hiding*, volume 2137 of *Lecture Notes in Computer Science*, pages 245–257. Springer.

Backes, M., Lorenz, S., Maffei, M., and Pecina, K. (2010). Anonymous webs of trust. In *10th Privacy Enhancing Technologies Symposium, PETS 2010*, LNCS series. To appear.

Borisov, N., Danezis, G., Mittal, P., and Tabriz, P. (2007). Denial of service or denial of security? In Ning, P., di Vimercati, S. D. C., and Syverson, P. F., editors, *ACM Conference on Computer and Communications Security*, pages 92–102. ACM.

Camenisch, J. and Lysyanskaya, A. (2005). A formal treatment of onion routing. In Shoup, V., editor, *CRYPTO*, volume 3621 of *Lecture Notes in Computer Science*, pages 169–187. Springer.

Chatzikokolakis, K. and Palamidessi, C. (2006). Probable innocence revisited. *Theor. Comput. Sci.*, 367(1-2):123–138.

Chaum, D. (1981). Untraceable electronic mail, return addresses, and digital pseudonyms. *Commun. ACM*, 24(2):84–88.

Damiani, E., di Vimercati, S. D. C., Paraboschi, S., Pesenti, M., Samarati, P., and Zara, S. (2003). Fuzzy logic techniques for reputation management in anonymous peer-to-peer systems. In Wagenknecht, M. and Hampel, R., editors, *Proceedings of the 3rd Conference of the European Society for Fuzzy Logic and Technology*, pages 43–48.

Damiani, E., di Vimercati, S. D. C., Paraboschi, S., Samarati, P., and Violante, F. (2002). A reputation-based approach for choosing reliable resources in peer-to-peer networks. In Atluri, V., editor, *ACM Conference on Computer and Communications Security*, pages 207–216. ACM.

Dingledine, R., Freedman, M. J., Hopwood, D., and Molnar, D. (2001). A reputation system to increase mix-net reliability. In Moskowitz, I. S., editor, *Information Hiding*, volume 2137 of *Lecture Notes in Computer Science*, pages 126–141. Springer.

Dingledine, R., Mathewson, N., and Syverson, P. F. (2004). Tor: The second-generation onion router. In *USENIX Security Symposium*, pages 303–320. USENIX.

Dingledine, R. and Syverson, P. F. (2002). Reliable MIX cascade networks through reputation. In Blaze, M., editor, *Financial Cryptography*, volume 2357 of *Lecture Notes in Computer Science*, pages 253–268. Springer.

ElSalamouny, E., Krukow, K. T., and Sassone, V. (2009a). An analysis of the exponential decay principle in probabilistic trust models. *Theor. Comput. Sci.*, 410(41):4067–4084.

ElSalamouny, E., Sassone, V., and Nielsen, M. (2009b). HMM-based trust model. In Degano, P. and Guttman, J. D., editors, *Formal Aspects in Security and Trust*, volume 5983 of *Lecture Notes in Computer Science*, pages 21–35. Springer.

Evans, N. S., Dingledine, R., and Grothoff, C. (2009). A practical congestion attack on Tor using long paths. In *Proceedings of the 18th USENIX Security Symposium*.

Feigenbaum, J., Johnson, A., and Syverson, P. (2007). Probabilistic analysis of onion routing in a black-box model. In *Proceedings of the 2007 ACM workshop on Privacy in electronic society*, WPES '07, pages 1–10, New York, NY, USA. ACM.

Freedman, M. J. and Morris, R. (2002). Tarzan: a peer-to-peer anonymizing network layer. In Atluri, V., editor, *ACM Conference on Computer and Communications Security*, pages 193–206. ACM.

Halpern, J. Y. and O'Neill, K. R. (2005). Anonymity and information hiding in multiagent systems. *Journal of Computer Security*, 13(3):483–512.

Hamadou, S., Palamidessi, C., Sassone, V., and ElSalamouny, E. (2009). Probable innocence in the presence of independent knowledge. In Degano, P. and Guttman, J. D., editors, *Formal Aspects in Security and Trust, FAST 2009*, volume 5983 of *Lecture Notes in Computer Science*, pages 141–156. Springer.

Hamadou, S., Sassone, V., and Palamidessi, C. (2010). Reconciling belief and vulnerability in information flow. In *IEEE Symposium on Security and Privacy*, pages 79–92. IEEE Computer Society.

Hopper, N., Vasserman, E. Y., and Chan-Tin, E. (2010). How much anonymity does network latency leak? *ACM Trans. Inf. Syst. Secur.*, 13(2).

Jakobsson, M. (1999). Flash mixing. In *Annual ACM Symposium on Principles of Distributed Computing, PODC 99*, pages 83–89.

Johnson, A. and Syverson, P. F. (2009). More anonymous onion routing through trust. In *CSF*, pages 3–12. IEEE Computer Society.

Johnson, A., Syverson, P. F., Dingledine, R., and Mathewson, N. (2011). Trust-based anonymous communication: adversary models and routing algorithms. In Chen, Y., Danezis, G., and Shmatikov, V., editors, *ACM Conference on Computer and Communications Security*, pages 175–186. ACM.

Krukow, K., Nielsen, M., and Sassone, V. (2008). A logical framework for history-based access control and reputation systems. *Journal of Computer Security*, 16(1):63–101.

McLachlan, J. and Hopper, N. (2008). Don't clog the queue! circuit clogging and mitigation in P2P anonymity schemes. In Tsudik, G., editor, *Financial Cryptography*, volume 5143 of *Lecture Notes in Computer Science*, pages 31–46. Springer.

McLachlan, J., Tran, A., Hopper, N., and Kim, Y. (2009). Scalable onion routing with Torsk. In Al-Shaer, E., Jha, S., and Keromytis, A. D., editors, *ACM Conference on Computer and Communications Security*, pages 590–599. ACM.

Murdoch, S. J. and Danezis, G. (2005). Low-cost traffic analysis of tor. In *IEEE Symposium on Security and Privacy*, pages 183–195. IEEE Computer Society.

Nambiar, A. and Wright, M. (2006). Salsa: a structured approach to large-scale anonymity. In Juels, A., Wright, R. N., and di Vimercati, S. D. C., editors, *ACM Conference on Computer and Communications Security*, pages 17–26. ACM.

Neff, C. A. (2001). A verifiable secret shuffle and its application to e-voting. In *ACM Conference on Computer and Communications Security*, pages 116–125.

Ohkubo, M. and Abe, M. (2000). A length-invariant hybrid mix. In Okamoto, T., editor, *ASIACRYPT*, volume 1976 of *Lecture Notes in Computer Science*, pages 178–191. Springer.

Pappas, V., Athanasopoulos, E., Ioannidis, S., and Markatos, E. P. (2008). Compromising anonymity using packet spinning. In Wu, T.-C., Lei, C.-L., Rijmen, V., and Lee, D.-T., editors, *ISC*, volume 5222 of *Lecture Notes in Computer Science*, pages 161–174. Springer.

Reiter, M. K. and Rubin, A. D. (1998). Crowds: Anonymity for web transactions. *ACM Trans. Inf. Syst. Secur.*, 1(1):66–92.

Rennhard, M. and Plattner, B. (2002). Introducing MorphMix: peer-to-peer based anonymous internet usage with collusion detection. In Jajodia, S. and Samarati, P., editors, *Proceedings of the 2002 ACM workshop on Privacy in the Electronic Society, WPES*, pages 91–102. ACM.

Sassone, V., ElSalamouny, E., and Hamadou, S. (2010a). Trust in Crowds: probabilistic behaviour in

anonymity protocols. In *Symposium on Trustworthy Global Computing, TGC 2010*, volume 6084 of *LNCS*. Springer.

Sassone, V., Hamadou, S., and Yang, M. (2010b). Trust in anonymity networks. In Gastin, P. and Laroussinie, F., editors, *CONCUR*, volume 6269 of *Lecture Notes in Computer Science*, pages 48–70. Springer.

Singh, A. and Liu, L. (2003). Trustme: Anonymous management of trust relationships in decentralized P2P systems. In Shahmehri, N., Graham, R. L., and Caronni, G., editors, *Peer-to-Peer Computing*, pages 142–149. IEEE Computer Society.

Smith, G. (2009). On the foundations of quantitative information flow. In de Alfaro, L., editor, *FOSSACS*, volume 5504 of *Lecture Notes in Computer Science*, pages 288–302. Springer.

Syverson, P., Tsudik, G., Reed, M., and Landwehr, C. (2001). Towards an analysis of onion routing security. In *INTERNATIONAL WORKSHOP ON DESIGNING PRIVACY ENHANCING TECHNOLOGIES: DESIGN ISSUES IN ANONYMITY AND UNOBSERVABILITY*, pages 96–114. Springer-Verlag New York, Inc.

Syverson, P. F., Goldschlag, D. M., and Reed, M. G. (1997). Anonymous connections and onion routing. In *IEEE Symposium on Security and Privacy*, pages 44–54. IEEE Computer Society.

Wang, X., Chen, S., and Jajodia, S. (2007). Network flow watermarking attack on low-latency anonymous communication systems. In *IEEE Symposium on Security and Privacy*, pages 116–130. IEEE Computer Society.

Wang, Y. and Vassileva, J. (2003). Trust and reputation model in peer-to-peer networks. In Shahmehri, N., Graham, R. L., and Caronni, G., editors, *Peer-to-Peer Computing*. IEEE Computer Society.

Wiangsripanawan, R., Susilo, W., and Safavi-Naini, R. (2007). Design principles for low latency anonymous network systems secure against timing attacks. In Brankovic, L., Coddington, P. D., Roddick, J. F., Steketee, C., Warren, J. R., and Wendelborn, A. L., editors, *Proc. Fifth Australasian Information Security Workshop (Privacy Enhancing Technologies), AISW 2007*, volume 68 of *CRPIT*, pages 183–191. Australian Computer Society.