Adapting P2P Mixnets to Provide Anonymity for Uplink-Intensive Applications

Francesco Buccafurri[a], Vincenzo De Angelis[b] and Sara Lazzaro[c]
Department of Information Engineering, Infrastructure and Sustainable Energy (DIIES), Università Mediterranea di Reggio Calabria, Via dell'Università 25, 89122 Reggio Calabria, Italy

Keywords: Mixnet, Anonymous Communication.

Abstract: Anonymity in Web and Internet communication is a widely investigated problem. Mixnets represent certainly the most concrete and effective approach to achieving the above goal. In general, the drawback of these approaches is that anonymity has a price in terms of traffic overhead and latency, when the global adversary model is adopted. On the Internet, to achieve scalability and not to require relevant infrastructure and network-protocol changes, only P2P overlay protocols can be used. In recent years, we are seeing a change in Internet traffic. Due to IoT, cloud storage, WSN, M2M, etc., uplink traffic is increasingly growing. An interesting problem to address is whether this new traffic configuration may enable new strategies for improving the effectiveness of P2P mixnet-like approaches. In this paper, we investigate this problem, by considering the most representative Internet-scale P2P mixnet, called Tarzan, which is designed to obtain strong anonymity while preserving low-latency applications. We experimentally demonstrate that changing the cover traffic from bidirectional to unidirectional by making tunnels cyclic is advantageous in the case of uplink-intensive applications. The outcomes of the paper can thus give a contribution to improve mixnet-based approaches in the future Internet.

1 INTRODUCTION

Anonymity in Web and Internet communication is a widely investigated problem (Shirazi et al., 2018). The most known and used anonymous protocol is Tor (Dingledine et al., 2004). However, as well-known, anonymity is easily broken under even weak threat models (Karunanayake et al., 2020). As a matter of fact, very severe threat models are realistic in which global (i.e., with a global view of the network traffic) passive adversaries or malicious participants are allowed. The most effective approaches existing in the literature that achieve the above goal are based on the concept of mixnet (Chaum, 1981) including cover traffic. Mixnet protocols rely on intermediate servers (called *mix-nodes*) that mix the messages coming from different sources to hide the relationship between the incoming messages to and the outcoming messages from the mix-nodes.

On the Internet, to achieve scalability and not to enforce relevant infrastructure and network-protocol changes, only P2P overlay routing protocols can be adopted. Moreover, with P2P approaches, we avoid the risk of the adversary gaining control of the servers, and, thus, more robust solutions can be implemented.

In recent years, we are seeing a drastic change in Internet traffic. Due to IoT, cloud storage, Wireless Sensor Networks (WSN), Machine-to-Machine networks (M2M), etc., uplink traffic is increasingly growing (Oueis and Strinati, 2016; Yang and Larsson, 2019; Shafiq et al., 2013; Berger et al., 2015).

Among other use cases, a relevant application context is file storage in the cloud. The trend is to have the entire local disk uploaded to the cloud (or not to have a local disk at all). The amount of information uploaded to the cloud is much greater than the information downloaded by the user. An interesting question to pose is whether this new traffic configuration may enable new strategies for improving the effectiveness of P2P mixnet-like approaches.

In this paper, we study this problem, by focusing our attention on the *anonymity trilemma* (Das et al., 2018), which states the existence of a trade-off between three metrics: anonymity, latency, and cover traffic. For P2P-based mixnets, we consider the

[a] https://orcid.org/0000-0003-0448-8464
[b] https://orcid.org/0000-0001-9731-3641
[c] https://orcid.org/0000-0002-0846-4980

Buccafurri, F., De Angelis, V. and Lazzaro, S.
Adapting P2P Mixnets to Provide Anonymity for Uplink-Intensive Applications.
DOI: 10.5220/0012077100003555
In *Proceedings of the 20th International Conference on Security and Cryptography (SECRYPT 2023)*, pages 73-84
ISBN: 978-989-758-666-8; ISSN: 2184-7711

most representative one, called *Tarzan* (Freedman and Morris, 2002), which is designed to be used at Internet scale, for web and other low-latency applications.

The result achieved in our paper is that changing the cover traffic from bidirectional to unidirectional, by making tunnels cyclic, is quite advantageous in the case of uplink-intensive applications. The above claim has been demonstrated experimentally by choosing Tarzan as a reference mixnet.

We argue that the choice of a specific mixnet for our study is not critical for the contribution given in this paper, due to the fact that all mixnets have the same structural functioning. Actually, our contribution should be seen not as an improvement of an existing protocol (i.e., Tarzan) but as a proposal of a new paradigm of mixnet suitable for an emerging application context. On the other hand, despite its age, Tarzan is the only effective proposed P2P anonymous routing protocol guaranteeing low latency even in large-scale Internet scenarios. Indeed, the protocol allows a client to anonymously contact a server through a tunnel whose length is independent of the number of nodes participating in the P2P network. Tarzan implements a P2P overlay network at the IP layer, in which peers collaborate with each other to implement anonymous tunnels through which a client may reach a proxy node (called PNAT) from which the server is reached. Another advantage of Tarzan with respect to recent state-of-the-art approaches is that, unlike the emerging mixnets that adopt centralized and explicit shuffling nodes (Piotrowska et al., 2017), the P2P approach makes the solution more robust against possible attacks on the nodes of the route (or their collusion). Indeed, all the nodes of the network are potentially sender or relay nodes and then there are no few explicit targets for the attacker. The only relevant approach that implements a P2P overlay network is (Shen et al., 2021). However, it does not work at the IP layer and, moreover, the length of each tunnel is $log\ n$, where n is the number of nodes of the network. Therefore, unlike Tarzan, the latency is growing with the number of nodes. Hence, the protocol is not suitable for low-latency applications when the number of users scales at huge values, as may happen in Internet scenarios. The study conducted in this paper leads to the definition of a new P2P overlay anonymous protocol, called *C(yclic)-Tarzan*, which outperforms Tarzan in the case of uplink-intensive applications. Specifically, regarding the anonymity trilemma, we show that for uplink-intensive applications, by fixing the same latency and the same cover traffic volume, C-Tarzan offers a greater cardinality of the anonymity set than Tarzan.

The paper is organized as follows. In Section 2, we investigate the related work. In Section 3, we provide the background notions about the Tarzan protocol. In Section 4, we formulate the problem addressed in this paper and give the intuition of our approach. The detailed protocol is presented in Section 5. We perform an analytical study of the latency in Tarzan and C-Tarzan in Section 6 and provide an experimental validation of our approach in Section 7. The security of the proposed approach is examined in Section 8. Finally, in Section 9, we draw our conclusions.

2 RELATED WORK

Anonymous Communication Networks (ACN) (Xia et al., 2020; Shirazi et al., 2018) are networks in which users are provided with anonymity services protecting their privacy also against possible censorship. An ambitious goal to achieve is to offer anonymity guarantees against passive eavesdroppers (including a global adversary) and malicious participants. As stated in (Danezis and Diaz, 2008), to achieve this goal, dummy traffic needs to be injected into the network to hide the actual traffic.

In the literature, two main approaches leveraging dummy traffic are available. The first is based on *buses* (Hirt et al., 2008; Beimel and Dolev, 2003; Young and Yung, 2014). In this solution, a predetermined route is used by the sender to anonymously communicate with the destination. However, this technique is not scalable on a large network, since it requires an Eulerian path passing through all the nodes, which leads to a prohibitive cost in terms of latency. The second approach is represented by the *mixnets* (Chaum, 1981) which, in general, offers a lower latency with a price in terms of cover traffic. Some recent mixnet proposals exist (Kotzanikolaou et al., 2017; Van Den Hooff et al., 2015; Piotrowska et al., 2017; Ben Guirat et al., 2021). Anyway, some drawbacks should be taken into account. For example, as recently stated in (Alexopoulos et al., 2017), the work proposed in (Kotzanikolaou et al., 2017) suffers from very large communication overhead. Regarding (Van Den Hooff et al., 2015), as stated by the authors themselves, the high end-to-end latency makes the protocol not suitable for low-latency applications such as web browsing. Moreover, these approaches rely on a server-oriented architecture, which is known to be less robust against possible attacks on the nodes of the route (Shen et al., 2021) and less scalable than P2P architecture (Shirazi et al., 2018).

Therefore, the state of the art of P2P approaches for low-latency applications is represented by Tarzan (Freedman and Morris, 2002), which is a work with

high impact in the (even current) scientific literature. Actually, another P2P mixnet proposal, less recent but adopted in practice, is I2P (Zantout et al., 2011). However, it suffers from different vulnerabilities such as brute-force attacks or timing attacks. Then, the authors suggest adopting some mitigations, such as constant-rate cover traffic.

Our paper strongly refers to (Freedman and Morris, 2002), which is chosen as a reference P2P-mixnet to prove the claim that, for uplink-intensive applications, a new paradigm of mixnet that includes only unidirectional cover traffic is advantageous. Observe that uplink-intensive applications are becoming more and more common in recent years (Oueis and Strinati, 2016; Yang and Larsson, 2019). Some examples of uplink-dominant applications are represented by M2M (Nikaein et al., 2014; Centenaro and Vangelista, 2015), Industrial IoT (Kwon et al., 2016), and Wireless-Sensor-Network (Dester et al., 2018). Furthermore, intrinsically, cloud-based applications increase the uplink bandwidth demand with respect to traditional client-server applications (Sun et al., 2020). Finally, the evolution of social networks toward the so-called metaverse will result in significant growth in uplink-traffic demand (Cheng et al., 2022).

3 BACKGROUND: THE TARZAN PROTOCOL

In this section, we provide the main background notions about the Tarzan protocol (Freedman and Morris, 2002). Tarzan is a P2P anonymous IP network overlay. Each node, in order to communicate anonymously with a destination, builds a tunnel composed of a sequence of nodes in which the last node communicates with a special node, called *PNAT*, which acts as a proxy towards the destination. Being Tarzan a mixnet, as shown in Section 4, the cardinality of the anonymity set increases exponentially with the length of the tunnel. Each intermediate node of the tunnel acts as a relay by forwarding the messages coming from the previous node. Anyway, since it does not know its position in the tunnel, it is not able to identify the originator of the traffic. Each node is associated with a group of nodes called *mimics*, which are used for the construction of the tunnel. Mimics are selected by using a gossip protocol and a *lookup* function based on a distributed hash table (DHT). Specifically, each node maintains a three-level hierarchy DHT in which the peer nodes are inserted in a given position according to their IP addresses. This table offers a *lookup* function that, given a string as input, returns as output an IP address of a node of the network. Observe that the input can be any arbitrary string. To select k mimics, each node a invokes the function $lookup^i(a.ipaddr)$ for $1 < i \leq k+1$ where $a.ipaddr$ represents the IP address of a. If each node selects k mimics, we expect, on average, that each node has $2k$ mimics. The DHT offers two advantages. First, since the DHT is shared by all the nodes, mimic selection is publicly verifiable and then this prevents an adversary node from selecting more than k mimics. The second advantage is that the mimics for a node are randomly selected in different IP domains, so that if an adversary controls an entire domain, by generating a huge number of malicious nodes in that domain, it does not increase the probability that a malicious node of such domain is selected as a mimic.

To send messages through this tunnel, the initiator exchanges a symmetric key with each node of the tunnel. Moreover, a *hop-by-hop* symmetric key between any pair of adjacent mimics is exchanged. This procedure is similar to the construction of a virtual circuit in the Tor protocol (Dingledine et al., 2004). Once exchanged these keys, the messages can be sent through the tunnel encrypted in a layered fashion. Furthermore, the hop-by-hop communications are encrypted with hop-by-hop symmetric keys. The same tunnel is used also for the response. A node establishes with each of its mimics a bidirectional cover traffic flow into which real data can be indistinguishably inserted.

4 PROBLEM FORMULATION AND BASIC APPROACH

As stated in the introduction, this paper aims to study how to adapt P2P mixnets to uplink-intensive applications. To do this, we refer to Tarzan.

In mixnets (and in Tarzan too), there are three main metrics to consider (Das et al., 2018): latency, amount of cover traffic, and cardinality of the anonymity set. Often, the latency is a project constraint as well as the anonymity degree. Therefore, adapting Tarzan to our setting means to find a solution that, under the same cover traffic level (that cannot be increased for the above reasons) and a fixed latency, offers a better anonymity degree than Tarzan.

As a measure for the cover traffic, the degree of the nodes can be considered. Indeed, the more links occur in the network the more cover traffic has to be generated. Moreover, Tarzan requires bidirectional cover traffic in each link to use the same path as the forward and response route. This also allows Tarzan not to have to exclude possible candidate nodes from the anonymity set due to traffic direction incompatibilities. Therefore, a challenge could be to elimi-

nate the bidirectionality of cover traffic still preserving the Tarzan-like approach. This is the purpose of our proposal. The idea is that unidirectional traffic could still be enabled in Tarzan protocol by rearranging the mimics of a node in such a way that they form a cycle. Once mimics are so organized, we can build a tunnel as in Tarzan, but requiring that two adjacent nodes in the tunnel belong to a cycle. This way, the response can be routed by moving back, at each hop between two nodes, by traveling the entire cycle involving these nodes. Thus, no bidirectional traffic is needed. This idea is sketched in Figure 1, in which the red lines represent the forward path and the green lines represent the cycles traveled by the response.

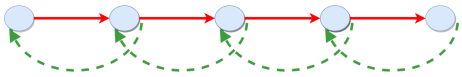
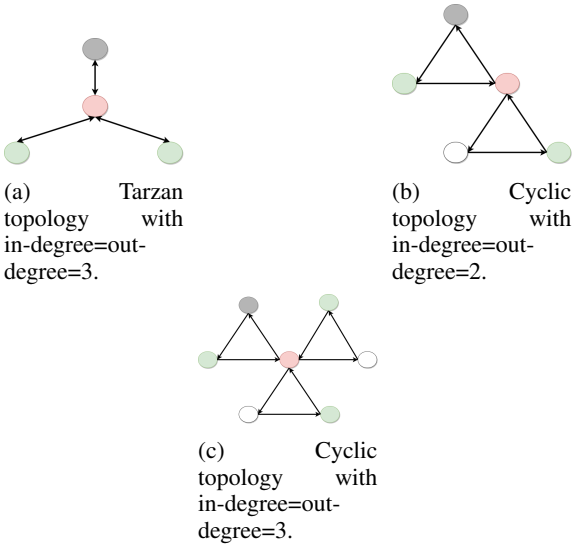


Figure 1: Forward path (red arrow) and return path (green arrow).

However, there might be a price in terms of latency to pay when applying this cyclic approach, since, in general, the response would go through a longer path than the forward path. Instead, in Tarzan, forward and return paths are the same. Therefore, the application of this idea deserves to be studied. This is just the aim of this paper. The first immediate consideration is that it is convenient to minimize the size of cycles. Being Tarzan bidirectional links equivalent to 2-node cycles, the minimum dimension for non-trivial cycles is the case of 3-node cycles. On the other hand, it is intuitive to understand that no advantage can derive from having bigger cycles. A much less clear point is to understand whether we have to pay a price also in terms of anonymity set. This question derives from the following qualitative analysis.

We start by considering the uncertainty at two hops in the standard Tarzan topology and a two-hop equivalent topology in which cycles are enabled. This is represented in Figure 2. Specifically, in Figure 2a, we represent the standard Tarzan topology in which each node has three mimics. Suppose that the gray node receives a message from the red node. In this case, the candidate senders, at a maximum distance of two hops, are the red node and the two green nodes. The same uncertainty is obtained in the cyclic topology represented in Figure 2b in which, again, the candidate senders, at a maximum distance of two hops, are the red node and the two green nodes.

Regarding the cover traffic, we observe that in Figure 2a, we have three bidirectional links while in Figure 2b we have four unidirectional links, thus saving two unidirectional links. Therefore, it appears that keeping the same uncertainty, we have a significant



(a) Tarzan topology with in-degree=out-degree=3.

(b) Cyclic topology with in-degree=out-degree=2.

(c) Cyclic topology with in-degree=out-degree=3.

Figure 2: Uncertainty at two hops.

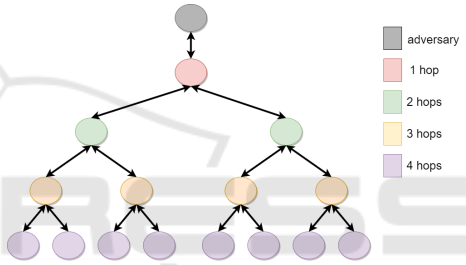


Figure 3: Extension of Figure 2a.

reduction in cover traffic.

Unfortunately, we can realize that the growth of the cardinality of the anonymity set for the cyclic approach is slightly slower than that of standard Tarzan. We can understand this just by considering the case of tunnel length equal to four. To see this, we extend the topologies of Figures 2a and 2b, in Figures 3 and 4 respectively, to include tunnels with a maximum length of four hops. In this case, the anonymity set of Figure 3 contains 15 nodes, while the anonymity set of Figure 4 contains 11 nodes. Observe that, even though the cardinality of the anonymity set of the cyclic approach is smaller than that of Tarzan, the growth of both is exponential in the length of the tunnel. Moreover, we have to take into account also the price in terms of latency required in the cyclic approach. However, the advantage in terms of cover traffic is maintained with respect to Tarzan. Therefore, it is interesting to understand what happens if we compare the standard Tarzan with the cyclic version by considering two topologies that determine the same cover traffic. The effect at two hops is highlighted in Figure 2c in which there are 6 unidirectional links equiva-

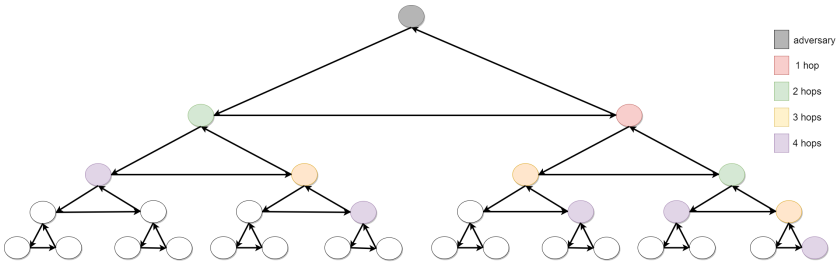


Figure 4: Extension of Figure 2b.

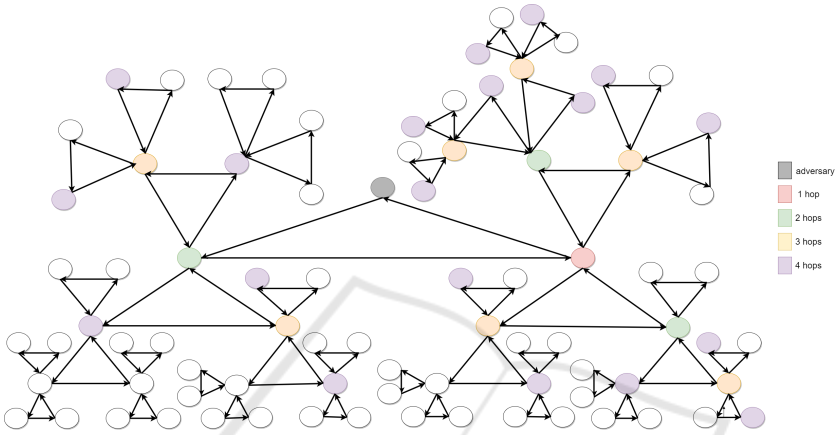


Figure 5: Extension of Figure 2c.

lent to three bidirectional links of Tarzan. Therein, we can see that the candidate senders are the red node and the three green nodes. Therefore, the uncertainty at two hops is increased. The extension to four hops of Figure 2c is represented in Figure 5 in which the anonymity set contains 30 nodes. Therefore, under the same cover traffic, the cyclic approach offers a greater cardinality of the anonymity set. However, the price in terms of latency still remains. Clearly, in Tarzan, the latency depends only on the tunnel length. In the cyclic approach, it mostly depends on the tunnel length, and in a small measure also depends on the node degree. Moreover, the disadvantage of the cyclic version depends also on the balance between downlink and uplink traffic (the more the weight of the downlink, the more the disadvantage). In fact, the price we pay in terms of latency is related to the downlink traffic for the return path, which is in general longer than the forward path. Thus, the problem we want to study is the following: In the cyclic approach, can we reduce the tunnel length to reduce latency and still be able to have a cardinality of the anonymity set greater than Tarzan? Though in general the answer to this question might be negative, it is interesting to understand what happens when there is an unbalance between the amount of uplink and downlink traffic. As we will describe in the sequel of the paper, the result we achieve is that for uplink-intensive networks, the above approach is definitely advantageous.

5 C-TARZAN

In this section, we propose a new protocol, called *Circular Tarzan (C-Tarzan)*, based on the cyclic approach introduced in the previous section. The idea is to move from bidirectional links (adopted in Tarzan) to unidirectional links. This is possible if the response is routed through the cycles to which the mimics belong. As discussed above, we consider cycles of three nodes to minimize the price in terms of latency.

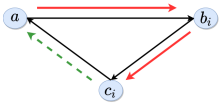
To build the cycles among mimics nodes, we design a new mimic selection algorithm that differs from that of Tarzan. We assume that the same Tarzan DHT table (with the *lookup* function) is used in C-Tarzan for the mimic selection. Each node a chooses k' mimics through the *lookup* function (see Section 3) as in Tarzan. Specifically, a selects $b_i = lookup^i(a.ipaddr)$ for $1 < i \leq k' + 1$. Each chosen mimic b_i can verify the correctness of the selection. Anyway, differently from Tarzan, a unidirectional link directed from a to b_i is established. At this point, each b_i will choose a mimic $c_i = lookup^i(a.ipaddr||b_i.ipaddr)$ and a unidirectional link directed from b_i to c_i is established.

Observe that since the function *lookup* accepts any arbitrary string as input and returns an IP address of a node of the network, it is guaranteed that the node c_i always exists in the network. c_i can verify the correctness of the mimic selection started by a, involving the node b_i. Finally, to close the cycle, a unidirectional link is established from c_i to a.

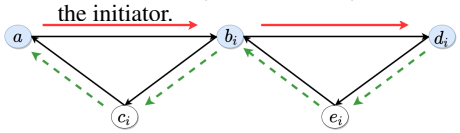
It is easy to see that each node has on average $6k'$ mimics. Indeed, each node A *selects directly* k' mimics $B_1, \dots B_{k'}$ to build k' cycles. In each cycle involving the node B_i, there will be a node C_i that establishes a link with A to close the cycle. Then, A will have further k' mimics $C_1, \dots C_{k'}$, for a total of $2k'$ mimics. At this point, on average, A *is selected directly* by k' nodes to build further k' cycles. This leads to further $2k'$ mimics for A. Finally, on average, A *is selected indirectly* by k' nodes that, in turn, are selected directly by other k' nodes to build cycles. As before, further $2k'$ mimics for A are obtained. Therefore, since unidirectional links are established between pairs of mimics, each node has, on average, $6k'$ unidirectional links ($3k'$ outgoing and $3k'$ ingoing).

We recall that, in Tarzan, if a node selects k mimics, it has, on average, $2k$ mimics and then $2k$ bidirectional links corresponding to $4k$ unidirectional links. Therefore, by considering the number of links as a measure of cover traffic, we have that, to obtain the same level of cover traffic in Tarzan and C-Tarzan, we have to set k' such that $6 \cdot k' = 4 \cdot k$ i.e., $k' = \frac{2}{3} \cdot k$.

At this point, we discuss how the messages are forwarded anonymously towards the destination and the latter can reply to the initiator. As in Tarzan, we assume that a symmetric hop-by-hop key is exchanged preliminarily between mimics. To enable the communication, we need to redefine the entire building process of the tunnel. Specifically, the initiator a selects, as first relay, one of its *outgoing* mimics b_i, i.e., a mimic b_i such that a directed link from a to b_i exists. Similarly to the standard Tarzan protocol, a needs the set of the (outgoing) mimics of b_i and to exchange a symmetric key with b_i. Anyway, since the link between a and b_i is unidirectional, a reply cannot be sent directly from b_i to a, because it would be not covered by dummy traffic. Therefore, to enable the reply, we define the function $C.next$ that can be invoked by a node C. This function receives as input a node B and returns as output the node A, such that there exist: (i) a direct link from B to C, (ii) a direct link from C to A, (iii) a direct link from A to B. Observe that, the *next* function leverages the fact that each node locally stores all the cycles it belongs to. Therefore, for a node C, given a node B as input, it is straightforward to compute the next of the node C (i.e., $A = C.next(B)$) in the cycle $BCAB$.



(a) Second relay in the same cycle of the initiator.



(b) Second relay in a different cycle from the initiator.

Figure 6: Second relay selection.

Then, b_i encrypts the response for a by using the hop-by-hop key exchanged with a and forwards this message to $c_i = b_i.next(a)$. This encrypted message is encrypted, in turn, by b_i with the hop-by-hop key exchanged with c_i. At this point, c_i decrypts the message, invokes the function *next* to retrieve $a = c_i.next(b_i)$, encrypts the message again with its hop-by-hop key exchanged with a, and forwards it to a. Observe that, even though c_i knows that real traffic has to be forwarded to a from b_i, c_i does not know the content of it, and then it has no more information than b_i about the fact that a is the actual initiator or just an intermediate node of the tunnel. Once obtained the outgoing mimics of b_i, a selects a new mimic among them, say d_i, and needs to exchange a symmetric key and the set of outgoing mimics of d_i. Now, two cases may occur. The first case is that $d_i = c_i$ i.e., a, b_i, d_i are in the same cycle and d_i coincides with c_i. In this case, the list of mimics of c_i can be communicated directly through the link between c_i and a.

The second (complementary) case occurs when d_i has no common cycle with a. In this case, the list of mimics has to be forwarded from d_i to a through b_i. To enable the communication between d_i and b_i, since no direct link exists from d_i to b_i, we apply the approach discussed above. Specifically, d_i forwards this list through another node $e_i = d_i.next(b_i)$.

These two cases are represented in Figures 6a and 6b, respectively. Therein, we represent by a red arrow the forward communication between the initiator and the second relay of the tunnel, and by a green dashed arrow the backward communication from the second relay to the initiator. The building of the tunnel proceeds iteratively until the last node. Once the tunnel is set, the initiator can communicate with the recipient through this tunnel as in the standard Tarzan protocol.

Regarding the response by the recipient, the approach used to enable the exchange of information between a node of the tunnel and a previous node is applied. Specifically, at each hop of the tunnel start-

ing from the last node until the initiator, if a direct link exists between a node and a previous node of the tunnel, then the response is directly forwarded through this link, otherwise the response is forwarded through an intermediate node.

Concerning the security of C-Tarzan, as we will see in Section 8, the security results obtained for Tarzan are still valid in C-Tarzan. As a matter of fact, our protocol extends Tarzan with a feature that only changes the way in which the tunnel is built (including mimic selection), but preserves all the other features of the protocol, including the way the ingoing and outgoing traffic is set at each link (Freedman and Morris, 2002) to guarantee unobservability.

6 LATENCY COMPARISON

In the previous sections, we mentioned that our solution introduces a price in terms of latency, assuming the same cover traffic and the same tunnel length in Tarzan and C-Tarzan. To give an answer to the question of Section 4, we have to quantify this price.

To perform an analytic analysis, we use as a measure of this metric the number of hops traveled by a message in the forward path and in the return path.

We introduce the following notation. We denote by τ the average delay of the links of the network. We start by evaluating the latency for Tarzan. We denote by h the tunnel length of Tarzan and by L_f and L_r the latency of the forward path and the latency of the return path of Tarzan, respectively. Since the same tunnel is used both for the request and the response, it is easy to see that $L_f = L_r = (h+2)\cdot\tau$, where the term 2 derives from the fact that there is one hop between the last node of the tunnel and the PNAT and one hop from the PNAT and the destination.

Consider now C-Tarzan. We denote by h' the tunnel length and by L'_f and L'_r the latency of the forward path and the latency of the return path, respectively. For the forward path, no difference with Tarzan exists and then $L'_f = (h'+2)\cdot\tau$. On the other hand, for the return path, it is not trivial to estimate the number of hops, since it depends on the tunnel construction. We can provide an approximation of the return latency representing an upper bound of its actual value.

We omit the calculation and report the obtained results. For h' even, the latency of the return path of C-Tarzan is: $(\frac{h'}{2}\cdot(\frac{1}{d}\cdot 1+\frac{d-1}{d}\cdot 4)+2)\cdot\tau = (h'\cdot(2-\frac{3}{2\cdot d})+2)\cdot\tau$. On the other hand, for h' odd, the latency is: $((h'-1)\cdot(2-\frac{3}{2\cdot d})+4)\cdot\tau$. By considering equally likely the events that h' is odd and h' is even, we conclude that the return latency for C-Tarzan is: $L'_r = (h'\cdot(2-\frac{3}{2\cdot d})+\frac{3}{4\cdot d}+2)\cdot\tau$. Observe that L'_r increases as d increases. Indeed, as d increases, the probability that a mimic of the tunnel is selected in a different cycle increases. Then, the response requires more hops and the return latency increases.

7 EXPERIMENTS

Through this section, we perform an experimental validation of C-Tarzan by highlighting the conditions under which it outperforms Tarzan.

Metrics and Experiment Setting. As already introduced, we consider three metrics: cover traffic, latency, and cardinality of the anonymity set. Regarding the cover traffic, we use as a measure the number of ingoing and outgoing links of the nodes, by considering that every link concurs, on average, with the same portion of cover traffic. As discussed in Section 5, to obtain the same cover traffic in Tarzan and C-Tarzan, we have to set $k' = \frac{2}{3}\cdot k$. Regarding the latency, as seen in Section 6, to obtain the same total latency (forward latency plus return latency) we need to set h' such that $L_f + L_r = L'_f + L'_r$ i.e., $h' = \frac{2h-\frac{3}{4\cdot d}}{3-\frac{3}{2\cdot d}}$. However, since we are interested in studying what happens when the balance between uplink and downlink traffic varies, we introduce two coefficients w_f and w_r, such that $w_f + w_r = 2$, to associate with the forward latency and the return latency, respectively. For example, $w_f = w_r = 1$ represents a balanced traffic between uplink and downlink, while $w_f = 2$ and $w_r = 0$ represents only uplink traffic. Therefore, the condition to satisfy is $w_f\cdot L'_f + w_r\cdot L'_r = w_f\cdot L_f + w_r\cdot L_r$, that leads to: $h' = \frac{2\cdot h-\frac{3}{4\cdot d}\cdot w_r}{w_f+\frac{4\cdot d-3}{2\cdot d}\cdot w_r}$.

Now, we denote by $AS(k,h)$ the cardinality of the anonymity set of Tarzan obtained as a function of k and h. Furthermore, we denote by $AS'(k',h')$ the cardinality of the anonymity set of C-Tarzan obtained as a function of k' and h'. Thus, the question now is whether, by setting k' and h' according to the previous equations, it holds that AS' is greater than AS. If this is the case, then our approach introduces an advantage with respect to Tarzan.

The values of AS and AS' are computed via simulation. Furthermore, in order to obtain realistic results, we do not use directly the upper bound provided in Section 6, but we find experimentally the values of h and h' leading to the same latency for Tarzan and C-Tarzan, respectively (actually, verifying the results obtained in Section 6). To summarize, we find the values (h,k,h',k') that satisfy the following system.

$$\begin{cases} k' = \frac{2}{3} \cdot k \\ w_f + w_r = 2 \\ w_f \cdot L'_f + w_r \cdot L'_r = w_f \cdot L_f + w_r \cdot L_r \\ AS' \geq AS \end{cases} \quad (1)$$

In detail, the simulation has been performed in JAVA as follows. We considered a network of 100,000 nodes. First, we set some values of w_f (and, then, $w_r = 2 - w_f$), k', and h' for C-Tarzan and, then, we generated a topology (the links are obtained considering that each node selects *directly* k' mimics to build cycles). On this topology, we measured the average degree of each node counting both the actual ingoing and the outgoing links (cover traffic), the actual number of hops that a request and the corresponding response have to cross on a path of height h' (measure of latency), and the cardinality of the corresponding anonymity set. We repeated the experiment with the same parameters for 100 rounds (by varying the topology) to obtain steady results.

At this point, by the first equation of the system (1), we set $k = \frac{3}{2} \cdot k'$. Then, by using the value $w_f \cdot L'_f + w_r \cdot L'_r$ obtained experimentally for C-Tarzan and by recalling that $L_f = L_r = (h+2) \cdot \tau$, by the second and third equations of the system (1), we found the proper value of $h = \frac{w_f \cdot L'_f + w_r \cdot L'_r - 4 \cdot \tau}{2 \cdot \tau}$.

Then, we performed again 100 rounds of simulation with k, h to measure the cover traffic, latency, and anonymity set of Tarzan. We confirmed that the obtained values of cover traffic and latency are the same as C-Tarzan (with an error of less than 1 % for both). Therefore, we obtain an experimental validation of the fact that the first three equations of (1) hold.

Results. In this section, we compare Tarzan and C-Tarzan in terms of cardinality of the anonymity set, by setting the same cover traffic and same latency.

In the first analysis, we show as the cardinality of the anonymity set for both protocols varies as the cover traffic increases. We plot in the y-axis the ratio between the cardinality of the anonymity set of C-Tarzan AS' and the cardinality of the anonymity set of Tarzan AS. In the x-axis, we consider the degree d representing the number of outgoing (or ingoing) links in C-Tarzan (as defined in Section 6) that is equal to the number of bidirectional links in Tarzan (to obtain the same cover traffic). The results of this analysis are reported in Figures 7,8,9, for different values of h' and w_f.

We represent with a dashed black line the ratio equal to 1. When the plots exceed this line, C-Tarzan outperforms Tarzan (in terms of the cardinality of the anonymity set). We observe that our performance (for a fixed h') decreases as d increases. This happens because, as d increases, the latency of C-Tarzan increases, then the tunnel length of Tarzan h (that offers the same latency of C-Tarzan) increases too. Therefore, the cardinality of the anonymity set of Tarzan increases. Even though the cardinality of the anonymity set of both protocols has a polynomial growth with d, the exponential growth of the cardinality of the anonymity set of Tarzan with h is dominant. Therefore, as d increases, the ratio between AS' and AS decreases. Regarding w_f, as it increases (by considering the same d), the performance of C-Tarzan increases. This happens because an increasing weight w_f represents predominant uplink traffic that leads to lower total latency for C-Tarzan (since the return path is longer than the forward path). This implies that the

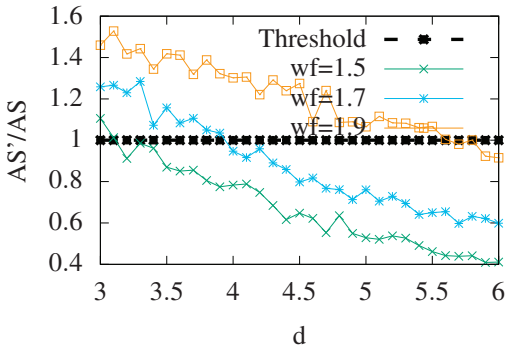


Figure 7: Anonymity set ratio vs cover traffic d with h'=3.

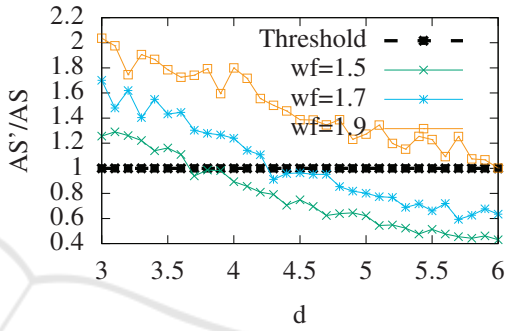


Figure 8: Anonymity set ratio vs cover traffic d with h'=4.

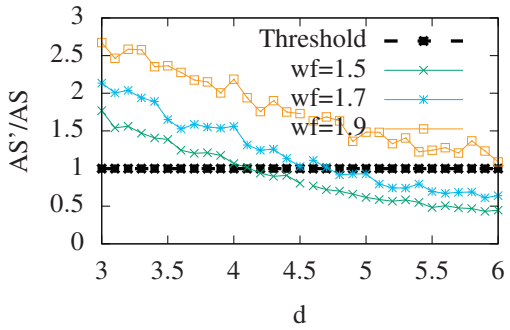


Figure 9: Anonymity set ratio vs cover traffic d with h'=5.

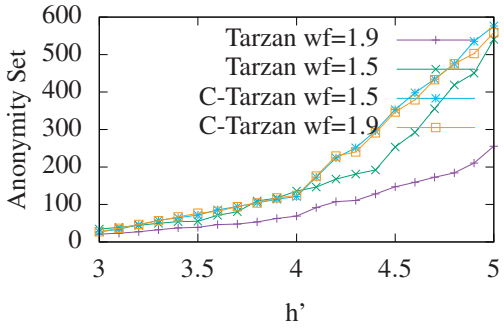


Figure 10: Anonymity set vs h' with d=4.

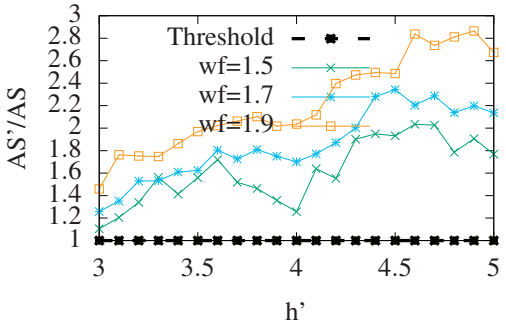


Figure 11: Anonymity set ratio vs tunnel length h' with d=3.

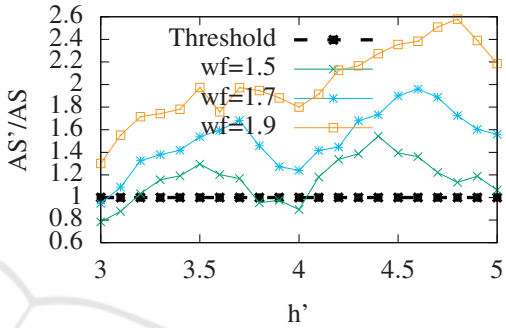


Figure 12: Anonymity set ratio vs tunnel length h' with d=4.

tunnel length h of Tarzan, which offers the same latency, decreases and then AS decreases too.

We observe that, until a certain level of cover traffic (corresponding to some d), it is advantageous to employ the C-Tarzan protocol, while when this threshold is exceeded, Tarzan is more convenient. Moreover, in the condition of increasing uplink traffic, this threshold also increases by making C-Tarzan suitable within a higher range of cover traffic level.

Observe that lower values of d are desirable since they represent cover traffic injected in the network. But, lower values of d result in an acceptable cardinality of the anonymity set in absolute terms (in relative terms C-Tarzan outperforms Tarzan). Indeed, as we discuss in the sequel, the anonymity set increases exponentially with h and h'. Then, with a small increment of h', we are able to obtain a good cardinality of the anonymity set still outperforming Tarzan. Just an example, with $d = 4$ and $h' = 4$, we obtain a cardinality of the anonymity set of about 100.

We conclude this section, by showing as the performances of C-Tarzan vary with respect to Tarzan as h' varies. The plot in Figure 10 shows AS and AS' as h' varies with two different values of w_f and $d = 4$. As expected, AS' increases exponentially with h'. Moreover, when h' increases, h increases too (to offer the same latency), and then also AS increases exponentially. Observe that AS' with $w_f = 1.5$ is essentially (modulo experimental error) the same as AS' with $w_f = 1.9$. Indeed, AS' does not depend on w_f. On the contrary, h depends on the total latency of Tarzan, which is equal to the total latency of C-Tarzan that, in turn, depends on w_f. Therefore, as w_f increases, h decreases and AS decreases too.

To conclude this section, in Figures 11, 12, and 13, we show the ratio between the anonymity set of Tarzan and C-Tarzan as h' varies for different values of w_f and d.

According to the previous analysis, C-Tarzan outperforms Tarzan for low d and for increasing w_f. Regarding h', we observe a fluctuating behaviour in which there are some ranges of h in which there is an increasing trend of the ratio and other ranges in which there is an opposite trend. This is due to a compensation effect between the growth of the cardinality of the anonymity set and the latency. In particular, for C-Tarzan, when h' increases, AS' increases, and the total latency increases too. Anyway, in some ranges, the increment of latency is limited. This leads to an increment of the tunnel length of Tarzan h that is not sufficient to obtain a cardinality of the anonymity set AS which compensates for the growth of AS'. On the contrary, once h' reaches a peak value, the effect of the growth of the latency assumes a more relevant role by leading to values of h corresponding to the cardinality of the anonymity set AS able to compensate for the growth of AS'. As a final remark, observe that, in this analysis, we show the advantage of our approach

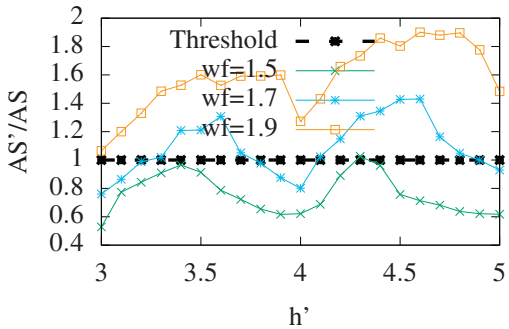


Figure 13: Anonymity set ratio vs tunnel length h' with d=5.

just in terms of cardinality of the anonymity set (under the same latency and cover traffic level). Clearly, this advantage can be translated into an advantage in terms of latency or cover traffic, by fixing the same cardinality of the anonymity set for both protocols.

8 SECURITY ANALYSIS

In this section, we analyze the security of C-Tarzan by following the same approach as the security analysis of Tarzan (Freedman and Morris, 2002).

P2P Model. As in Tarzan, we start by analyzing the security aspects related to the P2P nature of the protocols. As in (Freedman and Morris, 2002), our protocol aims to hide: *sender activity*, *sender content*, *recipient activity*, and *recipient content*. We say that an adversary detects sender activity when it discovers that the sender is sending something. Instead, it detects sender content when it links sender activity to the plaintext of a message. By definition, the exposition of sender content implies the exposition of sender activity. Similar definitions apply to recipient activity and recipient content.

Now, we consider two attackers: *static adversary* and *adaptive adversary*. A static adversary has the capability to corrupt a certain number of nodes in the P2P network before observing any system behavior. It can inspect packets and it can conduct timing analyses to determine the correlation between packets from the same tunnel seen at different relays. Clearly, the same happens for C-Tarzan. As for the adaptive adversary, in addition to the capabilities of a static adversary, it is able to choose which machine to compromise after observing any system behavior. Obviously, in principle, such an adversary would be able to discover sender (recipient) activity and sender (recipient) content if it has the capability to compromise nodes in a very short time. Hence to protect against an adaptive adversary, (Freedman and Morris, 2002) suggests choosing the duration of the tunnel to be less than the time to compromise a single node in the P2P network. Moreover, each tunnel should be built including different nodes each time. For C-Tarzan, there is no reason to require a tunnel lifetime greater than Tarzan tunnels. Moreover, also for C-Tarzan, nodes for tunnels can be refreshed each time. Therefore, the above mitigations can be applied also in the case of C-Tarzan. Coherently with the analysis of Tarzan, for the above reasons, the security analysis only considers the static adversary. Since all nodes can both originate and forward traffic, a malicious node included in a tunnel can just guess that its predecessor in a tunnel is the actual sender of an observed message with some confidence. This confidence is estimated in (Freedman and Morris, 2002) via probabilistic analysis. Due to space reasons, in this paper, we cannot include a similar analysis for C-Tarzan. However, given the fact that more mimics are present in C-Tarzan (to obtain the same cover traffic), we can argue that also the level of confidence obtained from a malicious node in C-Tarzan is lower than in Tarzan. Although we do not have space to prove this claim, we can give the intuition. When a malicious node is selected as a member of a tunnel, having more mimics in the network implies more nodes as possible predecessors (of the malicious node) in the tunnel. Even though in C-Tarzan, not all the possible predecessors of a node have the same probability to be the actual predecessor, in our case, we can say that this probability is at least the same as in Tarzan. Then we can conclude that the level of confidence regarding the sender activity provided by C-Tarzan is at least the same as that provided by Tarzan. We can state that C-Tarzan offers the same protection as Tarzan to sender (recipient) activity and sender (recipient) content. Specifically, if a node included in a tunnel is compromised, sender activity can be exposed only by guessing that its predecessor is the actual sender and this can only be done with a certain confidence level. Concerning sender content, since only a PNAT can read the plaintext content of a sender's message, it can only be exposed if the first and last nodes of a tunnel are both compromised. However, regarding the compromise of the first node, the same reasoning as above applies. Therefore, sender content can only be discovered with a certain probability. Finally, as for recipient content and recipient activity, it is sufficient to compromise the last relay to expose them both. The next aspect analyzed in (Freedman and Morris, 2002) regards mimic selection. Considering C-Tarzan, we first show that our mimic selection process does not introduce any threat with respect to the original Tarzan protocol. To this aim, we first introduce some notations. Say M the number of malicious domains and N the overall number of domains in the P2P network.

The authors in (Freedman and Morris, 2002) show that, for Tarzan, the following holds: **Claim 1**: Nobody can bias an initiator's choice of relays; and **Claim 2**: A node selects a malicious mimic with probability M/N.

In the following, we show that these claims hold for C-Tarzan too. Regarding **Claim 1**, malicious nodes may attempt to bias the initiator's mimic selection to increase its frequency of using malicious relays in its tunnel. However, we can demonstrate that if **Claim 1** holds for Tarzan, it also holds for C-Tarzan.

To show this, let us consider a node a that has to

select its mimics. Recall that, for each mimic, say b, directly selected by a, a mimic c is indirectly selected by b to close the cycle. As for b, there is no difference with respect to Tarzan. In other words, it is chosen through the same publicly verifiable procedure (i.e., the $lookup$ function with the IP address of a as input). Therefore, no control on the choice of b is given to a. Therefore, we have to check if the selection of c introduces some threats. Observe that also c is selected by using the same publicly verifiable procedure. This time, the lookup function is computed with $a||b$ as input. Clearly, there is no difference with respect to the previous selection in terms of capability of the adversary (even in the case of collusion between a and b) of the node c, because everyone (including c) can verify the expected output of the $lookup$ function. Hence, we conclude that **Claim 1** holds also for C-Tarzan.

Concerning **Claim 2**, we make the following considerations. In Tarzan, the probability to select a malicious mimic is M/N. This is due to the fact that, as explained in Section 3, mimics for a node are randomly selected in different IP domains via a three-level hierarchical DHT. Hence, an adversary controlling an entire domain M (thus generating a huge number of malicious nodes in that domain), does not have a greater probability that a malicious node of such domain is selected as a mimic. This also holds for C-Tarzan, since we also adopt the same DHT for the mimic selection process. Moreover, being **Claim 1** also true for C-Tarzan, the selection of a malicious mimic in a cycle does not increase the likelihood to include another malicious mimic in the same cycle. Therefore, **Claim 2** holds for C-Tarzan.

Traffic Analysis. Another security aspect investigated in (Freedman and Morris, 2002) is Traffic analysis. This analysis focuses on: (1) the way in which the traffic is managed by nodes and (2) the likelihood that the adversary may guess the position of the initiator in a tunnel via traffic analysis.

Concerning (1), in C-Tarzan we adopt the same strategies as Tarzan (i.e., the management of ingoing and outgoing traffic flows as in Section 3.7.3 of (Freedman and Morris, 2002)), then no security issue arises in C-Tarzan not already addressed in Tarzan.

Regarding (2), we observe that in C-Tarzan the forward message is sent through a Tarzan-like tunnel. Then the adversary capabilities to perform traffic analysis on the forward path are the same as Tarzan. Unlike Tarzan, the return path involves only a subset of nodes involved in the forward path, plus some additional nodes. First, observe that the power of traffic analysis attacks (aimed at identifying the initiator) performed by multiple malicious nodes in a tunnel is highly increased when these nodes are able to correlate forward packets with the corresponding responses. This is possible, in principle, in Tarzan, because forward and return paths coincide. Conversely, this capability appears much less likely in C-Tarzan, in which forward and return paths differ from each other and involve different sets of nodes.

9 CONCLUSION

In this paper, we propose an approach to improve the effectiveness of P2P mixnet-like approaches for uplink-intensive applications. The core idea of our proposal consists of moving from bidirectional to unidirectional cover traffic by arranging cyclic tunnels from the sources to the destinations. To show its applicability, we take as a reference mixnet the Tarzan protocol and study how it can be extended by adopting our cyclic approach. This led to the definition of a new protocol, called C-Tarzan. The performance of C-Tarzan is evaluated according to the three main metrics in the field of anonymous communications: latency, amount of cover traffic, and cardinality of the anonymity set. We performed an in-depth experimental validation highlighting the conditions under which it is more advantageous to employ C-Tarzan instead of Tarzan. The main result, arising from the conducted analysis, is that C-Tarzan outperforms Tarzan in terms of cardinality of the anonymity set above uplink-traffic thresholds, depending on the value of the other parameters, until a relevant improvement for uplink-intensive applications. This confirms the general validity of our cyclic approach, even though the computed thresholds are referred to the chosen mixnet (i.e, Tarzan). Obviously, if we keep the same anonymity set, cover traffic is reduced. This is relevant, because cover traffic means overhead, also in terms of energy consumption. Being the solution (and Tarzan too) designed for large-scale networks, this aspect has a practical impact and is meaningful. As resulting from the experiments, this advantage is obtained for low degrees of mixnet. However, this is coherent with the above consideration, because increasing the degree of the nodes of the mixnet involves either increasing the overall cover traffic or (if the cover traffic is not increased) reducing the bandwidth of each tunnel, thus increasing latency. Thus, with our solution, we are able to keep the degree low in favor of bandwidth and overall cover-traffic saving, obtaining better results in terms of anonymity.

REFERENCES

Alexopoulos, N., Kiayias, A., Talviste, R., and Zacharias, T. (2017). Mcmix: Anonymous messaging via secure multiparty computation. In *26th {USENIX} Security Symposium ({USENIX} Security 17)*, pages 1217–1234.

Beimel, A. and Dolev, S. (2003). Buses for anonymous message delivery. *Journal of Cryptology*, 16(1).

Ben Guirat, I., Gosain, D., and Diaz, C. (2021). Mixim: Mixnet design decisions and empirical evaluation. In *Proceedings of the 20th Workshop on Workshop on Privacy in the Electronic Society*, pages 33–37.

Berger, S., Simsek, M., Fehske, A., Zanier, P., Viering, I., and Fettweis, G. (2015). Joint downlink and uplink tilt-based self-organization of coverage and capacity under sparse system knowledge. *IEEE Transactions on Vehicular Technology*, 65(4):2259–2273.

Centenaro, M. and Vangelista, L. (2015). A study on m2m traffic and its impact on cellular networks. In *2015 IEEE 2nd World Forum on Internet of Things (WF-IoT)*, pages 154–159. IEEE.

Chaum, D. L. (1981). Untraceable electronic mail, return addresses, and digital pseudonyms. *Communications of the ACM*, 24(2):84–90.

Cheng, R., Wu, N., Chen, S., and Han, B. (2022). Will metaverse be nextg internet? vision, hype, and reality. *arXiv preprint arXiv:2201.12894*.

Danezis, G. and Diaz, C. (2008). A survey of anonymous communication channels. Technical report, Technical Report MSR-TR-2008-35, Microsoft Research.

Das, D., Meiser, S., Mohammadi, E., and Kate, A. (2018). Anonymity trilemma: Strong anonymity, low bandwidth overhead, low latency-choose two. In *2018 IEEE Symposium on Security and Privacy (SP)*, pages 108–126. IEEE.

Dester, P. S., dos S Filho, F. H. C., and Cardieri, P. (2018). Performance analysis of uplink traffic for machine type communication in wireless sensor networks. In *2018 IEEE 87th Vehicular Technology Conference (VTC Spring)*, pages 1–5. IEEE.

Dingledine, R., Mathewson, N., and Syverson, P. (2004). Tor: The second-generation onion router. Technical report, Naval Research Lab Washington DC.

Freedman, M. J. and Morris, R. (2002). Tarzan: A peer-to-peer anonymizing network layer. In *Proceedings of the 9th ACM Conference on Computer and Communications Security*, pages 193–206.

Hirt, A., Jacobson, M., and Williamson, C. (2008). Taxis: scalable strong anonymous communication. In *2008 IEEE International Symposium on Modeling, Analysis and Simulation of Computers and Telecommunication Systems*, pages 1–10. IEEE.

Karunanayake, I., Ahmed, N., Malaney, R., Islam, R., and Jha, S. (2020). Anonymity with tor: A survey on tor attacks. *arXiv preprint arXiv:2009.13018*.

Kotzanikolaou, P., Chatzisofroniou, G., and Burmester, M. (2017). Broadcast anonymous routing (bar): scalable real-time anonymous communication. *International Journal of Information Security*, 16(3):313–326.

Kwon, J.-H., Lee, H.-H., Lim, Y., and Kim, E.-J. (2016). Dominant channel occupancy for wi-fi backscatter uplink in industrial internet of things. *Applied Sciences*, 6(12):427.

Nikaein, N., Marina, M. K., Manickam, S., Dawson, A., Knopp, R., and Bonnet, C. (2014). Openairinterface: A flexible platform for 5g research. *ACM SIGCOMM Computer Communication Review*, 44(5):33–38.

Oueis, J. and Strinati, E. C. (2016). Uplink traffic in future mobile networks: Pulling the alarm. In *International Conference on Cognitive Radio Oriented Wireless Networks*, pages 583–593. Springer.

Piotrowska, A. M., Hayes, J., Elahi, T., Meiser, S., and Danezis, G. (2017). The loopix anonymity system. In *26th {USENIX} Security Symposium ({USENIX} Security 17)*, pages 1199–1216.

Shafiq, M. Z., Ji, L., Liu, A. X., Pang, J., and Wang, J. (2013). Large-scale measurement and characterization of cellular machine-to-machine traffic. *IEEE/ACM transactions on Networking*, 21(6):1960–1973.

Shen, T., Jiang, J., Jiang, Y., Chen, X., Qi, J., Zhao, S., Zhang, F., Luo, X., and Cui, H. (2021). Daenet: Making strong anonymity scale in a fully decentralized network. *IEEE Transactions on Dependable and Secure Computing*, pages 1–1.

Shirazi, F., Simeonovski, M., Asghar, M. R., Backes, M., and Diaz, C. (2018). A survey on routing in anonymous communication protocols. *ACM Computing Surveys (CSUR)*, 51(3):1–39.

Sun, Y., Liu, Q., Chen, X., and Du, X. (2020). An adaptive authenticated data structure with privacy-preserving for big data stream in cloud. *IEEE Transactions on Information Forensics and Security*, 15:3295–3310.

Van Den Hooff, J., Lazar, D., Zaharia, M., and Zeldovich, N. (2015). Vuvuzela: Scalable private messaging resistant to traffic analysis. In *Proceedings of the 25th Symposium on Operating Systems Principles*, pages 137–152.

Xia, Y., Chen, R., Su, J., and Zou, H. (2020). Balancing anonymity and resilience in anonymous communication networks. *Computers & Security*, page 102106.

Yang, H. and Larsson, E. G. (2019). Can massive mimo support uplink intensive applications? In *2019 IEEE Wireless Communications and Networking Conference (WCNC)*, pages 1–6. IEEE.

Young, A. L. and Yung, M. (2014). The drunk motorcyclist protocol for anonymous communication. In *2014 IEEE Conf. on Communications and Network Security*, pages 157–165. IEEE.

Zantout, B., Haraty, R., et al. (2011). I2p data communication system. In *Proc. of ICN*, pages 401–409. Citeseer.