electronics

MDPI

Article

A Privacy-Preserving Routing Protocol Using Mix Networks in Opportunistic Networks

Depeng Chen [1,2,*], Carlos Borrego [3] and Guillermo Navarro-Arribas [1,2]

1 Department of Information and Communications Engineering, Universitat Autònoma de Barcelona, 08193 Bellaterra, Spain; guillermo.navarro@uab.cat
2 CYBERCAT—Center for Cybersecurity Research of Catalonia, 43003 Tarragona, Spain
3 Department of Mathematics and Computer Science, Universitat de Barcelona, 08007 Barcelona, Spain; carlos.borrego@ub.edu
* Correspondence: depeng.chen@uab.cat

Received: 15 August 2020; Accepted: 19 October 2020; Published: 22 October 2020

Abstract: This paper focuses on the problem of providing anonymous communications in opportunistic networks. To that end, we propose an approach using Mix networks that enables a relatively simple solution. Opportunistic networks present some constraints that make the deployment of typical network anonymity solutions difficult or infeasible. We show, utilizing simulations on the basis of real mobility traces, that the proposed solution is feasible for some scenarios by introducing a tolerable penalty in terms of message delay and delivery. To investigate the impact of routing strategies, we offer two different methods to select Mix nodes. From the experiment results, we show the trade-off between network performance and security.

Keywords: opportunistic networks; mix networks; privacy

1. Introduction

We live during a global deployment of computer networks and network devices. This technology has allowed for the creation of a wide range of different network-based applications that enable collaborative behavior among users to share their personal interests, such as their hobbies and professions. During the last few years, the research community has proposed different solutions to allow for mobile nodes to communicate with each other, even in extreme cases where no end-to-end connection is guaranteed. This network paradigm has been termed as opportunistic networks (OppNets) [1]. In these opportunistic social networks, users carry mobile devices to communicate with each other and share data as in traditional social networks. Nevertheless, when it comes for privacy in OppNets, not all that glitters is gold.

There is a range of applications in OppNets, such as crisis management, battlefield coordination, wildlife monitoring, transportation engineering, and remote healthcare. They are typically used in environments that require networks to be tolerant of long delays, interruptions, and high error rates. Nodes in these networks communicate with each other in an opportunistic manner. It is common to use routing strategies on the basis of the so-called store–carry–forward strategies. That is, a node can store and carry a message until it finds another node to forward it.

Because of the unstable connection property of OppNets, it is difficult to directly use traditional network privacy and anonymity mechanisms. Current proposals are mostly centered on providing extensions or strategies to adapt onion routing for OppNets. Although these could be interesting approaches, some OppNets scenarios need to address the problem of traffic analysis. Contrary to the use of onion routing on the Internet, for example, in some OppNets scenarios, having an attack model where the attacker has access to the whole communication medium is not rare. Typical OppNets use

wireless connections where traffic analysis imposes a dangerous threat. Attacks can exploit message flow to detect the whole onion routing path, which leaks the communication pattern between sender and receiver, assuming that all network traffic is not extremely high. A common approach to deal with this problem is to generate noisy network traffic to mask actual traffic. Another alternative that has not been investigated in depth is the use of mix networks (mixnets). In this paper, we propose the use of a mixnet mechanism to provide privacy in opportunistic networks.

Mixnets can be used to provide anonymous communication in OppNets without high impact on the normal operation of the network. Some nodes from the network act as mix nodes enabling the implementation of different strategies. To the best of our knowledge, our anonymous schema is the first work to propose the use of mixnets in OppNets. Our main contributions are the following: First, we designed a mixnet-based communication approach for OppNets. Second, we validated and showed the particularities of our proposed schema in the ONE simulator (a common network simulator especially focused towards OppNets). Third, we analyzed the performance and privacy property of our proposed mechanism.

This paper is organized as follows. In Section 2, we introduce the current related work. We then present our anonymous-routing mechanism using mixnets in Section 3. Section 4 describes our experiment simulation on the ONE simulator. Lastly, in Section 5, we conclude our proposed work.

2. Motivation and Related Work

In OppNets, we can have unstable nodes with unpredictable movements producing a regularly changing network topology. Hence, end-to-end routing paths between source and destination nodes may be difficult to establish. The reliability of data transmission and network-traffic control, which are provided by traditional transport protocols such as TCP, are very inefficient [2] in OppNets. This also means that, in order to provide privacy and anonymous communication services, traditional network mechanisms cannot be directly applied [3].

Several proposals addressed anonymous communications in OppNets, and most of them are based on some application of onion routing. For instance, ARDEN [4] proposes an attribute-based encryption method to provide strong anonymity guarantees. It is based on a traditional onion-routing architecture where nodes can act as onion routers. ARDEN divides nodes into several groups, and each group shares the same key. Thus, nodes from the same group can forward the message, which can improve delivery probability.

Sakai et al. [5] designed an onion-based routing protocol and extended the existing protocol with group onions into multicopy forwarding. The main contributions of this paper were performance and security analyses of onion-based anonymous routing for DTNs.

In [6], Chen et al. proposed an onion-based scheme to ensure anonymous communications in predictable opportunistic networks (POppNets). A POppNet is a network where end-to-end connectivity is not guaranteed, and node communication happens in an opportunistic manner, but the behavior of the network can be predicted in advance. The predictability of such networks can be exploited to simplify some mechanisms of more generic OppNets where there is no prior knowledge on the network behavior.

Alternatively, [7] provided a new structure for the history table of nodes in an OppNet in which the personal information of participants in the network is hidden from other nodes. To send a message from a sender node to the final receiver node, the optimized route, composed of different nodes, is identified by inspecting the history table that was created by each node on the basis of previous interactions. In this paper, a new privacy-preserving history-based (PPHB) routing mechanism was proposed on the basis of historical location tracking. Bakiras et. al used a random walk process and encrypted exchanged message to ensure anonymous communications in OppNets [8]. Their work leveraged the simple opportunistic mechanism to store–carry–forward the message from end to end.

In general, and particularly in solutions based on onion routing, some problems might arise if we consider traffic analysis attacks. In an OppNet, the transmission medium is usually the air,

making it relatively easy to gather traffic data by anyone with physical proximity to the devices. If we consider an adversarial model where the attacker has access and can monitor all network interactions, traffic analysis techniques can be used to correlate source and destination in common onion routing based approaches. This adversarial model is relatively realistic for some scenarios. Consider, e.g., an OppNet built for a conference where nodes move around a relatively small physical space. OppNet routing protocols usually provide some sort of traffic analysis resistance, in the sense that a packet might be broadcast or sent to nodes which are neither the destination nor an intermediate node in the final delivery path. This does not prevent the identification of the message source, and in most cases does not prevent the attacker from gaining some knowledge about intermediate and destination nodes.

In order to provide an anonymous communication method that is robust enough against this adversarial traffic analysis model, we considered the use of mixnets for OppNets. The idea of using mixes providing anonymous communication was first proposed by Chaum in 1981 [9]. In Chaum's untraceable system, mix nodes encrypt messages, and use the stack to store and shuffle the messages. Freedman proposed a peer-to-peer mix network called Tarzan [10] that exploits layered encryption and multihop routing to achieve anonymous communication. We introduce a similar approach for OppNets, considering the particularities of these networks and, more precisely, the typical routing strategies used in these networks.

3. Anonymous Routing Using a Mix Network

Our proposal is based on the use of a mixnet in OppNets. The main idea is to be able to use a relatively simple mixnet schema without requiring complex cryptographic mechanisms. This approach is not only is feasible in OppNet scenarios, but can also set the basis for interesting solutions to anonymous communications in dynamic networks.

3.1. Opportunistic Mix Network

We are dealing with networks where nodes can move, appear, and disappear, and all links are opportunistic. Following the notation from [6] we can see an OppNet as an undirected dynamic graph $G(V, E)$, where V is the set of nodes and E the set of edges. Each edge denotes a connection between two nodes that can be used in both directions during a time range. Usually, an edge is denoted as $e = (u, v, t, \lambda)$, showing that it starts at time t and has duration λ. Nodes can also appear and disappear from the network; without loss of generality, we can assume that the disappearance of a node is captured by the fact that it has no edges (both having a node without connections or the fact that the node disappears is equivalent for our purposes). Graph $G^*(V^*, E^*)$ is the static undirected graph obtained from G by considering all its edges without time constraints (all edges are in the graph independently of time); we have that $E^* = E$. We used $N^*(v)$ to denote all neighbors of node $v \in V*$ in G^*, that is, $N^*(v) = \{u \mid (v, u, t_i, \lambda_i) \in E, \forall i\}$.

In order to implement the mixnet, each network has a set of m mix nodes $\mathcal{M} = M_1, \ldots, M_m$, such that $\mathcal{M} \subseteq V$. Our proposal exploits the fact that there could be several mix nodes, and the source node is free to choose which ones it uses to send a message. Moreover, the sender can choose the number of mix nodes to use in a cascade fashion (network) from set $\mathcal{M}$. In this sense, our approach follows what could be denoted as a restricted free route mix network—it is a free route mixnet [11,12]—but the selection of nodes is limited to a subset of the nodes of the network. We assumed that there was a key distribution mechanism, so each mix node had an asymmetric key pair, and the public key of each mix node was known to all nodes.

We considered a typical decryption mixnet. For example, consider sender node V_s that chooses three mix nodes, M_1, M_2, M_3, to send message x to destination V_d. Each node has a corresponding public key $PK_s, PK_1, PK_2, PK_3, PK_d$. To that end, V_s builds an onion-based encryption scheme, such as

$$PK_1(r_1, PK_2(r_2, PK_3(r_3, PK_D(r_d, x))))$$

where r_i are random nodes ensuring that messages cannot be correlated to their encrypted versions in any step. The message is received by M_1, decrypted, and sent to M_2 after performing the mix. Each node does the same until the message arrives at destination V_d. Each node V_i waits until it has received k_i messages, and then forwards these k_i messages, making it hard for an attacker to correlate input messages with output messages.

In the general case, mix nodes are a subset of the whole network nodes, $\mathcal{M} \subseteq V$. The source node selects a path of mix nodes $M_i \in \mathcal{M}$ of length p, and encrypts message x as

$$PK_1(r_1, \ldots PK_p(r_p, PK_D(r_d, x)) \ldots)$$

We assumed that each mix node had enough capacity to perform the cryptographic operations, and memory to store the required k_i messages. Which nodes act as mix nodes depends on the specific scenario and the parameterization of our approach. If we assume that we can choose such nodes, this can have implications in several ways. Having more mix nodes could provide more privacy. The sender can choose randomly from the set of mix nodes, and thus make it harder for an attacker to follow the message through the network. On the other hand, having many mix nodes can enlarge the delay associated with each mix since each node needs to wait for k_i messages. We can also consider other approaches to select potential mix nodes; for instance, it seems reasonable that those nodes with more interactions are obviously more interesting to use. We discuss this in Section 3.3.

3.2. Message Routing

Routing messages in our opportunistic mixnet approach differs greatly from a common mixnet implemented in a more classical network. Messages might not be easily routed, and source routing cannot generally be used. Thus, when the source node chooses the path of mix nodes, delivering the message to such nodes might require routing it through other intermediary nodes. To that end, our proposal uses epidemic routing.

In epidemic routing, each node forwards a message to all its neighbors until the message reaches its destination. It can produce much flooding in the network and, although it is not the most efficient routing strategy for OppNets, it is the most generic and basic approach, and served as a good basis to evaluate our proposal.

As an example, Figure 1, shows a possible message delivery from source node V_s to destination node V_d. The source node chooses three mix nodes, M_1, M_2, M_3. The delivery of the message needs to use intermediary nodes, denoted as V_i for $= 1, \ldots, 8$. The dashed lines denote epidemic routing, which is used to deliver the message from V_s to M_1, then from M_1 to M_2, and so on. We denoted these routes between each mix node, and source and destination nodes as routing steps. The number of intermediate nodes used in each routing step depend on the network behavior at the moment. As described before, each mix node has to wait until receiving k_i messages, which introduces an additional penalty to message-delivery time.

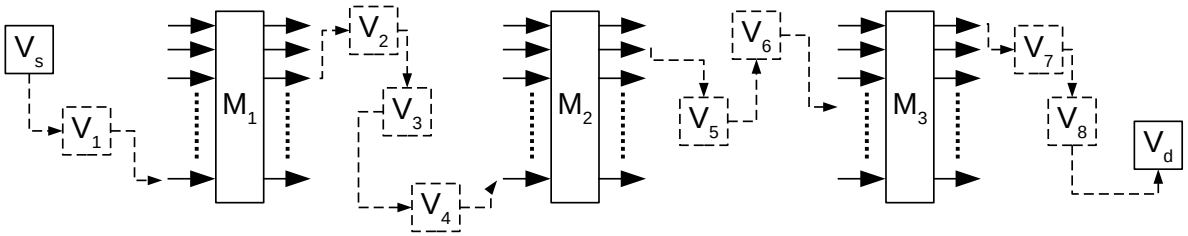


Figure 1. Example of opportunistic mixnet.

3.3. Choosing Mix Nodes

As described before, the localization of mix nodes in the network is an important issue. In order to evaluate our opportunistic mixnet proposal, we chose a set of network nodes $\mathcal{M}$ that could act as mix

nodes. The size of $\mathcal{M}$, denoted as $|\mathcal{M}|$, or more precisely its size with respect to $|V|$, determines the overall performance of the mixnet. Apart from its size, another important decision is how to choose such a set $\mathcal{M}$. There are existing routing strategies in free route mixnets to optimize the anonymity of the system [13]. These strategies, however, assume that all nodes are reachable, and that routing does not opportunistically occur. In order to evaluate the use of our proposal, we provide two selection strategies focused on the performance of the opportunistic mixnet.

The two different strategies that can be used to select set $\mathcal{M}$ are:

- Random selection: $\mathcal{M}$ is randomly selected among all nodes V. This is a relatively simple idea where nodes acting as mixes can be any node from the network. No other characteristic is considered.
- Centrality-based selection: $\mathcal{M}$ is selected as the $|\mathcal{M}|$ nodes from V with higher centrality. Here, the idea is to promote the use of highly connected nodes to act as mixes. This should improve the performance of the overall network. In order to simulate this idea, we look at nodes with higher centrality. As the centrality measure, we used the historic number of neighbors of each node. That is, a node with more contacts over time has higher centrality. In our case, we used a connection-based centrality measure that captured how many connections or interactions have a given node. More precisely, the centrality of a node $v \in V$ is denoted as $C(v) = |N^*(v)|$. In real situations, this could be equivalent to selection nodes that we know that have higher centrality or connectivity, e.g., in some vehicular networks, road-side units are known by every member of the network, and they are also central nodes.

Intuitively, the centrality-based strategy yields better delivery ratios and delivery time. Those nodes have better connectivity, and messages are more efficiently routed among them.

The size of set $\mathcal{M}$ with respect to V also has performance and privacy implications, but such implications might not be that obvious. For instance, a big set of potential mix nodes imply:

- Increased privacy: as more nodes can be used as mix nodes, the attacker needs to monitor more nodes attempting to correlate inputs with outputs. This introduces confusion as a means of privacy. Privacy is actually determined by the number of messages that each mix node accepts and the number of mix nodes each message uses. The fact that there are more mix nodes present in the network hinders the task of the attacker in monitoring such nodes, and can provide better privacy due to the distribution of links in the entire mix network [13].
- More time delays: as there are more mix nodes that can be used, filling each node with its corresponding k_i messages takes more time, and the delay introduced by each mix node is expected to be higher. This obviously depends on the actual traffic of the network (number of messages exchanged over time), but in general, more mix nodes produce higher delay times.

3.4. Threat Model

The objective of the adversary is to link the sender and receiver in our OppNets system. Specifically, we assumed that the attacker could access our OppNets system. It can wiretap, forward, and delete a message. Meanwhile, the attacker can also compromise the user and mix nodes in the system. However, we assumed that at least some mixes were honest.

Our threat model was based on [14]. There is no perfect security defender that could resist all kinds of attacks. In our assumption, we mainly considered three different threat types.

- Traffic analysis. Attacker A can conduct a traffic analysis to probe the link between sender node S and receiver node D. A can obtain access to every single message M, and its goal is to speculate that S and D are communicating with each other in high probability.
- Compromised user nodes. We also assumed that attacker A could compromise user nodes. Thus, A can obtain message M and forward it. In our scenarios, we assumed that at least two users were honest.

- Compromised mix nodes. Mix nodes can also be compromised, but at least one of them should be honest. The compromised mix nodes can conduct a link attack to deanonymize the system.

4. Evaluation

In this section, we evaluate our opportunistic-mixnet proposal. We first introduce the scenarios and simulation environment. Then, we describe three metrics to evaluate the performance. Lastly, we present the results.

4.1. Simulation Environment and Scenarios

We present an experiment using an enhanced version of the Opportunistic Network Environment (ONE) simulator [15] that included our anonymous mixnet routing. We chose different scenarios to analyze the performance of our schema. Node contacts from the scenarios were defined by physical contacts obtained from real mobility traces from the Crawdad database (http://crawdad.org/), a community resource for collecting wireless data at Dartmouth College. Each scenario corresponded to a different network with different nodes, and different mobility and contact patterns.

The first scenario, Info5, was based on real mobility traces [16] obtained during the 2005 edition of the Infocom conference over the course of almost three days. Contacts from these mobility traces represent 22,459 contacts from 41 different nodes. The second scenario, Cambridge, was based on 10,641 real contact traces from 51 students from the system research group of the University of Cambridge carrying small devices for six days [17]. Lastly, the third scenario, Taxis [18], contained 449,226 mobility contacts from 304 taxis during one month in the city of Rome.

Table 1 summarizes the main characteristics of the three scenarios. These include the connections per minute, which is the average number of connections a network has in one minute during the whole duration of the scenario; total connection time, which is the sum of the duration of all connections in the network, $\sum_{e_i \in G} \lambda(e_i)$; and the maximal and average degree of all nodes $v \in V^*$, where the degree of the node is determined by $N^*(v)$ (see Section 3.1).

Table 1. Characteristics of 3 scenarios.

	Scenarios		
	Cambridge	**Info5**	**Taxis**
Number of nodes	51	41	304
Connections per minute	0.9979	8.035	15.13
Total connection time (s)	1,383,082	2,145,974	6,189,716
Max node degree	12	14	12
Average node degree	0.2996	0.6433	0.4713

These scenarios are commonly used in the evaluation of routing algorithms in the OppNet literature. They provide a realistic setup to measure the feasibility of our proposal.

In our experiment, we set the simulation time of 86,400 s (24 h). That is, we simulate each scenario for one day.

4.2. Performance Metrics

One of the main concerns when using not only mixnets, but any means to provide anonymous communications in a network, is the penalty that such solutions could worsen overall network performance. This is especially relevant in OppNets, where we already do not usually have good performance in message delivery. The use of mix nodes introduces latency in message delivery and decreases the delivery ratio. The goal is to see whether these penalties are tolerable. To that end, we measured message latency and delivery ratio in several simulations for the three proposed scenarios, and using different setups and parameterizations.

We mainly focused our experiments to observe:

- Message latency: the average time that it takes for a message to reach its destination.
- Delivery ratio: ratio of successfully delivered messages.
- Overhead ratio: the factor of number of relayed messages minus delivered messages, divided by the number of delivered messages. The overhead ratio shows the number of network resources used in the process of delivering a message to its receiver.
- Average routing-step path length: the average number of hops that one packet needs to traverse in each routing step using epidemic routing (see Section 3.2). That is, the number of nodes used to route a message from the source node to the first mix node, from the first mix node to the second, and so on.

There are two main parameters that influence how our proposal is deployed in a given scenario. One is the percentage of mix hosts, i.e., the percentage of nodes that can be chosen as mix hosts or the size of $\mathcal{M}$ with respect to V. These are 20%, 40%, 60%, 80%, and 100%.

The other parameter is the number of mix nodes used in cascade. That is, the number of mix nodes that each message gets through. In our experiment, we set the number of mix nodes from 0 to 6. Here, the value set to 0 means using the general broadcast method. We used simple broadcast to transmit our message from one mix to another. To simulate a real OppNet environment, the TTL (time-to-live) of each message was set to a relatively large value of 5.5 h (20,000 s). We assumed that each node had enough buffer to store the messages. During the simulation, every 100 s, a randomly selected node sent a 1 byte message to a randomly chosen node. This was performed in all three scenarios. In our simulation, we set parameter k_i to 10 for each mix node i mentioned in previous sections.

4.3. Results

Running the simulations, we could measure the introduced penalty by using the opportunistic-mixnet approach in the three above mentioned scenarios.

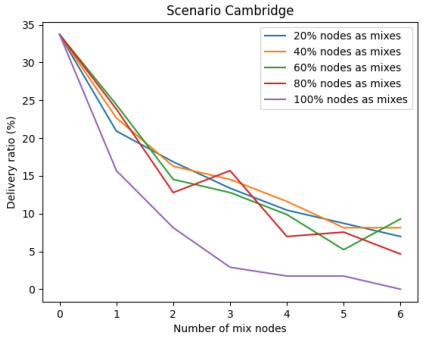
Figure 2 shows the delivery ratio for each scenario. In each case, the figure shows the delivery ratio based on the number of nodes used in cascade to communicate. 0 mix nodes correspond to the case of not using our opportunistic-mixnet approach. Each line corresponds to a different size for the set $\mathcal{M}$ of mix nodes. We used sizes corresponding to 100%, 80%, 60%, 40%, and 20% of the total nodes. This size was randomly selected from the whole set of nodes of the network. Similarly, Figure 3 shows the same data, but the selection of the set of mix nodes was performed using node centrality.

In general, the selection of nodes with higher centrality as mix nodes provided slightly better results in terms of delivery ratio. This improvement was, however, relatively small in most cases.

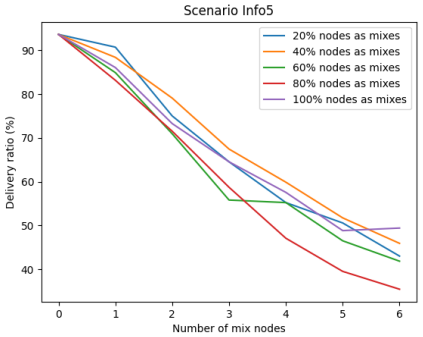
Using a higher number of nodes as mix nodes, that is, a larger size for set $\mathcal{M}$, usually provided lower delivery ratios. This is because increasing the number of mix nodes also increases the delay introduced by each. A source node randomly selects mix nodes from $\mathcal{M}$. With more possible nodes, the distribution of messages for each mix node decreases, and mix nodes need more time to receive their corresponding k_i messages to be able to flush them. Increasing the size of $\mathcal{M}$ increases the privacy provided by the system, but on the other hand introduces a clear observable penalty.

In Figures 4 and 5, we show the latency of delivered messages for each scenario using different sizes for set $\mathcal{M}$. Those sizes are 80%, 60%, 40%, and 20% of the total number of the node for each case. The delay is also shown for a different number of cascaded mix nodes. As per the previous figures, the 0 number of mix nodes is the case of sending messages without the opportunistic mixnet.

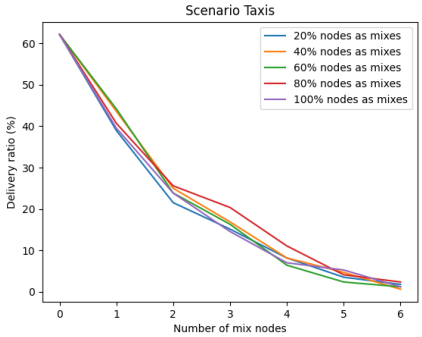
Again, Figure 4 shows the case when the set of mix nodes was chosen randomly, and Figure 5 when those nodes were selected by their centrality.



(**a**) Cambridge.

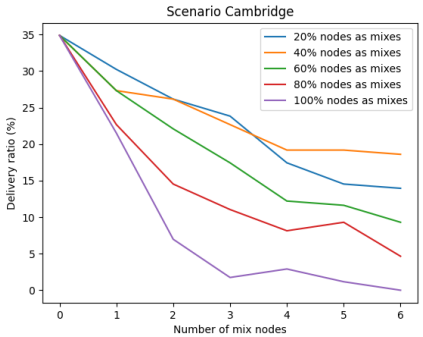


(**b**) Info5.

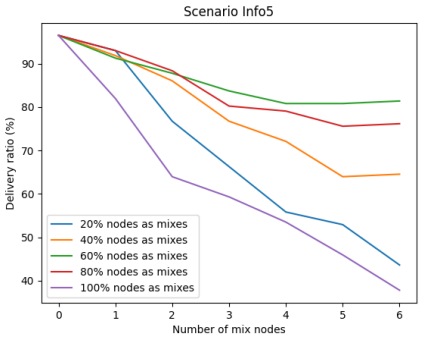


(**c**) Taxis.

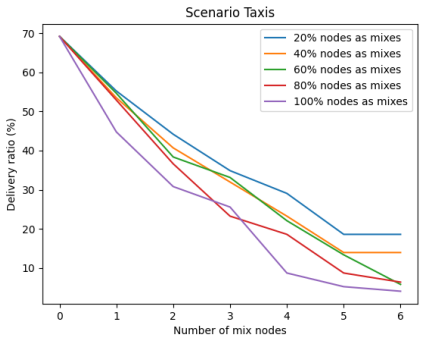
Figure 2. Delivery ration with random mix nodes.



(**a**) Cambridge.



(**b**) Info5.

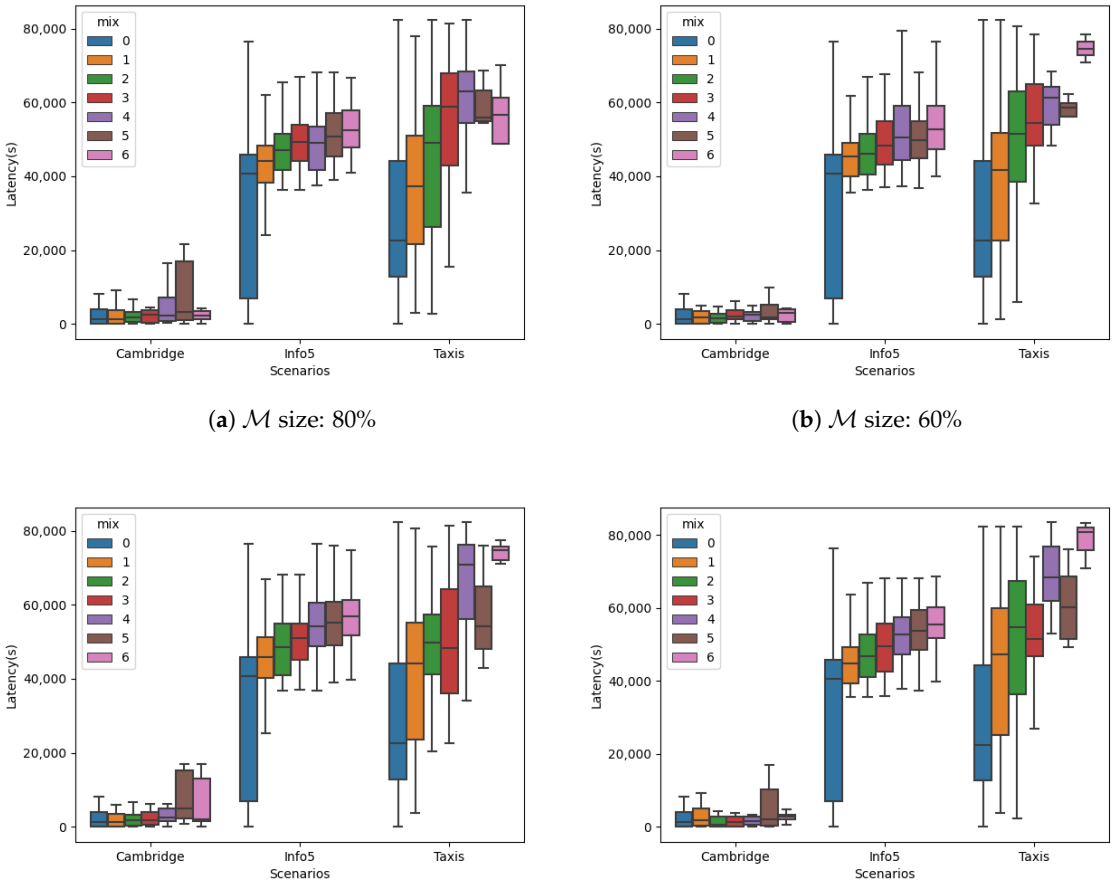


(**c**) Taxis.

Figure 3. Delivery ratio with centrality mix nodes.

The obtained results in this case are analogous to the ones observed for the delivery ratio. When using nodes on the basis of their centrality as mix nodes, we obtained slightly better results. The same happens for smaller sizes of set $\mathcal{M}$.

Overall, the introduced penalty by using the opportunistic-mixnet approach was not very large, and could easily be tolerable in most scenarios and applications of opportunistic networks.



(**a**) $\mathcal{M}$ size: 80%

(**b**) $\mathcal{M}$ size: 60%

(**c**) $\mathcal{M}$ size: 40%

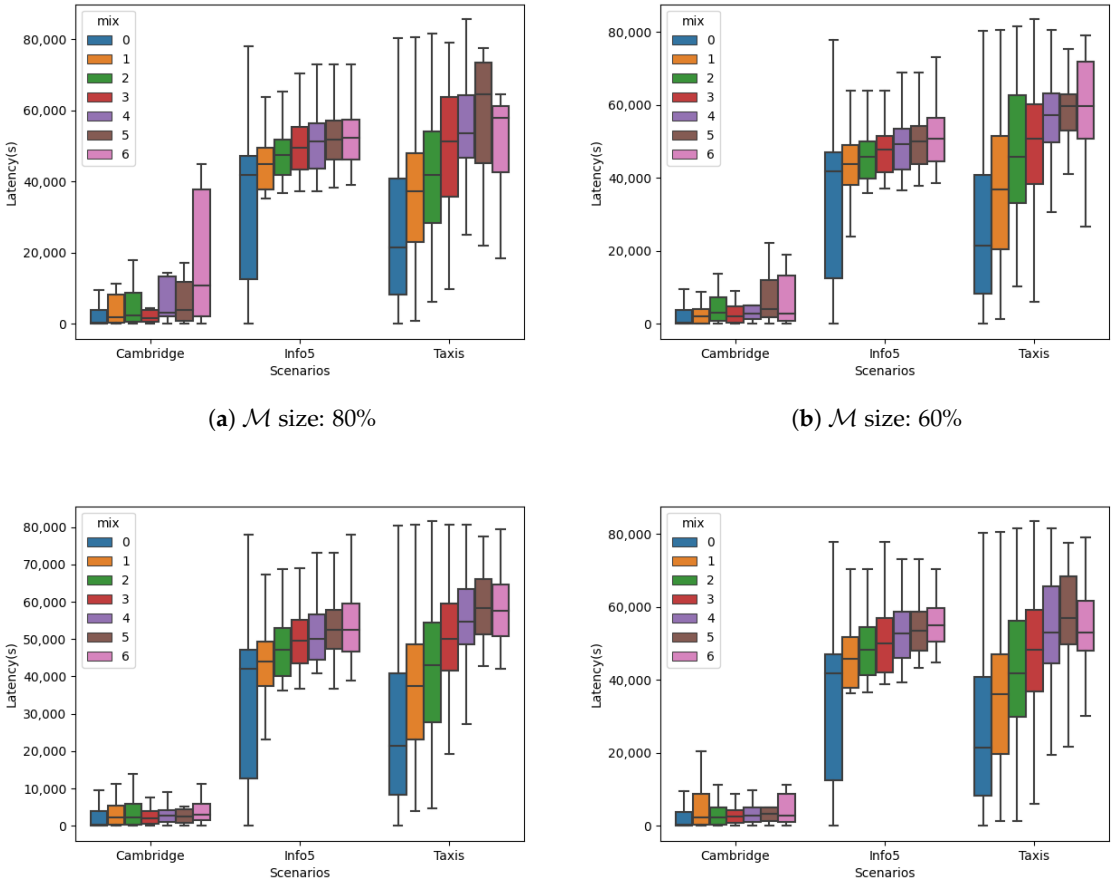
(**d**) $\mathcal{M}$ size: 20%

Figure 4. Latency with random mix nodes.

In Tables 2 and 3, we can see the overhead ratio of our proposed mix networks methods with random and centrality mix nodes. The overhead ratio can be exploited to present the volume of used network resources to deliver a message to its receiver. From the result, we can see that the overhead ratio of the Taxis scenario was much higher than those of the Cambridge and Info5 scenarios. The main reason is that there are more nodes in the Taxis scenario compared with in the two other scenarios, which leads to more relayed messages during transmission. This also shows that the number of mixes had more of an effect on the overhead ratio of the Taxis scenario than those of the Cambridge and Info5 scenarios.

We can also check the average routing-step path length in each case, that is, the number of hops that a message needs to traverse between each mix node. This is shown in Tables 4 and 5 when selecting mix nodes randomly or on the basis of their centrality, respectively.

In general, the path length for each step is kept constant, and the use of the mix nodes does not increase such length. Obviously, as the number of nodes increases, the number of paths from source to destination also increases, e.g., if we have an average step-path length of 4, using one mix node results in path length of 8, and using 3 results in overall path length of 12. This is something already expected by the way an opportunistic mixnet works. We saw, however, that the penalty introduced in terms of delivery ratio or overall path delay is relatively tolerable.



(**a**) $\mathcal{M}$ size: 80% (**b**) $\mathcal{M}$ size: 60%

(**c**) $\mathcal{M}$ size: 40% (**d**) $\mathcal{M}$ size: 20%

Figure 5. Latency with centrality mix nodes.

Table 2. Overhead ratio with random mix nodes.

Size of $\mathcal{M}$	0	1	2	3	4	5	6
Cambridge							
20%	39.7	37.0	36.7	35.9	37.1	36.9	36.5
40%	39.7	36.6	37.4	34.9	35.8	35.3	34.5
60%	39.7	36.6	37.7	35.7	34.5	35.4	35.4
80%	39.7	35.2	37.3	35.1	34.9	34.5	34.8
100%	39.7	41.2	44.1	43.4	42.3	43.4	43.0
Info5							
20%	38.1	37.5	39.1	39.5	39.7	39.3	39.8
40%	38.1	37.4	37.6	38.0	37.7	38.2	37.7
60%	38.1	38.2	38.7	39.2	38.4	38.1	38.0
80%	38.1	39.0	38.6	39.4	39.4	39.3	38.6
100%	38.1	38.4	39.1	38.4	38.3	38.5	38.1
Taxis							
20%	275.4	285.5	292.7	290.9	299.2	306.5	310.1
40%	275.4	271.9	284.3	285.4	289.2	298.8	302.0
60%	275.4	274.5	277.2	274.7	292.0	302.0	296.8
80%	275.4	282.2	287.3	269.2	284.4	279.9	280.0
100%	275.4	284.4	294.5	289.9	290.7	305.4	305.9

Table 3. Overhead ratio with centrality mix nodes.

Size of $\mathcal{M}$	0	1	2	3	4	5	6
Cambridge							
20%	38.5	33.3	33.2	32.5	33.4	34.5	34.1
40%	38.5	36.2	34.3	33.9	34.3	34.4	33.8
60%	38.5	36.2	33.6	35.3	34.6	34.3	34.7
80%	38.5	35.0	34.6	36.1	34.3	33.6	35.1
100%	38.5	37.5	42.4	50.5	39.2	40.2	42.7
Info5							
20%	37.6	37.5	39.3	39.8	40.2	39.7	40.1
40%	37.6	37.7	37.8	38.3	38.2	38.4	37.9
60%	37.6	37.8	37.5	37.5	37.2	36.8	36.6
80%	37.6	37.4	37.4	39.3	37.4	37.3	36.8
100%	37.6	38.4	39.5	39.3	39.0	39.2	39.1
Taxis							
20%	277.8	259.3	246.4	237.9	234.7	236.9	230.2
40%	277.8	263.0	252.1	243.4	243.3	242.2	234.8
60%	277.8	258.2	253.0	242.2	244.0	240.4	244.9
80%	277.8	265.5	261.6	263.7	254.2	263.7	255.7
100%	277.8	287.1	281.5	267.8	284.4	290.1	270.3

Table 4. Average routing-step path length with random mix nodes.

Mix Nodes	20%	40%	60%	80%	100%
Cambridge					
0	3.6	3.6	3.6	3.6	3.6
1	3.5	3.4	3.4	3.7	3.8
2	3.4	3.5	3.5	3.3	3.6
3	3.3	3.5	3.7	3.9	3.5
4	3.2	3.8	3.5	3.5	3.7
5	3.4	3.3	3.4	3.7	3.6
6	3.3	3.5	3.6	3.7	3.7
Info5					
0	3.7	3.7	3.7	3.7	3.7
1	3.9	4.0	4.0	3.8	3.8
2	3.9	3.9	4.0	4.1	3.9
3	3.8	4.0	3.9	3.8	4.0
4	3.7	4.0	3.8	3.7	4.0
5	3.7	3.8	3.8	3.9	3.9
6	3.6	3.9	3.7	3.7	3.8
Taxis					
0	5.6	5.6	5.6	5.6	5.6
1	5.7	5.4	5.5	5.5	5.6
2	5.8	5.5	5.4	5.6	5.7
3	5.4	5.2	5.5	5.5	5.6
4	5.7	5.4	5.7	5.8	5.7
5	5.9	5.8	5.6	5.5	5.7
6	6.0	5.2	5.1	5.9	6.2

Table 5. Average routing-step path length with centrality mix nodes.

Mix Nodes	20%	40%	60%	80%	100%
Cambridge					
0	3.3	3.3	3.3	3.3	3.3
1	3.2	3.7	3.6	3.9	3.7
2	3.2	3.6	3.4	3.6	3.3
3	3.0	3.4	3.6	4.0	4.1
4	2.9	3.7	3.7	3.7	3.4
5	3.1	3.6	3.6	4.0	3.7
6	2.9	3.7	3.6	3.9	4.0
Info5					
0	3.9	3.9	3.9	3.9	3.9
1	3.7	3.7	3.9	3.8	3.8
2	3.8	3.8	3.6	3.9	3.8
3	3.4	3.7	3.6	4.0	4.0
4	3.6	3.8	3.6	3.6	3.9
5	3.5	3.7	3.7	3.8	3.9
6	3.3	3.8	3.7	3.9	3.9
Taxis					
0	6.2	6.2	6.2	6.2	6.2
1	5.6	5.7	5.7	5.9	5.7
2	5.3	5.5	5.7	5.5	5.7
3	5.3	5.3	5.5	5.6	5.3
4	5.4	5.3	5.5	5.6	5.8
5	5.4	5.3	5.5	5.7	5.5
6	6.4	5.4	5.4	5.4	5.3

To analyze the buffer storage of each node, we saw in simulations that the average time in which a node holds messages in mix nodes is less than 1% of total delivery time. This percentage is so small since there is a lot of traffic in the network, and the minimal number of messages to forward a message is guaranteed 99% of the time. Because of this, this buffer utilization did not affect the performance of the latency of delivery time.

4.4. Anonymity Analysis

Our mixnet-based proposal can ensure sender and receiver anonymity, and sender–receiver unlinkability.

Sender anonymity: message M cannot be linked to any sender S. In our mixnet proposal, receiver R cannot trace back sender S, as R can only monitor its previous node. A single mix node only knows the previous and successor node. Even if the attacker compromises a number of mix nodes, it is still impossible to connect the sender and receiver if there is at least one honest mix node.

Receiver anonymity: message M cannot be linked to any receiver R. When message M is forwarded in our OppNet schema, it goes through a number of mix nodes, since there is at least one mix node that cannot be compromised. The attacker cannot obtain any information from message M. Thus, our proposed mixnet method can provide receiver anonymity.

Sender–receiver unlinkability: It is essential to trace the routing path in the OppNet to link sender and receiver. However, this was not the case in our scenarios. For any honest sender S and receiver R, message M could be encrypted and rerandomized with mix nodes.

4.5. Privacy Analysis Compared with Onion-Routing-Based Scheme

In this section, we compare our proposed mix network scheme with the possible use of an onion-routing (OR) scheme. For this purpose, we introduced the concept of attack range that denotes the communication area that the attacker can access. The attacker can gain access to all messages

exchanged in this area. As we previously mentioned, one of the drawbacks of common OppNet scenarios is that it might be relatively easy to monitor the entire communication network. In general, we assumed that, if the attacker can monitor the communication of the entire onion-routing circuit, then the attacker could break the circuit. This means that the attacker could monitor the whole circuit or communication path, violating the privacy of the circuit from sender and receiver nodes.

For example, suppose that a user sends a message from source node V_s to destination node V_d using three onion routers OR_1, OR_2, OR_3, $V_s \rightarrow OR_1 \rightarrow OR_2 \rightarrow OR_3 \rightarrow V_d$. In order to go from one node to the other, the message might follow an epidemic routing path through different network nodes as described in Section 3.2. That is, we have something similar to the scenario described in Figure 1, but using onion routers instead of mix nodes. In this case, if all nodes are inside attack range, then the attacker can trace communication $V_s \rightarrow V_d$. Depending on the size of the attack range, the attacker could trace a given number of communications.

In order to show the case in our OppNet scenarios, we simulated the attack range with the random walk mode in the ONE simulator. The entire simulation movement area was 800×800. The number of ORs is three (i.e., this is the common number of onion router used in Tor). Table 6 indicates the percentage of onion-routing circuits that an attacker could break in terms of the size of the attack range, which is expressed as a percentage of the total geographical area of the network.

From the table, we can see that the bigger the attack range is, the smaller the general privacy is achieved using onion routing. With regard to our mix network scheme, it is impossible to connect all packets with the whole circuit even if the attack range is 100%. This is due to the nature of the mixers, which prevents the correlation of source and destination even if the attacker can monitor the whole network. So, the privacy of our mix network scheme remains the same no matter the range that the attacker can access.

Table 6. Attack range with broken-circuit percentage.

Attack Range (%)	Broken Circuits (%)	Scheme
0	0	OR
0	0	Mix networks
20	16.7	OR
20	0	Mix networks
40	46.7	OR
40	0	Mix networks
60	63.3	OR
60	0	Mix networks
100	100	OR
100	0	Mix networks

5. Conclusions

In this paper, we proposed the use of a mix network to provide anonymous communications in OppNets. Our approach is to rely on a free routing mixnet with opportunistic routing between mix nodes. We showed that the introduced performance penalty by using such a system in common OppNet scenarios is tolerable, making it an interesting solution to consider.

As we pointed out in Section 2, using an onion-routing mechanism cannot ensure anonymous communication well in some situations, as it is often easier to be compromised with entry and exit nodes. Our mixnet-based proposal has interesting advantages as compared to existing approaches that are mainly based on some form of onion routing. One of the main advantages is making the system more secure against traffic analysis. Some typical OppNet scenarios involve wireless communications that can be easily monitored. Even monitoring the whole network is not unrealistic. The use of mix nodes provides more resistance to such attacks at the expense of introducing a larger penalty to the performance of the network, both in terms of message-delivery ratio and delay. Our goal here was to show that such a penalty can be tolerable in most cases.

There are some potential extensions to our presented work. As network traffic in OppNets significantly affects the attacker's judgement of the flow of each message, using dummy traffic can be a useful tool to enhance the difficulty of linking the relationship between sender and receiver. Dummy traffic can be exploited to increase the complexity of message flow and hide end-to-end communication intention [19]. It is also essential to investigate the trade-off between dummy traffic and network performance. Another future work will be the investigation of more complex mix networks. In this paper, we exploited a simple mix network mechanism to provide anonymous communication in OppNets. However, there are more other mix networks methods [20,21] that can be potential techniques to improve anonymous communications in OppNets.

Author Contributions: Conceptualization, D.C. and G.N.-A.; methodology, D.C. and C.B.; software, D.C. and C.B.; validation, D.C., C.B. and G.N.-A; formal analysis, D.C., C.B.; data curation, D.C.; writing—original draft preparation, D.C.; writing—review and editing, G.N.-A, D.C. and C.B.; visualization, D.C.; supervision, G.N.-A and C.B. All authors have read and agreed to the published version of the manuscript.

Funding: This work was partly supported by the Catalan AGAUR 2017SGR-463 project, and the Spanish Ministry of Science and Innovation TIN2017-87211-R project. Depeng Chen acknowledges the support from a China Scholarship Council grant.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Denko, M.K. *Mobile Opportunistic Networks: Architectures, Protocols and Applications*; CRC Press: Boca Raton, FL, USA, 2016.
2. Soelistijanto, B.; Howarth, M.P. Transfer reliability and congestion control strategies in opportunistic networks: A survey. *IEEE Commun. Surv. Tutor.* **2013**, *16*, 538–555. [CrossRef]
3. Magaia, N.; Borrego, C.; Pereira, P.; Correia, M. PRIVO: A privacy-preserving opportunistic routing protocol for delay tolerant networks. In Proceedings of the 2017 IFIP Networking Conference (IFIP Networking) and Workshops, Stockholm, Sweden, 12–15 June 2017; pp. 1–9.
4. Shi, C.; Luo, X.; Traynor, P.; Ammar, M.H.; Zegura, E.W. Arden: Anonymous networking in delay tolerant networks. *Ad Hoc Netw.* **2012**, *10*, 918–930. [CrossRef]
5. Sakai, K.; Sun, M.T.; Ku, W.S.; Wu, J.; Alanazi, F.S. Performance and security analyses of onion-based anonymous routing for delay tolerant networks. *IEEE Trans. Mob. Comput.* **2017**, *16*, 3473–3487. [CrossRef]
6. Chen, D.; Navarro-Arribas, G.; Pérez-Solà, C.; Borrell, J. Message anonymity on predictable opportunistic networks. *J. Ambient. Intell. Humaniz. Comput.* **2019**, 1–14. [CrossRef]
7. Rashidibajgan, S.; Doss, R. Privacy-preserving history-based routing in Opportunistic Networks. *Comput. Secur.* **2019**, *84*, 244–255. [CrossRef]
8. Bakiras, S.; Troja, E.; Xu, X.; Naves, J.F. Secure and Anonymous Communications Over Delay Tolerant Networks. *IEEE Access* **2020**, *8*, 88158–88169. [CrossRef]
9. Chaum, D.L. Untraceable electronic mail, return addresses, and digital pseudonyms. *Commun. ACM* **1981**, *24*, 84–90. [CrossRef]
10. Freedman, M.J.; Morris, R. Tarzan: A peer-to-peer anonymizing network layer. In Proceedings of the 9th ACM Conference on Computer and Communications Security, Washington, DC, USA, 18–22 November 2002; pp. 193–206.
11. Moeller, U.; Cottrell, L.; Palfrader, P.; Sassaman, L. Mixmaster Protocol Version 2. Internet Draft Draft-Sassaman-Mixmaster-03, Internet Engineering Task Force, 2004. Available online: http://tools.ietf.org/html/draft-sassaman-mixmaster-03. (accessed on 15 August 2020).
12. Gulcu, C.; Tsudik, G. Mixing E-mail with Babel. In Proceedings of Internet Society Symposium on Network and Distributed Systems Security, San Diego, CA, USA, 22–23 February 1996; pp. 2–16.
13. Danezis, G. Mix-Networks with Restricted Routes. In *Privacy Enhancing Technologies*; Dingledine, R., Ed.; Springer: Heidelberg, Germany, 2003.
14. Borrego, C.; Amadeo, M.; Molinaro, A.; Jhaveri, R.H. Privacy-Preserving Forwarding Using Homomorphic Encryption for Information-Centric Wireless Ad Hoc Networks. *IEEE Commun. Lett.* **2019**, *23*, 1708–1711. [CrossRef]

15. Keränen, A.; Ott, J.; Kärkkäinen, T. The ONE simulator for DTN protocol evaluation. In Proceedings of the 2nd International Conference on Simulation Tools and Techniques. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), Rome, Italy, 2–6 March 2009; p. 55.
16. Akestoridis, D.G. CRAWDAD Dataset Uoi/Haggle (v. 2016-08-28): Derived from Cambridge/Haggle (v. 2009-05-29). 2016. Available online: http://crawdad.org/uoi/haggle/20160828/one (accessed on 15 August 2020).
17. Leguay, J.; Hui, P.; Crowcroft, J.; Scott, J.; Lindgren, A.; Friedman, T. CRAWDAD Dataset Upmc/Content (v. 2006-11-17). 2006. Available online: http://crawdad.org/upmc/content/20061117 (accessed on 15 August 2020) .
18. Bracciale, L.; Bonola, M.; Loreti, P.; Bianchi, G.; Amici, R.; Rabuffi, A. CRAWDAD Dataset Roma/Taxi (v. 2014-07-17). 2014. Available online: https://crawdad.org/roma/taxi/20140717 (accessed on 15 August 2020).
19. Oya, S.; Troncoso, C.; Pérez-González, F. Do dummies pay off? limits of dummy traffic protection in anonymous communications. In *International Symposium on Privacy Enhancing Technologies Symposium*; Springer: Heidelberg, Germany, 2014; pp. 204–223.
20. Chaum, D.; Javani, F.; Kate, A.; Krasnova, A.; de Ruiter, J.; Sherman, A.T.; Das, D. *cMix: Anonymization by High-Performance Scalable Mixing*; USENIX Security: San Francisco, CA, USA, 2016.
21. Alexopoulos, N.; Kiayias, A.; Talviste, R.; Zacharias, T. MCMix: Anonymous messaging via secure multiparty computation. In Proceedings of the 26th {USENIX} Security Symposium ({USENIX} Security 17), Vancouver, BC, Canada, 16–18 August 2017; pp. 1217–1234.