Anonymous Short Communications over Social Networks

Francesco Buccafurri(✉), Vincenzo De Angelis, Maria Francesca Idone, and Cecilia Labrini

University of Reggio Calabria, Via Graziella, Loc. Feo di Vito, Reggio Calabria, Italy
{bucca,vincenzo.deangelis,mariafrancesca.idone,cecilia.labrini}@unirc.it

Abstract. Social networks nowadays attract a huge number of people and are an extraordinary platform for applications going beyond the basic mission of connecting people. In this context, anonymity of senders and recipients of messages could play an important role to protect privacy in specific cases such as proximity-based services, surveys, crowdfunding, and e-democracy, in which only short communications should be protected. As the social network provider could be semi-trusted (i.e., honest-but-curious), it could play as global passive adversary, monitoring the flow of all the messages exchanged in the social network. In this paper, we propose a new anonymous communication protocol tailored to our application context. We show that our proposal offers a good solution to the trade-off between traffic overhead and communication latency, better than the application of existing anonymous overlay routing protocols.

Keywords: Privacy · Anonymous communication · Social networks

1 Introduction and Motivations

Social networks probably represent the most disrupting digital innovation of the last twenty years. Different kinds of applications are nowadays implemented on top social networks. However, their power could be better exploited in some application contexts, such as e-democracy and e-participation, provided that communication anonymity is supported against a potentially honest-but-curious social network provider. Consider for example the domain of market surveys that a given company wants to submit to a given population for marketing purposes. Implementing this through an existing social network could dramatically increase the effectiveness and efficiency of the campaign, by easily reaching a large population and by allowing more sophisticated sample selections than traditional methods. A similar application context is that of innovative participatory models of crowdfunding, in which a company might leverage existing social networks to enable the anonymous participation of crowdfunders in voting or opinion evaluation related to the governance of the company.

J. Garcia-Alfaro et al. (Eds.): SecureComm 2021, LNICST 399, pp. 43–63, 2021.
https://doi.org/10.1007/978-3-030-90022-9_3

In the cases above, the communication should happen between anonymous users and explicit entities. However, interactions requiring anonymity among users might occur in other cases, such as anonymous opinion exchange [10], or proximity-based services, delivered on the basis of the proximity of two users who detect this condition by mutually exchanging some messages.

An important point is that the above applications are characterized by the common denominator that only a few asynchronous messages have to be exchanged in anonymous form. Therefore, we do not need the full communication power usually aimed in the domain of anonymous communication networks [22], supporting low-latency and connection-oriented communication.

The problem is not trivial. Indeed, the social network provider should be seen, in a realistic threat model, as a global (at least) passive adversary, able to monitor the whole flow of messages among users. Moreover, privacy is achieved if not only the content of messages is protected against the adversary, but also the communication itself.

A few proposals concerning anonymous communication in social networks are available [10,16] (to the best of our knowledge). However, [16] only deals with anonymous group communication (therefore it does not cover the application targets described earlier), and [10] does not provide sender anonymity against the global passive adversary.

The goal of this paper is to achieve communication anonymity against the global passive adversary in social-network applications (like those described earlier) in which only a few asynchronous messages should be exchanged. If we refer to existing centralized social networks, the only way to follow is of course to require the collaboration of social-network users to implement an overlay network over the application layer provided by the social network itself. Indeed, an alternate approach based on a centralized party playing as anonymizer has very limited effectiveness in our threat model, as shown in [24].

Thus, to not to masquerade the core of the problem under the chosen application context (i.e., social networks), we have to refer to any P2P overlay anonymous routing approach resistant to the global passive adversary.

It is well known that, to have anonymity against the global passive adversary we need to include cover traffic (i.e., dummy traffic to hide the actual messages) [6], which, in our context, results in bandwidth and CPU overhead (thus also battery consumption) for social-network users. Therefore, we have to reduce cover traffic as much as possible.

Existing P2P overlay network routing techniques resisting to the global passive adversary are based either on *mixnets* [2,8,11], or on *buses* [1,9,27].

As shown in the paper, state-of-the-art mixnet-based approaches, require a high amount of cover traffic. On the other hand, mixnets support low-latency applications and bursting-traffic services, as well as the notion of connection. As observed earlier, these are not features required in our application setting.

In the approach based on buses, anonymity is achieved by implementing routes (either deterministic [1,9] or non-deterministic [27]) independent of the intended communication, which senders and receivers can opportunistically exploit. With this approach, cover traffic is drastically reduced with respect to mixnets (at the price of higher latency). Indeed, here, no mixing is adopted and the incoming (cover) traffic, for each node, has exactly one 1-hop source, and each node can indifferently play the role of sender, recipient, or relay node. However, both deterministic (in which the fixed route is an Eulerian path passing through all the nodes) and non-deterministic approaches (in which the latency highly increases with the number of nodes) are unrealistic in scenarios with a huge number of nodes like social networks.

In this paper, we propose a new approach, tailored to the chosen application context (i.e., social networks), that uses the concept of fixed deterministic routes of buses to minimize cover traffic. Differently from buses, deterministic routes are not used to hide inside both senders and receivers, but to obtain separately anonymity of senders and recipients, thus allowing small predetermined cyclic routes (one route in which the sender is anonymized and one route in which the recipient is anonymized) and, then, acceptable latency.

The proposed technique is analytically compared with buses and mixnets, showing that buses produce much higher latency and mixnets require much more cover traffic. Moreover, through experiments, our technique is tested to be feasible in terms of latency, for the chosen applications. The proposed protocol, besides anonymity, also includes a certain degree of redundancy to take into account the possibility that some users are not collaborating.

The structure of the paper is the following. The related literature is analyzed in Sect. 2. We provide some basic notions needed to understand the proposed protocol in Sect. 3. The protocol is presented in Sect. 4 and it is compared with the state-of-the-art approaches in Sect. 5. In Sect. 6, we report the results of an experimental validation of our approach. The security of the protocol is analyzed in Sect. 7. Finally, in Sect. 8, we draw our conclusions.

2 Related Work

As mentioned in the introduction, our paper can be related to anonymous P2P overlay networks, in which participants act as both end users and relay nodes that route the data packets of others. A survey on anonymous communication systems is reported in [22]. Anonymous routing protocols existing in the literature might include full cover traffic. The class of techniques requiring cover traffic (also partially partial) suffers from the drawback that they are not resistant against the global passive adversary. Therefore, they do not reach the objective aimed in this paper, in which the social network provider plays as a global passive adversary. The famous Onion [19], Crowds [20], AP3 [15], and Octopus [25], fall within this class of protocols.

To resists against the global passive adversary, the inclusion of cover traffic is needed. As stated in the introduction, two main different approaches are possible: *buses* or *mixnets*. Our paper mainly relates to buses [1,9,27], by introducing a way to modulate anonymity (by passing from full anonymity to k-anonymity) with an advantage in terms of message latency.

There is a huge literature on mixnets, originally proposed in [4]. The combination of mixnets with cover traffic is a very common approach [8,11,12,26]. In general, by mixing cover with data traffic, one can add noise to the underlying traffic patterns to defeat traffic analysis. A typical approach is to insert cover traffic to maintain a constant transmission rate on any link, so that the traffic patterns observed by the adversary stay unchanged and leak no identifying information. [5] is a very recent mixnet-based approach resistant to the global adversary, but it is designed at network layer (thus it is not easy to understand how to implement network *Autonomous Systems* in a P2P fashion). Despite its age, Tarzan [8] still keeps a relevant position in the landscape of P2P anonymous overlay networks, as provides an effective way to implement the concept of mixnet combined with cover traffic even though some aspects (e.g., the adoption of a gossip protocol) might limit its scalability. Due to space limitations, we cannot analyze the huge literature regarding mixnets. However, in Sect. 5, we provide an analytical comparison of mixnet-based approaches with our proposal, based on a simplified model of mixnets extracted from [8].

Another important class of approaches is based on DC-Nets [22], which uses secure multi-party cryptographic protocols among the participant in the communication. One the most recent proposals falling into this class is [16]. It proposes an approach based on DC-net that allows users to anonymously publish their messages in a group of users. However, it does not provide point-to-point anonymous communication between two users. Moreover, users cannot communicate externally to the group. In general, this class of approaches is not applicable in our context. Indeed, they expect all participants to be involved in every run of the protocol and require pairwise shared keys between the participants. This is impractical in the social network domain. Moreover, as a matter of fact, they are more effective for anonymous group communication.

3 Background

In this section, we provide some basic notions useful for the comprehension of the protocol. They are *Identity-based Encryption (IBE)* and *Anonymity.*

Identity-Based Encryption. Identity-based encryption (IBE) is a type of public-key encryption in which the public key of a user is represented by some unique information associated with the user's identity. In our setting, the public key is the social network ID of the user. The advantage of IBE is that each user may encrypt a message for another user without requiring the public key to any external party. Formally, an IBE scheme consists of four algorithms:

Setup(k): it takes as input a security parameter k and outputs a master secret key MSK and master public key MPK.

Extract(MPK, MSK, ID): it takes as input the master public key MPK, the master secret key MSK and a parameter ID representing the identity of a user. It outputs a private key d associated with the user's identity ID.

Encrypt(MPK, ID, M): it takes as input the master public key MPK, a parameter ID representing the identity of a user and a message M. It outputs a ciphertext C intended for the user with identity ID.

Decrypt(MPK, C, d): it takes as input the master public key MPK, a ciphertext C, and a private key d. It outputs the decryption C of a message M.

IBE requires the presence of a trusted third party called Private Key Generator (PKG). In a setup phase, PKG invokes Setup (k) to obtain MPK and MSK. MPK is provided to all the users and MSK is kept secret by PKG. When a user u with identity ID_u wants to obtain a secret key associated with such an identity ID_u, she/he contacts PKG that invokes Extract (MPK, MSK, ID_u) to obtain d_u and sends it to u. Obviously, PKG sends d_u after making sure about the identity of u, for example through the help of an Identity Provider [18]. If another user y wants to send a message M to u (whose identity ID_u is known to y), then y invokes Encrypt (MPK, ID_u, M) to obtain the ciphertext C and sends C to u. Finally, u invokes Decrypt (MPK, C, d_u) and retrieves the message M. Observe that y does not interact with any party during the encryption process and, as will be clear in Sect. 4, this goes in favour of anonymity. Furthermore, we require that the adopted IBE scheme used is probabilistic and anonymous. Probabilistic means that the same message encrypted more times results in different ciphertexts. Anonymous means that the identity of the recipient of a ciphertext is hidden. The schemes [3,7], for example, satisfy our requirements.

Anonymity. The notion of *anonymity* may have different meanings, depending on context. Anyway, a comprehensive characterization of this concept is given in [17], which this section is inspired to. To enable anonymity of a given item, we have always to refer to an appropriate set of items with, potentially, the same attributes. This leads to the following general definition: "Anonymity of a subject means that the subject is not identifiable within a set of subjects, the anonymity set." The content of the anonymity set depends on the kind of item we are considering. For example, if we want to protect the privacy of who is performing a certain action, the anonymity set is composed of the subjects that are possible actors. In the context of message communication, we have the following specific definitions.

Sender Anonymity: the condition in which the adversary cannot *sufficiently* identify the sender in a set of potential senders.

Recipient Anonymity: the condition in which the adversary cannot *sufficiently* identify the recipient in a set of potential recipients.

Relationship Anonymity: the condition in which the adversary cannot *sufficiently* identify that a sender (in a set of potential senders) and a recipient (in a set of potential recipients) are communicating.

Observe that in the above definitions, with the use of the term sufficiently, we mean that "there is a possibility to quantify anonymity" with a proper threshold.

In our paper, the notion of *communication k-anonymity* defined in [23] is adopted. k-anonymity guarantees that in a network with n honest participants, the adversary is notable to guess the sender or recipient of a particular message with probability non-negligibly greater than $1/k$, where k is a configurable constant possibly smaller than n.

4 The Anonymity Protocol

We denote by SN the social network provider (we refer to a classical centralized social network model). Every user is associated with a numeric *SN identity* SI which is the ID used by SN to address the user (e.g., the ID_user of Facebook).

An anonymous IBE (identity-based encryption) system exists, relying on a PKG independent of SN and referring to the domain of SN identities. Obviously, the price in terms of confidentiality threat that the adoption of IBE in general entails, if the PKG misbehaves, is mitigated by the fact that the encrypted messages are privately exchanged within the social network, thus they are not publicly accessible. In the setup phase, the user demonstrates to the PKG the ownership of a given SI through the SN profile to obtain the private key.

Anonymity is obtained through a cooperative approach, involving the users of SN. The collaboration is thought as a special feature of certain application domains, each forming a *collaboration community*. The users of each community require anonymity when a certain type of service is delivered. In a real-life setting, we can imagine that the presence of a user in an application domain, besides the voluntary adhesion, could be conditioned to other features, such as the verification of the profile (to exclude fake profiles). Consider, for example, an application domain devoted to the submission of anonymous surveys. Besides anonymity (and the other typical requirements of e-voting systems), Sybil attacks (based on the proliferation of dummy identities) must be prevented.

An important feature required for the domains is that, if for the specific application the geographical location (e.g., the IP zone) can be a quasi-identifier with respect to the action that we want to anonymize, then the domain has to be restricted to the geographical area. For example, for national surveys not requiring more specific geographical information, the domain can include only profiles belonging to the national territory. In general, depending on the application, privacy guarantees taking into account also l-diversity [14] and t-closeness [13] should be provided, possibly by a suitable TTP. For space reasons, it was not possible to address this aspect in this paper, even though classical approaches can be applied.

4.1 The Ring Schema

In this section, we describe the structural elements of the model which the solution relies on. We start with the definition of *application domain*.

Definition 1. *An application domain A is a tuple $\langle ID_A, N_A, k_A, r_A \rangle$, where $ID_A \subseteq \mathbb{N}$ is a finite set of the SN identities of the involved users, N_A represents the cardinality of ID_A, $k_A \in \mathbb{N}^+$ is said privacy level, and $r_A \in \mathbb{N}^+ \setminus \{1\}$ is said redundancy level.*

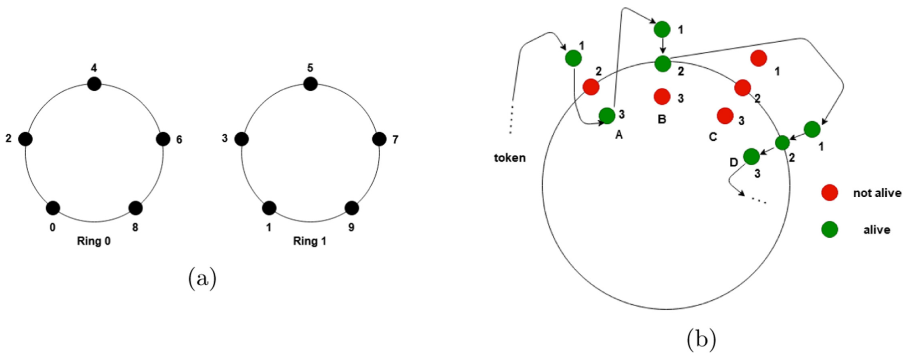


Fig. 1. (a) Ring schema for $\alpha = 2$ and $k_A = 5$; (b) Example of fragment of the route of a token in a ring.

The meaning of the privacy level regards the objective of our protocol, which is anonymity. Anonymity regards sender and recipient of a message (thus relationship) and is reached by requiring a sufficient degree of uncertainty. Given an application domain A, the privacy level k_A represents just the obtained degree of uncertainty, in the sense that the adversary can identify an item (sender or recipient) with probability not greater than $\frac{1}{k_A}$.

As it will be clearer below, the redundancy level indicates the average number of users mapped to a node of a virtual circuit we build for message communication. Virtual circuits are circular circuits, called *rings*. These are the elements of the *ring schema*, defined as follows.

Definition 2. *Given an application domain $A = \langle ID_A, N_A, k_A, r_A \rangle$, a number n_A such that $n_A = \alpha \cdot k_A$ (for any $\alpha \in \mathbb{N}^+$) and $\frac{N_A}{n_A} \geq r_A$, the set $D_A = \{0, \dots, n_A - 1\}$, and the function $f_A : D_A \to D_A$ such that $f_A(x) = x + \alpha$ **mod** n_A, the (α-)ring schema (of A) is the set of the equivalence classes induced by f_A each containing all the elements of D_A congruent modulo α. Each class is called (α-)ring (of A). A ring is identified by the canonical representative of the equivalence class. Given an element $x \in D_A$, we denoted by $ring(x)$ (with $ring(x) \in D_A$) the canonical representative of the ring which x belongs to. The elements of a ring are called nodes.*

It is easy to see that the cardinality of an α-*ring schema* is α and that the identifiers of the classes are $0, 1, \dots, \alpha - 1$. Moreover, all the classes are of cardinality k_A. Observe that the ring schema is completely defined by the parameters N_A, k_A, r_A, and α. The parameters of the adopted ring schema are notified to all the users of the application domain.

An example of ring schema for $\alpha = 2$ and $k_A = 5$ (then $n_A = 10$) is reported in Fig. 1a. Therein, being $\alpha = 2$, we have 2 rings (i.e., ring 0 and ring 1) each including $k_A = 5$ nodes. Moreover, for example, $ring(2) = 0$ and $ring(9) = 1$.

From now on, throughout the paper, assume given an application domain $A = \langle ID_A, N_A, k_A, r_A \rangle$ and an α-ring schema, for a given α.

The ring schema is the basic notion of our solution, because allows us to identify a topological structure suitable to support a cover-message-based mechanism which *hides* senders and recipients through mutual collaboration of the users of the application domain. To do this, the so far abstract nodes of rings have to be associated with users of the domain. This is done by uniformly mapping (through a classical hash function h) the set of SN identities of a domain ID_A to the set of nodes in D_A.

Definition 3. *A user mapping on an α-ring schema of A is any function h : $ID_A \rightarrow D_A$ such that the probability that $h(x) = h(y)$ (for each $x, y \in ID_A$, such that $x \neq y$) follows the uniform distribution.*

From now on, we consider, as a user mapping, the hash function h such that $h(x) = x$ **mod** n_A, for each SI $x \in ID_A$. It is well-known that this function allows us to fulfil the condition required by Definition 3. However, a different user mapping could be adopted, also by taking into account possible specific characteristics of the set ID_A. Observe that, since $\frac{|ID_A|}{|D_A|} = \frac{N_A}{n_A} \geq r_A$, we obtain an average number of collisions bounded by r_A. Also the user mapping is notified to the users.

The intuition behind the user mapping is to intentionally build (on the basis of the function f_A), virtual cyclic circuits of groups of users. In other words, if the ring v_0 is $\{v_0, \ldots v_{k_A-1}\}$, where $v_i < v_j$ for $0 \leq i < j \leq k_A - 1$, then $f_A(v_i) = v_{i+1}$, for each $0 \leq i < k_A - 1$ and $f_A(v_{k_A-1}) = v_0$. Moreover, with each v_i $(0 \leq i \leq k_A - 1)$, a set of at least r_A users (in the average), identifiable by reversing the function h, is associated. The multiplicity of users associated with nodes has the scope to give redundancy to these virtual cyclic circuits (as we better explain in the next subsection). Observe that, through the user mapping, each user belongs to exactly one node in a ring of a given domain.

The user mapping is not materialized by the users. As we will see later, they just could need to compute some of its values. Instead, SN stores a hash table H (based on h) materializing the user mapping in such a way that if a user requires to know the group of SN identities mapped to a given node v of a ring, SN can efficiently provide the correct answer just by accessing the hash table at the index v. Therefore, a user, starting from the knowledge of a given SI, say x, can determine which is the node associated with this SI just by computing $h(x)$, then can calculate all the sequence of nodes of the ring (for example, the node at distance j from x is obtained as $f_A^j(h(x))$), and can retrieve from SN the list of (at least) r_A SIs (in the average) associated with any node of the ring. We can assume that each user knows always the SIs associated with each node of the ring which the user belongs to. This information is called *configuration of the ring*. Any change of the configuration of the ring is communicated by SN, as we will see in Sect. 4.3. For each SI, say y, thanks to the IBE scheme, x can send an encrypted message to y such that only y can decrypt it without relying on a PKI. In contrast, if SN plays the role of PKI or the PKI is internal, the request of a public key results in a leakage of privacy. Anyway, IBE is not strictly necessary. We chose to adopt IBE because it is the most suitable scheme from a theoretic point of view. If, for real-life implementation of our solution, we want to avoid IBE, an external PKI or even an internal PKI with proper measures to prevent privacy leakage (based on timing decoupling between PKI requests and message communication) might be adopted. This is another subject of interest as a future work.

4.2 Redundancy

In this section, we better discuss the role of the redundancy level. Our protocol is collaborative and requires that users, maybe through a delegated app, perform some actions. Regarding this aspect, we introduce the following. First, we fix a (suitably high) probability threshold τ, and say that an event is *sufficiently* guaranteed if it occurs with probability not below τ.

Liveness Assumption. We assume that, if we pick a user, the probability she/he is alive is p. Then, we set an integer value r such that it is sufficiently guaranteed that among r users at least one is alive. r is any value such that $1-(1-p)^r \geq \tau$. It is worth noting that, as it will be clear in the sequel of the section, our system does not require any kind of registry of the active social network users available to many parties (and, thus, potentially critical from the privacy point of view).

k_A-Anonymity Threshold. For reasons that will be clear later, according to the liveness assumption, to sufficiently guarantee k_A-anonymity, the redundancy level r_A must be set in such a way that the minimum number of users associated with a ring is no smaller than $k_A \cdot r$ because it is sufficiently guaranteed that in a ring at least k_A users are alive. The value $k_A \cdot r$ is called k_A-*anonymity threshold*.

4.3 System Update

The previous definitions do not take into account possible updates regarding an application domain. These are joins and leaves of users, managed as follows.

User Join. When a new user U joins the application domain A, once an SI is assigned to this user, say SI_U, it suffices for SN to include the new user in the hash table H at the index $h(SI_U)$. As this addition cannot threaten the number of expected users alive in the affected ring (i.e., the ring which the node $h(SI_U)$ belongs to), the join does not impact the ring schema. The users of the affected ring are notified about the changes occurred in the ring, in such a way that the local information about the configuration of the ring is made coherent. No further action is required.

User Leave. When a user U with SN identifier SI_U leaves the application domain A, SN has to remove the user from the hash table H, at the index $h(SI_U)$. The number of users is obviously updated as $N'_A = N_A - 1$. Similarly to the case of join, the users of the affected ring are notified about the changes occurred in the ring, in such a way the local information about the configuration of the ring is made coherent. However, the event might impact the ring schema if the minimum number of the users of the affected ring goes below the k_A-anonymity threshold (i.e., $k_A \cdot r$) that sufficiently guarantees k_A-anonymity, according to the liveness assumption. If this is the case, n_A has to be decreased. Therefore, SN finds $\alpha' < \alpha$ for which the above threshold is exceeded. Then, SN has to redistribute the users in the new hash table (of size $n'_A = \alpha' \cdot k_A$), with a computational cost of $O(N'_A)$. SN has to notify to all the users of the domain the updated parameters of the ring schema (i.e., N'_A and α') and the new ring configurations. No computation overhead is required user-side.

Even though the worst case for leaves triggers a server-side (albeit linear, in the number of users) computational overhead, we can argue that, in real-life cases, communities tend to grow, or, at worst, joins and leaves balance. Therefore, with proper "safety margins" applied to the set value of α, the above worst case happens very rarely. Observe that, also the growth of the number of users might trigger the resizing of the hash table even though this is not necessary for the correctness of the ring schema. Indeed, it could be opportune not to have rings with an actual privacy level much higher than the required value. Also these aspects will be deepened as a future work, but do not seem to represent an actual drawback of the solution.

Finally, concerning system updates, one could think that drastic changes of the system not corresponding to changes of the communication characteristics can enable classical intersection attacks, thus breaking anonymity. However, this is not the case, because we only consider short communications, whose lifetime is certainly much less than the lifetime of the substantial ring structure.

4.4 Cover-Message Mechanism

At this point, we introduce the cover-message mechanism mentioned earlier which is at the basis of the anonymity service provided by our solution.

Consider a ring $\{v_0, \ldots v_{k_A-1}\}$. With a certain rationale, we choose r users belonging to the nodes of the rings responsible for maintaining the circulation of dummy messages called *tokens*. r is the value that determines the k_A-anonymity threshold (i.e., $k_A \cdot r$). Therefore, it is sufficiently guaranteed that at least one responsible user is alive. Now, we define how the token is built. It is a fixed-length message with three fields: $\langle C_M, C_D, B\rangle$, where C_M is a ciphertext (including either a dummy message or the intended message), and C_D is a ciphertext too, (including either SN or a destination user/node). B is a bit indicating if the token is empty ($B = 0$) or filled ($B = 1$). The exact meaning of the above fields will be clarified with the description of the communication primitives below. Every token turns in the ring in which it was born hop by hop by crossing, for each node, any alive user associated with this node. This is done according to the increasing value of the corresponding SIs. To formally describe the above mechanism, we need the following definition, introducing the notion of *next* alive user for a given user in a ring.

Definition 4. *Given a node v of a ring with at least an alive user, we denote by $first(v)$ the lowest SI associated with v and by $last(v)$ the highest SI associated with v. The closeness between two alive users with SIs x and y belonging to a ring (denoted by $closeness_A(x, y)$), is recursively defined as follows:*

- $closeness_A(x, y) = 0$, *if* $x = y$;
- $closeness_A(x, y) = |\{z \in ID_A \mid z \neq x \text{ is alive}, h(z) = h(x), x \leq z \leq y\}|$, *if* $h(x) = h(y)$ *and* $x < y$;
- $closeness_A(x, y) = closeness_A(x, last\,(h(x))) + closeness_A\,(first(h(y)), y) + j \cdot (N_A - 1)$ *for the least* $j > 0$ *such that* $f_A^j(h(x)) = h(y)$, *otherwise.*

We define the function $next_A : ID_A \rightarrow ID_A$ *as follows. For any* x *alive,* $next_A(x)$ *is the user (alive) with SI* $y \neq x$ *of the ring such that the closeness between* x *and* y *is minimum.*

In words, the next user of a user with SI x is the first alive user that is encountered by moving first in the node of the ring $h(x)$ in the direction of increasing SIs, and then (if there is no alive user in $h(x)$ with SI higher than x) to the closest node according to the function f_A with alive users and, therein, by taking the user with lowest SI.

According to Definition 4, the token is sent by an alive user x that received it to the user $next_A(x)$, and proceeds in the ring with the same rule. Therefore, it is not said that the token moves from the node v_i to the next node $f_A(v_i)$, because a jump is possible (in the case no alive user is present in the node $f_A(v_i)$).

At each hop, the token is encrypted with IBE under the identity of the next node so that an external eavesdropper cannot distinguish an empty token from a filled token. For efficiency reasons, the token is encrypted with a symmetric on-the-fly key which is, in turn, encrypted with IBE and sent along with the token. When a node receives the token, first it decrypts (by using IBE) the symmetric key and then the token. For the sake of presentation, when we refer to IBE encryption, we mean the above procedure.

In Fig. 1b, an example of fragment of a route followed by a token is depicted. In the figure, we highlight only 4 nodes of a ring, each composed of 3 users (in general, this number could vary among nodes). Green circles represent alive users, while red circles denote non-alive users. The token turns in the ring according to the function *next* of Definition 4.

SN publishes a cross-domain random $R \in \mathbb{N}^+$, with a certain rounding protocol obtained as PRNG verifiable by the users. R, implicitly identifies a user per ring, called *bridge user*, as follows. To identify the bridge user of her/his ring, a user with SI x has to find the alive user with the lowest SI belonging to the node $f_A^j(y)$, such that $j \geq 0$ is the least value such that at least an alive user is in $f_A^j(y)$ and $y = ring(h(x)) + R \cdot \alpha \,\mathbf{mod}\, n_A$. Observe that all the nodes of a ring identify the same bridge user. The bridge node is responsible for sending the messages outside the ring (playing the role of *exit user*) or to inject into the ring the messages coming from outside (playing the role of *entry user*).

4.5 Communication Primitives

In this section, we define the communication primitives supporting anonymity against the adversary (as we will see in Sect. 7, it is SN) and utilizable to build privacy-preserving applications. We have three primitives, defined as follows.

P1: Anonymous Sending to Explicit Recipient. It consists in a communication of a message M starting from an SN user U towards a given destination D (possibly SN) requiring that the sender U is anonymous and the recipient is explicit.

U belongs to the node $h(U)$ of a ring. U knows the current bridge user X. In this case, X plays the role of exit user. Then, U waits for the earliest empty token of the ring (recall that any user, when receives a token, to decide if forwarding or processing it, has to decrypt it because the token is encrypted with its identity) and fills it by setting its fields $\langle C_M, C_D, B\rangle$ as follows: C_M=Encrypt(MPK, D, M) (i.e., the IBE encryption with the identity D of the message M), C_D=Encrypt(MPK, X, D) (i.e., the IBE encryption with the identity X of the destination D), $B = 1$ (that represents the fact that the token is filled). Recall that MPK is the master public key of the IBE scheme. We denote by T the so obtained token. Now, U encrypts the filled token T with the identity of the user $Z = next_A(U)$. Then the token is sent to Z. The token turns in the ring until the node X which is the only user able to decrypt C_D, thus obtaining D. If D is a node, then X identifies (with the collaboration of SN to know the alive SIs) the bridge user Y of the ring of D and sends C_M to Y. Otherwise (i.e., D is either a user or SN), X sends C_M to D. Then, it sets B to 0 and forwards the token in the ring to $next_A(X)$.

P2: Anonymous Response by Explicit Recipient. It consists in a communication from an explicit entity belonging to SN (possibly SN itself) to a user U in response to an anonymous sending initiated by U. This primitive requires that the recipient U remains anonymous.

Since this primitive works in response to an anonymous sending, we can refer to the bridge user from which such a communication directly arrived, say X. The explicit entity starting this primitive just has to send the message to X. This response is actually directed to a user of the ring which X belongs and could be encrypted with an on-the-fly key contained in the original message. Anyway, X injects the response in the ring just by waiting for the earliest empty token and filling it with the response. In this case, C_M contains the response, $B = 1$, and C_D remains undefined. Then, the filled token turns in the ring until a user U is able to decrypt C_M. This is the actual recipient of the response. It does not empty the token and just forwards it. This operation is done by X when the token reaches it again by setting B to 0 and by further forwarding it in the ring.

P3: Anonymous Sending to Anonymous Recipient. It consists in a communication of a message M starting from an SN user with SI U towards a given destination user with SI D requiring that both U and D are anonymous.

The primitive invokes first Primitive **P1** with message M'= Encrypt (MPK, D, M), directed as explicit recipient to $Z = f_A^j\ (h(D))$ for some random j between 0 and $k_A - 1$. Observe that U can encrypt to D without prior asking SN for any information thanks to IBE. Then, when the bridge user of the ring of Z receives this message, it injects the message in the ring as in Primitive **P2**, thus eventually reaching the actual destination D. Observe that a possible reply of D to the message M sent by U can be done by using the same primitive.

5 Comparison with the Existing Approaches

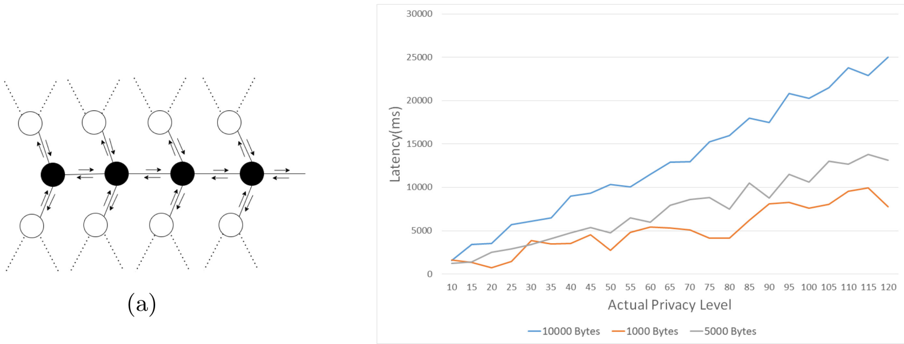
As mentioned in the introduction, there are two possible approaches in the literature that can be used to obtain communication anonymity against the global passive adversary: buses and mixnets.

Buses obtain anonymity by implementing routes (either deterministically [1, 9] or non-deterministically [27]) independent of the intended communication, which senders and receivers can opportunistically exploit.

In deterministic buses [1,9], the route is an Eulerian path, thus a path involving all the nodes of the network. This leads to huge latency times in case of social networks, as the number of nodes of an application domain is potentially huge. It is worth noting that, for a given application domain, for those cases in which primitive **P3** (i.e., anonymous sender to anonymous receiver) is enabled, deterministic buses require to include in the route all possible senders and receivers, therefore all the users of the domain. Consider that, given an application domain $\langle ID_A, N_A, k_A, r_A \rangle$, for our method, the latency time is $\Omega(k_A)$, while for deterministic buses it is $\Omega(N_A)$. In real-life applications, we expect that $N_A \gg k_A$. Coherently with the experiments shown in Sect. 6, any P2P emulation over social networks requires that each anonymous hop between two users is implemented by locally encrypting the message, sending it to the central social network, and then sending it again to the next user. It is easy to see that one hop takes a time of magnitude of 10^{-1} seconds. Therefore, for a realistic domain of 10^4 users, the latency time for a given message communication is 10^3 seconds, which is not suitable for the considered applications (e.g., proximity testing). However, we obtain a very high privacy level, that is 10^4 (in the sense that the adversary identifies the sender or the recipient with probability 10^{-4}. In contrast, with our technique, we can modulate the privacy protection to the still reasonable value of 10^3 to obtain a communication latency of magnitude 10^2 seconds. For a privacy level of order 10^2, we have latency times of magnitude 10^1 (which corresponds to the results obtained in Sect. 6).

Consider now non-deterministic buses [27]. This technique leads to very high latency times, as analytically highlighted in the paper itself. Indeed, the delivery time follows an equation of the form $\frac{K_1}{\left(1-\left(\frac{n-2}{n-1}\right)^{K_2}\right)}$, where K_1 and K_2 are suitable constants and n is the number of nodes of the network, meaning that, for large values of n, we have huge latency time. Indeed, the simulation conducted in [27], which does not take into account the emulation over social networks, in a network with only 2048 nodes (and thus a maximum privacy level of magnitude 10^3) produces average latency time of 20 min. If we overlay the protocol over the social network layer, then this time (roughly) doubles because, as observed earlier a 1-hop P2P message exchange corresponds to 2 point-to-point message exchange (i.e., from sender to server and from server to recipient).

Consider now mixnets. For mixnets, we adopt a simplified yet general model extracted from [8], in which bi-directional cover traffic over any link of the overlay network is enabled (this is necessary to hide communications against the global passive adversary). The idea is to obtain the anonymity set by mixing the traffic



(b)

Fig. 2. (a) Mixnet with $n = 4$ and $m = 2$; (b) Latency of Onion-ring vs actual privacy level.

at each hop of the communication and by hiding the real traffic inside cover traffic. This fan-out mechanism allows us to obtain that the cardinality of the anonymity set increases exponentially with the length of the communication path. In Fig. 2a, we represent a simple mixnet with a degree mixing 2 (i.e., the messages of 2 senders are mixed into a receiver at each step). This way, for a communication path of length l, the anonymity set resulting from the knowledge of a given receiver, has cardinality 2^l. However, to obtain this level of uncertainty, as clearly stated in [8], bi-directional cover traffic should be injected, over all the links of the network. For simplicity, we assume that cover traffic is injected at constant rate so that the amount of traffic can be represented just by the number of links in which it is injected. Let denote by m the mixing degree and by n the number of nodes participating in the mixnet. Note that, to achieve the exponential grow of anonymity degree with the number of hops mentioned before, we require that the mixing involves always new nodes, as depicted in Fig. 2a.

Hence, the number of links allowing us to protect the communication among n nodes is $(m + 1) \cdot n$ as the degree of each node is $m + 1$ and the last node (for example, in Fig. 2a, the total number of links is $12 = 4 \cdot 3$, where $n = 4$ and $m = 2$ and the degree of each node is 3). As cover traffic is bi-directional, the estimation of the total amount of cover traffic is $2 \cdot (m + 1) \cdot n$. This means that the minimum required cover traffic is $6 \cdot n$, as, to enable the fan-out mechanism, $m \geq 2$ should hold.

Consider now our technique. Suppose we have n alive nodes. They are partitioned, in average, among $\frac{n}{k}$ rings each of size k. In other words, k represents the average number of alive users in each ring. Accordingly to the Liveness Assumption, $k \geq k_A$. We refer to k as the *actual privacy level*.

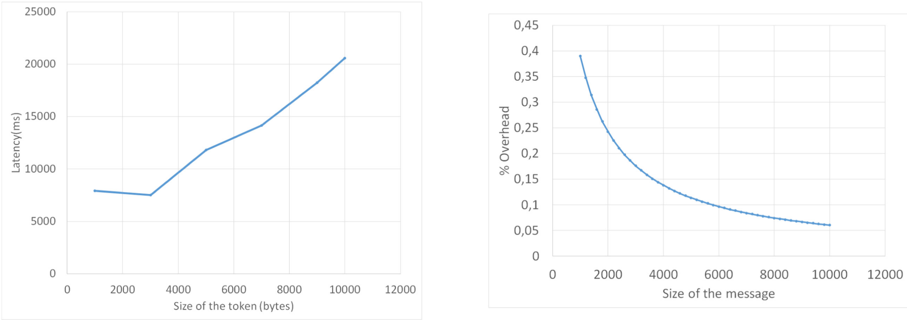
In our approach, cover traffic corresponds just to tokens 1-directionally turning in the rings. Therefore, the number of links is just n which also represents the total amount of cover traffic. Moreover, at least (when m is fixed to the minimum value) we reduce cover traffic of a multiplicative factor equal to 6.

Consider that, about communication latency, we can say that, to achieve the same actual privacy level k in the mixnet, we have to set $l = log_m k$. So, the mixnet technique used should adopt fixed-length tunnels like in [8,21]. Therefore, for equal actual privacy level k, we have a communication path of length k in our case and length $log_m k$ if the mixnet model is adopted. Thus, the advantage we obtain in terms of reduction of cover traffic results in a price in terms of communication latency. However, as mentioned before, this is not critical for our application domains as well highlighted in Sect. 6, in which communication times are evaluated.

6 Experiments

Through this section, we conduct an experimental validation of our approach regarding communication latency, by using the popular social network Twitter. In our analysis, we measured the time elapsed between the sending of message by a user and the reception by another user. To do this, we developed a JAVA application that receives the tokens from the users, decrypts, re-encrypts, and sends them to another user. Therefore, our measurements take into account all the aspects involved in our protocol (including IBE and Symmetric encryption). We implemented the exchange of tokens between users through the exchange of *Direct Messages* on Twitter. Each user receives a Direct Message that contains the token (encoded in Base64), decrypts, and re-encrypts it. Then, the token is sent (encoded in Base64) to another user. To send Direct Messages, we used the library Twitter4j. We applied the IBE scheme [7] and AES-128 for symmetric encryption. To automatically trigger the decrypt/re-encrypt/sending process, we rely on the Twitter Account Activity APIs. These webhook-based APIs send a notification message, each time an event occurs, to a web application we have developed. In our setting, the event is the reception of a Direct Message. To receive webhook events, we need to register a public URL of our web application. As suggested by the Twitter Documentation, to test locally our application, we can use *ngrok* that allows us to create a https tunnel (required by the Twitter Account Activity) and redirects every request to the local port where our web application is running. This inter-mediation would be not necessary in a real-life implementation of our protocol. Therefore, we measured the delay introduced by ngrok and subtract it by the total time. The results of our experiment are reported in Figs. 2b, and 3a.

In Fig. 2b the communication latency vs the actual privacy level is reported. The relationship is almost linear and this confirms the results of Sect. 5. We show three plots for different sizes of the token. We can see that, for actual privacy level $k = 100$, the delays are of 8 seconds for tokens of 1000 bytes and of 20 seconds for tokens of 10000 bytes. These communication-latency times appear compliant with the envisaged applications.



(a) (b)

Fig. 3. (a) Latency vs Token Size with actual privacy level equals to 100; (b) Overhead percentage vs size of the message.

The plot in Fig. 3a shows as the latency varies as the size of the token varies with an actual privacy level $k = 100$. After an initial plateau caused by a threshold effect, the grow is nearly linear. Finally, the last analysis we conducted regards the overhead introduced by the encryption schemes adopted. We denote by m the size in bytes of the message to be sent. We recall that the structure of the token is $\langle C_M, C_D, B\rangle$. C_M is obtained first by applying AES128 on m by using a random on-the-fly key of 16 bytes. As well-know, the overhead introduced by AES128 is the padding of the last block, therefore, in the worst case, it is 16 bytes. Finally, the on-the-fly key is encrypted by using the anonymous IBE scheme. To encrypt a 16 bytes key, the adopted IBE scheme produces a ciphertext of 192 bytes. Therefore, the total size of C_M is $m + 16 + 192$ bytes. With a similar reasoning, we can realize that the size of C_D is $d+16+192$, where d is the size of the destination address. We assume $d = 16$ bytes (it is enough to represent 2^{128} addresses). Finally, the token has to be further encrypted for the next node. The total size will be $m + 640$ bytes. In Fig. 3b, we report the overhead percentage vs the size of the message.

7 Security Analysis

In this section, we sketch a security analysis of our solution. We start by defining the threat model **TM**. We introduce the following three assumptions:

A1: Application domains are formed in such a way that the background knowledge does not allow the adversary to have more information than sender/recipient uniform distribution.

A2: The token has a fixed length so that, thanks to the IBE scheme, only on the basis of the content, it is not possible to distinguish an empty from a filled token.

A3: We assume that in each ring there are at least k_A users alive.

Observe that Assumption **A1** is intrinsically coherent with the concept of application domain itself, in the sense that users of the domain are someway homogeneous with respect to the interest in the delivered services.

Note that **A3** is sufficiently guaranteed if the liveness assumption holds and the number of users in any ring is not below the k_A-anonymity threshold (see Sect. 4.2).

Adversary Model. We consider the case of a global passive adversary able to monitor the flow of all the messages exchanged in the social network. Realistically, the adversary can be only SN itself.

According to the adversary model above, in which the users are not possible adversaries, we can observe that the anonymity of the adopted IBE has not a role, since any message encrypted by IBE is further encrypted hop-by-hop, so that the identity of the recipient cannot be accessed by the adversary even in case of a non-anonymous IBE. However, the adoption of an anonymous IBE just avoids the unnecessary leakage of information against the users of the social networks collaborating in a given communication. Concerning the significance of the adversary model, we stress the fact that, in real-life, the most serious threat is not the curiosity of some buddies, but the possible massive surveillance and misuse of personal sensitive information from the side of the social network provider.

Security Properties. We study the security properties *sender anonymity (**SA**)* and *recipient anonymity (**RA**)*, as defined in Sect. 3. Observe that if one of them is satisfied, then also *relationship anonymity* holds. The above security properties are guaranteed if the adversary is able to identify the protected item (sender or recipient) with probability not higher than $\frac{1}{k_A}$.

Theorem 1. *The security properties **SA** and **RA** hold in the threat model **TM**.*

Proof. The only point of the ring from which the adversary can draw some information more than the random guess to identify senders or recipients, is the bridge user, say it X. Indeed, this is the only point of the ring in which the possible transition empty/filled or filled/empty of a token could be in principle related to the observable incoming or outcoming traffic in/from the bridge. Transitions occurring in other points are not identifiable, due to the Assumptions, with probability higher than $\frac{1}{k_A}$. Therefore, we have to consider the following two cases (they are the only cases potentially helpful for the adversary). Either (1) the adversary observes incoming traffic in X (i.e., X could play the role of entry user), or (2) the adversary observes outcoming traffic from X (i.e., X plays the role of exit user). In case (1), two alternatives are possible. Either (1).a is the case in which X actually injects a token in the ring inserting the message coming from outside, or (1).b X empties a token circulating in the ring as a final step of Primitive **P2** or **P3**, and then X cannot process the incoming traffic. In the case (1).a, the adversary can infer that a recipient exists for this token. The only way to draw more information about this recipient is to follow the token and to observe it when it reaches the bridge. At this point, the adversary can detect a transition filled/empty in X, but this does not give any additional information about where the message has been received. Therefore, due to the Assumptions, the adversary cannot identify the recipient with probability higher than $\frac{1}{k_A}$. Consider now case (1).b. The adversary knows that the token injected by X is empty. Two cases may hold. Either (1).b.1 the token, after a turn, reaches X still empty, or (1).b.2 the token, after a turn, reaches X filled. The adversary may detect which of the above cases ((1).b.1 or (1).b.2) holds, just by observing if outcoming traffic arises from the arrival of the token (i.e., X, after the turn, plays the role of exit user). In the positive case, we are in case (1).b.2, otherwise we are in case (1).b.1. In the latter case, neither senders nor recipients have been involved in this turn. In case (1).b.2, a sender filled the token somewhere in the turn. But, due to the Assumptions, the adversary cannot identify the sender with probability higher than $\frac{1}{k_A}$.

In case (2) (i.e., X plays the role of exit user), the adversary can infer that the token injected by X into the ring is empty. Two cases may hold. Either (2).a the token, after a turn, reaches X still empty, or (2).b the token, after a turn, reaches X filled. The adversary may detect which of the above cases ((2).a or (2).b) holds, just by observing if outcoming traffic arises from the arrival of the token (i.e., X, after the turn, plays again the role of exit user). In the positive case, we are in case (2).b, otherwise we are in case (2).a. In the latter case, neither senders nor recipients have been involved in this turn. In case (2).b, a sender filled the token somewhere in the turn. But, due to the Assumptions, the adversary cannot identify the sender with probability higher than $\frac{1}{k_A}$. The proof is then concluded.

8 Discussion and Conclusions

The main conclusion we can draw from this paper is that when considering the problem of anonymous communication in the social network setting in a specific yet meaningful application domain (i.e., short-communication applications), it is possible to obtain a better result than simply miming any existing anonymous P2P overlay routing protocol. To avoid possible misunderstanding, it is worth noting that the choice of the social-network setting is not opportunistic or orthogonal to our study. It is done because social networks collect all the conditions that are the basis for our protocol (therefore, our protocol can be applied to any setting with similar features). Specifically:

(1) Social networks can be the domain in which the considered applications requiring short-communication anonymity are implemented. It is not evident that this kind of applications might occur in other contexts as they require that users are pre-registered in the system and that fake identities are prevented. Indeed, the social network provider or the party providing the service (if different) can require a secure initial digital identity verification. In other words, the membership of users to a social network could be an effective way to control the subscription to an application domain to avoid fake profiles and Sybil attacks.

(2) The global-passive-adversary threat model is realistic because of the power of the social network provider in existing centralized social networks; this is not the case for generic networks or P2P systems.
(3) Our protocol requires some central management functions, which can be executed by the social network provider. A realistic business model could exist because the users participate in the protocol to obtain privacy and the social network could become a provider of *anonymity as a service* with privacy guarantees and other correlated features, thus attracting larger market segments. Note that the participation of a user in the system has a price in terms of bandwidth and CPU. But this happens also in every P2P or collaborative approach, thus it is not per se a drawback of our solution.

As a future work, we plan to better address the construction of application domains. This is indeed a non-trivial task that we could not treat together with the definition of the anonymous communication protocol (which is the aim of this paper). Indeed, a deep analysis of this aspect, requires the exact definition of the considered use cases, and for each of them, the strategy we have to follow to guarantee that Assumption **A1** stated in Sect. 7 is verified. Another direction of our future research is the full implementation of the system in a "homemade" social network to give a proof-of-concept.

Acknowledgement. This paper is partially supported by Project POR FESR/FSE 14/20 Line A (Action 10.5.6) and Line B (Action 10.5.12).

References

1. Beimel, A., Dolev, S.: Buses for anonymous message delivery. J. Cryptology **16**(1), 25–39 (2003)
2. Bennett, K., Grothoff, C.: GAP – practical anonymous networking. In: Dingledine, R. (ed.) PET 2003. LNCS, vol. 2760, pp. 141–160. Springer, Heidelberg (2003). https://doi.org/10.1007/978-3-540-40956-4_10
3. Brakerski, Z., Lombardi, A., Segev, G., Vaikuntanathan, V.: Anonymous IBE, leakage resilience and circular security from new assumptions. In: Nielsen, J.B., Rijmen, V. (eds.) EUROCRYPT 2018. LNCS, vol. 10820, pp. 535–564. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-78381-9_20
4. Chaum, D.L.: Untraceable electronic mail, return addresses, and digital pseudonyms. Commun. ACM **24**(2), 84–90 (1981)
5. Chen, C., Asoni, D.E., Perrig, A., Barrera, D., Danezis, G., Troncoso, C.: Taranet: traffic-analysis resistant anonymity at the network layer. In: 2018 IEEE European Symposium on Security and Privacy (EuroS&P), pp. 137–152. IEEE (2018)
6. Danezis, G., Diaz, C.: A survey of anonymous communication channels. Tech. rep., Technical Report MSR-TR-2008-35, Microsoft Research (2008)
7. De Caro, A., Iovino, V., Persiano, G.: Fully secure anonymous HIBE and secret-key anonymous IBE with short ciphertexts. In: Joye, M., Miyaji, A., Otsuka, A. (eds.) Pairing 2010. LNCS, vol. 6487, pp. 347–366. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-17455-1_22

8. 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, pp. 193–206 (2002)
9. Hirt, A., Jacobson, M., Williamson, C.: Taxis: scalable strong anonymous communication. In: 2008 IEEE International Symposium on Modeling, Analysis and Simulation of Computers and Telecommunication Systems, pp. 1–10. IEEE (2008)
10. Kacimi, M., Ortolani, S., Crispo, B.: Anonymous opinion exchange over untrusted social networks. In: Proceedings of the Second ACM EuroSys Workshop on Social Network Systems, pp. 26–32 (2009)
11. Kotzanikolaou, P., Chatzisofroniou, G., Burmester, M.: Broadcast anonymous routing (bar): scalable real-time anonymous communication. Int. J. Inf. Secur. **16**(3), 313–326 (2017)
12. Le Blond, S., Choffnes, D., Zhou, W., Druschel, P., Ballani, H., Francis, P.: Towards efficient traffic-analysis resistant anonymity networks. ACM SIGCOMM Comput. Commun. Rev. **43**(4), 303–314 (2013)
13. Li, N., Li, T., Venkatasubramanian, S.: t-closeness: privacy beyond k-anonymity and l-diversity. In: 2007 IEEE 23rd International Conference on Data Engineering, pp. 106–115. IEEE (2007)
14. Machanavajjhala, A., Kifer, D., Gehrke, J., Venkitasubramaniam, M.: l-diversity: privacy beyond k-anonymity. ACM Trans. Knowl. Discovery Data (TKDD) **1**(1), 3-es (2007)
15. Mislove, A., Oberoi, G., Post, A., Reis, C., Druschel, P., Wallach, D.S.: Ap3: cooperative, decentralized anonymous communication. In: Proceedings of the 11th Workshop on ACM SIGOPS European Workshop, pp. 30-es (2004)
16. Nosouhi, M.R., Yu, S., Sood, K., Grobler, M.: Hsdc-net: secure anonymous messaging in online social networks. In: 2019 18th IEEE International Conference On Trust, Security And Privacy In Computing and Communications/13th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE), pp. 350–357. IEEE (2019)
17. Pfitzmann, A., Hansen, M.: A terminology for talking about privacy by data minimization: anonymity, unlinkability, undetectability, unobservability, pseudonymity, and identity management (2010)
18. Recordon, D., Reed, D.: Openid 2.0: a platform for user-centric identity management. In: Proceedings of the Second ACM Workshop on Digital Identity Management, pp. 11–16 (2006)
19. Reed, M.G., Syverson, P.F., Goldschlag, D.M.: Anonymous connections and onion routing. IEEE J. Sel. Areas Commun. **16**(4), 482–494 (1998)
20. Reiter, M.K., Rubin, A.D.: Crowds: anonymity for web transactions. ACM Trans. Inf. Syst. Secur. (TISSEC) **1**(1), 66–92 (1998)
21. Rennhard, M., Plattner, B.: Practical anonymity for the masses with MorphMix. In: Juels, A. (ed.) FC 2004. LNCS, vol. 3110, pp. 233–250. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-27809-2_24
22. Shirazi, F., Simeonovski, M., Asghar, M.R., Backes, M., Diaz, C.: A survey on routing in anonymous communication protocols. ACM Comput. Surv. (CSUR) **51**(3), 1–39 (2018)
23. Von Ahn, L., Bortz, A., Hopper, N.J.: K-anonymous message transmission. In: Proceedings of the 10th ACM Conference on Computer and Communications Security, pp. 122–130 (2003)
24. Wang, G., Wang, B., Wang, T., Nika, A., Zheng, H., Zhao, B.Y.: Whispers in the dark: analysis of an anonymous social network. In: Proceedings of the 2014 Conference on Internet Measurement Conference, pp. 137–150 (2014)

25. Wang, Q., Borisov, N.: Octopus: a secure and anonymous dht lookup. In: 2012 IEEE 32nd International Conference on Distributed Computing Systems, pp. 325–334 (2012)
26. Wang, W., Motani, M., Srinivasan, V.: Dependent link padding algorithms for low latency anonymity systems. In: Proceedings of the 15th ACM Conference on Computer and Communications Security, pp. 323–332 (2008)
27. Young, A.L., Yung, M.: The drunk motorcyclist protocol for anonymous communication. In: 2014 IEEE Conference on Communications and Network Security, pp. 157–165. IEEE (2014)