On the anonymity of privacy-preserving many-to-many communication in the presence of node churn and attacks

Jörg Daubert[a,b] and Tim Grube[b] and Max Muehlhaeuser[b] and Mathias Fischer[c]
[a]AGT International, Germany
[b]Telecooperation Group, Technische Universität Darmstadt / CASED, Germany
{daubert, tim.grube, max}@tk.informatik.tu-darmstadt.de
[c]Networking and Security Group, International Computer Science Institute, Berkeley, USA
mfischer@icsi.berkeley.edu

***Abstract*—Anonymity can protect from political repression in Online Social Networks (OSNs) as well as from undesired profiling, e.g., by advertisement companies, in todays' Internet. P2P-based anonymous publish-subscribe (pub-sub) is a highly-scalable approach to protect anonymity while enabling efficient many-to-many communication between services and users. However, churn and the resulting overlay degradation in P2P-based pub-sub systems require repairs and optimizations to maintain anonymity and efficiency. This paper analyzes attacks on such repair and optimization functions to disclose participants. For that, we apply a strong attacker model that combines large-scale traffic monitoring with malicious insiders. Furthermore, we propose and evaluate heuristic countermeasures. Our findings indicate that some attacks can be mitigated at reasonable costs. However, churn seems to remain a major threat to anonymity.**

***Index Terms*—Anonymity, Overlay, Attack, Optimization**

I. Introduction

Pub-sub is a distributed programming paradigm that enables many-to-many communication. Such communication is a typical use case of P2P-based systems, where participants form a loosely coupled group based upon interests, and all of them communicate with each other. For instance, hashtags in the OSN Twitter resemble the pub-sub paradigm. Participants embed hashtags in their published tweets while participants also receive tweet streams given a hashtag. Pub-sub connects such producers and consumers of information without requiring them to know each other. Information is routed according to interests (subscriptions), which are expressed by consumers (*subscribers*) along topics. Producers (*publishers*) annotate their notifications with such topics, and hand them over to brokers (*forwarders*), which then disseminate the notifications to the subscribers. In a P2P pub-sub system, every node can take over all roles.

Popular Online Social Networks (OSNs), e.g., Twitter and Facebook, play an important role in today's dissemination of information. In particular, the dissemination of time-critical information, the publication of news under political repression, and censorship emphasize the importance of OSNs. However, OSNs leak private information, e.g., social relations, personal preferences and communication behavior, to OSN operators, the advertisement industry, criminals, and governments. Hence, privacy-preserving OSNs and middleware to build such systems have emerged. Such privacy-preserving solutions attempt to protect the confidentiality of information and the anonymity of participants at the same time.

Anonymous pub-sub systems [1] apply techniques from *Tor* [2], *MIXnets*, and *Crowds* [3] directly to the domain of pub-sub. Information is not only encrypted to protect confidentiality, but also relayed with no IDs attached to protect anonymity by increasing the size of *anonymity sets*. However, anonymization-only services such as *Tor* and *MIXnets* do not adapt well to the many-2-many communication in OSNs. Due to their semi-central nature, they can be easily blocked [4].

Anonymous P2P-based pub-sub [1] extends secure pub-sub [5] that only provides confidentiality by anonymity and does not presume a (semi-) central infrastructure. P2P solutions suffer from node churn—nodes joining and leaving overlays or the whole system—and maintain low message overhead by optimizing the topology of overlays [6]. Such mechanisms can be exploited by anonymity attackers, too, to reduce anonymity sets.

The analysis of anonymity is mainly based on two major attacker models: a *global observer*, i.e., a passively monitoring attacker with global knowledge, or a single *malicious insider*. However, collusion of multiple malicious insiders and the combination of both attacker models should be considered as well [7]. Furthermore, anonymous P2P pub-sub requires periodic overlay optimizations, such as position changes for anonymity protection, optimizations as a result of node churn, and to optimize the overlay for an efficient message delivery. However, such optimizations have not been analyzed regarding their impact on the resulting anonymity protection so far.

In this paper, we use a powerful combination of global observer attacker and malicious insider to analyze the anonymity properties of a pub-sub system. This attacker abuses the two overlay optimizations *cover traffic* and *position changes*, as well as churn, to break anonymity of subscribers. Furthermore, we propose countermeasures against such attacks based upon opportunistic node behavior. We evaluate attacks and countermeasures via extensive real-world simulations of the anonymous pub-sub system used in [7] and earlier work.

Our results indicate that the combined attacker can successfully de-anonymize subscribers in anonymous P2P-based

pub-sub. However, our proposed countermeasures can mitigate or even prevent these kinds of attacks completely at low additional signaling overhead.

The remainder of this paper is structured as follows: Section II summarizes notation, construction, and optimization of anonymous pub-sub. Section III explains attacks on these optimizations, and proposes countermeasures against the attacks. Section IV summarizes our evaluation of the proposed attacks and countermeasures. Section V summarizes this paper.

II. Anonymous P2P publish-subscribe

This section provides a formal model for anonymous pub-sub systems, proposes our anonymity attacker model, and introduces our construction for anonymous unstructured P2P pub-sub. Furthermore, this section discusses the overlay optimizations cover traffic and position changes as well as the impact of churn.

A. System model

A distributed pub-sub system is realized on top of a basic overlay, denoted as a graph $G = (V, E)$ of nodes V with edges $E \subset V \times V$. An attribute overlay $\mathcal{M}_a = (V_a, E_a)$ distributes messages for an attribute $a \in \mathcal{A}$ (interests). The participants $V_a = \mathcal{P}_a \cup \mathcal{S}_a \cup \mathcal{F}_a$ in $\mathcal{M}_a$ consists of publishers $\mathcal{P}_a$, which disseminate new information for interest a, and subscribers $\mathcal{S}_a$, which are interested in information about a. The nodes $\mathcal{F}_a$ are not interested in information about a but rather another attribute a'. Thus, they contribute to the attribute overlay by relaying messages about a.

Within an overlay $\mathcal{M}_a$, messages are disseminated over direct connections $E_a = \{(n_1, n_2) \in E : n_1, n_2 \in V_a\}$. Using $\forall p \in \mathcal{P}_a \, \forall s \in \mathcal{S}_a : path_a(p, s)$ one obtains all traversed nodes on a shortest path between publisher p and subscriber s in $\mathcal{M}_a$. The set $N(v) = \{w \mid \forall (v, w) \in E\}$ contains the neighbors of v in V; the set $N_a(v) = \{w \mid \forall (v, w) \in E_a\}$ contains the neighbors of v in $\mathcal{M}_a$. $N_a^+(v)$ denotes the overlay predecessors and $N_a^-(v)$ the successors regarding the notification flow. The time $t \in T : T = \{1, 2, \dots\}$ indicates snapshots, e.g., G^t and $\mathcal{S}_a^t$. Every system snapshot is represented by an increment of t.

A pub-sub system is inherently dynamic. Node churn $\varphi \in [0, 1]$ denotes the ratio of nodes that are subject to churn during a time interval $[t_i, t_{i+1}]$. Churn occurs in two characteristics: first, nodes join and leave G, which also affect overlays $\mathcal{M}_a$. Second, subscribers and publishers join and leave attribute overlays $\mathcal{M}_a$ but not G. Nodes send a subscription message (m_{sub}) to join an attribute overlay. Advertisement messages m_{adv} from publishers indicate the availability of overlays. To leave an attribute overlay, they have to send an unsubscription (m_{unsub}) or unadvertisement (m_{unadv}), depending on their role in the attribute overlay. When nodes fail, their neighbors act as if they received an unsubscription / unadvertise message from that node.

B. Attacker model

The anonymity attacker attempts to reveal all nodes subscribing to or publishing to an attribute, i.e., given a, identify $\mathcal{P}_a$ and $\mathcal{S}_a$. To measure the attack success via a set-based metric [7], the attacker proposes a candidate set $\mathcal{S}'_a$ (and $\mathcal{P}'_a$ respectively) with likely subscribers. This is particularly difficult as subscribers may also have the role of a forwarder, i.e., $|\mathcal{S}_a \cap \mathcal{F}_a| \geq 0$. With a perfect attack, the attacker achieves $\mathcal{S}'_a = \mathcal{S}_a$; with perfect anonymity $V_a = \mathcal{S}'_a$.

Various capability models for attackers have been proposed: a global attacker that observes all communication, as well as malicious insiders [2], [3]. We use a strong attacker, whose capabilities combine global observer and malicious insider [7]. The global observer provides the topology G and the attribute overlays $\mathcal{M}_a$ by tracing message flows. Furthermore, the attacker estimates sizes of the publisher and subscriber sets $|\mathcal{P}_a|'$ and $|\mathcal{S}_a|'$. The active insiders send, receive, and decipher messages under the coordination of the global observer. For instance, to identify subscribers, we also assume that the active insider possesses the keys of a publisher to send valid messages on behalf of a publisher to the subscribers. This attacker model is very strong, but we believe that compromised devices (active insider) and large-scale traffic analysis (global passive attacker) have to be indeed considered with respect to the Snowden/NSA leaks.

The success of the attacker can be measured by information gain $g_t = H_{t-1} - H_t$ that expresses what the attacker "learned" during an attack [7]. To calculate the gain, we establish a probability distribution $\mathbf{v}$ at every time t containing the probability of every candidate node $v \in V$ being a subscriber in $\mathcal{S}_a$. Then, we calculate the difference for all nodes compared to the previous time $t - 1$ as given by Equation (1). Based on these differences, we calculate the Shannon entropy via Equation (2). The difference of this entropy between time t and $t - 1$ denotes the gain.

$$p_{diff}(v_i, t) = 1 - |\mathbf{v}_{t-1}[v_i] - \mathbf{v}_t[v_i]| \quad (1)$$

$$H_t = -\sum_{v_i \in |V_a|} p_{diff}(v_i, t) * log_2(p_{diff}(v_i, t)) \quad (2)$$

C. Anonymous publish-subscribe overlay construction

To establish the attribute overlay $\mathcal{M}_a$, publishers in $\mathcal{P}_a$ distribute routing information on a by flooding advertisement m_{adv} in G. Interested subscribers $\mathcal{S}_a$ reply via subscriptions m_{sub}. Nodes acting as forwarders $\mathcal{F}_a$ ensure the connection by forwarding messages. Thus, all three sets together form $\mathcal{M}_a$ through which publishers distribute notifications m_{notif} [7]. As a result, message dissemination in $\mathcal{M}_a$ causes low message overhead and low latency.

Other constructions such as [1] flood subscriptions rather than advertisements, but result in similar overlay networks when distributing notifications.

Subscribers and publishers leave overlays by sending m_{unsub} and m_{unadv} messages. Forwarders $f \in \mathcal{F}_a$ act on

behalf of publishers and subscriber and thus use the same messages. A subscriber s may need to reconnect to $\mathcal{M}_a$ if s lost its connection via a $f \in \mathcal{M}_a$. Hence, repair mechanisms have to be performed, e.g., gossiping with remaining neighbors of s or localized flooding by neighbors of f.

D. Anonymity-enhancing overlay optimizations

After their initial construction, churn degrades requirements such as *minimal notification delivery delay*. Therefore, optimizations to the overlay are required to maintain these requirements. Furthermore, *anonymity* may not be reached via the initial overlay construction, and thus has to be established via overlay optimization as well. We discuss two of those:

a) Cover traffic: Overlay topologies expose properties [8] that can be exploited by an anonymity attacker, e.g., to reveal leaf nodes as subscribers. Cover traffic can eliminate topological properties. MIX networks eliminate such properties, but are not applicable in low-latency systems and cause signalling overhead. Receiver-bound cover traffic [9] protects receivers by relaying messages to further nodes, concealing their role as original receiver. Thus, no waiting times are required.

We adapt this concept as an overlay optimization. Subscribers without "outgoing" overlay neighbors in $\mathcal{M}_a$ select neighbors from G, and add them to the attribute overlay. Formally, such a subscriber s has one neighbor in $\mathcal{M}_a$ but more edges in G (Equation (3)). The subscriber then selects one or more nodes v that would receive notifications via s if they were subscribers. Constraint (4) formalizes this condition.

$$s \in \mathcal{S}_a : |N_a(s)| = 1 \wedge |N(s)| > 1 \quad (3)$$

$$v \in V \setminus V_a \wedge v \in N(s) \wedge path_a(p, v) = (path_a(p, s), v) \quad (4)$$

A node v must not be part in the attribute overlay, v must be a neighbor of s, and s must be the direct predecessor of v on the shortest path from a publisher p to v. The subscriber s cannot assess the latter condition with only local knowledge and thus has to guess. The subscriber s then adds node v to the overlay $\mathcal{M}_a$, e.g., by informing v to subscribe to a. As a result, s is no longer "exposed as a leaf" on the one hand, and the potential anonymity set grows as V_a—the number of nodes in the overlay—grows.

b) Position changes: While cover traffic improves anonymity, it does not change the inner overlay structure. Position changes, usually used to minimize the delivery delay of notifications [10], can solve this issue. A random *position change* can be realized by adjacent nodes in $\mathcal{M}_a$ switching their positions. That is, nodes w, x switch positions, e.g., the path (v, w, x, y) becomes (v, x, w, y).

The topology now does not expose undesired properties as every node in V_a can take any position in $\mathcal{M}_a$. However, insider attacks abusing this mechanism have to be identified and analyzed.

E. Churn induced overlay changes

Churn adds dynamics to a P2P system by participants that join and leave the system. While the accurate characterization of churn in a system is a challenge [6], the mere dynamics exhibit information that can be abused by an attacker.

This paper distinguishes two types of churn: churn in the basic overlay, and churn in the attribute overlay. While nodes leaving the attribute overlay $\mathcal{M}_a$ have no immediate impact on the topology—a subscriber from $\mathcal{S}_a$ just becomes a forwarder from $\mathcal{F}_a$—joining $\mathcal{M}_a$ may cause subsequent overlay optimizations. Furthermore, leaving the basic overlay G may even require overlay repairs as new paths have to be formed to connect $\mathcal{M}_a$.

III. Anonymity attacks and countermeasures

This section structures our anonymity attacker model according to its capabilities and proposes four attacks to reveal anonymity. We propose three countermeasures to prevent or at least mitigate these attacks.

A. Attack based on churn

The global observer monitors network links and thus observes churn, e.g., when a node joins or leaves the basic overlay G and the attribute overlay $\mathcal{M}_a$ simultaneously.

The global observer simply has to observe changes in the communication patterns before and after node churn. In case communication paths change as a result of node churn, this information can be used to classify nodes from V_a to be forwarders in $\mathcal{F}_a$. Similar attacks on other anonymization services have been reported as *intersection attack* [11].

(a) s_2 joins late, ideally s_1 would be connected via dotted arc

(b) s_1 and s_3 leave, dotted edges disappear

Fig. 1: Effects of churn.

Fig. 1a depicts the case of the shaded node s_2 joining at time $t+1$. Initially, only s_1 is connected with p. Thus, $\mathcal{S}_a^t = \{s_1\}$, $\mathcal{F}_a^t = \{f_1, f_2\}$, and $path_a^t(p, s_1) = (p, f_1, f_2, s_1)$ as s_2 has not joined at time t yet. Once s_2 joins, it directly connects with p as $path_a^{t+1}(p, s_2) = (p, s_2)$. However, $path_a^{t+1}(p, s_1)$ should now become (p, s_2, s_1) rather than (p, f_1, f_2, s_1).

As a result, both subscribers s_1, s_2 are exposed as leaf nodes at $t+1$, whereas ideally only s_1 should be exposed. Moreover, s_2 may use cover traffic (cf. Section II-D) and select s_1 for that, which is valid according to Constraint (4). However, s_1 then receives duplicate messages, indicating the use of cover traffic to the global observer.

Fig. 1b depicts the case of the shaded nodes s_1, s_3 leaving at $t+1$. As s_1 leaves $\mathcal{M}_a$, f_1 may also leave $\mathcal{M}_a$ as it is no longer required to relay message at $t+1$. This "ripple effect" allows the global observer to reason that $f_1 \in \mathcal{F}_a$ and $f_1 \notin \mathcal{S}_a$.

Likewise, when s_3 leaves $\mathcal{M}_a$ at $t+1$, s_2 will remain in $\mathcal{M}_a$ as it is interested in the notifications for a. Hence, the global observer reasons that $s_2 \in \mathcal{S}_a$ and $s_2 \notin \mathcal{F}_a$.

In summary, node churn exposes subscribers over time. Furthermore, churn affects the overlay topology, which in consequence has impacts on the overlay optimization that might lead to further exposure. Thus, the effects of churn must be carefully considered when introducing new anonymity-enhancing overlay modifications.

B. Attack on cover traffic

Cover traffic is susceptible to attacks when used in combination with other overlay optimizations such as position changes. An active insider can stimulate such optimizations to cause benign nodes to react and thus to leak valuable information to the global observer.

(a) Initial overlay (b) s_1, f_1 changed positions

Fig. 2: Cover traffic under position changes.

Cover traffic is initiated by a node $s_1 \in \mathcal{M}_a$ that intents to cover itself by starting to relay messages to node f_2. However, this might create a topological anomaly when using additional position changes. When two nodes, f_1, s_1 change positions, s_1 keeps f_2 as cover node, as f_2 is not part of $\mathcal{M}_a$. This exchange of neighbors looks abnormal from the global observer's point of view.

The global observer monitors such abnormal exchanges (Fig. 2a). At time t, f_2 is cover node of s_1, f_1 intents to change positions with s_1. Fig. 2b: at time $t+1$, f_1 changed positions with s. Hence, the global observer reasons that s_1 is covering itself, and that f_2 is a cover node. The global observer therefore assigns these nodes to $\mathcal{S}'_a$ and $\mathcal{F}'_a$ respectively.

The malicious insider can extend this attack by forcing position changes, i.e., the active insider takes position of node f_1 (cf. Fig. 2a) to force s_1 to change the position with it. Using this method, the malicious insider targets uncertain nodes, i.e., nodes that have not been assigned to $\mathcal{S}'_a$ and $\mathcal{F}'_a$, and cause observable activity in dormant overlays.

In summary, while cover traffic is an optimization to protect anonymity, it can cause the opposite effect in combination with other overlay optimization.

C. Attack via overplay spamming

Overlay optimizations such as cover traffic and position changes are supposed to prevent the global observer from observing properties of the attribute overlay. However, in combination with a malicious insider, the overlay can be spammed with messages. This allows the global observer to observe overlay optimizations as each spammed message that is transmitted within an attribute overlay provides a topology snapshot. As a result, the global observer can *reenact* these optimizations and identify exposed leaf nodes (subscribers).

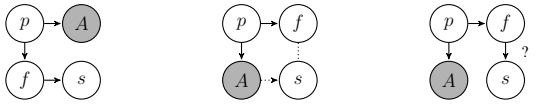
To observe position changes, the malicious insider must send messages faster than the duration in between position changes. Following the Nyqvist-Shannon sampling theorem, the malicious insider must a least keep a message rate as given by $rate$ in Equation (5) that adapts the theorem to the position change mechanism. The equation assumes that control messages for position changes are embedded in covert traffic, e.g., heartbeats, that are sent for instance every three seconds. For that, heartbeats must be padded and encrypted (content remains covert) to be able to hold control messages.

$$rate = \frac{1}{2 \times 4 \times 3s}; \quad changes_{max} = \frac{|V|}{2 \times N(v)_{avg} - 1} \quad (5)$$

Changing positions of two nodes requires at least four non-concurrent messages, two messages for a two-phase commit protocol and two messages for the neighbor handover. The symbol $changes_{max}$ in Equation (5) provides an approximation for the number of simultaneous position changes. To achieve this message rate, the active insider has to take over the position of a publisher and send notifications. To avoid the detection of overlay spamming, the insider may operate several malicious nodes and spread the burden among these nodes to reduce the sending rate per node.

D. Node cornering attack

In the node cornering attack, the malicious insider disconnects an overlay node from the overlay. The global observer then observers the behavior of this node. If the nodes re-establishes its connection to the overlay via an alternate path, it is a subscriber interested in notifications from this overlay. If not, the node is a forwarder.



(a) Change (b) Disconnect (c) Observe

Fig. 3: Corner attack: A changes into a position close to s, disconnects s, and then observes s.

To perform this attack, the malicious insider first has to "corner" the target node, then disconnect it from the overlay. Fig. 3 illustrates this approach. Fig. 3a shows the initial setup. To analyze s, malicious insider A first has to get close to s. For that, A uses the position change method. Fig. 3b shows that A has trapped s in a corner and interrupts the overlay connection. Fig. 3c shows now the global observer just need to observe the connection between f and s. If f starts to relay notifications to s, he can be sure that s is a subscriber. The Procedure cornerAttack shows the corner attack as recursive algorithm. However, every corner attack requires time for the malicious insider to get into position.

Procedure cornerAttack(N_a^+(v), N_a^-(v))

```
for w ∈ N_a^-(v) do
    if N_a^-(w) = ∅ then
        disconnect(w) // destroy edge (v,w);
        if observeReconnect(w) // via global observer then
            v[w] ← 1 // definitive subscribe;
        else
            v[w] ← 0 // definitive not a subscriber;
        normalize(v) // v always sums up to 1;
    else
        changePosition(v,w) // v becomes w and vice versa;
        cornerAttack(N_a^+(w), N_a^-(w));
```

E. Countermeasures

We propose two countermeasures to mitigate the introduced attack schemes. First, opportunistic node behavior (ONB) and second, publisher slow start (PSS).

With ONB, cover nodes should join an attribute overlay via a subscription and act like a normal overlay member (ONB1). Furthermore, nodes with low traffic join random attribute overlays to generate additional cover traffic (ONB2).

With PSS, publishers should wait before sending notifications after an attribute overlay has been established. Then, overlay optimization can be performed without the global observer being able to observe the initial notification flow.

ONB1 differs from cover nodes. With cover nodes, the node s covering itself relays notifications to the cover node. This approach has the drawback that s may forward to an unstable cover node f. The cover node may not perform all overlay optimizations. This can be exploited via one of the introduced attacks.

With ONB1, s asks f to join the overlay. If f is not suitable, e.g., s is not in $path_a(p, f)$, f ignores the request rendering the suboptimal path attack ineffective. If f is suitable, f subscribes to s and thus joins the set $\mathcal{S}_a$ from the global observrs perspective and behaves normally, e.g., also follows position changes and attempts to reconnect after an overlay disconnect. That renders the node cornering attack ineffective. However, the additional round-trip requires time and may thus still expose s during that duration. Furthermore, also f behaves like a member $\mathcal{S}_a$, it is actually part of $\mathcal{F}_a$ as it does not possess necessary secrets to decipher messages for $\mathcal{S}_a$.

To mitigate this drawback, nodes $v \notin V_a$ may opportunistically subscribe to a (ONB2). Therefore, s might not be exposed in the first place. Furthermore, the ratio $|\mathcal{S}_a|/|V_a|$ decreases as $\mathcal{F}_a$ and thus V_a grow. Thus, the set V_a exposes less statistical information about $\mathcal{S}_a$.

With PSS, publishers $\mathcal{P}_a$ wait after the establishment of the attribute overlay for overlay optimizations to take place before sending notifications. This behavior ensures that the global observer cannot observe or reverse the initial overlay topology $\mathcal{M}_a^0$. However, as a malicious publisher may not comply with this countermeasure, forwarders have to suppress notifications during that duration as well.

TABLE I: Effectiveness of countermeasures against attacks.

	Countermeasures		
Attacks	ONB1	ONB2	PSS
On churn	✓	?	✗
On cover traffic	✓	?	✗
Overlay spamming	✗	?	✓
Node cornering	✓	✓	✗

$$t_{wait} \approx 2 \times |path_a(p, s)|_{avg} \times 12s \tag{6}$$

This duration can be estimated by nodes, e.g., Equation (6) approximates the duration for the position change overlay optimization (4 messages times $3s$ for the heartbeat interval each). In addition, a rate limit must be enforced by forwarders to prevent publishers from sending notifications at optimal sampling rate (cf. Equation (5)).

The presented countermeasures address the attacks as stated by Table I. In particular, no single countermeasure protects against all presented attack. The ONB is a probabilistic countermeasure. Thus, its success depends upon the overlay topology as well as parametrization μ. ONB does not introduce an additional vulnerability. However, a malicious publisher may completely prevent PSS by ignoring the waiting time. The following section elaborates how well the presented attacks disclose participants and how good the countermeasures protect against these attacks via an extensive simulation.

IV. Evaluation

This section analyzes the proposed attacks and countermeasures in a realistic simulation setup. For that, we answer the question regarding the attackers gain from observing churn as well as cover traffic; how much ONB mitigates these attacks; what waiting time for PSS works best to counter the attacker; how much the attacker gains via the node cornering attack.

A. Simulation setup

To evaluate attacks and countermeasures, we use the *OMNeT++*[1] simulation model from [7]. The anonymous pub-sub application is implemented based upon UDP. All malicious insiders and the global observer share joint data via an out-of-band channel. This data contains the probability distribution $\mathbf{v}$ over all nodes V_a as shown in Equation (1). The simulation is build upon a fixed basic overlay. On top the anonymous pub-sub system constructs attribute overlays. The generation follows the random graph model with a size of $|V| \in [100, 1000]$ nodes and a varying ratio of edges to ensure a stable neighborhood size for increasing $|V|$. Thus, we obtain an average diameter in $\mathcal{M}_a$ of 6.51 ($|V| = 100$) up to 17.35 ($|V| = 1000$). We designate $|\mathcal{S}_a|/|V| = 0.1$ of these nodes as subscribers $\mathcal{S}_a$. Furthermore, we use the probability μ to indicate the likelihood of choosing a neighbor as cover node. The set $\mathcal{C}_a$ contains the opportunistic cover nodes for ONB.

[1] http://www.omnetpp.org

TABLE II: Simulation parameters.

Parameters and default values	
$\|V\| \in [100, 1000]$	// size of the basic overlay
$\|S_a\|/\|V\| = 0.1$	// ratio of subscribers
$\mu = 0.4$	// cover neighbor probability
$\|\mathcal{C}_a\|/\|V\| \in [0, 0.5]$	// ratio of ONB2 nodes
$\varphi \in \{0.05, 0.1, 0.2\}$	// ratio of node churn
$runs \in \{200, 300\}$	// repetitions per setup

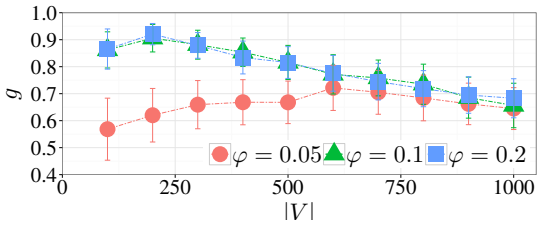


Fig. 4: Influence of churn on the attackers' information gain.

We repeat every experiment $runs \in \{200, 300\}$ times and calculate the 95% confidence intervals. Table II summarizes the parameters.

For the attacker, we assume that the global observer knows the topology of the basic overlay G. Furthermore, the global observer provides background information in terms of the subscriber ratio $|S_a|/|V|$. The malicious insider contributes secrets known to a publisher from $\mathcal{P}_a$. The attacker attempts to expose the set of subscribers $\mathcal{S}_a$. To measure success, we use the normalized information gain per subscriber g (cf. Section II-B) as metric.

B. Impact of churn

The effects of churn allow the global observer to distinguish subscribers from forwarders. However, it remains unclear how much the global observer gains from churn.

We simulate with the settings from Table II and vary the number of nodes $|V|$ as well as the ratio of nodes involved in churn $\varphi \in \{0.05, 0.1, 0.2\}$ and measure g.

We expect a linear correlation of g with φ as the global observer can observe more overlay repairs with increasing φ. Furthermore, we expect an increase of g with $|V|$ due to longer path lengths in the overlay and thus more nodes affected by churn.

Fig. 4 depicts the results of the simulation. As expected, g increases with φ for small amounts of nodes. However, the global observer does not seem to benefit from high churn for higher number of nodes—the results for $\varphi = 0.05$ and $\varphi = 0.2$ become almost indistinguishable. This is because more subscribers become part of every $path_a(p, s)$. These nodes cause the path to remain stable (persists for many snapshots T). Hence, the global observer identifies less forwarders.

In summary, the simulation results indicate that churn poses a high risk to nodes being de-anonymized.

C. Cover nodes under overlay optimizations

Cover traffic can be observed by the global observer when overlay optimizations take place. ONB1 and ONB2 can mitigate this problem.

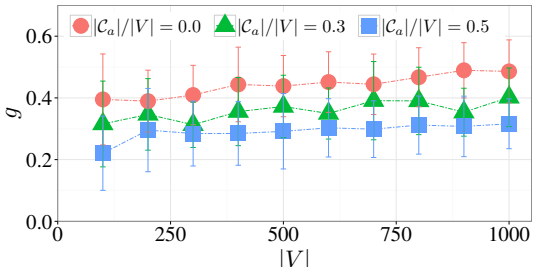


Fig. 5: Overlay optimizations with cover traffic and ONB.

We simulate with the same parameters as before, and set the probability of choosing a neighbor for cover traffic to $\mu = 0.4$ for ONB1. We variate the number of nodes as well as the ratio of opportunistic cover subscribers $|\mathcal{C}_a|/|V| \in \{0, 0.3, 0.5\}$, and measure g.

We expect a linear correlation between $|\mathcal{C}_a|/|V|$ and g as it becomes difficult for the attacker with increasing $|\mathcal{C}_a|/|V|$ to identify subscribers because less overlay nodes are forwarders. Furthermore, g should reach 0.5 with $|\mathcal{C}_a| = |S_a|$ as there are equally many subscribers and opportunistic cover nodes.

Fig. 5 shows the results of the simulation. As expected, the attacker achieves high information gain with cover traffic without ONB2 Likewise, opportunistic cover nodes significantly lower the gain. However, the attacker seems to gain slightly better with high number of nodes. This happens as $|path_a(p, s)|$ increases with higher $|V|$ and thus more optimization observations per node.

In summary, cover traffic reveals exploitable information to the attacker during overlay optimizations. Opportunistic cover nodes (ONB) mitigate this issue. However, a significant number of opportunistic nodes exceeding the number of real subscribers is required to achieve good protection.

D. Overlay spamming

With overlay spamming via the malicious insider, the global observer can observe overlay optimizations and thus eliminate the protection of position changes. PSS as countermeasure mitigates this attack, but the waiting time has to be adjusted correctly.

We use the same parameters as before, and vary the number of nodes $|V|$, and thus also the the average path length $|path_a(p, s)|_{avg}$. We measure the time $t[s]$ it takes every overlay node in V_a to change positions as often as $2 \times |path_a(p, s)|_{avg}$ with $runs = 300$. This allows each node to "travel" from any position in $\mathcal{M}_a$ to any other position. The simulation model reflects characteristics of a real distributed system, e.g., conflicts due to adjacent nodes attempting to change positions simultaneously. We resolve this issue via a two-phase commit protocol [12], where nodes commit to perform a position change before the actual change.

We expect a logarithmic correlation between $|V|$ and t under the assumption that $|path_a(p, s)|_{avg}$ can be approximated with the height of a tree under the random graph model. Hence,

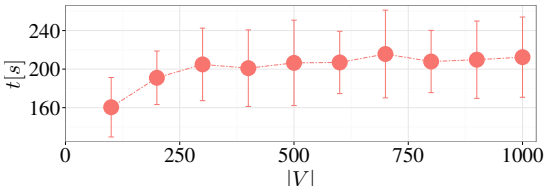


Fig. 6: Waiting time over graph size for PSS.

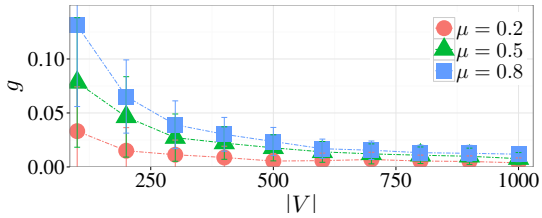


Fig. 7: Information gain with corner attack.

an estimation of the waiting time should be possible without global knowledge of $|V|$.

Fig. 6 depicts the results of the simulation. The waiting time quickly converges towards $213.41s$. This is more than an extrapolation based upon Equation (6) and the overlay diameter of $6.51-17.35$ (the diameter approximates $2\times|path(p,s)|_{avg}$). This happens in particular for small overlays as the two-phase commit protocol fails more often due to conflicts of parallel position changes.

In summary, the waiting time can be well estimated. However, a waiting time of 3-4 minutes may not be always acceptable. This waiting time can be reduced by increasing the rate of overlay control messages at the cost of increased message overhead.

E. Node cornering

With node cornering, the malicious insider exploits the position changes to explore the overlay guided by the global observer. The global observer observes the node behavior after the internal one disconnects them. However, it remains unclear how many cover nodes the attack can identify.

We use the same parameters as before, vary the number of nodes $|V|$ as well as $\mu \in [0.2, 0.5, 0.8]$, and measure g.

We expect an increase of g with μ, as the attacker should be able to identify more cover nodes. Furthermore, we expect a slight decrease in g with $|V|$ due to a slower growth of $|path_a(p,s)|_{avg}$ compared to the growth of $|V|$.

Fig. 7 depicts the results of the simulation. As expected, the gain increases with μ and drops quickly with increasing $|V|$. However, even with high μ, the overall gain for a single malicious insider remains low compared to other attacks.

In summary, this attack is not effective as a single malicious insider can only learn when moving into one branch of the overlay. The malicious insider can only compensate by waiting for other nodes to initiate position changes and thus "pull" the malicious insider back again. Alternatively, the malicious insider could collude. Still, the attack is invasive and could be detected by collaborating nodes.

V. Conclusion

In this paper, we discussed attacks on unstructured anonymous pub-sub overlays under churn. We proposed a strong attacker model that combines a malicious insider and a global observer to analyze participant anonymity under overlay optimizations and churn. We described attacks and analyzed them via a realistic simulation model.

Churn causes overlay repairs and thus enables the attacker to observe the overlay. This observation is powerful for de-anonymizing subscribers. ONB only partially mitigates this.

Cover traffic is a proven method to protect anonymity. However, it can also causes anomalies during optimizations. Observing these anomalies is a strong method to de-anonymize subscribers. Moreover, ONB which increases signaling overhead, has only a limited mitigation effect.

Invasive attacker behavior, such as overlay spamming and node corning, has only minor impact on the attackers success. Furthermore, such attacks are easy to detect and effective prevent, e.g., via rate limits and initial delay.

In summary, churn poses a high risk for anonymity and is hard to mitigate. Future work will focus on attack prevention methods to counter the effects of churn.

Acknowledgment

This work has been partially funded by IITP grant funded by the Korean government (MSIP) (No.B0101-15-1292). This work has been co-funded by the DFG as part of project B.2 within the RTG 2050 "Privacy and Trust for Mobile Users". This work was supported by CASED (www.cased.de). The authors would like to thank Roman Pilipchuk for his support.

References

[1] A. Shikfa, M. Önen, and R. Molva, "Privacy and confidentiality in context-based and epidemic forwarding," *ComCom*, vol. 33, no. 13, pp. 1493–1504, 2010.

[2] R. Dingledine, N. Mathewson, and P. F. Syverson, "Tor: The second-generation onion router," in *USENIX*. USENIX, 2004, pp. 303–320.

[3] M. K. Reiter and A. D. Rubin, "Crowds: Anonymity for web transactions," *ACM TISSEC*, vol. 1, no. 1, pp. 66–92, 1998.

[4] P. Winter and S. Lindskog, "How the great firewall of china is blocking tor," in *USENIX FOCI Workshop*. USENIX, 2012.

[5] C. Raiciu and D. S. Rosenblum, "Enabling confidentiality in content-based publish/subscribe infrastructures," in *SecureComm*. IEEE, 2006, pp. 1–11.

[6] D. Stutzbach and R. Rejaie, "Understanding churn in peer-to-peer networks," in *ACM IMC*. ACM, 2006, pp. 189–202.

[7] J. Daubert, T. Grube, M. Mühlhäuser, and M. Fischer, "Internal attacks in anonymous publish-subscribe P2P overlays," in *NetSys*. IEEE, 2015, pp. 1–8.

[8] L. Singh and J. Zhan, "Measuring topological anonymity in social networks," in *IEEE GrC*. IEEE, 2007, pp. 770–774.

[9] N. Mallesh and M. Wright, "Countering statistical disclosure with receiver-bound cover traffic," in *ESORICS*, ser. LNCS, vol. 4734. Springer, 2007, pp. 547–562.

[10] B. Schiller, S. Roos, A. Höfer, and T. Strufe, "Attack resistant network embeddings for darknets," in *IEEE SRDS Workshops*. IEEE Computer Society, 2011, pp. 90–95.

[11] J. Raymond, "Traffic analysis: Protocols, attacks, design issues, and open problems," in *Designing PETs*, ser. LNCS, vol. 2009. Springer, 2000, pp. 10–29.

[12] P. A. Bernstein, V. Hadzilacos, and N. Goodman, *Concurrency Control and Recovery in Database Systems*. Addison-Wesley, 1987.