Drac: An Architecture for Anonymous Low-Volume Communications

George Danezis[1], Claudia Diaz[2], Carmela Troncoso[2], and Ben Laurie[3]

[1] Microsoft Research Cambridge
gdane@microsoft.com
[2] K.U. Leuven/IBBT, ESAT/SCD-COSIC
firstname.lastname@esat.kuleuven.be
[3] Google, Inc.
ben@links.org

Abstract. We present Drac, a system designed to provide anonymity and unobservability for real-time instant messaging and voice-over-IP communications against a global passive adversary. The system uses a relay based anonymization mechanism where circuits are routed over a social network in a peer-to-peer fashion, using full padding strategies and separate epochs to hide connection and disconnection events. Unlike established systems, Drac gives away the identity of a user's friends to guarantee the unobservability of actual calls, while still providing anonymity when talking to untrusted third parties. We present the core design and components of Drac, we discuss the key ways in which it challenges our current concepts of anonymity and provide an initial simulation-based security analysis.

1 Introduction

Anonymous communications are important since the addressing, timing and volume of traffic can in some cases leak as much information as its content [37]. This is particularly true for real-time communications, as instant messages or phone calls can be indicative of imminent intentions or plans, e.g. in military command and control systems, or sensitive personal information, like medical status or family life, in civilian settings. Despite this, few systems have been proposed to provide strong anonymity against global passive adversaries for private communications.

Drac aims to provide strong anonymity and traffic analysis guarantees for real-time communications. This is achieved though a peer-to-peer relay based architecture. We assume that the traffic relayed is regular or low volume such as voice-over-IP (VoIP) or instant messaging (IM) respectively. This allows us to use a traffic padding regime and destroy any information leaking from patterns of traffic. Communication sessions are started and ended synchronously to further limit the information leakage.

We also design the trust model of Drac around a friend-of-a-friend architecture: communications between friends are unobservable, and communications with further contacts in the network are anonymous. Despite the anonymity sets being smaller, they are harder than random anonymity sets, in that they are correlated between sessions and an adversary has to infiltrate the social circle of a user to perform insider attacks. Finally, we assume that both parties to a conversation use Drac for their communications and have incentives to stay on-line and relay third party traffic even when they are not communicating: this provides unobservability [27] and is a natural architecture to support incoming voice calls or instant messages.

The aim of this work is to introduce the Drac design and provide a preliminary analysis of anonymity and unobservability. Unobservability is an unusual property, and even defining it or measuring it in a system represents novel challenges. Three aspects of the system are studied though simulations: the anonymity provided against the presence system, and the anonymity and unobservability of communications towards a global passive adversary.

The paper is organised as follows: Sect. 2 presents previous work and building blocks used in Drac; Sect. 3 presents a high level model of Drac and its components; Sect. 4 shows the preliminary evaluation results; finally we discuss some further aspects of Drac in Sect. 5 and offer our conclusions in Sect. 6.

2 Drac and related work

High-latency anonymous communications were introduced by David Chaum [6], and have been implemented in deployed systems such as mixmaster [22] and later mixminion [8]. Those systems are economical in that they do not require cover traffic. On the downside, they delay communications significantly, making it difficult to have a real-time conversation as is required for IM or VoIP.

Onion Routing systems, including Tor [13], provide low latency communications for web-browsing cheaply, by sacrificing security against a global passive adversary. Yet such adversaries are realistic and can be implemented through sampling [24], indirect network measurements [23], or eavesdropping on key Autonomous Systems (AS) [15]. Web browsing loads are bursty and high-bandwidth such that any traffic padding regime would be uneconomical. IM and VoIP loads on the other hand are more regular, or simply low-bandwidth, allowing link and end-to-end padding strategies to be affordable if high security is required.

The ISDN-mix system [26] was specifically designed to provide real-time anonymous communications. As ISDN-mixes, Drac creates connections synchronously in epochs to maintain connection anonymity, but does not implement cascades and does not use the custom ISDN infrastructure to support its operation – instead we assume that the communications are taking place over IP, using off-the-shelf routers.

In this work we are not overly concerned with the cryptographic details of Drac. There exist well established, provably secure, cryptographic constructions to support relaying anonymized messages [9] and extending anonymous connections [16, 18]. Similarly we assume that a padding regime is established that makes the output channels traffic statistically independent of the input channels [31, 34, 36]. This can be done simply by sampling a traffic schedule for the output channel independently and before even seeing the input channels, and sticking to it by adding cover traffic if there is not enough, or dropping messages if the queues become too long.

The trust model Drac uses is a version of restricted routes [7], where paths are created over friendship links. The impact of social networks on anonymity has been studied before [12], and recent work [17] has looked at modifying the global trust assumption common in contemporary anonymous channels. Yet we are the first to propose boldly making use of a social network as the backbone of anonymous paths.

Finally, the analysis we provide follows the information theoretic metrics proposed in [30, 11]. The probabilistic analysis we perform is very much a first analysis of the system, as it is heuristic, and does not take into account all constraints known to the adversary. A full Bayesian analysis [32] would be required to do this, and is the subject of future work. A full analysis of the impact of long term disclosure attacks [19] is also necessary: Drac is designed to provide smaller, but harder anonymity sets, than other systems. The fact that anonymity sets of different epochs are highly correlated (as routing is embedded over a social graph) invalidates previous results and performance bounds of these attacks [25]. These models have so far assumed anonymity sets contain random users, whereas in Drac these are highly correlated and composed of the social surroundings of users.

3 The Drac system

At the core of Drac we have a social network formed by N *users* (or *nodes*.) Each user u_i in this social network is connected to a set of *friends* $\mathcal{F}_i$. We assume that friends have a strong trust relation, and that they use each other to relay communications. For this purpose, friends share cryptographic keys (or at least a weak secret to bootstrap a cryptographic key) that they can use to establish secure communication *links*. Besides communicating with her friends, a user u_i also interacts with a set of *contacts* $\mathcal{C}_i$ to whom she is not connected in the social network. Contacts are people that a user may wish to talk to, but does not necessarily trust for relaying her connections (e.g., a relationship between a patient and her doctor.) We consider that contacts exchange their pseudonyms and establish a long term symmetric key offline (e.g., the patient meets the doctor at the clinic.) Finally, we assume that relationships with friends are public, thus known to the adversary (e.g., extracted from a social network web site [3],) but that relationships with contacts are secret and must be concealed by Drac.

3.1 Establishing communications with Drac

Upon connection to the network, a user establishes low bandwidth bi-directional *heartbeat connections* with each of her friends in order to make her availability known to them. These connection are padded at a very low rate, and are used for signaling purposes (creating and extending connections, starting communications, etc,) as well as for establishing connections with the private presence server (as explained in Sect. 3.2.) In Drac we strictly separate the control plane from the data plane: signalling and presence packets are embedded and routed in the heartbeat traffic such that an external observer cannot differentiate between dummy heartbeat packets and actual messages. Figure 1(left) shows the heartbeat connections between six users $\{u_A, \ldots, u_F\}$ in which $\mathcal{F}_A = \{u_C, u_F\}$, $\mathcal{F}_B = \{u_C, u_E\}$, $\mathcal{F}_C = \{u_A, u_B, u_D\}$, and so forth. By observing heartbeat connections, an adversary does not gain extra knowledge about users, as the friendships are considered public, and the timing and volume of heartbeat traffic does not leak any further information.

Users wish to communicate with contacts, but they are not connected to them in the network. For this purpose each u_i has an *entry point* E_i that she uses to indirectly establish communications. In each epoch users build a *circuit* of depth D to their entry points (using their heartbeat channels.) We describe the circuit creation process using the example network shown in Fig. 1:

1. User u_A selects at random one of her friends to be the first hop of the circuit. Say she chooses u_C from $\mathcal{F}_A = \{u_C, u_F\}$. They establish a secure *link* using their long-term key K_{AC}, and generate a session key k_{AC}.
2. u_A requests u_C to choose a friend at random and extend the circuit to her.
3. User u_C selects a friend at random, say u_D, and creates a new secure link using K_{CD}. Through the extended circuit, u_A and u_D establish a session key k_{AD}. As u_C chooses one of her friends at random to route u_A's traffic, it may be the case that u_A is chosen to participate in her own circuit.
4. Steps 2 and 3 are iterated D times using friends of friends as next hops in the path. The last user in the circuit is the entry point E_A of u_A. In the example above, if $D = 2$, we say that u_D is u_A's entry point E_A. As members of the circuit are chosen at random, u_A may end up being her own entry point.

We note that u_A needs to know her entry point to establish communications with contacts, and thus E_A needs to provide its identity to u_A at the end of the circuit creation process.

The circuit depth D is a security parameter of the system. Longer circuits increase the anonymity provided by Drac as they make tracing communications to their originator more difficult, while shorter circuits result in smaller anonymity sets, as shown in Sect. 4. We consider that the adversary can observe all links, and knows how many circuits are routed through each of them, but does not know the correspondences between inputs and outputs at each node.

Friends communicate with each other through direct links. To ensure that the communication is fully unobservable, both users still establish circuits of depth D in the network, but at least one of them has to choose the other as first hop. When a user u_i with entry point E_i, wants to communicate with one of her contacts u_j with entry point E_j, she requests E_i to extend the circuit to E_j. We call the connection between two entry points *bridge*, and denote it as B_{ij}. We note that bridges between users that are not friends are visible, as they stand out with respect to the edges in the underlying social network, and the

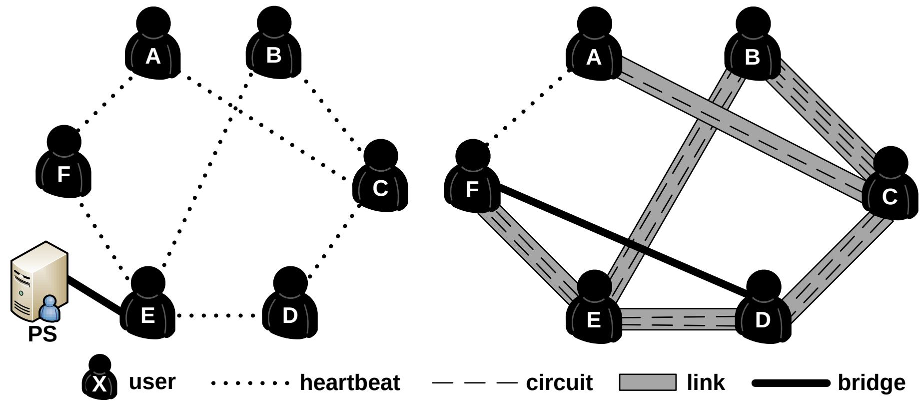


Fig. 1. Underlying social network and connection to the presence server (left.) Adversary's observation of an epoch (right.)

heartbeat channels that the adversary observes. If the entry points of u_i and u_j are friends, an adversary can still observe that there is an extra circuit in the system. However, she cannot distinguish this bridge from other links that are part of a connection between a user and her entry point. Further, when E_i is the same as E_j, no bridge is created and an adversary cannot detect that there is a communication.

To ensure confidentiality of communications, u_i and u_j encrypt messages using the keys that they share with each other, and with the nodes that they use for transit. We denote by $\mathrm{E}_k(M)$ the encryption of message M under key k. Upon receiving a message, an intermediate node processes it using the session key shared with the originator of the message. After processing, the node checks whether the message is addressed to itself. If the result is still a ciphertext the message is relayed to the next node in the circuit, or dismissed at the last node.

Example. Let us consider that u_X talks to u_W through two of her friends u_Y and u_Z (which whom she shares session keys k_{XY} and k_{XZ} respectively,) and two of u_W's friends u_U and u_V (with whom u_W shares k_{WU} and k_{WV}.) u_X and u_W share a session key k_{XW} that they create as explained in Sect. 3.2. The route can be depicted as:

$$u_X \rightarrow u_Y \rightarrow u_Z \Rightarrow u_U \rightarrow u_V \rightarrow u_W$$

where a bridge B_{XW} has been created between u_Z and u_U.

If u_X wishes to package a message M for u_W she encrypts it under k_{XW}, k_{XZ}, and k_{XY}, and sends:

$$u_X \rightarrow u_Y : \mathrm{E}_{k_{XY}}(\mathrm{E}_{k_{XZ}}(\mathrm{E}_{k_{XW}}(M)))$$

The message gets relayed and decrypted by u_Y and u_Z. User u_Z sends to u_U $E_{k_{XW}}(M)$ through the bridge B_{XW}. Then, the message is encrypted under the keys of u_U and u_V. The following message arrives to u_W:

$$u_V \rightarrow u_W : \mathrm{E}_{k_{WV}}(\mathrm{E}_{k_{WU}}(\mathrm{E}_{k_{XW}}(M)))$$

3.2 Private presence server

Users can establish communications with their friends or contacts, and thus need to be reachable by them. To communicate with friends, users can use their direct heartbeat channels. For initiating communications with a contact, we require a private presence server that allows u_i to be reachable by her contact u_j. The presence server is assumed to be cooperative (i.e., follows the protocols) but untrustworthy (i.e., it could be colluding with the adversary in order to deanonymize its users.) In our scheme, we draw some ideas from the Apres [20] system, but we introduce several modifications in order to adapt it to the context of Drac. For simplicity, we only consider one presence server in this work, but we note that Drac could be trivially extended to support several servers.

Each user u_i has a long term identifier ID_i that is known by all her contacts, but not by the presence server. We note that a user u_i may have several IDs, each corresponding to a circle of contacts, so that contacts belonging to different "circles" cannot find out that they know the same user. In order to have unlinkability between time periods and avoid long-term pseudonymous profiling by the presence server, the identifier IDJ_i of u_i in a given time period T is computed as $IDJ_i = H(T, ID_i)$, where $H(x, y)$ is an HMAC of x with key y. As T is published by the presence server, u_i and her contacts are able to compute IDJ_i from her long term identifier ID_i.

In order to be reachable by her contacts, u_i creates a circuit of depth D_p (D_p may or may not be equal to D) to her presence server PS using the heartbeat channels. This presence circuit is built following the same procedure as the one used to construct communication circuits from users to entry points. When the connection is D_p hops long, u_i instructs the last node, E_{P_i}, to send the IDJ_i encrypted with the key of PS to PS. At this point, u_i has an open connection to her presence server, who can list IDJ_i as online.

In Fig. 1(left) we show the heartbeat connections in one epoch. These connections carry *presence circuits* that are unobservable to the attacker. For instance, let us consider that the presence circuit from u_A runs through users u_F and u_E. An adversary can see the bridge between u_E and PS, but cannot distinguish whether this connection comes from u_A (through u_A-u_F-u_E), u_C (through u_C-u_B-u_E or u_C-u_D-u_E), or u_E (through u_E-u_B-u_E, u_E-u_F-u_E, or u_E-u_D-u_E.)

Let us assume u_B wants to communicate with her contact u_A. First, u_B constructs a circuit to PS through the heartbeat channels in a similar way as u_A did to register her presence. We assume that u_A and u_B share a long-term secret key K_{AB}, and that they know each other's long-term IDs (ID_A and ID_B.) User u_B creates a message for PS with the form:

$$E_{PK_{PS}}(IDJ_A, E_{K_{AB}}(E_B, g^{r_B}))\,,$$

where PK_{PS} is the public key of PS, K_{AB} is the shared secret between u_A and u_B, E_B is the entry point of u_B, and r_B is a randomly generated number. PS decrypts the message with its private key, and checks if a user with identifier IDJ_A is connected. If this is the case, then it forwards $E_{K_{AB}}(E_B, g^{r_B})$ through the presence circuit of u_A; otherwise, it ignores u_B's request.

When u_A gets the message from PS, she tries to decrypt it with all her contact keys. When she identifies that the right key is the one corresponding to u_B, she retrieves the entry point E_B of u_B and g^{r_B}. u_A may now decide to communicate with u_B. We note that, if u_A decides to ignore u_B's request for communication, u_B does not know whether or not u_A received the request, or even whether she is online. Should u_A be willing to talk to u_B, she requests her entry E_A to prepare a bridge to E_B for the next epoch. At the beginning of the communication, u_A sends the second part of the Diffie-Hellman key exchange, g^{r_A}, so that the conversation is encrypted with a session key $k_{AB} = g^{r_A r_B}$.

In order to preserve forward secrecy of requests for communications, u_A and u_B update their shared key K_{AB}. In this way, neither of them can be coerced to decrypt an earlier intercepted message. The new key K'_{AB} is computed as: $K'_{AB} = H(k_{AB}, K_{AB})$.

There are some differences between Drac's presence mechanism and Apres [20]. The most important one concerns the way ID's are managed. In Apres, the ID's correspond to relationships (i.e., u_A and u_B share ID_{A+B},) and when u_A connects to the presence server she provides all the ID's she shares with her contacts, plus some extra ones to prevent the server from identifying her by her number of ID's. The main disadvantage of this approach is that, even in the absence of communications, the presence server can see the number of online user relationships. Given a clustered group of contacts who are often online, the presence server may be able to identify the relationships and link the identities between epochs.

3.3 An epoch in Drac

Figure 1(right) shows the adversary's observation of an epoch in which users $\{u_A, \ldots, u_F\}$ are online in Drac using $D = 2$ (for simplicity, we denote user u_X as X in the reminder of this section.) We omit the connections to the presence server in the figure for the purpose of this example. The communication circuits (represented as - - -) created by the users are the following: A-C-D, B-C-B, C-D-E, D-E-F, E-B-C, and F-E-B. The last node in each circuit is the entry point of the initiator of the circuit, e.g., D is E_A, the entry point of A. Besides, a secure link (represented as ▬) has been created between every pair of nodes that route a circuit. Note that there is no link between A and F, because no circuit is relayed through them. However, the adversary can still observe the heartbeat connection between them (represented as $\cdots$.)

In the epoch shown in the figure two communications are taking place. First, F and B are communicating. As both share the same entry point (E_F=E_B=B,) no bridge is created and the communication is fully unobservable for the attacker. A and D are having the second conversation, and they have created a bridge between their entry points E_A=D and E_D=F (represented as ▬.) Although this bridge is distinguishable by the attacker, it is not possible to determine from the observation that A and D are the communication end points. For example, a plausible alternative that would yield the same observation would be that there is only one communication between D and F, and that the circuits are as follows: A-C-D, B-C-D, C-B-C, D-E-F, E-B-E, and F-E-D.

By looking at the circuit connections, the adversary is not able to link users with their entry points because they not only send messages through their own circuit, but also act as "mixes" [6] relaying the traffic of others. Thus, when several circuits traverse a node it is not possible for the adversary to distinguish which input circuit corresponds to which output. As noted in the previous section, all connections must be activated synchronously at the beginning of an epoch. Otherwise, the adversary would see connections ripple down the network when they are created and be able to link users with their entry points. Thus, users must prepare connections in advance during the previous epoch, using the heartbeat channels. For this they have to i) perform key exchanges with all nodes in the circuit to their entry points, ii) find the entry points of the contacts with whom they want to communicate, and iii) instruct their entry point to prepare a bridge to their contact's entry points. We note that this procedure requires users to register their identities for the next epoch when they sign up in the presence server. If two friends want to communicate, they do not need to find their corresponding entry points, but just inform each other through their direct heartbeat connection.

In this paper we restrict our analysis to one epoch, and leave the study of the epoch duration's impact on performance, usability, and security as a subject of future work.

4 Evaluation

4.1 Experimental setup

In order to perform a preliminary analysis of the anonymity and unobservability properties provided by Drac, we have implemented a software simulator.[1] We have tested three topologies for the network graph that describes how users are connected to their friends: small-world networks [35], scale-free networks [2], and random networks. We note that although these topologies do not necessarily resemble real social networks, they are still of theoretical interest as they allow us to study separately the effects of clustering and power law distributions on the security properties of Drac. Experiments with real social network's graphs should be conducted in order to understand the level of protection offered by a potential deployment of Drac.

The simulator generates networks of N nodes (users) with an average of f edges (friends) selected according to the network topology, and f randomly selected contacts. We simulate a single epoch per experiment. First we simulate

[1] The code will be made available by the authors upon request.

the epoch preparation phase, in which each user u_i prepares a communication circuit of depth D hops to her entry node E_i. In addition, users register at the presence server through a heartbeat circuit of depth D_p. We denote the last node in the presence circuit as E_{Pi}. We consider scenarios in which 10% of the N users are communicating with contacts through bridges that connect their respective entry nodes.

Second, we record the observation of the adversary after connections have been activated in the beginning of the epoch. We recall that the adversary observes:

- The heartbeat connections between each pair of users u_i and u_j who share a friendship relationship.
- The connections from the end of the presence circuits (i.e., from the entry nodes E_{Pi}) to the presence server.
- The number of communication circuits routed between each pair of nodes u_i and u_j, which is inferred by looking at the amount of bandwidth used.
- The bridge links B_{ij} that connect the entry nodes E_i and E_j in a communication between two contacts u_i and u_j.

Given the observation of the adversary, in each experiment we randomly select a target user and compute her presence anonymity, communication anonymity, and communication unobservability as described in the next three sections. The results shown in the following sections combine samples from a thousand experiments for each simulation scenario. The baseline simulation scenario is a small-world network of 500 users, with 10 friends and 10 contacts each, and circuit depths D and D_p of three hops. These are the default parameters used in the experiments unless indicated otherwise.

4.2 Anonymity towards the presence server

We first examine the anonymity provided by Drac towards the presence server. Let us consider a user u_A who registers at the presence server with pseudonym IDJ_A in a given epoch. The presence server knows that IDJ_A corresponds to a node that is connecting to it through a presence circuit of depth D_p, which is routed over the heartbeat connections. The last node in this circuit is visible to the presence server, and we denote it by E_{PA}.

In addition, we assume that the adversary can see all the heartbeat connections in the network. We recall that, as explained in Sect. 3, heartbeat connections exist between any two users who share a friendship relationship, and that heartbeat traffic is always the same regardless of whether one, several, or no presence circuits are routed over the heartbeat connection.

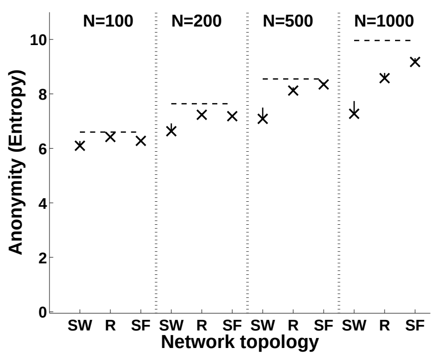
Given this information, IDJ_A may correspond to any of the users u_i connected to E_{PA} by D_p hops in the network of heartbeat channels. Let $\Pr_i[E_{PA}]$ be the probability that user u_i is u_A. We compute $\Pr_i[E_{PA}]$ by enumerating all possible circuits that start at E_{PA} and lead to u_i after D_p hops, taking into account that nodes may appear several times in the paths. Let $\mathcal{P}_i$ be the total number of such paths leading to u_i, $\Pr_i[E_{PA}]$ is computed as:

$$\Pr_i[E_{PA}] = \frac{\mathcal{P}_i}{\sum_{j=1}^{N} \mathcal{P}_j} \ , \ 1 \le i \le N$$

We compute the anonymity of u_A towards the presence server as the entropy H_A of the distribution of $\Pr_i[E_{PA}]$ over all users [11, 30].

$$H_A = -\sum_{i=1}^{N} \Pr_i[E_{PA}] \log_2 \Pr_i[E_{PA}]$$

Figure 2(left) shows the anonymity of Drac towards the presence server for small-world (SW), scale-free (SF), and random (R) networks of sizes between $N = 100$ and $N = 1000$. The dashed horizontal line indicates the maximum achievable anonymity for a network of size N, which is computed as $\log_2 N$. The '**x**' marks the median anonymity for 1000 experiments (each corresponding to an independent target user,) and the vertical line traversing the '**x**' indicates the first and third quartiles of the distribution of anonymity results.



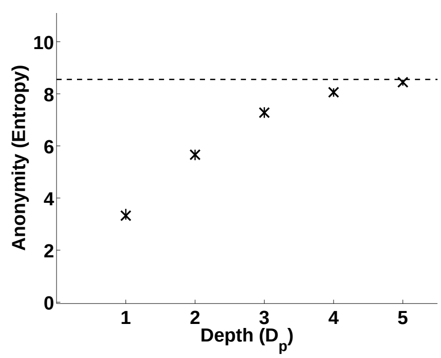


Fig. 2. Anonymity towards the presence server, depending on the network size and topology (left;) and on the depth of the circuits with the baseline parameters (right.)

As we can see in the figure, small-world network topologies provide the lowest anonymity for any network size, and as the network grows their performance becomes worse compared to the other two topologies. This is due to the high degree of clustering of small-world networks, which prevents Drac from taking full advantage of bigger networks: independently of the network size, u_A's connections stay mostly in its own neighborhood. Random networks provide near-optimal anonymity for small network sizes, but as the networks grow the best anonymity performance is shown by scale-free networks. Scale-free networks show a power law degree distribution and grow with preferential attachment. This implies that these networks have some nodes with a very high degree, which grows with the size of the network. High-degree nodes act as mixing hubs that increase anonymity. We choose small-world network topologies in the remaining simulation scenarios in order to test Drac in the least favorable conditions (highly clustered networks) and estimate a lower bound on the anonymity that it offers.

The critical security parameter of the Drac system is the depth of the circuits – which is a system design parameter, as opposed to the network topology or the average number of friends per user. As shown in Figure 2(right) longer presence circuit depths increase the anonymity provided by Drac, at the cost of more communication latency – as the messages need to travel more hops before reaching their destination. In an real-world implementation of Drac, the depth parameter D_p can be tuned to trade bandwidth, latency, and anonymity requirements for any given network, as discussed in Section 5.

4.3 Contact communication anonymity

We recall that communications between *friends* are unobservable to the adversary (see Sect. 3.1.) Let us consider that users u_A and u_F are *contacts* who are communicating in a given epoch. We assume that the bridge connection B_{AF} between their respective entries, E_A and E_F, is observable to the adversary (i.e., we assume that E_A and E_F are not friends.) Note that this is a worst-case scenario, as the bridge B_{AF} may not be distinguishable to the adversary if E_A and E_F are friends, and it is fully unobservable when both users share the same entry; i.e., when $E_A = E_F$.

Starting from the fact that an observable bridge B_{AF} evidences that two contacts are communicating, we evaluate the anonymity of each of the two communicating users separately. This is done by analyzing which users may have constructed a communication path ending, respectively, in entries E_A and E_F. Note that this evaluation does not measure end-to-end anonymity. The reason why it is not straightforward to compute end-to-end anonymity is because in Drac the adversary does not have certainty that a given user is communicating, as opposed to systems that do not use dummy traffic [8, 14, 29]. Information theoretic anonymity metrics [11, 30] operate under the assumption that the adversary knows that user u_A is communicating, and then measure the uncertainty of the adversary in identifying the other end of the communication (i.e., *who talks to whom*.) In contrast, Drac provides communication unobservability properties, implying that the adversary is not certain of *who is talking* in the first place. The next section provides a preliminary analysis of unobservability in Drac. In this section, we evaluate the anonymity of user u_A with respect to an adversary that observes the bridge at E_A.

The analysis methodology is similar to the presence anonymity explained in the previous section. The adversary explores all possible circuit paths of depth D and records the frequency with which each user u_i appears as initiator of the candidate circuit that ends in E_A. The main difference with the computation of presence anonymity is that in this case the adversary can see the number of circuits routed between each pair of nodes (by looking at the amount of bandwidth used.)

Figure 3 shows the results of our simulations for the contact communication anonymity provided by Drac in various network conditions. The left-hand side of the figure compares contact communication anonymity for small-world (SW), scale-free (SF), and random networks (R), of $N = 100$ to $N = 1000$ users. We can see that small-world networks provide the lowest anonymity, while scale-free networks provide the best anonymity of the three topologies, for similar reasons as pointed out in the previous section. We note the anonymity sets in this case are smaller than for the presence circuits. The first factor reducing anonymity is that the adversary has additional information with respect to presence – the number of circuits per link. Another factor that reduces communication anonymity with respect to presence anonymity is that communication links are more sparse than heartbeat links. Users route on average $D+1$ communication circuits – regardless of the size of the network and the average number of friends f – and several circuits may be routed to the same friend. Thus, nodes will maintain fewer communication links with friends than heartbeat connections – and at most, the same.

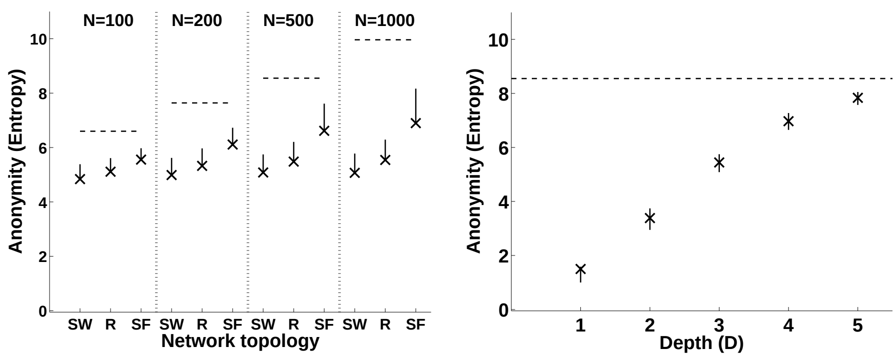


Fig. 3. Anonymity of contact communications towards a global passive adversary, depending on the network size and topology (left;) and on the depth of the circuits with the baseline parameters (right.)

For a constant circuit depth D, Drac provides more anonymity in bigger networks (particularly for scale-free topologies.) We note though that the gap grows between the achieved contact communication anonymity, and the maximum achievable (represented in the figure by dashed horizontal lines) – indicating that longer connection depth would be required to fully take advantage of bigger networks.

In Figure 3(right) we show the variation of anonymity with the security parameter D. As we can see, increasing the depth of the circuits can push the contact communication anonymity of Drac arbitrarily close to the maximum achievable (for a given network size.)

4.4 Contact communication unobservability

In this section we provide a preliminary analysis of the unobservability of communications between contacts provided by Drac. In particular, we look at how well the adversary can correctly guess whether or not user u_A is communicating with a contact.

Let C be the total number of contact communications taking place in a given epoch, and let $\mathcal{E}$ be the set of entry nodes routing bridge connections for those communications. If all communications create a bridge connection, then $|\mathcal{E}| = 2C$; if m pairs of communicating contacts share the same entry node, then $|\mathcal{E}| = 2(C - m)$.

We denote by $\Pr_i[E_j]$ the probability that u_i is the user whose entry node is $E_j \in \mathcal{E}$. We compute $\Pr_i[E_j]$ by enumerating all possible circuits that start at E_j and lead to u_i after D hops (note that $\sum_{i=1}^{N} \Pr_i[E_j] = 1$, but that $\sum_{j=1}^{|\mathcal{E}|} \Pr_A[E_j]$ is not necessarily one.) The probability $\Pr[u_A]$ that u_A is one of the $|\mathcal{E}|$ users communicating with a contact through *any* of the entry nodes in $\mathcal{E}$ is computed as:

$$\Pr[u_A] = \frac{\sum_{j=1}^{|\mathcal{E}|} \Pr_A[E_j] \prod_{k=1,k\neq j}^{|\mathcal{E}|}(1 - \Pr_A[E_k])}{\sum_{j=1}^{|\mathcal{E}|} \Pr_A[E_j] \prod_{k=1,k\neq j}^{|\mathcal{E}|}(1 - \Pr_A[E_k]) + \prod_{k=1}^{|\mathcal{E}|}(1 - \Pr_A[E_k])}$$

We assume that the adversary knows the total number of contact communications C, and can correctly identify *all* bridge connections. We construct the following test to compare Drac to an ideal system that provides perfect unobservability – in which the adversary's best guess is to choose at random:

- First, the adversary computes $\Pr[u_i]$ for all users $u_i, 1 \leq i \leq N$.
- The adversary constructs a set $\mathcal{S}$ with the $2C$ users with higher probabilities, and another set $\mathcal{R}$ with $2C$ randomly chosen users. The set $\mathcal{R}$ models the guess of the adversary for the ideal system.
- We randomly select a user u_A who *is* communicating with a contact, and we test if $u_A \in \mathcal{S}$, and if $u_A \in \mathcal{R}$. We repeat this experiment a thousand times and compare the success rate of the Drac adversary with respect to the success rate of ideal system's (random) adversary.
- We perform the same experiment choosing a user u_Z who is *not* communicating, and compare the success rate of the adversaries of Drac and the ideal system by testing the rate with which $u_Z \in \mathcal{S}$, and $u_Z \in \mathcal{R}$.

Figure 4 shows the results of our tests for a small-world network of 500 nodes in which there are $C = 25$ contact communications, each involving two users. The left-hand side of the figure shows the results of our test for a user u_A who is communicating. As we can see, when connections have depth $D = 1$ the adversary is able to correctly guess that u_A is communicating in more than half of the experiments. When the depth increases to $D = 4$, the advantage of the Drac adversary becomes negligible with respect to the adversary of the ideal system (who guesses at random.)

The right-hand side of the Figure 4 shows the results when testing a user u_Z who is not communicating. As in the previous case, the Drac adversary has an advantage for small circuit depths D, but as D increases her success rate becomes no better than random guessing.

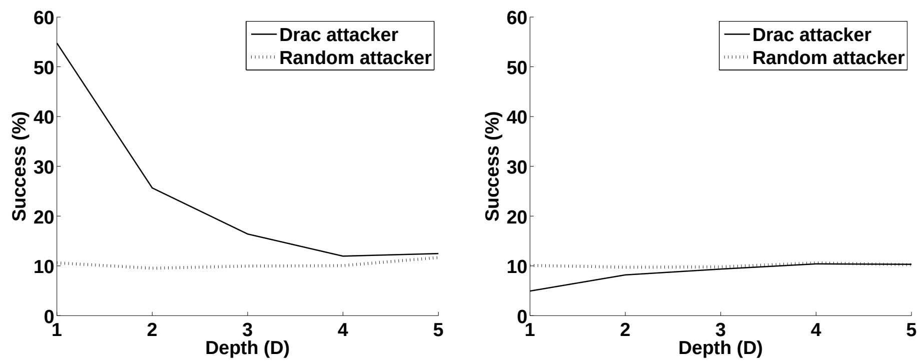


Fig. 4. Comparison of Drac and random adversary success rate in determining that a user is communicating, given that when 10% of the users are communicating. The left-hand side shows the results for a user u_A who *is* communicating, and the right-hand side for a user u_Z who is *not* communicating

5 Discussion

We have so far provided a high-level description of Drac. In this section we discuss some specifics regarding real world performance, trade-offs, overheads and details of the trust model.

Drac is designed to support real-time, low-volume communications such as IM and controversially VoIP. What makes VoIP different from web-traffic is the extreme predictability of the traffic of a VoIP call, despite the tighter requirements to make it useable. A mouth-to-ear delay of more than 50 ms makes voice reflection annoying and a delay of more than 250 ms makes a two-way conversation difficult. As an indication the free Speex[2] codec allows for a sampling rate of 8 kHz and a bit rate of 2.15 kbps (say 3 kbps to take into account some cryptographic overhead). A compressed sample is generated for every 20 ms of speech, with a look-ahead of 10 ms; i.e., 50 packets a second at a sampling rate of 8 kHz, which corresponds to telephony quality. Each node in Drac needs to establish two such channels (2 kpbs) one for incoming and one for outgoing voice, relayed though multiple nodes. This bandwidth is well within the capabilities of contemporary broadband connections, and a dedicated infrastructure could be cheaply built using off-the-shelf routers to support large number of calls (e.g., for

[2] `http://www.speex.org`

a diplomatic network). Since VoIP is delay sensitive, it is reasonable for nodes to discard packets that have been sitting in a queue for longer than 250 ms, indicating that a UDP based implementation [28] would be preferable for Drac. IM traffic has much less stringent requirements, with a couple of messages a second being necessary, each only a few hundreds of bytes long.

As discussed in the evaluation section the length of the path of each circuit is a key security parameter in Drac. This length is also the key contributor to the overhead of the system: $D + 1$ hops per node would mean that the system would consume $N \cdot (D + 1) \cdot 2 \cdot 3$ kbps at any time, even if there are no calls in progress (each node will be expected to carry $(D + 1) \cdot 2 \cdot 3$ kbps on average.) Research suggests that denial-of-service attacks become more likely when paths are longer [4], but the friend-of-a-friend topology used to route makes it less likely that malicious nodes are present on any hop of short paths. Finally, although in this paper we have assumed that D and D_p are constant for all users, there might be some advantages in allowing users to specify their own circuit lengths, as the adversary has to guess the length as well as the exact sequence of nodes in the circuit.

The trust model used in Drac is one of the most novel, and controversial design choices. We argue that relaying communications over a friend-of-a-friend network provides some security advantages. First, it makes denial-of-service and related attacks [4] less likely, and social defenses against sybil attacks can be readily deployed [10]. Moreover, circuit creation does not require a centralized directory and trust infrastructure, which favors network scalability. Drac also avoids network discovery and random sampling attacks present in other peer-to-peer designs [21]. Users have incentives to route traffic [1] for their friends, and the relative stability of a social graph allows for tit-for-tat strategies to penalise free-loading. Finally, the stability of the social graph also invalidates the models of many traffic analysis attacks that assume anonymity sets to contain a random selection of users alongside the target: filtering out the correlated "noise" from those anonymity sets will be nuch more difficult under Drac.

On the down side, paths over social graphs need to be longer to achieve good levels of anonymity, and the length depends on the mixing properties of the social graph [7]. Finally, this design choice exposes the long term social network of the user to the adversary: in many cases the purpose of an anonymity network is hiding exactly those relationships. We have taken the view that long term relations are doomed to be exposed through long term attacks [19]. We instead opt to make those visible to better anonymize casual conversations with unusual contacts. Despite the fact that a relation is visible, actual communication events between friends are designed to be unobservable – a stronger guarantee than the usual anonymity. These choices present a novel trust and protection profile in the anonymity design space.

6 Conclusions

Drac is the first system to be designed to withstand a global passive adversary to protect instant messaging or voice-over-IP conversations. The low-volume and regularity of such traffic makes the use of padding practical, compared with padding high variance connections carrying web-traffic. The overhead of Drac is still high, as users relay circuits over each other all the time. We argue that for IM this overhead is still practical, since the original traffic volumes are low to start with. For VoIP a broadband connection should suffice to participate in Drac, following the current "volunteer" model of Tor [14]. For other deployments a dedicated IP infrastructure could also be reasonable – as some high-profile recent communication security failures illustrate, even some well funded state level actors do not currently have a secure traffic analysis resistant diplomatic network [33]. Our design for Drac could perfectly well fulfill that role.

The design of Drac also borrows features from peer-to-peer designs that suppress the distinction between users and infrastructure, with the novel twist of using a friend-of-a-friend network as a communication and trust backbone. This seriously limits the potential for sybil attacks, provides incentives for relaying traffic, and leads to more stable anonymity sets. All these features require a renewed analysis of past attacks to incorporate them, but we are hopeful they will present advantages over the traditional model of routing over a random graph.

Finally, Drac is fundamentally different from other designs regarding the security properties it provides: it reveals the social graph to the adversary, but provides a stronger property – unobservability of communications. Anonymity is provided when pseudonymous contacts have a conversation. This mixture of properties is likely to be useful in different contexts from the traditional anonymity properties that try to hide relationships against a partial adversary. Our analysis of these properties, albeit preliminary, seems promising but many of the definitions, attacks, and analysis frameworks in the literature will have to be adapted to this new context. This work is a first contribution in this direction.

Acknowledgements.

The authors would like to thank Josep Balasch for editorial comments that improved the readability of the paper. C. Diaz and C. Troncoso are funded by the Fund for Scientific Research in Flanders (FWO). This work was supported in part by the IAP Programme P6/26 BCRYPT of the Belgian State.

References

1. Alessandro Acquisti, Roger Dingledine, and Paul Syverson. On the Economics of Anonymity. In Rebecca N. Wright, editor, *Proceedings of Financial Cryptography (FC '03)*. Springer-Verlag, LNCS 2742, January 2003.
2. Albert-Laszlo Barabasi and Eric Bonabeau. Scale-free networks. *Scientific American*, 288(5):60–69, 2003.

3. Joseph Bonneau, Jonathan Anderson, and George Danezis. Prying data out of a social network. In Nasrullah Memon and Reda Alhajj, editors, *ASONAM*, pages 249–254. IEEE Computer Society, 2009.
4. Nikita Borisov, George Danezis, Prateek Mittal, and Parisa Tabriz. Denial of service or denial of security? In Peng Ning, Sabrina De Capitani di Vimercati, and Paul F. Syverson, editors, *ACM Conference on Computer and Communications Security*, pages 92–102. ACM, 2007.
5. Nikita Borisov and Philippe Golle, editors. *Privacy Enhancing Technologies, 7th International Symposium, PET 2007 Ottawa, Canada, June 20-22, 2007, Revised Selected Papers*, volume 4776 of *Lecture Notes in Computer Science*. Springer, 2007.
6. David Chaum. Untraceable electronic mail, return addresses, and digital pseudonyms. *Communications of the ACM*, 24(2):84–88, February 1981.
7. George Danezis. Mix-networks with restricted routes. In Roger Dingledine, editor, *Privacy Enhancing Technologies*, volume 2760 of *Lecture Notes in Computer Science*, pages 1–17. Springer, 2003.
8. George Danezis, Roger Dingledine, and Nick Mathewson. Mixminion: Design of a type III anonymous remailer protocol. In *IEEE Symposium on Security and Privacy*, pages 2–15. IEEE Computer Society, 2003.
9. George Danezis and Ian Goldberg. Sphinx: A compact and provably secure mix format. In *IEEE Symposium on Security and Privacy*, pages 269–282. IEEE Computer Society, 2009.
10. George Danezis and Prateek Mittal. Sybilinfer: Detecting sybil nodes using social networks. In *NDSS*. The Internet Society, 2009.
11. Claudia Diaz, Stefaan Seys, Joris Claessens, and Bart Preneel. Towards measuring anonymity. In *Designing Privacy Enhancing Technologies, Proceedings of PET'02*, pages 54–68. Springer-Verlag, LNCS 2482, 2003.
12. Claudia Díaz, Carmela Troncoso, and Andrei Serjantov. On the impact of social network profiling on anonymity. In Nikita Borisov and Ian Goldberg, editors, *Privacy Enhancing Technologies*, volume 5134 of *Lecture Notes in Computer Science*, pages 44–62. Springer, 2008.
13. R. Dingledine, N. Mathewson, and P. Syverson. Tor: The second-generation onion router. *Proceedings of the 13th USENIX Security Symposium*, 2, 2004.
14. Roger Dingledine, Nick Mathewson, and Paul F. Syverson. Tor: The second-generation onion router. In *USENIX Security Symposium*, pages 303–320. USENIX, 2004.
15. Nick Feamster and Roger Dingledine. Location diversity in anonymity networks. In Vijay Atluri, Paul F. Syverson, and Sabrina De Capitani di Vimercati, editors, *WPES*, pages 66–76. ACM, 2004.
16. Ian Goldberg. On the security of the tor authentication protocol. In George Danezis and Philippe Golle, editors, *Privacy Enhancing Technologies*, volume 4258 of *Lecture Notes in Computer Science*, pages 316–331. Springer, 2006.
17. Aaron Johnson and Paul F. Syverson. More anonymous onion routing through trust. In *CSF*, pages 3–12. IEEE Computer Society, 2009.
18. Aniket Kate, Gregory M. Zaverucha, and Ian Goldberg. Pairing-based onion routing. In Borisov and Golle [5], pages 95–112.
19. Dogan Kesdogan, Dakshi Agrawal, Dang Vinh Pham, and Dieter Rautenbach. Fundamental limits on the anonymity provided by the mix technique. In *IEEE Symposium on Security and Privacy*, pages 86–99. IEEE Computer Society, 2006.
20. Ben Laurie. Apres - a system for anonymous presence. Technical report.

21. Prateek Mittal and Nikita Borisov. Information leaks in structured peer-to-peer anonymous communication systems. In Paul Syverson, Somesh Jha, and Xiaolan Zhang, editors, *Proceedings of the 15th ACM Conference on Computer and Communications Security (CCS 2008)*, pages 267–278, Alexandria, Virginia, USA, October 2008. ACM Press.
22. Ulf Möller, Lance Cottrell, Peter Palfrader, and Len Sassaman. Mixmaster Protocol — Version 2. Draft, July 2003.
23. Steven J. Murdoch and George Danezis. Low-cost traffic analysis of tor. In *IEEE Symposium on Security and Privacy*, pages 183–195. IEEE Computer Society, 2005.
24. Steven J. Murdoch and Piotr Zielinski. Sampled traffic analysis by internet-exchange-level adversaries. In Borisov and Golle [5], pages 167–183.
25. Luke O'Connor. Entropy bounds for traffic confirmation. Technical Report 2008/365, IACR, October 2008.
26. A. Pfitzmann, B. Pfitzmann, and M. Waidner. ISDN-MIXes: Untraceable Communication with Small Bandwidth Overhead. *Informatik-Fachberichte*, pages 451–463, 1991.
27. Andreas Pfitzmann and Marit Köhntopp. Anonymity, unobservability, and pseudonymity - a proposal for terminology. In Hannes Federrath, editor, *Workshop on Design Issues in Anonymity and Unobservability*, volume 2009 of *Lecture Notes in Computer Science*, pages 1–9. Springer, 2000.
28. Joel Reardon. Improving Tor using a TCP-over-DTLS tunnel. Master's thesis, University of Waterloo, September 2008.
29. Michael K. Reiter and Aviel D. Rubin. Anonymous web transactions with crowds. *Commun. ACM*, 42(2):32–38, 1999.
30. Andrei Serjantov and George Danezis. Towards an information theoretic metric for anonymity. In *Designing Privacy Enhancing Technologies, Proceedings of PET'02*, pages 41–53. Springer-Verlag, LNCS 2482, 2002.
31. Paul Syverson, Gene Tsudik, Michael Reed, and Carl Landwehr. Towards an analysis of onion routing security. In *Design Issues in Anonymity and Unobservability (PET 2000)*, pages 96–114. Springer LNCS 2009, 2000.
32. Carmela Troncoso and George Danezis. The bayesian traffic analysis of mix networks. In Ehab Al-Shaer, Somesh Jha, and Angelos D. Keromytis, editors, *ACM Conference on Computer and Communications Security*, pages 369–379. ACM, 2009.
33. Miron Varouhakis. Greek intelligence and the capture of PKK leader abdullah ocalan in 1999. Studies in Intelligence Vol. 53, No.1 (Extracts), March 2009.
34. Parvathinthan Venkitasubramaniam, Ting He, and Lang Tong. Relay secrecy in wireless networks with eavesdroppers. In *Proceedings of the Allerton Conference on Communication, Control and Computing*, 2006.
35. Duncan J. Watts and Steven H. Strogatz. Collective dynamics of 'small-world' networks. *Nature*, 393:440–442, 1998.
36. Charles Wright, Scott Coull, and Fabian Monrose. Traffic morphing: An efficient defense against statistical traffic analysis. In *Proceedings of the Network and Distributed Security Symposium - NDSS '09*. IEEE, February 2009.
37. Charles V. Wright, Lucas Ballard, Scott E. Coull, Fabian Monrose, and Gerald M. Masson. Spot me if you can: Uncovering spoken phrases in encrypted voip conversations. In *IEEE Symposium on Security and Privacy*, pages 35–49. IEEE Computer Society, 2008.