

# Measuring Anonymity: The Disclosure Attack

Anonymity services hide user identity at the network or address level but are vulnerable to attacks involving repeated observations of the user. Quantifying the number of observations required for an attack is a useful measure of anonymity.

DAKSHI  
AGRAWAL  
IBM Watson  
Research  
Center

DOGAN  
KESDOGAN  
Aachen  
University of  
Technology

**T**he ubiquity of networked computers has made increasing amounts of user information available, threatening user privacy. With access to this information, an adversary can build profiles and obtain sensitive personal information about an individual's political views, health, mobility pattern, social circle, and so on. Users therefore should be able to determine when, how, why, and to what extent their personal information is revealed to others.<sup>1</sup>

The most basic defense against information theft is the careful deployment of cryptographic techniques, which guarantees the privacy of exchanged messages. At the network level, however, a message's address information attributes it to both sender and receiver. Cryptography cannot hide this address information. Consequently, a network operator or intruder can read and collect a user's interactions with which to derive user-specific profiles.

If cryptography is the foundation of digital privacy, then anonymity of communication is its bedrock—without anonymous communication, cryptography-based privacy would be very weak indeed. Every message has two *anonymity sets*—groups of identities assigned to the message's sender or receiver to help hide its identity.<sup>2</sup> Communication acts by anonymity set participants are not attributable to individual participants, and thus remain anonymous. But what if the communication act is repeated multiple times? Will it remain anonymous? A new type of traffic analysis attack—the *disclosure attack*—can reidentify a targeted user's communication partners even if the user employs an anonymity technique.

In this article, we analyze the disclosure attack, measuring its effects against user communications protected by an anonymity technique. We estimate how many

times an attacker must observe a user's anonymous communication acts to find all the user's communication partners. Because the disclosure attack is suboptimal, our analysis provides an upper bound for the number of observations an attacker would need to break an anonymity technique. Our analysis lets anonymity service participants select parameters according to their privacy needs and build a more dynamic and transparent anonymity system.

## Anonymity model

We can measure an anonymity technique's fundamental protection limits on two dimensions:

- *Cryptographic message protection*, including practical metrics such as complexity-theoretic or computational security (which assumes that integer factorization is a difficult problem).
- *Attacker distribution*, such as the *omnipresent attacker*—an attacker able to access and observe all communication lines in a system—and locally present attacker models. Assuming the most distributed attacker provides the most stringent measure of anonymity; a weaker attacker model assumes that the attacker is somewhat restricted and cannot access all communication lines in the system.

To cope with the varieties and combinations of these two dimensions, we must abstract both the internal structure of anonymity techniques and the assumed attacker model. Thus, our formal model assumes that the anonymity technique can build a secure anonymity set against the assumed attacker model. It also assumes that

the attacker is strong enough to identify the anonymity set but can't uncover the protected user.

**Protecting traffic information**

A Mix is an anonymity technique in which an intermediary communication node collects and forwards several messages to hide the identity of the message's senders and receivers.<sup>3,4</sup> A Mix collects a batch of  $b$  messages of equal length from  $b$  distinct senders, discards repeats, changes their appearance (that is, the bit pattern), and forwards the messages to the recipients in a different order. This process hides the relationship between the message sender and its recipient from everyone but the Mix and the message sender. (See the "Mix in Action" sidebar for an example scenario.)

Using  $d$  Mixes in a static order (a *cascade*) or an arbitrary chosen order (a *network*) improves a message's protection against adversaries controlling a Mix. In such cases, using multiple Mixes hides the message sender and recipient from all adversaries in the network who do not control all the Mixes a message passes through. Clearly, Mixes require carefully designed procedures for discarding repeat messages, changing message appearance, and reordering messages in a batch. Furthermore, Mixes should be independently designed and produced and should have independent operators.

**Abstracting Mixes**

Because an adversary can easily determine anonymity sets at the network level, Mixes assume that all network links are observable. Thus, by observing messages to and from an anonymity service, an attacker can determine anonymity sets. Rather than discuss the technical details of any anonymity technique here, we abstract them using the following properties<sup>2</sup>:

- In each anonymous communication, a subset  $A'$  of all senders  $A$  sends a message to a subset  $B'$  of all recipients  $B$ —that is,  $A' \subset A$  and  $B' \subset B$ , as Figure 1 illustrates. In a particular system, the set of all senders  $A$  can be the same as the set of all recipients  $B$ .
- The size of the sender anonymity set is  $|A'| = b$ , where  $1 \leq b \ll |A|$ .
- The size of the recipient anonymity set is  $|B'| = n$ , where  $1 \leq n \ll |B|$  and  $n \leq b$ —that is, several senders can communicate with the same recipient.

The typical values for  $|A|$ ,  $|B|$ ,  $|A'|$ , and  $|B'|$  vary from implementation to implementation and with the environment in which they operate. Stefan Köpsell, Hannes Federrath, and Marit Hansen present an implementation they call Web-Mixes, in which  $(|A|)$  is around 20,000.<sup>5</sup> They don't give typical values for  $|A'|$  for Web-Mixes, but we generally expect  $|A'| < 100$ .

**Mix in Action**

A Mix is an intermediary relay station that hides a message's appearance, including its bit pattern and length. It also hides the temporal relationships (or order) among incoming and outgoing messages. An ideal Mix implementation prevents even an omnipresent attacker (an attacker that observes all incoming and outgoing lines) from linking an incoming message to an outgoing one.

For example, say Alice generates a message  $Message_{Bob}$  to Bob with constant length (add padding bits or split). A sender protocol recursively encrypts the message with public keys  $c_{Bob}$  and  $c_{Mix}$ :  $[[Bob, Message_{Bob}]] := c_{Mix}(RN, Bob, c_{Bob}(Message_{Bob}))$ . This act is similar to enclosing a letter in successive envelopes starting with the recipient. Padding a one-time random number  $RN$  within the encryption will avoid replay attacks.

A Mix hides the message's appearance by decrypting it with a private key  $d_{Mix}$  and strips off the unique random numbers:  $c_{Mix}(RN, Bob, c_{Bob}(Message_{Bob})) \rightarrow Bob, c_{Bob}(Message_{Bob})$ . Using the letter analogy, Mix removes the outermost envelope and finds the inner envelope with the recipient's address. Next, the Mix forwards the inner envelope to the intended recipient after reordering the incoming messages.

To hide a message's order, the Mix collects three messages from distinct users,  $[[Bob, Message_{Bob}]]$ ,  $[[Dave, Message_{Dave}]]$ ,  $[[Cleo, Message_{Cleo}]]$ , and forwards them (after decryption) randomly.

An attacker observing all incoming and outgoing lines from the Mix can only deduce that Alice has communicated with one of the individuals  $\{Bob, Dave, Cleo\}$ .

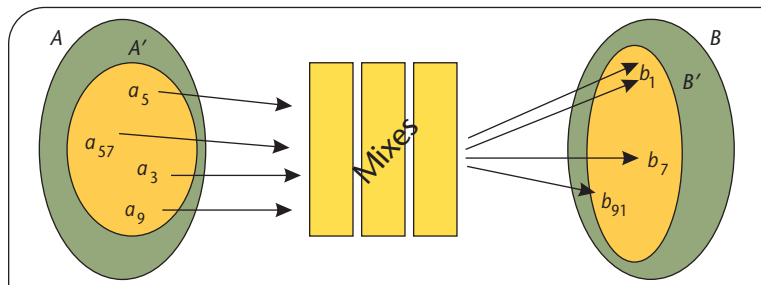


Figure 1. Formal model of an anonymity set. In any anonymous communication, a subset  $A'$  of all senders  $A$  sends a message to a subset  $B'$  of all recipients  $B$ .

**Attack model**

We model attacks by considering a bipartite graph  $G = (A \cup B, E)$  with partite sets  $A$  and  $B$ . The set of edges  $E$  describes the hidden relationships between senders and recipients—that is, a sender  $a$  and a recipient  $b$  are connected by an edge if  $b$  is a recipient of the messages sent by  $a$ . An intruder or adversary must reconstruct the portion of the bipartite graph connected directly to a targeted sender by discovering its edges, as Figure 2 shows.

We assume the attacker in our model notices each anonymous communication act. Each act gives the adversary a randomly selected  $A'$  and  $B'$  (that is,  $A' \subset A$  and

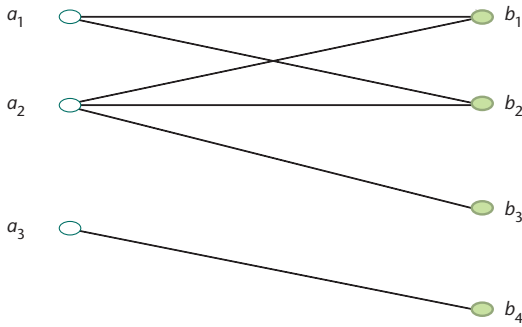


Figure 2. A bipartite graph showing the peer-to-peer relations between senders  $A = \{a_1, a_2, a_3\}$  and recipients  $B = \{b_1, b_2, b_3, b_4\}$ .

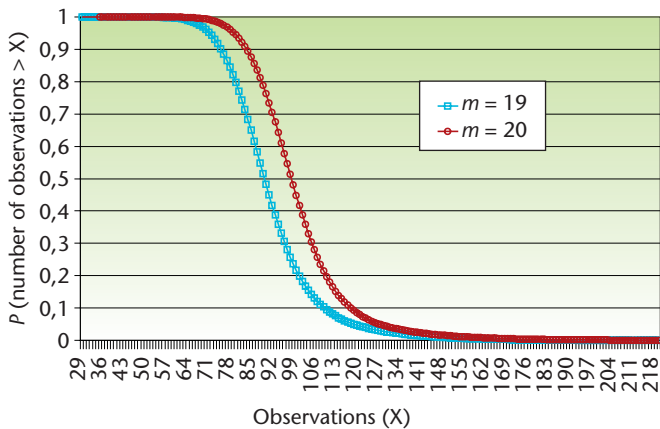


Figure 3. Probability that the number of required observations is greater than  $X$  for  $N = 20,000$  and  $b = 50$ .

$B' \subset B$ ), and a set of possibilities for the hidden edges. The interesting question is how long the attacker must observe anonymity sets  $A'$  and  $B'$  to obtain exact knowledge about a targeted sender's peer partners.

### Traffic analysis attacks

Several researchers, including Mathew Wright and his colleagues,<sup>6</sup> suggest several attacks that exploit an implementation's weakness—for example, in protocol or timing—to compromise Mixes. Our starting point is more general. We assume a technique has no implementation weakness and focus on every anonymity technique's core functionality: the anonymity set. In fact, given a user's communication pattern, we ask whether and how well anonymity sets can hide it.

### Deterministic exact attack

Assuming the attacker has observed all possible combinations of sender and receiver anonymity sets, we can identify all communication relationships among communicating partners.<sup>7</sup> We illustrate this claim using the

bipartite graph in Figure 2 with a sender anonymity size of two. In this environment we have the following cases:

- If  $A' = \{a_1, a_2\}$ , the attacker observes  $\{\langle b_1, b_1 \rangle, \langle b_1, b_2 \rangle, \langle b_2, b_2 \rangle, \langle b_1, b_3 \rangle, \langle b_2, b_3 \rangle\}$ .
- If  $A' = \{a_1, a_3\}$ , the attacker observes  $\{\langle b_1, b_4 \rangle, \langle b_2, b_4 \rangle\}$ .
- If  $A' = \{a_2, a_3\}$ , the attacker observes  $\{\langle b_1, b_4 \rangle, \langle b_2, b_4 \rangle, \langle b_3, b_4 \rangle\}$ .

We can now construct the bipartite graph in Figure 2. We know from case 1 (recipient sets  $\langle b_1, b_1 \rangle$  and  $\langle b_2, b_2 \rangle$ ) that  $b_1$  and  $b_2$  are peer partners of  $a_1$  and  $a_2$ . We also know that  $b_4$  is not a partner of  $a_1$  or  $a_2$ . When we combine these facts with case 2, we can deduce that  $b_4$  should be  $a_3$ 's communication partner, and  $a_1$  has only  $b_1$  and  $b_2$  as its partners. Finally, we know from case 3 that only  $a_2$  has the peer partner  $b_3$ , and  $a_3$  has only one peer partner  $b_4$ . Thus, for the given example, the bipartite graph can be fully identified (the attack's constructive proof is available elsewhere<sup>7</sup>).

### Disclosure attack

A disclosure attack is a probabilistic traffic-analysis attack used to identify all communication partners of a targeted user. Assume Alice uses the system to hide her number of communication partners,  $m$  where  $1 \leq m \ll |B|$ . We assume that the attacker knows  $m$ . We can probabilistically justify this critical assumption. Assume  $\mathbf{m}$  is the real number of partners in a given time period. If the attacker overestimates  $m$  such that  $m = \mathbf{m} - k$  where  $k$  is a positive integer, the adversary cannot apply the disclosure attack's learning phase. Thus, the attacker adjusts  $m$  to  $m := m - 1$  and reapplies the first phase until it succeeds. Of course, the attacker can ensure a correct estimate for  $m$  only with enough observations to apply the disclosure attack. "Enough" depends on the environment's stochastic structure, as Figure 3 shows. Other works analyze in detail how many observations are enough for a correct estimate of  $m$ .<sup>8,9</sup>

To perform the attack, the attacker simply records all recipient sets  $B'$  that include one of Alice's peer communication partners. For simplicity, we increase time  $t$  by one each time Alice sends a message. Thus, when Alice communicates for the first time,  $t = 1$ ; when she communicates for the second time,  $t = 2$ ; and so on. We denote the recipient set at time  $t$  by  $B'_t = \{b_t^1, \dots, b_t^m\}$ .

A disclosure attack has a *learning phase* and an *excluding phase*.

**Learning phase.** In this phase, the attacker's task is to find  $m$  mutually disjoint recipient sets—that is, each set has only one peer partner of Alice—by observing Alice's incoming and outgoing messages.

In technical terms, for mutually disjoint recipient sets  $(B_{j_1}', \dots, B_{j_m}')$ ,  $B_{j_x}' \cap B_{j_y}' = \emptyset$  if  $x \neq y$ . After the learning phase, the attacker can be sure that each set  $B_{j_x}'$  contains only one of Alice's peer communication partners. (If any

of the mutually disjoint sets has more than one of Alice's peer partners, the total number of peer partners would exceed  $m$ , contrary to our assumptions.)

For example, assume that Alice communicates frequently with Bob and Dave using the Mix technique (See the "Mix in Action" sidebar). Assume that the attacker knows that Alice communicates with two people and observes three recipient sets:  $B_1 = \{\text{Bob, Dave, Cleo}\}$ ,  $B_2 = \{\text{Henry, Dave, Cleo}\}$ , and  $B_3 = \{\text{Bob, Julia, Kim}\}$ . Because  $B_2 \cap B_3 = \emptyset$ , the attacker chooses  $\{B_2, B_3\}$  as the next step's basis sets. Because the attacker only notes sets in which Alice is a sender, each basis set contains at least one of Alice's peer communication partners. Therefore, two basis sets together contain both of Alice's communication partners.

**Excluding phase.** The attacker's task in this phase is to observe new recipient sets until all Alice's nonpeer partners are excluded from the basis sets. To do this, the attacker intersects a new recipient set with the basis sets. Three possible outcomes exist:

- *No intersection.* Contrary to our assumption, because none of the peer communication partners in the basis sets appear in the recipient set.
- *Intersection only with one basis set.* The attacker knows that Alice's peer partner must be in the intersection (excluding act).
- *Intersection with more than one basis set.* The attacker cannot tell which intersection contains Alice's peer partner.

In other words, to refine the recipient sets ( $B_{j_1}'$ , ...,  $B_{j_m}'$ ), the attacker takes a new recipient set  $B'$  that intersects only one prior recipient set—that is,  $B' \cap B_{j_x}' \neq \emptyset$  and  $B' \cap B_{j_y}' = \emptyset$  for all  $x \neq y$ , and refines  $B_{j_x}'$  to  $B_{j_x}' \cap B'$ . The refinement process only excludes Alice's nonpeer partners from  $B_{j_x}'$ . The attacker continues the refinement process until each of the sets  $B_{j_1}'$ , ...,  $B_{j_m}'$  contains only one user. Because we exclude only Alice's nonpeer partners in the excluding stage, the remaining  $m$  users in  $B_{j_1}'$ , ...,  $B_{j_m}'$  are clearly Alice's communication partners.

## Disclosure attack analysis

The disclosure attack is an NP-complete problem. The proof, detailed elsewhere,<sup>10</sup> is technical and involves showing that the learning phase of the disclosure attack is equivalent to the well-known NP-complete Clique problem.<sup>11</sup>

To analyze the disclosure attack, we use a formal model, described next. We make certain assumptions that simplify analysis and exposition without affecting the number of observations required by the disclosure attack. These assumptions include the following:

- $N$  system users send messages to each other. The number of senders (the batch size) in each anonymous com-

munication is  $b$ , where  $1 < b \ll N$ . (We assume constant batch size for mathematical convenience.) Two senders can have the same recipient in a batch. Thus, the size  $n$  of the recipient anonymity set fulfills the condition  $n \leq b$ .

- Alice is a sender using the system to hide her  $m$  communication partners.
- Alice chooses a communication partner in each communication uniformly among her  $m$  partners, while the other senders choose their communication partners uniformly among all  $N$  recipients. (This assumption ensures that all system users are equally likely to be in a recipient set as Alice's nonpeer partners, and is not required for the disclosure attack.)

We use several strategies to estimate the average number of observations required to complete the disclosure attack's learning and excluding phases.

## Simulation results

To determine the statistical characteristics of the number of observations a disclosure attack requires, we developed a disclosure attack simulator. Because the learning phase is an NP-complete problem, writing such a simulator is tricky. We use enhanced backtracking algorithms by transforming the learning phase problem into a binary constraint satisfaction problem.<sup>12</sup> We supplement this transformation with knowledge about the observations, significantly reducing the search space. For example, we use the fact that any recipient set containing two of Alice's peer partners doesn't belong in the learning phase output. More details on the simulator are available elsewhere.<sup>9</sup>

To illustrate the simulation results, we consider an anonymity-providing system with the parameters  $N = 20,000$ ,  $b = 50$ , and  $m = 20$ . For each simulation in this article, we computed a 95 percent confidence interval for the mean and stopped repeating the simulations when the confidence interval was no more than 5 percent of the computed mean. Figures 4 and 5 shows the average number of observations required at the learning and excluding stages as a function of  $N$ ,  $b$ , and  $m$  as the two other parameters remain fixed.

As Figures 4 and 5 show, for both phases of the disclosure attack, the required number of observations rises sharply as parameters  $m$  and  $b$  increase above a threshold and parameter  $N$  decreases below a threshold. Before this sharp rise, the number of observations required at both stages is too low—often fewer than 50—to provide adequate anonymity. Furthermore, before the sharp rise, the number of required observations is independent of one or more system parameters  $N$ ,  $b$ , and  $m$ . In these *weak operational regions* a system operator partially loses the ability to control the anonymity the system provides by varying system parameters. Clearly, a user wants to avoid weak operational regions and

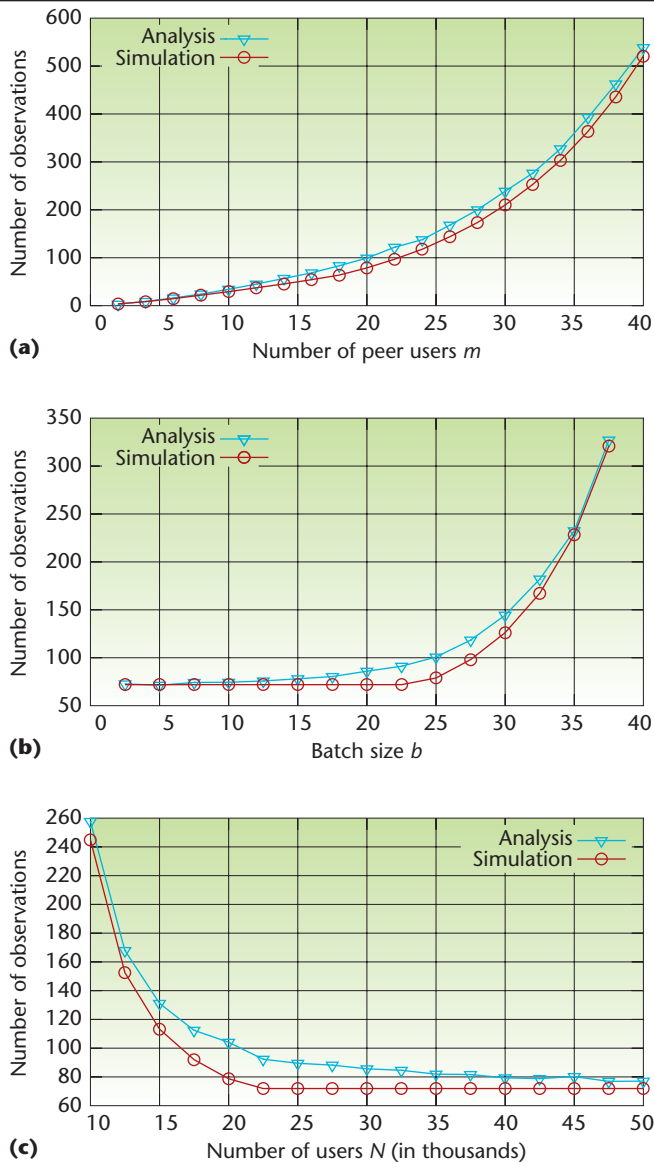


Figure 4. Number of observation required for the learning phase when (a)  $N = 20,000$  and  $b = 50$ , (b)  $N = 20,000$  and  $m = 20$ , and (c)  $b = 50$  and  $m = 20$ .

choose system parameters  $N$ ,  $b$ , and  $m$  to ensure a certain level of anonymity.

To measure anonymity using a disclosure attack, we evaluated an anonymous Web-surfing scenario. We parameterized the number of a user's favorite Web sites by  $m$  and asked the user to choose Web sites (that is, recipient sets) according to a Zipf distribution.<sup>9</sup> We chose Zipf because empirical data has shown that it works well with Web site popularity measurements.<sup>13</sup> Using the results of this evaluation, users can vary  $m$  to choose their anonymity level.

Our simulations show that an attacker mounting a disclosure attack faces two bottlenecks in the learning phase:

- A *nonpeer* bottleneck results from the requirement that nonpeer recipients in  $m$  sets are disjoint.
- A *peer* bottleneck results from the requirement that all  $m$  peer recipients should occur in  $m$  sets.

For a given set of system parameters, one of the two requirements is statistically more difficult to satisfy than the other. Consequently, the bottleneck corresponding to the more difficult requirement prolongs the learning phase most.

The weak operational region boundaries (the  $x$ -axis values in Figure 4 where the curves start their sharp rise) coincide with the parameter values, which correspond to a transition from a nonpeer bottleneck to a peer bottleneck. The same is true for the excluding phase. Therefore, we can identify weak operational regions by finding the transition between the two bottlenecks. We've derived analytical formulas for this transition, letting us identify weak operational regions without simulation.

### Learning phase analysis

To estimate the number of observations needed for the disclosure attack's learning phase, the attacker must collect  $m$  mutually exclusive recipient sets in which Alice is a participant. To avoid unnecessary repetition, we assume that

- Alice participates in all batches, and
- Only one of Alice's peer partners is in the recipient sets of all batches.

We can safely make the first assumption because the attacker is only interested in the batches in which Alice participates. We make the second assumption and provide a correction factor to account for the possibility that another sender could send a message to one of Alice's peer partners, making a batch useless for the learning phase's purpose.

Consider a brute-force attacker who keeps the collection of all groups of mutually exclusive recipient sets. (The brute-force attacker is also the most powerful attacker.)

Let  $G^k(l)$ ,  $k = 1, 2, \dots, l = 1, 2, \dots, k$ , be the collection of all groups of  $l$  mutually exclusive recipient sets after observing  $k$  batches. An attacker observing the  $(k + 1)$ th recipient set would obtain  $G^{k+1}(l)$  as follows: for  $l = 1, 2, \dots, k$ ,  $G^k(l) \subset G^{k+1}(l)$  as the mutual exclusivity of the recipient sets in  $G^k(l)$  is not affected by new observations.

Furthermore, the attacker would see if the new observation is mutually exclusive with any group in  $G^k(l - 1)$ . We can add the new observation that is mutually exclusive with a group in  $G^k(l - 1)$  to the group, thereby increasing the group's cardinality to  $l$ , and add the modified group to  $G^{k+1}(l)$ . This process continues until the attacker obtains  $m$  mutually exclusive recipient sets—in other words, the learning phase ends when  $G^k(m) \geq 1$  for some  $k$ .

This reasoning gives rise to two heuristic equations:

$$|G^{(k+1)}(l)| = |G^{(k)}(l-1)| \pi(l) + |G^{(k)}(l)| \quad \text{and} \quad (1)$$

$$|G^{(k+1)}(1)| = |G^{(k)}(1)| + 1, \quad (2)$$

where  $|A|$  denotes set  $A$ 's cardinality, and  $\pi(l)$  is the probability that a new recipient set will be mutually exclusive with a group of  $(l-1)$  mutually exclusive recipient sets. These equations are heuristic rather than precise because they involve several approximations.

To derive Equation 2, we approximate that the probability of seeing the same recipient set in two different batches is zero. The second approximation is more subtle: The actual evolution of the number of groups in  $G^k(l)$  is probabilistic—that is,  $|G^k(l)|$  will vary from one trial to another, yet Equation 1 computes  $|G^k(l)|$  deterministically. The approximation here involves replacing certain random variables with their expected values.

A more technical analysis of these assumptions is available elsewhere.<sup>10</sup> In this article, we compare the results of these equations to the simulation results to prove our assumptions' validity.

Using Equations 1 and 2 recursively, we calculate the value of  $k$  for which

$$\arg \min_k |G^k(m)| \geq 1.$$

Let  $\mathcal{K}$  denote this value of  $k$ . Recall that this estimate only considers batches in which only one of Alice's peer partners participates. The probability that other  $(b-1)$  senders in a batch do not send a message to one of Alice's peer partners is  $((N-m)/N)^{(b-1)}$ . Therefore, the estimate of the required number of batches for the learning phase in which Alice participates is

$$\mathcal{K} / \left( \frac{N-m}{N} \right)^{b-1}.$$

We can improve this estimate by incorporating the necessary condition that before the learning phase ends, the attacker must observe all Alice's peer partners in the recipient sets. Observing Alice's  $i$ th peer partner requires on average  $m/(m-i+1)$  batches; thus we obtain our final estimate for the learning phase from

$$T_l \approx \max \left[ \frac{\mathcal{K}}{\left( \frac{N-m}{N} \right)^{b-1}}, \sum_{i=1}^m \frac{m}{m-i+1} \right]. \quad (3)$$

Figure 4 compares the analytical result obtained from Equation 3 with the simulation results. For all three parameters, the analytical estimate is only slightly less than the actual number of observations required for the learning stage. The most prominent differences between the estimates and the simulation results occur around the knee of the curves, which coincide with the term change in Equation 3. The second term in the equation provides

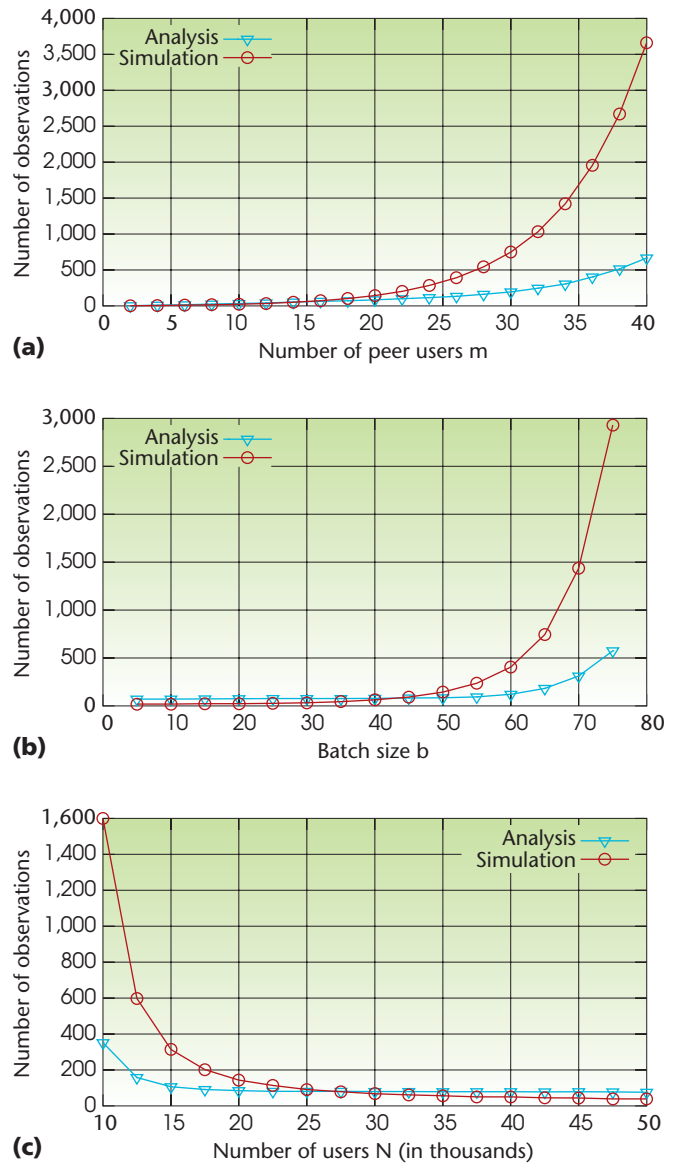


Figure 5. Number of observation required for the excluding phase when (a)  $N = 20,000$  and  $b = 50$ , (b)  $N = 20,000$  and  $b = 20$ , and (c)  $b = 50$  and  $m = 20$ .

the number of observations for the flat part of the curve, and the first term provides the number of observations for the rising part. During the transition from the first term to the second, our estimate becomes less accurate.

### Excluding phase analysis

An attacker starts the excluding phase with the  $m$  mutually exclusive recipient sets  $O_1, O_2, \dots, O_m$  obtained in the learning phase. The sets  $O_1, O_2, \dots, O_m$ , each have cardinality  $b$ . (We assume that in a batch, two different senders do not send messages to the same receiver, an acceptable assumption in an open system.<sup>8</sup>) In the excluding phase, if the attacker finds a recipient set  $R_l$  that inter-

**Table 1. Threshold values of  $N$ ,  $m$ , and  $b$  for three typical cases.**

PARAMETERS	THRESHOLD		
	$N$	$M$	$b$
$N = 20,000, m = 20, b = 50$	15,718	28	56
$N = 400, m = 10, b = 10$	369	11	10
$N = 20,000, m = 40, b = 100$	105,602	91	138

sects with only one set  $O_l$  among  $O_1, O_2, \dots, O_m$ , the attacker replaces  $O_l$  with the intersection of  $O_l$  and  $R_t$ . This process reduces the cardinality of  $O_l$ , and in the end, each of the sets  $O_1, O_2, \dots, O_m$  contains only one recipient. These recipients are Alice's  $m$  peer partners.

We model the excluding phase as a discrete stochastic process with state  $X = (O_1, O_2, \dots, O_m)$ . Let  $X_t$  denote the process state after the attacker observes the  $t$ th batch. Because the state  $X_t$  only depends on the previous state  $X_{t-1}$  and the  $t$ th batch,  $X_t$  is a Markov process.

If we assume that the nonpeer recipients in a new batch occur uniformly across all  $N$  users, the process  $Y$  given by  $Y = (|O_1|, |O_2|, \dots, |O_m|)$  also becomes a Markov process. This is because the recipient sets  $O_1, O_2, \dots, O_m$  are mutually exclusive at the end of the learning phase and remain mutually exclusive during the excluding phase. The mutual exclusivity of sets  $O_1, O_2, \dots, O_m$  and uniformly occurring nonpeer recipients in a new batch render the recipient's individual identities unimportant for determining the sets' sizes. In other words, the mutual exclusivity of  $O_1, O_2, \dots, O_m$  and uniformly occurring nonpeer recipients in a batch imply that the number of recipients in the set  $O_l$ , where  $1 \leq l \leq m$ , after observing the  $t$ th batch depends only on the number of recipients in these sets before observing the  $t$ th batch. Because we're only interested in the average number of observations required to reach the state  $Y = (1, 1, \dots, 1)$ , it suffices to analyze the Markov process  $Y$ .

In theory, recognizing  $Y$  as a Markov process lets us obtain an upper bound on the average number of observations required for the excluding phase. However, the large size of the state-space of  $Y$  given by  $b^m$  makes this statistic difficult to compute. The state space of  $Y$  is large even for moderate values of  $b$  and  $m$  ( $b = 25$  and  $m = 10$ , for example).

To efficiently compute the required upper bound, we look at the number of recipients in the first recipient set,  $Y_1 = |O_1|$ , and stipulate that the total number of nonpeer recipients in other recipient sets is given by  $\gamma = (m-1)(E(Y_1) - 1)$ . Under this stipulation, the state space of  $Y_1$  only has  $b$  states, and we can compute the transition probability of  $Y_1$  relatively easily. Using the standard formulas for Markov processes and the transition probabilities, we can compute the average number of observations required to make  $Y_1 = 1$ . Computation details are available elsewhere.<sup>10</sup>

Figure 5 compares the results from the methods outlined here with the simulation results. These graphs show that once the number of observations start their sharp rise, our analysis overestimates the number of observations required for the excluding phase. This overestimation is a result of discarding observation sets during the analysis if they are not immediately useful. In a disclosure attack, the attacker keeps an observation set for future use. In all our tests, our estimates fall within one order of magnitude of the numbers obtained by simulations. At present, we have no rigorous proof of the tightness of the bound provided by our estimates. Different extreme cases we have tried, however, give us confidence that we can use our estimates to compute an upper bound on an anonymity system's protection limit.

### Weak operational region analysis

As discussed earlier, to analytically compute the weak operational region, we must identify the point at which the disclosure attack transitions from a nonpeer bottleneck to a peer bottleneck in the learning phase.

Consider a nonpeer bottleneck caused by Alice's nonpeer recipients. After the attacker has collected  $(i-1)$  mutually exclusive recipient sets, Alice has  $(i-1)(b-1)$  nonpeer recipients in these sets. Therefore, her choices for  $(b-1)$  nonpeer recipients in the  $i$ th mutually exclusive recipient set decreases from  $(N-m)$  to  $(N-m-(i-1)(b-1))$ . The impact of previously observed nonpeer recipient sets is greatest when the attacker wants to find the  $m$ th mutually exclusive recipient set. Then, the number of eligible candidates decreases from  $(N-m)$  to  $(N-m-(m-1)(b-1))$ , and the probability of finding  $(b-1)$  unobserved nonpeer recipients becomes

$$\frac{(N-m-(m-1)(b-1))^{(b-1)}}{(N-m)^{(b-1)}} \quad (4)$$

For a peer bottleneck, the attacker would find it difficult to obtain a recipient set with a previously unobserved peer partner of Alice. An attacker with  $(i-1)$  mutually exclusive recipient sets has already observed  $(i-1)$  peer partners of Alice, and the probability of observing a previously unobserved peer partner is  $(m-i+1)/m$ . Clearly, the bottleneck is most severe when the attacker tries to obtain Alice's  $m$ th peer partner. In that case, the probability of observing the  $m$ th peer-partner is only  $1/m$ .

We obtain the boundary of the transition region by equating the probability of each bottleneck. Specifically, we set

$$\frac{(N - m - (m - 1)(b - 1))^{(b-1)}}{(N - m)^{(b-1)}} = \frac{1}{m}. \quad (5)$$

By fixing two parameters in Equation 5, we can compute the threshold value of the third parameter when the disclosure attack goes from one bottleneck to another in the learning phase. Table 1 shows the threshold values for three typical cases. Comparing these thresholds with the simulation results (given for the graph in Figure 4a) shows that Equation 5 precisely estimates the transition region boundary.

In this article, we demonstrate how to estimate fundamental limits on the anonymity provided by an anonymity technique. Our future work will take these limits and implement an intelligent assistant that helps users control and protect their privacy. □

## Acknowledgments

This article benefited tremendously from comments by Dieter Rautenbach, Stefan Penz, Lexi Pimenides and discussion with them. Stefan Penz performed the disclosure attack implementation.

## References

1. OECD Guidelines on the Protection of Privacy and Transborder Flows of Personal Data, Organization for Economic Cooperation and Development, 1981.
2. A. Pfitzmann and M. Köhntopp, "Anonymity, Unobservability, and Pseudonymity—A Proposal for Terminology," *Designing Privacy Enhancing Technologies: Proc. Int'l Workshop Design Issues in Anonymity and Observability*, LNCS 2009, Springer-Verlag, 2000, pp. 1–9.
3. D. Chaum, "Untraceable Electronic Mail, Return Addresses, and Digital Pseudonyms," *Comm. ACM*, vol. 24, no. 2, 1981, pp. 84–88.
4. G. Danezis, R. Dingledine, and N. Mathewson, "Mixminion: Design of a Type III Anonymous Remailer Protocol," *Proc. IEEE Symp. Security and Privacy*, IEEE CS Press, 2003, pp. 28–41.
5. S. Köpsell, H. Federrath, and M. Hansen, "Erfahrungen mit dem Betrieb eines Anonymisierungsdienstes," (Experiences with the Operation of an Anonymity Service) *Datenschutz und Datensicherheit*, vol. 27, no. 3, 2003 (in German).
6. M. Wright et al., "An Analysis of the Degradation of Anonymous Protocols," *ISOC Network and Distributed Systems Security Symp.*, 2002.
7. D. Kesdogan and D. Rautenbach, *Towards Optimal Disclosure Attack*, tech. report, Computer Science Dept. Informatik IV (Communication and Distributed Systems), Technical Univ. of Aachen, 2003.
8. D. Kesdogan, D. Agrawal, and S. Penz, "Limits of Anonymity in Open Environments," *Proc. 5th Int'l Workshop Information Hiding*, LNCS 2578, Springer-Verlag, 2002.
9. S. Penz, *Security Analysis and Evaluation of Anonymity Techniques in Open Environments*, master's thesis, Computer Science Dept. Informatik IV (Communication and Distributed Systems), Technical Univ. of Aachen, 2002 (in German).
10. D. Agrawal, D. Kesdogan, and S. Penz, "Probabilistic Treatment of Mixes to Hamper Traffic Analysis," *Proc. IEEE Symp. Security and Privacy*, IEEE CS Press, 2003, pp. 16–27.
11. M.R. Garey and D.S. Johnson, *Computers and Intractability*, Freeman, 1979.
12. R. Dechter and D. Frost, *Backtracking Algorithms for Constraint Satisfaction Problems*, tech. report, Information and Computer Science (ICS) Dept., Univ. of Calif., Irvine, Sept. 1999.
13. B. Krishnamurthy and J. Rexford, *Web Protocols and Practice: HTTP/1.1, Networking Protocols, Caching, and Traffic Measurement*, Addison-Wesley, 2001.

**Dakshi Agrawal** is a research staff member at the IBM Watson Research Center. His research interests include security and privacy of digital communications and policy-based computing. He has a bachelor's degree from the Indian Institute of Technology, Kanpur, a master's from Washington University in St. Louis, and a doctorate from the University of Illinois, Urbana-Champaign, all in electrical engineering. Contact him at [agrawal@us.ibm.com](mailto:agrawal@us.ibm.com).

**Dogan Kesdogan** is an assistant professor in the Computer Science Department at Informatik 4 (Communication and Distributed Systems) at the Technical Univ. of Aachen. His research interests include communication networks, security, and privacy. He has a doctorate in computer science from the Aachen University of Technology. Contact him at [kesdogan@acm.org](mailto:kesdogan@acm.org).