Data Caching for Enhancing Anonymity

Rajiv Bagai and Bin Tang
Department of Electrical Engineering and Computer Science
Wichita State University
Wichita, Kansas 67260–0083, USA
Email: {rajiv.bagai, bin.tang}@wichita.edu

***Abstract*—The benefits of caching for reducing access time to frequently needed data, in order to improve system performance, are already well-known. In this paper, a proposal for employing data caching for increasing the level of anonymity provided by an anonymity system is presented. This technique is especially effective for user sessions containing bidirectional communication, such as anonymous web browsing. A framework is first constructed for capturing the effect of attacks on anonymity systems that have the ability to serve some incoming user requests from their cache. A system-wide metric is then presented for measuring the anonymity provided by such systems. It is shown that the anonymity level of such systems rises with the amount of data caching performed by them. This behavior is illustrated in an example threshold mix network.**

***Index Terms*—Anonymity Metrics, Caching, Privacy.**

I. Introduction

Applications of the Internet that require anonymous communication are becoming increasingly visible and their need is expected to grow significantly. Anonymous web surfing, chatting, e-mailing and e-voting are just some examples of tasks that require anonymity. As the underlying architecture of the Internet was not designed with such applications in mind, it has become necessary to develop special systems that bridge this gap and provide the needed anonymity.

Two fundamental directions in which research on anonymity systems is usually performed are:

- *System architecture*, which includes designs of anonymity systems, various techniques and strategies that underlie them, study of attacks they are vulnerable to, etc., and
- *Anonymity metrics*, which includes methods for measuring the level of anonymity provided by systems, evaluating and comparing robustness of systems against given attacks, etc.

The work described in this paper spans both of the above directions.

Our architectural contribution is a proposal to adopt the technique of data caching in an anonymity system that, in addition to resulting in the expected performance gain due to caching, has the benefit of enhancing the level of anonymity provided by the system. While most currently popular techniques for providing anonymity, such as message routing via proxies and onion encryption, do so at the expense of *increasing* message latency, data caching has the advantage of providing anonymity in addition to *decreasing* latency.

As our metric contribution, we develop a new metric for measuring the anonymity level of a system equipped with caching abilities, and show that the greater the amount of caching performed by the system, the higher its resulting anonymity level. We also illustrate this phenomenon of rising anonymity in an example threshold mix network. To the best of our knowledge, our work is the first attempt at quantifying the anonymity gains resulting from caching.

A. Related Work

The most popular architecture to date for anonymity systems is a mix network, proposed by Chaum [1], which is a collection of proxy machines that jointly relay messages between senders and receivers connected to the network. TOR [6] and Mixminion [4] are well-known examples of real systems based on this architecture.

Not much work has yet been done, however, on employing data caching as a technique for achieving anonymity, despite the recognition made by Shubina and Smith [14] of the potential of caching for this purpose. They proposed providing anonymity by a proactive procurement of web content from available caches, such as Google cache, but since then there has not been any follow-up work in this direction, such as an analysis of anonymity gains thus achieved or a study of caching strategies for maximizing anonymity, etc. Kim and Kim [9] also noted the possibility of using caching for achieving anonymity, but their work focuses on server anonymity in unstructured P2P systems. In this paper, we study caching performed by mix-based anonymity systems.

The subject of anonymity metrics, on the other hand, has enjoyed much research attention. Chaum [2] suggested usage of the size of the set of possible users, within which a particular user blends in, as a measure of anonymity provided to that user. Serjantov and Danezis [11] proposed an entropy-based measure that takes into account the probabilities assigned by an attacker to different users for being the sender (or receiver) of a particular message. Diaz, Seys, Claessens and Preneel [5] gave a better entropy-based metric that can be used to compare systems with different number of users. Tóth, Hornák and Vajda [15] argued to additionally consider as a metric, the maximum probability an attacker can assign to any user. Kelly *et al.* [8] is a good survey of well-known work in the area of anonymity metrics.

All approaches mentioned above for measuring anonymity are from the point of view of a single user of the system or message. In contrast, Edman, Sivrikaya, and Yener [7] proposed a framework for measuring the anonymity provided

by a system as a whole. Their method is based upon the size of the set of possible input-output pairs of messages that any particular input-output message pair blends in.

In this paper, we first develop a system-wide metric that is a generalization of this metric for systems that perform data caching. We then show that the anonymity level of such a system rises with increased caching, and illustrate this phenomenon in an example threshold mix network. Such networks were first proposed by Serjantov, Dingledine and Syverson [12].

B. Paper Outline

The rest of this paper is organized as follows. Section II gives an overview of the approach of Edman *et al.* [7] for measuring anonymity. It presents the underlying system model for which their metric is constructed, some examples of attack results on such systems, and the rationale for their metric. Section III presents our approach that first removes some of the limitations of the system model of [7] and then equips the system with an ability to cache data. This section then presents our generalized anonymity metric for systems that employ caching. It goes on to showing that the level of anonymity of a system enjoys a monotonically increasing relationship with the amount of caching performed in it. Section IV highlights this relationship in an example threshold mix network. Finally, Section V contains conclusions from our work and gives several directions for future work.

II. A System-Wide Anonymity Metric

In this section we give an overview of the metric proposed by Edman, Sivrikaya, and Yener [7] for the level of anonymity provided by an anonymity system. Their metric is not specific to any particular architecture of the underlying system, such as one organized as a common network of mixes introduced by Chaum [1], but is applicable to any system that attempts to provide anonymity to communication transmitted via it. Moreover, the metric gives a *system-wide* measure of the effectiveness of the anonymity system, unlike most other approaches that typically measure the anonymity provided from the perspective of a *single* user or message in the system.

A. The Underlying Model

Let S be the set of input messages observed by a passive observer having entered an anonymity system, and T be the set of output messages observed by the same observer having exited from that system. We assume that every input message eventually appears as an output message, and that the anonymity system does not generate any output messages by itself. Thus, there is a one-to-one correspondence between S and T, and the sizes of these sets are identical. We let m denote their size, i.e. $|S| = |T| = m$.

The main goal of the anonymity system is to prevent the observer from determining the underlying correspondence between S and T. It may attempt to achieve that goal by employing a number of techniques such as:

- modifying message encoding by encryption/decryption to prevent message bit-pattern comparison by the observer,
- outputting messages in an order other than the one in which they were received to prevent sequence number association by the observer, etc.

The maximum anonymity this system can strive to achieve is when for any particular output message in T, each of the input messages in S is a possible candidate to be the one that exited the system as that message in T. We depict this situation by the complete bipartite graph $K_{m,m}$ between S and T, as shown in Figure 1(a). Any edge $\langle s_i, t_j \rangle$ in this graph indicates that the incoming message s_i could possibly have been the outgoing message t_j.

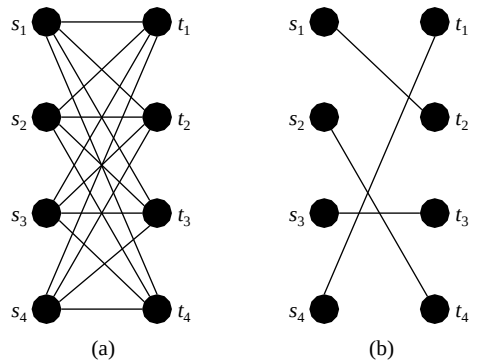

Fig. 1. (a) Complete anonymity. (b) An instance of no anonymity.

An attacker, on the other hand, attempts to eliminate as many edges as possible from the complete bipartite graph of Figure 1(a), due to their infeasibility concluded from the attack. After a completely successful attack, for each outgoing message t in T, the attacker would have identified exactly one possible incoming message that could have exited the system as t. In other words, the attacker would have obtained a *perfect matching* between the input and output messages of the system. In this case, the system is thus considered to provide no anonymity. There are $m!$ possible perfect matchings between S and T, an arbitrary one of which is shown in Figure 1(b).

B. Some Attack Examples

Figures 1(a) and 1(b) correspond to the two extreme situations, namely complete anonymity and no anonymity at all. In general, after having detected some input-output pairings as infeasible, an attack would result in a bipartite graph that lies somewhere in between these two extreme ends. Let this graph of possible input-output pairings resulting from an attack be called the *candidacy graph* of that attack. Exactly which edges of the complete bipartite graph are missing from an attack's candidacy graph will depend upon how much information is available to that attack.

For example, if messages passing through an anonymity system are not padded to become of the same size, then message sizes can clearly be used to rule out input-output message pairings that are of different sizes. Many anonymity systems therefore pad their messages to become of equal size.

However, systems in which all messages are of the same size are then constrained to have some maximum route length for messages, due to the fact that each message (upon leaving the sender) contains in it the addresses of all proxy servers it will go via. Serjantov and Danezis [11] present an attack that exploits knowledge of this maximum route length. The example of Figure 2 shows how this attack can render certain input-output pairings as infeasible.

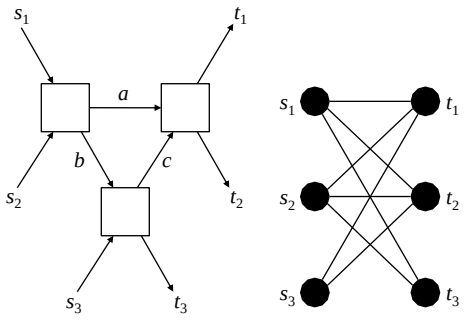

Fig. 2. (a) Route length attack. (b) Candidacy graph.

Figure 2(a) shows an anonymity system with three mix nodes, $S = \{s_1, s_2, s_3\}$, and $T = \{t_1, t_2, t_3\}$. Messages a, b, and c are within the system. Suppose the maximum route length for messages in this system is 2, i.e. any s_i can pass through at most 2 mix nodes. If an attacker knows the maximum route length of this system, and can observe messages entering and leaving *each* mix node, then he can infer that message c must be s_3, because if it were either s_1 or s_2 the route length condition is violated as then that message would have passed through 3 mix nodes. Therefore, t_3 cannot be s_3. Figure 2(b) shows the candidacy graph resulting from this analysis, from which the edge $\langle s_3, t_3 \rangle$ is thus absent.

Edman *et al.* [7] give another attack that notes the times at which messages enter an anonymity system and exit from it. If in addition to this, the attacker knows the minimum and/or maximum latency of messages in the system, a candidacy graph better than the complete bipartite graph can be arrived at, as shown in the example of Figure 3.

Entry times	Exit times
$s_1 = 1$	$t_1 = 4$
$s_2 = 2$	$t_2 = 5$
$s_3 = 4$	$t_3 = 7$
$s_4 = 5$	$t_4 = 8$

(a)

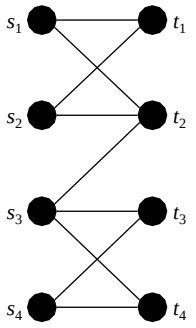

Fig. 3. (a) Message entry and exit times. (b) Candidacy graph.

Suppose each message entering a certain anonymity system comes out after a delay of between 1 and 4 time units. If 4 messages enter and exit this system at times shown in Figure 3(a), then s_1 must be either t_1 or t_2, because the other outgoing messages, namely t_3 and t_4, are outside the possible latency window of s_1. Similar reasoning can be performed on all other messages to arrive at the candidacy graph of this attack, shown in Figure 3(b).

C. Anonymity Metric

Given a candidacy graph obtained after performing an attack, the number of possible perfect matchings allowed by that graph between senders and receivers is a good indication of the anonymity left in the system in the aftermath of the attack.

For any bipartite graph $G = (U, V, E)$, where $|U| = |V|$, we let $\widehat{G}$ denote the number of one-to-one correspondences between U and V allowed by G. In other words, $\widehat{G}$ is the cardinality of the following set:

$$\{f : U \to V \quad | \quad f \text{ is a bijection, and } \forall u \in U, \langle u, f(u) \rangle \in E\}.$$

$\widehat{G}$ is also known as the *permanent* of the 0-1 adjacency matrix of G (see Servedio and Wan [13] for the permanent of a matrix).

When $G = (S, T, E)$, where $|S| = |T| = m \geq 1$, is the candidacy graph of an attack, then $\widehat{G}$ is essentially the number of perfect matchings between the incoming and outgoing messages possible after the attack. For a completely toothless attack, G is the complete bipartite graph $K_{m,m}$, as in Figure 1(a), and $\widehat{G} = m!$. If, on the other hand, G is the result of the most successful attack that has succeeded in correlating all incoming messages with their outgoing counterparts, as in Figure 1(b), then $\widehat{G} = 1$. For the attack of Figure 2, $\widehat{G}$ can be seen to be 4, and for that of Figure 3, $\widehat{G}$ can be seen to be 4 as well.

We define the anonymity level of the system after an attack with candidacy graph G as:

$$d(G) = \begin{cases} 0 & \text{if } m = 1, \\ \dfrac{\log(\widehat{G})}{\log(m!)} & \text{otherwise.} \end{cases}$$

As $\widehat{G}$ perfect matchings between S and T are still possible after the attack (of the $m!$ total possible before the attack), $\log(\widehat{G})$ / $\log(m!)$ captures the amount of information the attacker still needs to reveal the entire communication pattern of the system. The value of $d(G)$ always lies between 0 and 1. When $d(G) = 0$, $\widehat{G} = 1$ and the system provides no anonymity to any sender. When $d(G) = 1$, $\widehat{G} = m!$, i.e. the system is providing maximum anonymity. For the attack of Figure 2, the anonymity level is log(4) / log(6) $\approx$ 0.774, and for that of Figure 3, the anonymity is log(4) / log(24) $\approx$ 0.436.

It is instructive to observe that most other entropy-based metrics, such as of Serjantov and Danezis [11] or of Diaz *et al.* [5], are for single messages. Such metrics thus have reasonable intuitive interpretations with as well as without

normalization by the number of messages or users in the system. The metric defined above, on the other hand, is for the *entire* anonymity system. Here, normalization (i.e. division by $\log(m!)$) is necessary for correct measurement, as illustrated by the following argument.

Suppose a new mix is added to the system in Figure 2(a) with s_4 as its only input and t_4 the only output. The candidacy graph of this system will have the edge $\langle s_4, t_4 \rangle$, in addition to those in Figure 2(b). The anonymity level of the modified system is clearly lower than that of the original one, as it exposes s_4 completely. However, $\widehat{G} = 4$ is still the same for both systems, so by itself is not a good metric. The normalized metric, on the other hand, decreases from log(4) / log(6) $\approx$ 0.774 to log(4) / log(24) $\approx$ 0.436, which is in line with our intuitive lowering of the anonymity level.

III. Data Caching

The model of the anonymity system presented in the previous section has the following two characteristics:

- the communication carried by the system is only *unidirectional*, and
- the system simply relays *all* messages.

The above characteristics are limiting. Firstly, such systems are often used by clients to anonymously access resources from servers, such as web-browsers obtaining files from web-servers during a session of anonymous web-surfing. In such applications, clients need to send anonymous requests to servers and receive responses to their requests, i.e. communication via the anonymity system is bidirectional. Secondly, if the anonymity system is equipped with the ability to store some requested content in its cache, subsequent client requests for those cached resources can be satisfied by the system itself. As those requests do not need to be relayed to their respective end servers, they can be blocked by the anonymity system. In this section we present a model of an anonymity system with such caching abilities, and show that this added functionality improves the overall anonymity provided by the system.

A. A New Model

We consider an anonymity system that maintains one or more internal caches. The system might fill its caches either in an eager and proactive manner, such as by fetching in advance contents of web pages frequently accessed via it, or in a lazy and reactive manner, as by storing a copy of some of the server responses that pass through it. Upon receiving any request message from a sender, it first determines if that request can be served by cached content already present in the system. If so, that incoming request is served from within and it does not appear as an outgoing message of the system.

If S and T are, as before, the sets of input and output messages of this system, respectively, this system behavior results in $|S| \geq |T|$. Note that, in general, a perfect matching between S and T is not possible any more.

Of the $|S| = m$ input messages of the system, suppose only $|T| = n$ appear as output, where $m \geq n$. The remaining $m - n$ messages are assumed to be blocked by the system due to caching. The candidacy graph of an attack will now be a bipartite graph between S and T whose set of edges will be a subset of the set of edges in the complete $K_{m,n}$ bipartite graph.

As an example, suppose the system of Figure 2(a) does not produce message t_3 as output because it can service that request from its cache. Thus, for this system, $S = \{s_1, s_2, s_3\}$, and $T = \{t_1, t_2\}$. The candidacy graph of the same attack on this modified system is essentially that in Figure 2(b), from which the vertex t_3 is deleted, along with all edges connected to it. The resulting candidacy graph is shown in Figure 4(a). It is, in fact, the complete bipartite graph $K_{3,2}$, because the attack can now eliminate no input-output message pair as infeasible.

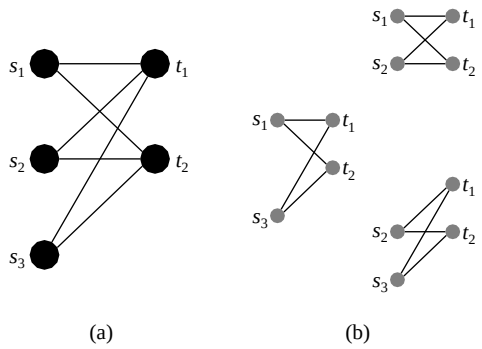

Fig. 4. (a) Candidacy graph G of route length attack on system of Figure 2, modified with cache for message t_3. (b) All left-projections of G.

B. A Generalized Anonymity Metric

As for measuring the anonymity provided by a system that performs caching, we begin by making the following two observations:

- The $m - n$ blocked messages do not appear as output, so cannot be correlated with any outgoing message. The anonymity received by these blocked messages is therefore considered to be maximal, since an observer cannot deduce *whom* their senders communicated with.
- Although there may not be a perfect matching between S and T (if their sizes are different), the n outgoing messages must be some n of the m incoming messages. The *total* number of possible matchings allowed by a candidacy graph, between T and any subset of S of size n, is a now a good indication of the anonymity provided by the system after the attack.

For any set U, and $n \geq 0$, we let $\mathcal{S}_n(U)$ be the set of all subsets of U of size n.

Definition 1: A bipartite graph $P = (W, V, F)$ is called a left-projection *of a bipartite graph $G = (U, V, E)$, if:*

1) $W \in \mathcal{S}_{|V|}(U)$, *and*
2) $F = \{\langle u, v \rangle \in E \mid u \in W\}$.

In other words, P is a subgraph obtained from G by removing any $|U| - |V|$ vertices from U (and edges connected to those vertices). We let $\mathcal{L}(G)$ denote the set of all left-projections of

G. The following proposition follows immediately from the above definition.

Proposition 1: If G is a complete bipartite graph $K_{m,n}$, where $m \geq n$, then each $P \in \mathcal{L}(G)$ is a complete bipartite graph $K_{n,n}$.

Figure 4 is an example of the situation mentioned in the above proposition. The candidacy graph of Figure 4(a) is complete, with $m = 3$ and $n = 2$. Figure 4(b) shows all its left-projections. They are all complete as well, with just different vertex labels. The following proposition is also straightforward.

Proposition 2: Any labeled bipartite graph $G = (U, V, E)$ has $\binom{|U|}{|V|}$ distinct left-projections.

We now develop our metric for the anonymity provided by a system after an attack that results in a candidacy graph $G = (S, T, E)$, where $|S| = m \geq 1$, $|T| = n \geq 0$, and $m \geq n$. We define the system's anonymity level as:

$$d'(G) = \begin{cases} 0 & \text{if } m = n = 1, \\ 1 & \text{if } n = 0, \\ \dfrac{1}{m}\left[(m-n) + \dfrac{n \log\left(\sum_{P \in \mathcal{L}(G)} \widehat{P}\right)}{\log(\binom{m}{n}\, n!)}\right] & \text{otherwise.} \end{cases}$$

The above metric can be viewed as the arithmetic mean of the anonymity provided by the system to each of the m incoming messages. Its value also lies between 0 (for no anonymity) and 1 (for full anonymity).

As observed earlier, of all the m input messages, $m - n$ messages get blocked by the system and do not appear as output, so the anonymity provided to those $m - n$ messages is maximal, i.e. 1.

The total number of perfect matchings allowed by G for the remaining n messages is the sum of the number of their perfect matchings allowed by all the left-projections of G, i.e.

$$\sum_{P \in \mathcal{L}(G)} \widehat{P}.$$

Moreover, there are $\binom{m}{n}$ left-projections of G, each of which can allow a maximum of $n!$ perfect matchings.

The following theorem shows that the above metric d' is a generalization of the old metric d, in that the two coincide when the system cannot make use of caching to eliminate any output message.

Theorem 1: If $m = n$, $d'(G) = d(G)$.

Proof: Both metrics are 0 for $m = 1$. When $m > 1$, since $m = n$, $\mathcal{L}(G)$ is the singleton set $\{G\}$, and $\binom{m}{n} = 1$. Thus, in this case, $d'(G)$ simply reduces to $\log(\widehat{G})$ / $\log(m!)$, which is $d(G)$. ∎

C. Enhanced Anonymity

Although it is intuitive that caching enhances the anonymity provided by a system, we now formally prove that fact. We begin by revisiting the example of Figure 4 and, as an exercise, compute the anonymity of that attack according to our new metric. For the candidacy graph G of Figure 4(a), $m = 3$ and $n = 2$. All left-projections of this candidacy graph are shown in Figure 4(b). It is easily seen that for each $P \in \mathcal{L}(G)$, $\widehat{P} = 2$. The anonymity level $d'(G)$ is thus

$$\frac{1}{3}\left[1 + \frac{2\,\log(2+2+2)}{\log(\binom{3}{2}\ 2!)}\right],$$

which is 1. On the other hand, the anonymity level of the candidacy graph of Figure 2(b) was computed in Section II-C to be log(4) / log(6) ≈ 0.774. Caching has thus resulted in an increase in the anonymity of this system by eliminating the outgoing message t_3. It can be verified that a somewhat modest gain in anonymity, from 0.774 to 0.849, is achieved by eliminating instead any one of the other messages, t_1 or t_2. If, however, both t_1 as well as t_2 can be eliminated, the system's anonymity level is higher, 0.877. We now show that such a gain in the anonymity can always be expected by caching.

Definition 2: A bipartite graph $H = (U, Q, F)$ is called a right-clipping *of a bipartite graph $G = (U, V, E)$, denoted $G \succeq H$, if:*

1) *$V \supseteq Q$, and*
2) *$F = \{\langle u, v\rangle \in E \mid v \in Q\}$.*

In other words, H is a subgraph obtained from G by removing zero or more vertices from V (and edges connected to those vertices). The following proposition follows immediately from the above definition.

Proposition 3: If G is labeled, it has $2^{|V|}$ right-clippings.

Note also, that $\succeq$ is a reflexive, antisymmetric, and transitive relation on bipartite graphs, resulting in the following proposition.

Proposition 4: $\succeq$ is a partial order.

It is instructive to observe that, adding caching into an anonymity system essentially reduces the candidacy graph of a given attack to some right-clipping of it. Our main result of this section will demonstrate a monotonic relationship between the amount of caching performed by a system and the level of anonymity provided by it. We first establish the following lemma pertaining to clipping *exactly* one right-vertex of a candidacy graph.

Lemma 1: Let $G = (U, V, E)$ and $H = (U, Q, F)$, where $|U| = m \geq |V| = n + 1$, and $|Q| = n$. If $G \succeq H$, then

$$\frac{(n+1)\ log\left(\sum_{P \in \mathcal{L}(G)} \widehat{P}\right)}{log(\binom{m}{n+1}\ (n+1)!)} \leq 1 + \frac{n\ log\left(\sum_{P \in \mathcal{L}(H)} \widehat{P}\right)}{log(\binom{m}{n}\ n!)}.$$

Proof: G has one vertex, called v, in addition to those in H, i.e. $V - Q = \{v\}$. Let X be the number of perfect matchings in H of Q, i.e. $X = \sum_{P \in \mathcal{L}(H)} \widehat{P}$. Consider any such perfect matching of Q in some fixed $P_0 = (U_0, Q, F_0) \in \mathcal{L}(H)$. As shown in Figure 5, since $|U_0| = n$, the vertex v can be connected in G to at most all of $m - n$ vertices of $U - U_0$. There are thus at most $m - n$ perfect matchings in G of V that preserve the given perfect matching in H. Therefore,

$$\sum_{P \in \mathcal{L}(G)} \widehat{P} \leq (m - n)X.$$

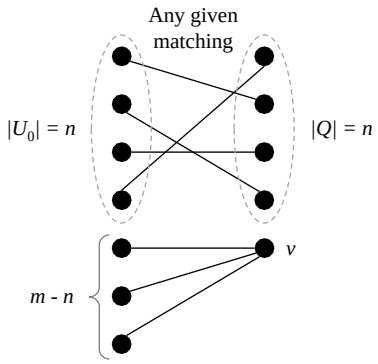

Fig. 5. All possible connections of vertex v in G, for any given perfect matching of Q in H.

Also, let $Y = \binom{m}{n} n!$. It is easily verified that

$$\tbinom{m}{n+1} (n+1)! = (m-n)Y.$$

It thus suffices to show that

$$1 + \frac{n \log(X)}{\log(Y)} - \frac{(n+1)\log((m-n)X)}{\log((m-n)Y)} \geq 0.$$

Both denominators in the above expression are positive. The numerator of the expression obtained by converting the above to have a common denominator for its terms is

$$\log(Y)\log((m-n)Y) + n\log(X)\log((m-n)Y) \\ -(n+1)\log((m-n)X)\log(Y).$$

It remains to be shown that the above is always non-negative. By using elementary properties of logarithms and performing algebraic simplification, the above expression can be easily transformed to

$$[\log(Y) - \log(X)]\ [\log(Y) - \log((m-n)^n)].$$

As Y is the *maximum* number of perfect matchings possible in H of Q, and X is the *actual* number of them, it follows that $Y \geq X$, thus $\log(Y) - \log(X) \geq 0$. Also, since $Y = \prod_{i=m-n+1}^{m} i$, which is a product of n values, each of whom is larger than $m-n$, we have that $Y \geq (m-n)^n$. Thus, $\log(Y) - \log((m-n)^n) \geq 0$, and the lemma holds. ∎

We now show that by increasing the amount of data caching performed by an anonymity system, the level of anonymity provided by that system is also increased.

Theorem 2: Let $G = (U, V, E)$ and $H = (U, Q, F)$, where $|U| = m \geq |V| = n_G \geq |Q| = n$. If $G \succeq H$, then

$$d'(G) \leq d'(H).$$

Proof (By weak induction on $n_G - n$): In the base case, $n_G - n = 0$. The theorem follows trivially, as now $G \succeq H$ implies $G = H$.

In the inductive case, $n_G - n > 0$. By the inductive hypothesis, for all bipartite graphs $H' = (U, Q', F')$, such that $|Q'| = n+1$ and $G \succeq H'$, we have that $d'(G) \leq d'(H')$. Of these, there are exactly $n_G - n$ graphs H' for which, additionally, $H' \succeq H$. Since $n_G > n$, firstly, at least one such graph exists, and secondly, $m > 1$. Let H_0 be that graph. From the definition of $d'(H_0)$, we have

$$d'(H_0) = \frac{1}{m}\left[m - n - 1 + \frac{(n+1)\log\left(\sum_{P\in\mathcal{L}(H_0)}\widehat{P}\right)}{\log(\binom{m}{n+1}\ (n+1)!)}\right].$$

By applying Lemma 1, we obtain

$$d'(H_0) \leq \frac{1}{m}\left[m - n + \frac{n\log\left(\sum_{P\in\mathcal{L}(H)}\widehat{P}\right)}{\log(\binom{m}{n}\ n!)}\right].$$

The above expression is $d'(H)$, and the theorem follows from the transitivity of $\succeq$. ∎

IV. Application to Threshold Mix Networks

Threshold mix networks were introduced by Serjantov, Dingledine and Syverson [12] as a high-latency strategy to counter input-output message correlation by operating a mix in iterative rounds. A message is initially multiply encrypted, and each mix that it passes through decrypts its outermost layer. In each round, a mix in such networks collects some threshold number m of incoming messages and outputs a decrypted version of them in some different order. The decryption prevents correlation by simple message bit-pattern comparison. The main goal of this strategy is to prevent correlation by message entry/exit sequence numbers by flushing messages out in an order other than the one in which they came in. Waiting for m messages to arrive before sending them out adds latency, which is the price paid for achieving this goal.

From the point of view of anonymity, an important difference between this model and the one we have been considering so far in this paper is that any incoming message now blends in with other messages only in its own round. This has a detrimental effect on the overall anonymity of the system, especially as rounds progress. Let after r rounds, $n \leq m$ be the average number of messages sent out in each round (where the remaining $m - n$ incoming messages per round were served by the system's internal cache).

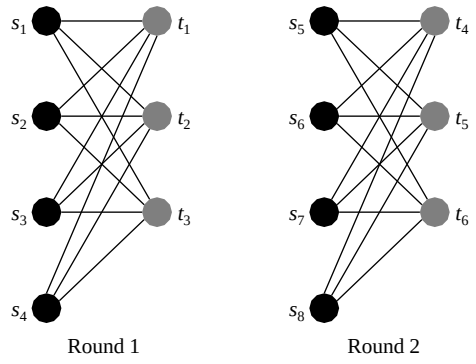

Fig. 6. Candidacy graph without any attack after two rounds, with $m = 4$ and $n = 3$.

Figure 6 shows the candidacy graph without any attack after two rounds, with $m = 4$ and $n = 3$. Since there is no attack,

the candidacy graph after the first round is $K_{4,3}$, for which the level of anonymity is $d' = (1/4) * ((4-3) + 3\log(24) \,/\, \log(24)) = 1$. However, after two rounds, as shown in Figure 6, the candidacy graph is $K_{4,3} \cup K_{4,3}$, for which the anonymity level can be seen to have been lowered from $d' = 1$ to $d' = (1/8) * ((8-6) + 6\log(24) \,/\, \log(56)) = 0.842$.

In general, after r rounds, the overall anonymity provided by the mix thus far is

$$\frac{1}{m}\left[m - n + \frac{nr\,\log(\binom{m}{n}\, n!)}{\log(\binom{mr}{nr}\,(nr)!)}\right].$$

Table I shows how the anonymity level of an example system with $m = 100$ varies with the number of rounds and the average amount of caching in each round performed by the system.

Round Number	Average Caching Level				
	0%	20%	40%	70%	100%
1	1	1	1	1	1
2	0.7136	0.7889	0.851	0.9303	1
3	0.6069	0.708	0.7927	0.9022	1
4	0.5477	0.6625	0.7594	0.8861	1
5	0.5086	0.6322	0.7371	0.8752	1
6	0.4805	0.6102	0.7208	0.8671	1
7	0.4588	0.5932	0.7082	0.8609	1
8	0.4415	0.5796	0.698	0.8558	1
9	0.4273	0.5683	0.6895	0.8516	1
10	0.4153	0.5587	0.6824	0.8481	1

TABLE I
ANONYMITY IN A THRESHOLD NETWORK WITH $m = 100$, OVER DIFFERENT AVERAGE CACHING LEVELS AND MIX ROUNDS

In the above table, the case of 0% caching corresponds to $n = m$. If, on an average, the system responds to 20% of the incoming messages by its cache, then $n = 0.8\ m$. As an extreme case, if all incoming messages can be handled by the system's cache, i.e. $n = 0$, then the level of anonymity provided by the system after each round is 1, which is the maximum.

Figure 7 depicts the anonymity variation given in Table I in the form of an easier to visualize graph. It is evident from the graph that the anonymity provided by the system reduces as the rounds progress, but increases with the amount of caching performed in the system.

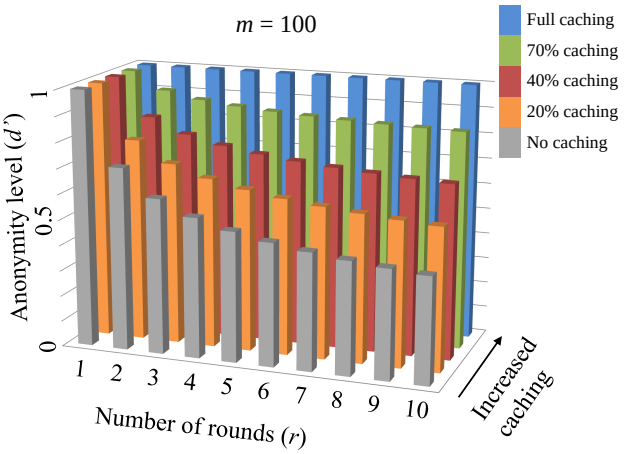

Fig. 7. Graphical representation of anonymity levels in Table I.

V. CONCLUSIONS AND FUTURE WORK

Anonymity systems that are used for applications such as anonymous web surfing, first attempt to provide sender anonymity for requests sent by the users to servers, and then receiver anonymity for the responses sent back by the servers to the requesting users. If such systems have the ability to cache some server responses, then those cached contents can be used for future user requests for the same resources, without having to relay those requests to the servers. We have shown that the overall anonymity provided by such systems to their users is higher than that provided by systems that do not employ caching. In fact, our results show that the greater the number of requests that can be served from the system's internal cache, the higher the anonymity. To the best of our knowledge, our work is the first attempt at quantifying the anonymity gains resulting from caching.

Most currently popular techniques for providing anonymity, such as message routing via proxies and onion encryption, result in increasing message latency and are thus carried out at the *expense* of performance. In contrast, data caching has the advantage of providing anonymity in addition to *improving* performance, as it clearly also results in reduction of bandwidth usage and data access latency. Extra storage capacity of mix nodes for holding the caches and extra effort needed to keep those caches up-to-date are two trade-offs for the gains achieved by caching. As storage is constantly becoming more readily available, the first trade-off is insignificant. However, maintaining cache consistency incurs traffic overhead and, if not dealt with carefully, could cause unreasonable performance degradation. Analysis of this aspect of caching in anonymity systems, we have for now left as future work.

Our technique opens up several other possible future research directions, as outlined below.

A network of mix nodes, as proposed by Chaum [1], is a popular architecture of current anonymity systems, such as TOR [6] and Mixminion [4]. In these systems, messages are routed over paths of mix nodes, and these paths are constructed dynamically from the pool of available mix nodes at any time. An interesting direction for future research is developing strategies for selecting mix nodes for placement of caches, integrated with strategies for constructing paths of mix nodes in order to maximize the system's anonymity level. If a user's access patterns can be anticipated, then that user's requests can be routed over paths that are more likely to contain caches that can satisfy those requests.

The Statistical Disclosure Attack (SDA), proposed by Danezis [3], is known to be a very powerful attack, targeted against a single user. The attack uncovers the servers often contacted by that user. Many defense strategies have been proposed for thwarting this attack, of which the one by Mallesh

and Wright [10] of sending dummy messages generated from within the anonymity system is particularly effective. We plan to combine that strategy with our caching technique, such as for example by generating a dummy request each time a genuine request is served by a cache. Such a combined strategy should be an even more effective countermeasure against SDA, especially if the system has an intelligent strategy to send its dummy messages to strategically chosen servers.

The edges of the candidacy graphs we considered in this paper are all equally likely. However, an attacker may be able to assign probabilities to them by some function $p(u, v)$ indicating the probability that the incoming message u is the outgoing message v. An anonymity metric for systems with probabilistic attacks is presented in Edman *et al.* [7]. We are currently working on arriving at a metric for such systems in the presence of caching.

ACKNOWLEDGMENT

The research described in this paper has been partially supported by the United States Navy Engineering Logistics Office contract no. N41756-08-C-3077.

REFERENCES

[1] D. Chaum. Untraceable electronic mail, return addresses, and digital pseudonyms. *Communications of the ACM*, 24(2):84–88, 1981.

[2] D. Chaum. The dining cryptographers problem: Unconditional sender and recepient untraceability. *Journal of Cryptology*, 1:65–75, 1988.

[3] G. Danezis. Statistical disclosure attacks: Traffic confirmation in open environments. In *Proceedings of Security and Privacy in the Age of Uncertainty*, pages 421–426, Athens, May 2003.

[4] G. Danezis, R. Dingledine, and N. Mathewson. Mixminion: Design of a type iii anonymous remailer protocol. In *Proceedings of the IEEE Symposium on Security and Privacy*, pages 2–15, May 2003.

[5] C. Diaz, S. Seys, J. Claessens, and B. Preneel. Towards measuring anonymity. In *Proceedings of the 2nd Privacy Enhancing Technologies Workshop*, pages 54–68, 2002.

[6] R. Dingledine, N. Mathewson, and P. Syverson. Tor: The second-generation onion router. In *Proceedings of the 13th USENIX Security Symposium*, pages 303–320, August 2004.

[7] M. Edman, F. Sivrikaya, and B. Yener. A combinatorial approach to measuring anonymity. In *Proceedings of the IEEE International Conference on Intelligence and Security Informatics*, pages 356–363, 2007.

[8] D. Kelly, R. Raines, M. Grimaila, R. Baldwin, and B. Mullins. A survey of state-of-the-art in anonymity metrics. In *Proceedings of the 1st ACM Workshop on Network Data Anonymization*, pages 31–39, 2008.

[9] B. Kim and K. Kim. Efficient caching strategies for Gnutella-like systems to achieve anonymity in unstructured P2P file sharing. In *Proceedings of the 6th Conference on Next Generation Information Technologies and Systems*, pages 117–128, Kibbutz Shefayim, Israel, 2006.

[10] N. Mallesh and M. Wright. Countering statistical disclosure with receiver-bound cover traffic. In *Proceedings of the 12th European Symposium on Research In Computer Security*, pages 547–562, Dresden, Germany, 2007.

[11] A. Serjantov and G. Danezis. Towards an information theoretic metric for anonymity. In *Proceedings of the 2nd Privacy Enhancing Technologies Workshop*, pages 41–53, 2002.

[12] A. Serjantov, R. Dingledine, and P. Syverson. From a trickle to a flood: Active attacks on several mix types. In *Proceedings of the 5th International Workshop on Information Hiding*, Noordwijkerhout, The Netherlands, October 2002.

[13] R. A. Servedio and A. Wan. Computing sparse permanents faster. *Information Processing Letters*, 96(3):89–92, 2005.

[14] A. M. Shubina and S. W. Smith. Using caching for browsing anonymity. *ACM SIGEcom Exchanges*, 4(2):11–20, 2003.

[15] G. Tóth, Z. Hornák, and F. Vajda. Measuring anonymity revisited. In *Proceedings of the 9th Nordic Workshop on Secure IT Systems*, pages 85–90, Espoo, Finland, 2004.