Breaking the Collusion Detection Mechanism of MorphMix

Parisa Tabriz and Nikita Borisov

University of Illinois at Urbana-Champaign
{tabriz, nikita}@uiuc.edu

Abstract. MorphMix is a peer-to-peer circuit-based mix network designed to provide low-latency anonymous communication. MorphMix nodes incrementally construct anonymous communication tunnels based on recommendations from other nodes in the system; this P2P approach allows it to scale to millions of users. However, by allowing unknown peers to aid in tunnel construction, MorphMix is vulnerable to colluding attackers that only offer other attacking nodes in their recommendations. To avoid building corrupt tunnels, MorphMix employs a collusion detection mechanism to identify this type of misbehavior. In this paper, we challenge the assumptions of the collusion detection mechanism and demonstrate that colluding adversaries can compromise a significant fraction of all anonymous tunnels, and in some cases, a majority of all tunnels built. Our results suggest that mechanisms based solely on a node's local knowledge of the network are not sufficient to solve the difficult problem of detecting colluding adversarial behavior in a P2P system and that more sophisticated schemes may be needed.

1 Introduction

Over 20 years ago, David Chaum introduced the *mix* as a communication proxy to hide the correspondence between messages coming into and going out of a system [4]. Since then, this design has been extensively used to build anonymous systems ranging from remailers [6] to low-latency communication systems for anonymous Internet access [2, 7, 9, 20].

Most mix network designs use a relatively small and fixed set of mix servers for forwarding all traffic, usually on the order of several dozen. The current deployment of the Tor [7] network has been pushing this limit somewhat, with several hundred servers in operation, but the network cannot grow much larger without major changes to its design and implementation. This imposes a limit on how much traffic these networks can handle and therefore the size of the user population; already, the Tor network carries close to half the amount of traffic of its stated capacity, and the actual capacity of the network may in fact be significantly lower.[1] Furthermore, some recent traffic analysis techniques take advantage of the relatively small number of nodes to enumerate them all in the search for the forwarders of a particular stream [12].

MorphMix [20] represents an alternative, peer-to-peer design for anonymous networks. Each MorphMix user runs a node that both generates anonymous

---

[1] For example, drops in traffic volumes are correlated with drops in the stated capacity of the network [15], indicating that some lower limit may already be reached.

traffic of its own and acts as a mix server, forwarding anonymous traffic for others. This allows the capacity of the network to grow in proportion to the number of users. Even as the number of users reaches millions, the requirements on any single node are small; in particular, each node knows only a limited number of other nodes, and uses recursive queries of neighbors and neighbors' neighbors in order to find other nodes. For these reasons, MorphMix, or a similar design, holds the most promise for providing a global, widely-used anonymous communications infrastructure.

The limited knowledge at each node, however, can create a problem for MorphMix security. As each node relies on its neighbors to learn about other nodes in the system when building anonymous tunnels, a set of colluding malicious nodes could easily direct many tunnels to the colluding set and therefore compromise anonymity. To defend against such attacks, MorphMix introduced a new collusion detection mechanism (CDM). The original analysis of this mechanism considered several attack strategies and determined that in all cases, the number of tunnels that could be compromised by all colluding nodes is small [20].

In this paper, we present a new attack on the MorphMix collusion detection mechanism that is far more effective than those considered in the original analysis. Our key observation is that because the CDM relies solely on local knowledge and observations, attackers can effectively model the state of the CDM at each node and tailor their strategy accordingly. The attackers can therefore avoid detection for much longer and compromise a significant percentage of all tunnels constructed in MorphMix; in some cases, attackers can compromise the majority of all tunnels built. Our results show that the CDM introduced by MorphMix is not an effective means of defending against collusion attacks and that further research is needed to solve this important problem in decentralized peer-to-peer anonymous networks.

This paper is organized as follows. In Section 2, we review MorphMix and its collusion detection mechanism. In Section 3, we describe our attack. We present simulation results in Section 4 and in Section 5, we present both immediate countermeasures and permanent changes necessary to prevent this attack.

2 MorphMix

2.1 MorphMix and Anonymous Tunnels

MorphMix is a circuit-based mix network consisting of many MorphMix clients, or *nodes*, that act as both connection initiators and routers for the network. Each MorphMix node maintains a limited number of *virtual links* via TCP connections to *neighbor* nodes within the system. One unique feature of Morph-Mix is that the route a node uses for its connection, an *anonymous tunnel*, is constructed iteratively by other participating nodes in the system. We briefly describe MorphMix's anonymous tunnel construction, and refer the reader to [18] for a more detailed look at the MorphMix system and tunnel construction protocol.

An anonymous tunnel consists of the node establishing the connection, the *initiator*, zero or more *intermediate* nodes, and the *final* node of the anonymous

tunnel. Similar to other onion routing mix networks like Tor, MorphMix uses fixed size messages and layered encryption across each link of the anonymous tunnel to prevent against traffic analysis attacks and protect message content, respectively.

When an initiator node, a, wants to create an anonymous connection, it first establishes a shared key with one of its neighboring nodes, say b, which will be used to encrypt messages sent across that link of the tunnel. If a decides to extend the tunnel, it asks node b to recommend a *selection* of nodes from b's neighbors to use as a next hop in the anonymous tunnel. Node a then chooses from the offered selection a node, say c, to append to b in the tunnel. Node a establishes a symmetric key with c via b that it will use for encryption across the next link in the tunnel. To prevent against b performing a man-in-the-middle attack between a and c, a selects a *witness* node from the nodes it already knows[2] to establish the symmetric key between a and c. There are other attacks that can be considered if the witness is in collusion with b, but to simplify our presentation, we ignore this case and assume the witness is always an honest node. Once a tunnel to c is established, a can ask c for a selection of nodes to extend the tunnel further; this process continues until a has finished appending nodes to the tunnel.

By having the last node select the next hop to append during tunnel construction, MorphMix nodes only need to maintain state information about their local neighbors. This allows MorphMix to scale independently of the number of nodes in the system. However, an immediate threat is introduced when a malicious node is appended because it helps determine the next hop in the tunnel. To prevent a malicious node from offering selections biased with other malicious nodes, MorphMix employs a *collusion detection mechanism* (CDM) to identify this behavior and prohibit this form of attack. MorphMix assumes that a tunnel is compromised, or *malicious*, if the first intermediate and final node are both controlled by an attacker. Otherwise, it considers the tunnel *fair*.

2.2 Collusion Detection Mechanism

Similar to other anonymous systems, we assume the primary goal of an attacker in MorphMix is to link communications between initiators and recipients of a connection. While low-latency systems are generally more susceptible to traffic analysis, MorphMix is vulnerable to a more immediate attack when colluding nodes try to append other colluding nodes during tunnel construction. Morph-Mix detects this behavior by performing collusion detection on each offered selection.

If attackers own a whole range of IP addresses, it would be easy for them to operate many MorphMix nodes. To limit this threat, MorphMix distinguishes individual nodes from each other by their 16-bit IP address prefix. We refer to this prefix as the node's /16 subnet. It is much more costly and difficult for attackers to own nodes in many unique /16 subnets than it is for them to own many nodes in one or fewer /16 subnets. The CDM is built on the following two assumptions: honest selections will be comprised of nodes selected randomly

---

[2] MorphMix uses a peer discovery mechanism to learn about other nodes in the network to use for witness and neighbor selections.

from many different /16 subnets and malicious selections will be comprised of mostly or all colluding nodes coming from a limited portion of /16 subnets in MorphMix.

Each MorphMix node maintains a fixed size *extended selection list*, L_{ES}, of entries consisting of the concatenation of the selection and the 16-bit IP prefix of the node that sent that selection; we refer to this entry as an *extended selection* and each 16-bit IP prefix in the entry as the node's subnet. When a tunnel initiator receives a new selection, it compares it to each entry in its L_{ES} and calculates the proportion of node subnets that have been seen multiple times to those that have been seen only once. The computed correlation is expected to be larger for a colluding selection than it is for an honest selection because colluding nodes will limit their selections to only other nodes in their colluding set and honest nodes will offer selections consisting of neighbors that have been chosen more or less randomly from all nodes in the system. The algorithm, described in [20], is repeated below:

> Correlation Algorithm
> 1. Build a set ES_N consisting of the 16-bit IP address prefixes of the nodes in the new extended selection.
> 2. Define a result set ES_R which is empty at first.
> 3. Compare each extended selection ES_L in the extended selections list L_{ES} with ES_N. If ES_N and ES_L have at least one element in common, add the elements of ES_L to ES_R.
> 4. Count each occurrence of elements that appear more than once in ES_R and store the result in m.
> 5. Count the number of elements that appear only once in ES_R and store the result in u.
> 6. Compute the correlation c which is defined as $c = \frac{m}{u}$ if $u > 0$, or ∞ otherwise.

Each MorphMix node remembers the correlations it has computed over recent extended selections and represents these in a *correlation distribution*. Every time a node receives a new extended selection, it computes a correlation and updates this distribution according to an exponential weighted moving average. The results in [18] show that these distributions can often be characterized as having two peaks, one formed by the aggregate contribution of honest nodes and one formed by the aggregate contribution of malicious nodes (see Figure 2a). From this distribution, MorphMix determines a threshold point between the two peaks, the *correlation limit*, which has the property that correlations greater than this limit are malicious with high probability and correlations less than this limit are honest with high probability.

During tunnel construction, the initiator calculates the correlation of every extended selection it receives and compares this to its correlation limit. If any extended selection during the setup is detected as malicious, the tunnel is torn down and not used. Otherwise, the tunnel is considered fair and used for anonymous connections.

There are two important assumptions to highlight from the collusion detection mechanism that form the intuition behind our attack:

1. An extended selection from a colluding node will overlap with many other colluding entries stored in the L_{ES}, resulting in a large c.
2. The c of a malicious extended selection will, in general, be higher than the correlation limit determined for that node.

By limiting the number of selections that overlap in a victim's L_{ES}, a colluding adversary can keep c low such that it frequently falls beneath the correlation limit and is not detected during tunnel construction.

3 Attacking the Collusion Detection Mechanism

In this section, we define the adversary necessary to perform this attack, describe the attacker's goal, and present a general description of the attack.

3.1 Attacker Model

Anyone with access to the Internet and a MorphMix client can actively participate in MorphMix. Because attackers can so easily join, contribute, and exit the system, we assume an active, internal adversary. Specifically, we assume that there is some subset, n_c, of all MorphMix nodes, n, that is comprised of colluding nodes from unique subnets that are participating in MorphMix. In reality, the number of colluding nodes may be larger than n_c, but because the CDM does not differentiate between two nodes from the same subnet, we only consider the number of colluding nodes that can be represented by unique subnets. We assume the colluding set will conspire to choose how they offer selections to a victim node, but otherwise, behave honestly.

We specify that n_c will be comprised of nodes representing a percentage, C, of the unique subnets in MorphMix, where C can realistically range from 0% to 40%. This range represents different attackers present in the system, from a small group of attackers to an organization of moderate resources to an even larger network of compromised zombie machines. This range similarly follows the assumptions made by the MorphMix authors. Consequently, we analyze the success of our attack using a colluding set ranging in size from 5% to 40% of the unique subnets in MorphMix.

3.2 Attacker Goal

We assume that the goal of attackers is to link a connection initiator with some outgoing stream. Attackers can achieve this goal by owning the first intermediate and final node in an anonymous tunnel. This will happen with probability C^2 during normal MorphMix behavior. Our attackers, however, accomplish linkability by owning every node in the tunnel. We aim to show that by using intelligent selections, colluding attackers can expect to compromise every node in C anonymous tunnels built by some victim node.

3.3 Attack Description

Our attack is based on this simple intuition: Because each node's CDM is based on only the local knowledge stored in its L_{ES}, attackers can model and manipulate the L_{ES} to avoid being detected. To accomplish this, colluding nodes should only offer other colluding nodes in their selections, and they should organize and offer selections to a victim in such a way that they have the least overlap with other malicious selections in the victim's L_{ES}. The analysis in [18] simulates attackers offering selections that are comprised of nodes *randomly* chosen from the set of all colluding nodes. By being more intelligent with their selections, colluding attackers can limit the number of nodes in extended selections that contribute to m in the correlation algorithm. The attack works as follows:

Intelligent Selection Attack
1. For every victim, v, construct a list of selections, S_v, comprised of only colluding nodes such that there is no overlap in node subnets between any selection entry in S_v.
2. Maintain a global pointer, p_g, that keeps reference to a selection in S_v to be offered in the next attack attempt.
3. When v contacts any colluding node to be a first intermediate node and any subsequent node in a new anonymous tunnel, we offer to v the selection pointed to by p_g and increment p_g. If p_g pointed to the last element in S_v, we set p_g to be the first element of S_v and iterate once again through all elements in the list.

The above attack assumes that a colluding node can determine if it is the first intermediate node *during* tunnel construction. MorphMix makes this difficult by using the same witness mechanism for every step of tunnel construction, including the first. Therefore, a node cannot determine whether it is the first node or a later node from only the messages exchanged in the append protocol. Measuring message delays can help determine the position in the list, but in our attack, a node must decide if it is in the first position before returning the selection, with not enough messages exchanged to measure timings.

We therefore modify our attack to return selections from S_v for all tunnels arriving at a colluding node from v, including ones that may have originated from nodes other than v. After the fact, once the tunnel is fully constructed, it is easy to determine whether it originated from v by counting the number of links after v, since all nodes past v will be colluding. (The effects of variable tunnel lengths will be discussed in Section 5.)

Intelligent Selection Attack (Revised)
1. Whenever v requests a selection from a colluding node, we begin the attack by assigning a local pointer, p_l to the selection referenced by p_g and offer that selection to v. We cannot verify if v is the initiator of the tunnel or a node appended to a tunnel started by some other node, v'.
2. For every successive selection request, we increment p_l in S_v and offer the new selection that p_l points to.

Users per Subnet	Percentage of Overnet
1	72%
2	14%
3	7%
4	4%
5	2%
6	1%

Table 1: Node distribution according to Overnet traffic traces. For example, 72% of Overnet is composed of users coming from unique subnets, 14% of Overnet is composed of users coming from subnets with two active users, etc.

3. After the tunnel is created, we determine if the tunnel initiator was v or some v' by measuring the tunnel length.

4. If the tunnel was initiated by v, we update p_g to hold the value referenced by p_l. Otherwise, some v' was the initiator of this tunnel. Since our attack selections are stored in the L_{ES} of v', they can still be used against v. In this case, p_g maintains its original value.

Next we will use simulations to determine how effective this attack is at avoiding the collusion detection mechanism.

4 Simulation

4.1 MorphMix Settings

Because MorphMix does not have a substantial user base, we are unable to execute this attack on a live system. Instead, we simulate many tunnel constructions using the CDM from the MorphMix client prototype [19] and investigate the effects of the attack on one node, the victim node. We evaluate how successful the attack is based on how many tunnels we can compromise, what proportion of all tunnels constructed can be compromised, and how long the attack can run successfully.

The analysis in [18] simulates construction of 5000 anonymous tunnels from a network of 10,000 MorphMix nodes with every node coming from a unique /16 subnet. We believe this is not indicative of a realistic user distribution for an unstructured, decentralized network such as MorphMix. The real Internet is composed of a high concentration of users from certain subnets and we choose to represent this imbalance. Additionally, each node's correlation limit in Morph-Mix is based on the correlations of all recent selections it has seen, both honest and malicious. Because of this, it is important to simulate as realistic a network distribution as possible, namely, one that consists of users coming from both common and unique subnets. We look to a popular P2P system of similar structure, the Overnet/eDonkey file-sharing system, to provide more realistic statistics [1].

Resulting from traffic probes taken during 2003, Overnet consisted of the subnet to node distribution displayed in Table 1 [3]. We simulated 5000 tunnel

constructions consisting of only honest selections from both the original node distribution in [18] and our own node distribution based on the Overnet traces. The average correlation limit in the original distribution was .145 ($\sigma = .005$) and the average correlation limit using the Overnet trace was .172 ($\sigma = .005$), a significant difference given identical network parameters between the two simulations aside from the underlying node distribution. From Figure 2a, we can see that the distance between the peak formed by honest selections and the peak formed by malicious selections is already very small. Increasing the correlation limit by even a small amount will make this distance even closer and the correlation limit harder to define. While the Overnet spread may not exactly represent a deployed MorphMix system, we believe it is more indicative of P2P use than the one used in the original MorphMix simulation. For this reason, we follow this distribution during our experimentation.

Aside from this change, we use the same fixed tunnel size of 5 nodes and compute the size of the L_{ES} and the size of node selections identical to [18].[3] Every node in MorphMix maintains virtual links to neighbors that are chosen as the first intermediate node during tunnel construction. Virtual links with other nodes are established more or less randomly from all of the nodes known to a user. We consider that C tunnels will actually begin with a colluding node and $(1 - C)$ will begin with an honest node and be impossible to compromise.

In [17], the authors of MorphMix do an additional analysis of the CDM, taking into account node tunnel acceptance rates and uptime probabilities. We have chosen to ignore these additional constraints in our simulation for simplicity, but believe they would only strengthen the success of the attack: while honest nodes may occasionally refuse to accept new tunnels and leave the network due to limited capabilities, malicious nodes are likely to devote more resources to their attacks and have higher acceptance probabilities and network uptimes. Therefore, we can expect even fewer than $(1 - C)$ tunnels would have a first intermediate honest node.

4.2 Attacker Settings

Attackers will blindly assume that any initial selection request from v (and all subsequent selection requests) are contributing to a tunnel initiated by v. If v was not the tunnel initiator though, the attack selections destined for v actually arrived at some other node, v', and are stored in that node's L_{ES}.

If our attack is being executed against many victim nodes in MorphMix, attacker selections destined for one victim may accidentally be misdirected to a different victim. To minimize the effect these misdirected selections have on the collusion detection mechanism, the attackers should use a different random permutation of n_c when constructing S_v for each different victim v. When v' receives a selection destined for v, it will appear as a random sample of nodes from $S_{v'}$, and that is in fact how we model misdirected selections in our simulations. More specifically, whenever a node tries to extend a tunnel that starts

---

[3] In MorphMix, tunnel size is fixed for the duration of the session and has a default value of 5 nodes. While this value can be changed upon restart, we assume most users would keep the default configuration. We address the effects of violating this assumption in Section 5.

C	Honest Tunnels	Malicious Tunnels	Percentage Compromised
5%	3337.9	6.8	0.2% ($\sigma = 0.1\%$)
10%	2951.4	33.8	1.1% ($\sigma = 0.2\%$)
15%	2283.2	470.1	17.1% ($\sigma = 1.5\%$)
20%	1930.0	860.4	30.8% ($\sigma = 1.1\%$)
30%	1171.5	1384.0	54.2% ($\sigma = 2.4\%$)
40%	450.9	1847.5	80.4% ($\sigma = 2.3\%$)

(a) Uninterrupted attack execution.

C	Honest Tunnels	Malicious Tunnels	Percentage Compromised
5%	4251.9	51.8	1.2% ($\sigma = .2\%$)
10%	4161.2	146.9	3.4% ($\sigma = .2\%$)

(b) Optimized attack execution.

Table 2: Tunnel construction for range of attackers.

with an honest node, with probability C a malicious next node is chosen, who will then return a misdirected selection. The misdirected selection is represented as a random set of nodes from n_c. If the tunnel is then extended further, we assume the malicious node will carry out the attack and provide more misdirected selections, once again represented by a random sample from n_c.

We simulate the modified attack as described in Section 3.3 by first creating a random permutation of attacking nodes and storing this ordering into S_v. We select the first k nodes, where k is the selection size, and continue to cycle through S_v to create unique selections.

We briefly explored more sophisticated ways of creating S_v such that more selections can be made with minimal overlap, however, our initial results showed that even a basic organization of how colluding selections are offered is enough to result in significant attacker success.

4.3   Attack Execution

We execute the attack during 5000 tunnel construction attempts by a single victim node and calculate how many successful tunnels are constructed. In Morph-Mix, a node creates, on average, a new anonymous tunnel every 2 minutes. Therefore, creating 5000 tunnels is roughly equivalent to one week of constant MorphMix usage. In Table 2a, we can see that the attack results in a significant portion of anonymous tunnels being compromised using intelligent selections. If colluding adversaries control nodes in more than 15% of the represented subnets in MorphMix, they are able to compromise at least that percentage of tunnels constructed by victims. Attacking levels above 30% result in the majority of all constructed tunnels being compromised by an attacker. While adversaries that control nodes in less unique subnets cannot claim quite as high statistics, by
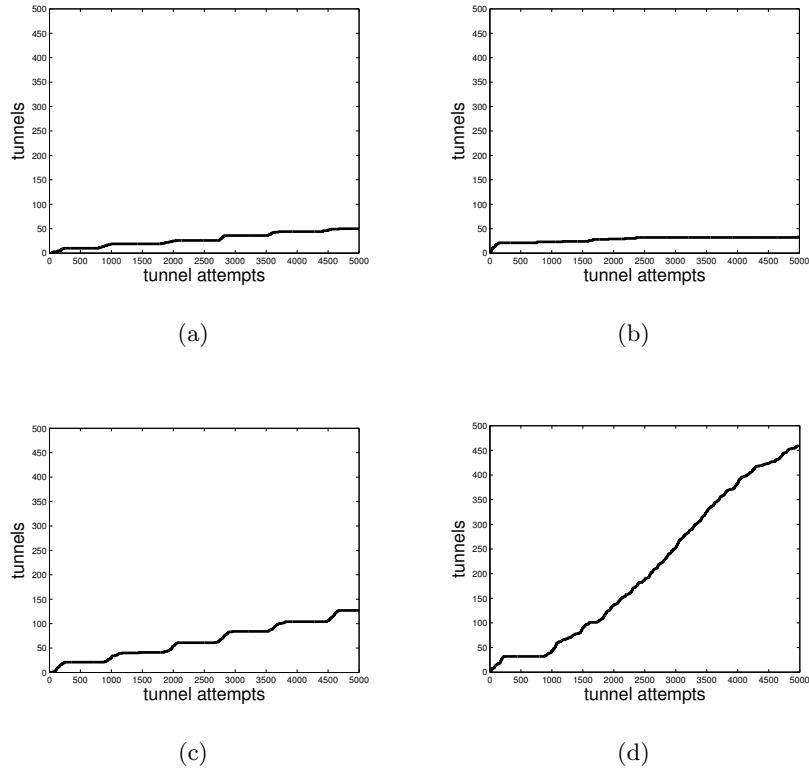
Fig. 1: Successfully constructed tunnels from colluding adversaries with (a) $C = 5\%$ executing an optimized attack, (b) $C = 10\%$ executing an uninterrupted attack and (c) an optimized attack, and (d) $C = 15\%$ executing an uninterrupted attack.

slightly adjusting the attack, they can still successfully compromise more than C^2 anonymous tunnels.

4.4 Optimized Execution for Smaller Adversaries

The main problem that attackers have when they own few nodes in unique subnets is that they are more limited in the number of unique selections they can create. If they continue the attack uninterrupted, these selections will begin to overlap in a victim's L_{ES}, causing the correlation and chance of detection to raise. If attackers owning nodes in less than 15% of the unique subnets in MorphMix attack uninterrupted, they will eventually saturate the victim's L_{ES} and be detected with high probability once they starts repeating selections. In this case, they can optimize their attack by using intelligent selections to build tunnels until they runs out of unique selections and then behave normally until

the victim's L_{ES} has cleared. Because nodes evict the oldest entries from their L_{ES}, attackers can estimate how long it will take for the victim's L_{ES} to be cleared based on how often and at what rate the victim creates tunnels. Both of these parameters have initial values in each MorphMix client, and even if they are changed, they are limited by a small range of realistic values.

We test this strategy during 5000 tunnel attempts using identical simulation parameters and present the results in Table 2b. In Figure 1, we compare the results of the uninterrupted and optimized attack for C ranging from 5% to 15%. We note that attackers with $C = 10\%$ executing an uninterrupted attack can only compromise around 30 tunnels before they have saturated the L_{ES} and cannot compromise any more tunnels. However, if they attack until they run out of unique selections and then wait until the node's L_{ES} has cleared, they can compromise almost five times as many tunnels and can continue to attack the victim in this manner indefinitely.

This interrupted strategy is only necessary for colluding attackers with limited resources. Specifically, it is only necessary for those with nodes in less than 15% of the uniquely represented subnets in MorphMix. As seen in Figure 1d, attackers with $C = 15\%$ can continue to compromise tunnels indefinitely without waiting for a victim's L_{ES} to clear.

In theory, there is an improved strategy that a limited attacker can use when behaving honestly. Attackers should provide selections that consist of very few malicious nodes and many other unique honest nodes during this honest behavior period. This way, the victim's L_{ES} becomes filled with selections that will overlap with future attacking selections, yet make a large contribution to u in the correlation algorithm. This will, in turn, lower the correlation and decrease the chance of future detection.

5 Attack Countermeasures

In Section 2.2, we reviewed the CDM and how the correlation limit is determined in practice. As shown in [18], when colluding adversaries provide selections of nodes that are *randomly* selected from only participating attackers, the contribution of these selections to the correlation distribution forms a distinguishable second peak to the right of the contribution of honest selections. We reproduce this effect by simulating an attacker that controls nodes in 20% of the unique subnets in MorphMix and attacks with selections of only randomly chosen malicious nodes. The resulting correlation distribution and correlation limit are displayed in Figure 2a. Next, we simulate the same adversary using *intelligent* selections. The results in Figure 2b show that using this method destroys the dual-peak characteristic of the correlation distribution. This, in turn, creates a less meaningful correlation limit, crippling the detection mechanism.

An immediate countermeasure to this attack might be to increase the number of nodes in the tunnel and increase the number of entries in the L_{ES}. Increasing the number of nodes in the tunnel would force the attackers to use more selections for each tunnel. This would cause their attacking selections to overlap much sooner in the L_{ES}, driving up the correlation before as many tunnels can be compromised. Increasing the number of entries in the L_{ES} has a similar
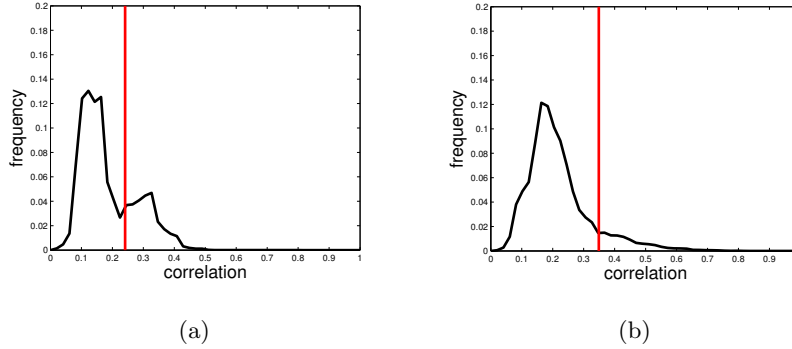
Fig. 2: Correlation distribution and correlation limit of (a.) random colluding selections and (b.) intelligent colluding selections.

consequence because it allows each node to store more attacker selections at one time. An immediate drawback to this approach is that it has a two-fold impact on system performance. Increasing the size of the tunnel will increase connection latency as messages will need to be routed through more nodes. Increasing the size of the L_{ES} will require greater storage and more computation for each execution of the CDM during tunnel construction.

Alternatively, one might introduce variable length tunnels into MorphMix. If an attacker doesn't initially know the true length of the tunnel, it is more difficult to determine if he owns the first and last nodes; however, tunnel length is limited by a small range of realistic values. The analysis in [18] noted that while longer tunnels (eg. 10 nodes) offer greater protection than shorter tunnels (eg. 3 nodes), they also incur a higher connection latency and result in higher bandwidth usage by the MorphMix network. They also will increase the chances of tunnel failure if a node leaves the network or purposefully breaks a connection. Taking this into account, an attacker can estimate the distribution of tunnel lengths in MorphMix and the probability that it has compromised the entire tunnel. Variable length tunnels may slightly weaken our attack, but they do not eliminate it.

New users to MorphMix are especially vulnerable to the intelligent selection attack. Since new users enter the system with an empty L_{ES}, attackers are guaranteed to successfully compromise a significant portion of a new user's initial tunnels, regardless of the L_{ES} size. This type of initial behavior in MorphMix will presumably limit its adoption. Most importantly, neither of these methods prevents the attack, and instead, only delays its success. As described in Section 4.2, attackers can optimize their attack strategy so that they can still build a significant number of anonymous tunnels given fewer unique attack selections.

The general limitation of the CDM is that it only considers a node's local knowledge when detecting collusive behavior. Specifically, it distinguishes between honest and colluding selections based only on the selections the individual node has previously seen, not taking into account the behavior of the rest of the

network when calculating its own correlation limit. Also, because nodes evict the oldest entries from their L_{ES}, attackers can estimate when a victim's L_{ES} will be cleared of attacking selections and then once again begin the attack. These two factors make it easy for attackers to not only model and manipulate what a node has stored in its L_{ES}, but improve their attack strategy based on this information. Although the CDM may be adjusted to capture some possible attack strategies, attackers can stay one step ahead by modeling the state and algorithms of the CDM at each node and crafting the best possible response, consisting of both honest and colluding selections.

An effective collusion detection mechanism for MorphMix requires a more global perspective of the network. One instance of this might be to enforce a double check on any offered selection during tunnel construction. Every time a tunnel initiator wants to append a node to its tunnel, he contacts a unique witness to help establish the symmetric key between the initiator and the new end node. Additionally, it is the witness that chooses which node from the offered selection to use for the next hop. Requiring that a selection correlation fall beneath the correlation limit of the initiator and witness when appending a node may double the chances of it being detected; however, it may also adversely affect the false positive rate when evaluating honest selections. Nevertheless, while this may improve the mechanism's detection rate, it still doesn't provide a thorough view of network behavior and more sophisticated schemes are likely needed.

6   Related Work

6.1   P2P Anonymous Systems

In response to some of the weaknesses of single point proxy systems and centralized mix networks, attention turned toward distributed solutions to anonymous networking. Crowds [16] aims to provide anonymity to people using the Internet by blending and forwarding web requests among other users in their "crowd". Because no user can distinguish between receiving a web request from an initiator or just another forwarding user, sender anonymity is preserved. Hordes [11] is a variant of Crowds that uses multicast services to anonymously route replies back to the initiator. From a high level, both Crowds and Hordes provide anonymity through plausible deniability because each user issues requests on behalf of other unidentifiable users in the crowd. Both systems are examples of condensed P2P systems and use a central directory to keep track of users currently in the crowd. Whenever a user enters of leaves the system, each node in the crowd must be updated with this change in status. A major disadvantage to this approach is it severely restricts the number of users a crowd can support; thus, these systems are only appropriate for small sized networks.

Mix networks, on the other hand, enforce anonymity by providing sender and receiver unlinkability. Tarzan [9] is a P2P overlay for anonymous networking. Like MorphMix, each Tarzan client is a mix. Users achieve anonymity by using layered encryption and multi-hop routes relayed through other Tarzan nodes. Distinctive to Tarzan, each user selects its own route through a restricted set of nodes and cooperates in system cover traffic to prevent initiator identification from traffic analysis. To learn about other nodes in the system for

anonymous routes, Tarzan users continually contact their peers and download current neighbor lists which provide each Tarzan node with a shared global view of the network. This approach, however, severely limits the scalability of the system. Tarzan peer selection is similar to MorphMix in that peers are chosen among distinct IP prefixes instead of their whole IP address; however, no additional collusion detection mechanism is present in the system.

6.2 Collusion Detection

The problem of detecting colluding adversaries in distributed systems is not unique to MorphMix. Without a trusted central authority, it was shown in [8] that large P2P systems are vulnerable to "Sybil attacks" in which a small number of entities can present multiple identities and compromise a disproportionate share of the system. While the Sybil attack has been extensively studied in the area of ad-hoc wireless and sensor networks [13], it has been less threatening to real-world P2P systems. MorphMix, for example, obviates the threat of Sybil attacks by using each node's IP address as an implicit certification that it is unique within the system.

Collusion detection, however, is still very much an open problem. It has been more specifically examined in the realm of reputation based systems, such as information retrieval on the web [14] and P2P file sharing networks. In [22], they study the effects of collusive behavior to improve Google page rank of indexed web pages and propose modifications to the page rank algorithm to prevent this type of gaming. The Eigentrust algorithm [10] for P2P file sharing networks provides a way to compute a global trust value of peers based on their peer interactions in the system. One limitation of this approach is the requirement of universally trusted root nodes, a feature often lacking in most P2P systems. The authors in [5] propose a reputation management protocol, P2PRep, for peers participating in P2P file sharing networks. Their protocol consists of weighted voting according to peer credibility that determines the reputation of other peers in the network. As opposed to peer reputation, Credence [21] attempts to deter pollution in file sharing systems by computing reputations on the actual shared information as opposed to the peers.

While the success of these P2P approaches is promising, many seem specifically suited for file sharing networks where the ultimate goal is detecting malicious content in the system. Also, there is often clear evidence of misbehavior in file sharing networks, however, this is rarely the case in anonymous networking systems. It is still uncertain if reputation management schemes can be applied to anonymous systems like MorphMix without disrupting node anonymity and user unlinkability within the system.

7 Conclusion

We have presented an attack to MorphMix that breaks the collusion detection mechanism during anonymous tunnel construction, thus devastating the anonymity guarantees initially proposed by the system. We assume an internal adversary of different resource levels and demonstrate that this attack can

successfully compromise many tunnels in both a strong and weak setting. This attack highlights an inherent weakness in the MorphMix CDM. Namely, the mechanism only considers a node's local view of the network when detecting collusive behavior. This allows attackers to model a victim's local knowledge and manipulate its content to prevent detection of compromised anonymous tunnels.

Our results show that MorphMix does not effectively address the problem of detecting colluding nodes in peer-to-peer anonymous networks and this problem is worthy of future research. Peer-to-peer approaches such as MorphMix are currently the only solution that can scale to very large numbers of users and offer a promise of a truly global and widely used anonymous communication infrastructure. Therefore, solving the problem of collusion is an important step towards widespread adoption of anonymity technologies.

8 Acknowledgements

References

1. eDonkey File Sharing System, 2003.
2. Oliver Berthold, Hannes Federrath, and Stefan Köpsell. Web MIXes: A System for Anonymous and Unobservable Internet Access. In *Workshop on Design Issues in Anonymity and Unobservability*, pages 115–129, 2000.
3. R. Bhagwan, S.Savage, and G. Voelker. Understanding Availability. In *2nd International Workshop on Peer-to-Peer Systems*, 2003.
4. David Chaum. Untraceable Electronic Mail, Return Addresses, and Digital Pseudonyms. *Communications of the ACM*, 24(2):84–88, 1981.
5. Fabrizio Cornelli, Ernesto Damiani, Sabrina De Capitani di Vimercati, Stefano Paraboschi, and Pierangela Samarati. Choosing Reputable Servents in a P2P Network. In *WWW*, pages 376–386, 2002.
6. George Danezis, Roger Dingledine, and Nick Mathewson. Mixminion: Design of a Type III Anonymous Remailer Protocol. In *Proceedings of the 2003 Symposium on Security and Privacy*, pages 2–15. IEEE Computer Society, May 11–14 2003.
7. Roger Dingledine, Nick Mathewson, and Paul F. Syverson. Tor: The Second-Generation Onion Router. In *USENIX Security Symposium*, pages 303–320, 2004.
8. Douceur. The Sybil Attack. In *International Workshop on Peer-to-Peer Systems (IPTPS), LNCS*, volume 1, 2002.
9. Freedman and Morris. Tarzan: A Peer-to-Peer Anonymizing Network Layer. In *SIGSAC: 9th ACM Conference on Computer and Communications Security*. ACM SIGSAC, 2002.
10. Sepandar D. Kamvar, Mario T. Schlosser, and Hector Garcia-Molina. The Eigentrust Algorithm for Reputation Management in P2P Networks. In *WWW*, pages 640–651, 2003.
11. Brian Neil Levine and Clay Shields. Hordes: a Multicast-Based Protocol for Anonymity. *Journal of Computer Security*, 10(3):213–240, 2002.

12. Steven J. Murdoch and George Danezis. Low-Cost Traffic Analysis of Tor. In *IEEE Symposium on Security and Privacy*, pages 183–195, 2005.

13. James Newsome, Elaine Shi, Dawn Song, and Adrian Perrig. The Sybil Attack in Sensor Networks: Analysis & Defenses. In *Proceedings of the Third International Symposium on Information Processing in Sensor Networks (IPSN-04)*, pages 259–268, New York, April 26–27 2004. ACM Press.

14. Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The PageRank Citation Ranking: Bringing Order to the Web. Technical Report SIDL-WP-1999-0120, Stanford University, November 1999.

15. Peter Palfrader. Number of Running Tor Routers, 2006.

16. Reiter and Rubin. Crowds: Anonymity for Web Transactions. *ACMTISS: ACM Transactions on Information and System Security*, 1, 1998.

17. Rennhard and Plattner. Practical Anonymity for the Masses with MorphMix. In *FC: International Conference on Financial Cryptography*. LNCS, Springer-Verlag, 2004.

18. Marc Rennhard. PhD thesis, Swiss Federal Institute of Technology Zurich.

19. Marc Rennhard. MorphMix prototype v0.1, 2004.

20. Marc Rennhard and Bernhard Plattner. Introducing MorphMix: Peer-to-Peer Based Anonymous Internet Usage with Collusion Detection. In *WPES*, pages 91–102, 2002.

21. Kevin Walsh and Emin Gun Sirer. Fighting Peer-to-Peer SPAM and Decoys with Object Reputation. In *Proceedings of the Third Workshop on the Economics of Peer-to-Peer Systems (P2PECON)*, 2005.

22. Zhang, Goel, Govindan, Mason, and Van Roy. Making Eigenvector-Based Reputation Systems Robust to Collusion. In *International Workshop on Algorithms and Models for the Web-Graph (WAW), LNCS*, volume 3, 2004.