Compulsion Resistant Anonymous Communications

George Danezis and Jolyon Clulow

University of Cambridge, Computer Laboratory,
William Gates Building, 15 JJ Thomson Avenue,
Cambridge CB3 0FD, United Kingdom.
{George.Danezis, Jolyon.Clulow}@cl.cam.ac.uk

Abstract. We study the effect compulsion attacks, through which an adversary can request a decryption or key from an honest node, have on the security of mix based anonymous communication systems. Some specific countermeasures are proposed that increase the cost of compulsion attacks, detect that tracing is taking place and ultimately allow for some anonymity to be preserved even when all nodes are under compulsion. Going beyond the case when a single message is traced, we also analyze the effect of multiple messages being traced and devise some techniques that could retain some anonymity. Our analysis highlights that we can reason about plausible deniability in terms of the information theoretic anonymity metrics.

1 Introduction

Research on anonymous communication and mix networks has in the past concentrated on protecting users' anonymity against an eavesdropping adversary. Chaum in his seminal paper [7] argues that mix networks are secure against a global passive adversary, able to eavesdrop on all the communication links. Recent circuit based systems, such as Tarzan [16], MorphMix [22] and Tor [15], provide security against a partial passive adversary, that can only eavesdrop on part of the network. Some designs [12] additionally address active adversaries, that can modify data to gain information about the obscured routing of a message. Finally mix systems protect against corrupt intermediaries by distributing the mixing functionality across many nodes in the network.

Assuming a partial eavesdropping adversary, controlling only a small subset of network nodes one can show that very little information is leaked about the correspondence between actual senders and receivers of messages. It is often argued that such a threat model is more realistic than a global passive adversary, that is prohibitively expensive in a wide area network. On the other hand it does not encompass one of the most common threats, namely the *threat of compulsion of an honest node*.

We will study the impact on anonymity if an adversary is able to ask for honest nodes' private keys or the decryption of arbitrary material. Such an adversary,

if unbounded, would have a devastating effect on the security properties of a traditional mix network: they would be able to decrypt all layers of all packets and recover all hidden identity information. For this reason it is more instructive to think of such attacks in terms of the cost necessary to de-anonymize a message, *ie.* the number of honest nodes under compulsion necessary to achieve it.

In this work we introduce techniques to make compulsion attacks both more expensive, more visible and reduce the information they provide. The cost of compulsion attacks is raised by introducing some uncertainty, though multicasting steps, in the routing. Given that routing of messages is cheaper than compulsion attacks, this penalizes the adversary much more than it hinders the normal operation of the system. This technique bears some resemblance to source routed cover traffic.

Often it is assumed than an adversary is *shy* and would prefer the attack not to be known, particularly to the ultimate target. This is reasonable since advance warning of being under surveilance would allow one to destroy evidence, hide, or lie. We describe techniques that make it difficult to hide the fact that a packet is being traced, through using *compulsion traps*, loops in the routing that are designed to give advance warning of a successful attack.

Finally we make the results of an attack much less certain, by allowing both intermediate nodes to plausibly lie, and the final node to pretend to be merely an intermediary. An adversary is in these cases not able to find any bit string that would contradict these claims. This allows mix systems to provide initiator/receiver anonymity properties, and retain some anonymity, even if all nodes are under compulsion.

We used established research into measuring anonymity, and show that resistance to compulsion and plausible deniability can be measured within that framework. In other words compulsion is simply another form of attack that leads to the system providing varying degrees of anonymity depending on the intensity. We also show that traffic analysis, the tracing of multiple messages, has a devastating impact even against hardened systems. To protect against it we should question the conventional wisdom of relaying messages through completely random nodes.

2 The Compulsion Threat Model

The traditional anonymous communication threat model assumes an adversary that can eavesdrop or even modify traffic on a proportion of the network links. Furthermore it assumes that some proportion of the system nodes are subverted and directly controlled by the adversary. This threat model has been used in the context of cryptological research for some time, and can indeed express a wide spectrum of threats. On the other hand it suffers some biases from its military origins, that do not allow it to define some very important threats to anonymous communication systems.

Anonymous communication systems are often deployed in environments with a very strong imbalance of power. That makes each of the participants in a network, user or intermediary, individually vulnerable to *compulsion attacks*:

> A node under compulsion has to provide a service or information that is, to the adversary, indistinguishable from the genuine one. We will call protocols that allow a different service or information to be provided other than the one under normal operation, compulsion-resistant.

These compulsion attacks are usually expensive for all parties and cannot be too wide or too numerous.

A typical example of a compulsion attack would be a court order to keep and hand over activity logs to an attacker. Such a legal challenge led to the closure of the first remailer `anon.penet.fi` [18, 19, 17]. This can be targeted at particular intermediaries or can take the form of a blanket requirement to retain and make accessible certain types of data. Another example of a compulsion attack could be requesting the decryption of a particular ciphertext, or even requesting the secrets necessary to decrypt it. Both these could equally well be performed without legal authority by just using the threat of force.

Parties under compulsion could be asked to perform some particular task, which bears some similarity with subverted nodes. For example, this is an issue for electronic election protocols where participants might be coerced into voting in a particular way.

Note that compulsion and coercion cannot be appropriately modeled using the concept of subverted nodes from the traditional threat model. The party under compulsion is fundamentally honest but forced to perform certain operations that have an effect which the adversary can observe either directly or by requesting the information from the node under compulsion. The information or actions that are collected or performed under coercion are not as trustworthy, from the point of view of an adversary, as those performed by a subverted node since the coerced party can lie and deceive in an attempt not to comply. At the same time system designers should not assume that honest parties will always lie when under compulsion simply because they can – it is more prudent to assume that only nodes benefiting from deceiving the adversary are required to lie. Good compulsion-resistant designs will maintain their security properties even when all other, non-interested nodes, are cooperating with the adversary.

Election protocols [8, 1] are specifically designed to allow voters to freely lie about how they voted, and *receipt-freeness* guarantees that there is no evidence to contradict them. Other protocols and systems attempt to provide *plausible deniability*, the ability to deceive the coercer and reveal partial or wrong information. Forward secure communications [6] guarantee the privacy of past conversations, by updating and deleting old keys. Forward secure signature schemes [2] make sure that signature keys leaked cannot be used to sign valid documents in past epochs. The steganographic file system [3] allows users to deny the existence of some stored files, while chaffinch allows users to deny the existence of some communication stream [9].

3 Introduction to mix systems

A mix, as introduced by David Chaum [7], is a network node that hides the correspondence between its inputs and outputs. It does that by receiving encoded inputs that it decodes using a private key, to outputs that are bitwise unlinkable with the inputs. Furthermore in order to disrupt the timing characteristics of traffic, several messages are batched together before being decoded and sent out in a random order.

In order to prevent a single corrupt mix compromising the anonymity of all messages traveling though it, a chain of mixes can be used. As long as one of them is honest, the message will be provided with some anonymity. The way one can select the series of mixes is called the topology of the network. If only one sequence is possible, we call such a network a cascade, and conversely if all possible sequences are permitted we call the network free route [4].

Technically a message to travel though a mix network has to be encoded using the public keys of all intermediate mix nodes. This is done by recursively encrypting the message in layers, starting with the last node in the path, and ending with the first. Each layer contains some routing information that allows the message at each intermediary to be routed to the next mix. It is also conventional to include a session key, and perform most cryptographic operations using symmetric primitives rather than asymmetric, and therefore expensive, operations.

Aside from the sender encoding messages to send them anonymously though the network, mix network can be used to route anonymous replies. In this case the sender includes a bit-string in the anonymous message that can be used by the receiver to send a message back. This process does not reveal any information about the identity of the original sender. We call these bit stings, anonymous reply blocks, and they are constructed in a similar way to messages, but with no payload. They contain the final address of their recipients, recursively encoded under the keys of intermediate mixes, with routing information in each layer. To route a message using a reply block, the message is appended to the block, and sent though the network. As it travels each intermediary mix decodes the reply block (with the last mix retrieving the final address), while at the same time encoding the reply message. This guarantees unlinkability, since both parts of the message are changed, and allows the final recipient, to decode the message since it knows the secret contained in the reply block, that he manufactured.

A good mix packet format would be expected, aside from providing bitwise unlinkability, to have a set of other properties. It should hide the total path length and the position on the path from intermediate mixes. It should also provide replies that are indistinguishable from other messages, and be resistant to tagging attacks. Both Mixminion [12] and the newer Minx [13] packet formats provide these properties.

Figure 1 illustrates the routing of normal sender anonymous messages, as well as anonymous replies. The notation $[x]_y$ means that message x is encoded under key y. The keys with capital letters subscripts $(K_A, \ldots, K_F)$ are the public keys of the corresponding nodes, while keys with lowercase subscripts

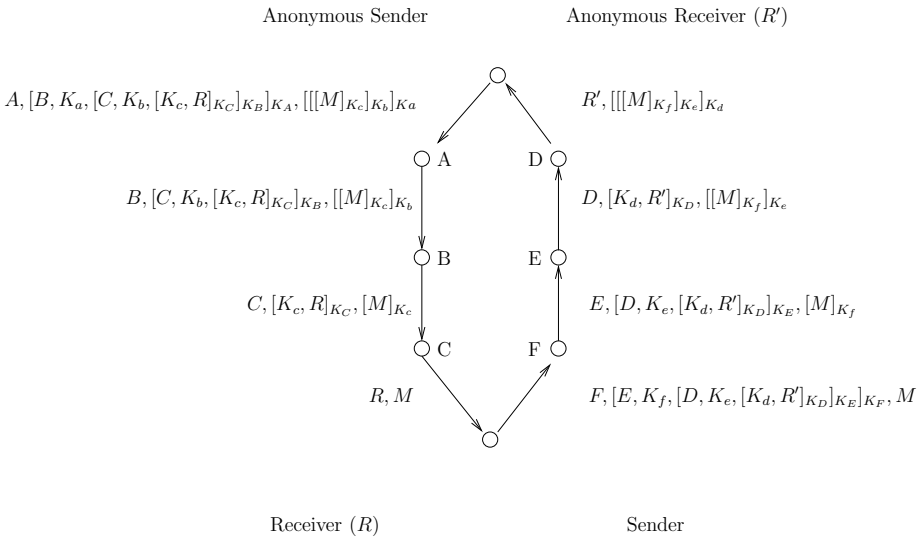

Fig. 1. Sending and replying to anonymous messages

$(K_a, \ldots, K_f)$ are symmetric session keys. The reply block, that the anonymous sender has constructed and included in the message M is the string: $F, [E, K_f, [D, K_e, [K_d, R']_{K_D}]_{K_E}]_{K_F}$. The figure abstracts away quite a few crucial details, such as the exact encoding scheme, the padding added to keep messages constant length, the duplicate detection mechanisms that discard already seen messages. The full details of how to engineer such schemes can be found in [12, 13]. Nevertheless this simplified model allows us to describe the vulnerability to compulsion attacks, as well as ways of defending against them.

4 Using compulsion to attack mix systems

The security of mix systems relies on the correspondence between inputs and outputs of some mix on the message path to remain hidden. There is no cryptographic way of ensuring this (since it is impossible to prove that information was not leaked), and therefore some degree of trust is placed on the mixes. Even if these are not subverted and controlled by the adversary, an attacker can compel them to either decode a ciphertext or hand over any key material they possess.

Both types of compulsion attacks assume that the adversary is in possession of a ciphertext to decode. These can be intercepted by a global passive adversary, or even by a partial eavesdropping adversary that watches all anonymous messages sent by a particular user. Note that there is little value in intercepting messages in the middle of the network since they have already been stripped of information that could lead to their sender. Therefore an eavesdropper will try to intercept messages as close to their senders as possible, and then sequentially compel intermediate mixes to decode them. After a number of compelled mixes

equal to the path length, an adversary will be able to link the targeted sender with the receiver of their messages.

Roger Dingledine [14] was first to point out that this mechanisms makes reply blocks more vulnerable to attack, than normal messages. The attacker has no need to eavesdrop on the network to get a first packet since the reply block is readily available and encodes all information needed to trace its creator. Therefore given a single use reply block and a number of compelled nodes equal to the path length, the identity of the user that has sent the message containing the reply block is revealed. Since this is the simplest and most devastating compulsion attack we will concentrate on making it more difficult.

Many stream based system such as Tor [15], MorphMix [22] and Cebolla [5], avoid using onion encrypted messages and reply blocks by recursively opening a bidirectional channel though the intermediaries. This architecture is very effective for supporting streams initiated by the anonymous client towards a server. It does not support very well streams initiated by a client towards an anonymous receiver, since the channel must be kept open (either using real network connection, or virtual labels and circuits), and connected to a well known public server. In particular Tor's implementation of 'Rendezvous point', are still susceptible to compulsion attacks. Since the connection though the tor network has to be kept alive all the time, an adversary only needs to compel each node in the route to reveal their predecessor, until the hidden service is uncovered.

We will see how to strengthen traditional mix systems against such compulsion attacks.

5 Protecting mix systems

In this section we present the technical details of the constructions we devised in order to strengthen mixed communications against compulsion attacks and in particular reply blocks, since they are the most vulnerable to this attack. We first consider the case of tracing a single reply block and then, in section 6, tracing multiple reply blocks. We will make some assumptions about the nature of the mix network, but present our solutions in the context of the abstract mix architecture of figure 1.

First we assume a peer-to-peer mix network, where all clients are also mix servers for other nodes. This is a challenging design for reasons not related to compulsion. The main difference from traditional, client-server, mix networks is the need to distribute the full list of all participating nodes and their keys at all times. A failure to distribute the whole list might result in attacks (as described in section 4.2.7 of [11].) Furthermore this has to be done in a way that is not manipulable by an adversary that tries to flood the network with corrupt nodes. Implementing such an infrastructure and key distribution in wide area networks are active, but separate, subject of research.

The abstract model of a mix net we use can in practise be implemented using the deployed Mixminion [12] or the proposed Minx [13] packet formats. These have been designed with some common requirements in mind: they allow

multiple mix nodes to relay an encoded mix packet, making sure that the packet is cryptographically unlinkable at different hops. Single use reply blocks can be used to reply to anonymous senders, and their transport is indistinguishable from other packets. Both systems hide the total length of the chain of mixes, and the position of each mix node on the path of the message. Finally they are not vulnerable to tagging attacks – if a message is modified in an attempt to gain information it ends up getting discarded.

On the other hand the respective behavior of the two different, mixminion and minx, formats is different when it comes to discarding 'malformed' or potentially 'modified' messages. A mixminion node will discover that a message header is modified, using a message digest, and discard the message immediately. If the body of the message only has been modified then it will be forwarded, an all-or-nothing transform will take care of destroying the message contents, when it reaches a 'swap point'. On the other hand Minx nodes cannot tell if messages have been modified, since they do not contain any redundancy at all, and are indistinguishable from noise. The decoding procedure just ensures that modified messages turn into random bit-strings, and are therefore routed using a random walk around the network, until they are randomly discarded. Mixminion does never randomly route messages around since all routing information has to be well formed. These differences will have some impact on how the compulsion resistance mechanisms are implemented.

5.1 Multicast steps

First we note the asymmetry between the cost of relaying a message and the cost of compelling a node to reveal secrets. Relaying consumes a bit of bandwidth and computation time. On the other hand compelling a node can only happen after a prolonged conflict, legal or physical, with a very high cost for all parties. We shall use this asymmetry to protect against compulsion.

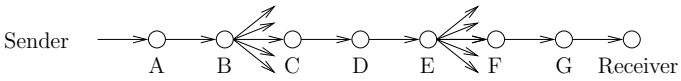

Fig. 2. Routing with multicast steps

A standard message travels through a mix network using a sequence of intermediate mix nodes. The routing information is encrypted to intermediary nodes, and they have to use their secrets in order to decrypt it. A way of pushing up the cost of compulsion would be to include some *multicast steps* into the routing, where the message is sent to a set of nodes at once. Only one of these nodes has the necessary secret to correctly decode the message, and continue routing it. The adversary, or even the node that is compelled to surrender the decoded message, has no way of knowing which of the nodes included in the multicast is

the correct one. Therefore the adversary will have to try them one by one until a correct decryption is provided to trace that step.

In the case of Mixminion the nodes receiving a message that cannot be decrypted using their secrets will discard it immediately. As a result an adversary compelling them can be sure that they are not the nodes expected to route the message, since he can ask for all secrets and check for a well formed message. For each multicast step, multicasting the message to K nodes, the cost to the mix network during normal operation is $\mathcal{O}(K)$. On the other hand the adversary will have to query the multicast nodes one by one until one of them decodes the message correctly. This requires the adversary to query on average about $\frac{K}{2}$ nodes per multicast step until a correct decryption is provided.

If Minx is used to implement *multicast steps*, neither the nodes nor an adversary that compels them can tell if the message was intended for them. They will simply decode it and pass it along. This results in an exponential growth of the effort required to process the message, that makes this scheme impractical in the case of Minx. At the same time it also means that the adversary has to compel an enormous number of nodes, hoping that one of them will be the final node relaying the message.

5.2 Compulsion traps

Some adversaries would prefer to trace messages using compulsion, without the ultimate recipient of the message knowing. Often this would allow the target to eliminate evidence, to destroy key material, and physically hide. We shall therefore assume that our adversary is *shy* and forces mix nodes not to hide whether they are under compulsion.

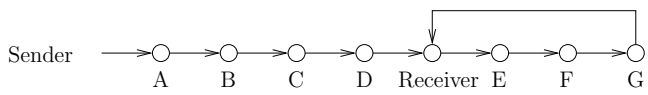

Fig. 3. Routing topology of a compulsion trap

We modify the path selection procedure, that we use to select intermediary nodes in the mix network, to provide some advance warning of an attack. The sender of the message or reply block includes itself on the path that their own message is routed through – we call this a *compulsion trap*. As a result a reply block that is being traced back would require the attacker to compel the target to decode the message. The target can do this and provide a valid message, that still has to get routed to reach its ultimate destination. The adversary has no way of knowing which node on the path (aside from when it reaches the last) is the target, while the target gets some advance notice that tracing is taking place.

The overheads that this technique imposes on the system are negligible during the normal operation of the system. When a reply block and a message are being

routed through the system, the receiver can decode the message the first time it sees it, since it can recognize the reply block as his. There is no need to route the actual message through the loop, since it eventually leads back to the same user. On the other hand if an adversary is tracing the message, the target node will provide a decryption, and force the adversary to compel more nodes, until he is eventually led back to the target. This means that more compulsion operations have to be performed by the adversary, than hops when honest nodes relay the message. During normal operation the message is not relayed for more hops than a message that is not using this scheme, and therefore the latency of messages is also not affected by this scheme.

5.3 Plausibly deniable routing

Compulsion traps might give an advance warning of a compulsion attack being performed against the network, but do not allow the ultimate recipient of a traced reply block to remain anonymous in the long run. As we have seen the normal operation of the mix network does not require the reply message to travel through all nodes specified, but only until the first occurrence of the recipient node. The question then arises: why including the same node again further down in the path? There is no reason, and one can include a random selection of other nodes instead.

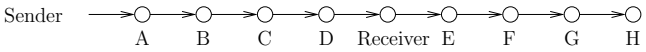

Fig. 4. Plausibly deniable recipient

The resulting construction, of including in the routing information a tail of random nodes, provides *plausible deniability*. That means that the adversary cannot determine with certainty that a particular node in the path was the intended recipient of the message. All nodes can provide a plausible theory to explain why they have been involved in relaying the message: most can claim that they were merely relaying the message. The last node can explain that they were just selected at random – which is true, and plausible given this strategy in known to be used. Note that the only node to be lying is the actual receiver and there is no information held by the adversary, that can contradict the false claim that the message is merely being relayed at this point[1].

Plausible deniability was introduced by Michael Roe [23], as the property that is complementary to non-repudiation, the inability to deny something. Early

[1] Showing that a node is the actual receiver of the message would involve proving that they know a subset of the symmetric keys K_f, K_e, K_d, R' of figure 1. These can be generated on the fly using a master passphrase, and unless they are stored the claim will be difficult to prove. This assumption does not require the use of secure hardware for the receiver or any other nodes.

work has assumed that unless something is digitally signed, it is plausibly deniable. This is not quite true since other evidence, such as logging or the use of a time stamping service, can still produce a very high degree of confidence in an action or a sequence of events, that will make it difficult to deny it.

At the same time little effort has been made to quantify the degree of plausible deniability. When it comes to an actor trying to deny they performed an action, plausible deniability can be measured using established measures of anonymity [24]. Therefore the plausible deniability property proposed can be analyzed as an anonymity property. We can at each stage of the compulsion attack assign a probability to each actor in the network of being the receiver. Then we use the established measure of anonymity, which is the entropy of this probability distribution, to assess how much plausible deniability is provided [24].

We shall compute the anonymity of the proposed scheme as a function of the compulsion effort of the adversary. We denote the number of hosts under compulsion as k. We assume there are N participants, in a peer-to-peer mix network. To simplify things we also assume that the total route length is of a fixed size l. The real recipient of the message was an equal probability of being at any position, while the other relays are chosen at random.

After k mixes have been forced to decode the reply block, and provide the adversary with the next recipient, the adversary knows k candidates that are equally likely to be the receiver. All other $N - k$ nodes also have an equal probability of being the receivers in case he is not in the set of nodes under compulsion. This is the case with probability $\frac{l-k}{l}$, namely the probability the target has chosen a position on the route further away from the part that has been traced back. The probability distribution that describes how likely each node is to be the receiver is the following (after k nodes under compulsion):

$$\Pr[i|k] = \begin{cases} \frac{k}{l}\frac{1}{k} & \text{if } i \text{ in the compelled set} \\ \frac{l-k}{l}\frac{1}{N-k} & \text{otherwise} \end{cases} \tag{1}$$

We can easily calculate the entropy ($\mathcal{H}$) of this distribution ($U(x)$ denotes the uniform distribution over x elements).

$$\mathcal{H}(\Pr[i|k]) = \mathcal{H}\left(\frac{k}{l}, \frac{l-k}{l}\right) + \frac{k}{l}\mathcal{H}(U(k)) + \frac{l-k}{l}\mathcal{H}(U(N-k)) \tag{2}$$

$$= \log l + \left(\frac{l-k}{l}\right)\log\frac{N-k}{l-k} \tag{3}$$

This formula is in line with our intuitions. When there is no compulsion ($\mathcal{H}(\Pr[i|0])$) then the effective anonymity set size is equal to $\log N$, since all the network participants are equally likely to have been the receiver. The adversary is missing $\log N$ bits of information to uniquely identify the receiver. When all nodes in the route have been compelled we have $\mathcal{H}(\Pr[i|l]) = \log l$, since only the compelled nodes are equally likely to be the ultimate receivers of the message. Note that even when all participants are under compulsion, there is still some anonymity left.

6 From single message tracing to traffic analysis

We have looked in detail how tracing a single message would be more expensive if our countermeasures are used, and calculated the anonymity that would remain. Serious traffic analysis usually tries to infer the identity, or other characteristics, of communicating parties using repeated patterns of communication. We therefore need to assess the security of the proposed countermeasures against a compelling adversary that traces back multiple reply blocks.

Like in Crowds [21] allowing the actual receiver to be present at any position of the path opens the *compulsion trap* and *plausibly deniable routing* to predecessor attacks as first presented by Wright et al. [26] and analyzed further by Shmatikov [25]. Since mixing is taking place, one could try to skew the probability distribution describing the placement of the receiver on the path to gain some security against this attack. This would not add any security against compulsion attacks (beyond making compulsion slower to the same degree as the latency increases), and therefore we shall not discuss this countermeasure any further. Instead we will analyze the security of the base scheme and present in section 6.1 an alternative solution that relies on routing amongst a fixed set of 'friends'.

In the case of the *multicast steps* and the *compulsion trap* constructions, and adversary that compels enough nodes will eventually reach and identify the ultimate recipient. Therefore traffic analysis of multiple reply blocks can only be expected to yield a performance benefit (minimize the effort of the attacker).

In the case of multicast steps there is no such performance benefit for the attacker, and the optimal strategy is to trace a single block until the final node, which will also be the receiver. When attacking a compulsion trap based system tracing two single use reply blocks in parallel might prove to be cheaper. Assuming that all relay nodes are chosen at random, and the receiver will appear on both paths, a node that appears in both paths is the receiver with a very high probability.

We shall use the setup of figure 3, to illustrate how an adversary should decide between tracing one reply block until the end, or tracing two in parallel, when compulsion traps are used. We will assume that all reply blocks have a total path length of l, and the compulsion trap, the position in the route where the receiver node insert itself, is $k \in [1, l-1]$ (node that in figure 3 the last node is always the receiver), and it follows a uniform distribution over all positions $k \sim U(1, l-1)$ and therefore $\Pr[k] = \frac{1}{l-1}$. On average the compulsion trap point k, where the receiver has included himself in the path, will be at:

$$E[k] = \sum_{k=1}^{k=l-1} k \Pr[k|k \sim U(1, l-1)] = \frac{1}{l-1} \sum_{k=1}^{k=l-1} k = \frac{l-1}{2} \tag{4}$$

An attacker tracing two reply blocks in parallel will expect to have to compel $K_1 + K_2$ nodes, where both are independent random variables that follow the uniform distribution above, until it reaches the same recipient node, in both paths. Since $E[K_1 + K_2] = 2E[k] = l - 1$, a attacker will do marginally better by

tracing two reply blocks in parallel. Tracing a single reply blocks is guaranteed to pay off after l compulsion steps. This slight efficiency improvement might be offset by the possible false positives, that would be the result of the same node appearing randomly in the two paths without being the receiver. This probability, in our example, equals (from section 2.1.5 of [20], that uses the notation $m^{(n)} = m(m-1)\dots(m-n+1)$):

$$P_{\text{false}} = 1 - \frac{N^{(2k)}}{\left[N^{(k)}\right]^2} \tag{5}$$

Note how the probability of false positives goes quickly towards zero as N grows.

The method we presented that provides *plausible deniability* guarantees some anonymity even when all nodes on the path of one reply have been compelled. When two reply blocks to the same receivers are available an adversary reduces greatly the anonymity of the receiver. A second couple of nodes, one in each path, that are the same is extremely unlikely as N grows:

$$P_{\text{false}} = 1 - \frac{N^{(2(l-1))}}{\left[N^{(l-1)}\right]^2} \tag{6}$$

We should conclude that given the above models a mix system, even if it implements our countermeasures, can be subject to traffic analysis attacks to uncover the ultimate receiver of reply blocks. Therefore we must modify the way nodes are selected on the path to recover some plausible deniability and some anonymity even when faced with overwhelming compulsion. There are two ways in which one can select nodes to include in the path to enhance anonymity, the first is to route amongst a smaller group of friends, the second is to setup stings, that look to an attacker performing traffic analysis like the final receiver.

6.1 Routing amongst friends

As the network of mix participants (and therefore nodes) grows, traffic analysis attacks become more certain. The probability a random node is on the path of a reply block twice, becomes smaller, and as a result the probability of the attacker observing a false positive decreases quickly to zero. In order to strengthen the network against such attacks it might be worthwhile to form routes in a non random manner. The objective being to maximize the number of common nodes in the paths of different reply blocks.

First note that a particular node choosing routes that always include a fixed set of nodes foils the traffic analysis attacks. These set nodes can be arranged in a cascade but this is not necessary: they simply need to be present at some point, before or after the receiver, on the path.

To avoid the attacks against route selection described in section 4.2.7 of [11] the choice of the fixed set must remain private to the creator of the reply block. Care should also be taken for the set not to be inferred easily by corrupt nodes or an observer (although this goes beyond the strict compulsion threat model).

One way of inferring the set of 'friends' used is to observe the destination of messages sent by the user, or the origin of those received. Nodes that are always used will appear with much higher frequency than other ones. This can allow an adversary to infer the membership of this set. In turn this allows corrupt nodes on the network to infer that connection between these nodes are very likely to be carrying messages to the particular receiver.

A mixed approach, where a set of fixed nodes is used in conjunction with random nodes seems preferable. The reply block paths are constructed in such a way that between each node from the fixed 'friends' set there is a randomly chosen node. This prevents an observer or a corrupt node from trivially inferring the membership of the fixed set.

Using a fixed set of nodes of size F, guarantees that under any circumstances, including the tracing of multiply reply blocks, the effective anonymity set size will be at least $-\log(F+1)$. An attacker will not be able to distinguish between the actual receiver and the fixed set of nodes that is always used to route traffic. The additional security provided does not come for free: all F nodes must be on the path, along with another F random nodes. Therefore the latency will be on average proportional to the minimum degree of anonymity required. This fact might influence the design decisions behind systems that aim to prevent compulsion to provide less mixing (less latency per mix) and opt for longer routes instead.

6.2 Setting someone else up

An attacker that naively relies on compulsion in order to infer the final recipient of a reply block must be very cautious. In the case of *compulsion traps* construction, where the final receiver is meant to be twice on the path of the message, it is easy to incriminate another node. A user can create a path that contains a loop at a different position than itself. For example the path [A, B, C, D, Receiver, E, F, D], contains a loop, that might look to the adversary as a compulsion trap when actually node D is unaware that the message is traveling twice through it (since it does not know all the keys that are necessary to decode the reply block and recognize it at each phase).

For each technique that the adversary could use for tracing a reply block (except the traffic analysis of different reply blocks), the adversary could create such a structure to convince the adversary that someone else is the receiver. This provides even more plausible deniability if the adversary does trace the message using compulsion and makes an inference as above.

7 Conclusions

We have presented the effects that compulsion powers might have in undermining the security of a mix network. Our analysis is based on an abstract model of how a mix network works and should be applicable to a large variety of architectures.

Three concrete techniques were presented to make such attacks more expensive for the adversary, tracing the especially vulnerable anonymous reply blocks. Introducing *multicast steps* in the routing increases the number of nodes that have to be compelled to trace, but also makes normal routing more expensive. *Compulsion traps* allow a user to get advance warning of a reply block being traced, and might allow them to eliminate evidence or make tracing more difficult by deleting key material necessary for further tracing [10]. Finally *plausibly deniable routing* provides some anonymity even though all nodes on the path of the reply block have been compelled.

Conceptually we have shown that plausible deniability properties can, when they describe uncertainty about identity, be quantified using the established anonymity metrics. As an example we have calculated the remaining anonymity after a compulsion attack against a plausibly deniable routing scheme. Furthermore we have challenged the optimality of choosing random routes through the network, since it maximizes the effectiveness of traffic analysis, and opted for routing amongst a set of 'friends'. All the techniques we have presented, can be combined to provide a strengthened mix network against compulsion.

The anonymity provided by mix systems is crucially dependent upon trusting third parties. Therefore quantifying the effects of an adversary with compulsion powers against honest nodes is necessary. Other security protocols, or whole systems, that rely on trusting third parties might benefit from such an analysis: how much security can be retained even when trusted players are all forced to turn bad?

Acknowledgements The authors would like to express their gratitude to the anonymous reviewers for their valuable comments and corrections.

References

1. Alessandro Acquisti. Receipt-free homomorphic elections and write-in ballots. Technical Report 105, International Association for Cryptologic Research, May 2 2004.
2. Ross Anderson. Two remarks on public-key cryptology. Available at `http://www.cl.cam.ac.uk/ftp/users/rja14/forwardsecure.pdf`, 1997. Invited Lecture, ACM-CCS '97.
3. Ross Anderson, Roger Needham, and Adi Shamir. The steganographic file system. In David Aucsmith, editor, *Information Hiding (IH'98)*, volume 1525 of *LNCS*, pages 73–82, Portland, Oregon, USA, 15-17 April 1998. Springer-Verlag.
4. Rainer Bohme, George Danezis, Claudia Diaz, Stefan Kopsell, and Andreas Pfitzmann. Mix cascades vs. peer-to-peer: Is one concept superior? In *Privacy Enhancing Technologies (PET 2004)*, Toronto, Canada, May 2004.
5. Zach Brown. Cebolla – pragmatic IP anonymity. In *Ottowa Linux Symposium*, June 2002.
6. Ran Canetti, Shai Halevi, and Jonathan Katz. A forward-secure public-key encryption scheme. In Eli Biham, editor, *EUROCRYPT*, volume 2656 of *Lecture Notes in Computer Science*, pages 255–271. Springer, 2003.

7. David Chaum. Untraceable electronic mail, return addresses, and digital pseudonyms. *Communications of the ACM*, 24(2):84–88, February 1981.
8. David Chaum. Secret-ballot receipts: True voter-verifiable elections. *RSA Laboratories Cryptobytes*, 7(2):14–27, Fall 2004.
9. Richard Clayton and George Danezis. Chaffinch: Confidentiality in the face of legal threats. In Fabien A. P. Petitcolas, editor, *Information Hiding workshop (IH 2002)*, volume 2578 of *LNCS*, pages 70–86, Noordwijkerhout, The Netherlands, 7-9 October 2002. Springer-Verlag.
10. George Danezis. Forward secure mixes. In Jonsson Fisher-Hubner, editor, *Nordic workshop on Secure IT Systems (Norsec 2002)*, pages 195–207, Karlstad, Sweden, November 2002.
11. George Danezis. Designing and attacking anonymous communication systems. Technical Report UCAM-CL-TR-594, University of Cambridge, Computer Laboratory, 2004.
12. George Danezis, Roger Dingledine, and Nick Mathewson. Mixminion: Design of a Type III Anonymous Remailer Protocol. In *IEEE Symposium on Security and Privacy*, Berkeley, CA, 11-14 May 2003.
13. George Danezis and Ben Laurie. Minx: A simple and efficient anonymous packet format. In *Workshop on Privacy in the Electronic Society (WPES 2004)*. ACM, October 2004.
14. Roger Dingledine. Personal communication, 2003.
15. Roger Dingledine, Nick Mathewson, and Paul Syverson. Tor: The second-generation onion router. In *Proceedings of the 13th USENIX Security Symposium*, August 2004.
16. Michael J. Freedman and Robert Morris. Tarzan: A peer-to-peer anonymizing network layer. In Vijayalakshmi Atluri, editor, *ACM Conference on Computer and Communications Security (CCS 2002)*, pages 193–206, Washington, DC, November 2002. ACM.
17. Johan Helsingius. Johan helsingius closes his internet remailer. `http://www.penet.fi/press-english.html`, August 1996.
18. Johan Helsingius. Johan helsingius gets injunction in scientology case privacy protection of anonymous messages still unclear. `http://www.penet.fi/injunc.html`, September 1996.
19. Johan Helsingius. Temporary injunction in the anonymous remailer case. `http://www.penet.fi/injuncl.html`, September 1996.
20. Alfred J. Menezes, Paul C. Van Oorschot, and Scott A. Vanstone. *Handbook of Applied Cryptography*. CRC Press, 1996. ISBN: 0-8493-8523-7.
21. Michael Reiter and Aviel Rubin. Crowds: Anonymity for web transactions. *ACM Transactions on Information and System Security (TISSEC)*, 1(1):66–92, 1998.
22. Marc Rennhard and Bernhard Plattner. Introducing MorphMix: Peer-to-Peer based Anonymous Internet Usage with Collusion Detection. In *Workshop on Privacy in the Electronic Society (WPES 2002)*, Washington, DC, USA, November 2002.
23. Michael Roe. *Cryptography and Evidence*. PhD thesis, University of Cambridge, Computer Laboratory, 1997.
24. Andrei Serjantov and George Danezis. Towards an information theoretic metric for anonymity. In Roger Dingledine and Paul Syverson, editors, *Privacy Enhancing Technologies workshop (PET 2002)*, volume 2482 of *LNCS*, pages 41–53, San Francisco, CA, USA, 14-15 April 2002. Springer-Verlag.

25. Vitaly Shmatikov. Probabilistic analysis of anonymity. In *Computer Security Foundations workshop (CSFW-15 2002)*, pages 119–128, Cape Breton, Nova Scotia, Canada, 24-26 June 2002. IEEE Computer Society.
26. Matthew Wright, Micah Adler, Brian Neil Levine, and Clay Shields. An analysis of the degradation of anonymous protocols. In *Network and Distributed Security Symposium (NDSS '02)*, San Diego, California, 6-8 February 2002.