This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE INFOCOM 2010 proceedings
This paper was presented as part of the main Technical Program at IEEE INFOCOM 2010.

A Node-failure-resilient Anonymous Communication Protocol through Commutative Path Hopping

Fengjun Li [1], Bo Luo [2], Peng Liu [1], Chao-Hsien Chu [1]
[1] College of IST, The Pennsylvania State University
[2] Department of EECS, The University of Kansas

***Abstract*—With rising concerns on user privacy over the Internet, anonymous communication systems that hide the identity of a participant from its partner or third parties are highly desired. Existing approaches either rely on a relative small set of pre-selected relay servers to redirect the messages, or use structured peer-to-peer systems to multicast messages among a set of relay groups. The pre-selection approaches provide good anonymity, but suffer from node failures and scalability problem. The peer-to-peer approaches are subject to node churns and high maintenance overhead, which are the intrinsic problems of P2P systems.**

In this paper, we present CAT, a node-failure-resilient anonymous communication protocol. In this protocol, relay servers are randomly assigned to relay groups. The initiator of a connection selects a set of relay groups instead of relay servers to set up anonymous paths. A valid path consists of relay servers, one from each selected relay group. The initiator explores valid anonymous paths via a probing process. Since the relative positions of relay servers in the path are commutative, there exist multiple anonymous yet commutative paths, which form an anonymous tunnel. When a connection encounters a node failure, it quickly switches to a nearest backup path in the tunnel through "path hopping", without tampering the initiator or renegotiating the keys. Hence, the protocol is resilient to node failures. We also show that the protocol provides good anonymity even when facing types of active and passive attacks. Finally, the operating cost of CAT is analyzed and shown to be similar to other node-based anonymous communication protocols.

I. Introduction

The goal of anonymous communication is to hide the identity of a communication participant (the initiator/recipient or their association) from being known by its partner and other third parties. In today's Internet, where privacy is a critical issue, anonymous communication is becoming desirable in many applications, from anonymous e-mail, web browsing to e-voting and online chatting. However, exiting Internet protocols are mainly designed to support efficient and robust packet transmission. They often fall short of addressing the anonymity requirement in communications. Security mechanisms such as link encryption protect messages enroute from being eavesdropped, however, the messages can still be tracked using various sophisticated traffic analysis techniques.

Several anonymous communication systems have been proposed to address the anonymity requirements. The first generation of such approaches focus on defending against timing attacks [5], [4]. Centralized proxies running Chaum's MIXes are introduced as relay nodes to reorder, delay and pad packets against timing analysis [6]. More recent anonymous routing proposals, such as Tor [20], [9], Crowds [18], Tarzan [11], and MorphMix [19], allow applications to set up an anonymous path by selecting a sequence of proxies from a fixed core set as relay nodes. Session keys are negotiated to wrap the payload and the next-hop route in nested layers so that each relay node only knows its predecessor and successor. Therefore, not only the data stream but also the connection is anonymized. However, these approaches are vulnerable to relay node failures. Once the initiator chooses a specific set of relay nodes to construct the anonymous path, the path is immutable. Hence, the failure of any relay node in a path leads to path failure. Moreover, these systems are sluggish in responding to path failures. After detecting a path failure from end-to-end timeouts, the system needs to reset its active relay set and renegotiate the session keys to construct a new anonymous path, which cannot "inherit" or reuse anything from the failing path to save time. Such path-recovery strategy also makes these approaches vulnerable to passive attacks, for example, the predecessor attack proposed by Wright et. al. [22]. Furthermore, the anonymous approaches atop a static set of relay proxies suffer from scalability problem. As mentioned in [15], the famous anonymous approach Tor [9] is reaching its capacity limit in supporting an increasing population. To address the scalability problem, peer-to-peer anonymous approaches, such as Hordes [12], AP3 [14], Cashemere [23], and Salsa [15], etc, have been proposed. These approaches benefit from the structured P2P network for scalability, however, they still suffer from the intrinsic problems of peer-to-peer systems, such as node churns and high maintenance overhead.

In this paper, we present a node-failure-resilient anonymous communication protocol based on commutative path hopping. In this protocol, relay servers are randomly assigned to relay groups. Nodes of the same relay group share a pair of public and private commutative group keys. The initiator of a connection selects a set of relay groups instead of relay nodes to construct the anonymous paths. A valid path consists of relay nodes, one from each selected relay group. The initiator explores valid anonymous paths via a probing process. Each probe contains information about the selected relay groups and corresponding session keys, where session keys are encrypted in a way that only the first member of each selected relay group can recover the key for the group. A successful probe explores multiple anonymous paths that form a *C*ommutative-path-enabled *A*nonymous *T*unnel. Hence, we name our protocol *CAT*.

978-1-4244-5837-0/10/$26.00 ©2010 IEEE

This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE INFOCOM 2010 proceedings
This paper was presented as part of the main Technical Program at IEEE INFOCOM 2010.

By "commutative", we mean that the anonymous tunnel has the ability to conduct – when needed – per-packet path hopping without "forgetting" the connection context. Path-hopping is dramatically different from re-establishing a new anonymous path since it does not require any new session key to be renegotiated. Due to the unique path-hopping capability, once a relay node detects a node failure in an active anonymous path, it quickly traces back to the predecessor of the failing node, and then automatically switches to a nearest "backup" path. Since the group keys and the session keys are all commutative, the wrapped payload can still be recovered, regardless of the order of the relay nodes in the path. Hence, CAT provides good resilience to node failures via initiator-free path recovery, and good resilience to passive attacks due to less frequent path reconstructions.

The rest of the paper is organized as follows. After an overview of the related work in Section II, we present the system architecture and discuss protocol details in Section III. Next, we analyze the proposed protocol on anonymity and resilience against types of attacks in Section IV and Section V, respectively. The cost and performance of CAT are evaluated in Section VI. Finally, we conclude our work in Section VII.

II. Background and Preliminaries

A. Anonymous communication systems

Research on anonymous communications can be traced back to Chaum's *MIX* [6] in 1981, which was first designed to provide sender-recipient *unlinkability* for electronic mails. In particular, MIX servers pad messages into fixed-size packets, and reorder outgoing packets to defend against timing attacks. Later, [20], [4] have applied the concept of Chaum's MIX to form a MIX cascade or a MIX network in anonymous Web browsing. To avoid mix servers revealing confidential information, onion routing with layered encryption was introduced in [20] and implemented in Tor [9]. In onion routing, the initiator selects nodes from a pool of mix-based relays to form an anonymous path, and encrypts the payload and the route to the next-hop with the public key of each relay in the path, recursively. Most of existing anonymous approaches [9], [11], [19], [23] adopt the well-known Onion Routing paradigm to wrap the payload and the sequence of relays in layers to hide the identifiable information from intermediate relays and third parties.

However, these approaches are vulnerable to node failures, since the failure of any relay node in a path causes path failure. Frequent node failures lead to frequent system reset, which results in further scalability problems and vulnerability to passive attacks. For instance, Wright et. al. presented a special logging attack, called the *predecessor attack* [22], based on a simple observation: if a compromised node intercepts a message, the predecessor node will have a higher probability to be the original sender than other nodes. Therefore, the attacker counts the number of rounds in which a set of messages are sent through a compromised node. If the sender continuously communicates with a particular recipient, it will be identified as the predecessor node with a higher probability. [22] claimes many anonymous protocols/systems [20], [18], [11], [12] are vulnerable to this attack.

Recently, peer-to-peer anonymous routing systems have been proposed to enlarge the pool of relays, such as Tarzan [11], Hordes [12], AP3 [14], Cashmere [23], and Salsa [15]. These approaches benefit from the structured P2P framework for scalability. However, they also suffer from the intrinsic problems of peer-to-peer systems, such as node churns and high maintenance overhead. Cashmere [23], built on a structured overlay, addresses the problem of node churns. It assigns relay nodes to relay groups according to the prefix of their node identifiers. When a relay node fails, its predecessor selects another node in the same relay group to continue packet forwarding. Thus, a single relay node failure is masked by overlay routing. However, Cashmere floods the payload in relay groups, which introduces high maintenance overhead. Moreover, P2P-based approaches rely on peer-to-peer lookups to locate relay servers. But new research shows that attacks against the lookup mechanism would cause severe information leaks [15].

B. Commutative encryption

Commutative cipher is a special class of encryption algorithm that has the property of being *commutative* [21]. An encryption algorithm $E(.) : \mathcal{M} \times \mathcal{K} \rightarrow \mathcal{M}$ is commutative if it satisfies the following properties: for any $m, m' \in \mathcal{M}, m \neq m'$, any $K_1, ..., K_n \in \mathcal{K}$, and any permutations of i, j:

1) $E_{K_{i_1}}(...(E_{K_{i_n}}(m))) = E_{K_{j_1}}(...(E_{K_{j_n}}(m)))$,
2) $Pr[E_{K_1}(...(E_{K_n}(m))) = E_{K_1}(...(E_{K_n}(m')))] < \varepsilon$,

where $\mathcal{M}$ is the message space and $\mathcal{K}$ is the key space.

Two commonly used commutative ciphers are the *Pohlig-Hellman algorithm* based on Elliptic Curve Cryptography (ECC), and the *Shamir-Omura algorithm* based on RSA cryptography. Commutative cipher has been widely used in the literature for various purposes, e.g. protecting secrecy and privacy in e-commerce [2], [3], [7], data mining, information sharing [1], [8], and network routing [13]. In this work, we adopt commutative encryption to wrap the payload in layers.

III. Architecture and Design

Anonymous communication systems allow client applications to set up a connection through a set of relay nodes (a.k.a. relays) to protect the identity of the initiator (i.e. *sender anonymity*), the identity of the target recipient (i.e. *recipient anonymity*), as well as their association (i.e. *sender-recipient unlinkability*) from being known by irrelevant parties. Our goal is to design an anonymous protocol that provides good sender anonymity and sender-recipient unlinkability, especially in the presence of frequent node failures. In this section, we first introduce the system architecture, and then elaborate the CAT protocol in details.

A. System architecture

At macro level, our design allows a relay to determine its next-hop relays following certain construction requirement to achieve better flexibility and robustness in path construction.

This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE INFOCOM 2010 proceedings
This paper was presented as part of the main Technical Program at IEEE INFOCOM 2010.

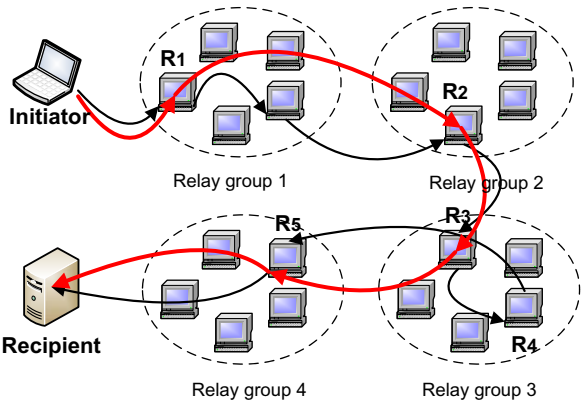

Fig. 1. An example of anonymous path construction in CAT.

More specifically, the initiator of a connection selects a set of relay groups, instead of a sequence of relay nodes, to construct an anonymous path. The payloads of this connection are wrapped with communicative keys so that they can be recovered in any arbitrary order with the correct keys. A simple illustration of such anonymous path construction in CAT is shown in Figure 1. In this example, the initiator selects four relay groups without specifying any relay node. Through a path probing, the first relay node of each group are selected to set up an anonymous path (denoted by the red line).

In this work, we consider an anonymous communication system consisting of n relays, each employing a MIX anonymizer. We assume the relays are randomly but evenly assigned to m relay groups. Ideally, each relay group has $k = n/m$ relays. Relays in the same relay group share a pair of public and private keys, e.g. (pk_i, sk_i) for group$_i$, which are commutative. Similar to other anonymous communication approaches built atop the public-key infrastructure (PKI) [23], our system assumes a proxy server, called *Central Authority (CA)*, for group key generation and distribution. Each relay learns the pair of group keys of its own group and the public group keys of other groups from the CA though some out-of-band mechanisms.

The client (i.e. the initiator) connects to the system through a special user-level process, called *anonymous proxy* (AP). AP is responsible for preprocessing the messages of client applications before sending them to the anonymous network. Such preprocessing includes proper encryption and padding according to a user-selected parameter ℓ, which denotes the number of relay groups used to construct the anonymous path.

B. The CAT protocol

At micro level, the protocol is performed in three steps: (1) it explores multiple anonymous paths atop a set of relays through a path probing process and constructs an anonymous tunnel; (2) it negotiates session keys that are reusable in the anonymous tunnel to encrypt the payload; and (3) when a node failure occurs in the active path, it transparently switches to the nearest backup path through path hopping.

1) Path probing: to initiate a connection, the AP first randomly selects ℓ relay groups, $\{G_1, ..., G_\ell\}$, out of m candidate relay groups. m **flags** are created to denote the status of m relay groups, 1 for each chosen relay group, and 0 for the others. Next, the AP generates the probe with a unique stream identifier pid, and assigns a symmetric session key k_{probe} and ℓ commutative session keys $\{SK_1, ..., SK_\ell\}$ to the probe. The main body of the probe includes the encrypted probe session key and m indicators:

$$\mathbf{Probe} = pid| < ... < k_{probe} >_{pk_1} ... >_{pk_\ell} |\mathbf{I_1}|...|\mathbf{I_m}$$

where $\mathbf{I_i} =<< SK_i >_{k_{probe}} |\mathbf{flag} >_{pk_i}$ is the indicator for the ith relay group. For a relay group that is not chosen for the probing, the corresponding session key field is padded with 0s.

Once the probe is prepared, the AP randomly selects two relays from its candidate relay list to forward the probe. Each selected relay checks the corresponding indicator in the probe to decide its status in the probing process. For example, when a relay R_o in group G_o receives the probe, it extracts the oth flag by decrypting indicator I_o with the private group key of G_o. The status of R_o in the probing process is determined by the recovered flag. A flag set to 1 indicates that group G_o has been chosen for constructing the anonymous path, and relay R_o is the first relay in G_o that receives the probe. In response, R_o needs to (1) set the flag to 0, (2) cache $< SK_o >_{k_{probe}}$ and pad the field with all 0s, and (3) apply its private group key on $< ... < k_{probe} >_{pk_1} ... >_{pk_\ell}$ to unwrap one layer of the encrypted symmetric probe key. If the flag recovered by R_o is 0, it indicates either G_o has not been selected by the AP or it has already been processed by another relay of G_o. In either case, R_o marks itself as "unused" for this probing. After the operations, the probed relay selects two other relays for the next hop, and continues the probing process.

To avoid long probing process, we set a threshold T_f for the number of failed attempts. Each time a probe finds an unused relay, it decreases T_f by 1. When T_f decreases to 0, the probing process stops. The last relay creates an "abort" message to notify all the relays enroute about the failed probing, and the cached session keys will be cleared. A successful probing process stops whenever k_{probe} is completely recovered. In this case, the last relay creates a "confirmation" message with k_{probe} in plaintext for all the active relays enroute. Each active relay retrieves the encrypted commutative session key from its cache, and decrypts it with k_{probe}. During this process, an "unused" relay enroute will quit the path after passing necessary information (e.g. its successor id) to the predecessor. In the end, when the AP receives the "confirmation" message, an anonymous path is finally constructed. With one probe, the AP typically receives more than one "confirmation" message. The AP picks one anonymous path as the active path. Together, all the commutative paths form the *anonymous tunnel* that supports path hopping.

The **advantages** of such path probing are three-fold: first, the initiator is not required to select relays to pre-define a static anonymous path. Hence, no global knowledge about the relays is necessary. Secondly, after choosing the relay

This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE INFOCOM 2010 proceedings
This paper was presented as part of the main Technical Program at IEEE INFOCOM 2010.

groups, the initiator releases the control over the construction of the anonymous tunnel. This introduces more dynamics and randomness in path construction. Finally, since the payload is encrypted with the commutative keys, it can be unwrapped in any arbitrary order. Hence, the positions of the relays in the final anonymous paths are mutable.

2) Payload encryption: once the connection is constructed, the AP prepares the payload by wrapping it with the pre-negotiated session keys in a layered fashion,

$$\mathbf{Payload_i} =< \mathbf{Payload_{i-1}} >_{SK_i}, 1 \leq i \leq \ell$$

Due to the commutative property of the session keys, the payload can be decrypted with ℓ correct keys in any arbitrary order. When the payload is relayed along the anonymous path, its encryption layers will be peered off one by one, until it reaches the last relay. Due to such layered decryption, the input and output differ at intermediate relays. This effectively defends against certain passive traffic analysis attacks.

3) Path hopping: a successful probing often discovers several anonymous paths concurrently. Ideally, if each relay selects two other relays for the next-hop, the constructed anonymous paths span like a binary tree rooted at the initiator. Multiple paths form a virtual anonymous tunnel. The anonymous paths and the virtual tunnel explored by a probe can be identified by the probe identifier pid. So, relays enroute put pid in its routing table and route the packets accordingly. However, compromised relays may link two packets with a same pid and break the anonymity. So, CAT lets each relay enroute replace the old pid with a random path identifer, and cache the tuple $[pid_{old}, pid_{new}, relay_{next}]$ in its routing table.

A virtual tunnel is always constructed for the communication (e.g. TCP streams) between one initiator-recipient pair. Packets of the same stream may be routed through different anonymous paths in the tunnel, independently. For each packet, one path is selected as the "active path", and the others are maintained as "backup" paths. If the current active path fails, the packet automatically switches to the nearest backup path. We call this type of "switching" *path-hopping*. More specifically, per-packet path-hopping is managed as follows: when a relay detects the failure of its active path, it checks its routing table for other available paths in this tunnel. If no backup path exists, the relay notifies its predecessor until a backup path is found or the path failure is detected by the initiator.

Sometimes, connections initiated by the same initiator (with different recipient) are allowed to reuse a virtual tunnel (before it expires) to share the construction cost and to obtain better network efficiency. However, multiple connections sharing one virtual tunnel may reduce the unlinkability between the initiator and the recipient.

C. Discussions

Before further analysis on CAT's anonymity and security, there are a few issues and design choices that we would like to discuss in details.

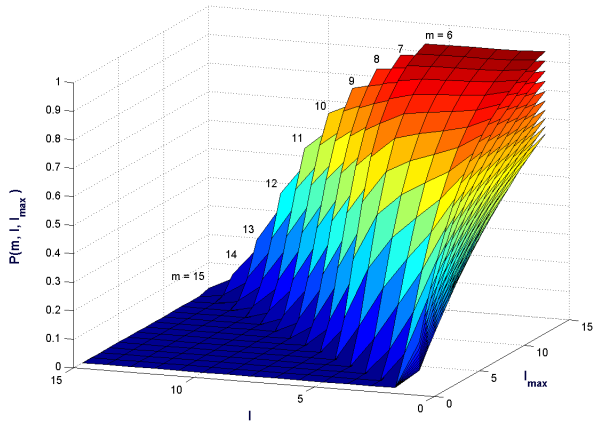

Fig. 2. We calculate $P(m, \ell, \ell_{max})$ in typical parameter settings: $6 \leq m \leq 15$, $1 \leq \ell \leq m$, and $\ell \leq \ell_{max} < 15$.

1) Tuning the parameters for probing: when ℓ relay groups are selected for path probing, ideally, the probe would discover 2^ℓ different paths. However, for protocol efficiency, we should set a limit for the number of hops that a probe can experience. As a result, some probes may fail to find a valid anonymous path within given hops. The probability that a probe successfully finds (at least) one valid path is affected by several parameters. In this section, we explore the dominant parameters, and explain how to tune them to direct an efficient path probing.

According to the protocol, a probe is allowed to make T_f unsuccessful attempts before it fails. Hence, the maximal hops a probe can experience is $\ell_{max} = \ell + T_f$. We calculate $P(m, \ell, \ell_{max})$, the probability that a probe finds at least one valid anonymous path within ℓ_{max} hops, as follows:

$$P(m, \ell, \ell_{max}) = \sum_{i=1}^{\ell_{max}-\ell+1} f(i)[1-(1-P(m-1, \ell-1, \ell_{max}-i))^2],$$

where $f(i) = \binom{\ell_{max}}{i}(\frac{1}{m})^i(1-\frac{1}{m})^{\ell_{max}-i}$ and $P(1,1,1) = 1$.

We calculate the probability in typical parameter settings , and show the results in Figure 2. This help us to understand how the probability is affected by different parameters (e.g. m, ℓ, and ℓ_{max}). From Figure 2, we see that the probability of discovering at least one valid anonymous path increases along with the increase of ℓ_{max} (i.e. maximal number of hops), but it decreases with m (i.e. number of relay groups) and ℓ (i.e. number of chosen relay groups). Considering ℓ should not be too large to maintain the efficiency of the probing process, we suggest its value should be less than 15. When ℓ is set to 15, we notice the probability drops significantly along with the increase of m. Hence, a large m is not recommended. We also notice that the probability is around 0.5 when $\ell_{max} - \ell \geq 6$. This suggests an acceptable range for ℓ should be $[3, \ell_{max}-6]$.

2) Re-issuing relay group membership: relays can infer information about the other relays (e.g. membership) from the responses of neighbor relays in the path probing process. For example, if a relay quits the path during construction and recommends another relay to its predecessor, the predecessor

This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE INFOCOM 2010 proceedings
This paper was presented as part of the main Technical Program at IEEE INFOCOM 2010.

easily infers that they belong to two different relay groups. Moreover, an attacker controlling a large set of compromised servers (e.g. Botnets, or Sibyl attacks [10]) may attempt to obtain group secrets by inserting a large amount of compromised nodes into different relay groups. It is particularly dangerous if the attacker obtains the private group keys of all the relay groups (see discussion in Section V). Therefore, it is critical to find countermeasures against such attacks. A simple yet efficient solution is to renew group membership after a predefined period. Once being renewed, the private groups will be reissued.

3) Resilience to node failures: most anonymous communication systems suffer severely from node failures, especially the ones that use pre-defined static paths. In such systems, the success of message delivery relies on every relay node successfully forwarding the message to the next relay. A relay node may fail due to intermittent network failures or its leaving from the system. Many anonymous communication systems, including Tor [9], recruit their relay nodes from volunteer servers, whose uptime is highly dynamic and beyond control. Meanwhile, peer-to-peer anonymous approaches are facing very frequent node churns, i.e. node joining or leaving the system. Therefore, the performance of an anonymous protocol/system against node failures is an important metrics to evaluate its feasibility.

In anonymous approaches atop pre-defined static paths, one relay node's failure leads to path failure. Moreover, node failures may not be detected at its first occurrence. Once a path failure is detected, the initiator needs to launch a new path construction process to establish a new anonymous path and negotiate new session keys. This means a long break between path corruption and recovery. On the contrary, CAT is resilient to node failures. First, in most cases, it is the predecessor relay instead of the initiator that detects a node failure. Then, path recovery is triggered and accomplished via per-packet path-hopping. The initiator would notice the path failure only after all anonymous paths in the tunnel fail. Therefore, short-lived intermittent failures are masked out, resulting in less path reconstruction.

IV. Anonymity Analysis

The primary goal of an anonymous protocol is to support private and anonymous communications. Anonymity is a probability-based concept with respect to "the state of being unidentifiable" to the attackers [16]. In data communication, there are three types of anonymity : *sender anonymity*, *recipient anonymity*, and *sender-recipient unlinkability* [17]. In this section, we analyze CAT on the sender anonymity and sender-recipient unlinkability.

A. Sender anonymity

Sender anonymity indicates how probably a relay is viewed as the initiator of a connection by an attacker. In CAT, messages along an anonymous path are encrypted, so that the attacker finds no clue of the initiator from message content. Therefore, it is not reasonable for him to associate the messages with any particular relay. However, the attacker's knowledge changes if a compromised relay under his control is selected to construct the anonymous path.

The predecessor of the first compromised relay that intercepts the message is more suspicious to be the initiator, than other relays. This is the predecessor attack which was first proposed by Reiter and Rubin in [18]. In this attack, the attacker identifies the predecessor of the first compromised relay that intercepts the message, and makes a guess that this predecessor is the initiator. Furthermore, Reiter and Rubin proposed a probability model to calculate the chance of such guess being correct. They defined an event I that "the initiator is correctly identified by the attacker", i.e. correct guess; and an event $H_k, k \geq 1$ that "the first compromised relay occurs at the kth position of an anonymous path". To ease the presentation, H_{k+} denotes $H_k \vee H_{k+1} \vee ....$ When a compromised relay intercepts the message, the chance of a correct guess is $P(I|H_{1+})$, which can be further calculated as

$$P(I|H_{1+}) = \frac{P(I \wedge H_{1+})}{P(H_{1+})} = \frac{\sum_{i=1}^{\infty} P(I|H_i)P(H_i)}{P(H_{1+})},$$

considering H_ks are exclusive. Furthermore, we look at the nominator of $P(I|H_{1+})$. $P(I|H_i)$ indicates the probability of correct guess under the condition that the attacker's collaborators first intercept the message at the ith position of the path. Therefore, $P(I|H_1) = 1$, and $P(I|H_k) = 0, k \geq 2$. Hence, we have $P(I|H_{1+}) = \frac{P(H_1)}{P(H_{1+})}$. In Crowds, Reiter and Rubin showed that $P(I|H_{1+}) = 1 - p_f(1 - f)$, where p_f is the forwarding probability [18].

To better understand this result, we view this probability as a posterior probability conditioned on the fact that the message is intercepts by one or multiple compromised relays. Due to the stochastic nature of the random selection, the attacker can only tell that "the predecessor is the initiator of the connection" with certain probability. If there are more than one compromised relays in a path, only the predecessor of the first compromised relay counts. Note that, in Crowds [18], each message is identified by its destination. When multiple compromised relays intercept several messages, they are able to recognize whether the messages are the same. In this way, the attacker is always able to notice his collaborators in the same path, and identify the first one. Hence, the attacker's chance is determined by the probability that the message is first sent to a compromised relay over the overall probability that the message is intercepted by (at least) one compromised relay.

Also note that, the predecessor attack is passive, as pointed out in [22]. compromised relays still carry on relay duties following the protocol to avoid drawing attentions to themselves. Based on this understanding, we believe that CAT provides better sender anonymity than Crowds. In other words, we believe the probability for an attacker to correctly guess the predecessor of the first compromised relay is the initiator is smaller in CAT than in Crowds. We elaborate the reasons as follows.

This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE INFOCOM 2010 proceedings
This paper was presented as part of the main Technical Program at IEEE INFOCOM 2010.

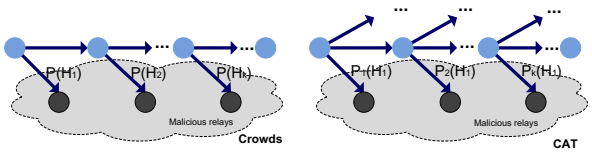

Fig. 3. In Crowds, the first compromised relay receiving the message is identified; however, in CAT, false-positive occurs due to different versions of a same message.

First, each message in CAT appears differently when it traverses multiple relays. More specifically, as described in Section III-B, once created, each message is encrypted with ℓ commutative symmetric keys. When a relay processes the message, it alters the encrypted secret by unwrapping one layer. Hence, a message appears in ℓ different versions. That says, when multiple compromised relays intercept different versions of the same message at different positions, they can hardly link them together from the content, as shown in Figure 3. Furthermore, the incoming packets are permuted at each relay by the embedded Mix. The attacker has little chance to correlate different versions of the message via timing attacks. Therefore, the attacker will more likely treat these different versions as different messages. Thus, the attacker may target at multiple candidate predecessors while there is only one initiator. Since the attacker's chance of correctly guessing the initiator of a connection depends on the first compromised relay that intercepts the message, such indistinguishability of "first compromised relay" will significantly minish the chance for the attacker to correctly guess the initiator.

Obviously, if the attacker can correlate different versions of a message, he can determine the first compromised relay that intercepts the message, and thus targets at its predecessor. For example, if we make an extreme assumption that there is only one message in the system, all the messages can be confirmed as from a same initiator. In this case, the attacker gets his "best guess". Next, we calculate this probability in such an extreme scenario, and compare the result with the one in Crowds.

Now, let us assume the per-hop delay is consistent since it is dominated by Mix delay, and the probing path length is $L \geq \ell$. Hence, when multiple compromised relays intercept such message, the attacker can identify the first interceptor based on timestamps. We now calculate the probability that the first compromised relay occurs at the kth position of the path, $P(H_k)$, in CAT. Since the initiator randomly selects two relays as the first relays, the chance for a compromised relay to occur in the first position is $P(H_1) = 1 - \binom{n-c-1}{2}/\binom{n-1}{2}$. Similarly, we find $P(H_2) = \frac{\binom{n-c-1}{2}}{\binom{n-1}{2}}[1-\frac{\binom{n-c-2}{2}^2}{\binom{n-2}{2}^2}]$, and $P(H_k)$ is:

$$P(H_k) = \frac{\binom{n-c-1}{2}}{\binom{n-1}{2}} \cdot \frac{\binom{n-c-2}{2}^{\sum_{i=1}^{k-2} 2^i}}{\binom{n-2}{2}^{\sum_{i=1}^{k-2} 2^i}} \cdot [1 - \frac{\binom{n-c-2}{2}^{2^{k-1}}}{\binom{n-2}{2}^{2^{k-1}}}],$$

for $3 \leq k \leq L$.

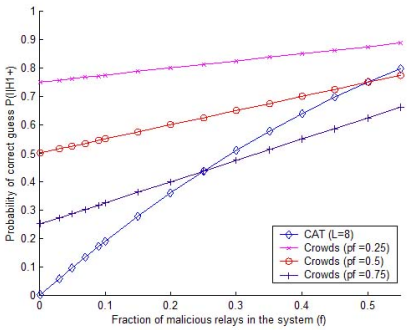

Fig. 4. Compare the probability that the initiator of a message is correctly identified by the attacker in CAT and Crowds.

For a large n, we have a good approximation as:

$$\frac{\binom{n-c-2}{2}}{\binom{n-2}{2}} = \frac{(n-c-2)(n-c-3)}{(n-2)(n-3)} \approx (\frac{n-c}{n})^2 = (1-f)^2.$$

Therefore, H_k is simplified as:

$$H_k = \begin{cases} 1 - \frac{(n-c)^2}{n^2} = 1-(1-f)^2, & k=1; \\ (1-f)^{\sum_{i=1}^{k-1} 2^i}(1-(1-f)^{2^k}), & 2 \leq k \leq L. \end{cases}$$

Finally, we have:

$$P(I|H_{1+}) = \frac{H_1}{H_{1+}} = \frac{1-(1-f)^2}{1-(1-f)^{(2^1+2^2+\ldots+2^L)}}.$$

The result is dominated by the nominator, $1-(1-f)^2$. Even with very small L, the denominator quickly increases to 1. Comparing with the result of Crowds [18], CAT achieves better anonymity performance than Crowds when $p_f < 1-f$. In Figure 4, we compare the probability that the initiator of a message is correctly identified by the attacker using CAT with the one using Crowds. Obviously, when the number of compromised relays is not large in the system (e.g. less than 30%), our protocol provides better sender anonymity than crowds.

Furthermore, Reiter and Rubin defined that a path initiator is *probable innocence* if $P(I|H_{1+}) < 1/2$. To achieve such *probable innocence*, CAT requires the fraction of compromised relays f <30%, while in Crowds, f should be smaller than $1 - \frac{1}{2p_f}$, which is about 33% when p_f is set to 0.75.

Since every relay employs a MIX, the system is less vulnerable to the timing attacks. Also, since the messages have different versions at different positions of the path, an adversary may have no clue to correlate them without launching collusive attacks. In this sense, we argue that with the same number of relays in a path, our approach provides similar degree of sender anonymity as the onion-routing-based approaches [9].

For the attacker, the other $n-c-1$ good relays are equally likely to be the initiator, with a same probability, $\frac{1-P(I|H_{1+})}{n-c-1}$. Following the definition of the entropy-based anonymity measurement, we calculate the *sender anonymity* provided by CAT. As shown in Figure 5, the sender anonymity is a function of the path length L and the fraction of compromised relays f.

This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE INFOCOM 2010 proceedings
This paper was presented as part of the main Technical Program at IEEE INFOCOM 2010.

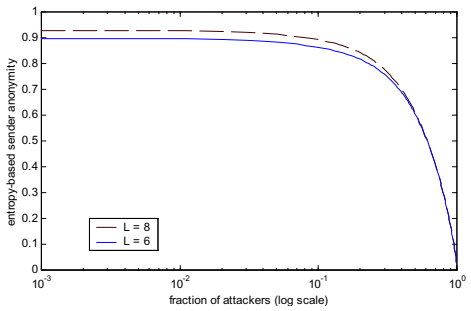

Fig. 5. Measurement of the sender anonymity provided by CAT.

B. Unlinkability

In previous analysis, we find the probability distribution of a good node as being the sender. Next, we need to find the probability of a good node j as being the recipient.

In the CAT approach, if there is no compromised relay enroute, the attacker will have no clue about the identity of the recipient. However, if one compromised relay is chosen as a relay node in the forwarding path, it will introduce its gangs as much as possible. Thus, the attacker may have chance to break or degrade the recipient anonymity.

Let S_L be the relay groups chosen by the AP, and S_k be the relay groups that include one or more compromised relays. In a valid path of length L, if a probe meets the first compromised node at position $r+1$, it is expected that the probe has already passed through r good relay nodes.

In this case, the compromised relay will collude with all the other compromised nodes to try to occupy the remaining positions in the path. If $S_k \supseteq S_L - S_r$, the colluding nodes can control the remaining $L-r$ hops, and reveal the identity of the recipient. If $S'_k = S_k \cap (S_L - S_r)$ is not empty, the colluding node can only control k' consecutive hops. Since the suffix position (of the path) is still chosen from good nodes, the identity of the recipient will not be revealed. Overall, the attacker can assign the probability of a good node j as being the recipient as:

$$p_j = \begin{cases} \frac{m}{(m-k')(N-c)-m}, & S'_k \neq empty \\ \frac{1}{N-c}, & S'_k = empty \end{cases}$$

Assume an attacker assigns a probability p_{ij} for each pair of nodes (n_i, n_j) as being the sender and recipient of a message. The *sender-recipient unlinkability* can be calculated following the similar entropy-based method. Moreover, if the attacker believes n_i is the sender with a probability p_i and n_j is the receiver with a probability p_j, then $p_{ij} = p_i p_j$, and the unlinkability of sender and receiver can be calculated as below,

$$H_u = \frac{-\sum p_{ij} log_2(p_{ij})}{-n^2 \cdot (\frac{1}{n^2}) log_2(\frac{1}{n^2})} = \frac{-\sum p_{ij} log_2(p_{ij})}{2 log_2(N)}.$$

Thus, together with p_i calculated from Section IV-A, we could quantify the unlinkability provided by CAT.

V. Security Analysis

In this section, we discuss common attacks against anonymous systems that have been reported in recent literature, and analyze how well CAT defend against them. These attacks often assume a power attacker that controls a fraction of relays (or nodes). With compromised relays in the system, the attacker may control partial or even the entire anonymous path, and examine the network traffic (locally or globally) to break the anonymity of other relays actively (e.g. intersection attacks) or passively (e.g. predecessor attacks, traffic attacks).

A. Control initiators' circuits

The health of anonymous paths is the basis of anonymous communication. If an attacker controls all the relays on the initiator's path, he would easily associate the initiator with the recipient, and break all types of anonymity. Similar to other dynamic path approaches (e.g. MorphMix [19]), CAT is particularly vulnerable to this attack since the initiator cannot select relays for the anonymous path. On the contrary, if a compromised relay is selected at any hop, it attempts to bias node selection process to put more compromised relays into the route. To mitigates this threat, CAT lets the initiator select a number of relay groups. Since the group membership is randomly assigned and frequently refreshed, and the membership is kept secret among the relays, it is difficult for the attacker to construct a path with all compromised relays in limited number of tries. However, it is particularly dangerous if the attacker puts (at least) one compromised relay into each relay group. In this way, he can obtain all the private group keys, and thus have a good chance to intercept the commutative session key during probing. The attacker's chance can be modeled as a "balls-and-bins" probability problem, similar as discussed in [15]. When the number of relay groups is small (10 to 20), the attacker may possibly control enough number of compromised relays to launch the attack. Therefore, it is critical to find other mechanisms to defend against this attack.

The first and the most common countermeasure is to limit the number of relays that are associated with a unique owner. There are several research against botnets and sybil attacks. Here, we adopt a simple solution that limits the number of joining requests from the same IP. We also cache the membership of a node when it quits the system. Next time, when it rejoins before membership refreshing, it will be assigned to the same group. Another countermeasure is to refresh the relay group membership periodically, so that it will be more difficult for the attacker if he tries to learn the group distribution and compromise relays with a clear target.

B. Traffic analysis

CAT employs a MIX server at relay, to introduce padding, reordering outgoing, and packet delays at each hop to defend against most of the typical traffic analysis. Moreover, some of the special features of the CAT protocol make it even harder for the attacker to distinguish messages from different streams. In CAT, an initiator sends encrypted messages to the recipient over multiple paths in the same anonymous tunnel.

This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE INFOCOM 2010 proceedings
This paper was presented as part of the main Technical Program at IEEE INFOCOM 2010.

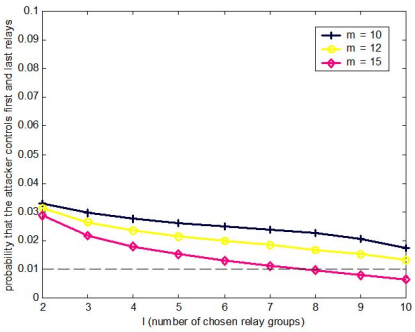

Fig. 6. The probability that the attacker controls the first and last relays.

The path along which an encrypted payload would be routed is randomly decided by all the relays enroute. Even the initiator itself has no knowledge or control. Moreover, payload is unwrapped with one layer at each relay, which automatically alters its appearance.

However, like most other anonymous systems, CAT also suffers from end-to-end timing attacks, if the attacker targets two ends of a stream, and correlates messages from their timing characteristics. The attacker can do so if he controls both the first and the last relays of an anonymous path. In most approaches, such as Tor, Crowds, etc., the relays are randomly chosen. Therefore the chance of the compromised relays being chosen at both ends is $(\frac{c}{n})^2$. Now, we analyze the attacker's chance of success in our system.

In CAT, the chance of a compromised relay being selected as the first relay is $\frac{c}{n}$. All the compromised relays that have been chosen would bias the relay selection process to increase the chance of their gangs. However, their chance still depends on three conditions: (1) the probe has experienced $\ell - 1$ selected groups before it is sent out for the last group probe; (2) the hop limit ℓ_{max} has not been reached; and (3) before selecting a compromised node, the probe is sent to the nodes belonging to either the irrelative groups or the experienced groups. The probability of satisfying the first two conditions is $\sum_{i=1}^{\ell_{max}-L+1} P(m-1, \ell-1, \ell_{max}-i)$. And the probability of satisfying condition (3) is $\frac{c}{n}[(1-\frac{1}{m})(1-\frac{c}{n})]^{i-1}$. Here, the attacker's chance of controlling the first and last relays is

$$\mathtt{P} = \sum_{i=1}^{\ell_{max}-\ell+1} \frac{c^2}{n^2}[(1-\frac{1}{m})(1-\frac{c}{n})]^{i-1} P(m-1, \ell-1, \ell_{max}-i)$$

Although the above formula does not provide a closed-form solution, we can calculate the probability recursively. We assume the attacker controls 10% of the relays in the system (100 out of 1000 relays), and calculate the probability under typical parameter settings (ℓ_{max}=15, and m = 10, 12 and 15). The results are shown in Figure 6. In general, the attacker's chance in CAT is slightly higher (with an average 0.0146 when $m = 15$) comparing with the one in other systems (e.g. 0.01 when $\frac{c}{n} = 0.1$).

C. Predecessor Attack

As introduced in Section II, an attacker can launch a logging attack, called *predecessor attack*, to identify the initiator of a connection as the monitored predecessor with high occurrence. The predecessor attack is very effective, as pointed out in [22], especially when two parties continuously communicate across path reformations. As indicated in [22], the generic attack is effective on an anonymous protocol when it is adopted to support recurring and trackable connections. Many anonymous protocols [20], [18], [11], [12] are subject to this attack.

To launch the predecessor attack, an attacker, who controls a number of relays, logs the predecessor of the first malicious relay that intercepts a message when the path is reformed. Statistically, the initiator will be logged more often than other relays, after a large number of path reformations, the attacker may identify the initiator with higher confidence.

If CAT is adopted to support certain applications in which the initiator needs to remain connected with a recipient for a long period of time, it still suffers from the predecessor attack. However, at path failures, CAT allows "hopping" among a set of backup paths instead of a complete path reconstruction. Since the hopping operations does not involve the initiator, observation of such hopping will not increase the attacker's change of correct guess. As a result, less frequent path reconstruction leads to better resilience against the predecessor attack.

VI. Cost and Performance

In the proposed CAT protocol, messages that are encrypted with commutative session keys are routed among a number of relays from the chosen set of relay groups in any arbitrary sequence. As discussed in Section III and Section IV, it achieves good performance against node failures and attacks such as the predecessor attack. In this section, we analyze the relative costs in operating CAT from three aspects: (1) the burden on nodes; (2) the burden on links; and (3) the burden on encryption operations. We compare the costs with other node-based systems, and show that CAT introduces no significant, if not less, overhead.

Burden on nodes. Three types of nodes exist in CAT, a centralized node *CA*, anonymous proxies, and relay servers. The burden on nodes basically measures the communication costs to maintain a graph of candidate relay nodes. Obviously, the *CA* is the one with the largest cost since it is responsible for actively maintaining status of all other nodes, as well as membership management and key (i.e. group keys) management. The cost at *CA* is $O(n)$, where n is the entire network. APs and relays also keep a list of active nodes, which causes a total cost of $O(n^2)$. However, if APs and relays update their lists from the *CA*, the overall communication cost will be reduced. In summery, CAT introduces a similar, sometime a little smaller, burden on nodes comparing with other node-based anonymous systems, e.g. Crowds [18], Tor [9].

Burden on links and communication latency. Burden on links actually measures the bandwidth cost in message for-

This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE INFOCOM 2010 proceedings
This paper was presented as part of the main Technical Program at IEEE INFOCOM 2010.

warding. In CAT, assume ℓ relay groups are chosen to construct the path. Although the probe is allowed to pass at most ℓ_{max} hops, after a path is setup, its length is determined as ℓ. Furthermore, although multiple paths may be constructed, conceptually only one path is using at a time. Therefore, the bandwidth cost is still $O(\ell)$.

Burden on encryption operations. In node-based anonymous systems, traffic between any two nodes is encrypted with negotiated link keys to protect confidentiality. Also, approaches based on Onion Routing paradigm wrap the payload and route information with the public keys of a set of relays. Other approaches such as [11], [23] adopted symmetric key encryption for payload and amortizes asymmetric crypto operations across sessions.

CAT adopts all of the above encryption operations, as well as commutative symmetric and asymmetric encryptions at each intermediate relay in both probing and message forwarding. Thus, it introduces a high computational cost at the AP and the intermediate relays. To obtain an idea, we assume the AP select ℓ relay groups to construct the paths.

In path probing, the AP takes ℓ symmetric key encryptions to prepare the encrypted session keys, and another ℓ asymmetric key encryptions to wrap the session keys into ℓ layers. Each intermediate relay, if it is the first node from its group receiving the probe, applies one asymmetric decryption to unwrap one layer of the secret, and applies another symmetric decryption to uncover the commutative session key.

In message forwarding, an AP applies ℓ symmetric encryption operations to prepare a data packet, and an intermediate relay only needs to apply one symmetric decryption. Also note that, no additional cost for path hopping since the systematic session keys are cached at relays in backup paths. In summery, total computation cost is $O(\ell)$.

VII. Conclusion

We present CAT, an anonymous communication protocol that features commutative path hopping and commutative payload wrapping. In addition to providing good anonymity, CAT is resilient to relay node failures, as well as active and passive attacks.

In CAT, instead of pre-selecting all the relay nodes on a path, the initiator chooses a set of relay groups to construct a virtual tunnel via probing. Each tunnel consists multiple anonymous paths, and each path has a sequence of relays connected in arbitrary order. Payloads are encrypted by commutative session keys and transported through the tunnel. In the event of path failure caused by node failures, CAT hops to the nearest backup path in the same tunnel, without tampering with the initiator or altering payload encryption. In addition to node-failure resilience, we show that CAT provides similar, if not better, anonymity and resilience against predecessor attack, compared with node-based approaches such as Crowds. We also show that CAT only introduces minimal operation overhead to archive these goals.

Acknowledgement

This work was supported by NSF CNS-0916469, NSF CNS-0905131, and ARO MURI: Computer-aided Human Centric Cyber Situation Awareness.

References

[1] R. Agrawal, A. Evfimievski, and R. Srikant. Information sharing across private databases. In *SIGMOD '03: Proceedings of the 2003 ACM SIGMOD*, pages 86–97, 2003.

[2] F. Bao, R. H. Deng, and P. Feng. An efficient and practical scheme for privacy protection in the e-commerce of digital goods. In *ICISC '00: Proceedings of the Third International Conference on Information Security and Cryptology*, pages 162–170, 2001.

[3] F. Bao, R. H. Deng, P. Feng, Y. Guo, and H. Wu. Secure and private distribution of online video and some related cryptographic issues. In *ACISP '01: Proceedings of the 6th Australasian Conference on Information Security and Privacy*, pages 190–205, 2001.

[4] O. Berthold, H. Federrath, and S. Kpsell. Web mixes: A system for anonymous and unobservable internet access. pages 115–129, 2000.

[5] J. Boyan. The anonymizer: Protecting user privacy on the web. *Computer-Mediated Communication Magazine*, 4(9), September 1997.

[6] D. L. Chaum. Untraceable electronic mail, return addresses, and digital pseudonyms. *Communications of the ACM*, 24(2):84–90, 1981.

[7] S.-C. Cheung, H.-F. Leung, and C. Wang. A commutative encrypted protocol for the privacy protection of watermarks in digital contents. In *HICSS '04: Proceedings of the Proceedings of the 37th Annual Hawaii International Conference on System Sciences (HICSS'04)*, 2004.

[8] C. Clifton, M. Kantarcioglu, J. Vaidya, X. Lin, and M. Zhu. Tools for privacy preserving distributed data mining. *ACM SIGKDD Explorations*, 4(2), 2003.

[9] R. Dingledine, N. Mathewson, and P. Syverson. Tor: The second-generation onion router. In *the 13th USENIX Security Symposium*, 2004.

[10] J. R. Douceur. In *IPTPS '01: Revised Papers from the First International Workshop on Peer-to-Peer Systems*, pages 251–260, London, UK.

[11] M. J. Freedman and R. Morris. Tarzan: A peer-to-peer anonymizing network layer. In *Proceedings of the 9th ACM Conference on Computer and Communications Security (CCS 2002)*, Washington, D.C.

[12] B. N. Levine and C. Shields. Hordes: A multicast based protocol for anonymity. *Journal of Computer Security*, 10:213–240, 2002.

[13] H. Y. S. Lu. Commutative cipher based en-route filtering in wireless sensor networks. In *Vehicular Technology Conference*, volume 2, pages 1223– 1227, Sept. 2004.

[14] A. Mislove, G. Oberoi, A. Post, C. Reis, P. Druschel, and D. S. Wallach. AP3: cooperative, decentralized anonymous communication. In *EW11: Proceedings of the 11th workshop on ACM SIGOPS European workshop*, page 30, 2004.

[15] P. Mittal and N. Borisov. Information leaks in structured peer-to-peer anonymous communication systems. In *CCS '08: Proceedings of the 15th ACM conference on Computer and communications security*, pages 267–278, 2008.

[16] Pfitzmann, Andreas, and M. Köhntopp. Anonymity, unobservability, and pseudonymity - a proposal for terminology. In *Designing Privacy Enhancing Technologies*, pages 1–9, 2001.

[17] A. Pfitzmann and M. Waidner. Networks without user observability. *Comput. Secur.*, 6(2):158–166, 1987.

[18] M. K. Reiter and A. D. Rubin. Crowds: anonymity for Web transactions. *ACM Transactions on Information and System Security*, 1(1):66–92, 1998.

[19] M. Rennhard and B. Plattner. Introducing morphmix: peer-to-peer based anonymous internet usage with collusion detection. In *WPES '02: Proceedings of the 2002 ACM workshop on Privacy in the Electronic Society*, pages 91–102, 2002.

[20] P. F. Syverson, D. M. Goldschlag, and M. G. Reed. Anonymous connections and onion routing. In *SP '97: Proceedings of the 1997 IEEE Symposium on Security and Privacy*, page 44, 1997.

[21] S. A. Weis. PhD thesis.

[22] M. Wright, M. Adler, B. N. Levine, and C. Shields. An analysis of the degradation of anonymous protocols. In *In Proceedings of the Network and Distributed Security Symposium - NDSS 02. IEEE*, 2002.

[23] L. Zhuang, F. Zhou, B. Y. Zhao, and A. Rowstron. Cashmere: Resilient anonymous routing. In *NSDI*, 2005.