A Quantitative Analysis of Anonymous Communications

Yong Guan, Xinwen Fu, Riccardo Bettati, and Wei Zhao

***Abstract*—This paper quantitatively analyzes anonymous communication systems (ACS) with regard to anonymity properties. Various ACS have been designed & implemented. However, there are few formal & quantitative analyzes on how these systems perform. System developers argue the security goals which their systems can achieve. Such results are vague & not persuasive. This paper uses a probabilistic method to investigate the anonymity behavior of ACS.**

In particular, this paper studies the probability that the true identity of a sender can be discovered in an ACS, given that some nodes have been compromised. It is through this analysis that design guidelines can be identified for systems aimed at providing communication anonymity. For example, contrary to what one would intuitively expect, these analytic results show that the probability that the true identity of a sender can be discovered might not always decrease as the length of communication path increases.

***Index Terms*—Anonymous communication, network security, rerouting, sender/receiver anonymity.**

I. Acronyms & Names[1]

ACS	anonymous communication system
Anonymizer	a web proxy to achieve web-browsing anonymity
Anonymous Remailer	anonymous e-mail service
DARPA	USA Defense Advanced Research Projects Agency
DC-net	an approach to achieve anonymity
L1	strategy of selecting paths with fixed length
L2	strategy of selecting paths with variable length of geometric distribution
L3	strategy of selecting paths with variable length of uniform distribution
LPWA	Lucent Personalized Web Assistant
Mix	an approach to achieve anonymity
PipeNet	an anonymous protocol
T1	strategy of selecting paths without cycles
T3	strategy of selecting paths with disjoint cycles
T3	strategy of selecting paths with arbitrary nonreflective cycles

Notation

$\mathbf{N}$	number of nodes in the system
$\mathbf{V}$	$\{v_1, v_2, \ldots, v_N\}$: set of nodes in the system
$\mathbf{M}$	number of compromised nodes, for $M \leq N$
$\mathbf{L}$	path length
$\mathbf{L}'$	guessed $\mathbf{L}$ by the adversary
$\mathbf{K}$	number of compromised nodes on the rerouting path, $0 \leq K \leq \min(M, L)$
$\mathbf{s}$	true sender of the message
$\mathbf{R}$	receiver of the message
$\mathbf{R.P}$	immediate predecessor of the receiver
$\mathbf{NS}$	set of nodes which are definitely not the true sender
$\mathbf{PS}$	set of nodes which are likely to be the true sender
$\mathbf{CP}$	set of compromised nodes on the rerouting path
$\mathbf{CNP}$	set of compromised nodes which are not on the rerouting path
$< t_{C_i}, P_{C_i}, S_{C_i}, C_i >$	information reported from the compromised node, C_i
t_{C_i}	time instant when the message traverses C_i
P_{C_i}	immediate predecessor of C_i
S_{C_i}	immediate successor of C_i
$\mathbf{\Omega}$	all possible event, ω, that the adversary may observe
p_f	forwarding probability
$f_1, f_2, \ldots, f_J$	J completely identified path-fragments
$f(i, j, L)$	$\Pr\{$the true sender s can be identified for instance $\#j$ that there are i compromised nodes on the path with length $L\}$
F	the fact that the adversary collected, including path segments $f_1, f_2, \ldots, f_J$ and the order thereof

Manuscript received July 19, 2002. This work was supported in part by NSF under Contract Number EIA-0081761, DARPA under Contract Number F30602-99-1-0531, and Texas Higher Education Coordinating Board under its Advanced Technology Program. Responsible Editor: N. Ye.

Y. Guan is with the Department of Electrical and Computer Engineering, Iowa State University, Ames, IA 50011 USA (e-mail: guan@ee.iastate.edu).

X. Fu, R. Bettati and W. Zhao are with the Department of Computer Science, Texas A&M University, College Station, TX 77843-3112, USA (e-mail: xinwenfu@cs.tamu.edu; bettati@cs.tamu.edu; Zhao@cs.tamu.edu).

Digital Object Identifier 10.1109/TR.2004.824826

[1]The singular & plural of an acronym are always spelled the same.

II. Introduction

THIS paper quantitatively analyzes ACS with regard to anonymity properties. With the rapid growth and public acceptance of the Internet as a means of communication and information dissemination, concerns about privacy & security on the Internet have grown. Anonymity becomes a basic requirement for many on-line Internet applications, such as E-Voting, E-Banking, E-Commerce, and E-Auctions. Anonymity protects the identity of a participant in a networked application. Many ACS have been developed, which protect the identity of the participants in various forms; sender anonymity protects the identity of the sender, while receiver anonymity does this for the receiver. Mutual anonymity guarantees that both parties of a communication remain anonymous to each other. Finally, some systems provide unlinkability-of-sender-and-receiver. In such systems, no one can infer the communication relation between the sender & receiver, except the sender & receiver themselves.

Among these various forms of anonymity, sender anonymity is most demanded in the current Internet applications. In E-Voting, for example, a cast vote should not be traceable back to the voter. Similarly, payments using E-Cash should be nontraceable. Finally, users may generally not want to disclose their identities when visiting web sites. Thus, this paper focuses primarily on sender anonymity.

Sender anonymity is most commonly achieved by transmitting the message to its destination through one or more intermediate nodes, to hide the true identity of the sender. The message is effectively rerouted along what is called the rerouting path. This paper studies rerouting-based anonymous communication systems in terms of their ability to provide sender anonymity. The selection of rerouting paths is critical for this kind of ACS. The 2 key issues in path selection are:

1) how to choose the path length, and
2) how to choose the path topology.

This paper studies how different 'path selection strategies' affect the ability to provide sender anonymity. For a given anonymous communication system, this ability is measured as the probability that the true identity of a sender can be discovered.

This investigation assumes a passive adversary model. An adversary can compromise one or more nodes in the system. An adversary agent at such a compromised node can gather information about messages that traverse the node. If the compromised node is involved in the message rerouting, it might be able to discover and report the immediate predecessor and successor node for each message traversing the compromised node. The adversary is assumed to collect all the information from the compromised nodes, and to attempt to derive the identity of the sender of a message.

The following sections describe several insightful results based on a quantitative analysis of ACS.

- Contrary to intuition, the probability that the true identity of a sender can be discovered might not always decrease as the path length increases.
- The complexity of the path topology does not have an important impact on this probability. While paths with complicated topology perform better than simple ones, the difference is relatively small.
- As anticipated, the probability that the true identity of a sender can be discovered increases when the number of compromised nodes in the system increases. In particular, the deterioration of the system behavior goes sublinearly as the number of compromised nodes increases.

This study showed that several well-known ACS are not using the best path selection strategies. Therefore these results are very helpful for the current & future development of ACS.

Section 2 gives an overview of the ACS. Section 3 presents the system model, and discusses the key issues in path selection for ACS. Section 4 describes the threat model, and discusses how the adversary determines the identity of the sender of a message under this model. Section 5 reports the analytic & numeric results. Section 6 presents several remarks.

III. Overview of Anonymous Communication Systems

This section surveys the past work related to anonymity, including DC-Net [5], [39], Mixes [4], [18], [20], Anonymizer [1], Anonymous Remailer [2], LPWA [10], Onion Routing [13], [31], [33], [34], Crowds [26], Hordes [28], Freedom [12], and PipeNet [8].

Many ACS have been designed & implemented to provide various types of anonymity, such as sender anonymity, receiver anonymity, mutual anonymity, unlinkability of sender & receiver, or combinations thereof. As mentioned in the Introduction, sender anonymity is typically most demanded in current Internet applications.

Systems providing sender anonymity can be categorized into two classes:

1) rerouting-based systems, and
2) nonrerouting-based systems.

To the best of our knowledge, DC-Net [5] is the only nonrerouting-based ACS. In DC-Net, each participant shares secret coin flips with other pairs, and announces the parity of the observed flips to all other participants and to the receiver. The total parity should be an even number, because each flip is announced twice. By incorrectly stating the parity that the sender has seen, this causes the total parity to be an odd number. Thus the sender can send a message to the receiver. The receiver gets the message if it finds that the total parity is odd. No-one, except the sender, knows who sends it. The advantage of DC-Net over rerouting-based systems, is that it does not introduce extra overhead in terms of longer rerouting delays & an extra amount of rerouting traffic. It relies, however, on an underlying broadcast medium, which comes at great expense as the number of participants increases. Due to its lack of scalability in practice (e.g., the number of participants), none of the current on-line applications use this method. The remainder of this paper therefore focuses on rerouting-based systems.

Most widely-used ACS use rerouting of a message through a number of intermediate nodes. The sender sends the message to such an intermediate node first. This node then forward the message either to the receiver, or to another intermediate node, which then forward the message again. After the message traverses the first intermediate node, the sender cannot be identified through the information kept in the header of the message. Even though rerouting introduces extra delay and typically increases the amount of traffic due to longer routes, this approach

is scalable & practical when such overheads are within tolerable limits.

The remainder of this section briefly overviews several such communication systems, categorizing them according to their path selection strategies.

Anonymizer [1] provides fast, anonymous, interactive communication services. Anonymizer in this approach is essentially a web proxy that filters out the identifying headers and source addresses from web client requests. Instead of a user's true identity, a web server can only learn the identity of the Anonymizer-Server. In this approach, all rerouting paths have a single intermediate node, which is the Anonymizer-Server.

Anonymous Remailer [2] is mainly used for e-mail anonymity. It uses rerouting of an e-mail through a sequence of mail remailers, and then to the recipient such that the true origin of the e-mail can be hidden.

Onion-routing [13], [31], [34] provides anonymous Internet connection services. It builds a rerouting path within a network of onion-routers, which in turn are similar to real-time Chaum Mixes [4]. The basic idea of the mix approach is that each message is sent over a series of independent stations (mixes). A mix is a store-and-forward device which accepts a number of fixed-length messages from different sources, discards repeats, performs a cryptographic transformation on the messages, and then outputs the message to the next destination in an order not predictable from the order of inputs.

Onion Routing I [31], [33], [34] uses a network of 5 Onion Routing nodes operating at the Naval Research Laboratory. It forces a fixed length (5 hops) for all routes.

Onion Routing II [33] can support a network of up to 50 core Onion Routers. For each rerouting path through an onion routing network, each hop is chosen at random. The rerouting path may contain cycles. Its path selection approach is borrowed from Crowds [26]. The anticipated route length is completely determined by the weight of a Bernoulli trial.

Crowds [20] aims at protecting the users' web-browsing anonymity. Like Onion Routing, the Crowds protocol uses a series of cooperating proxies (called jondos) to maintain anonymity within the group. Unlike Onion Routing, the sender does not determine the entire path. Instead, the path is chosen randomly on a hop-by-hop basis. Cycles are allowed on the path. Once a path is chosen, it is used for all the anonymous communication from the sender to the receiver within a 24-hour period. At some specific time instant, new members can join the crowd and new paths can be formed.

Freedom Network [12] also aims at providing anonymity for web browsing. Freedom is similar to Onion Routing. It consists of a set of proxies which run on top of the existing Internet infrastructure. To communicate with a web server, the user first selects a sequence of proxies to form a rerouting path, and then uses this path to forward the requests to its destination. The Freedom Route Creation Protocol allows the sender to randomly-choose the path, but the path length is fixed at 3 intermediate nodes [35]. The Freedom client user interface does not allow the user to specify a path containing cycles.

Hordes [28] uses multiple jondos similar to those used in the Crowds protocol to anonymously route a packet toward the receiver. It uses multicast services, however, to anonymously-route the reply back to the sender instead of using the reverse path of the request. Similar to Crowds, Hordes also allows cycles on the forwarding path.

Lucent Personalized Web Assistant [10] uses a single proxy server which accepts connections from the sender of an anonymous connection, and forward them on to the host that the sender wishes to contact anonymously. Obviously, this system uses the rerouting path with only 1 intermediate node, similar to Anonymizer.

PipeNet [8] is a simple anonymous protocol, based on the idea of virtual link encryption. Before the sender starts to send the data, it establishes a rerouting path. PipeNet always generates a rerouting path with 3 or 4 intermediate nodes.

IV. System Model and Key Issues in Path Selection

The system model used in the following discussion is an abstraction of the systems mentioned in Section 2. It therefore lends itself well to discussing the key issues in rerouting-based ACS.

A. System Model

A rerouting-based ACS consists of a set of N nodes $V = \{v_i : 0 \leq i < N\}$, which collaborate with each other to achieve anonymity. Following general practice, assume that the receiver R is always compromised, and therefore is not included as part of the N nodes. For this paper, the network is modeled at the transport layer, and every host can communicate with every other host. The network therefore can be modeled as a clique. An edge in this graph represents a direct path (with no intermediate nodes) from a source host to a destination host (possibly through some routers in the network). To hide the true identity of the sender, the message is transmitted from source to destination through one or more intermediate nodes. The path traversed by the message is a rerouting-path, and is described as follows

$$\langle s, I_1, \ldots, I_L, R \rangle, \tag{1}$$

$s \in V$ is the sender, $I_k \in V (1 \leq k \leq L)$ is the intermediate node $\#k$ on the path, R is the receiver, and $R \notin V$.

Fig. 1 shows a system of 16 nodes; Node 0 is the sender of a message. The message is transmitted along the rerouting path $< 0, 5, 2, 7, 11, 8, R >$ determined by the anonymous communication system, and finally arrives at Node R. In this example, the message has traversed 5 intermediate nodes. Path-length is defined as the number of intermediate nodes on the path; therefore the path length is 5.

B. Path Selection

Either before or during the transmission of a message, the rerouting-based ACS must construct a rerouting path from the source to the destination. Fig. 2 shows a framework for how this can be done. (The steps in Fig. 2 are often made only implicitly in real systems; e.g., the protocol might impose a path length of a given fixed size, and a limited number of rerouting nodes might make a selection of a rerouting sequence irrelevant.)

From Fig. 2, it is clear that the key steps in path selection are:

1) choose the length of the rerouting path (path length), and
2) choose the sequence of intermediate nodes on the path.

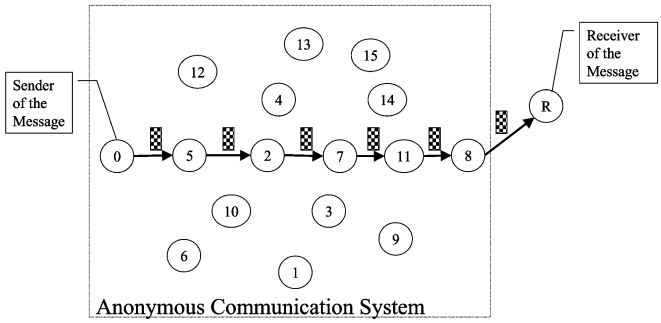

Fig. 1. System Model.

INPUT: s, R: source s and destination R of a message

1. Select path length L;
2. Choose a sequence of intermediate nodes $I_1, I_2, \ldots, I_L$;
3. Return the path $< s, I_1, I_2, \ldots, I_L, R >$.

Fig. 2. Rerouting Path Selection Algorithm.

1) Choosing Path Length: Two kinds of strategies can be used: fixed-length and variable-length. For variable-length, the path length is a r.v. conforming to a specific probability distribution. Onion-Routing I & Freedom use fixed-length strategies, whereas Crowds & Onion-Routing II use variable-length strategies. The system developer must decide the type of path length selection (fixed or variable) and its parameters. To study the effect of path length strategy, consider the 3 typical path-length strategies:

- Strategy L1: Fixed Length. Whenever a sender wants to send a message to some receiver, the ACS chooses a fixed number of intermediate nodes to form the rerouting path to transfer the message.
- Strategy L2: Variable Length with geometric path-length distribution. The path length is randomly chosen as a nonnegative number conforming to the distribution, for $x \geq 0$,

$$\Pr\{L = x\} = (1 - p_f) \cdot p_f^x; \tag{2}$$

  $p_f \equiv$ the forwarding probability which controls the s-expected length of the rerouting path.
- Strategy L3: Variable Length with uniform path-length distribution. The path length is randomly chosen as a nonnegative number between 2 values, a & b, following a uniform distribution

$$\Pr\{L = x\} = \frac{1}{b-a}, \quad a \leq x < b.$$

2) Choosing Path Topology: Once the path length is defined, the rerouting path is chosen by randomly selecting intermediate nodes. Depending on whether a node can be chosen on a rerouting path more than once, classify paths either as:

- simple paths: no cycle is allowed, or
- complicated paths: cycles are allowed.

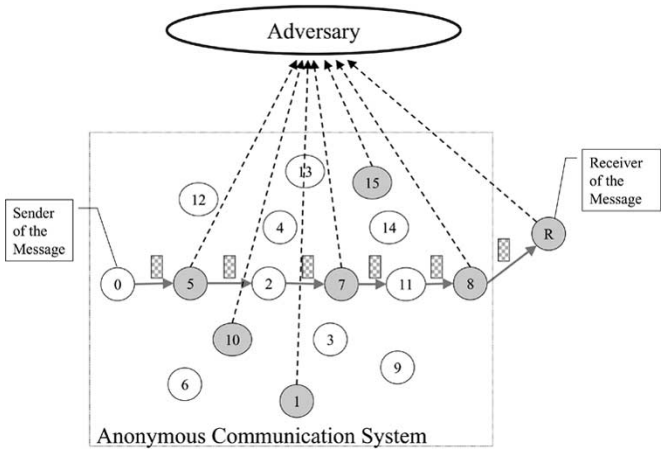

Fig. 3. Threat Model.

To study the effect of path topology, consider the following 3 strategies:

- Strategy T1: the rerouting path does not allow cycles. No intermediate node can show up in more than one different place on the rerouting path.
- Strategy T2: the rerouting path allows disjoint cycles, which are cycles that do not share common nodes. Cycles must be nonreflective. Only the intermediate node, as the starting & ending point of a cycle, can show up exactly 2 times.
- Strategy T3: the rerouting path allows arbitrary nonreflective cycles. This is a path which ends at the node where it begins, and the immediate successor of a node on this path cannot be this node itself. Any intermediate node can show up arbitrary times at any place. Reflective cycles (a loop which is an edge that connects a node to itself) are not considered because they have no effect on the adversary's ability to infer the sender identity. To the adversary, the reflective cycle can be regarded as the node itself.

Clearly, Strategy T1 can be used to construct the simple path. Strategy T2 & Strategy T3 are for the complicated path.

V. Threat Model and Adversary Algorithm

Section 4.1 defines the adversary's capabilities in term of a threat-model. Section 4.2 describes how the adversary can take advantage of these capabilities to monitor the network activities, and use collected information to determine the identity of the sender of a message.

A. Threat Model

This paper considers a passive adversary model. By passively monitoring messages in transit, the adversary collects information and detects the identity of senders. To have access to messages, the adversary has previously compromised a number of nodes $\{C_k : 1 \leq k \leq M, C_k \in V\}$. The receiver is assumed to be compromised as well[2]. An agent of the adversary at a compromised node observes & collects all the information

[2]This assumption proves true in many realistic situations: For example, an e-mail author might want to hide its identity from the recipient. Similarly, a visitor to a web page might want to hide its identity from the web server.

TABLE I
SPECIAL NOTATION.

Notation	Description
$\mathbf{N}$	number of nodes in the system
$\mathbf{V}$	$V = \{v_1, v_2, \ldots, v_N\}$ set of nodes in the system
$\mathbf{M}$	number of compromised nodes, $M \leq N$
$\mathbf{L}$	path length
$\mathbf{L'}$	guessed path-length by the adversary
$\mathbf{K}$	number of compromised nodes on the rerouting path, $0 \leq K \leq \min[M, L]$
$\mathbf{s}$	true sender of the message
$\mathbf{R}$	receiver of the message
$\mathbf{R.P}$	immediate predecessor of the receiver
$\mathbf{NS}$	set of nodes which are definitely not the true sender
$\mathbf{CP}$	set of compromised nodes which are on the rerouting path
$\mathbf{CNP}$	set of compromised nodes which are not on the rerouting path
$< t_{C_i}, P_{C_i}, S_{C_i}, C_i >$	information reported from the compromised node C_i
t_{C_i}	time instant when the message traverses C_i
P_{C_i}	immediate-predecessor of node C_i
S_{C_i}	immediate-successor of node C_i
$\mathbf{\Omega}$	all possible events that the adversary can observe

in the message, and thus reports the immediate predecessor & successor node for each message traversing the compromised node. Assume also that the adversary collects this information from all the compromised nodes, and uses it to derive the identity of the sender of a message. In Fig. 3, nodes $1, 5, 7, 8, 10, 15$, and R are compromised. A message is transmitted along the rerouting path $< 0, 5, 2, 7, 11, 8, R >$, determined by a path selection algorithm (assuming that Strategies L1 & T1 are used) of the anonymous communication system from its source 0, to its destination R. In this case, the true sender 0 can definitely be determined by the adversary, because the adversary can construct the path $< 0, 5, 2, 7, 11, 8, R >$ completely based on the information collected from the compromised nodes.

This analysis is based on the worst-case assumption in the following sense:

- The sender has no information about the number or identity of compromised nodes. The route selection therefore does not rely on any knowledge about which nodes are compromised. Thus, some compromised nodes might be on the rerouting path.
- The adversary has full knowledge of the path selection algorithm.
- The adversary collects all the information from the agents on the compromised nodes, and attempts to derive the true identity of the sender.
- To simplify this discussion, without loss of much generality, assume that messages which traverse these compromised (malicious) nodes on the path can be correlated; i.e., one can determine whether a message received by a compromised node is the same one received by another compromised node on the path at an earlier time. For many anonymous communication systems, e.g., Crowds, this is possible by comparing the payload no matter whether it is encrypted or not (Neither end-to-end payload encryption nor link-by-link payload encryption can prevent such type of correlation). For future work, we leave more complicated cases, such as MIX-type systems, or any case where the messages cannot be completely correlated.

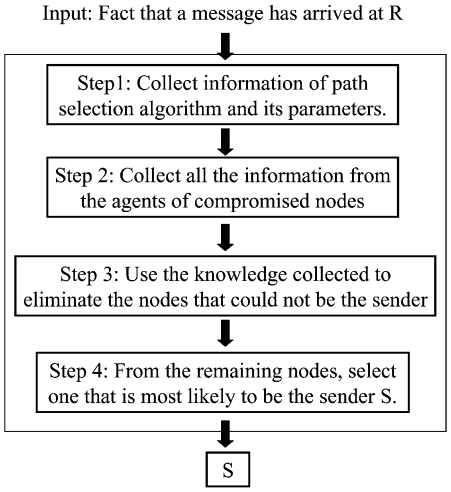

Fig. 4. Framework for Adversary Algorithm (Fixed Path Length).

In previous evaluations of ACS, various attack models were assumed [26], [28], [33]. Many of these models are special cases of the model described here.

- Attack by observing-respondent [28] and end-server [26]: These 2 cases correspond to the case where the receiver is compromised.
- Attack by active & passive path traceback [28], and collaborating-jondos [26]: These 2 examples correspond to the case in the model in this paper where the rerouting path can be reconstructed by the attacker using the routing information, or other monitoring information from at least $\lceil (L/2) \rceil$ nodes on the rerouting path compromised.
- Attack by local eavesdropper [26], [28]: This corresponds to the case in the model in this paper where the sender of the message is compromised.

To formally discuss how the adversary derives the identity of the sender, Table I introduces a special list of notations.

Section 4.2 discusses how the adversary derives the identity of the sender for fixed & variable path lengths.

B. Overview of Adversary Algorithm for Fixed Path Length

The Adversary Algorithm derives the true identity of the message sender. Fig. 4 shows the framework for the adversary algorithm for fixed path lengths. In the first 2 steps, the adversary collects the information about the path selection algorithm and its parameters, and all the information from those compromised nodes. Following this, the adversary tries to eliminate the nodes which could not be the sender. Finally, the adversary algorithm returns a node S from the remaining nodes as the source of a message.

Define the detection-probability as the probability that the true sender s can be discovered: $\Pr\{S = s\}$. This measure is used to evaluate the anonymity behavior of anonymous communication systems.

C. Details of the Adversary Algorithm for Fixed Path-Lengths

This section discusses the details of each step in the adversary algorithm in Fig. 4.

First, the adversary collects the information about the path selection algorithm & its parameters, and all the information about network activities from those compromised nodes. The information collected by the adversary can be classified into 2 types:

- static (off-line) information, and
- dynamic (on-line) information.

Static information includes the knowledge about the path selection algorithm & its parameters, especially the path-length distribution. Dynamic information is collected at run-time, and is based on network activities, e.g., when & where messages come from and go to.

Step 1) `Collect information about the path selection algorithm & its parameters`. The adversary might already have full or partial knowledge of the path selection algorithm, and its parameters for the anonymous communication system under attack. Steps 2 & 3 assume that the adversary has all the necessary knowledge about the path-selection algorithm, with the exception of seed values for any random-number generators used by the path-selection algorithm. This assumption follows common practice in all security-related areas, because it makes little sense, for example, to assume that encryption algorithms are not known to the adversary.

Step 2) `Collect information from the agents residing on compromised nodes.` As discussed in Section 4.1, an adversary agent at a compromised node can discover & report the immediate predecessor & successor node for each message traversing the compromised node. Dynamic information is collected by the compromised nodes in the system as follows: Every compromised node on the path, say Node C_i, reports the tuple

$$< t_{C_i}, P_{C_i}, C_i, S_{C_i} >, \tag{3}$$

t_{C_i} is the time instant when the message traverses Node C_i, P_{C_i} is the immediate predecessor of Node C_i, and S_{C_i} is the immediate successor of Node C_i. The $(M-K)$ compromised nodes not on the path implicitly report that they saw no message. After collecting information from all compromised nodes, the adversary can sort these tuples collected by time t_{C_i} in ascending order. Let the ordered tuples be:

$$< t_{C_1}, P_{C_1}, C_1, S_{C_1} >, < t_{C_2}, P_{C_2}, C_2, S_{C_2} >, \ldots, < t_{C_K}, P_{C_K}, C_K, S_{C_K} > .$$

These collected tuples are ω, the fact the adversary can learn by the observation of the system. In the remainder of this paper, F is used to represent all the information obtained by the adversary in Steps 1 & 2.

Following this, the adversary attempts to derive the probability that the true sender s can be identified

$$\Pr\{S = s \,|\, F = \omega\}, \quad s \in V. \tag{4}$$

TABLE II
ELIMINATION RULES FOR SIMPLE PATHS.

Rule	Pre-condition	Action
R1	(P_{C_1} = NULL) and ($S_{C_1} \neq$ NULL)	NS := NS $\cup (V \setminus \{C_1\})$
R2	($P_{C_1} \neq$ NULL) and the partial path from C_1 to R is from C_1 to is completely-identified and its length is L'	NS := NS $\cup (V \setminus \{P_{C_1}\})$
R3	$\forall\, v \in$ CNP	NS := NS $\cup \{v\}$
R4	$L > 0$	NS := NS $\cup \{R.P\}$
R5	$\forall k > 1$, $C_k \in$ CP	NS := NS $\cup \{C_k, P_{C_k}, S_{C_k}\}$
R6	$P_{C_1} \neq$ NULL	$NS := NS \cup \{C_1, S_{C_1}\}$
R7	($P_{C_1} \neq$ NULL) and the partial path from C_1 to R is completely-identified and its length is less than L'	$NS := NS \cup \{P_{C_1}\}$

$F = \omega$ represents the event that the adversary collected is exactly the fact ω (those tuples).

Step 3) `Determine a set of potential senders.` Based on the information collected in the first 2 steps, the adversary eliminates nodes from the set of possible senders of the message by applying the following rules. The node set V can be classified into 3 subsets:

i) NS,
ii) $\{$ Node P_{C_1} : Node P_{C_1} is the immediate predecessor of the first compromised node C_1 on the path$\}$,
iii) PS.

The NS represents the set of nodes excluded from being the true sender. The remaining nodes, except Node P_{C_1}, form the set of potential senders, PS. $\mathrm{PS} \cup \{P_{C_1}\}$ contains the nodes likely to be the true sender.

It follows that

$$\mathrm{NS} \cup \{P_{C_1}\} \cup \mathrm{PS} = V; \tag{5}$$

$$|\mathrm{NS}| + |\mathrm{PS}| + 1 = N. \tag{6}$$

For each node $v \in \mathrm{NS}, \Pr\{S = v \,|\, F = \omega\} = 0$. For each node $v \in \mathrm{PS}$, $\Pr\{S = vF = \omega\} > 0$. Node P_{C_1} can also be the sender if $\Pr\{S = P_{C_1} \,|\, F = \omega\} > 0$. It follows that the true sender must be a node in the set $\{P_{C_1}\} \cup \mathrm{PS}$.

Now construct node set NS. The set of non-senders, NS, is determined with an elimination algorithm. Initially, NS is the empty set $\emptyset$, and PS is the system-node set V. Following this, a sequence of nodes is eliminated from PS (and so added to NS) by applying a set of elimination-rules.

For simple paths, Table II lists a set of elimination rules.

In the remainder of this Section 4.3, the partial path between 2 compromised nodes on the rerouting path has been completely identified when all the nodes on the rerouting path have been identified between the 2 compromised nodes. For simple paths, this means that all the nodes on the partial path are either compromised nodes, or the immediate predecessors or immediate successors of compromised nodes; For 2 "adjacent" compromised nodes (there is no compromised node between them), the immediate successor of the first compromised node is either the immediate predecessor of the second compromised node, or the second compromised node.

If the sender itself is compromised ($c_1 = s$), then the sender can be trivially identified. All remaining nodes are therefore nonsenders. (**Rule R1**)

If the adversary can construct the whole rerouting path, and the predecessor of the first compromised node is the head of the path, then the sender again is identified. All remaining nodes are therefore nonsenders. (**Rule R2**)

Compromised nodes not on the path do not see a message. Therefore they cannot be the true sender, and so can safely be added to NS. (**Rule R3**)

Whenever $L > 0$, i.e., there is at least 1 intermediate node, the immediate predecessor of the receiver cannot be the true sender. Otherwise, the path length would be 0. (**Rule R4**)

Except for the immediate predecessor of the first compromised node, none of the compromised nodes themselves, and their immediate predecessors & immediate successors can be the true sender. (**Rules R5** & **R6**)

The immediate predecessor of the first compromised node, p_1, could not be the true sender if the path from this first compromised node to the receiver R can be completely identified, and the length of this identified path is less than L. Otherwise, the path length would be less than L. Thus, p_1 is not the sender. (**Rule R7**)

So far, NS is constructed.

For complicated paths, the elimination rules are more complicated, but similar to that for simple paths.

Step 4) `Among all possible senders, select the one most likely to be the sender.` The adversary determines which node is most likely to be the true sender among the set of potential senders $\{P_{C_1}\} \cup \mathrm{PS}$, and returns this node as the sender of the message.

First, calculate $\Pr\{S = P_{C_1} \,|\, F = \omega\}$. We can definitely determine $\Pr\{S = P_{C_1} \,|\, F = \omega\} = 0$ if Elimination Rules $R1$ or $R7$ have been applied, or $\Pr\{S = P_{C_1} \,|\, F = \omega\} = 1$ if Elimination Rule $R2$ has been applied.

Otherwise, by law of total probability, $\Pr\{S = P_{C_1} \,|\, F = \omega\}$ is derived as follows

$$\Pr\{S = P_{C_1} \,|\, F = \omega\} = \sum_{k=0}^{l'} \Pr\{S = P_{C_1} \,|\, F = \omega, C_1 = I_k\} \cdot \Pr\{C_1 = I_k \,|\, F = \omega\} = \Pr\{C_1 = I_1 \,|\, F = \omega\} \tag{7}$$

where I_k represents the kth intermediate node on the path.

Calculation of $\Pr\{C_1 = I_k \,|\, F = \omega\}$ is in [17].

If $s = P_{C_1}$, then

$$\Pr\{S = s \,|\, F = \omega\} = \Pr\{S = P_{C_1} \,|\, F = \omega\}.$$

Otherwise, calculate $\Pr\{S = s \,|\, F = \omega\}$ as follows.

In some cases, the physical location of the sender & of the compromised nodes, or some other information, can give rise to an *a priori* probability p_i for Node i to be the true sender. Thus calculate the probability that the true sender can be discovered as

$$\Pr\{S = s \,|\, F = \omega\} = (1 - \Pr\{S = P_{C_1} \,|\, F = \omega\}) \cdot \frac{p_s}{\sum_{v \in \mathrm{PS}} p_v}. \tag{8}$$

In most cases, the adversary does not have such knowledge. So in the remainder of this paper, without loss of generality, it is assumed that any node in PS can be the sender of a message with equal probability. Thus, the probability that the identity of the sender can be discovered is

$$\Pr\{S = s \,|\, F = \omega\} = \frac{1 - \Pr\{S = P_{C_1} \,|\, F = \omega\}}{|\mathrm{PS}|} = \frac{1 - \Pr\{S = P_{C_1} \,|\, F = \omega\}}{N - |\mathrm{NS}| - 1}. \tag{9}$$

D. Correctness & Completeness of Elimination Rules

For an elimination algorithm to be effective, it must eliminate as many nonsenders as possible, without ever eliminating the true sender. This section shows that the elimination-algorithm defined by rules $R1$ through $R7$ satisfies these requirements.

For an elimination algorithm to never mistakenly-eliminate the true sender, the elimination rules must be correct.

Definition 1: **Correctness of elimination rules**: A set of elimination rules is correct if these rules do not mistakenly eliminate the true sender. A correct elimination algorithm uses a correct set of elimination rules.

For an elimination algorithm to output the node most likely to be the true sender, the elimination rules must be complete.

Definition 2: **Completeness of elimination rules**: A set of elimination rules is complete if no other node can be correctly eliminated after this set of rules is applied. A complete elimination algorithm uses a complete set of elimination rules.

For a simple path, it can be shown that rules $R1$ to $R7$ are correct & complete.

Theorem 1: Correctness of Elimination Rules—Rules $R1$ to $R7$ never mistakenly eliminate the true sender.

Theorem 2: Completeness of Elimination Rules—No other nodes can be eliminated (added into NS) after rules $R1$ to $R7$ are applied.

The proofs of Theorems 1 & 2 are in [17].

Similarly, the correctness & completeness of elimination rules for the case of complicated paths can be proved.

All the elimination algorithms partition the set of nodes into potential senders and nonsenders. Without further information available, all potential senders have the same probability of being the true sender. It therefore follows that a correct & complete elimination algorithm performs best among all elimination algorithms; i.e., it finds the true sender with highest probability.

E. Extension to Variable Path-Length

Section 3 discussed how randomized re-routing decisions at the intermediate nodes give rise to path lengths which follow a probability distribution. The adversary algorithm for fixed path-lengths in Fig. 4 is extended to this case.

Let the path-length follow the distribution

$$\Pr\{L = i\} \text{ for } a \le i \le b. \tag{10}$$

To calculate $\Pr\{S = s \,|\, F = \omega\}$, the path length must be derived. For fixed-length paths, the length is known to the adversary. However, for variable-length paths, given the fact ω that the adversary learns from the observation of the system, the adversary first derives the lower & upper bounds of possible path length, and then derives the probability that each node can be identified as the true sender.

As discussed in Section 4.1, the adversary collects information from the M compromised nodes in the system. $(M - K)$ compromised nodes, not on the path, report that no message traversed them, and that each of K compromised nodes on the path reports a tuple of information about when & where the message came from & went to:

$$\langle t_{C_k}, P_{C_k}, C_k, S_{C_k} \rangle, \quad \text{where } 1 \le k \le K.$$

In general, these tuples can be concatenated. Formally, 1 tuples $< t_{C_i}, P_{C_i}, C_i, S_{C_i} >$ and $< t_{C_{i+1}}, P_{C_{i+1}}, C_{i+1}, S_{C_{i+1}} >$ can be concatenated if and only if

$$S_{C_k} = A_{C_{k+1}}, \tag{11}$$
$$S_{C_k} \ne C_{k+1}, \tag{12}$$
$$A_{C_{k+1}} \ne C_k; \tag{13}$$

or if

$$S_{C_k} = C_{k+1}, \tag{14}$$
$$A_{C_{k+1}} = C_k. \tag{15}$$

Thus, an adversary at some time instant can observe a sequence of J fragments of a path, which the message traverses

$$< *, f_1, *, f_2, \ldots, *, f_J >, \tag{16}$$

where f_i is the ith completely-identified partial-path fragment, and any 2 fragments can not be combined; i.e., each fragment contains the maximum possible tuples. The partial path between 2 compromised nodes on the re-routing path has been completely identified when all the nodes on the re-routing path have been identified between the 2 compromised nodes. This means that all the nodes on the partial path are either compromised nodes, or the immediate predecessors or immediate successors of compromised nodes; and for 2 "adjacent" compromised nodes (there is no compromised node between them), the immediate successor of the first compromised node is either the immediate predecessor of the second compromised node, or is the second compromised node. The adversary knows the number of these completely identified path fragments, and the length of each path fragment.

Now derive the lower & upper bounds of possible path lengths. For example, the lower bound c is the sum of the lengths of each completely identified path fragments, because all the constructed path fragments can be adjacent (i.e., there is no intermediate node between any 2 time-adjacent path fragments). The upper bound d can also be defined to be $(b - (M - K))$, where b is the maximum path-length defined in the path-length distribution $\Pr\{L = i\}$ for $a \le i \le b$.

Unless the path from the sender to the receiver has been completely identified, every node except the $(M-K)$ compromised nodes reporting no message traversed are always possible to be on the rerouting path. As Theorem 3 shows, there are no better bounds than c & d.

Theorem 3: The lower & upper bounds (c & d) derived above are tight, i.e., there do not exist bounds e & f satisfying $e < c \le l \le d < f$.

Theorem 3 can be easily proved by contradiction.

As assumed, the adversary knows the path-length distribution of the variable path-length strategy, i.e., $\Pr\{L = l\}$, where $0 \le a \le l \le b$; by law of total probability, the adversary can calculate $\Pr\{S = s \,|\, F = \omega\}$ by:

$$\Pr\{S = s \,|\, F = \omega\} = \sum_{l'=c}^{d} \Pr\{S = s \,|\, F = \omega, L' = l'\} \cdot \Pr\{L' = l' F = \omega\}, \tag{17}$$

where

$$\Pr\{L' = l' \,|\, F = \omega\} = \frac{\Pr\{F = \omega \,|\, L = l'\} \cdot \Pr\{L = l'\}}{\sum_{i=c}^{d} \Pr\{F = \omega \,|\, L = i'\} \cdot \Pr\{L = i\}}, \quad \text{for } c \le l' \le d. \tag{18}$$

$\Pr\{S = s \,|\, F = \omega, L' = l'\}$ can be calculated by the algorithm for the fixed path-lengths discussed in Section 4.3, for any given fixed path-length l'.

From the law of total probability,

$$\Pr\{S = s\} = \sum_{\omega \in \Omega} \Pr\{S = s \,|\, F = \omega\} \cdot \Pr\{F = \omega\}, \tag{19}$$

Ω is the set of all possible events which the adversary might observe. As discussed in Section 4.2, this measure is used to evaluate the anonymity behavior of ACS.

VI. Security Analysis

This section analyzes the impact of path selection strategies on the probability that the true identity of the sender can be discovered ($\Pr\{S = s\}$). The analytic results are presented first; then the numerical results are presented.

A. Analytic Results

Derive the conditional probability that the identity of the true sender is discovered, given that there are exactly i compromised nodes on a randomly selected simple path. This provides the basis for deriving general formulas for the probability that the true sender can be discovered. Then, derive closed-form solutions for special cases.

There are $\binom{L+1}{i}$ instances of simple paths which have length L, and contain exactly i compromised nodes[3]. For example, let $L = 4$ & $i = 1$; then there are $\binom{4+1}{1} = 5$ instances:

$$\begin{aligned}
&\text{Instance0}: \ S^* \ i_1 \ i_2 \ i_3 \ i_4 \ R^* \\
&\text{Instance1}: \ S \ i_1^* \ i_2 \ i_3 \ i_4 \ R^* \\
&\text{Instance2}: \ S \ i_1 \ i_2^* \ i_3 \ i_4 \ R^* \\
&\text{Instance3}: \ S \ i_1 \ i_2 \ i_3^* \ i_4 \ R^* \\
&\text{Instance4}: \ S \ i_1 \ i_2 \ i_3 \ i_4^* \ R^*
\end{aligned}$$

i_k represents intermediate node $\#k$ on the path; S is the sender; and R is the receiver. The * indicates that a node is compromised.

Let $f(i, j, L)$ be the detection probability for instance j

$$\Pr\{S = s \,|\, i, j, L\} = f(i, j, L). \tag{20}$$

The value for $f(i, j, L)$ is calculated by applying the elimination rules to the instance. For example, for Instance 3, it is known that i_2, i_3, & i_4 cannot be the sender (per rules R1, R2, & R3, respectively). Thus, for this instance, the nonsender set is

$$\text{NS} = \{i_2, i_3, i_4\}. \tag{21}$$

Then, by (9),

$$f(1, 3, 4) = \frac{1}{N - |\text{NS}|} = \frac{1}{N - 3}. \tag{22}$$

But for Instance 1, because the true sender s is the immediate predecessor of the compromised node i_1, according to (7): $f(1, 1, 4) = (1/2)$.

Formally, a procedure for computing $f(i, j, L)$ can easily be established by using the elimination rules as operators. The details of the procedure are not given here due to space limitation. With $f(i, j, L)$, one can obtain the probability that the sender's identity can be discovered for a given number of compromised nodes, as illustrated in lemma 1:

Lemma 1: For a simple path of length L, and i compromised nodes, the conditional probability that the identity of the true sender s is discovered is

$$\Pr\{S = s \,|\, i\} = \frac{1}{\binom{L+1}{i}} \sum_{j=0}^{\binom{L+1}{i}-1} f(i, j, L). \tag{23}$$

The lemma follows directly from the law of total probability, and the definition of $f(i, j, L)$.

1) General Formula: Based on lemma 1, derive the general results on the probability that the sender can be identified. First, deal with the case of simple paths with fixed length.

[3]The receiver is always compromised. Thus the i compromised nodes here do not include the receiver.

Theorem 4: For a system with M compromised nodes, and a simple path of length L, the probability that the true sender s can be discovered is

$$\begin{aligned}
&\Pr\{S = s\} \\
&\quad = \sum_{i=c}^{d} \left[\left(\frac{1}{\binom{L+1}{i}} \sum_{j=0}^{\binom{L+1}{i}-1} f(i, j, L) \right) \cdot \frac{\binom{M}{i}\binom{N-M}{L+1-i}}{\binom{N}{L+1}} \right], \qquad (24) \\
&\quad c \equiv \max[0, L + 1 - (N - M)], \qquad (25) \\
&\quad d \equiv \min[M, L]. \qquad (26)
\end{aligned}$$

Proof: Recall that the path-selection algorithm constructs the rerouting path by randomly choosing the intermediate nodes. All the nodes are chosen as intermediate nodes on the path with equal probability. Also recall that the path-selection algorithm does not know which nodes are compromised.

The number K of compromised nodes on the path can be any integer between c & d. By the law of total probability,

$$\Pr\{S = s\} = \sum_{i=c}^{d} \Pr\{S = s \,|\, K = i\} \cdot \Pr\{K = i\}; \tag{27}$$

$\Pr\{K = i\}$ represents the probability that a simple path of length L contains exactly i compromised nodes. Because both the nodes on the rerouting path & the compromised nodes are randomly chosen, we have

$$\Pr\{K = i\} = \frac{\binom{M}{i}\binom{N-M}{L+1-i}}{\binom{N}{L+1}}. \tag{28}$$

Substitute (23) & (28) into (27); the result is (24). ◀

Next, consider the case of simple paths with variable length. Formally, let r.v. L be the path length chosen by the sender, r.v. C be the number of compromised nodes on the selected path, L' be the path length considered by the adversary, and J be an index marking instances of paths that have length L and i compromised nodes.

Theorem 5: Assume that, in a system with M compromised nodes, simple paths with variable-length are used, and that the path length L conforms to the probability distribution $\Pr\{L = l\}$, where $a \le l \le b$. The probability that the true sender s can be discovered is

$$\begin{aligned}
\Pr\{S = s\} &= \sum_{l=a}^{b} \sum_{i=c}^{d} \frac{\sum_{j=0}^{z} \sum_{l'=u}^{v} g(l', i, j, l) \cdot q(l', i, j, l)}{\binom{l+1}{i}} \\
&\quad \cdot \Pr\{L = l\}, \\
c &\equiv \max[0, l + 1 - (N - M)], \\
d &\equiv \min[M, l], \\
z &\equiv \binom{l+1}{i} - 1, \\
u &\equiv h(i, j, l), \\
v &\equiv b - (M - i), \\
g(l', i, j, l) &\equiv \Pr\{S = s \,|\, L' = l', J = j, C = i, L = l\}, \\
q(l', i, j, l) &\equiv \Pr\{L' = l' J = j, C = i, L = l\}; \qquad (29)
\end{aligned}$$

and $h(i,j,l)$ is the minimum path length that the adversary can derive for case $\#j$ when the true path length is l, and i nodes on the path are compromised.

Like $f(\cdot)$, the functions $g(\cdot)$ & $q(\cdot)$ can be computed by applying operators related to the rules used in the Adversary Algorithm in Section 4.5.

Sketch of Proof: By the law of total probability,

$$\Pr\{S=s\} = \sum_{l=a}^{b} \Pr\{S=s \,|\, L=l\} \cdot \Pr\{L=l\}. \quad (30)$$

$\Pr\{S=s \,|\, L=l\}$ can be further computed by applying the law of total probability; (27) follows. ◀

So far, the general analytic results for the case of simple paths have been obtained. A similar approach can be taken to deal with the case of complicated paths. The only difference is that for the latter, the construction of the instance list for a path of length L with i compromised nodes is appreciably more complicated [17].

2) Special Cases: Section 5.1.1 derived the general results for computing the probability that the true sender can be discovered. This section analyzes 2 special cases to obtain closed-form formulas. While these 2 special cases are simple, their closed-form formulas help to analytically verify certain properties observed in the numerical analysis in Section 5.2.

In special case #1, consider a system using a fixed-length simple path with exactly 1 compromised node. The probability that the true sender can be discovered can be easily determined as follows.

Theorem 6: For a system having exactly 1 compromised node & using a fixed-length path, and for $1 \le N \le 3$,

$$\Pr\{S=s\} = 1. \quad (31)$$

When $N=4$,

$$\Pr\{S=s\} = \begin{cases} \frac{3}{4}, & \text{L=1 or L=2;} \\ \frac{7}{8}, & \text{L=3.} \end{cases} \quad (32)$$

When $N \ge 5$, see (33) at the bottom of the page. In special case #2, consider a system that uses variable-length paths conforming to the distribution

$$\Pr\{L=x\} = \begin{cases} p, & \text{x=0;} \\ 1-p, & \text{x=1.} \end{cases} \quad (34)$$

For this case, Theorem 7 can be obtained.

Theorem 7: For a system using variable-length paths with a length distribution conforming to (34),

$$\Pr\{S=s\} = \frac{M}{N} + (1-p)\cdot\frac{M}{(N-1)}\cdot\left(1-\frac{M}{N}\right) + \left(p + \frac{N-1-M}{N-1}\cdot(1-p)\right)\cdot\left(1-\frac{M}{N}\right) \cdot \left(p^2 + \frac{1}{N-1-M}(1-p)^2\right). \quad (35)$$

The proofs of these 2 theorems are in [17].

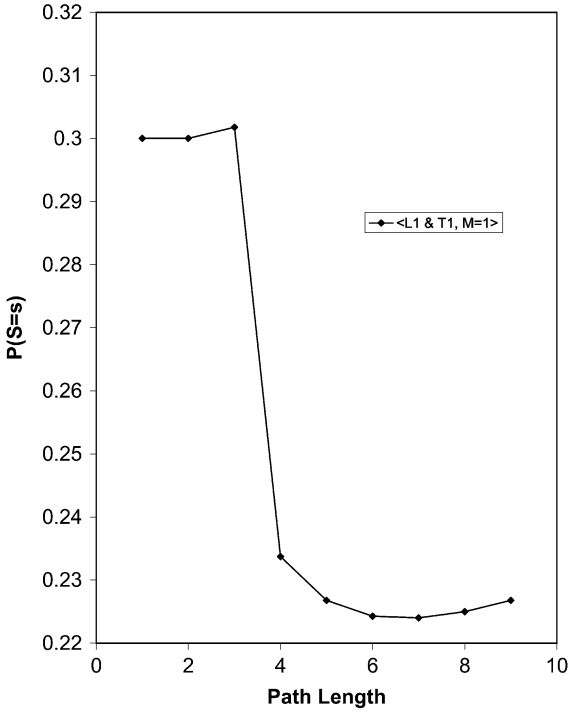

Fig. 5. $Pr\{S=s\}$ vs. path length. ($N=10$, and fixed length).

B. Observations

This section numerically computes the values of $\Pr\{S=s\}$ of several anonymous communication systems; and analyzes how path length, path topology, and the number of compromised nodes impact the values of $\Pr\{S=s\}$.

Throughout this section, $< L_i$ & $T_j, M=k >$ represents a system which uses Strategy L_i for path-length selection, Strategy T_j for path-topology selection, and has k nodes compromised.

1) Impact of Path Length:

a) Case of Fixed-Length Paths: Fig. 5 compares the detection probability of the sender for various systems with fixed path-length selection. Consider the following observations:

- In many cases, the value of $\Pr\{S=s\}$ decreases as the path-length increases. This coincides with general intuition; the more a message gets rerouted, the more difficult it is for the adversary to infer the sender.
- However, $\Pr\{S=s\}$ is not always monotonically decreasing as the path-length increases. For example, for the systems $< L_1$ & $T_1, M=1 >, \Pr\{S=s\}$ reaches its minimum at $L=7$. After that, $\Pr\{S=s\}$ becomes an increasing function of L.

The second observation is against intuition. One would anticipate that anonymity would be better with longer paths. The results here show that this is not always true. This phenomenon

$$\Pr\{S=s\} = \begin{cases} \frac{3}{N}, & L=1 \text{ or } L=2; \\ \frac{2}{N} + \frac{N^2-5N+7}{N\cdot(N-2)\cdot(N-3)}, & L=3; \\ \frac{2N^2-7N+4}{N\cdot(N-2)\cdot(N-3)} + \frac{2L^2+(6-4N)\cdot L+N^2-4}{N\cdot(N-2)\cdot(N-4)\cdot(L-1)}, & L \ge 4. \end{cases} \quad (33)$$

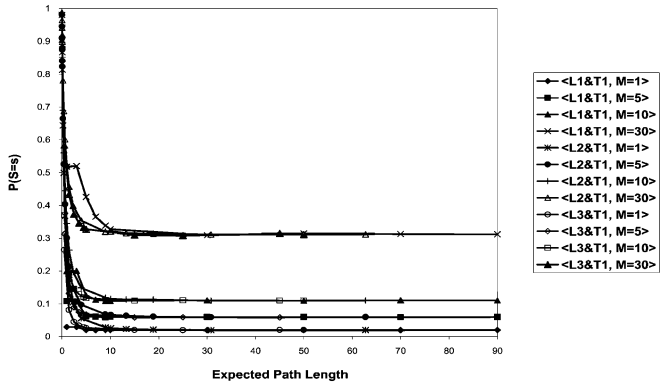

Fig. 6. $\Pr\{S = s\}$ vs. s-Expected Path-Length. ($N = 100$, Simple Path, and Variable Length).

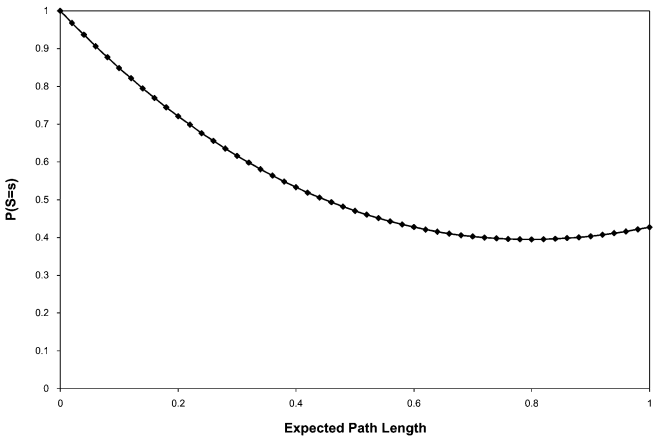

Fig. 7. $\Pr\{S = s\}$ vs. s-Expected path length. ($N = 11, M = 2$, Variable-Length Distribution (43).

can be explained: As the path-length increases, the possibility increases that compromised nodes are chosen as intermediate nodes. In this way, the adversary would gain better information about the path, improving its chance to identify the sender.

This observation can be confirmed analytically. From (33),

$$\frac{\partial \Pr\{S=s\}}{\partial L} = \frac{4L+6-4N}{N\cdot(N-2)\cdot(N-4)\cdot(L-1)} - \frac{2L^2+(6-4N)\cdot L+N^2-4}{N\cdot(N-2)\cdot(N-4)\cdot(L-1)^2}. \quad (36)$$

When $(\partial \Pr\{S=s\})/(\partial L) = 0$, then $\Pr\{S=s\}$ achieves its minimum value. In this case, it follows that $\Pr\{S=s\}$ achieves its minimum value when the path length is 7 (the extreme point is 6.657) for $N = 10$ & $M = 1$. Also, consider (33); by applying classical calculus techniques, it is easy to show that $\Pr\{S = s\}$ reaches its maximum at $L = 3$. That is, in this kind of system, increasing the path length does not necessarily improve the performance. It is best to choose $L = 7$; and it is worst to use $L = 3$.

b) Case of Variable-Length Paths: Here, the path-length is a r.v. Consider the following distributions of path length. Some of them are used in existing anonymous communication systems:

- Fixed path length

$$\Pr\{L=x\} = \begin{cases} 1, & \text{x = c;} \\ 0, & \text{otherwise.} \end{cases} \quad (37)$$

- Variable path-length conforming to geometric distribution

$$\Pr\{L=x\} = (1-P_f)\cdot P_f^x, \quad \text{for } x \geq 0. \quad (38)$$

- Variable path-length conforming to uniform distribution

$$\Pr\{L=x\} = \begin{cases} \frac{1}{b-a}, & a \leq x \leq b; \\ 0, & \text{otherwise.} \end{cases} \quad (39)$$

Fig. 6 shows the numerical results about detection probability under different path-length distributions. Fig. 6 suggests the following observations:

- When the s-expected path length is sufficiently large (e.g., larger than 15), the detection probability under various path-length distributions becomes very close, indicating that the length-distribution here is not critical.
- When the s-expected path length is small, $\Pr\{S = s\}$ is sensitive to the s-expected path length. Generally, when the s-expected path-length increases, $\Pr\{S = s\}$ decreases in most cases. Again, in some special cases, this may not be true.
- When the s-expected length is small (e.g., less than 10), the uniform distribution performs the best.

The nonmonotonicity of $\Pr\{S = s\}$ can again be verified with a closed-form formula. Consider the simple case where the variable path-length conforms to the distribution defined in (34); $(1-p)$ is the s-expected path length.

From (35),

$$\frac{\partial \Pr\{S=s\}}{\partial p} = -A\cdot p^2 - B\cdot p - C,$$
$$A \equiv \frac{3M\cdot(M-N)^2}{(N-1)\cdot(M-N+1)\cdot N},$$
$$B \equiv \frac{2(M-N)\cdot(-N^2+N+2M\cdot N-M^2+M)}{(N-1)\cdot(M-N+1)\cdot N},$$
$$C \equiv \frac{(M-N)\cdot(-M^2+M\cdot N-2-4M+2N)}{(N-1)\cdot(M-N+1)\cdot N}. \quad (40)$$

When $(\partial \Pr\{S=s\})/(\partial p) = 0$, i.e., $p = (-B \pm \sqrt{B^2-4AC})/(2A)$, then $\Pr\{S=s\}$ achieves its minimum value. There might be 2 values for p here; choose the one in $[0,1]$. Fig. 7 illustrates the impact of p on $\Pr\{S = s\}$. In this case, $\Pr\{S = s\}$ achieves its minimum value 0.395 when the s-expected path length is 0.796.

2) Impact of Path Topology: Due to the space limitation, consider only the impact of path topology under fixed path-length strategy here.

Compare the 3-path topology selection strategies of T1, T2, T3. Path lengths are fixed.

Fig. 8 shows the impact of complexity of path topology on $\Pr\{S = s\}$. The system with complicated paths (i.e., using strategies T_2 & T_3) performs better than the system with simple paths. However, the difference is relatively small.

3) Impact of Number of Compromised Nodes: Figs. 9, & 10 show how M, the number of compromised nodes, affects $\Pr\{S = s\}$. As anticipated, the probability that the true sender can be discovered increases as the number of compromised nodes increases. This conforms to general intuition. In addition, the deterioration of the system behavior goes sublinearly as the number of compromised nodes increases.

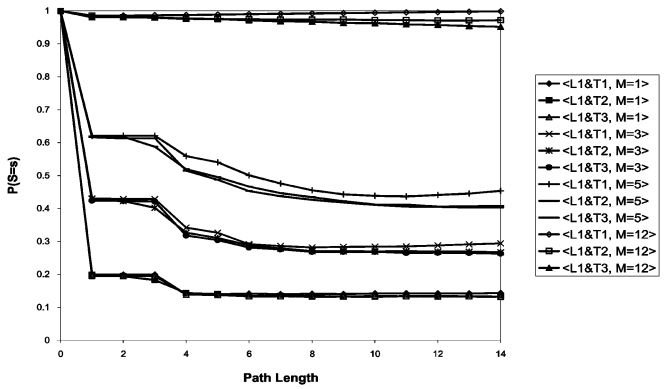

Fig. 8. $\Pr\{S = s\}$ vs. Path Length. ($N = 15$ and Fixed Length).

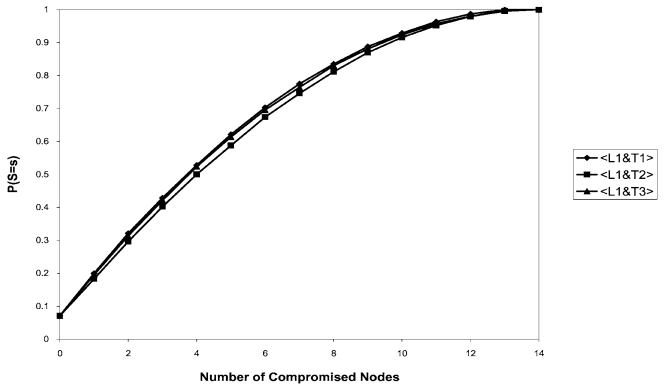

Fig. 9. $\Pr\{S = s\}$ vs. Number of Compromised Nodes($N = 15$ and Fixed Length).

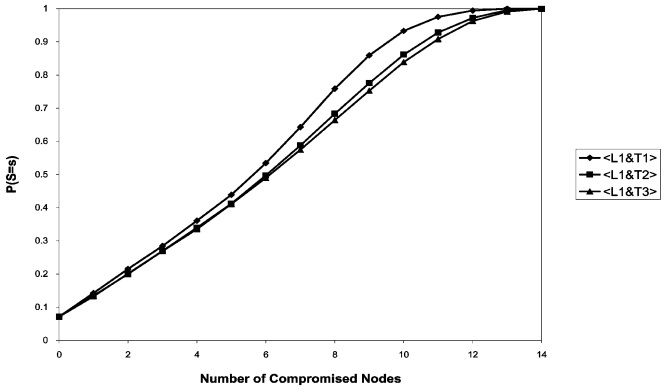

Fig. 10. $\Pr\{S = s\}$ vs. Number of Compromised Nodes($N = 15$ and Fixed Length $L = 10$).

VII. Remarks

This paper quantitatively analyzes the anonymity behavior of ACS, under various rerouting path-selection strategies. Several previously-used path selection methods are considered and the behavior of the adversary was modeled. The system anonymity was measured in terms of the probability that the identity of a message-sender can be discovered by the adversary. The main results from this study are:

1) A general intuition has been that the longer the rerouting path, the better the system's anonymity. While this is true in many cases, the analytic result here shows that the anonymity of the system might NOT always be improved as path-length increases. In some cases, a longer path of rerouting can result in a worse anonymity. Formulas are provided which allow designers to identify optimal rerouting path lengths under varying conditions.
2) While complicated paths perform better than simple ones, the difference on system anonymity due to different complexity of path topology is usually relatively small.
3) As anticipated, system anonymity becomes more vulnerable as the number of compromised nodes increases. In addition, the deterioration of the system behavior goes sublinearly as the number of compromised nodes increases.

From these analytic results, several existing ACS are not using the best path-selection strategies. For example, Freedom ([12], [35]) uses simple paths with fixed length of 3 in the current system, which gives the worst anonymity when there is 1 compromised node in the system.

Thus, the results in this paper can help system developers to properly-design their path-selection algorithm, and consequently improve their ACS.

Acknowledgment

This work was partially sponsored by:

- NSF under Contract Number EIA-0081761,
- DARPA under Contract Number F30602-99-1-0531,
- Texas Higher Education Coordinating Board under its Advanced Technology Program.

References

[1] The Anonymizer [Online]. Available: http:/www.anonymizer.com/
[2] Anonymous Remailer [Online]. Available: http://www.lcs.mit.edu/research/anonymous.html
[3] O. Berthold, H. Federrath, and M. Köhntopp, "Project anonymity and unobservability in the Internet," in *Proc. 10th Conf. on Computers, Freedom Privacy: Challenging the Assumptions*, 2000, pp. 57–65.
[4] D. Chaum, "Untraceable electronic mail, return addresses, and digital pseudonyms," *Communications of the ACM*, vol. 24, no. 2, pp. 84–88, 1981.
[5] ——, "The dining cryptographers problem: Unconditional sender and recipient untraceability," *J. Cryptology*, vol. 1, no. 1, pp. 65–75, 1988.
[6] R. C. H. Cheng, "Generating beta variatew with nonintegral shape parameters," *Communications of the ACM*, vol. 21, no. 4, pp. 317–322, 1978.
[7] J. Claessens, B. Preneel, and J. Vandewalle, "Solutions for anonymous communication on the Internet," in *Proc. IEEE 33rd Annual Conf. on Security Technology*, 1999, pp. 298–303.
[8] W. Dai, PipeNet 1.1.
[9] J. D. Faires and R. L. Burden, *Numerical Methods*: PWS-KENT Publishing Co, 1993.
[10] E. Gabber, P. B. Gibbons, Y. Matias, and A. Mayer, "How to make personalized web browsing simple, secure, and anonymous," in *Proc. Financial Cryptography'1997*, LNCS 1318, 1997, pp. 17–31.
[11] J. E. Gentle, *Random Number Generation and More Monte Carlo Methods*: Springer-Verlag, 1998.
[12] I. Goldberg and A. Shostack, Freedom Network 1.0 Architecture and Protocols, 1999.
[13] D. Goldschlag, M. Reed, and P. Syverson, "Onion routing for anonymous and private internet connections," *Communications of the ACM*, vol. 42, no. 2, pp. 39–41, 1999.
[14] Y. Guan *et al.*, "Preventing traffic analysis for real-time communication networks," in *Proc. IEEE MILCOM1999*, vol. 1, Nov. 1999, pp. 744–750.
[15] Y. Guan, X. Fu, R. Bettati, and W. Zhao, "Efficient traffic camouflaging in mission-critical QoS-guaranteed networks," in *Proc. IEEE Information Assurance and Security Workshop*, June 2000, pp. 143–149.
[16] Y. Guan *et al.*, NetCamo: Camouflaging network traffic for QoS-guaranteed mission critical applications, in IEEE Trans, System, Man, and Cybernetics, vol. 31, no. 4, pp. 253–266, July 2001. Special Issue on Information Assurance.

[17] Y. Guan, X. Fu, R. Bettati, and W. Zhao, "A Quantitative Analysis of Anonymous Communications," Dept. of Computer Science, Texas A&M University, Technical Report TR01-016, July 2001.
[18] C. Gülcü and G. Tsudik, "Mixing e-mail with babel," in *Proc. 1996 Symp. Network and Distributed System Security*, 1996, pp. 2–16.
[19] C. A. R. Hoare, *Communicating Sequential Processes*: Prentice-Hall, 1985.
[20] A. Jerichow *et al.*, "Real-time mixes: A bandwidth-efficient anonymity protocol," *IEEE J. Selected Areas in Communications*, vol. 16, no. 4, pp. 495–509, 1998.
[21] Lucent Personalized Web Assistant (2001). [Online]. Available: http://www.bell-labs.com/projects/lpwa
[22] S. Kent and R. Atkinson, "Security Architecture for the Internet Protocol,", RFC2401, 1998.
[23] R. E. Newman-Wolfe and B. R. Venkatraman, "High level prevention of traffic analysis," in *Proc. Seventh Annual Computer Security and Applications Conf.*, 1991, pp. 102–109.
[24] ——, "Performance analysis of a method for high level prevention of traffic analysis," in *Proc. 8th Annual Computer Security and Applications Conf.*, 1992, pp. 123–130.
[25] M. Reed, P. Syverson, and D. Goldschlag, "Anonymous connections and Onion routing," in *IEEE J. Selected Areas in Communications*, vol. 16, 1998, pp. 482–494.
[26] M. K. Reiter and A. D. Rubin, "Crowds: Anonymity for web transactions," *ACM Trans. Information and System Security*, vol. 1, no. 1, pp. 66–92, 1998.
[27] S. Schneider and A. Sidiropoulos, "ESORICS'96," in *CSP and Anonymity*, E. Kurth, G. Martella, and E. Montolivio, Eds: Springer-Verlag, 1996, vol. 1146, pp. 198–218. LNCS.
[28] C. Shields and B. N. Levine, "A protocol for anonymous communication over the Internet," in *Proc. 7th ACM Conf. Computer & Communication Security*, Nov. 1–4, 2000, pp. 33–42.
[29] V. Scarlata, B. N. Levine, and C. Shields, "Responder anonymity and anonymous peer-to-peer file sharing," in *Proc. IEEE Int'l. Conf. Network Protocols (ICNP) 2001*, pp. 272–280.
[30] R. C. Summers, *Secure Computing*: McGraw-Hill, 1997.
[31] P. Syverson, D. Goldschlag, and M. Reed, "Anonymous connections and Onion routing," in *Proc. IEEE Symp. Security and Privacy, 1997*, pp. 44–54.
[32] P. Syverson and S. Stubblebine, "Group principles and the formalization of anonymity," in *World Congress on Formal Methods'1999*: Springer-Verlag, LNCS 1708, pp. 814–833.
[33] P. Syverson, G. Tsudik, M. Reed, and C. Landwehr, "Toward an analysis of Onion routing security," in *Proc. Workshop on Design Issues in Anonymity and Unobservability*, 2000, pp. 83–100.
[34] P. Syverson, M. Reed, and D. Goldschlag, "Onion routing access configuration," in *DISCEX 2000: Proc. DARPA Information Survivability Conf. & Expo*, pp. 34–40.
[35] Anton stiglic, in Personal Communication, Zero-Knowledge Systems Inc., May 2001.
[36] A. Teich, M. Frankel, R. Kling, and Y.-C. Lee, "Anonymous communication policies for the internet: Results and recommendations of the AAAS conference," *The Information Society*, vol. 15, no. 2, 1999.
[37] B. R. Venkatraman and R. E. Newman-Wolfe, "Performance analysis of a method for high level prevention of traffic analysis using measurements from a campus network," in *Proc. Tenth Annual Computer Security and Applications Conf.*, 1994, pp. 288–297.
[38] Report from the national workshop on internet voting released. presented at National Workshop on Internet Voting. [Online]. Available: http://www.netvoting.org/Resources/InternetVotingReport.pdf
[39] M. Waidner, "Unconditional sender and recipient untraceability in spite of active attacks," in *Proc. Eurocrypt*, 1989, pp. 302–319.
[40] M. Wright, M. Adler, B. N. Levine, and C. Shields, "An analysis of the degradation of anonymous protocols," in *Proc. ISOC Network and Distributed System Security Symp*, NDSS 2002, 2002.
[41] Zero-knowledge Systems (2001). [Online]. Available: http:/www.zero-knowledge.com/

Yong Guan is an Assistant Professor in the Department of Electrical and Computer Engineering at Iowa State University. He received his B.S. (1990) and M.S. (1996) in computer science from Peking University, China, and his Ph.D. (2002) in computer science from Texas A&M University. From 1990 to 1997, he worked as an assistant engineer (1990–1993) and lecturer (1996–1997) in Networking Research Group of Computer Center at Peking University, China. In 2002, he joined Iowa State University as a faculty member. His current research interests are in security issues in computer networks, distributed systems, and wireless and mobile ad-hoc networks.

Yong Guan received the best-paper award from the IEEE National Aerospace and Electronics Conference in 1998 and won 2nd place in graduate category of the Int'l. ACM student research contest in 2002.

Xinwen Fu is a Ph.D. student in the Department of Computer Science at Texas A&M University. He obtained his B.S. (1995) and M.S. (1998) degrees in electrical engineering from Graduate School of University of Science and Technology of China and Xian Jiaotong University, China, respectively. His research interests are Network Security and distributed systems. He won the 2nd prize in the International ACM student research contest in 2002.

Riccardo Bettati is an Associate Professor in the Department of Computer Science at Texas A&M University. He received his Diploma in Informatics (1988) from the Swiss Federal Institute of Technology (ETH), Zuerich, Switzerland, and his Ph.D. (1994) from the University of Illinois at Urbana-Champaign. From 1993 to 1995, he held a postdoctoral position at the Int'l. Computer Science Institute in Berkeley and at the University of California at Berkeley. His research interests are in real-time distributed systems, real-time communication, and network support for distributed applications.

Wei Zhao is currently an Associate Vice President for Research at Texas A&M University. He completed his undergraduate program in physics (1977) at Shaanxi Normal University, Xian, China. He received his M.Sc. (1983) and Ph.D. (1986) in computer and information science from the University of Massachusetts, Amherst. In 1990, he joined Texas A&M University where he has been a full professor in the Department of Computer Science since 1996. Between 1997 and 2001, he served as a department head.

He is an IEEE Fellow. His current research interests include secured real-time computing and communication, distributed operating systems, databases, and fault-tolerant systems. He has played critical leadership roles in projects NetEx and NetCamo. His research group has been recognized by various awards & prizes, including the outstanding paper award from the IEEE Int'l. Conf. on Distributed Computing Systems in 1992, the best paper award from the IEEE National Aerospace & Electronics Conf. in 1998, an award on technology transfer from the Defense Advanced Research Program Agency (DARPA) in 2002, and the 2nd prize in the international ACM student research contest in 2002.

He is an inventor for two US patents and has published over 180 papers in journals, conferences, and book chapters. He is active in professional services. He was an editor of the IEEE Trans. on Computers from 1992–1996. He is on the editorial board of the IEEE Trans. on Parallel and Distributed Systems. He was program chair'n. (1995) and general chair'n. (1996) of the IEEE Real-Time Technology and Applications Symposia. He served as program chair'n. (1999) and general chair'n. (2000) of the IEEE Real-Time Systems Symposia. He was the co-program chair'n. for the IEEE Int'l. Conf. on Distributed Computing Systems in 2001. He will be the co-general chair'n. of the IEEE Int'l. Conf. on Distributed Computing Systems in 2003. He will be Guest Editor for a special issue on security in parallel & distributed computing systems for the *IEEE Trans, on Parallel and Distributed Systems* to be published in 2003.