# Assuring Telecommunication Anonymity through Leakage Reduction

[1]Longy O. Anyanwu, Ed.D., [2]Jared Keengwe, Ph.D., [3]Gladys Arome, Ph.D.

[1]Mathematics and Computer Science, Fort Hays State University, Hays, Kansas, USA, loanyanwu@fhsu.edu
[2]Department of Teaching and Learning University of North Dakota, ND, USA , jared.keengwe@und.nodak.edu
[3]Department of Media and Instructional Tech. Florida Atlantic University, FL, USA, garome@fau.edu

***Abstract***

*Each Internet communication leaves trails here or there, that can be followed back to the user. Notably, anonymous communication schemes are purposed to hide users' identity as to personal, source and destination location and content information. Previous studies have shown that the average round trip times (RTT) leakage between network host location, $X_1$ and network destination location, $Y_1$, can be determined, [12]. Additionally, an attack from a web site with access to a network coordinate system can recover 6.8 bits/hr. of network location from one corrupt Tor router, [12]. Notably, no network capability is in existence to completely negate anonymity leakage in network latency, [12], thus, the minimization of anonymity leakage in network latency becomes critically salient. The purpose of this paper is to investigate network latency anonymity leaks, and propose practical techniques for their reduction. In this direction, we investigate the following technical question: what implementation techniques can be configured to truly reduce anonymity leaks using deployable systems. Here, an extension of the popular Tor security strategies and unique configuration of the popular network anonymity techniques (algorithms) for future implementation are presented.*

***Categories and Subject Descriptors:*** *Network security. Network anonymity loss reduction. Secure networks and communication. Anonymous communications.*

***General terms:*** *Network security, Reliable anonymity systems.*

## 1. Introduction

The Internet promises an ever-increasing variety of available services to anyone anywhere. This social and business convenience comes with compromises to privacy. On the Internet, users have few controls, if any, over the privacy of their actions. Each communication leaves trails here or there, and often, someone can follow these trails back to the user. Notably, anonymous communication schemes are purposed to hide users' identity as to personal, source and destination location, and content information. Frankly, anonymous communication on the Internet offers new opportunities but has ill-understood risks.

Two types of anonymity are required for complete anonymization [5]. Data anonymity filters out identifying data, such as the sender field in an e-mail. Connection anonymity obscures the communication patterns. Furthermore, there are four types of connection anonymity. Sender anonymity protects the identity of the initiator. Receiver anonymity protects the identity of the responder. Mutual anonymity [11] provides both sender and receiver anonymity. Unlinkability [15] means that an attacker cannot discern sender-receiver relationships. Even if the identity of one endpoint is compromised, the identity of the other endpoint cannot be linked to it. Thus, people create special networks to protect privacy, especially the identities of the entities participating in a communication via Internet connections. Thus, the main goal of the networks is to provide anonymity for their users. Each network employs some specific anonymous communication schemes (methods) to reach the goal.

## 2. Anonymous communication schemes

Anonymous communication is not new. It has been in use for a while now. As a matter of fact, the idea was first fielded by Chaum [3]. He suggested the transmission of messages through a server which mixes message packets from different users before forwarding them to the destination, thus, concealing the identity, location and destination, and content information between senders and receivers. Consequently, many anonymity schemes have emerged, and have been used rather widely. Network latency has become a major basis to construct de-anonymization schemes for network communication. It has also become a method of comparing network transmission rates. This method has infiltrated into the anonymity scheme market. High latency anonymity schemes deliver messages at long delay rates (such as the Mixmaster and Mixminion), [6]; [13]. With high latency schemes, more bandwidth is used. On the other hand, low latency systems transmit messages at a reasonably short delay rates. Such low latency protocol systems are typified by the popularly used Tor and AN.ON systems, [7]; [16]. The benefits of using low-delay anonymity are that anonymous communications use a variety application services including remote login and web browsing, although this functionality comes at the cost of reduced anonymity guarantees. In particular, most of these services are easily defeated by a global passive adversary using relatively straightforward attacks such as packet counting, [17]. Additionally, using packet counting attacks, an attacker with control of a subset, *S*, of the nodes in the system can trace a subset, *S*, of the connections made to colluding servers and subset, $S^2$of all connections running through the system, [19].

The possibility of using latency data in traffic analysis has been mentioned several times in previous works, apparently originating in 2001, [1]. Since then some studies have investigated the amount of network latency leakage [12]. Of course, to get an upper bound on the amount of information that can be leaked under the current Internet topology, the amount of information about a host that can be gained, given a precise estimate of its RTT to a randomly chosen host, may be measured. For the general Internet, in the above study, the MIT King data set was used [9]. Then for each source host A, we computed the expected number of bits in the RTT to a random destination host B by counting, for each B, the number of hosts C such that the confidence intervals for AB and AC overlapped. Taking this count as $N_B$ and the total number of hosts as N, the information gain for AB was computed as $\log_2$ (N/NB). The cumulative distribution of expected information gain for the two data sets used in the study is shown in Figure 1. For the King data set, the average number of bits from RTT per host is 3.64, the median is 3.8.

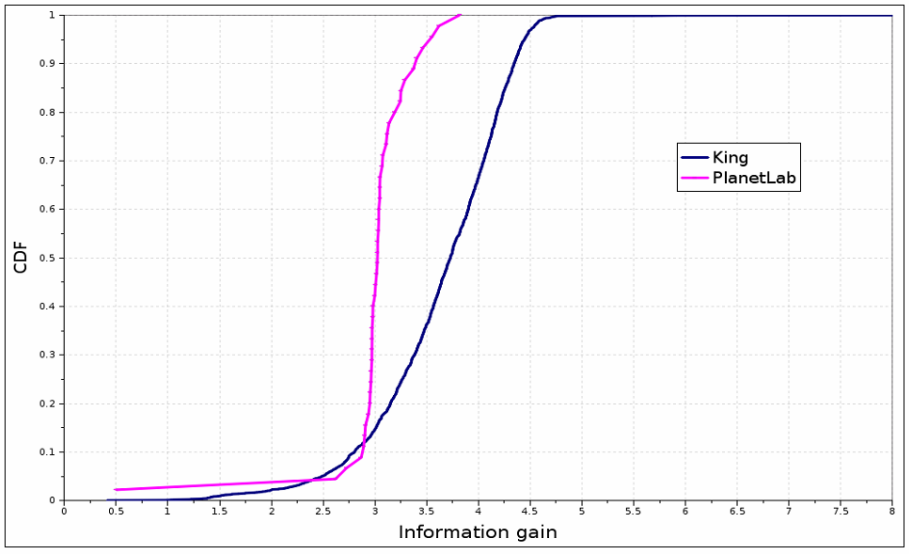


FIGURE 1: Cumulative distribution of expected information gain from RTT per host, for MIT King data set and PlanetLab nodes.

### 2.1. The Multicast Scenario

The multicast approach [2] provides a communication infrastructure that is reasonably resilient against both eavesdropping and traffic analysis. Using this protocol, entities representing applications communicate through a sequence of networked computing nodes, which is referred to as onion routers. Onion routers are generally application layer routers that realize Chaum MIXes. Onion routing connections proceed in three phases: connection setup phase, data transfer phase and connection termination phase. Over the Internet, anonymous systems [10], [18] use application level routing to provide anonymity through a fixed core set of MIXes as in the Onion Routing protocol. Each host keeps a global view of the network topology, and makes anonymous connections through a sequence of MIXes instead of making direct socket connections to other hosts. The relative percentage of malicious nodes and connectivity is shown in figure 2.

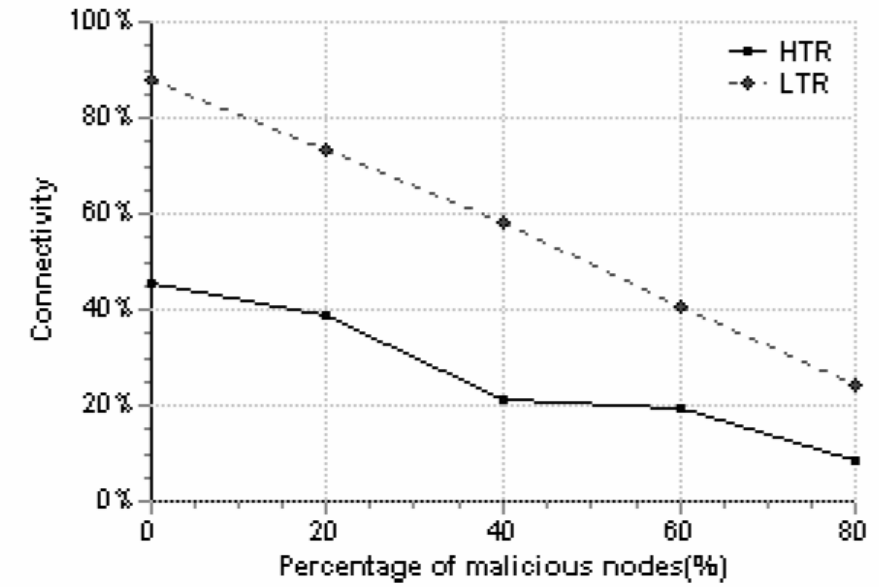


FIGURE 2: Connectivity vs. percentage of malicious nodes

Nonetheless, there are limitations to the capabilities of an attacker, simply because, access to any traffic of interest before it exits the anonymity service and arrives at his malicious servers is very rare. Some studies have investigated into the question: what information, outside of the actual bits of data packets delivered to the adversary, does a low-latency anonymity service leak, and to what extent does this leakage compromise the anonymity offered by the service? [12]). Several recent studies have explored the impact of the local attacker's access to information about the timing of events in a low-latency anonymity scheme, such as packet arrival times, using, for instance, the "circuit clogging" attack version of Murdoch and Danezis, [14], which relies on the observation that a sudden increase in the load of a Tor server will increase the latency of all connections running through it. Indeed, Murdoch and Danezis demonstrated how a corrupt Tor node and web server can exploit this property to determine the nodes in a Tor circuit, (the nodes that forward a given connection through the network).

### 2.2. The purpose of the paper.

The purpose of this paper is to investigate network latency anonymity leaks, and propose practical techniques for their reduction or even elimination. In this direction, we investigate the following technical question: what implementation techniques can be configured to truly reduce anonymity leaks? The emphasis is on deployable systems which provide strong anonymity against a strong attacker model for the Internet. The method used here, is to propose an extension of the popular Tor security strategies and to present a unique configuration of the popular network anonymity techniques (algorithms) for future implementation.

### 2.3. Typical Time-based (Latency) Attack

In such an attack, typically, the corrupt Tor node regularly sends packets on a loop through each Tor server, measuring the time the packets spend in transit. Then when the malicious server wishes to trace a connection, it modulates its throughput in a regular, on/off burst pattern. By correlating the delay at each Tor server against the timing of these burst periods, the attacker learns which nodes are in the circuit. Since the estimated number of Tor users (on the order of 105 as of April 2007) is less than the number of possible circuits (on the order of 108) seeing two connections that use the same circuit nodes is a strong indicator that the connections are from the same user. Thus at a minimum, timing information can leak the linkage between Tor connections.

Hopper, et al, in their paper titled "How Much Anonymity does Network Latency Leak?", made similar observations that typical malicious servers acting as local adversaries can observe the network latency of a connection made over a Tor circuit. They also observed that even in this scenario, if a client attempts to connect to two malicious servers (or make two connections to the same malicious server) using the same circuit, then the server-client RTTs of these connections (minus the RTT from the last node to the server) will be drawn from the same distribution, whereas other clients connecting to the server will have different RTTs. Based on this observation, they developed an attack on Tor that allows two colluding web servers to link connections traversing the same Tor circuit. The attack uses only standard HTTP, the most commonly mentioned Tor application layer, and requires no active probing of the Tor network and has very minimal bandwidth requirements. They tested this attack using several hundred randomly chosen pairs of clients and randomly chosen pairs of servers from the PlanetLab wide area testbed, [3], communicating over the deployed Tor network. Resultantly, they found suggestions that pairs of connections can have an equal error rate of roughly 17%, and the test can be tuned to support a lower false positive or false negative rate. Also, the publicly available MIT King data set, [9], a collection of pair wise RTTs between 1950 Internet

hosts, was analyzed to estimate the average amount of information that is leaked by knowing the RTT between a given host and an unknown host. The investigators found that, on average, knowing the RTT to a host from one known server yields 3.64 bits of information about the host ( and equivalently, it reduces the number of possible hosts from n to $n/2^{3.64} \approx 0.08n$). The expected bits gained per hour relative to connection, is shown in figure 3. Notably, several attack techniques exist. One widely used attack technique, the active client-identification attack, is briefly explained below.

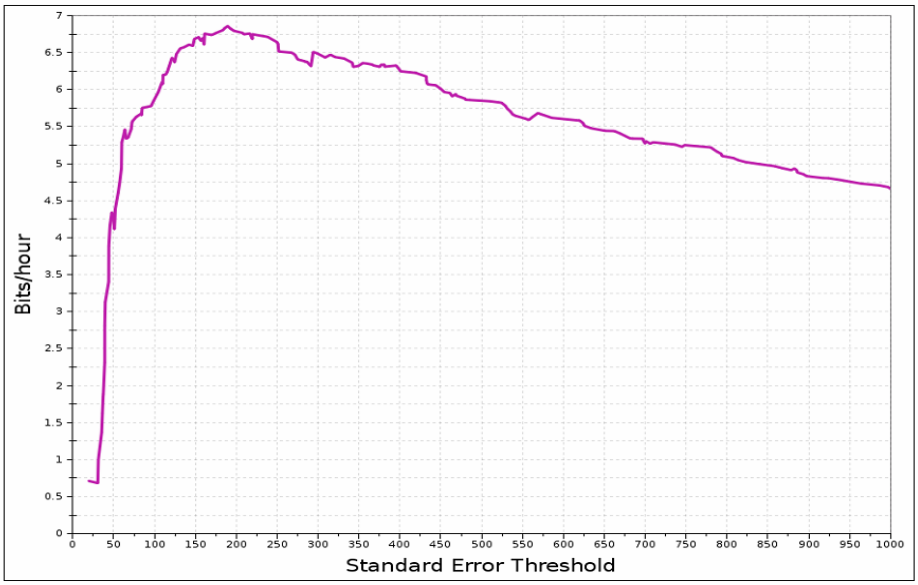


FIGURE 3: Expected bits per hour vs. 100-connection standard error threshold.

### 2.4. An Active Client-identification Attack.

Using standard protocols and taking advantage of repeated visits from a client, a malicious Tor server with access to latency oracle, can estimate the RTT between Tor servers and nodes in the RTT equivalence class of nodes of suspected client's locations. A simulated attack [12] was evaluated using over 200 runs with randomly chosen client/server pairs from the PlanetLab wide area testbed, using randomly chosen circuits among the currently deployed Tor nodes (as of Jan./Feb. 2007). The results suggest that a malicious server with a periodically reloading web page can recover, on average, about 6.8 bits of information about a client's location per hour. Thus a client's RTT equivalence class can be determined in 3 hours, on average. These results have serious implications for the design of low-latency anonymity schemes. In particular, they suggest that, without new ideas for path selection, adding delay to a connection may be unavoidable for security considerations. In turn, this has implications for design decisions: for example, if latency must be uniformly high, then TCP tunneling over such services will provide extremely low bandwidth; or if the latency of circuits can be masked with noise in the short term, then circuit lifetimes may need to be shortened. The Tor circuit constitutes the primary anonymous network system used in this study.

## 3. A brief overview of the tor system

The Tor system is a low-latency and bandwidth-efficient anonymizing layer for TCP streams. Its growing popularity and the availability of a test-bed deployment have proven to be a fertile ground for research on implementing and attacking low-delay anonymity schemes. Tor system works similarly to a circuit-switched telephone network, where a communication path, or circuit, is first established, over which all communication during a given session takes place. Anonymity is achieved by establishing that circuit through three nodes: an entry node, an intermediary (middleman), and an exit node. Only the entry node knows the identity of the client contacting it, in the form of its IP address. The middleman node knows the identities of both the entry and exit nodes, but not who the client is, or the destination he or she wishes to reach over the circuit. If the Tor server is an "exit" node, which provides a gateway between the Tor network and the Internet, it is responsible for making application-layer connections to hosts on the Internet, and serves as a relay between potentially non-encrypted Internet connections and encrypted Tor traffic. Thus, it knows the destination with whom the client wishes to communicate, but not the identity of the client. In this manner, no single node in the Tor network knows the identities of both communicating parties associated with a given circuit. All communications proceed through this encrypted tunnel.

Circuits are established iteratively by the client, who gets a list of Tor nodes and long-term keys from a directory service, selects a Tor node from that list (preferably one with high uptime and bandwidth), negotiates a communication key, and establishes an encrypted connection. To avoid statistical profiling attacks, by default each Tor client restricts its choice of entry nodes to a persistent set of three randomly chosen "entry guards". The circuit is then extended to additional nodes by tunneling through the established links. Link encryption, using ephemeral Diffie-Hellman key exchange for forward secrecy, is provided by SSL/TLS. To extend the circuit to another Tor node, the client tunnels that request over the newly-formed link. Traffic between Tor nodes is broken up into cells of 512 bytes each. Cells are padded to that size when not enough data is available. All cells from the client use layered (or "onion") encryption, in that if the client wishes for a message to be passed to example.com via Tor nodes A, B, and C (C being the exit node), the client encrypts the message with a key shared with C, then again with a key shared with B, and finally A. The message is then sent over the previously established encrypted tunnel to A (the entry node). A will peel off a layer of encryption, ending up with a message encrypted to B (note that A can not read this message, as A does not have the key shared between the client and B). A then passes on the message to B, who peels off another encryption layer, and passes the message to C. C removes the final encryption layer, ending up with a clear text message to be sent to example.com.

Messages can be any communication that would normally take place over TCP. Since there is significant cryptographic overhead (such as Diffie-Hellman key exchange and SSL/TLS handshake) involved with the creation and destruction of a circuit, circuits are reused for multiple TCP streams. However, anonymity can be compromised if the same circuit is used for too long, so Tor avoids using the same circuit for prolonged periods of time, giving circuits a client-imposed maximum lifetime1.

The biggest problem with the Tor network is its vulnerability to timing attacks. If an attacker sees a packet from the user to the first Tor router and shortly afterwards a packet from the last router to the final destination, it is possible to identify the user. This is an inherent issue of low-latency anonymizers and its solution is still an open research problem. Although, it has been suggested before that this information might be a potential avenue of attack [1], it is not known to us that leaking this information had any adverse effect on the anonymity provided by schemes like Tor. An example of a typical attack on a Tor system is briefly described bellow.

### 3.1. Typical Attack Against the Tor System:

When a client, using a timing-based attack, connects to the malicious web server, that server modulates its data transmission back to the client in such a way as to make the traffic pattern easily identifiable by an observer. At least one Tor server controlled by the adversary builds "timing" circuits through each Tor server in the network (around 800 as of January/February 2007[1]). These circuits all have length one, beginning and terminating at the adversarial Tor node. By sending traffic through timing circuits to measure latency, the adversary is able to detect which Tor servers process traffic that exhibits a pattern like that which the attacker web server is generating. Since Tor does not reserve bandwidth for each connection, when one connection through a node is heavily loaded, all others experience an increase in latency. By determining which nodes in the Tor network exhibit the server-generated traffic pattern, the adversary can map the entire Tor circuit used by the client.

Recent studies have suggested that an adversary may de-anonymize any stream for which that adversary controls the entry and exit nodes [19]. The probability of this occurrence in the short term (transient client connections) is $c(c-1)/r^2$, where c is the maximum number of nodes corruptible by the adversary in a fixed period of time, and r is the number of available Tor routers in the network. An adversary can determine if he or she controls the entry and exit node for the same stream by using a number of methods mentioned below, including fingerprinting and packet counting attacks. Indeed, it is expected that single-hop proxy services will leak more information about the client-proxy RTT, allowing fairly precise linking attacks, although the strength of the client location attack will be somewhat diminished against services that have a single proxy server location.

In summary, most existing literature on the topic focuses mainly on types of attacks and available individualized strategies for overcoming them. Notably, individualized solutions and troubleshooting leave a lot to be desired in today's ubiquitous and multi-platform communication applications. A holistic approach to network communication anonymity is critical.

### 3.2. Limitations of Existing Studies

1) The most serious of these limitations is the insufficiency of data on conditional information gain, that is, we cannot conclusively evaluate, from our data, how much additional information each run of an attack provides. This is due in part to limitations of an experimental method, which did not re-use clients; thus a "longitudinal" study may be needed to more accurately assess conditional information gain.

[1] TOR (the onion router) servers. http://proxy.org/tor.shtml, 2007.

2) Another limitation is that the client location attack assumes that a user repeatedly accesses a server from the same network location. This assumption may sometimes be invalid in the short term due to route instability, or in the long term due to host mobility. It seems plausible that the attack can still be conducted when circuits originate from a small set of network locations, such as a user's home and office networks, but the attack would be of little use in case of more frequent changes in network location. Thus, the question remains: Will replication of the experiments, using less widely-supported tools, such as persistent HTTP over Tor, produce the same results? Even the authors of a paper titled "How Much Anonymity does Network Latency Leak?" [12], themselves acknowledged that, of course, the answer is highly dependent on both the network topology (latency in a star topology would leak no information about a host's location) and the protocol in question. This is because it is conceivable that if so much noise is added to the network latency, the signal can be undetectable. Furthermore, there is room for evaluation of alternative methods of implementing RTT oracles, and perhaps for a more sophisticated testing procedure that avoids the expense of querying the RTT oracle for every pair of Tor entry node and candidate location.

## 4. Leakage reduction techniques

A number of techniques and best practices which can reduce the attacker's probability of success in client location attack have been suggested [12]. Four such combinations of configuration techniques are described below.

### 4.1. The utility of onion routers in the Tor System.

The use of onion routers in the Tor system can minimize the success probability of the Murdoch-Danezis attack by allocating a fixed amount of bandwidth to each circuit, independent of the current number of circuits, and doing "busy work" during idle time. This may undesirably compromise efficiency but certainly will hinder the success of client location attack. Additionally, by configuring Tor nodes to refuse to extend circuits to nodes which are not listed in the directory, their use of RTT oracles will be prevented. They can also drop ICMP ECHO REQUEST packets in order to raise the cost of estimating their network coordinates. Additionally, a DNS-based administrative disabling of Tor recursive lookups from "outside" will limit, if not preclude, the success of this attack. Although the security implications are not clear, making the Tor path selection algorithm latency-aware, by incorporating some notion of network coordinates into directory listings, thus, clients could construct circuits having an RTT close to one of a small number of possibilities, will reduce the high average circuit RTTs (of 5 sec[2]), reduce the effectiveness of latency-based attacks, and allow clients to explicitly trade-off some anonymity for better efficiency.

### 4.2. The Administrative Ingenuity Approach.

Tor administrative drop of ping packets and denial of other attempts to learn their network coordinates to accuracy, and the addition of sufficient delays (of forwarding data at the client) to make the RTT and timing characteristics servers independent of the underlying network topology, will hinder success probability. Furthermore, given the limited time period over which a Tor circuit is available for sampling, the introduction of high variance random delays in outgoing cells and selecting delays from an identical distribution at each Tor node would also make the timing distributions from different circuits look more alike, thus thwarting the circuit-linking attack. Of course, if the only way to thwart attacks based on latency and throughput is to add latency and restrict throughput, this would have serious implications for the design of low-latency anonymity systems and the quality of anonymity we can expect from such schemes

[2] Observed in the Hopper, et al study titled "How Much Anonymity does Network Latency Leak?", ***Communications*** of the ***ACM***, v.24 n.2 (2007), p.84-90.

### 4.3. The Multicast Technique to Network Anonymity.

With the multicast technique, the source node is disallowed from gathering and storing information about the network topology, [2]. Instead, the source node initiates a path establishment process by broadcasting a *path discovery* message with some trust requirements to all of neighboring nodes. Intermediate nodes satisfying these trust requirements insert their identification (IDs) and a session key into the *path discovery* message and forward copies of this message to their selected neighbors until the message gets to its destination. The intermediate nodes encrypt this information before adding it to the message. Once the receiver node receives the message, it retrieves from the message the information about all intermediate nodes, encapsulates this information in a multilayered message, and sends it along a reverse path in the dissemination tree back to the source node. Each intermediate node along the reverse path removes one encrypted layer from the message, and forwards the message to its ancestor node until the message reaches the source node. When the protocol terminates, the source node ends-up with information about all the trusted intermediate nodes on the discovered route as well as the session keys to encrypt the data transmitted through each of these nodes. The multicast mechanism and the layered encryption used in the protocol ensure the anonymity of the sender and receiver nodes.

The *path discovery* phase allows a source node *S* that wants to communicate securely and privately with node *R* to discover and establish a routing path through a number of intermediate wireless nodes. An important characteristic of this phase is that none of the intermediate nodes that participated in the *path discovery* phase can discover the identity of the sending node *S* and the receiving node *R*. The source node *S* triggers the *path discovery* phase by sending a *path discovery* message to all nodes within its transmission range. The *path discovery* message has five parts. The first part is the open part. It consists of message type, *TYPE*, trust requirement, *TRUST_REQ*, and a one-time public key, *TPK*. The trust requirement indicated by *TRUST_REQ* could be *HIGH, MEDIUM* or *LOW*. *TPK* is generated for each *path discovery* session and used by each intermediate node to encrypt routing information appended to the *path discovery* message. The second part contains the identifier IDR of the intended receiver, the symmetric key *KS* generated by the source node and *PLS* the length of the third part, *padding*, all encrypted with the public key *PKR* of the receiver.

### 4.4. The two-pronged Approach to Attack Detection.

Venkatraman and Agrawal [21] proposed an approach for enhancing the security of AODV protocol based on public key cryptography. In this approach, two systems, EAPS (External Attack Prevention System) and IADCS (Internal Attack Detection and Correction System) were introduced. EAPS works under the assumption of having mutual trust among network nodes while IADC runs by having the mutual suspicion between the network nodes. Every route request message carries its own digest encrypted with the sender's private key hash result in order to ensure its integrity. To validate established routes, route replies are authenticated between two neighbors along them. This approach, using the Onion Routing approach and trust management system to provide trust and anonymity for the path discovery (and hence for subsequent communications using this path), prevents external attacks.

## 5. CONCLUSION

Notably, some of these anonymity loss reduction configurations and techniques are mostly hardware-based. Some more, such as the multicast strategy, are mainly software-based, and yet others, such as the dropping ping request, are mostly implemented administratively. The design considerations of these techniques will, at a minimum, certainly assure the reduction of anonymity losses. Given that the signature or pattern of these attacks is not clearly known or constant, the incorporation of any combination of the proposed techniques will, no doubt, preclude the success of existing patterns of attack.

The variety of the proposed techniques spans network communications (whether single-cast or multicast). Although, there is a detailed exploration of Tor systems for their popularity in security assurance, the techniques and methods are equally applicable to other systems. Expectedly, if and where a security breach occurs, it is immediately detected and corrected using the two-pronged approach to attack detection and correction described in the paper

## 6. References


[1] Back, A., Moeller, U., and Stiglic, A. Traffic analysis attacks and trade-offs in anonymity providing systems. In Proc. Information Hiding Workshop (IH 2001) (April 2001), LNCS 2137, pp. 245–257

[2] Boukerche, A., El-Khatib, K., Xu, L., and Korba, L. A Novel Solution for Achieving Anonymity in Wireless Ad Hoc Networks. National Research Council of Canada and Institute for Information Technology. ACM PE-WASUN'2004, held in conjunction with the 7th ACM International Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems. Venice, Italy. October 4-6, 2004. NRC 47402.

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

[4] Chun, B., Culler, D., Roscoe, T., Bavier, A., Peterson, L., Wawrzoniak, M., and Bowman, M. Planetlab: an overlay testbed for broad-coverage services. SIGCOMM Comput. Commun. Rev. 33, 3 (2003), 3–12.

[5] Claessens, J. Preneel, B. and Vandewalle, J. Solutions for Anonymous Communication on the Internet. In *Proceedings of the International Carnahan Conference on Security Technology*, pages 298.303. IEEE, 1999.

[6] Danezis, G., Dingledine, R., and Mathewson, N. Mixminion: Design of a Type III Anonymous Remailer Protocol. In SP '03: Proc. 2003 IEEE Symposium on Security and Privacy (Washington, DC, USA, 2003), IEEE Computer Society, p. 2.

[7] Dingledine, R., Mathewson, N., and Syverson, P. F. Tor: The second-generation onion router. In Proc. 13th USENIX Security Symposium (August 2004).

[8] Federrath, H., et al. JAP: Java anonymous proxy. http://anon.inf.tu-dresden.de/.

[9] Gil, T. M., Kaashoek, F., Li, J., Morris, R., and Stribling, J. The "King" data set. http://pdos.csail.mit.edu/p2psim/kingdata/, 2005.

[10] Goldberg, I., and Shostack, A. Freedom network 1.0 architecture, November 1999.

[11] Guan, F, Fu, X. Bettati, R. and Zhoa, M. An Optimal Strategy for Anonymous Communication Protocols. In *Proceedings of 22nd International Conference on Distributed Computing Systems*, pages 257.266. IEEE, 2002.

[12] Hopper, N., Vasserman, E. Y., Chan-Tin, E. How Much Anonymity does Network Latency Leak? ***Communications*** of the ***ACM***, v.24 n.2 (2007), p.84-90.

[13] Moeller, U., Cottrell, L., Palfrader, P., and Sassaman, L. IETF draft: Mixmaster protocol version 2. http://www.ietf.org/internet-drafts/draft-sassaman-mixmaster-03.txt, 2005.

[14] Murdoch, S. J., and Danezis, G. Low-Cost Traffic Analysis of Tor. In SP '05: Proc. 2005 IEEE Symposium on Security and Privacy (Washington, DC, USA, 2005), IEEE Computer Society, pp. 183–195.

[15] Pfitzmann, A, and Waidner, M. Networks without User Observability. *Computers & Security*, 2(6):158.166, 1987.

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

[17] Serjantov, A., and Sewell, P. Passive attack analysis for connection-based anonymity systems. In Proc. ESORICS 2003 (October 2003).

[18] Syverson, P. F., Goldschlag, D. M., and Reed, M. G. Anonymous connections and onion routing. In *Proceedings of the IEEE Symposium on Security and Privacy* (Oakland, California, May1997), 44–54.

[19] Syverson, P., Tsudik, G., Reed, M., and Landwehr, C. Towards an analysis of onion routing security. In Designing Privacy Enhancing Technologies: Proc. Workshop on Design Issues in Anonymity and Unobservability (July 2000), H. Federrath, Ed., Springer-Verlag, LNCS 2009, pp. 96–114.

[20] TOR (the onion router) servers. http://proxy.org/tor.shtml, 2007.

[21] Venkatraman, L., Agrawal, D.P. Strategies for enhancing routing security in protocols for mobile ad hoc networks, in *Journal of Parallel and Distributed Computing, 63.2* (February 2003), Special issue on Routing in mobile and wireless ad hoc networks, Pages: 214 – 227, Year of Publication: 2003, ISSN:0743-7315


**Article received: 2009-01-03**