IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received October 1, 2018, accepted November 11, 2018, date of publication November 16, 2018,
date of current version December 18, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2881719

Deciding Your Own Anonymity: User-Oriented Node Selection in I2P

LIN YE[1], XIANGZHAN YU[1], JUNDA ZHAO[1], DONGYANG ZHAN[1], XIAOJIANG DU[2], AND MOHSEN GUIZANI[3]

[1]School of Computer Science and Technology, Harbin Institute of Technology, Harbin 150006, China
[2]Department of Computer and Information Sciences, Temple University, Philadelphia, PA 19122, USA
[3]Department of Computer Science and Engineering, Qatar University, Doha, Qatar

Corresponding author: Lin Ye (hityelin@hit.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61771166 and in part by the National Key Research & Development Plan of China under Grant 2016QY05X1000.

ABSTRACT With the development of Internet applications, anonymous communication technology plays a very significant role in protecting personal privacy. As one of the most popular anonymous communication systems, I2P provides strong anonymity through its encryption and communication schemes. However, I2P does not consider the users' preferences, which is difficult to meet the individual demands of specific users and then allows them to decide their anonymity. Thus, this paper proposes two novel user-oriented node selection algorithms that can effectively enhance the anonymity or reduce the communication delay over the I2P network. In order to choose proper nodes, we also investigate key factors to evaluate the nodes. Then, the basic node selection algorithm (BNSA) is proposed to group routing nodes and provide high-performance node candidates. Based on BNSA, the geographic-diversity-oriented node selection algorithm (GDNSA) and the communication-delay-oriented node selection algorithm (CDNSA) are proposed. These can improve the anonymity or communication performance of the I2P network. The GDNSA increases the attack difficulty by establishing tunnels that span multiple regions. In the meantime, the CDNSA reduces the communication delay of the tunnel by selecting the next hop node with the lowest communication delay. Finally, the mathematical analysis and experimental results show that the GDNSA has good resistance to collusion attacks, and the CDNSA reduces the communication delay in spite of weakening a little anonymity.

INDEX TERMS Anonymous communication, node selection, geographic diversity, communication delay, I2P.

I. INTRODUCTION

While the Internet facilitates our lives, it also brings serious concerns to personal privacy when the traffic goes across the network, such as web browsing, email and instant messaging. Attackers can sniffer the packets to identify communication relationships, which threatens personal privacy. Therefore, anonymous communication technology has become an ideal way to protect the communication security. The purpose of the anonymous communication is to hide confidential information of each end user, including the identity as well as the content, and avoid being observed and discovered by third parties. However, regarding anonymous communication systems, anonymity and communication efficiency have always been a trade-off. In general, the better anonymity the system has, the worse the communication efficiency will be. That is because anonymity enhancement is often accompanied by the complexity of the communication process, which will also result in a decrease in communication efficiency. Therefore, how to meet the individual demands of different users has become a hot topic in anonymous communication systems.

As one of the most popular anonymous communication systems, I2P provides strong anonymity through its complex encryption and communication schemes. However, I2P does not consider the users' preferences, which is difficult to meet individual demands of specific users and then allow them to decide their anonymity. In this paper, we propose two user-oriented node selection algorithms that can optimize the existing I2P. First, the Basic Node Selection Algorithm (BNSA) is proposed to classify tunnel nodes, which does not trust performance information provided by other nodes. BNSA evaluates all nodes through active measurement and obtains the high-performance ones. Second, to achieve better anonymity, the Geographic-diversity-oriented Node Selection Algorithm (GDNSA) tries to determine high-performance tunnel nodes from as many regions as possible. Since it is difficult for attackers to deploy a

large number of high-performance nodes in multiple regions, GDNSA has good resistance to collusion attacks. Third, the Communication-delay-oriented Node Selection Algorithm (CDNSA) is proposed to get better communication efficiency, which chooses the node with the lowest delay each time as the next tunnel node and reduces the communication delay of the entire tunnel.

The contributions of this paper are as follows:

1) A basic node selection algorithm is proposed to classify tunnel nodes, which can evaluate the nodes through active measurement and provide high-performance node candidates.
2) To enhance the anonymity, we propose a node selection algorithm based on geographic diversity to determine high-performance tunnel nodes from as many regions as possible, which increases the difficulty of collusion attacks.
3) To achieve better communication efficiency, we propose a node selection algorithm based on communication delay to determine the node with the lowest delay at each hop. This can reduce the overall communication delay of the tunnel.

The rest of this paper is organized as follows: Section II gives the background and summarizes the related work. Section III describes the proposed node selection algorithms. Section IV evaluates the effectiveness of our solution. Conclusions and future work are presented in Section V.

II. BACKGROUND & RELATED WORK

A. I2P NETWORK

The existing anonymous networks mainly built on the rerouting scheme, such as Tor [1], Tarzan [2], I2P [3], Crowds [4], etc., where the packets go through multiple anonymous nodes over the network. Each node modifies, fills, and forwards data packets, so that the messages pass through several nodes from the sender to the receiver, thereby realizing the protection of the identities and communication relationship.

I2P [3] is a low-latency anonymous communication network based on P2P networks and key management technologies [5]–[7], which originates from the Invisible Internet Project (IIP) [8]. I2P can integrate a wide range of applications, such as anonymous web hosting, web browsing, file sharing, and email. Meanwhile, the concerns on I2P also arise all over the world, especially for governments, because it can provide anonymity and prevent traffic analysis by encrypting and distributing communications. This has become a part of dark web where many illegal activities spring up.

I2P has four notable features, including garlic routing, address book, tunnel and network database. Garlic routing [9] is a variant of onion routing technology. Compared with the onion routing, the main difference is that garlic routing can encapsulate multiple messages in an encrypted data packet. The encrypted data packet is called ''garlic'', and the inside message is called ''clove''. The address book functionally refers to a mapping similar to DNS, which bridges the domain name of an anonymous application and the identity of the application provider. When each user joins the system, he will generate a 512-byte base64-encoded key called *destination*, which is the unique identifier for the user. If a user A provides an anonymous application, a domain name with the format xxx.i2p will be generated. Then, the record a.i2p=destination_A will be added to the address book, where a.i2p is the domain name and destination_A is the user A's *destination*. When other users want to use the anonymous application provided by user A, they can access a.i2p directly in the browser.

A tunnel is a unidirectional rerouting path of the I2P network. Accordingly, the tunnels can be divided into inbound and outbound ones depending on the transmission direction. As shown in Figure 1, an inbound tunnel is responsible for receiving messages while an outbound tunnel is responsible for sending messages. In one tunnel, the first hop node is called the gateway, and the last hop node is called the endpoint. Other nodes are called the participant nodes. Each tunnel is valid for 10 minutes. When a tunnel expires, the routing node needs to re-create a new tunnel. The length of a tunnel is determined by two non-negative numbers x and y specified by a user. The system generates a random number $r \in [-y, y]$ and the length of the tunnel is $max(x + r, 0)$.

Network Database, also known as NetDB, is built on top of Kademlia [10], which stores two kinds of information: RouterInfos and LeaseSets. The RouterInfos contains key information required for the communication between routing nodes, including the public key and the address of the routing node. The LeaseSets consists of multiple lease information where each lease information contains gateway information to a destination, that is, the gateway information of an anonymous user's inbound tunnel, including the address of inbound tunnel gateway, tunnel expiration time, and Tunnel ID, etc.

To anonymize messages, every node has a I2P router, which has many inbound and outbound tunnels. A message will go from one node's outbound tunnel to another node's inbound tunnel, and finally arrive at the destination. When A wants to communicate with B, A needs to request B's lease set from Network Database. After obtaining B's lease information, A sends the messages through its own outbound tunnel to the gateway of B's inbound tunnel, which transmits the messages along the inbound tunnel to B. Then, B can send back the response messages to A in a similar manner.

B. RELATED WORK

There are many existing efforts to enhance the efficiency and security of anonymous networks. The work in [11] summarizes the overall structure and core technologies of Tor and I2P, and compares their node selection strategies, performance and scalability. The study in [12] also presents a detailed and comprehensive comparison between Tor and I2P. HORNET [13] is a low-latency onion routing system that builds the high-speed end-to-end anonymous channels by leveraging next-generation network architectures. Using the random walk theory and crypto-types, the work

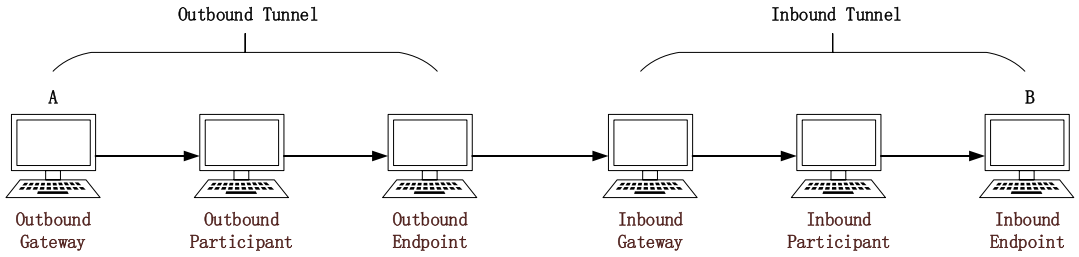


FIGURE 1. Outbound and inbound tunnels.

in [14] achieves multimodal behaviors to enhance the privacy of anonymous networks. The study in [15] implements a large-scale monitoring architecture for I2P networks, which deploys monitoring nodes in the I2P network database to collect query requests. The analysis results show that most of the I2P communication traffic comes from file sharing, followed by web browsing. It also evaluates the geographic location distribution and network activity of nodes, which proves that the group-based node division is feasible. EFTAN [16] is an improved model for Tor to promote the efficiency, security and anonymity, which divides the circuits into two parts to make sure that all nodes cannot know the identity of communication participants. Hydra [17] divides the existing useable nodes into relay groups, and then selects them randomly by using automatic algorithms. The work in [18] leverages a combination of node conditions (bandwidth and uptime of relays) and path condition (delays between the relays) to improve Tor's performance and security. TorPolice [19] performs privacy-preserving access control, which makes abuse-plagued service providers enforce access rules to enable the global access control for relays. The study in [20] evaluates I2P's design choices against performance and security, and compares it with non-anonymous peer-to-peer network. The work in [21] improves the speed of client browsing by selecting the path based on the bandwidth in the consensus file according to the customer's needs. Furthermore, Sybilhunter [22] has developed to detect Sybil relays in Tor based on their appearance and behavior. The study in [23] uses two exit relay scanners to discover either misconfigured or outright malicious relays in Tor. The work in [24] gives an analysis of family misconfiguration in Tor networks as well as the corresponding methods that discover and correct them.

On the other hand, many existing attacks can threaten anonymous communication systems [25]. Attackers can perform external attacks by monitoring from outside, and also leverage some internal nodes to attack anonymous networks. In anonymous communication systems based on rerouting, collusion attack is one of the most common attacks. The attackers, as well as malicious nodes, exchange relevant communication information by controlling multiple nodes on the rerouting path to infer a user's identity. The work in [4] proves that when the proportion of malicious nodes exceeds a certain threshold, the anonymity will be broken. The study in [26] introduces an attack that can determine the identity of an anonymous web service provider (Eepsite) in the I2P network. The attacker first estimates the set of nodes that Eepsite creates the tunnel, and then performs a DoS attack on these nodes. Thus, the malicious nodes controlled by the attacker replace these nodes to participate in the Eepsite tunnel creation, and finally the attacker can make some simple measurements to determine the identity of Eepsite. A DoS attack is also used to reduce the performance of normal nodes [27], thereby increasing the probability that the malicious nodes controlled by the attacker are selected to determine the identity of the anonymous service provider. Besides, the attacker can determine the identity of the sender and the receiver by observing the synchronization pattern of communication messages. The longer the attacker observes the synchronous communication, the more likely the communication nodes are to be associated. Flow marking attacks [28] control the ingress and egress nodes in the anonymous communication system, which can recognize the communication relationship between hosts by tracking such artificial marks. Our paper is also related to other security problems [29]–[31].

III. USER-ORIENTED NODE SELECTION ALGORITHMS

Essentially, the anonymity and communication delay mainly depend on the path length and node selection of the rerouting. Node selection is the process of determining which routing nodes in the network to deliver messages. The node selection algorithms provide standards and rules for relay nodes selection in the rerouting path. Different node selection algorithms will result in the differences of anonymity and communication delay.

In this section we first present the Basic Node Selection Algorithm (BNSA), which is based on the I2P original node selection algorithm. Based on BNSA, two node selection algorithms are proposed to enhance the anonymity or reduce the communication delay respectively, i.e., Geographic-diversity-oriented Node Selection Algorithm (GDNSA) and Communication-delay-oriented Node Selection Algorithm (CDNSA). Finally, we give the theoretical analysis on anonymous security of GDNSA and CDNSA.

A. BASIC NODE SELECTION ALGORITHM

The tunnel node selection algorithm of the I2P network works in source route mode, that is, the sender is responsible for selecting tunnel nodes and creating a tunnel. Unlike other

anonymous communication networks, routing nodes in I2P do not trust performance information provided by other routing nodes, because 'lazy' nodes may claim lower performance for free-riding or malicious ones may claim higher performance to increase the probability of joining a target tunnel deliberately. To this end, the routing nodes in I2P actively measure the performance of other routing nodes, such as bandwidth, tunnel creation success rate, workload and reachability. Meanwhile node evaluation continuously updates the status in real time, including the time how long a routing node responds to the query, the number of tunnel failures that a routing node participates, and the last communication time of a routing node.

In this paper, we still refer to a tunnel as the rerouting path, and tunnel nodes are selected according to node evaluation. However, the existing node evaluation mainly depends on two factors: capacity and speed. Therefore, in order to represent the comprehensive performance of routing nodes, we introduce several new evaluation factors to propose our BNSA.

1) NODE EVALUATION FACTORS

The node evaluation factors in BNSA include capacity, bandwidth, online time, reachability and delay.

a: CAPACITY

Capacity refers to the number of successful tunnels established with a routing node over a period of time proposed in [3]. Due to the operational overhead of a tunnel creation, it is necessary to evaluate routing nodes according to the willingness of responding to tunnel requests. Since there is limitation on bandwidth, CPU usage and the number of participating tunnels, routing nodes sometimes may reject or drop tunnel requests.

The capacity calculation is to estimate the number of tunnels that a routing node agrees to participate in the next hour through historical statistical information. The weight of historical statistical information decreases over time. The time interval for the evaluation is 10 minutes, 30 minutes, 1 hour and 1 day. The evaluation equation proposed in [3] is:

$$R = 4 \times r(10m) + 3 \times r(30m) + 2 \times r(1h) + r(1d) \quad (1)$$

where R is the weighted evaluation, and $r(t)$ represents the statistical information of the routing node participating in the tunnel at the most recent t time.

Since the overhead of tunnel creation is high, it is reasonable that the number of rejectors, non-respondors and failures should be considered. Therefore, the final formula for calculating $r(t)$ is:

$$r(t) = accepts - rejects - timeouts - 4 \times failures \quad (2)$$

where *accepts* represents the times that a routing node agrees to the participation of the tunnels, *rejects* represents the times that a routing node refuses to participate in the tunnels, *timeouts* represents the times that a routing node does not respond to participate in the tunnels, and *failures* represents the times that a routing node agrees to participate in the tunnels but tunnel testing has failed.

b: BANDWIDTH

Bandwidth is defined as a weighted result of a node's speeds in different time periods. A routing node counts the sent and received bytes in a tunnel created by itself in one minute as the speed of each participating node in the tunnel, and a tunnel node bandwidth in one minute is the average speed of all tunnels that it participates in, which is the basis of bandwidth estimator proposed in BNSA. The calculation formula is:

$$B = 4 \times b(1min) + 3 \times b(10min) + 2 \times b(1h) + b(3h) \quad (3)$$

where B represents the weighted bandwidth, and $b(t)$ represents the average of three maximum bandwidths in the most recent t time.

c: ONLINE TIME

Online time refers to the time how long a node is online. We believe that the longer the node is online, the more likely it is to remain online. If a node is offline during a test, the online time will be reset to 0. The calculation formula is:

$$T = ot/rt \quad (4)$$

where T is the score of the online time, ot is a node's online time, and rt is the system execution time.

d: REACHABILITY AND DELAY

Delay is measured every 10 minutes to determine whether the measured nodes are online or not. Obviously, the lower the delay is, the higher the node scores. Reachability is the requirement of a node selection, which means the connected routing nodes have the qualification to be selected in the process of establishing a tunnel. Delay will serve as a key reference for CDNSA.

2) BNSA DESIGN

BNSA divides the routing nodes into three groups, namely, Reachable Group (RG), High-Reliable Group (HRG), and High-Performance Group (HPG). HPG is a subset of HRG, and HRG is a subset of RG. In BNSA, all of the routing nodes are divided into different groups in each polling based on the evaluation factors such as capacity, bandwidth, online time. Specifically, we first initiate these three groups and then test the reachability of these nodes. All of the reachable nodes are added into the RG. After that, the reliability of each node in the RG is scored according to its capacity and online time. The nodes with high scores beyond the average score, are inserted into the HRG. In the HRG, every node's performance score is calculated according to its bandwidth, and the nodes with high scores beyond the average score are put into the HPG. Finally, HPG will be the basis of the GDNSA and CDNSA.

Algorithm 1 Basic Node Selection Algorithm

Input: AN
Output: HPG

```
RG = { }, HRG = { }, HPG = { }
for node in AN do
    if node is reachable then
        RG.add(node)
for node in RG do
    Rely[node] = R + k × T
aveRelay = average(Rely[all_nodes])
for node in RG do
    if Relay[node] > aveRelay then
        HRG.add(node)
if HRG.length < MinRely then
    insert MinRely - HRG.length more nodes into HRG
if HRG.length > MaxRely then
    pop HRG.length - MaxRely nodes from HRG
for node in HRG do
    Performance[node] = B[node]
avePerformance = average(Performance[all_nodes])
for node in HRG do
    if Performance[node] > avePerformance then
        HPG.add(node)
if HPG.length < MinPerf then
    insert MinPerf - HPG.length more nodes into HPG
if HPG.length > MaxPerf then
    pop HPG.length - MaxPerf nodes from HPG
return HPG
```

As shown in Algorithm 1, the input is the set of all known routing nodes AN, and the output is the HPG. BNSA has three steps:

Step 1. Test the reachability of all routing nodes in the AN and add the reachable ones into the RG.

Step 2. Calculate the reliability score of each routing node in the RG. The reliability score is a comprehensive score of capacity and online time. The reliability of a routing node is calculated as follows:

$$Rely = R + k \times T \tag{5}$$

where *Rely* represents a reliability score, R represents the capacity score, T represents an online time score, and k is a constant coefficient to adjust online time score, which is between 0 and 10. By default, $k = 1$. Then, we calculate the average reliability score *AveRely* in the RG as a threshold, beyond which the routing nodes with the reliability scores can be inserted into the HRG. We also use *MinRely* and *MaxRely* to indicate the lower and upper bounds of the number of HRG nodes respectively. If the number of routing nodes in the HRG is less than *MinRely*, more routing nodes with descending reliability scores will be selected into the HRG. If the number of routing nodes in the HRG is greater than *MaxRely*, only the *MaxRely* highest reliability routing nodes will be selected into the HRG.

Step 3. Calculate the performance score of each routing node in the HRG, that is, the bandwidth score, and obtain the average performance score *AvePerformance*. Then, the routing nodes in the HRG whose performance scores are beyond *AvePerformance* will be added into the HPG. Similarly, we also use *MinPerf* and *MaxPerf* to represent the lower and upper bounds of the number of HPG nodes respectively. If the number of routing nodes in the HPG is less than *MinPerf*, more routing nodes with descending performance scores will be selected into the HPG. If the number of routing nodes in the HPG is greater than *MaxPerf*, only the *MaxPerf* best performance routing nodes will be selected into the HPG.

B. GEOGRAPHIC-DIVERSITY-ORIENTED NODE SELECTION ALGORITHM

We refer to the routing nodes controlled by the attackers as the malicious ones. The malicious nodes not only could refuse to serve, but also can collect user information by conspiring with other malicious ones. If a tunnel has many malicious nodes, it is difficult to guarantee communication anonymity and system performance. In order to fight against such an attack on I2P, some approach increases the length of a tunnel to reduce the proportion of malicious nodes in the tunnel. However, a long tunnel will cause a significant increase in communication delay, because the encryption and decryption of data packets require high overhead. Some method tries to increase the size of the anonymous network to reduce the proportion of malicious nodes. But it requires a large number of trusted nodes, which is difficult to implement. Because there are more routing nodes in developed countries, such as Europe and the United States, it is likely that multiple nodes in the tunnel are located in the same region using a traditional node selection algorithm. The attackers can easily deploy their high-performance malicious nodes without a high cost to crack communication anonymous. To this end, we propose the GDNSA in this paper, which can increase attack cost by selecting routing nodes in different regions to create a tunnel.

GDNSA is an optimized node selection algorithm based on BNSA. As shown in Algorithm 2, the input is the HPG and tunnel length L. The output is routing node queue TN, containing $L-1$ routing nodes that participate in the tunnel. GDNSA first initializes the TN to an empty queue and divides the routing nodes in the HPG into n groups according to different regions. Then, it randomly arranges the n groups to obtain the regions queue AQ=<A_1, A_2,..., A_n>. After that, it performs $L-1$ routing node selection. In the k*th* selection, the first element A_i in the AQ is selected. If A_i is not empty, a routing node N_k is randomly selected from A_i to append to the TN tail. Then, N_k is removed from A_i and A_i is moved to the tail of AQ. At this time, AQ = <A_{i+1},..., A_n, A_1,..., A_i>, TN=<N_1, N_2, ..., N_k>, where $1 \le k \le L-1$, $i = k\%n$. Finally, the queue of $L-1$ routing nodes TN=<N_1, N_2, ..., N_{L-1}> can be generated.

Algorithm 2 Geographic-Diversity-Oriented Node Selection Algorithm

Input: HPG, L
Output: TN

```
TN = { }, AQ = { }, tail = 0
Routing nodes in HPG are divided into n groups
Randomly sort n groups and store them in AQ
for i = 1 to L-1 do
    A_i = AQ[0]
    if A_i is not empty then
        N_k = randomly_select_node(A_i)
        TN[tail] = N_k
        tail++
        A_i.remove(N_k)
        AQ.remove(A_i)
        AQ.append(A_i)
return TN
```

C. COMMUNICATION-DELAY-ORIENTED NODE SELECTION ALGORITHM

Sometimes the users expect anonymous systems to provide lower communication delay. For example, in instant messaging or web browsing, if the communication delay is too high, it will seriously affect user experience. The traditional node selection algorithm improves transmission efficiency by selecting routing nodes with higher bandwidth, but the increase in bandwidth does not mean a reduction in communication delay. For example, the routing nodes in tunnel A have better bandwidth, but the delay of each hop in tunnel A is higher. The bandwidth of the routing nodes in tunnel B is lower than that of tunnel A, but the delay of each hop in tunnel B is lower. If the communication data is small, it is obvious that tunnel B has a lower communication delay than tunnel A. Therefore, we propose the CDNSA, which achieves lower communication delay by reducing the delay of each hop in the tunnel.

CDNSA is illustrated in Algorithm 3. The input is the HPG and tunnel length L. The output is the routing node queue TN formed by $L-1$ routing nodes. First, we initialize TN to an empty queue and select the lowest delay $L-1$ nodes in the HPG to form LDS_0. Then, we randomly select a routing node N_1 and add it into TN. At this time, TN = <N_1>. After that, N_1 also uses $L-1$ nodes with the lowest delay in its HPG to form LDS_1, and randomly adds an unused routing node N_2 from the LDS_1 to TN. At this time, TN=<N_1, N_2>. CDNSA iterates continuously until TN=<$N_1, N_2, \ldots, N_{L-1}$> is generated.

D. ANONYMITY SECURITY ANALYSIS

1) SECURITY ANALYSIS OF GDNSA ANONYMITY

In this section, we mainly analyze the security of GDNSA for collusion attacks. The system anonymity will be seriously threatened if the majority or all of the routing nodes are malicious ones.

Assume that the total number of HPG is N, where the total number of malicious nodes is M, satisfying $M \leq N$.

Algorithm 3 Communication-Delay-Oriented Node Selection Algorithm

Input: HPG, L
Output: TN

```
TN = { }, tail = 0
Select L − 1 nodes with the lowest delay in the HPG to form LDS_0
N_1 = randomly_select_node(LDS_0)
TN[tail] = N_1
tail++
for i = 1 to L-2 do
    Select L − 1 nodes with the lowest delay from N_i' HPG to form LDS_i
    N_{i+1} = randomly_select_node(LDS_i)
    TN[tail] = N_{i+1}
    tail++
return TN
```

The number of regions is G, and the length of the tunnel is L, which satisfies $G < N$ and $L \geq 2$. It is assumed that N routing nodes are evenly distributed among G regions, that is, each region contains N/G routing nodes. A_0 indicates the probability that the anonymity is cracked where the tunnel is constructed by random node selection algorithm without region division:

$$A_0 = \frac{M}{N} \times \frac{M-1}{N-1} \times \ldots \times \frac{M-L+2}{N-L+2} = \prod_{k=0}^{L-2} \frac{M-k}{N-k} \quad (6)$$

A_1 indicates the probability that the anonymity is cracked where the tunnel is constructed by GDNSA when M malicious nodes are evenly distributed among G regions:

$$A_1 = \begin{cases} \left(\frac{\frac{M}{G} - \lfloor \frac{L}{G} \rfloor}{\frac{N}{G} - \lfloor \frac{L}{G} \rfloor}\right)^{L\%G} \times \prod_{k=0}^{\lfloor \frac{L}{G} \rfloor - 1} \left(\frac{\frac{M}{G}-k}{\frac{N}{G}-k}\right)^{G}, & L > G \\ \left(\frac{M}{N}\right)^{L-1}, & L \leq G \end{cases} \quad (7)$$

$A_i (i > 1)$ indicates the probability that the anonymity is cracked where the tunnel is constructed by GDNSA when M malicious nodes are evenly distributed in $\frac{G}{i}$ regions:

$$A_i = \begin{cases} \prod_{k=0}^{L-2} \frac{M \times (G - i \times k)}{N \times (G-k)}, & L \leq \frac{G}{i} \\ 0, & L > \frac{G}{i} \end{cases} \quad (8)$$

We assume that in HPG the number of routing nodes N = 10000, the number of regions G = 20, the numbers of malicious nodes M are 2000, 4000, 6000, 8000 respectively. When the tunnel length L is 3, 5, 7, the anonymity crack probabilities A_i are shown in Table 1.

Figure 2 compares the anonymity of the tunnels constructed by random node selection algorithm and GDNSA respectively when malicious nodes are distributed randomly in G and $G/2$ regions, which indicates:

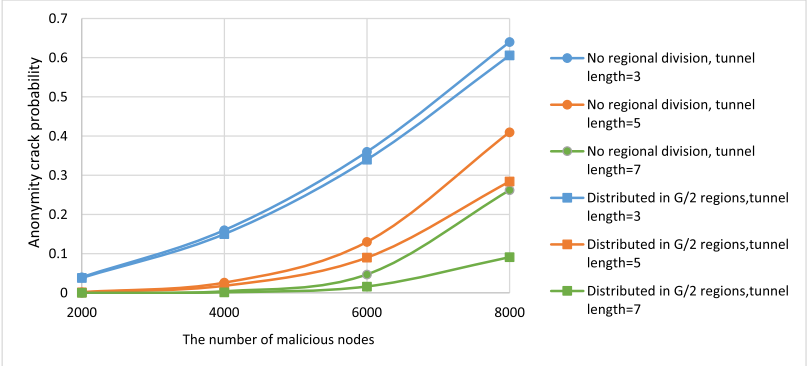


FIGURE 2. Random selection algorithm VS. GDNSA.

TABLE 1. Anonymity crack probability.

	L \ M	2000	4000	6000	8000
A_0	3	0.03998	0.16	0.36	0.64
	5	0.0016	0.026	0.13	0.41
	7	0.00006	0.004	0.047	0.262
A_1	3	0.04	0.16	0.36	0.64
	5	0.0016	0.0256	0.1296	0.4096
	7	0.00006	0.0041	0.04656	0.2621
A_2	3	0.038	0.15	0.34	0.606
	5	0.0011	0.018	0.0899	0.284
	7	0.00002	0.0014	0.0162	0.091
A_3	3	0.036	0.143	0.322	0.573
	5	0.0007	0.0115	0.0583	0.184
	7	0	0	0	0
A_4	3	0.0336	0.135	0.303	0.539
	5	0.00042	0.0068	0.034	0.108
	7	0	0	0	0

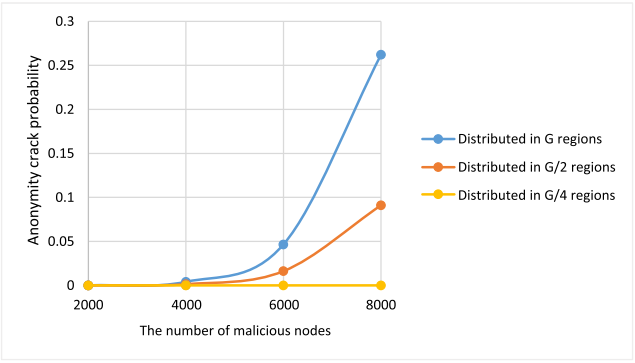


FIGURE 3. GDNSA with different malicious node distributions.

- The higher the proportion of malicious nodes is, the higher the probability of anonymity is cracked.
- The longer the length of the tunnel is, the lower the probability of anonymity is cracked.
- With the same length, the tunnel constructed by GDNSA is more secure than the tunnel constructed by random node selection algorithm.

Figure 3 shows a comparison of the anonymous security in a tunnel with a length 7 constructed by GDNSA with different distributions of malicious nodes. It can conclude that the fewer regions malicious nodes are distributed, the smaller the probability of the anonymity is cracked, and the higher the anonymity security will be. Because it is difficult for an attacker to deploy malicious nodes in multiple locations at the same time, GDNSA increases the difficulty of collusion attacks and enhances the security of the system.

2) SECURITY ANALYSIS OF CDNSA ANONYMITY

According to CDNSA, the first-hop node is selected from its own HPG. From the second hop, the node selection is determined by the participants in the tunnel. If the first node is a malicious one, it can control the entire tunnel by providing another malicious node. Therefore, CDNSA anonymity mainly relies on node selection in the first hop.

Suppose that there are N routing nodes in the HPG where the number of malicious nodes is M and tunnel length is L. The probability H_i that the tunnel selects the $L-1$ lowest delay nodes including i malicious nodes in the HPG is:

$$H_i = \frac{\binom{i}{M} \times \binom{L-i-1}{N-M}}{\binom{L-1}{N}} \tag{9}$$

The probability of first hop is a malicious node:

$$P = \frac{M}{N} \tag{10}$$

Figure 4 illustrates the comparison between CDNSA and random node selection algorithm, which implies:

- Compared with random node selection algorithm, CDNSA is more likely to be cracked with worse anonymity.
- CDNSA anonymity is merely related to the proportion of malicious nodes, regardless of tunnel length. The higher the proportion of malicious nodes is, the higher probability the first hop will be a malicious node.

It can conclude that CDNSA reduces communication delay with a certain loss of anonymity, so CDNSA is suitable for the users who care about communication delay more than anonymity.

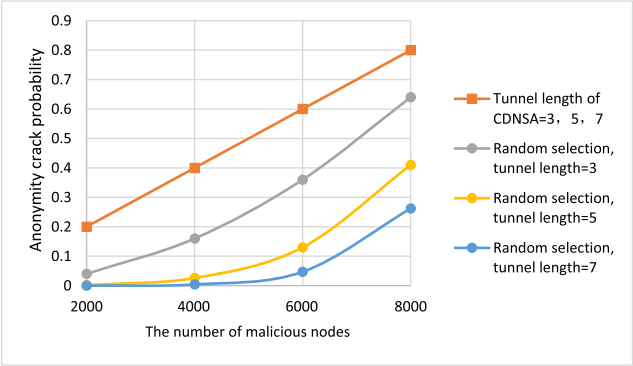


FIGURE 4. Random selection algorithm VS. CDNSA.

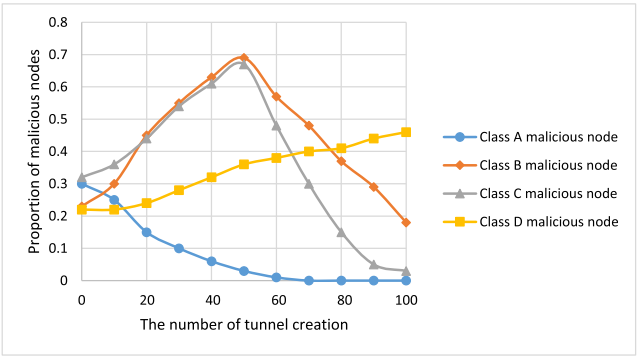


FIGURE 5. Comparison of different types of malicious nodes.

IV. EVALUATION

We use NS-3 to simulate an anonymous communication system. The nodes in different regions are tagged by their region number. If the difference between two region numbers is 1, it is assumed that these two regions are adjacent. The delay between two regions is proportional to the difference between two region numbers. For example, the communication delay between two regions a and b is $T \times |a - b|$ where T is a positive integer constant, indicating the basic communication delay between adjacent regions. Communication delay between the nodes in the same area T_{is} is a random positive integer between 1 and T, so $T_{is} = Random(1, T)$. The communication delay T_{os} of the nodes in different regions is the sum of the communication delay between two regions and the intra-region communication delay, which formula is $T_{os} = T \times |a - b| + T_{is}$.

A. BNSA EFFECTIVENESS

In order to evaluate BNSA's effectiveness, we investigate the proportion of malicious nodes in the HPG selected by BNSA. In the test, the number of routing nodes in the system is 1000, the number of malicious nodes is 200, the upper and lower bounds of the number of the nodes in the HRG are 300 and 200 respectively, and the upper and lower bounds of the number of the nodes in the HPG are 200 and 100 respectively. According to the behaviors of malicious nodes, we divide malicious nodes into four classes:

- Class A: malicious nodes directly discard tunnel creation requests.
- Class B: malicious nodes agree to the first 50 tunnel requests, but discard the next 50 tunnel requests.
- Class C: malicious nodes agree to all tunnel requests and normally serve at the first 50 times, but they discard the data of the sender at the next 50 times.
- Class D: malicious nodes behave the same as normal nodes.

Besides, the bandwidth of a malicious node is two times higher than that of a normal node. The tunnel creator selects nodes randomly in the HPG to create a tunnel.

We evaluate these four types of malicious nodes separately, and the results are shown in Figure 5, which indicates:

- Because Class A malicious nodes always refuse the requests of tunnel creation, their capacity scores are lower, and the proportion of malicious nodes in the HPG gradually decreases.
- Since Class B malicious nodes agree to the tunnel requests and their bandwidth performances are high during the first 50 tunnel creation processes, the proportion of malicious nodes in the HPG increases at the beginning. However, since they refuse to participate in other tunnels after the first 50 tunnels, the capacity scores gradually decrease. Therefore, it becomes more difficult to join the HPG, and the proportion in the HPG will decrease finally.
- Class C malicious nodes are similar to Class B. The first 50 tunnels make the proportion of malicious nodes increase, but after that, the proportion decreases. Because the penalty for tunnel testing failure is higher than the rejection, the proportion of Class C malicious nodes in the HPG decreases significantly faster than that of Class B malicious nodes.
- Class D malicious nodes behave the same way as normal nodes, and their bandwidth is higher than normal ones. With the increase of the tunnels to be created, the proportion of Class D malicious nodes in the HPG continues to increase.

It is concluded that BNSA has good resilience against Class A/B/C malicious nodes, but it cannot handle the Class D malicious nodes, which can break system anonymity through collusion attacks. According to the analysis in the previous section, GDNSA can fight against collusion attacks because it is difficult and costly for an attacker to deploy malicious nodes in multiple regions.

B. COMMUNICATION DELAY

In order to evaluate the communication delay, we investigate the one-way communication delay with GDNSA, CDNSA and random node selection algorithm respectively with different region divisions. In the experiment, the number of routing nodes is 1000, each node agrees to the tunnel requests with a probability of 50% and the basic delay between adjacent regions is 5 seconds.

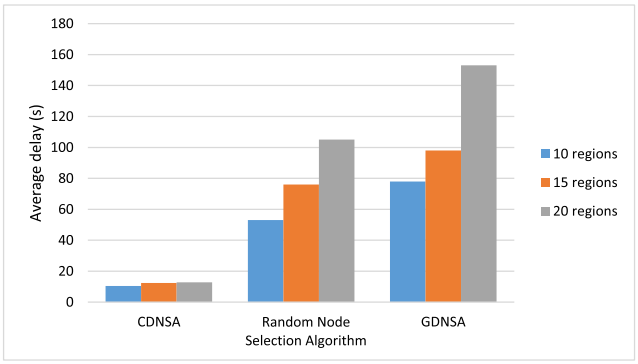


FIGURE 6. Average communication delay of different algorithms.

When the number of region divisions is 10, 15 and 20, the average one-way communication delay for creating 100 tunnels with the length 4 is as shown in Figure 6. Since CDNSA selects the nodes with the lowest delay, the overall delay of the tunnels created by CDNSA is much smaller than other algorithms. Because the nodes selected by GDNSA are located in different regions, these tunnels have the highest one-way communication delay. The more regions are involved in, the higher the average one-way communication delay is. The random node selection algorithm cannot guarantee the lowest communication delay, and there is no need to ensure that the nodes come from different regions. Therefore, its average one-way communication delay of the tunnel is between CDNSA and GDNSA.

V. CONCLUSION

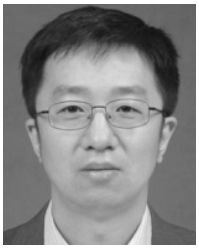
Considering the differentiated requirements of individual users to decide their own anonymity using an anonymous communication system, this paper investigates the node selection algorithms in I2P system. We propose several user-oriented node selection algorithms, i.e., Basic Node Selection Algorithm, Geographic-diversity-oriented Node Selection Algorithm and Communication-delay-oriented Node Selection Algorithm, which enable users to create the corresponding tunnels according to their own preferences. The theoretical analysis and experimental results prove the effectiveness of our node selection algorithms, which can enhance system anonymity or reduce communication delay, respectively. In the future, we will investigate the misbehaviors of I2P nodes that may be compromised. Differentiating them from normal nodes can help provide better node candidates and enhance the anonymity.

REFERENCES

[1] R. Dingledine, N. Mathewson, and P. Syverson, "Tor: The second-generation onion router," in *Proc. Usenix Secur.*, 2004, p. 21.
[2] M. J. Freedman and R. Morris, "Tarzan: A peer-to-peer anonymizing network layer," in *Proc. 9th ACM Conf. Comput. Commun. Secur.*, 2002, pp. 193–206.
[3] L. Schimmer, "Peer profiling and selection in the I2P anonymous network," in *Proc. PetCon*. Dresden, Germany: Technische Univ. Dresden, 2009, pp. 59–70.
[4] M. K. Reiter and A. D. Rubin, "Crowds: Anonymity for Web transactions," *ACM Trans. Inf. Syst. Secur.*, vol. 1, no. 1, pp. 66–92, 1998.
[5] Y. Xiao *et al.*, "A survey of key management schemes in wireless sensor networks," *Comput. Commun.*, vol. 30, nos. 11–12, pp. 2314–2341, Sep. 2007.
[6] X. Du, Y. Xiao, M. Guizani, and H.-H. Chen, "An effective key management scheme for heterogeneous sensor networks," *Ad Hoc Netw.*, vol. 5, no. 1, pp. 24–34, 2007.
[7] X. Du, Y. Xiao, M. Guizani, and H.-H. Chen, "Transactions papers a routing-driven elliptic curve cryptography based key management scheme for heterogeneous sensor networks," *IEEE Trans. Wireless Commun.*, vol. 8, no. 3, pp. 1223–1229, Mar. 2009.
[8] M. Ehlert, "I2p usability vs. Tor usability a bandwidth and latency comparison," Humboldt Univ. Berlin, Berlin, Germany, Seminar Rep., 2011, pp. 129–134. [Online]. Available: http://www.i2pproject.net/en/papers/bibtex#ehlert2011:usability-comparison-i2p-tor and http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.476.4614
[9] R. Dingledine, M. J. Freedman, and D. Molnar, "The free haven project: Distributed anonymous storage service," in *Designing Privacy Enhancing Technologies*. Berlin, Germany: Springer, 2001, pp. 67–95.
[10] P. Maymounkov and D. Mazières, "Kademlia: A peer-to-peer information system based on the XOR metric," in *Proc. Int. Workshop Peer-to-Peer Syst.*. Berlin, Germany: Springer, 2002, pp. 53–65.
[11] B. Conrad and F. Shirazi, "A survey on tor and i2p," in *Proc. 19th Int. Conf. Internet Monit. Protection (ICIMP)*, 2014, pp. 22–28.
[12] A. Ali *et al.*, "TOR vs I2P: A comparative study," in *Proc. IEEE Int. Conf. Ind. Technol.*, Mar. 2016, pp. 1748–1751.
[13] C. Chen, D. E. Asoni, D. Barrera, G. Danezis, and A. Perrig, "HORNET: High-speed onion routing at the network layer," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, 2015, pp. 1441–1454.
[14] M. A. Nia, R. E. Atani, and A. Ruiz-Martínez, "Privacy enhancement in anonymous network channels using multimodality injection," *Secur. Commun. Netw.*, vol. 8, no. 16, pp. 2917–2932, 2015.
[15] J. P. Timpanaro, I. Chrisment, and O. Festor, "A bird's eye view on the i2p anonymous file-sharing environment," in *Proc. Int. Conf. Netw. Syst. Secur.* Berlin, Germany: Springer, 2012, pp. 135–148.
[16] Y. Meng, Y. Liu, J. Fei, and Y. Zhu, "An efficient improved model for tor anonymous network," in *Proc. ICETA*, 2017, pp. 1–8.
[17] M. Wahal and T. Choudhury, "Hydra—Anonymous network routing mechanism," in *Proc. Int. Conf. Technol. Unmanned Syst. (INFOCOM)*, Dec. 2017, pp. 230–235.
[18] S. M. Milajerdi and M. Kharrazi, "A composite-metric based path selection technique for the Tor anonymity network," *J. Syst. Softw.*, vol. 103, pp. 53–61, May 2015.
[19] Z. Liu, Y. Liu, P. Winter, P. Mittal, and Y. C. Hu, "TorPolice: Towards enforcing service-defined access policies for anonymous communication in the Tor network," in *Proc. IEEE Int. Conf. Netw. Protocols*, Oct. 2017, pp. 1–10.
[20] J. P. Timpanaro, T. Cholez, I. Chrisment, and O. Festor, "Evaluation of the anonymous I2P network's design choices against performance and security," in *Proc. Int. Conf. Inf. Syst. Secur. Privacy*, Feb. 2015, pp. 1–10.
[21] K. Kiran, B. Vignesh, P. D. Shenoy, K. R. Venugopal, T. V. Prabhu, and M. S. E. Prasad, "Client requirement based path selection algorithm for Tor network," in *Proc. Int. Conf. Comput., Commun. Netw. Technol.*, Jul. 2017, pp. 1–6.
[22] P. Winter, R. Ensafi, K. Loesing, and N. Feamster, "Identifying and characterizing sybils in the Tor network," in *Proc. 25th Usenix Secur. Symp.*, Aug. 2016, pp. 1–28.
[23] P. Winter *et al.*, "Spoiled onions: Exposing malicious Tor exit relays," in *Proc. 14th Privacy Enhancing Technol. Symp. (PETS)*, Jul. 2014, pp. 304–331.
[24] X. Wang, J. Shi, and G. Li, "Towards analyzing family misconfiguration in Tor network," *Computer Science and its Applications* (Lecture Notes in Electrical Engineering book series), vol. 203. Dordrecht, The Netherlands: Springer, 2012, pp. 503–514.
[25] J.-F. Raymond, "Traffic analysis: Protocols, attacks, design issues, and open problems," in *Designing Privacy Enhancing Technologies*. Berlin, Germany: Springer, 2001, pp. 10–29.
[26] M. Herrmann and C. Grothoff, "Privacy-implications of performance-based peer selection by onion-routers: A real-world case study using I2P," in *Proc. Int. Symp. Privacy Enhancing Technol. Symp.* Berlin, Germany: Springer, 2011, pp. 155–174.
[27] D. McCoy, K. Bauer, D. Grunwald, T. Kohno, and D. Sicker, "Shining light in dark places: Understanding the Tor network," in *Proc. Int. Symp. Privacy Enhancing Technol. Symp.* Berlin, Germany: Springer, 2008, pp. 63–76.

[28] X. Fu, Y. Zhu, B. Graham, R. Bettati, and W. Zhao, "On flow marking attacks in wireless anonymous communication networks," *J. Ubiquitous Comput. Intell.*, vol. 1, no. 1, pp. 42–53, 2007.
[29] X. Hei and X. Du, "Biometric-based two-level secure access control for implantable medical devices during emergencies," in *Proc. IEEE INFOCOM*, Apr. 2011, pp. 346–350.
[30] X. Du and H. H. Chen, "Security in wireless sensor networks," *IEEE Wireless Commun. Mag.*, vol. 15, no. 4, pp. 60–66, Aug. 2008.
[31] S. Liang and X. Du, "Permission-combination-based scheme for Android mobile malware detection," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2014, pp. 2301–2306.

LIN YE received the Ph.D. degree from the Harbin Institute of Technology in 2011. From 2016 to 2017, he was a Visiting Scholar with the Department of Computer and Information Sciences, Temple University, USA. His current research interests include network security, peer-to-peer networks, network measurement, and cloud computing.



XIANGZHAN YU is currently a Professor with the School of Computer Science and Technology, Harbin Institute of Technology. His main research fields include network and information security, security of Internet of Things, and privacy protection. He has published one academic book and over 50 papers on international journals and conferences.

JUNDA ZHAO received the M.S. degree in computer science from the Harbin Institute of Technology in 2018.

DONGYANG ZHAN received the B.S. degree in computer science from the Harbin Institute of Technology in 2014, where he is currently pursuing the Ph.D. degree with the School of Computer Science and Technology. His research interests include cloud computing and security.

XIAOJIANG (JAMES) DU received the B.S. and M.S. degrees in electrical engineering from Tsinghua University, Beijing, China, in 1996 and 1998, respectively, and the M.S. and Ph.D. degrees in electrical engineering from the University of Maryland at College Park in 2002 and 2003, respectively. He is currently a tenured Professor with the Department of Computer and Information Sciences, Temple University, Philadelphia, USA. He has authored over 300 journal and conference papers in these areas, as well as a book published by Springer. His research interests are wireless networks, security, and systems.

MOHSEN GUIZANI received the bachelor's (Hons.) and master's degrees in electrical engineering and the master's and Ph.D. degrees in computer engineering from Syracuse University, Syracuse, NY, USA, in 1984, 1986, 1987, and 1990, respectively. He has served as the Associate Vice President of graduate studies with Qatar University, the Chair of the Computer Science Department, Western Michigan University, and the Chair of the Computer Science Department, University of West Florida. He is currently a Professor and the ECE Department Chair at The University of Idaho. His research interests include wireless communications and mobile computing, computer networks, mobile cloud computing, security, and smart grid.