Egyptian Informatics Journal 19 (2018) 199–205

Contents lists available at ScienceDirect

Egyptian Informatics Journal

journal homepage: www.sciencedirect.com

# Defense Against Protocol Level Attack in Tor Network using Deficit Round Robin Queuing Process

K. Sangeetha [a,*], K. Ravikumar [b]

[a] Bharathiyar University, Coimbatore, India
[b] Faculty of Computer Science Department, Tamil University, India

ARTICLE INFO

*Article history:*
Received 26 April 2017
Revised 25 January 2018
Accepted 24 March 2018
Available online 3 April 2018

*Keywords:*
Tor network
Novel traffic dividing and scheduling
Protocol-level attacks
Indistinguishability obfuscation
Fake traffic
Deficit Round Robin queuing

ABSTRACT

Among different types of network, Tor network is mostly referred as an onion-routing network which enables the anonymous communication and supports TCP applications over the network. The onion-routing is utilized by the Tor network for transmitting the information of users via virtual circuits which are created by several successive relays. The performance and security of the anonymous communication in Tor network have been improved by using Novel Traffic Dividing and Scheduling (NTDS) mechanism. Normally, this mechanism was used for preventing the basic versions of sniper attacks in the Tor network. On the other hand, protocol-level attacks were not possible to prevent since some malicious entry router may duplicate, alter, insert or delete the TCP stream cells from the sender. This may cause cell recognition errors at the exit router and these cell recognition errors were recognized at the destination which helps to identify the sender. Hence, in this article, a defense mechanism called Mid-DRRQ is proposed for protocol-level attacks against Tor network. In Mid-DRRQ mechanism, the protocol-level attacks are prevented by using indistinguishability obfuscation model which generates the fake traffic within the middle relay. This fake traffic is allowed on outgoing connections without any cover traffic. Moreover, the outgoing connections of middle relays are controlled by using Deficit Round Robin queuing process. Finally, the simulation results show that the proposed mechanism effectively prevents the protocol-level attacks against Tor network.

## 1. Introduction

The most necessity area on the network is an anonymous communication which appears ultimately within accomplish as a common commercial exploitation of Onion Routing [1]. Tor network is defined as the operational network for many decades and consists of nearly two thousand nodes and few hundred thousand users [2]. It is developed from Onion Router as open software for providing available online anonymity. The onion routing is referred as the levels of encryption where the original information is encrypted by using the destination information at many times and this information are transmitted through the virtual circuits consisting of several Tor relays. The relay node is used for decrypting the level of encrypted information and offering the encrypted information to the successive relay nodes in the circuit for decryption process. In Tor network, the original information is sent without denoting the destination information. It is used for different applications however various attacks have been detected in Tor network due to its growth of deployment [3,4]. The performance and anonymous communication in the Tor network was improved by Novel Traffic Dividing and Scheduling (NTDS) mechanism [5]. By using this mechanism, the traffic was divided according to the different metrics like throughput, delay, bandwidth capacity, and congestion level. The separated traffic was distributed over the each circuit. The basic versions of the sniper attacks were detected and prevented by using NTDS method. However, it was impossible to defend the protocol-level attacks in the Tor network which are also degrades the anonymity performance.

* Corresponding author at: 440/2, Murugalayam, Water Board Colony, Kattur, Alagapuram, Salem 636016, India.

*E-mail addresses:* sangeethaphdr@gmail.com (K. Sangeetha), ravikasi2001@yahoo.com (K. Ravikumar).

Peer review under responsibility of Faculty of Computers and Information, Cairo University.

Production and hosting by Elsevier

https://doi.org/10.1016/j.eij.2018.03.005
1110-8665/© 2018 Production and hosting by Elsevier B.V. on behalf of Faculty of Computers and Information, Cairo University.

Protocol-level attack is defined as the exploitation of specific feature or implementation bug of some protocol installed at the victim for consuming huge amount of its resources. These attacks are used for verifying that the Alice is communicating with Bob over Tor [6]. The attacker may control the entry and exit routers and execute protocol-level attacks by manipulating the cells related to the given circuit. The malicious entry routers contain the information such as source IP address, port with the circuit ID and the time of the cell being manipulated. Then the attacker launches the protocol-level attack in different ways such as duplication, modification, insertion or deletion of the information. Such modified information cells are transmitted to the exit router which detects the cell recognition errors occurred by the manipulated cells. By using these cell recognition errors, the attacker may confirm the target cell enters Tor through the malicious exit router and target cell exits Tor through the malicious exit router.

Hence in this article, a novel defense mechanism called MidDRRQ is proposed for mitigating the protocol-level attacks against Tor network. Initially, consider the protocol-level attack which may de-anonymize the users by modifying the cell. Then, the solution is provided for this attack by verifying the integrity of cells independently. In the proposed mechanism, indistinguishability obfuscation model is proposed for generating the fake traffic within the middle relays. The generation of fake traffic facilitates the outgoing connections based on the stochastic fair queuing process. However, the complexity of the stochastic fair queuing is high which increases the delay. Therefore, the delay is reduced by Deficit Round Robin (DRR) queuing process. Due to the DRR process, the outgoing connections on the middle relays are effectively controlled. Thus, the proposed mechanism effectively prevents the protocol-level attacks in the Tor network.

## 2. Related work

Tas et al. [7] proposed novel session initiation protocol based Distributed Denial-of-Service (DDoS) attacks and its defense mechanisms. Initially, the vulnerabilities in the Session Initiation Protocol (SIP) were investigated and the new vulnerabilities were identified in the SIP retransmission mechanisms. Then, two advanced attacks were developed by exploiting the vulnerabilities identified in the SIP retransmission mechanism. Moreover, a novel defense mechanism was introduced for preventing the proposed attacks effectively. However, the efficiency of the mechanism was not effectively analyzed. Ling et al. [8] proposed the new cell-counting based attack against Tor network. This attack was difficult for detecting and has ability for confirming the anonymous communication relationship among Tor network users. The transmission of cells from the target TCP stream were manipulated by an attacker at malicious exit router and the secret signal was embedded into the cell counter variation of the TCP stream [9]. The embedded signal was recognized by accomplice of the attacker at the entry router by using the proposed recovery algorithms. However, the defending against this attack was still complex due to basic design of Tor network. Danner et al. [10] proposed two algorithms in order to identify the DoS attack in Tor networks. In this algorithm, the efficient analysis of parameters was introduced for defining the DoS attacks along with the analytic model of the attacker's effectiveness. The DoS attackers were identified by means of number of circuit's generation approach which is linear in the number of relays in the network. Moreover, the set of assumptions about the attacker and the other probabilistic were also effectively proved. Panchenko et al. [11] investigated about the performance and anonymity of the Tor network. In this study, the bottlenecks in the Tor network were analyzed and path selection mechanism was introduced for utilizing the accessible capabilities in the heterogeneous network. The proposed mechanism was introduced based on the combination of distantly calculated recent load of the nodes and an evaluation of their highest capability. Hence, the onion routing performance was improved by using this mechanism. Ming [12] proposed stochastic fairness queuing mechanism for mitigating the flooding-based DDoS attacks in the network. In this mechanism, the feasibility of mitigating the UDP flooding attacks was explored by classless queuing principles like Stochastic Fairness Queuing (SFQ) and First Come First Served (FCFS). Then the comparative analysis was obtained between SFQ and FCFS on their efficiency and robustness in mitigation of UDP flooding based DDoS attacks. However, the defense mechanisms require more effective solutions. Gayathri and Karthick [13] proposed different queuing algorithms for mitigating the DDoS attacks. Here, the mitigation against DDoS attack was provided by the Stochastic Fairness Queuing (SFQ) algorithm and Class Based Queuing (CBS) algorithm which are proposed as the countermeasures for defending the DDoS attack. In addition, Machine Learning Automatic Defense System (MLADS) was utilized based on the Flexible Deterministic Packet Marking (FDPM) for tracing back the source of the DDoS attacks. However, the process overhead due to queuing algorithms was high. The comparison of related works with the proposed work is listed in Table 1.

Based on the drawbacks of the related works, the proposed mechanism is introduced for mitigating the protocol-level attacks by using Deficit Round Robin queuing & Indistinguishability obfuscation. It achieves reduced complexity and queuing delay.

The remaining article is programmed as follows: Section 2 explains the proposed defense mechanism for protocol-level attacks. Section 3 illustrates the simulation results of the proposed mechanism compared with the existing mechanism. Section 4 concludes the research work.

## 3. Proposed methodology

In this section, the proposed defense mechanism for mitigating the protocol-level attacks against Tor networks is explained. Initially, the model of indistinguishability obfuscation is proposed for generating the fake traffic within the middle relays. Then, the Deficit Round Robin (DRR) queuing algorithm is introduced for handling the outgoing connections on the middle relays.

### 3.1. Tor protocol

Each circuit is Tor network is shared among several TCP streams for minimizing the setup delay. In the Tor network, Onion Router (OR) communicates by using Transport Layer Security (TLS) sockets. This communication prevents the adversaries from having the control over network traffic for modifying the data, impersonating an OR and reading the plaintext data [14]. The users may choose the three OR and negotiate the virtual circuit. The command field can be Cell_Padding, Cell_Create/Cell_Created, Cell_Versions, Cell_Destroy or Cell_Relay. The Tor cell format is shown in Tables 2 and 3.

There are two types of cells such as control cells and relay cells. All cells have the header consists of Circ_ID which specifies the referring circuit and command field which describes what to do with the cell's payload. The payload is consisting of 509 bytes including with the multiple layers of encryption. In Relay cell format, the Recognized field indicates whether the cell is for this node or must be transmitted to next OR. The stream which is related with this cell is declared by Stream_ID field. Then, the Length field determines the actual length of cell's payload since cells are padded to the common fixed size. The application-level data is packed into the equivalent sized cells such as 512 bytes in Tor.

**Table 1**
Comparison of related works.

| Ref. No. | Type of attack detection | Methods used | Merits | Demerits |
|---|---|---|---|---|
| [7] | Dos/DDos attack | Session initiation protocol | Average CPU load is decreased | Efficiency was not effectively analyzed |
| [8,9] | Cell-counting-based attack | Anonymizer framework | Better effectiveness & feasibility | Detection of attack has still high complexity |
| [10] | DoS attack | A simplified, practical detection algorithm | Better efficiency | High computational complexity |
| [11] | Bottlenecks | Path selection mechanism | Better anonymity | High jitter |
| [12] | DDoS and flooding-based attacks | Stochastic fairness queuing & First come first served | More efficient & robust | Only throughput performance was analyzed. |
| [13] | DDoS attack | Stochastic fairness queuing, Class based queuing & Machine learning automatic defense system | Better efficiency and security | High process overhead |

**Table 2**
Tor control cell format.

| | Circ_ID | Command | Payload |
|---|---|---|---|
| Bytes | 2 | 1 | 509 |

Tor users utilize Onion Proxy (OP) for receiving the directory information, establishing circuits in the network and managing the application-level connections.

### 3.2. Indistinguishability obfuscation

An indistinguishability obfuscation is defined as the transformation iO of an input program iI(P) to an output program iO(P) and has following properties.

- For all security parameters $\gamma$ and for all programs P and for all inputs $a$, the obfuscated program $P' = iO(\gamma P)$ and the original program P should give the same output with probability 1 as follows:

$$Pr[P(a) = P'(a)] = 1 \tag{1}$$

  - For any pair of programs $P_1$ and $P_2$ which provides the similar output for all inputs $a$, obfuscated programs $O_1 = iO(\gamma, P_1)$ and $O_2 = iO(\gamma, P_2)$ must be indistinguishable with probability $1 - \alpha(\gamma)$ such as follows:

$$\begin{aligned} &Pr[\forall a.P_1(a) = P_2(a)] > 1 - \alpha(\gamma) \Rightarrow \\ &|Pr[D(O_1) = 1] - Pr[D(O_2) = 1]| < \alpha(\gamma) \end{aligned} \tag{2}$$

In Eq. (2), $D(\cdot)$ refers the decision function of the adversary which outputs 0 or 1 denoting that it recognizes the given program as an obfuscation of program $P_1$.

According to the properties, the Probabilistic Polynomial Time (PPT) adversary cannot differentiate obfuscated programs of two programs $P_1$ and $P_2$, if original programs operate like each other. $P_1$ and $P_2$ may have the negligible difference in their input or output functionality with the low probability. Therefore the adversary has no ability to notify which original program was utilized for generating the given obfuscated program even by inspecting the internals of their corresponding obfuscated programs.

### 3.3. Integrity checking and generation of fake traffic

In addition, integrity checking and fake traffic generation processes are performed for transmitting the Tor cells on middle nodes which prevents the protocol-level attacks. Initially, the structure of TOR cells must be updated for satisfying the integrity requirement in each OR. In this approach, one integrity field is considered for each OR in the cell. The format of TOR control cells and relay cells are shown in Tables 4 and 5.

Each field is filled by the correct digest value in which corresponding OR can be determined. In each layer, include the field for having the integrity field for all OR. After decryption of that layer, the integrity may be checked. The integrity field is replaced in Tor cells with three digest fields. In this mode, the encryption process is used for encrypting the integrity fields as payloads. The integrity field of $OR_1$ is encrypted with $OR_1$'s key whereas the integrity field of $OR_2$ and $OR_3$ are encrypted in multilayer manner. Then, every OR decrypts the cells. After decrypting the payload and set of integrity fields, each OR verifies the validity of the first integrity field. Then it will modify the content of its integrity field and rotates the array of integrity fields. Therefore the leaking of any information about the location of OR in the circuit is avoided.

Then fake traffics are generated and categorized in five major groups. The first group is end-to-end fake traffic which is also known as fake flow. The fake flows start from the original sender itself and will be delivered to the final receiver. The second group consists of fake packets. The fake packets are exchanged between neighbor mixes or between the sender and the first mix or between the last mix and receiver. The remaining three groups are consisting of three types of messages. The difference between messages and packets is that in opposite of packets, messages are multilayer encrypted. The encryption is performed with appropriate keys of path nodes. Prior-fake messages are started from the sender and end in the middle nodes. Mid-fake messages are exchanged between non-neighbor mix nodes. Also, the post-fake messages are started at mix nodes and end in the receiver.

In the proposed mechanism, the post-fake messages are introduced by middle OR such as $OR_2$. Across the key negotiation and circuit establishment, OP generates an instance of indistinguishability obfuscation and transmits it to the middle node $OR_2$. This process may generate the fake messages. Onion secret key of subsequent OR is kept in the obfuscated. Since the keys are obfuscated, $OR_2$ cannot extract them and compromise the anonymity. OP must

**Table 3**
Tor relay cell format.

| | Circ_ID | Command | Relay_Command | Recognized | Stream_ID | Integrity | Length | Relay_Payload |
|---|---|---|---|---|---|---|---|---|
| Bytes | 2 | 1 | 1 | 2 | 2 | 4 | 4 | 498 |

**Table 4**
Proposed tor control cell format.

| | Circ_ID | Command | Payload |
|---|---|---|---|
| Bytes | 2 | 1 | 509 |

transmit fake flows with a particular frequency for completing the defense mechanism if it has no data for transmitting across the circuit. Thus, $OR_2$ generates the fake messages by obfuscated method automatically.

Then the contents of post-fake messages are determined by assuming that the exit node is compromised by the attacker and since the traffic contents are revealed at the exit node, random content will disclose fake messages and attacker can defeat the defense easily. Therefore obfuscated method is used for requesting and generating the valid traffic which attacker cannot recognize it. The responses of these post fake messages can be managed by OP. These fake traffic messages are used for allowing outgoing cells or connections of middle relay nodes based on stochastic fair queuing process [15]. However, the process complexity of stochastic fair queuing is high. Therefore, in the proposed mechanism, Deficit Round Robin (DRR) queuing mechanism is introduced.

### 3.4. Deficit Round Robin (DRR) queuing algorithm

Deficit Round Robin (DRR) is defined as the scheduling algorithm for the network scheduler. In DRR, the scheduler handling $N$ flows is configured with quantity $Q_N$ for each flow [16]. The all $N$ flows are having the continuous stream of arbitrary sized packets arriving to the router and all these flows required to leave the router on the similar outgoing connection. The fairness index for a $N$th flow is defined as follows:

$$Fairness\ Index_N = \frac{FQ_N \cdot \sum_{i=1}^{N} f_i}{f_N} \tag{3}$$

$$\text{where } FQ_N = Max\left(\lim_{t\to\infty} \frac{sent_{N,t}}{sent_t}\right) \tag{4}$$

Packets coming in on different flows are stored in different queues. The deficit counter is used for keep tracking of the credit available to each queue. The variable quantum is referred as the number of bytes which each queue $N$ may send in each round $r$. Each queue $N$ is allowed for transmitting the packets in the first round subject to the restriction that $bytes_{N,1} \leqslant Q_N$. If there are no more packets in queue $N$ after the queue has been services, the state variable known as $Deficit\_Counter_N$ is reset to 0. Otherwise, the remaining number $(Q_N - bytes_{N,r})$ is stored in the $Deficit\_Counter_N$. In subsequent rounds, the amount of bandwidth usable by this flow is the sum of $Deficit\_Counter_N$ of the previous round included to $Q_N$.

The analysis of empty queues is avoided by keeping an auxiliary list $Active\_List$ which is the list of indices of queues which contain at least one packet. Whenever the packet arrives to the previously empty queue $N$, $N$ is added to the end of $Active\_List$. Whenever index $N$ is at the head of $Active\_List$, the mechanism services up to $Q_N + Deficit_Counter_N$of bytes from queue$N$; if at the end of this service opportunity, queue $N$ still has the packets to transmit, the index $N$ is moved to the end of $Active\_List$. Otherwise, $Deficit\_Counter_N$ is set to zero and index $N$ is removed from $Active\_List$. The algorithm of DRR is given below.

**Algorithm** (*DRR Mechanism*).

```
Initialization
$For(i = 0; i < n; i + 1)$
$Deficit\_Counter_N = 0;$
//Enqueuing Process
On arrival of packet $P$
$N = Extract\_Flow(P)$
$If(Exists\_In\_Active\_List(N) = False)$ then
$Insert\_Active\_List(N)$
$Deficit\_Counter_N = 0;$
$If(nofreebuffersleft)$then
$Free\_Buffer()$
$Enqueue(N, P)$
//Dequeuing Process
$While(True)$ do
$If(Active\_Listisnotempty)$
Remove head of $Active\_List$
$Deficit\_Counter_N = Q_N + Deficit\_Counter_N$
$While((Deficit\_Counter_N > 0)and(Q_N notempty))$
$Packet\_Size = Size(Head(Q_N))$
$If(Packet\_Size \leqslant Deficit\_Counter_N)$
$Send(Dequeue(Q_N))$
$Deficit\_Counter_N = Deficit\_Counter_N - Packet\_Size$
Else break
$If(Empty(Q_N))$
$Deficit\_Counter_N = 0$
Else $Insert\_Active\_List(N)$
End
```

Finally, the accuracy of protocol-level attacks is validated based on the correlation coefficient which is used for measuring the strength of correlation between the time of manipulating cells and the time of detecting the cell recognition errors. The correlation coefficient is defined as follows:

$$r = \frac{\sum_{x,y}(x - \bar{x})(y - \bar{y})}{\sqrt{\sum_x (x - \bar{x})^2}\sqrt{\sum_y (y - \bar{y})^2}} \tag{5}$$

In Eq. (5), $x$ refers the time of manipulating cells at the entry onion router, $y$ refers the time of cell recognition errors incurring at the exit router and $\bar{x}$ and $\bar{y}$ are mean values of $x$ and $y$ respectively. Based on this correlation coefficient the robustness of the proposed mechanism against protocol-level attacks is measured.

## 4. Results and discussion

In this section, the performance of the existing mechanisms and proposed mechanism is compared. The performance is evaluated by using Network Simulator version 2 (NS-2.34) for defense against protocol-level attacks and the comparison of Mid-DRRQ is made between NTDS method and existing Mid-defense mechanism. In this experiment, 100 nodes are used and two malicious OR are used as the Tor entry OR and exit OR. In which, the entry OR is located nearer to the client (Alice) and exit OR is located closer

**Table 5**
Proposed tor relay cell format.

| | Circ_ID | Command | Digest 1 | Digest 2 | Digest 3 | Relay_Command | Stream_ID | Length | Relay_Payload |
|---|---|---|---|---|---|---|---|---|---|
| Bytes | 2 | 1 | 6 | 6 | 6 | 1 | 2 | 2 | 484 |

to the server (Bob). The middle OR is selected by using the path selection algorithm. In addition, the considered path length in the proposed mechanism is three. The performance of the proposed mechanism is evaluated in terms of different metrics such as PDR, throughput, end-to-end delay, probability that a circuit selects the malicious routers based on percentage of malicious routers and path length. These metrics are used for analyzing the effectiveness of the path selection algorithm and also the anonymity of the Tor network.

### 4.1. Packet Delivery Ratio

Packet Delivery Ratio (PDR) is defined as the deviation of the number of delivered data packets to the destination or clients. It is computed as,

$$PDR = \frac{Number\ of\ packet\ delivered}{Number\ of\ packet\ transmitted} \quad (6)$$

According to the PDR, the packets are transmitted securely through the selected number of Tor circuits without any malicious routers.

Fig. 1 shows that the comparison of packet delivery ratio made between NTDS and Mid-DRRQ mechanism. X-axis considers the number of nodes and Y-axis denotes the packet delivery ratio in %. The graph result illustrates that the proposed Mid-DRRQ mechanism has high PDR compared to the NTDS approach. Thus, the highest PDR proves that the packets are transmitted to the destination with high secrecy level.

### 4.2. Throughput

Throughput is defined as the amount of data packets which are delivered successfully through the Tor OR in the network.

$$Throughput = \frac{Number\ of\ packet\ transmitted}{Time\ taken\ for\ packet\ transmission} \quad (7)$$

Based on the highest throughput, the stability of the nodes is improved and also the number of Tor circuits is minimized.

Fig. 2 shows that the comparison of throughput made between NTDS and Mid-DRRQ mechanism. In the graph, x-axis considers the number of nodes and y-axis denotes the throughput in %. The graph result illustrates that the proposed Mid-DRRQ mechanism has high throughput compared to the NTDS approach. Thus, the highest throughput improves the stability and reduces the number of Tor circuit in the network.

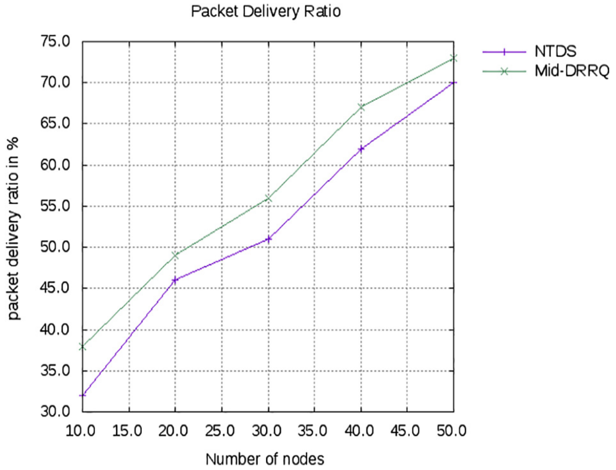

**Fig. 1.** Packet delivery ratio (%).

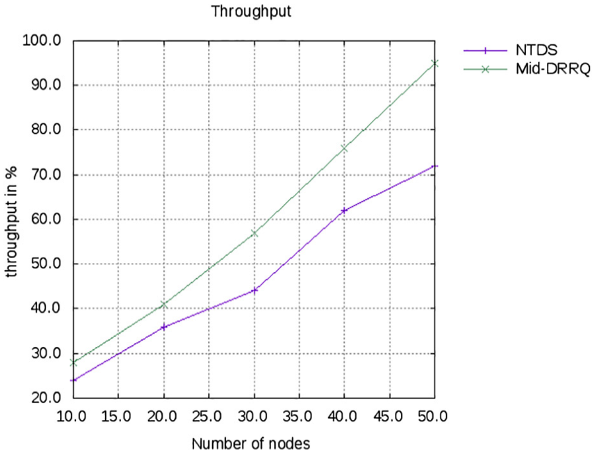

**Fig. 2.** Throughput (%).

### 4.3. End-to-end delay

End-to-end delay is defined as the time duration which is taken for the packet to be forwarded in the network between server (Bob) and client (Alice).

$$End - to - End\ Delay = \frac{Time\ taken\ for\ packets\ transmission}{Number\ of\ packet\ received\ by\ the\ destination} \quad (8)$$

Delay is considered as one of the performance metrics which indicates that the Tor network has less number of malicious routers while transmitting the packets from source to destination over the network.

Fig. 3 shows that the comparison of end-to-end delay made between NTDS and Mid-DRRQ mechanism. X-axis considers the number of nodes and Y-axis denotes the end-to-end delay (ms). The graph result illustrates that the proposed Mid-DRRQ mechanism has less end-to-end delay compared to the NTDS approach. Thus, less delay improves the anonymity of the Tor network.

### 4.4. Percentage of malicious routers versus probability that a circuit chooses malicious routers as entry & exit routers

The Probability that a circuit chooses the malicious routers is the probability value that the circuit selects the malicious onion

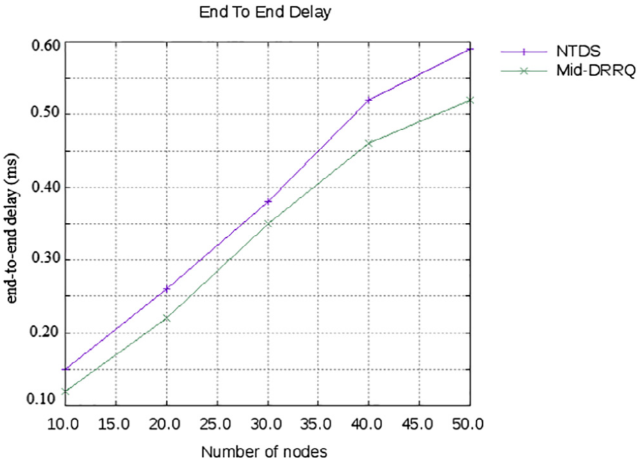

**Fig. 3.** End-to-end delay.

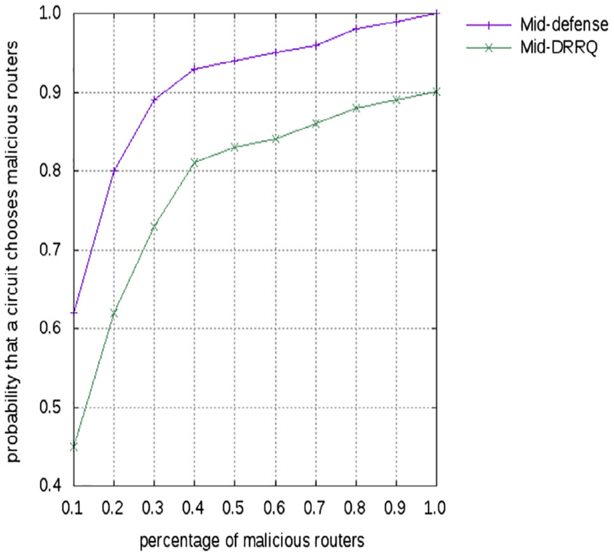

**Fig. 4.** Percentage of malicious routers versus probability that a circuit chooses malicious routers.

routers as entry and exit routers within the Tor network and it is computed as follows:

$$P(E) = \frac{E}{N} \cdot \frac{2Q - E}{N - E} \tag{9}$$

In Eq. (9), $N$ refers the number of onion routers in Tor network, $2Q$ refers the number of malicious onion routers and $E$ refers the number of malicious entry onion routers. The probability that a circuit chooses the malicious onion routers as entry and exit routers for the proposed Mid-DRRQ mechanism is reduced while increasing the percentage of malicious routers in Tor network i.e., the proposed mechanism improves the security of the Tor network by reducing the selection of malicious routers.

Fig. 4 shows that the comparison of probability that a circuit chooses the malicious routers made between existing Mid-defense and proposed Mid-DRRQ mechanism. X-axis considers the percentage of malicious routers and Y-axis denotes the probability that a circuit chooses malicious routers. The graph result illustrates that the proposed Mid-DRRQ mechanism has less probability that a circuit chooses malicious nodes while increasing the percentage of malicious routers which is compared to the NTDS approach. Thus, less probability of selecting the malicious routers by Tor circuit increases the security of the packet transmission without any attacks.

### *4.5. Path length versus probability that a circuit chooses malicious routers*

The probability that a circuit selects all malicious routers based on the path length $(j = 3)$ is calculated as follows;

$$P_j = \sum_{i=1}^{k}\left(b_i \sum_{j=1,j\neq i}^{k} \frac{b_j}{1-b_i}\left(\sum_{l=1,l\neq i,l\neq j}^{k} \frac{b_l}{1-b_i-b_j}\right)\right), \quad k < N \tag{10}$$

$$\text{where } b_i = \frac{B_i}{\sum_{i=1}^{k+N} B_i} \tag{11}$$

Path length is defined as the number of onion routers in the Tor network. In Eq. (10), $b_i$ refers the probability that the onion router chooses the bandwidth $B_i$. The probability that a circuit chooses all routers as malicious is decreased when path length of the Tor network is increased.

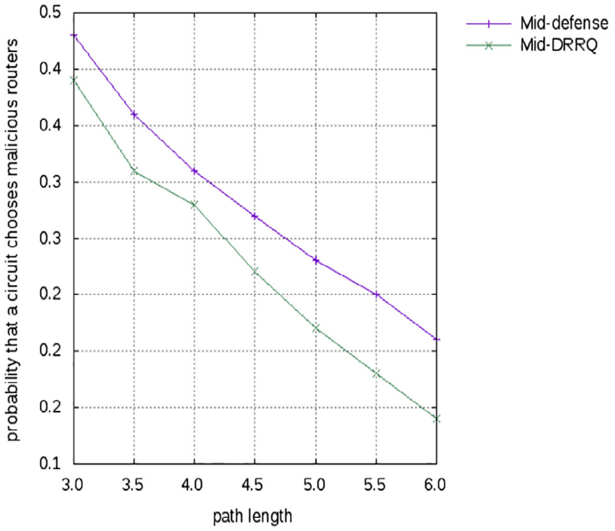

**Fig. 5.** Path length versus probability that a circuit chooses malicious routers.

Fig. 5 shows that the comparison of probability that a circuit chooses the malicious routers made between existing Mid-defense and proposed Mid-DRRQ mechanism. X-axis considers the path length and Y-axis denotes the probability that a circuit chooses malicious routers. The graph result illustrates that the proposed Mid-DRRQ mechanism has less probability that a circuit chooses malicious nodes while increasing the path length which is compared to the NTDS approach. Thus, reducing the probability that a circuit chooses all malicious routers refers the efficiency of the proposed mechanism and it indicates that the path length does not impact on the protocol-level attacks since these attacks only require malicious entry and exit routers along the Tor circuit.

## 5. Conclusion

In this article, the novel defense mechanism is proposed for mitigating the protocol-level attacks against Tor network. In this proposed defense mechanism, the model of indistinguishability obfuscation is used for defending the protocol-level attacks based on the two main approaches such as integrity checking and fake traffic generation within the middle relay nodes. The generated obfuscated fake traffic is forwarded on outgoing connections of middle relays by using the stochastic fair queuing model. However, the complexity and setup delay of queuing model is high. Therefore, the Deficit Round Robin algorithm is proposed for reducing both complexity and delay by enhancing the flow fairness and controlling the outgoing connections of middle relays. Finally, the experimental results prove that the proposed defense mechanism for protocol-level attacks has better performance than the other mechanisms.

## References

[1] Syverson P, Tsudik G, Reed M, Landwehr C. Towards an analysis of onion routing security. In: Designing privacy enhancing technologies; 2001. p. 96–114.

[2] Dingledine R, Mathewson N, Syverson P. Tor: the second-generation onion router. Washington DC: Naval Research Lab; 2004.

[3] Loesing K. Measuring the Tor network: evaluation of client requests to the directories. The Tor project. Technical report; 2009.

[4] Goodin D. Tor at heart of embassy passwords leak. The Regist; 2007.

[5] Sangeetha K, Ravikumar K. A novel traffic dividing and scheduling mechanism for enhancing security and performance in the tor network. Indian J Sci Technol 2015;8(7):689.

[6] Ling Z, Luo J, Yu W, Fu X, Jia W, Zhao W. Protocol-level attacks against Tor. Comput Netw 2013;57(4):869–86.

[7] Tas IM, Ugurdogan B, Baktir S. Novel session initiation protocol-based distributed denial-of-service attacks and effective defense strategies. Comput Secur 2016;63:29–44.

[8] Ling Z, Luo J, Yu W, Fu X, Xuan D, Jia W. A new cell-counting-based attack against Tor. IEEE/ACM Trans Netw (ToN) 2012;20(4):1245–61.

[9] Ling Z, Fu X, Jia W, Yu W, Xuan D, Luo J. Novel packet size-based covert channel attacks against anonymizer. IEEE Trans Comput 2013;62(12):2411–26.

[10] Danner N, Defabbia-Kane S, Krizanc D, Liberatore M. Effectiveness and detection of denial-of-service attacks in Tor. ACM Trans Inf Syst Secur (TISSEC) 2012;15(3):11.

[11] Panchenko A, Lanze F, Engel T. Improving performance and anonymity in the Tor network. In: International performance computing and communications conference (IPCCC); 2012. p. 1–10.

[12] Ming Y. Mitigating flooding-based DDoS attacks by stochastic fairness queueing. Adv Inf Sci Serv Sci 2012;4:145–52.

[13] Gayathri R, Karthick M. Defending DDoS attacks: implementation of SFQ with CBQ and MLADS. Int J Eng Res Technol (IJERT) 2014;3(4):2229–31.

[14] Soltani M, Najafi S, Jalili R. Mid-defense: mitigating protocol-level attacks in TOR using indistinguishability obfuscation. In: International ISC conference on information security and cryptology (ISCISC); 2014. p. 214–9.

[15] McLachlan J, Hopper N. Don't clog the queue! Circuit clogging and mitigation in P2P anonymity schemes. In: International conference on financial cryptography and data security; 2008. p. 31–46.

[16] Shreedhar M, Varghese G. Efficient fair queuing using deficit round-robin. IEEE/ACM Trans Netw 1996;4(3):375–85.