Computers & Security 150 (2025) 104224

Contents lists available at ScienceDirect

Computers & Security

journal homepage: www.elsevier.com/locate/cose

Full length article

Enhancing network tunnels anonymity through increasing combined traffic in a clustered structure

Reza Mirzaei [a], Nasser Yazdani [a], Mohammad Sayad Haghighi [a,b,*]

[a] School of Electrical and Computer Engineering, College of Engineering, University of Tehran, 1439957131, Iran
[b] School of Computer Science, Institute for Research in Fundamental Sciences (IPM), Iran

ARTICLE INFO

Keywords:
Anonymity network
Tor
I2P
Shannon entropy
Degree of anonymity
Secure protocol

ABSTRACT

Given the visibility of the tunnel creation phase within tunnel-based anonymity structures, an entity's traffic can be segregated based on the relay selection mechanism employed. Hence, the global attacker's capability to detect communications and undermine the anonymity of entities is heightened. Another factor that can aid the attacker in identifying the tunnels is the improper combination of created tunnels and variations in the positioning of a combined relay within those tunnels. One potential solution to address these issues is to combine tunnels traffic by restricting the list of selectable relays. This can be accomplished by taking into account the choices made by tunnel owners and the network structure, as well as ensuring that the common selected relays occupy the same positions within the tunnels. We propose a clustering structure with routing capabilities to establish an infrastructure for creating combined tunnels. Our method has two key pillars. Firstly, both the tunnel creation packets and data packets follow the same pattern, making it difficult for the attacker to differentiate tunnel creation traffic from regular network traffic. Secondly, by allowing entities to join different clusters and maintaining a high ratio of entities to the number of interfaces within each cluster, the probability of combining traffic from senders within the same cluster is significantly increased. These interfaces within the proposed structure are referred to as permanent relays. Given the hierarchical nature of the proposed structure, the positions of relays within the tunnels of a cluster remain consistent. To assess the effectiveness of the proposed structure, we employ the average degree of anonymity metric, which relies on the Shannon entropy concept. Simulation results demonstrate a substantial increase in the degree of anonymity achieved by the proposed structure in comparison to previous approaches.

1. Introduction

The importance of user privacy on the Internet necessitates the presence of mechanisms to address risks that pose a threat to users' information (Safaei Pour et al., 2023; Heurix et al., 2015). While encryption can safeguard the content of messages from attackers, the metadata within these messages still holds the potential to reveal the identity information of the communication endpoints. Anonymization emerges as a key solution for preserving the identities of communication parties. By leveraging anonymity, users can effectively conceal their identity information (Panchenko et al., 2009; SeyedHassani et al., 2019; Angel and Setty, 2016) and behavioral patterns (Haghighi and Mohamedpour, 2008, 2010; Sayad Haghighi and Aziminejad, 2020) from external attackers.

Conceptually, anonymity can be defined as follows (Pfitzmann et al., 1991a): "anonymity is the state of being not identifiable within a set of subjects, the anonymity set". In this definition, the anonymity set refers to "the set of all possible subjects who might cause an action". Consequently, the greater the number of actors associated with a specific action, the higher the level of anonymity for the primary actor. Several different architectures have been suggested for designing anonymity networks. These include structures based on cryptography (Chaum, 1988; Waidner and Pfitzmann, 1990; Golle and Juels, 2004), random-walk (Reiter and Rubin, 1998; Muñoz-Gea et al., 2008; Fleming et al., 2014), mixing (Chaum, 1981; Danezis et al., 2003; Pfitzmann et al., 1991b), tunneling (Dingledine et al., 2004; Anon., 2022) and others (Mislove et al., 2004; Nambiar and Wright, 2006; Goel et al., 2003). Among these architectures, tunneling-based structures utilizing onion routing have gained popularity due to their ability to offer a low-latency mechanism for message forwarding.

In this paper, we examine the vulnerabilities associated with tunneling-based anonymity mechanisms. To conduct a more precise evaluation, we employ the average degree of anonymity, which utilizes

* Corresponding author at: School of Electrical and Computer Engineering, College of Engineering, University of Tehran, 1439957131, Iran.
E-mail addresses: mirzaei.reza@ut.ac.ir (R. Mirzaei), yazdani@ut.ac.ir (N. Yazdani), sayad@ut.ac.ir (M. Sayad Haghighi).

https://doi.org/10.1016/j.cose.2024.104224
Received 5 January 2024; Received in revised form 12 August 2024; Accepted 26 August 2024
Available online 27 November 2024

Shannon information theory (Shannon, 1948) to quantify the level of certainty regarding an event. In order to explore the issues within tunneling-based structures, we consider the following threat model for the attacker, while assuming the central server is immune to compromise.

1. **Global**: The attacker possesses unrestricted access to the entire anonymity network and maintains complete supervision and control over all links within the network.
2. **Passive**: The attacker refrains from engaging in any destructive actions that could compromise the integrity of the messages exchanged within the anonymity network and solely focuses on monitoring the desired messages.
3. **External**: The attacker does not take part in the anonymity network and does not exert control over any of the entities within the network.

The primary issues encountered in the tunneling mechanism stem from the inadequate combination of entities' traffic within the network. Consequently, it becomes crucial to establish conditions that promote a high probability of traffic aggregation among entities. This, in turn, makes it more challenging for the attacker to discern their behavioral patterns. However, it should be noted that the attacker retains the ability to identify tunnels based on the detectability of the tunnel creation phase. In anonymity networks that utilize cover traffic, the actual messages are transmitted through tunnels, while the cover traffic is directed to random entities within the network without relay intervention. This approach effectively neutralizes the impact of cover traffic through tunnel identification. The proposed structure incorporates a relay selection procedure that enhances the likelihood of selecting combined relays. Furthermore, the combined relay is consistently positioned within the tunnels presented in a cluster. Consequently, this approach significantly amplifies the combination of traffic from active tunnels throughout the network. Moreover, in the proposed structure, the data transmission traffic and tunnel creation traffic exhibit similarity, making it difficult for the attacker to distinguish between them and identify the tunnel creation phase.

As illustrated in Fig. 1, the permanent relays play a crucial role in facilitating communication between clusters. Given the relatively larger number of entities compared to the permanent relays within a cluster, the probability of combining multiple tunnels at the initial relay originating from the same cluster is heightened. When multiple tunnels are combined, the position of a common relay within those tunnels remains consistent. Consequently, their traffic patterns become highly similar. In the proposed structure, an entity has the capability to engage in multiple clusters both as a regular participant (sender/receiver) and simultaneously as a permanent relay in several other clusters. The primary contributions of this paper can be summarized as follows:

* The study investigates weaknesses in establishing a tunnel within the anonymity structure, with a specific focus on the ability to combine traffic patterns. Key aspects of this examination include the simultaneous construction of tunnels, concurrent transmissions within the established tunnels, and the placement of common relays within these tunnels.
* Establishing a structure designed to enhance the likelihood of common relays being strategically positioned within the tunnels significantly increases the combination of traffic patterns. Consequently, the attacker's ability to differentiate the traffic sent by the senders becomes negligible.
* Presentation of a novel method that employs simulation and the calculation of the average degree of anonymity to assess the improvements achieved in comparison to the identified weaknesses.

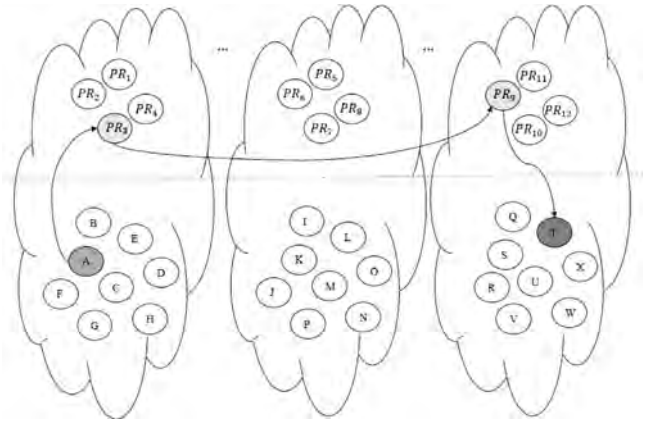

Fig. 1. The proposed structure based on clustering and permanent relays.

Our research introduces a novel clustered structure with enhanced secure routing capabilities aimed at improving anonymity in network communications. This new approach offers an alternative to existing anonymity networks like Tor and I2P. By integrating an innovative clustered architecture and advanced secure routing mechanisms, we enhance both anonymity and security features. While our structure employs a tunneling principle similar to traditional anonymity networks, it builds upon their foundational infrastructure and protocols to further strengthen anonymity and address existing vulnerabilities. Through dynamic relay selection and combined traffic management, we tackle common critical weaknesses, thereby boosting the network's overall anonymity and resilience. Our approach offers a robust solution for secure and anonymous communication.

In the proposed structure, packets are uniformly encrypted and sent in an identical format. The specific structure and purpose of each received packet are discernible only after decryption by the intended recipient, preventing attackers from identifying packet types from the transmitted traffic. Additionally, by standardizing the sizes of both tunnel creation and data transmission packets, our approach complicates adversaries' attempts to distinguish between different packet types based on size alone.

To combat adaptive adversaries and deep packet inspection, we incorporate enhanced traffic merging, dynamic relay selection, and frequent tunnel re-establishment. These measures are designed to enhance the robustness of our network against sophisticated threats, ensuring heightened security and anonymity for users.

The remaining sections of the paper are organized as follows. The next section provides a review of two widely-used architectures that employ onion routing and utilize a tunneling mechanism to establish anonymity networks. Section 3 discusses the problems associated with the tunneling mechanism, analyzing them through various scenarios. Section 4 introduces the proposed structure aimed at addressing these issues and improving the overall system. Section 5 presents the simulation results, security analysis and a comparison between different structures. It also discusses some limitations of the proposed structure and offers solutions to these challenges. Lastly, in Section 6, the article concludes with a summary of findings and conclusions.

2. Background and related work

Although our primary focus is on low-latency, tunnel-based approaches for achieving anonymity, it is also important to mention mix networks (Chaum, 1981; Sampigethaya and Poovendran, 2007; Ahmad and Kamal, 2019), which offer significant anonymity benefits. Mix

networks, provide an alternative method by rearranging and shuffling messages to obfuscate the connection between sender and receiver. In Mix networks, messages are encrypted and routed through a series of intermediate nodes, known as mixes, which shuffle and re-encrypt the messages to prevent eavesdroppers from linking the sender to the receiver. The system's effectiveness relies on techniques such as batching messages, introducing artificial delays, and using complex encryption algorithms to obscure communication patterns.

This approach provides a higher degree of anonymity but at the expense of increased latency. Despite the latency drawback, mix networks offer robust protection against traffic analysis attacks. Due to the excessive delay associated with these structures, they will be excluded from further consideration in this paper.

Two well-known anonymity networks that employ tunneling mechanisms to anonymize messages are Tor and I2P. Tor utilizes two-way tunnels, where messages are exchanged within the same tunnel for both sending and receiving. On the other hand, I2P separates input and output tunnels, keeping them distinct from each other. The subsequent explanation provides a brief overview of the general structure of these two anonymity networks.

Tor, introduced in 2004 by Dingledine et al. (2004), is the most prominent low-latency anonymity network that operates based on the onion structure. It has served as the foundation for the development of numerous new structures (Al-E'mari et al., 2023; Jawaheri et al., 2020; Gegenhuber et al., 2023). Over the years, significant efforts have been made to evaluate (Snader and Borisov, 2011; Panchenko et al., 2012; Sherr et al., 2009) and enhance both the anonymity (Mittal et al., 2011; Tang and Goldberg, 2010; Edman and Syverson, 2009) and performance (AlSabah et al., 2012, 2013; Geddes et al., 2014) of Tor.

Any entity that wishes to utilize the Tor network has the option to participate in the network as a relay, which enhances their anonymity. To create a tunnel, the entity initially obtains a list of onion routers in the network from each authority server. By comparing these lists (Ren and Wu, 2010), the entity considers the list that is agreed upon by the majority as the primary list. In the default and standard mode of operation, Tor employs tunnels with a length of 3.

As depicted in Fig. 2, the initial relay in the tunnel is referred to as the Entry Guard. This relay is chosen randomly from a list of three relays and remains unchanged for several months. The final relay in the tunnel is known as the Exit Relay, which is also selected randomly from all the onion routers that meet the desired exit policy (McCoy et al., 2008). The Middle Relay, positioned between the Entry Guard and Exit Relay, is chosen randomly from all the onion routers present in the network.

To transmit a message in Tor, the sender constructs the message using the onion structure and sends it through the tunnel. Tor utilizes *Build* and *Extend* packets to establish a tunnel, gradually extending it in a telescopic manner. However, these patterns can be exploited by attackers to differentiate between tunnel creation messages and data messages.

I2P, introduced in 2003 (Anon., 2022), is a low-latency anonymity network that differs from Tor in terms of its tunneling mechanism. In I2P, the traffic sent and received by a sender is exchanged through separate, one-way tunnels. Unlike Tor, which uses two-way tunnels for message exchange, I2P's structure is message-oriented. However, compared to Tor, the evaluation and research on I2P's structure have been relatively less explored (Schimmer, 2009; Timpanaro et al., 2012, 2011).

In I2P, the network utilizes two types of tunnels: Client tunnels for data transmission and Exploratory tunnels for administrative purposes. Each entity participating in the network creates at least one inbound tunnel to receive messages and at least one outbound tunnel to send messages. To distinguish tunnel-related messages from data messages, a field is included in the header of the messages. This field specifies the packet type. For TunnelBuild packets, the value of this field is either 21 or 23, while for TunnelBuildReply packets, it is either 22 or 24. This allows for the differentiation and separation of tunnel-related packets from regular data messages within the I2P network.

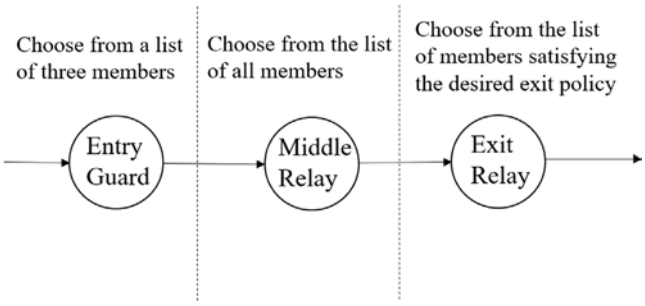

Fig. 2. Abstract view to Tor tunneling scheme.

The messages sent by the sender in the I2P network are encrypted using Garlic encryption. Garlic encryption follows a structure similar to onion cryptography, where a Garlic message contains messages intended for multiple destinations. This allows for efficient and secure transmission of multiple messages within a single encrypted package.

Table 1 categorizes studies related to Tor and I2P, organized by their architectures, issues, and suggested improvements.

In the subsequent sections of the paper, various scenarios will be explored to identify and analyze the problems that arise in tunnel-based structures. Additionally, solutions to address these problems will be proposed. By addressing these challenges, the aim is to enhance the effectiveness and security of tunnel-based anonymity networks.

3. Weaknesses of tunnel-based anonymity structures

Anonymity structures that rely on the creation of multi-use tunnels can be susceptible to attacks that compromise the privacy of communications. These structures face the challenge of being identified by attackers who can track and analyze the patterns of tunnel creation and the participants within each tunnel.

In these types of structures, the tunnel creation process follows identifiable patterns that can be observed by a global attacker. Since the tunnels are used for an extended period, the attacker can track the exchanged packets during tunnel establishment and observe the recurring traffic patterns over time. This enables the attacker to identify the tunnels present in the network and the entities participating in each tunnel. Once the tunnels and their associated entities are identified, the attacker can easily trace any packet injected by a sender and received by a receiver at the end of the tunnel.

This identification of tunnels and entities within them poses a significant threat to the anonymity provided by these structures. It allows the attacker to trace the flow of communications, potentially compromising the privacy and security of the users involved.

Addressing these vulnerabilities is crucial to ensuring the effectiveness of tunnel-based anonymity structures. In the subsequent sections, we will analyze the weaknesses of tunnel-based structures that lead to the identification of the complete path of a packet. We will provide solutions for each of these weaknesses. The examination will focus on four scenarios, namely: 1- Completely separate tunnels 2- Combined tunnels 3- Combined tunnels with simultaneous transmission 4- Combined tunnels with concurrent construction and simultaneous transmission. By delving into these scenarios and providing specific solutions for each weakness, we aim to improve the effectiveness of tunnel-based structures, ultimately enhancing the privacy and security of the communication network.

Table 1
Categorizing the review works.

Aspect	Tor	I2P
Architecture	- Multi-hop onion routing - Entry (guard) nodes, middle relays, and exit nodes	- Garlic routing with multiple messages encapsulated in a single message - Separate inbound/outbound tunnels
Issues	- Separation of network users from relays - Packet type leakage - Increasing the global attacker's ability to detect tunnels	- Higher latency - Packet size and type leakage - Increasing the global attacker's ability to detect tunnels
Proposed structure	Architecture Improvements	- Clustered structure with permanent relays - Dynamic relay selection - Adaptive traffic shaping - Simulation and evaluation show enhanced anonymity

3.1. Completely separate tunnels

The most straightforward scenario in tunnel-based structures, which can easily expose the entire packet path, is the isolation of tunnels from others in the network. In this scenario, when entities within a tunnel do not participate in any other tunnels over time, the transmission of messages within the tunnel reveals the message route through its discernible pattern. For instance, referring to Fig. 3(a), we can consider the tunnels created by entities [A, B, C, D] and [M, N, O, P] that exclusively forward their sender messages along fixed routes A–B–C–D and M–N–O–P. Consequently, the traffic patterns of these two tunnels can be readily identified and distinguished, allowing for the traceability of the sender and receiver when a message is transmitted within these tunnels.

In accordance with this scenario, any tunnel where none of its entities participate in any other tunnels can be easily traced and its communications disclosed. To address this vulnerability, the simplest solution is to employ combined routes during the tunnel construction process. This solution will be further explored in the following.

3.2. Combined tunnels

As previously discussed, if a tunnel exists in which none of its entities are actively present in other tunnels, its traffic pattern can be easily identified through message monitoring. To address this issue, it is necessary for every entity, when creating a tunnel, to ensure that at least one participant in that tunnel is also present in another tunnel within the network. To achieve this, a list of entities currently participating in tunnels can be maintained, and each entity selects at least one entity from this list to include in its own tunnel. The greater the combined participation of entities in different tunnels, the more challenging it becomes for attackers to identify the traffic pattern. For instance, let us consider the scenario depicted in Fig. 3(b). By utilizing the common entity 1 in the two tunnels mentioned earlier, the observed traffic patterns for the attacker would be as follows:

* When a packet is routed through the path A–B–C, the sender of the packet is entity A, and the receiver can be either entity D or entity P, each with a probability of $\frac{1}{2}$.
* When a packet is routed through the path M–N–C, the sender of the packet is entity M, and the receiver can be either entity D or entity P, each with a probability of $\frac{1}{2}$.
* When a packet is routed through the path C–D, the receiver of the packet is entity D, and the sender can be either entity A or entity M, each with a probability of $\frac{1}{2}$.
* when a packet is routed through the path C–P, the receiver of the packet is entity P, and the sender can be either entity A or entity M, each with a probability of $\frac{1}{2}$.

It should be noted that choosing a common entity when creating a tunnel does not completely solve the problem of revealing the traffic pattern. While it helps to mitigate the issue, other factors such as active participation in other tunnels and the position of the common entity within the tunnels are also crucial. These factors play a significant role in enhancing the anonymity and making it more challenging for attackers to identify the traffic patterns accurately. If only one of the tunnels is actively used during a certain period of time and no messages are sent in the other tunnel, the attacker can easily establish a connection between the two ends of the path through the common entity in that tunnel. In the example shown in Fig. 3(c), only tunnel A–B–C–D is active at time period t, allowing the attacker to distinguish the inactive tunnel. To address this issue, a solution involving simultaneous sending along with cover traffic can be employed, which will be further explored and discussed.

3.3. Combined tunnels with simultaneous transmission

To address the issue highlighted in the previous scenario, two methods can be employed, each of which incurs additional overhead for the structure. The first method involves employing continuous active traffic, where even if the senders do not have an actual message to transmit, they send cover and artificial messages within the tunnel to keep the tunnels active. However, this approach leads to a significant increase in network traffic as all links remain active at all times. Alternatively, the second method aims to mitigate the excessive traffic overhead by dividing time into continuous slots. In this approach, senders are only allowed to transmit packets at the beginning of each slot, and it is mandatory for all senders to send packets within each slot. The length of these slots can be carefully chosen to strike a balance between maintaining an acceptable level of network overhead and minimizing the time overhead. As depicted in Fig. 3(d), this method effectively combines traffic across all intervals, resulting in the convergence of traffic along common routes.

Despite the utilization of continuous traffic to maintain active tunnels, there remains the possibility for attackers to identify and separate these tunnels. The attackers can achieve this by leveraging the information acquired from the tunnel creation process. As previously mentioned, structures that rely on tunneling to transmit packets exhibit discernible patterns during the tunnel creation phase, distinguishing these packets from regular data packets. In other words, the packets associated with tunnel creation possess distinct packet structures and header information compared to ordinary data packets. Consequently, the intended entities have the capability to identify these specific message types. For instance, in I2P, the $TunnelBuild$ packets and $TunnelBuildReply$ packets can be recognized during the tunnel creation phase.

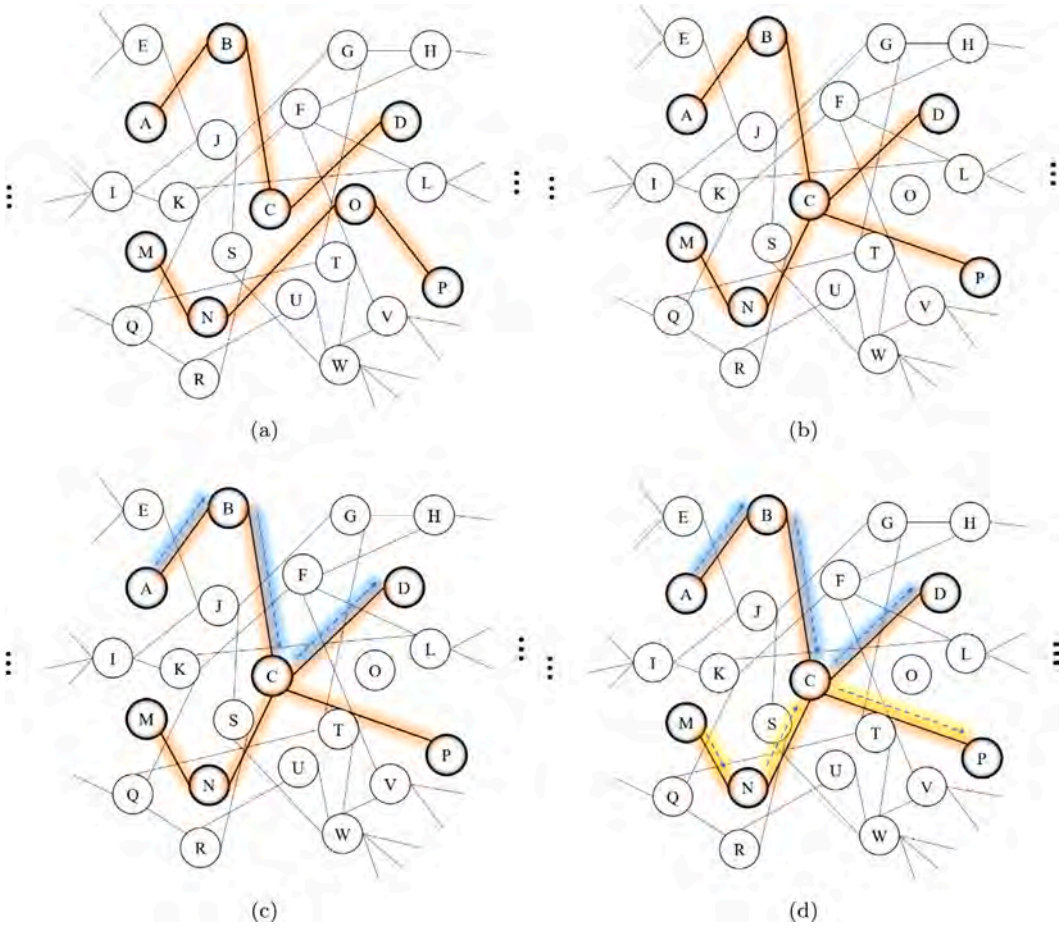

Fig. 3. The impact of different scenarios on tunnel-based anonymity structures: (a) Identifiable traffic pattern in completely separate tunnels (b) Combining the traffic in two tunnels by utilizing the common entity C (c) Separation of tunnels from each other due to the fact that only one of the tunnels is active (d) Simultaneous sending for combining traffic in common routes.

Considering this, if tunnel A–B–C–D is created by entity A at time t_1, and tunnel M–N–C–P is created by entity M at time $t_2 > t_1$, the attacker can examine the paths of these packets and discern the respective tunnels. Consequently, the tunnels become distinguishable from each other. Figs. 4(a) and 4(b) illustrate how the creation of tunnels at different times enables the attacker to identify these tunnels. In these figures, the $T-packet$ denotes the tunnel creation packets. For the sake of simplicity, the discrepancy in packet transmission times between different entities is disregarded.

3.4. Combined tunnels with concurrent construction and simultaneous transmission

To address this issue, it is necessary for the tunnel creation messages to reach the common entity within a very close time frame. Assuming, for simplicity, that the packet transmission time between different entities is the same and the common entity is located in the same position in the tunnels, the tunnel builder process can combine these types of packets at the common entity by sending them simultaneously. This can be achieved by allocating time slots and sending tunnel creation messages at the beginning of each slot. As shown in Fig. 4(c), if both tunnel creation messages are sent at the beginning of the same time slot (assuming the same packet transmission time between different entities), these packets are combined at entity C. As a result, it becomes challenging for the attacker to separate the tunnels.

Another issue arises from the positioning of common entities within tunnels. As depicted in Fig. 4(d), despite sending the tunnel creation packets at the beginning of the same time slots, the arrival time of these packets at entity C differs due to the varying positions of entity C in the two tunnels. Resolving this problem requires ensuring that the common entity occupies the same position in all tunnels, which can be challenging. Alternatively, the common entity could send the received messages simultaneously. However, this approach may introduce significant time overhead for the first received message.

In the subsequent section, a structure will be presented to address the aforementioned vulnerabilities. The proposed structure aims to overcome the issue of traffic separation by employing a combination of common and dedicated routes. The specific details of this proposed structure will be elaborated on in the following section.

4. Proposed structure

The main problems in the scenarios described above can be attributed to two reasons. First, there is a limitation in creating combined tunnels due to the large number of potential relays available. The second reason is the possibility of traffic separation within combined tunnels caused by the different positions of the combined relays within these tunnels. To address these issues, a clustering mechanism has been proposed, where entities can only participate in tunnel creation through permanent relays located at the cluster borders. In a three-hop tunnel, for instance, the first and second relays are selected from the sender's

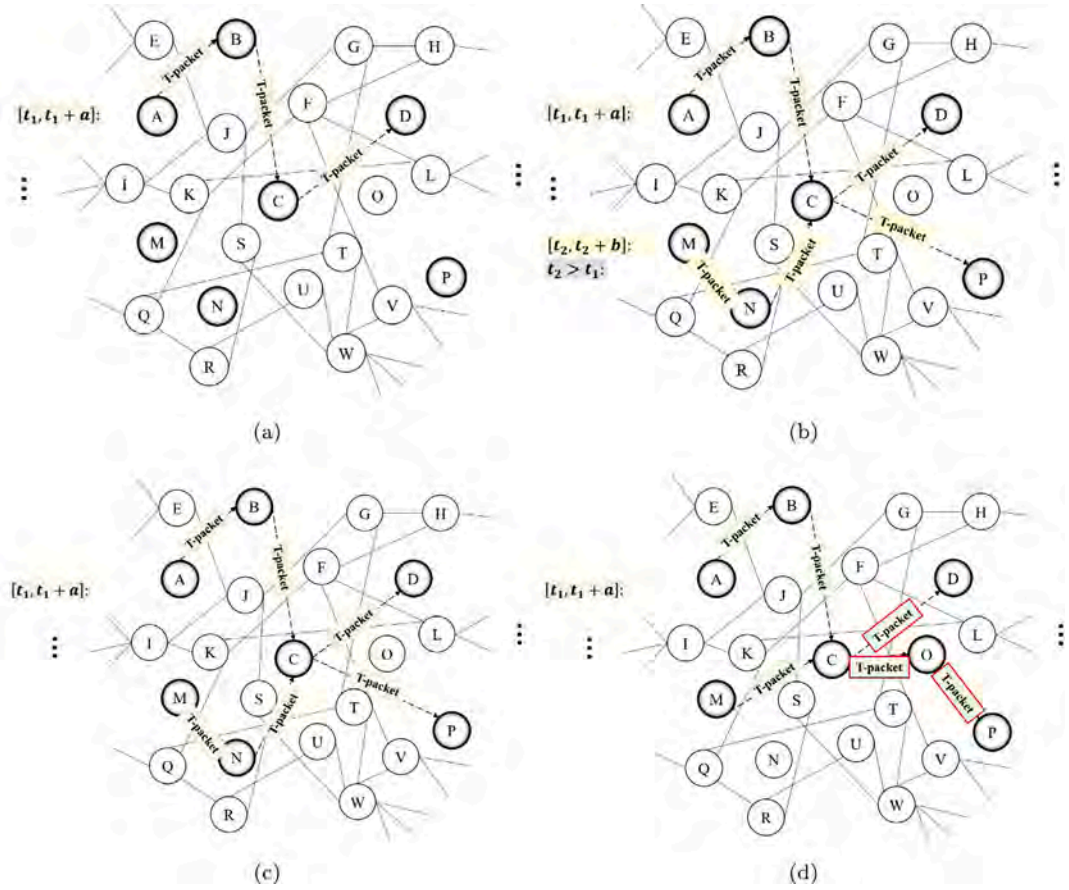

Fig. 4. The impact of different scenarios on combined tunnels: (a) Creation of tunnel A–B–C–D by entity A in time interval $[t_1, t_1 + a]$ (b) Creation of tunnel M–N–O–P by entity M in time interval $[t_2, t_2 + b]$ (c) Simultaneously sending tunnel creation packets in the same time slot to consolidate them in entity C (d) Separation of tunnel creation packets due to the difference in the location of common entity C in two tunnels.

cluster and the middle relay's cluster, respectively. The third relay is the middle relay itself. For example, in Fig. 1, sender A randomly selects the middle relay, entity T, and then selects a permanent relay from its own cluster and another permanent relay from the cluster of entity T. This process results in the creation of the tunnel $A \rightarrow PR_3 \rightarrow PR_9 \rightarrow T$.

Based on the selection of relays, the probability of combining tunnels originating or terminating in the same cluster increases because the second and third relays are chosen from a smaller set of members. In these tunnels, the arrangement of combined relays will be consistent. The steps for creating the structure and integrating entities into the network are explained below. Subsequently, the tunnel construction mechanism by the sender in the network will be described. The notations used in the following are listed in Table 2.

Table 2
Notations used in the proposed structure.

Notation	Description
$pubKey_i$	Self-signed public-key of entity i
$privKey_i$	Private-key j of entity i
VID_i	Virtual-ID of entity i based on its public-key
VID_Mask	Mask-field used by entities to obtain category number from VIDs
$SCID$	Category number of authority servers
S_k	Authority server k
m_l	A message generated by its sender
R_l	A randomly generated number by its sender
PR_i	Permanent Relay i
$key_{src \leftrightarrow PR_{src}}$	A shared-key between src and PR_{src}
SPR_S	Set of all PRs in the clusters
$SPR_{S_{SCID}}$	Set of all PRs in cluster $SCID$
$SVID_S$	set of all $\{VID, pubKey\}$ in the clusters

4.1. Network construction

The creation of the proposed structure involves two steps: registration of permanent relays and joining the network. The following sections explain each step in detail. It is important to note that, for simplicity and to avoid unnecessary complexity, the proposed structure utilizes a single trusted server for accessing network information. However, it is possible to extend the proposed solution to incorporate multiple trusted servers.

In the network, each entity possesses one or more Virtual IDs ($VIDs$), which are derived from their self-signed public keys. Initially, the trusted server determines the number of clusters and the number of permanent relays per cluster. Subsequently, based on this configuration, a VID_Mask is generated by the trusted server. This VID_Mask is then applied to the VID of each entity, resulting in their placement within a specific cluster, determined by their VID.

4.1.1. Registration of permanent relays

Any entity that wishes to become a permanent relay in the network first generates a pair of public and private keys. Using these keys, it calculates its Virtual-ID (VID) based on the following equation. The entity then sends its information to the trusted server. The process of

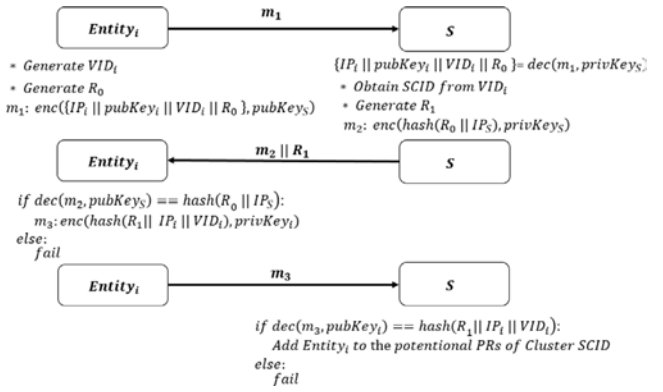

Fig. 5. Communications between an entity and the trusted server for permanent relay registration.

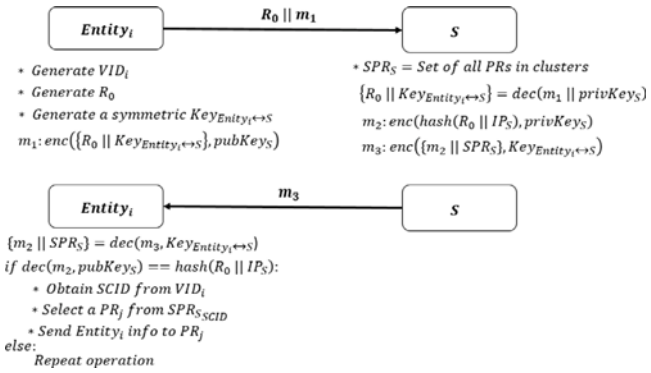

Fig. 6. Obtaining the list of permanent relays in the network from the trusted server.

information exchange is depicted in Fig. 5. In this figure, IP_i represents the IP address of entity i, while IP_S denotes the IP address of the server. Additionally, R_0 and R_1 are random nonces generated by the entity and the server, respectively. Definitions for other symbols can be found in Table 2. Upon receiving the information, the server adds the entity to the list of potential permanent relays.

$$VID_i = hash\left(pubKey_i\right) \tag{1}$$

Finally, the server randomly selects permanent relays from the lists of potential relays, taking into account the configured number of permanent relays for each cluster. This selection process can be repeated at regular intervals to rebuild the structure if necessary. Once the selection is complete, all entities can retrieve the list of selected permanent relays from the server, indicating which relays they should connect to in the network.

4.1.2. Joining the network

To join the network, each entity wishing to participate generates one or more pairs of self-signed public and private keys. Using Eq. (1), it calculates a unique VID for each public key. The entity then determines its cluster based on Eq. (2) and randomly selects one of the permanent relays belonging to that cluster. The entity proceeds by sending its VID information to the selected permanent relay.

$$SCID_i = VID_i \ \& \ VID\_Mask \tag{2}$$

Upon successful completion of this process, the selected permanent relay forwards the entity's information to all other permanent relays within its cluster. Additionally, the permanent relay sends the entity's VID and public key to the trusted server for inclusion in the network's list of VIDs. The entity establishes a shared symmetric key with each permanent relay within its cluster and regularly updates these keys at different time intervals. The message exchange during this step is illustrated in Figs. 6 and 7. In these figures, R_0 and R_1 are random nonces generated by entity i and the server, respectively. Additionally, IP_{PR_j} indicates the IP address of the permanent relay j, while IP_S represents the IP address of the server. SPR_S denotes the set of all permanent relays in the clusters, and $SPR_{S_{SCID}}$ represents the set of all permanent relays in cluster $SCID$. Definitions for other symbols can be found in Table 2.

4.2. Communication scheme in the proposed structure

To initiate communication or create a tunnel, the entity needs to obtain the list of available VIDs in the network from the trusted server.

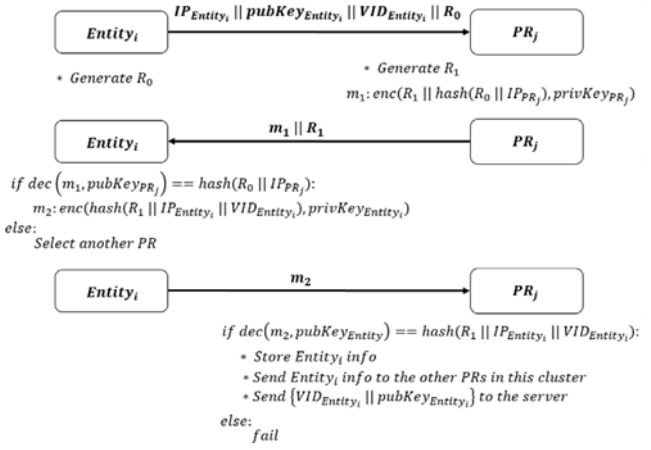

Fig. 7. Sending the VID information to the selected permanent relay by the entity.

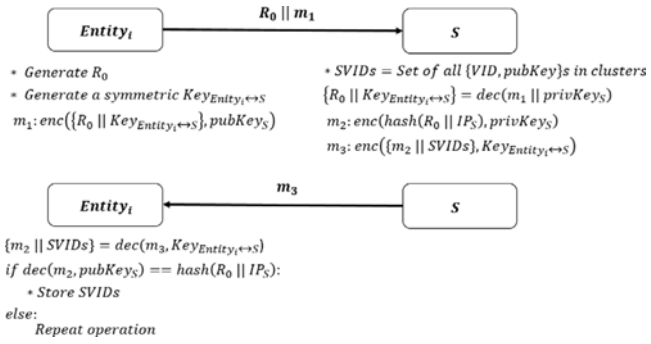

Fig. 8. Obtaining the list of available VIDs in the network from the trusted server.

Once the list is acquired, the entity selects a VID from the available options based on Eq. (2). It then identifies the corresponding cluster associated with this VID. To establish a tunnel, the entity selects one permanent relay randomly from its own cluster. To complete the list of three entities needed for tunnel creation, the entity also selects another permanent relay randomly from the cluster associated with the selected VID.

After selecting the entities participating in the tunnel, the entity encrypts messages containing the symmetric key and communication ID using onion layer encryption. Each layer is encrypted with the public key of the corresponding participating entity. These encrypted

Table 3
Parameters of simulations related to the examined scenarios.

Parameter	Value	
N	Sim-#1	[10, 20, 30, 40, 50, 60, 70, 80, 90, 100]
	Sim-#2	[10, 20, 30, 40, 50, 60, 70, 80, 90, 100]
	Sim-#3	[10, 20, 30, 40, 50, 60, 70, 80, 90, 100]
	Sim-#4	[10, 20, 30, 40, 50, 60, 70, 80, 90, 100]
	Sim-#5	50
Tunnel length	3	
Number of test iterations	100	
Relay selection	Sim-#1	Random
	Sim-#2	Restricted-Random
	Sim-#3	Restricted-Random
	Sim-#4	Restricted-Random
	Sim-#5	Restricted-Random
Active tunnels rate	Sim-#1	–
	Sim-#2	–
	Sim-#3	[0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1]
	Sim-#4	0.5
	Sim-#5	[0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1]
Concurrent tunnel-build rate	Sim-#1	–
	Sim-#2	–
	Sim-#3	–
	Sim-#4	[0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1]
	Sim-#5	[0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1]

messages are then sent to the selected entities. Once the messages are received, the tunnel is successfully created and can be used by the entity for communication. The entity can confirm the successful creation of the tunnel by sending a message through it. The steps involved in this process are depicted in Figs. 8 and 9(a). Also, the steps of sending messages to the final receivers through the tunnel are shown in Fig. 9(b). In these figures, R_0 is the random nonce generated by entity i, IP_S represents the IP address of the server, $SVID_S$ denotes the set of all VIDs and their related public keys in the clusters, m_{plain} is the basic plain message provided by the sender and $tunnel_{id}$ represents the randomly chosen ID for the tunnel. $IntermediateRelay$ is a relay randomly selected from the list of VIDs and $Receiver$ is the final endpoint. Definitions for other symbols can be found in Table 2. In the following section, we evaluate and compare the proposed structure with the previous structures to assess its effectiveness.

5. Simulation, security analysis and comparison

The primary goal of the simulations is to assess the attacker's capability to enhance his understanding of the network tunnels. Consequently, key prerequisites of these simulations included randomizing parameters influencing this assessment. These parameters encompassed the creation of tunnels within each time interval, the involvement of nodes in these tunnels, and the active tunnels during each interval. These simulations have been implemented using NS3 (Riley and Henderson, 2010). Additionally, for more accurate calculation of the average degree of anonymity, the Panda library (McKinney, 2011) and the Python programming language have been utilized (Privacy-Helper, 2024).

Effective factors influencing anonymity are examined based on the previously discussed scenarios. To conduct a more precise evaluation, we have utilized the Shannon entropy (Díaz et al., 2003) to calculate the degree of anonymity. Shannon entropy, rooted in information theory, offers a robust measure of uncertainty and randomness within a system, making it ideal for evaluating anonymity networks. Several factors make Shannon entropy particularly suitable for this purpose:

1. *Quantitative Measure of Uncertainty:* Shannon entropy offers a precise quantitative measure of the uncertainty in identifying users within the network. This approach enables accurate evaluation and comparison of different anonymity mechanisms.
2. *Scalability:* Entropy scales effectively with the size of the anonymity set. As the number of potential senders or receivers increases, entropy naturally rises, indicating a higher level of anonymity
3. *Robustness to Different Scenarios:* Shannon entropy is capable of assessing anonymity across diverse scenarios, including varying network sizes, traffic patterns, and adversary models. Its versatility makes it suitable for evaluating a wide range of anonymity network designs.

Shannon entropy is particularly effective for measuring anonymity because it captures the probabilistic nature of traffic analysis attacks. Unlike other distance metrics such as Hamming distance (Hamming, 1950) and Kullback–Leibler (KL) divergence (Kullback and Leibler, 1951), Shannon entropy is more suited for this purpose. KL divergence typically compares different models rather than measuring the inherent uncertainty within a single distribution, while Hamming distance mainly evaluates differences between individual observations rather than overall uncertainty, which is essential for assessing anonymity.

Assuming there are N active senders within the network during the specified time period, the entropy $H(X)$ for each message X transmitted in the network during that period can be determined using the following equation:

$$H(X) = \sum_{i=1}^{N} -p_i \log_2 (p_i) \tag{3}$$

According to Eq. (3), the maximum possible value of entropy, denoted as H_M, in the given system can be calculated as follows:

$$H_M = \log_2 (N) \tag{4}$$

Now, the degree of anonymity can be determined based on the normalized entropy. The highest degree of anonymity is achieved when all senders have an equal probability of being the source of the message. The lowest value is obtained when a specific sender has a probability of 1 of being the source of the message:

$$d = 1 - \frac{H_M - H(X)}{H_M} = \frac{H(X)}{H_M}, \quad 0 \le d \le 1 \tag{5}$$

To evaluate the proposed structures, a single message is sent by each sender in the network within the same time interval. The degree of anonymity is then calculated for each message. Finally, the average degree of anonymity (d_a) for the network is obtained by taking the average of the degree of anonymity of all messages. This value is calculated using the following equation:

$$d_a = \frac{\sum_{i=1}^{N} d_i}{N} \tag{6}$$

In the following simulations, the scenarios mentioned earlier are examined, focusing on the average degree of anonymity (d_a). The proposed structure is then evaluated based on this criterion to determine its effectiveness in maintaining anonymity. Furthermore, a comparison is made between the proposed structure, Tor, and I2P. The simulations are conducted using the parameters shown in Table 3. In the table, the term "Restricted-Random" denotes a selection process that includes restrictions to enhance the likelihood of combining traffic. The "Active tunnels rate" refers to the ratio of active tunnels to all tunnels within a specific time period, while the "concurrent tunnel-build rate" represents the ratio of created tunnels to all tunnels in a specific time period. The pseudo-code for the simulations is presented in Code 1. Additionally, to better illustrate these steps, a flowchart depicting the process of executing this pseudo-code is presented in Fig. 10. In each iteration, there is only one round for sending messages by active senders. Consequently, with successive iterations, the time sequence of sending in the network is established. The simulations is iterated multiple times, with active senders randomly changing in each iteration.

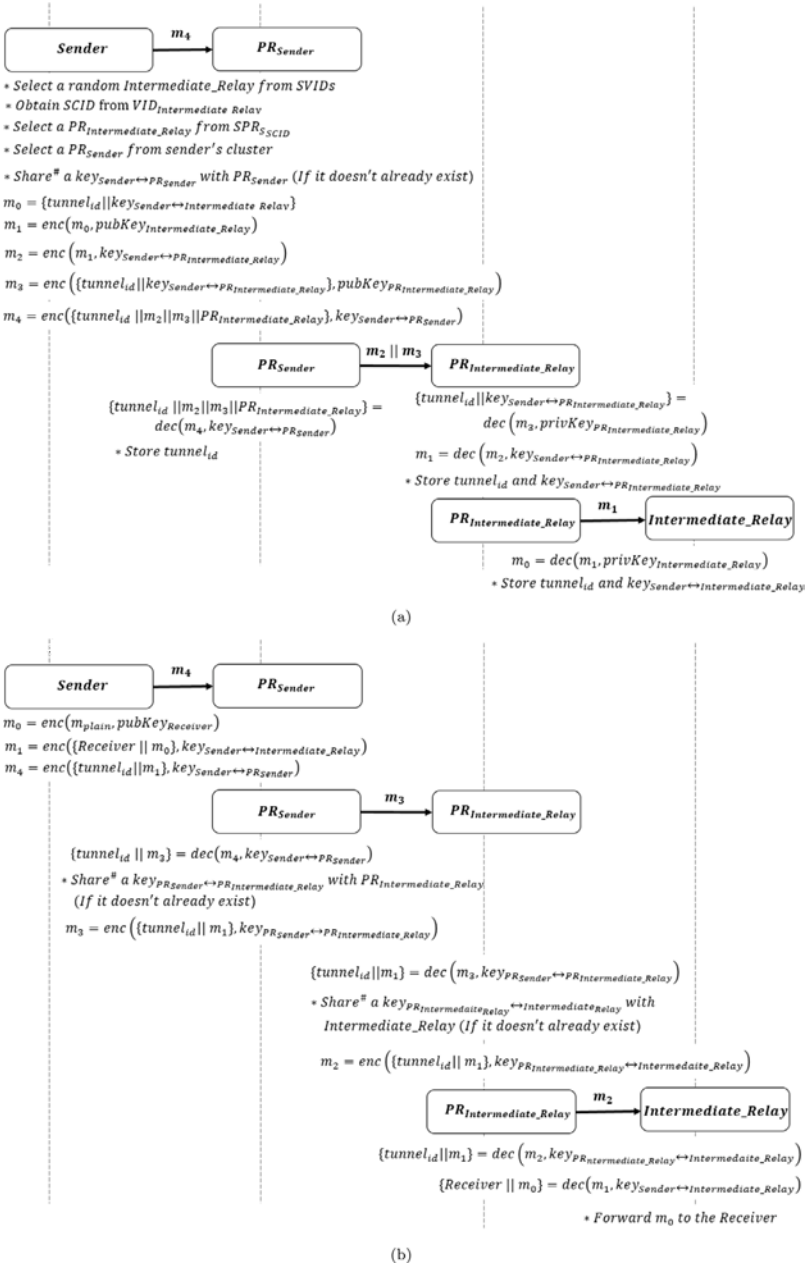

Fig. 9. Communication schem: (a) Creating the tunnel by sharing the symmetric keys generated by the sender. (b) Sending messages to the final receivers through the tunnel. (#-The sharing of this key is accomplished through authenticated Diffie–Hellman.).

When examining the communication between entities as a flow of data characterized by alternating periods of sending and silence, the effectiveness of the global attacker in identifying tunnels is actually heightened. This is attributed to the increased ease with which the attacker can discern traffic patterns in this mode. These patterns emerge due to prolonged durations of traffic flow and the heightened likelihood of adding or subtracting one flow amidst another data stream.

After establishing and utilizing the tunnels within the network, if we view the network as a collection of combined or distinct tunnels, a notable effect emerges consistently across all combined tunnel structures. Consequently, this effect can be regarded independently of the structure and uniformly across the tunnels. Thus, it will exert a consistent impact on the results derived from the simulations. It is worth noting that this environmental condition could be explored further as an appealing extension in future research endeavors. There are several solutions to address this challenge, three of which are outlined below:

1. Cover Traffic: Utilizing cover traffic presents a viable solution, which can be tailored depending on the type of structure in place. For instance, within the proposed structure, it is feasible to determine the initiation or termination times of coverage traffic sent by each entity. This information can be gleaned from feedback provided by the permanent relays to obscure the combined tunnels effectively.

2. Feedback Mechanism: Leveraging feedbacks dispatched from the permanent relays can serve to notify about the depletion of traffic

Pseudo-code for simulations

δ_{pe} represents the ratio of participation, δ_{tb} represents the ratio of potential senders and δ_{tg} represents the ratio of active senders in the network

1. Define the set of all entities as E and its size is represented as n_E
2. Define the set of all participant entities as E_P and its size is represented as $n_{E_P} = n_E * \delta_{pe}$
3. Define a set of random senders $S = \{\forall s_i \mid s_i \epsilon\ E\}$ and its size is represented as $n_S = n_E * \delta_{tb}$
4. Define the type of relay selection algorithm as trs
5. Define a set of the combined relays $CR = \{\}$
6. **For** Variable $i = 1$ to n_S :
7. **If** $trs == random$:
8. Define a set of random relays $R_{s_i} = \{\forall r_j \mid r_j \epsilon\ E_P\ and\ r_j \neq s_i\}$ and its size is represented as $n_{R_{s_i}} = n_t$
9. **Else**:
10. Choose a random number rnd_{s_i} between $[1, n_t]$
11. Define a set of random relays $R_(s_i) = \{(\forall r_j\ and\ j \neq rnd_{s_i} \mid r_j \epsilon\ E_P\ and\ r_j \neq s_i)$ or $(r_{rnd_{s_i}} \mid r_{rnd_{s_i}} \epsilon\ CR\ and\ r_j \neq s_i)\}$ and its size is represented as $n_{R_{s_i}} = n_t$
12. Merge CR and R_{s_i}
13. **End of If**:
14. Build the tunnel t_{s_i}
15. **End of For**
16. Define a set of random active senders $S_A = \{\forall s_k \mid s_k \epsilon\ S\}$ and its size is represented as $n_{S_A} = n_S * \delta_{tg}$
17. Define a set of sent messages $M = \{\ \}$
18. **For** Variable $k = 1$ to n_{S_A} :
19. Send the message m_{s_k} through the tunnel t_{s_k}
20. Add the m_{s_k} to M
21. **End of For**
22. Define the size of M as n_M
23. Define the sum of all calculated degree of anonymity as $DA_{sum} = 0$
24. **For** Variable $l = 1$ to n_M :
25. Calculate the degree of anonymity DA for m_l using Eq. 5
26. $DA_{sum} = DA_{sum} + DA_{m_l}$
27. **End of For**
28. Calculate the average degree of anonymity using Eq. 6 $(N = n_M)$

Code 1: The pseudocode shows a single iteration of the simulation.

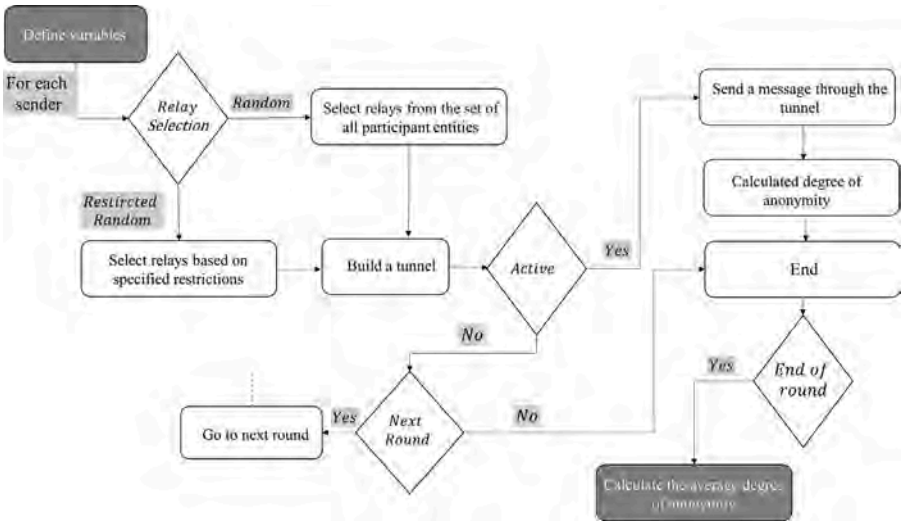

Fig. 10. Flowchart of the simulation steps.

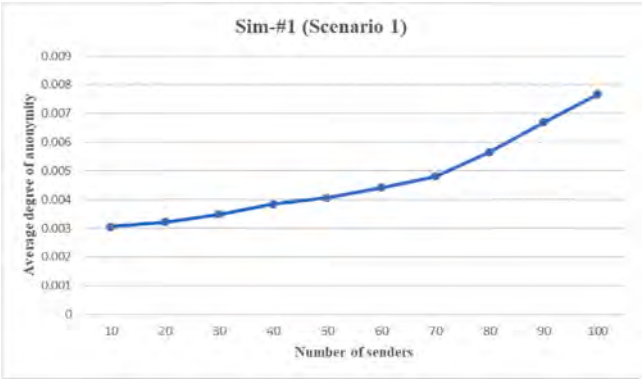

Fig. 11. Effect of increasing the number of tunnels on the average degree of anonymity.

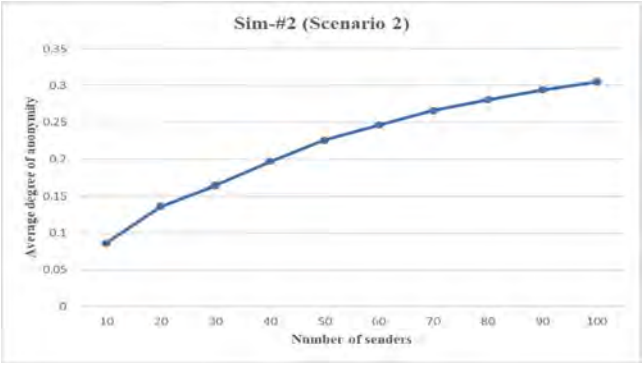

Fig. 12. Effect of Increasing the Number of Tunnels on the Average Degree of Anonymity with Restricted-Random Relay Selection.

within the combined tunnels, prompting a change in the tunnel identified by the attacker. This feedback loop enhances the adaptability of the network in response to evolving threats.

3. Shortened Tunnel Utilization: Another highly effective solution involves reducing the duration of tunnel usage. Ideally, employing disposable tunnels offers optimal security, although a balance must be struck between maximizing tunnel utility and ensuring tunnel validity over time. This approach mitigates the risk associated with prolonged exposure of tunnels to potential attackers.

5.1. Simulation

5.1.1. Sim-#1: Completely separate tunnels

The results of the simulation are presented in Fig. 11, which demonstrates the relationship between the number of tunnels and the average degree of anonymity of messages. As the number of tunnels increases, there is a higher probability of combining traffic, leading to an increase in the average degree of anonymity.

5.1.2. Sim-#2: Combined tunnels

In this scenario, the random selection of relays in the tunnel is constrained by the requirement to choose at least one relay from the list of entities already participating in another tunnel. This restriction imposes limitations on the selection process. The simulation results, as depicted in Fig. 12, demonstrate a notable increase in the average degree of anonymity when this restriction is applied compared to a completely random selection method.

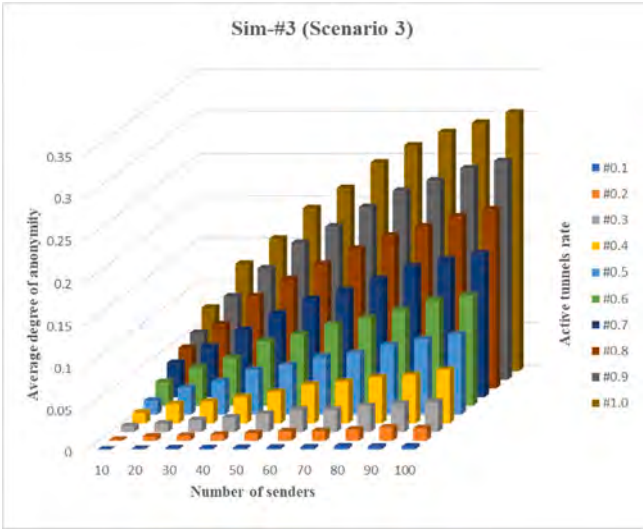

Fig. 13. Effect of active tunnels rate on the average degree of anonymity.

5.1.3. Sim-#3: Combined tunnels with simultaneous transmissions

In this scenario, the impact of the number of active tunnels within a specific time period on the average degree of anonymity has been examined. The active tunnels rate represents the proportion of active tunnels in the desired time period compared to the total number of tunnels in the network. In previous scenarios, this parameter was set to 1, indicating all tunnels were active. The simulation results, presented in Fig. 13, illustrate that increasing the number of tunnels, as well as the number of active tunnels in the network, directly contributes to an enhanced average degree of anonymity throughout the network.

5.1.4. Sim-#4, Sim-#5: Combined tunnels with concurrent construction and simultaneous transmission

In simulation 4, the relationship between the number of simultaneously created tunnels and the average degree of anonymity has been examined. The concurrent tunnel-build rate represents the number of tunnels created within a specific time period compared to the total number of tunnels in the network. In previous scenarios, this parameter was set to 1, indicating all tunnels were created concurrently. The simulation results, depicted in Fig. 14, demonstrate that as more tunnels are created within a time period simultaneously, there is a higher probability of traffic combination, leading to an increased degree of anonymity for the messages. Fig. 15 illustrates the results of Sim-#3 and Sim-#4 under the condition where all tunnels in the network are active, accompanied by a 95% confidence interval.

Simulation 5 examined the relationship between the concurrent tunnel-build rate and the active tunnels rate. The results, presented in Fig. 16, demonstrate a direct correlation between these two factors and the average degree of anonymity.

5.1.5. Sim-#6 - Sim-#10: The proposed structure

The simulations related to the proposed structure utilized the parameters shown in Table 4. The corresponding results are presented in Figs. 17 to 21. Fig. 17 illustrates that when the number of tunnels is approximately equal to $\frac{N}{2}$, the average degree of anonymity does not vary significantly with further increases in the number of tunnels. This suggests that in the proposed structure, if half of the entities have tunnels in the network, a satisfactory average degree of anonymity can be attained in comparison to the maximum achievable value.

Simulation 7 explores the impact of the number of permanent relays on the average degree of anonymity. As depicted in Fig. 18, the average

Table 4
Parameters Used in Simulations for the Proposed Structure.

Parameter	Value	
N	Sim-#6	[10, 20, 30, 40, 50, 60, 70, 80, 90, 100]
	Sim-#7	[10, 20, 30, 40, 50, 60, 70, 80, 90, 100]
	Sim-#8	[10, 20, 30, 40, 50, 60, 70, 80, 90, 100]
	Sim-#9	[10, 20, 30, 40, 50, 60, 70, 80, 90, 100]
	Sim-#10	50
Tunnel length	3	
Number of test iterations	100	
Relay selection	Restricted-Random	
Active tunnels rate	Sim-#6	[0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1]
	Sim-#7	1
	Sim-#8	1
	Sim-#9	1
	Sim-#10	0.5
Number of clusters	Sim-#6	4
	Sim-#7	4
	Sim-#8	4
	Sim-#9	[2, 4, 6, 8, 10, 12, 14, 16, 18, 20]
	Sim-#10	[2, 4, 6, 8, 10, 12, 14, 16, 18, 20]
Number of permanent relays per cluster	Sim-#6	5
	Sim-#7	[5, 10, 15, 20, 25, 30, 35, 40, 45, 50]
	Sim-#8	[1, 2, 3, 4, 5]
	Sim-#9	5
	Sim-#10	[5, 10, 15, 20, 25, 30, 35, 40, 45, 50]

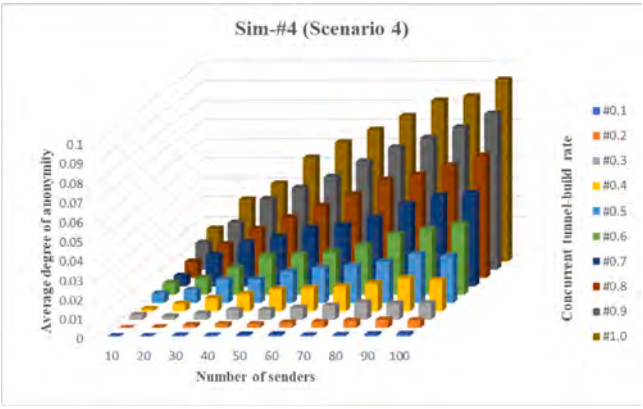

Fig. 14. Effect of concurrent tunnel-build rate on the average degree of anonymity.

degree of anonymity rises as the number of permanent relays decreases. The optimal range for selecting the number of permanent relays lies between 1 and 25, with the maximum value being $\frac{N}{\text{Number of clusters}}$. By conducting a more detailed analysis of simulation 7, as shown in Fig. 19, it can be observed that the average degree of anonymity experiences a more pronounced growth within the interval of [1, 5].

Simulation 9 focuses on examining the influence of the number of clusters on the average degree of anonymity. As illustrated in Fig. 20, the average degree of anonymity rises as the number of clusters decreases. The recommended range for selecting the number of clusters is between 1 and 4, with the maximum value obtained by $\frac{N}{\text{Number of permanent relays per cluster}}$. The results of Sim-#7 and Sim-#9 are demonstrated in Fig. 22, considering the scenario where all tunnels in the network are active. This is complemented by a 95% confidence interval.

Simulation 10 explores the relationship between the number of clusters and the number of permanent relays per cluster in relation to the average degree of anonymity. As depicted in Fig. 21, both parameters exhibit an inverse relationship in relation to the average degree of anonymity. Decreasing the number of clusters or decreasing the number of permanent relays per cluster leads to an improvement in the average degree of anonymity.

5.2. Security analysis

In this section, we will discuss the security aspects of the proposed structure, including server and the permanent relays. We will also cover some advanced attacks aimed at compromising the entity's anonymity and explain the suggested solutions to mitigate these risks. The provided protocols have been evaluated using ProVerif (Blanchet, 2016). The summary of these evaluation results is provided in Appendix. Further evaluation details and complete results can be found in this GitHub project (Privacy-Helper, 2024). As mentioned earlier; for the basic evaluation of protocols security, we consider the following threat model for the attacker:

1. **Global**: The attacker has unrestricted access to the entire anonymity network and exerts full control and oversight over all its links.
2. **Passive**: The attacker avoids any destructive actions that might compromise the integrity of the messages within the anonymity network, focusing solely on monitoring the targeted messages.
3. **External**: The attacker does not participate in the anonymity network or exert control over any of its entities.

In the following sections, we provide a detailed explanation of the security measures for servers and permanent relays in neutralizing attacks and offer solutions to eliminate vulnerabilities. We will then examine the capabilities of more advanced attackers and discuss appropriate countermeasures.

5.2.1. Server

In this paper, the assumption is made that the server is fully trusted, and the attacker cannot compromise the server. However, if the server were to be compromised by an attacker in a hypothetical scenario, the significance of this server and its impact on the network vary depending on the attacker's objective. If the attacker aims to compromise the server and manipulate its behavior within the network, it is crucial to note that remaining passive provides no advantage. The server must be active for the attack to succeed, which aids in detecting compromises and reducing their impact on the network. More details on this topic will be provided in the following section.

If the objective of an attack is to destroy a server and hinder its ability to support the network, it could indeed disrupt ongoing network operations. To address this challenge, employing parallel servers with a

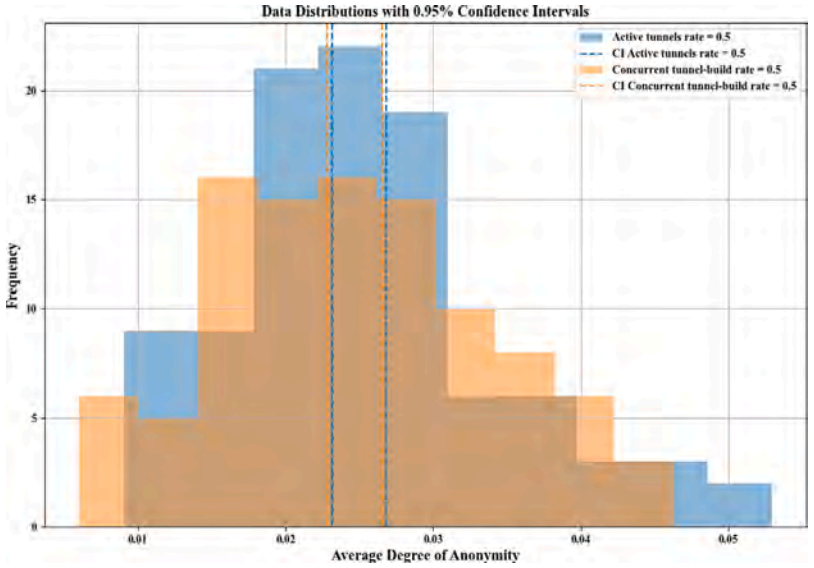

Fig. 15. Effects of active tunnels rate and concurrent tunnel-build rate on the average degree of anonymity (95% confidence interval).

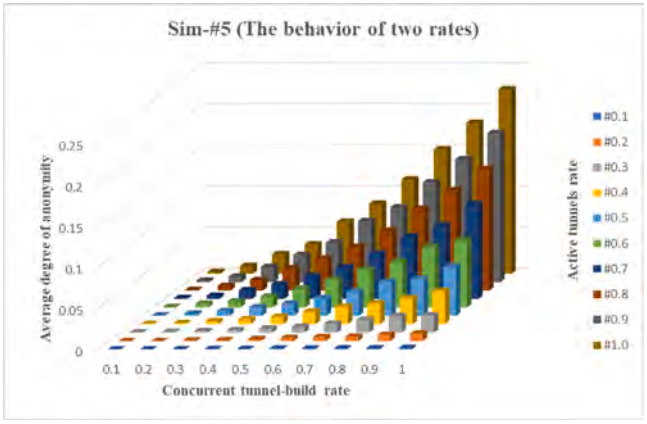

Fig. 16. The relationship between the active tunnels rate and the concurrent tunnel-build rate in relation to the average degree of anonymity.

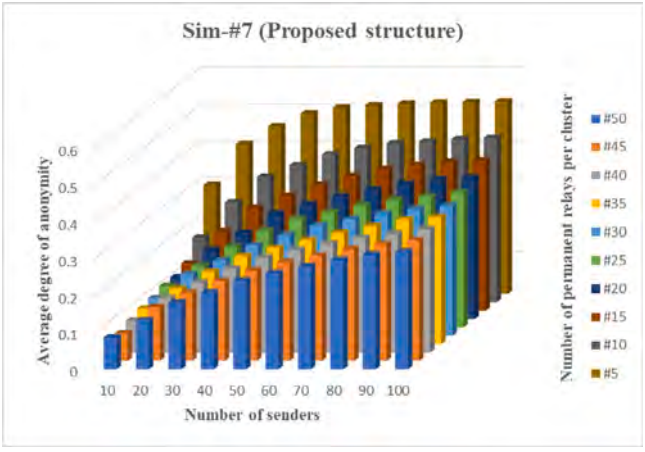

Fig. 18. Impact of reducing the number of permanent relays per cluster on the average degree of anonymity.

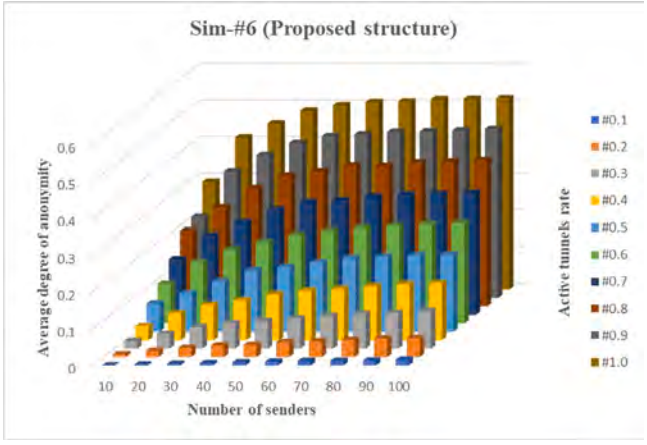

Fig. 17. The effect of increasing the number of active tunnels on the average degree of anonymity in the proposed structure.

majority consensus is a viable solution. The following sections provide a detailed explanation of the proposed solutions to counteract the attacker's objectives.

1. Malicious server

The trusted server in the network manages information about permanent relays and VIDs. It only interacts with entities in the following specific states: registering and receiving information about permanent relays and VIDs. Outside these interactions, the server does not handle data traffic between entities but serves purely as a storage resource.

If an attacker compromises the server, they could access information about permanent relays and VIDs. However, since this information can be legally obtained by other entities, its leakage does not harm the network. The server's main risk is if an attacker uses it to manipulate the lists of permanent relays or VIDs, which could affect entities joining the network.

Thus, if the server does not engage in any destructive actions with the lists it provides, it will not influence or control the behavior of

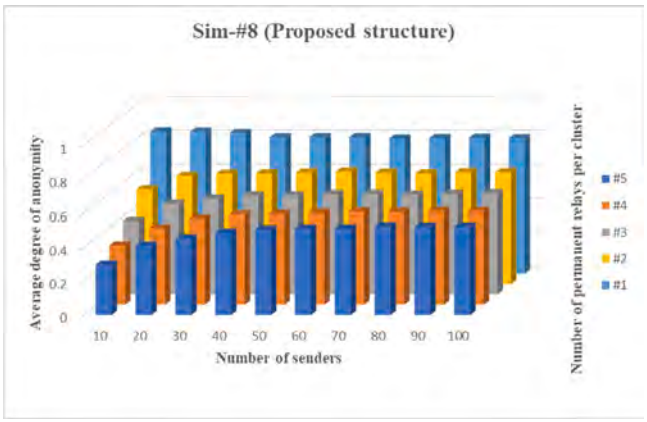

Fig. 19. Examining simulation 7 in more detail.

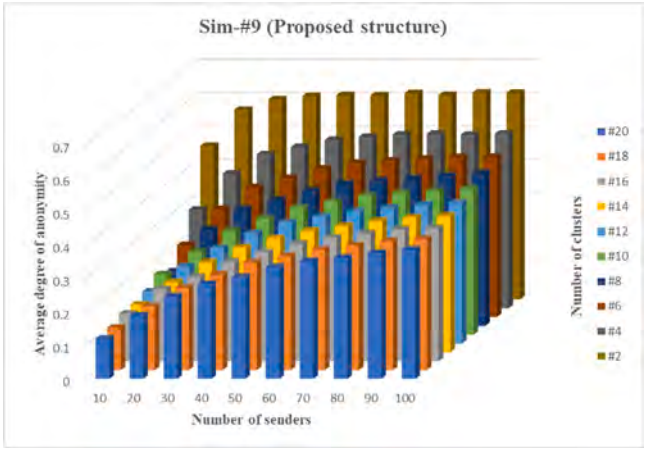

Fig. 20. Impact of reducing the number of clusters on the average degree of anonymity.

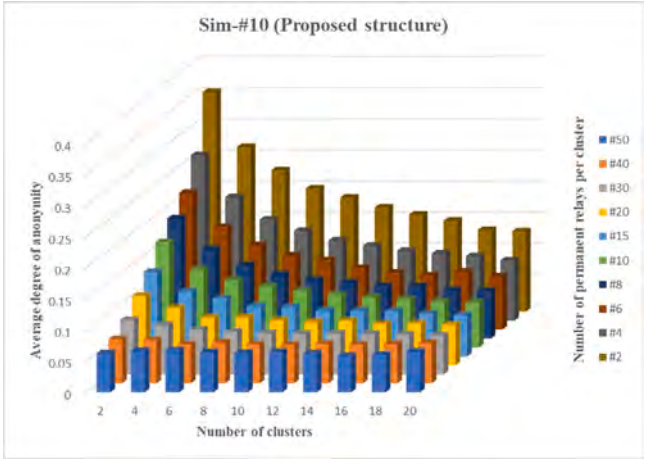

Fig. 21. The relationship between the number of clusters and the number of permanent relays per cluster in relation to the average degree of anonymity.

the entities. Since all information available to the server can be legally and accurately obtained by all entities, including global attackers, the server's passive role will not differ from that of an external global attacker in terms of network traffic analysis. Consequently, merely compromising the server without introducing malicious functionality will not enhance the attacker's capabilities. To ensure trustworthiness, entities and permanent relays periodically test the server's integrity by checking lists and verifying their own inclusion.

If a server compromise is detected, entities can alert the server support team. Once a certain number of warnings, known as the threshold (σ_T), is received, the support team can act to clean or replace the server. The threshold is adjustable based on network sensitivity (S_n) and trust in the reporting entity (T_i). In a highly sensitive network with high trust, this threshold can be very low, triggering immediate investigations upon the first report.

The threshold can be calculated using the following equation, based on the network's sensitivity (S_{Max}) and the degree of trust in an entity (T_i):

$$\sigma_{T_i} = \frac{S_n}{S_{Max}} * \frac{1}{T_i} \tag{7}$$

Each entity in the network is assigned a trust value between 0 and 1, starting at 1 by default. This value can adjust based on report accuracy, either halving or doubling within the permissible range. This system helps diminish the influence of malicious entities and emphasizes the importance of accurate reports from honest entities. To verify reports, the system tracks the number of reports received over a period. If this count exceeds a predefined threshold for the reporting entity, appropriate actions are taken. The report counts resets at the end of each period.

To identify a compromised server, entities evaluate the lists received from the server. These lists include permanent relays and VIDs. Entities can periodically request these lists. If they notice their own removal from the list, it indicates a potential server compromise. Entities can collaborate to identify and respond to malicious behavior by the server.

Entities seeking permanent relay positions must provide their public key and VID. However, when joining the network, entities do not provide this information, preventing the server from linking IP addresses to VIDs. This means the server cannot create targeted lists. Malpractices in the lists are likely detected through periodic requests. Permanent relays can collaborate to compare the server's list with their expected list of relays, allowing them to identify any deliberate omissions.

If a server is compromised but behaves honestly, it may go unnoticed without impacting the network. However, it can still be flagged and replaced if it starts acting maliciously. After corrective measures are taken, the network will be refreshed, allowing new entities to join with updated VIDs.

2. Server Malfunction Attack

Relying on a single central server can present challenges in servicing the network. To address this, a multi-server approach, similar to the directory authorities used in Tor, can be implemented. This involves multiple servers that exchange information regularly to avoid a single point of failure. Each server signs the hash of its list, and entities can compare these signed hashes from different servers to determine the most trusted list. A list is considered valid if it receives a majority of signatures. This setup makes it harder for attackers to compromise the network, as they would need to control more than half of the servers to cause disruption. The multi-server approach involves two main components:

1. *Redundant Servers:*

   * Multiple servers operate simultaneously, each holding duplicate network data. This redundancy ensures network stability even if one server fails or is compromised.

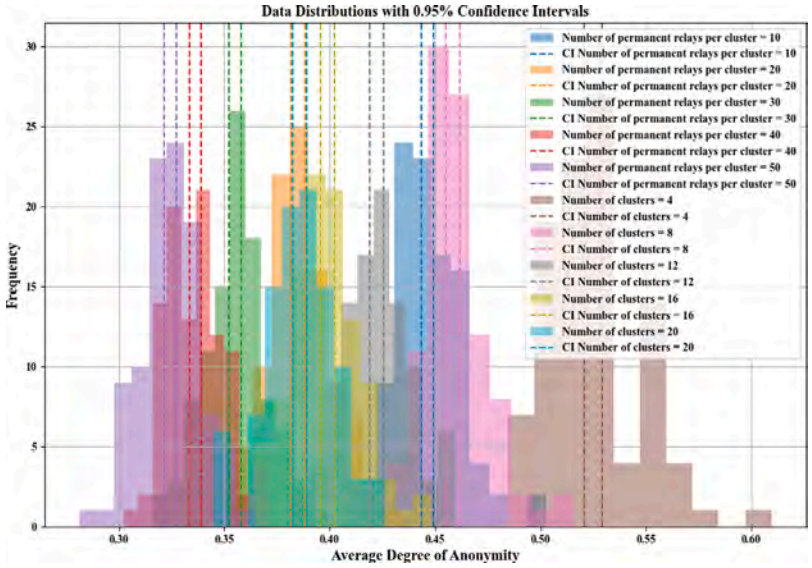

Fig. 22. Impacts of reducing the number of clusters and reducing the number of permanent relays on the average degree of anonymity (95% confidence interval).

* Servers periodically synchronize and validate each other's lists to prevent data manipulation and maintain consistency.

2. *Majority Consensus:*

* Decisions on relay selection and management are based on a majority consensus among servers. A list must receive majority approval to be considered valid.
* This method increases security, as attackers must compromise a majority of servers to affect the network.

Transitioning to a multi-server schema in the proposed structure does not require changes in the communication protocols between entities and servers. Instead, entities need to select a server and compare lists from multiple servers to find the majority-agreed list. Additionally, protocols for server-to-server communication are necessary to synchronize and authenticate lists, similar to those used in Tor.

Tor's directory authorities (Anon., 0000) use a consensus mechanism where each authority independently submits its network status, and the combined consensus requires majority approval. This process ensures that the final list is verified by a majority of authorities.

Several improvements and alternatives to Tor's directory structure have been proposed. Torsk (McLachlan et al., 2009) Uses a Distributed Hash Table (DHT) and buddy selection protocol for better scalability but adds overhead. . In Torsk, Tor's directory authorities take on the role of Neighborhood Authorities, tasked with providing certificates to neighboring nodes in the DHT space when nodes join or leave; a practice appeared in the literature in multiple instances (Haghighi et al., 2021). PIR-Tor (Mittal et al., 2011) Employs Private Information Retrieval (PIR) for enhanced privacy but has high computational and communication costs. ConsenSGX (Sasy and Goldberg, 2019) Uses trusted execution environments like Intel's SGX (Costan and Devadas, 2016) for optimized relay selection but faces adoption challenges.

The I2P operates as a fully decentralized network without central authorities (Anon., 2022). It uses untrusted "netfill" routers and a gossip protocol for updating network details, which can lead to partitioning attacks due to unverifiable information (Zantout and Haraty, 2011).

5.2.2. Permanent relays

Given that the investigation of the structures involves a global adversary of a passive and external nature, evaluations have been conducted under the assumption of non-infection of entities and permanent relays. However, if the attacker possesses the capability to compromise one of the permanent relays, the benefits derived from such an infection are restricted to establishing a relationship between the IPs and VIDs of entities present in the sub-cluster associated with that specific permanent relay. Essentially, the actual identity of the entity at either end of the tunnel becomes exposed to the attacker based on the location of the compromised permanent relay. In the context of Tor, where the attacker is universal and information distribution is limited, this information becomes easily accessible to the adversary.

Since the tunnel path is determined by the sender, the VID associated with the entity at the end of the tunnel is only visible to the permanent relay at the destination. Consequently, if the permanent relay at the source is compromised, only the entity at the beginning of the tunnel is exposed to the attacker. Conversely, if the permanent relay at the destination is compromised, only information pertaining to the end of the tunnel is revealed to the attacker. Therefore, if a permanent relay is compromised, it is not feasible to observe the entire route of the tunnels originating or terminating within the cluster associated with that permanent relay. Consequently, it can be asserted that the compromise of a single permanent relay alone does not pose a significant threat to identifying the tunnels in which that permanent relay participates.

If the attacker manages to compromise multiple permanent relays from different clusters, only the tunnels with both source and destination permanent relays selected from the list of compromised permanent relays will be exposed to the attacker. To delve into the impact of permanent relay compromise in greater detail, it is imperative to first elucidate the enhancement that the proposed structure brings to how tunnels are combined within the network. Subsequently, an examination of how permanent relay compromise affects these improvements becomes feasible. This approach allows for a comprehensive understanding of the effects of permanent relay compromise on the structure and functionality of the network's tunnel system.

In the proposed structure, sharing between one of the permanent relays of the source or destination is sufficient to combine the tunnels.

However, compromising both the permanent relays of the source and destination is necessary to fully identify the path. Given the presence of a global attacker in the network, the individual tunnels are already known to the attacker. Therefore, the advantage of compromising permanent relays lies in identifying the path of the combined tunnels for the attacker.

For simplicity, let us consider a scenario where there are n_C clusters in the proposed structure, each containing n_{PR} permanent relays and the attacker has compromised a permanent relay from each cluster. If we have a total of n entities, we can calculate the probability of combining tunnels Pr_{TC} as follows:

$$\begin{aligned} Pr_{TC} &= \frac{1}{n_{PR}} + \frac{1}{n_{PR}} - (\frac{1}{n_{PR}} * \frac{1}{n_{PR}}) \\ &= \frac{2n_{PR} - 1}{{n_{PR}}^2} \end{aligned} \tag{8}$$

Additionally, the probability Pr_I that one of the combined tunnels will pass through the compromised permanent relays, thereby fully revealing its route, can be calculated as follows. $Pr_{I_{PR_I}}$ represents probability of permanent relay compromise in source side and $Pr_{I_{PR_D}}$ represents probability of permanent relay compromise in destination side.

$$\begin{aligned} Pr_I &= Pr_{TC} * (Pr_{I_{PR_I}} * Pr_{I_{PR_D}}) \\ &= Pr_{TC} * (\frac{1}{n_{PR}} * \frac{1}{n_{PR}}) = \frac{2n_{PR} - 1}{{n_{PR}}^2} * (\frac{1}{n_{PR}})^2 \\ &= \frac{2n_{PR} - 1}{{n_{PR}}^4} \end{aligned} \tag{9}$$

Therefore, as observed, even in the event of permanent relay compromise, the number of combined tunnels in the proposed structure exhibits a significant improvement compared to previous structures. In previous structures, assuming the total number of entities equals n, the probability of combining tunnels would be $(2/n)$. When taking into account the significantly small value of n_{PR} in comparison to n, the probability of the proposed structure calculated from the following equation will be much higher than the probability of the combination in the previous structures.

$$\begin{aligned} Pr_{TC} - Pr_I &= \frac{2n_{PR} - 1}{{n_{PR}}^2} - \frac{2n_{PR} - 1}{{n_{PR}}^4} \\ &= \frac{2{n_{PR}}^3 - {n_{PR}}^2 - 2n_{PR} + 1}{{n_{PR}}^4} \\ &= \frac{2}{n_{PR}} - \frac{1}{{n_{PR}}^2} - \frac{2}{{n_{PR}}^3} + \frac{1}{{n_{PR}}^4} > \frac{2}{n} \end{aligned} \tag{10}$$

5.2.3. Advanced attacks

We are expanding our security analysis to encompass advanced attacks, such as adaptive adversaries and deep packet inspection. These types of attacks represent highly sophisticated attack strategies that present considerable challenges to anonymity networks. By comprehensively understanding and addressing these advanced threats, we aim to bolster the robustness and resilience of our proposed structure.

1. Adaptive Adversaries

Adaptive adversaries can dynamically adjust their strategies based on observed network behavior and responses, unlike static adversaries who use fixed methods. They continuously refine their techniques to exploit vulnerabilities more effectively. Here are our key strategies for mitigating threats posed by adaptive adversaries:

* *Dynamic Relay Selection:* Our proposed structure employs dynamic relay selection to counter adaptive adversaries. Relays are periodically chosen using a randomized algorithm. This approach makes it challenging for adversaries to predict and target specific relays over time, disrupting their ability to conduct long-term surveillance on any particular relay.
* *Frequent Tunnel Re-establishment:* Periodically re-establishing tunnels complicates adversaries' efforts to track traffic effectively. By frequently changing the paths through which data travels, we reduce the likelihood that adversaries can accurately map the network and correlate specific traffic patterns with individual users.
* *Adaptive Traffic Shaping:* Implementing adaptive traffic shaping techniques helps obscure traffic patterns. Standardizing packet sizes and adjusting the frequency of tunnel reconstruction for data packet transmission creates a more uniform traffic flow. This strategy hinders adversaries from identifying and analyzing distinct communication streams effectively.

Traffic analysis attacks pose a significant threat to anonymity networks. These attacks target the inference of communication patterns, participant identification, or user de-anonymization by analyzing traffic characteristics like timing, volume, and packet sizes. In the following, we briefly review some studies conducted to evaluate adaptive attacks and explain the solutions available in the proposed structure to address these attacks. Shahbar and Zincir-Heywood (2018) investigate the effectiveness of flow analysis in identifying encrypted traffic within anonymity networks. They utilize a range of flow-based features and machine learning classifiers to differentiate between various types of encrypted traffic. The study shows that flow analysis can achieve high accuracy in identifying encrypted traffic but acknowledges challenges in accurately classifying traffic amidst sophisticated adversaries. Our research proposes enhanced traffic obfuscation techniques and dynamic relay selection mechanisms to mitigate flow analysis attacks and enhance overall anonymity in such networks. Montieri et al. (2018) focus on classifying traffic originating from popular anonymity services like Tor, I2P, and JonDonym. They employ traffic features and machine learning techniques to differentiate various types of dark web traffic. The study concludes that traffic classification is highly accurate, posing a significant threat to user anonymity. It underscores the necessity for enhanced traffic obfuscation and anonymity mechanisms. In our research, we propose measures such as adaptive traffic shaping through randomized tunnel construction, enhanced tunnel clustering, and standardized packet sizes to bolster resistance against traffic classification attacks.

Tor, as one of the foremost anonymity networks, safeguards users' privacy by directing their communications through a series of relays. Despite its resilient architecture, Tor remains susceptible to a range of active and passive attacks (Alsabah and Goldberg, 2016). Below, we explore these advanced threats and detail how our proposed structure mitigates them to bolster the network's security and anonymity.

- *Traffic Analysis:*

  * *Attack Mechanism:* Adversaries monitor traffic patterns, volumes, and timing to correlate activities at different points in the network. By comparing the incoming and outgoing traffic at relays, they can identify patterns that suggest a relationship between specific senders and receivers.
  * *Countermeasures in Our Approach:*

    - *Traffic Obfuscation:* Our proposed structure implements traffic obfuscation techniques to disrupt traffic analysis. By standardizing packet sizes and varying the timing and frequency of data packets, we create a more uniform traffic flow that is harder to analyze.
    - *Cover Traffic:* Introducing cover traffic, or dummy traffic, helps to obscure actual communication patterns. This additional traffic makes it more challenging for attackers to distinguish between real and dummy traffic, thereby enhancing anonymity.

- *Website Fingerprinting:*

  * *Attack Mechanism:* Adversaries capture and analyze the packet sequences and sizes to identify the websites visited by users. Even encrypted traffic reveals patterns that can be matched against known website fingerprints.
  * *Countermeasures in Our Approach:*

    - *Standardized Packet Sizes:* We standardize the sizes of both tunnel creation and data transmission packets, making it difficult for attackers to derive meaningful fingerprints from packet size alone.
    - *Randomized Tunnel Construction:* By randomizing the relay selection and tunnel construction, we reduce the effectiveness of fingerprinting attacks that rely on specific patterns.

- *Traffic Injection:*

  * *Attack Mechanism:* Adversaries inject identifiable patterns or markers into the traffic at one point in the network and look for these markers at other points to trace the path of the traffic.
  * *Countermeasures in Our Approach:*

    - *Adaptive Relay Selection:* Our dynamic relay selection mechanism ensures that relays are periodically reselected based on a randomized algorithm. This randomness makes it difficult for injected markers to be consistently traced through the network.
    - *Packet Renewal and Encryption:* At each intermediate relay, the previous packet is ignored after processing and the entire message is encrypted and the packet is created from scratch.

Our proposed structure thwarts active and passive attacks through a blend of traffic obfuscation and adaptive relay selection. These measures significantly bolster the network's resilience against sophisticated threats, thereby elevating the level of anonymity and security for users. By continually refining our defense strategies and integrating robust countermeasures, we strive to establish a more secure and dependable anonymity network.

2. Deep Packet Inspection (DPI)

Deep packet inspection (DPI) involves analyzing the data portion of packets as they traverse a network. This capability enables adversaries to conduct thorough analysis and classification of traffic, which could lead to the identification and isolation of anonymity network traffic. To mitigate DPI threats, our approach incorporates the following strategies:

* *Encryption and Padding:* All data packets, including those for tunnel creation and data transmission, undergo encryption to prevent Deep Packet Inspection (DPI) from accessing their contents. Additionally, padding is employed to standardize packet sizes, making it difficult for DPI tools to differentiate between various types of traffic based solely on packet size.
* *Traffic Obfuscation:* Techniques for traffic obfuscation are utilized to obscure the nature of transmitted data. This includes renewing and modifying packet headers and metadata to resemble benign or less suspicious traffic, thereby lowering the likelihood of detection and classification by DPI tools.
* *Use of Cover Traffic:* Cover traffic, also known as dummy traffic, is introduced alongside genuine data transmissions. This increases overall traffic volume and complexity, making it challenging for DPI to separate meaningful communication from noise. Cover traffic is intentionally generated to mimic the characteristics of legitimate traffic, further complicating analysis by adversaries.

Table 5
Parameters of the delay simulation.

Parameter	Value
Number of entities	50
Maximum test iteration	200
Encryption type between the sender and the destination	ElGamal/AES-256
Encryption type between the sender and a relay	AES-256
Encryption type between the sender and a relay in proposed structure	ElGamal/AES-256
Encryption type between the sender and an permanent-relay	AES-256

5.3. Comparison

5.3.1. Message delay

In this simulation, there are 100 entities with random delay intervals ranging from 1 to 100 ms between each other. The parameters of this simulation are listed in Table 5. The tunnel length for Tor is set to 3, and for I2P, the length of the receiver's input tunnel and the sender's output tunnel are both set to 2. In order to ensure a fair comparison, the same encryption methods are used for message encryption in all structures. In scenarios where two nodes lack a shared key, a combination of Elgamal and AES encryption is employed to concurrently send the key and message. Conversely, if two nodes possess a shared key, they utilize this key for encrypting and decrypting the exchanged message. In the proposed structure, given that the number of permanent relays is significantly lower compared to the total entities and there is substantial communication between the entities and the permanent relays, it is reasonable to establish a symmetric shared key between them. Nevertheless, it is not mandatory to have a symmetric shared key for communication either with the relays or the intended destination of the message within the proposed structure. The structures used in the simulation are illustrated in Fig. 23.

To calculate and compare the message sending delay in these structures, the simulations have been conducted using NS3, and the implementation details along with the results are accessible at this git repository (Privacy-Helper, 2024). The message delay obtained from the simulation for each structure is illustrated in Fig. 24. Additionally, the comparison between the results, along with a 95% confidence interval, is depicted in Fig. 25. The simulation results were obtained by varying the maximum allowable delay between two nodes in the network. It can be observed that the proposed architecture introduces a slightly higher delay to the messages compared to Tor and I2P.

5.3.2. Average degree of anonymity

The parameters utilized in the simulation, comparing the proposed structure with the tunneling mechanism, are displayed in Table 6.

In this simulation, the proposed structure has been compared in four different states with varying numbers of permanent relays, while the tunneling mechanism has been compared in three states with different concurrent tunnel-build rates. As depicted in Fig. 26, the proposed structure consistently outperforms the standard tunneling method across all scenarios. The comparison's results are depicted in Fig. 27 where all tunnels in the network are active, accompanied by a 95% confidence interval.

In the next simulation, a comparison has been made between Tor, I2P, and the proposed structure. In Tor, the first relay (entry guard) is selected from a list of only 3 entities for an extended period of time. The members of this list are randomly chosen at the beginning of each interval, and the list is renewed after each interval.

For the sake of simplicity, we have simplified the selection process of an entry guard by considering it from all relays. This increases the probability of sharing tunnels compared to selecting from separate small sets. Additionally, the last relay is selected exclusively from

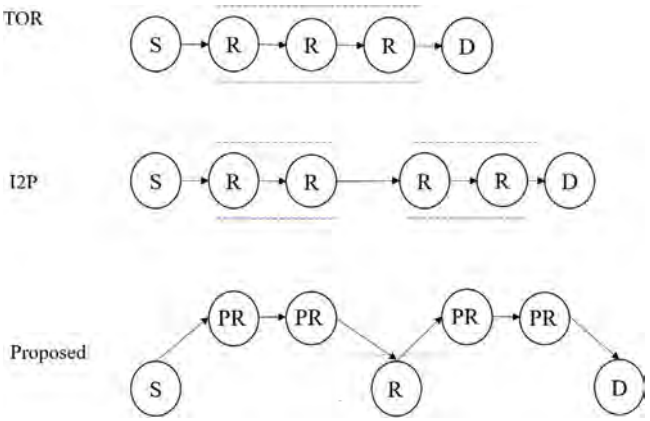

Fig. 23. The Structures Used in the Delay Simulation: R denotes a Relay, S denotes the Sender, D denotes the Destination, and PR denotes a Permanent-Relay.

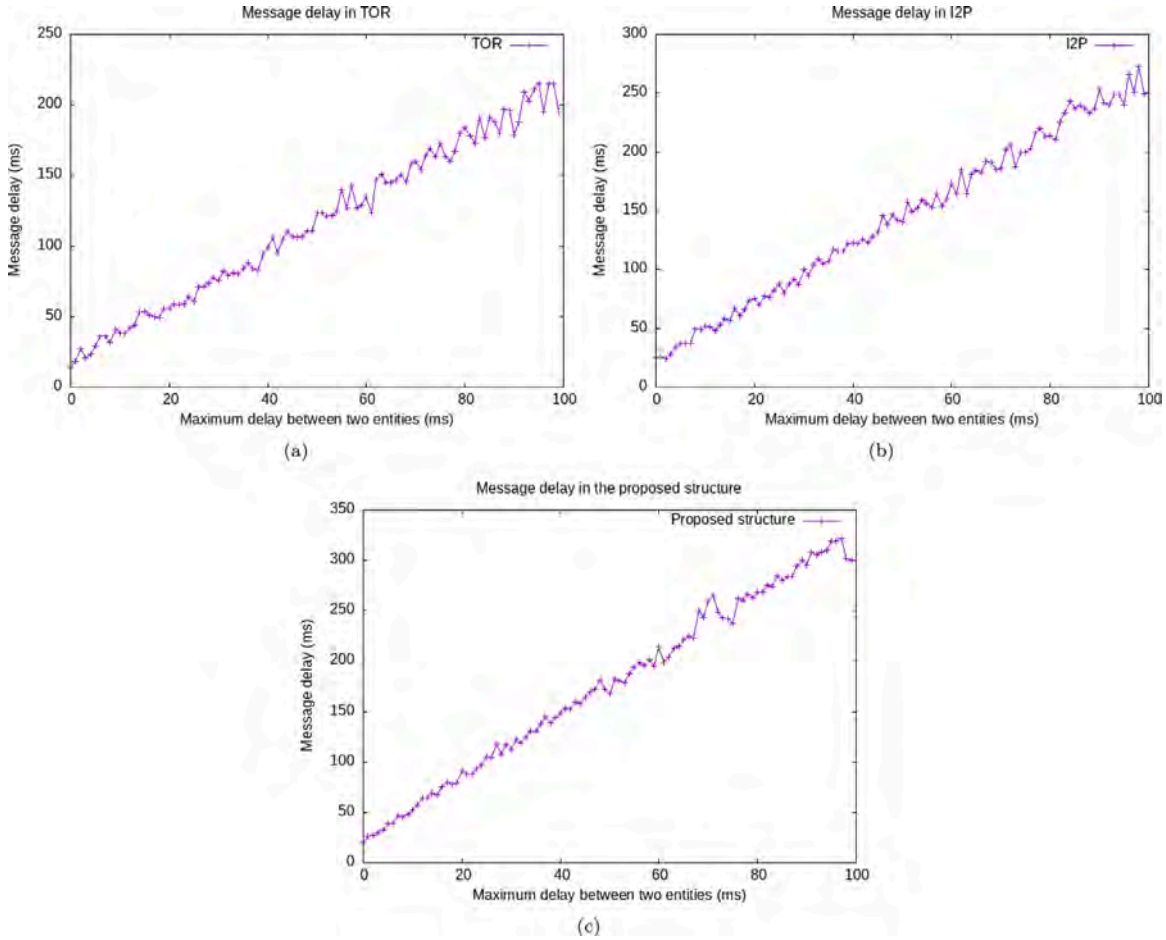

Fig. 24. Message delay in structures based on a different maximum possible delay value between two entities in the network: (a) Message delay in Tor (b) Message delay in I2P (c) Message delay in the proposed structure.

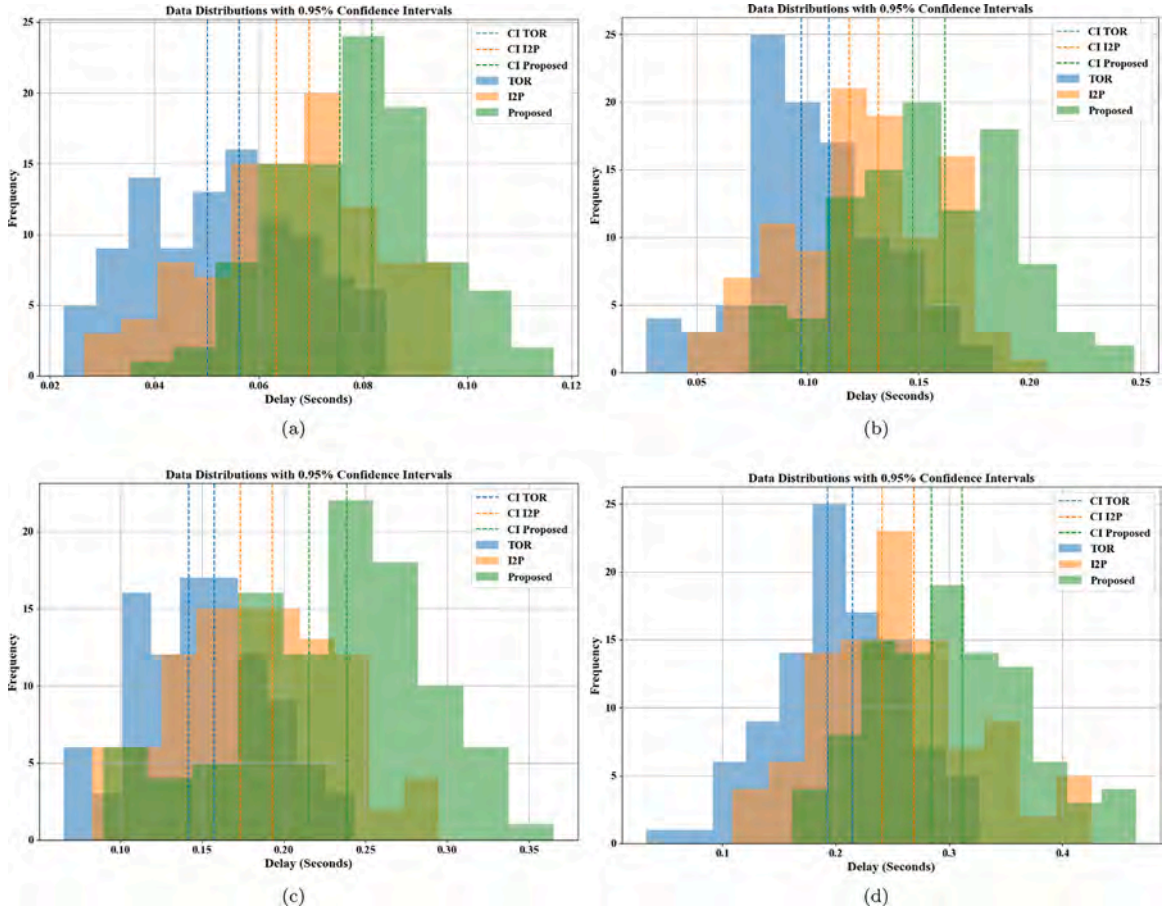

Fig. 25. The comparison of message sending delay: (a) Distributing nodes throughout the network with a maximum delay of 25 ms. (b) Distributing nodes throughout the network with a maximum delay of 50 ms. (c) Distributing nodes throughout the network with a maximum delay of 75 ms. (d) Distributing nodes throughout the network with a maximum delay of 100 ms.

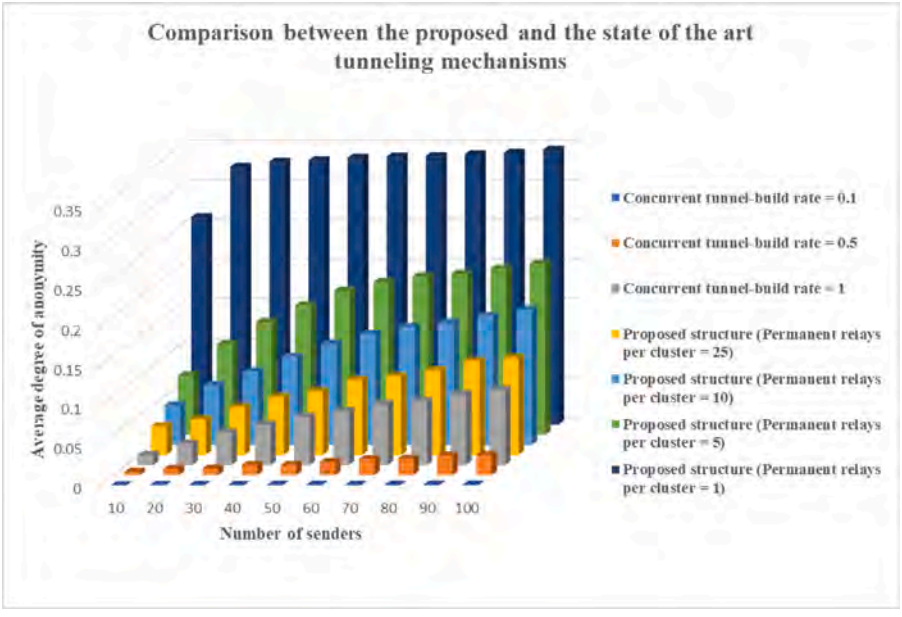

Fig. 26. Comparison between the proposed structure and the tunneling mechanism.

Table 6
Parameters of simulations related to the comparison between the proposed structure and the tunneling mechanism.

Parameter	Value
N	[10, 20, 30, 40, 50, 60, 70, 80, 90, 100]
Tunnel length	3
Number of test iteration	100
Relay selection	Restricted-Random
Active tunnels rate	0.5
Concurrent tunnel-build rate	[0.1, 0.5, 1]
Number of clusters	4
Number of permanent relays per cluster	[1, 5, 10, 25]

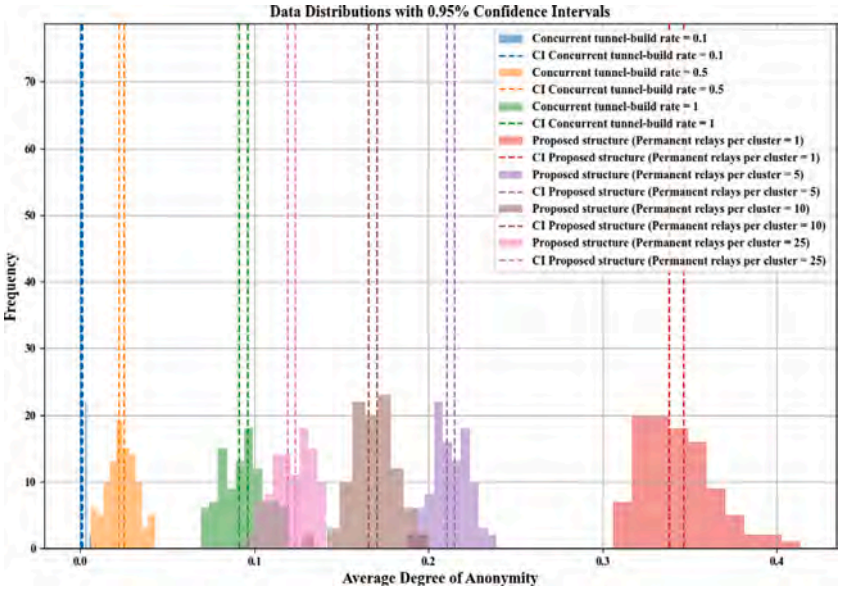

Fig. 27. Comparison between the proposed structure and the tunneling mechanism (95% confidence interval).

Table 7
Simulation parameters for comparing the proposed structure, Tor, and I2P.

Parameter	Value
N	[10, 20, 30, 40, 50, 60, 70, 80, 90, 100]
Tunnel length	3
Number of test iteration	100
Relay selection	Restricted-Random
Active tunnels rate	0.5
Concurrent tunnel-build rate	[0.1, 0.5, 1]
Number of clusters	4
Number of permanent relays per cluster	[1, 5, 10, 25]
Tor exit policy proportions	0.5

entities that offer the requested exit policy. In the simulation, we choose the first two relays restrictively and randomly from all entities, and the last relay is chosen from a subset of entities based on the proportion specified in the Tor exit policy proportions parameter in Table 7.

For simulating I2P, we have considered two tunnels of length 2 for each entity. One tunnel serves as the outbound tunnel for the sender, while the other serves as the inbound tunnel for receiving messages. Initially, the inbound tunnels are inactive. To activate an inbound tunnel, a random selection is made from all available inbound tunnels on the receiver side for each sender. This process ensures that the desired tunnel becomes an active tunnel. The parameters used for this simulation are listed in Table 8.

The results of simulation 12 are depicted in Fig. 28, indicating that the proposed structure outperforms both Tor and I2P in terms of anonymity across all scenarios. The results of this comparison are shown in Fig. 29, where all tunnels in the network are active, along with a 95% confidence interval. A summary of the simulation results for 50 senders is presented in Table 8. In this table, the proposed structure is configured with 4 clusters.

5.4. Discussion on proposal limitations and future work

While our proposed clustered structure with secure routing capabilities notably enhances anonymity and security in network communications, it is crucial to recognize and address its inherent limitations and the challenges linked to its implementation. This section delves into a detailed discussion of these limitations and proposes potential avenues to mitigate them effectively.

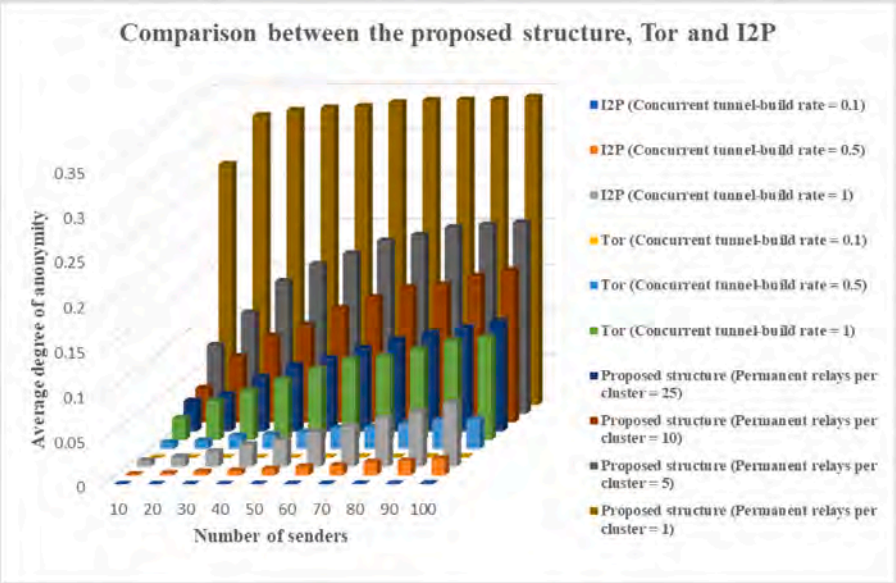

Fig. 28. Comparison between the proposed structure, Tor and I2P.

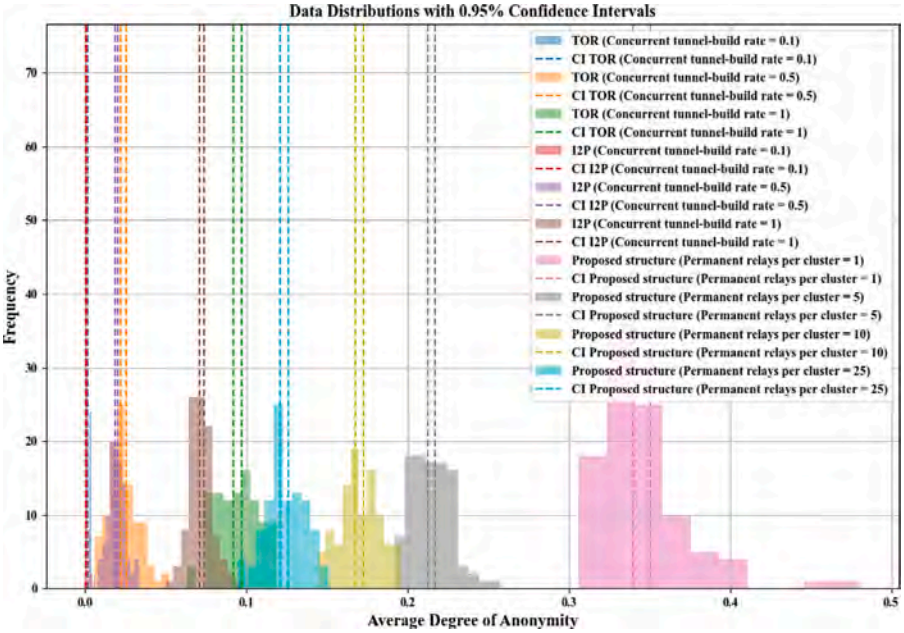

Fig. 29. Comparison between the proposed structure, Tor and I2P (95% confidence interval).

Table 8
Comparison between the proposed structure, Tor, and I2P ($N = 50$). CBT denotes concurrent tunnel-build rate, and #PRs denotes the number of permanent relays.

	Specific parameters		Average degree of anonymity	Delay
I2P	CBT rate	0.1	0.000271	125 ms
	CBT rate	0.5	0.007331	
	CBT rate	1	0.029504	
Tor	CBT rate	0.1	0.000425	103 ms
	CBT rate	0.5	0.01897	
	CBT rate	1	0.079073	
Proposed structure	#PRs per cluster	25	0.081694	154 ms
	#PRs per cluster	10	0.127316	
	#PRs per cluster	5	0.177891	
	#PRs per cluster	1	0.341821	

1. *Centralized Trusted Server:* Our proposal has limitations due to its reliance on a centralized trusted server for information management. To mitigate this risk, we propose a distributed server architecture where multiple servers share the responsibility of relay selection and management. Implementing a majority consensus mechanism among servers can ensure that no single server has undue influence over the network.
2. *Implementation Overhead:* Implementing the proposed clustered structure involves some complexities, particularly in terms of coordinating relay selection and managing clusters.
    - o *Relay Selection Process:* The dynamic and randomized relay selection process adds complexity to network management. To ensure that relays are periodically reselected and synchronized across the network, careful planning and execution are required. Developing automated tools and algorithms for managing relay selection and synchronization can help streamline this process. Additionally, continuous monitoring and adaptive adjustments are essential for maintaining optimal performance.
    - o *Latency Concerns:* The increased number of hops and the dynamic nature of relay selection may lead to higher latency, potentially impacting the usability of real-time applications like video conferencing and online gaming. To address this, optimizing the relay selection algorithm to balance security and performance can help reduce latency. Additionally, implementing adaptive traffic management techniques to prioritize latency-sensitive traffic can enhance the user experience.
3. *Dependence on Permanent Relays:* The reliance on permanent relays within clusters poses risks if their performance degrades. The performance of the network could be affected if permanent relays become overloaded or fail. Load balancing techniques and failover mechanisms can ensure that the network remains resilient and performs well even if some relays are underperforming.

Future studies could concentrate on optimizing the clustering mechanism and exploring network information distribution using multiple servers with a majority consensus mechanism or a Distributed Hash Table (DHT)-based structure. It is also essential to validate the effectiveness of our proposed structure through real-world deployment scenarios. Additionally, improving structural efficiency may involve creating methods to assign performance ratings to network entities, which could influence their selection as permanent relays or relays within tunnels. Furthermore, continuous monitoring and the use of adaptive traffic management techniques to prioritize latency-sensitive traffic can help maintain optimal performance.

6. Conclusion

We propose an architecture aimed at concealing the traffic patterns of entities by implementing combined tunnels within the network. This proposed structure addresses the limitations of the conventional tunnel creation mechanism and offers solutions for improvement. In the conventional approach, an adversary can distinguish individual tunnels from each other based on identifiable tunnel creation messages, enabling them to discern traffic patterns and potentially eliminate the effectiveness of cover traffic. In contrast, our proposed structure leverages clustering and the involvement of permanent relays in tunnel creation to enhance the likelihood of combined traffic, effectively connecting multiple senders to a single message. In this way, the tunnel creation phase becomes indistinguishable from regular data traffic, preventing the attacker from identifying it.

To evaluate and compare the proposed structure with previous structures, we utilized the average degree of anonymity as a metric. The average degree of anonymity was calculated by first determining the Shannon entropy for each message. The resulting entropy values were then scaled to a range of [0, 1] based on the maximum entropy value. The calculated value was subsequently divided by the total number of available messages, resulting in the average degree of anonymity. Through a comparative analysis between the proposed structure, Tor, and I2P, it was observed that the proposed structure consistently exhibited a higher average degree of anonymity compared to these two existing structures. This indicates that the proposed architecture provides enhanced anonymity for network communication when compared to conventional solutions like Tor and I2P. The higher average degree of anonymity in the proposed structure suggests a stronger safeguarding of user privacy.

CRediT authorship contribution statement

Reza Mirzaei: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Resources, Investigation, Conceptualization. **Nasser Yazdani:** Writing – review & editing, Validation, Supervision, Project administration, Methodology, Formal analysis. **Mohammad Sayad Haghighi:** Writing – review & editing, Validation, Supervision, Methodology, Investigation, Formal analysis.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Appendix. Protocol verification

A.1. Registration of permanent relays

```
--------------------------------------------------------------
Verification summary:

Query not attacker(vidI[]) is true.

Query not attacker(ipI[]) is true.

Query event(termEntity(x,y)) ==> event(acceptsServer(x,y)) is true.

Query inj-event(termServer(x)) ==> inj-event(acceptsEntity(x)) is true.

--------------------------------------------------------------
```

A.2. Obtaining the list of permanent relays

```
--------------------------------------------------------------
Verification summary:

Query not attacker(k[]) is true.

Query not attacker(sprS[]) is true.

Query event(termEntity(x)) ==> event(acceptsServer(x)) is true.

--------------------------------------------------------------
```

A.3. Sending the VID information to the permanent relays

```
--------------------------------------------------------------
Verification summary:

Query not attacker(vidI[]) is true.

Query not attacker(ipI[]) is true.

Query event(termEntity(x,y)) ==> event(acceptsPR(x,y)) is true.

Query inj-event(termPR(x)) ==> inj-event(acceptsEntity(x)) is true.

--------------------------------------------------------------
```

A.4. Obtaining the list of available $VIDs$

```
--------------------------------------------------------------
Verification summary:

Query not attacker(k[]) is true.

Query not attacker(svidS[]) is true.

Query event(termEntity(x)) ==> event(acceptsServer(x)) is true.

--------------------------------------------------------------
```

A.5. Creating a tunnel

```
----------------------------------------------------------------
Verification summary:

Query not attacker(kPRI[]) is true.

Query not attacker(kI[]) is true.

Query not attacker(tunnelId[]) is true.

----------------------------------------------------------------
```

A.6. Sending a message

```
----------------------------------------------------------------
Verification summary:

Query not attacker(message[]) is true.

Query not attacker(tunnelId[]) is true.

Query not attacker(receiverId[]) is true.

----------------------------------------------------------------
```

Data availability

No data was used for the research described in the article.

References

Ahmad, K., Kamal, A., 2019. Mix networks: Existing scenarios and future directions on security and privacy. Recent Pat. Eng. 14, http://dx.doi.org/10.2174/1872212114666191223125619.

Al-E'mari, S., Sanjalawe, Y., Fraihat, S., 2023. Detection of obfuscated tor traffic based on bidirectional generative adversarial networks and vision transform. Comput. Secur. (ISSN: 0167-4048) 135, 103512. http://dx.doi.org/10.1016/j.cose.2023.103512.

AlSabah, M., Bauer, K., Elahi, T., Goldberg, I., 2013. The path less travelled: Overcoming tor's bottlenecks with traffic splitting. In: De Cristofaro, E., Wright, M. (Eds.), Privacy Enhancing Technologies. Springer Berlin Heidelberg, Berlin, Heidelberg, ISBN: 978-3-642-39077-7, pp. 143–163.

AlSabah, M., Bauer, K., Goldberg, I., 2012. Enhancing tor's performance using real-time traffic classification. In: Proceedings of the 2012 ACM Conference on Computer and Communications Security. CCS '12, Association for Computing Machinery, New York, NY, USA, ISBN: 9781450316514, pp. 73–84. http://dx.doi.org/10.1145/2382196.2382208.

Alsabah, M., Goldberg, I., 2016. Performance and security improvements for tor: A survey. ACM Comput. Surv. 49, http://dx.doi.org/10.1145/2946802.

Angel, S., Setty, S., 2016. Unobservable communication over fully untrusted infrastructure. In: Proceedings of the 12th USENIX Conference on Operating Systems Design and Implementation. OSDI '16, USENIX Association, USA, ISBN: 9781931971331, pp. 551–569.

Anon., Tor Directory Protocol Specification. URL https://gitweb.torproject.org/torspec.git/tree/dirspec.txt.

Anon., 2022. I2P documentation. URL https://geti2p.net/en/docs. (Online; Accessed 16 August 2022).

Blanchet, B., 2016. Modeling and verifying security protocols with the applied pi calculus and ProVerif. Found. Trends Priv. Secur. (ISSN: 2474-1558) 1 (1–2), 1–135. http://dx.doi.org/10.1561/3300000004, URL http://dx.doi.org/10.1561/3300000004.

Chaum, D.L., 1981. Untraceable electronic mail, return addresses, and digital pseudonyms. Commun. ACM (ISSN: 0001-0782) 24 (2), 84–90. http://dx.doi.org/10.1145/358549.358563.

Chaum, D., 1988. The dining cryptographers problem: Unconditional sender and recipient untraceability. J. Cryptology (ISSN: 0933-2790) 1 (1), 65–75.

Costan, V., Devadas, S., 2016. Intel SGX explained. IACR Cryptol. ePrint Arch. 86, URL http://eprint.iacr.org/2016/086.

Danezis, G., Dingledine, R., Mathewson, N., 2003. Mixminion: Design of a type III anonymous remailer protocol. In: 2003 Symposium on Security and Privacy. pp. 2–15. http://dx.doi.org/10.1109/SECPRI.2003.1199323.

Díaz, C., Seys, S., Claessens, J., Preneel, B., 2003. Towards measuring anonymity. In: Dingledine, R., Syverson, P. (Eds.), Privacy Enhancing Technologies. Springer Berlin Heidelberg, Berlin, Heidelberg, ISBN: 978-3-540-36467-2, pp. 54–68.

Dingledine, R., Mathewson, N., Syverson, P., 2004. Tor: The second-generation onion router. In: Proceedings of the 13th Conference on USENIX Security Symposium - Volume 13. SSYM '04, USENIX Association, USA, p. 21.

Edman, M., Syverson, P., 2009. As-awareness in tor path selection. In: Proceedings of the 16th ACM Conference on Computer and Communications Security. CCS '09, Association for Computing Machinery, New York, NY, USA, ISBN: 9781605588940, pp. 380–389. http://dx.doi.org/10.1145/1653662.1653708.

Fleming, C., Zhou, X., Liu, D., Liang, H., 2014. DiffuseNet: A random walk based anonymity network. In: 2014 IEEE International Conference on Signal Processing, Communications and Computing. ICSPCC, pp. 877–881. http://dx.doi.org/10.1109/ICSPCC.2014.6986323.

Geddes, J., Jansen, R., Hopper, N., 2014. IMUX: Managing tor connections from two to infinity, and beyond. In: Proceedings of the 13th Workshop on Privacy in the Electronic Society. WPES '14, Association for Computing Machinery, New York, NY, USA, ISBN: 9781450331487, pp. 181–190. http://dx.doi.org/10.1145/2665943.2665948.

Gegenhuber, G.K., Maier, M., Holzbauer, F., Mayer, W., Merzdovnik, G., Weippl, E., Ullrich, J., 2023. An extended view on measuring tor AS-level adversaries. Comput. Secur. (ISSN: 0167-4048) 132, 103302. http://dx.doi.org/10.1016/j.cose.2023.103302.

Goel, S., Robson, M., Polte, M., Sirer, E.G., 2003. Herbivore: A Scalable and Efficient Protocol for Anonymous Communication. Cornell University, URL https://api.semanticscholar.org/CorpusID:14881715.

Golle, P., Juels, A., 2004. Dining cryptographers revisited. In: Cachin, C., Camenisch, J.L. (Eds.), Advances in Cryptology - EUROCRYPT 2004. Springer Berlin Heidelberg, Berlin, Heidelberg, ISBN: 978-3-540-24676-3, pp. 456–473.

Haghighi, M.S., Ebrahimi, M., Garg, S., Jolfaei, A., 2021. Intelligent trust-based public-key management for IoT by linking edge devices in a fog architecture. IEEE Internet Things J. 8 (16), 12716–12723. http://dx.doi.org/10.1109/JIOT.2020.3027536.

Haghighi, M.S., Mohamedpour, K., 2008. Securing wireless sensor networks against broadcast attacks. In: 2008 International Symposium on Telecommunications. pp. 49–54. http://dx.doi.org/10.1109/ISTEL.2008.4651270.

Haghighi, M.S., Mohamedpour, K., 2010. Neighbor discovery: Security challenges in wireless ad hoc and sensor networks. In: Bouras, C.J. (Ed.), Trends in Telecommunications Technologies. IntechOpen, Rijeka, http://dx.doi.org/10.5772/8469.

Hamming, R., 1950. Error detecting and error correcting codes. Bell Syst. Tech. J. 29, 147–160.

Heurix, J., Zimmermann, P., Neubauer, T., Fenz, S., 2015. A taxonomy for privacy enhancing technologies. Comput. Secur. (ISSN: 0167-4048) 53, 1–17. http://dx.doi.org/10.1016/j.cose.2015.05.002.

Jawaheri, H.A., Sabah, M.A., Boshmaf, Y., Erbad, A., 2020. Deanonymizing tor hidden service users through bitcoin transactions analysis. Comput. Secur. (ISSN: 0167-4048) 89, 101684. http://dx.doi.org/10.1016/j.cose.2019.101684.

Kullback, S., Leibler, R.A., 1951. On information and sufficiency. Ann. Math. Stat. 22 (1), 79–86.

McCoy, D., Bauer, K., Grunwald, D., Kohno, T., Sicker, D., 2008. Shining light in dark places: Understanding the tor network. In: Borisov, N., Goldberg, I. (Eds.), Privacy Enhancing Technologies. Springer Berlin Heidelberg, Berlin, Heidelberg, ISBN: 978-3-540-70630-4, pp. 63–76.

McKinney, W., 2011. Pandas : Powerful Python data analysis toolkit. URL http://pandas.sourceforge.net/.

McLachlan, J., Tran, A., Hopper, N., Kim, Y., 2009. Scalable onion routing with torsk. In: Proceedings of the 16th ACM Conference on Computer and Communications Security. CCS '09, Association for Computing Machinery, New York, NY, USA, ISBN: 9781605588940, pp. 590–599. http://dx.doi.org/10.1145/1653662.1653733.

Mislove, A., Oberoi, G., Post, A., Reis, C., Druschel, P., Wallach, D.S., 2004. AP3: Cooperative, decentralized anonymous communication. In: Proceedings of the 11th Workshop on ACM SIGOPS European Workshop. In: EW 11, Association for Computing Machinery, New York, NY, USA, ISBN: 9781450378079, pp. 30–es. http://dx.doi.org/10.1145/1133572.1133578.

Mittal, P., Olumofin, F., Troncoso, C., Borisov, N., Goldberg, I., 2011. PIR-Tor: Scalable anonymous communication using private information retrieval. In: 20th USENIX Security Symposium, USENIX Security 11. USENIX Association, San Francisco, CA.

Montieri, A., Ciuonzo, D., Aceto, G., Pescapè, A., 2018. Anonymity services tor, I2p, JonDonym: Classifying in the dark (web). IEEE Trans. Dependable Secure Comput. PP, http://dx.doi.org/10.1109/TDSC.2018.2804394.

Muñoz-Gea, J.P., Malgosa-Sanahuja, J., Manzanares-Lopez, P., Sanchez-Aarnoutse, J.C., Garcia-Haro, J., 2008. A low-variance random-walk procedure to provide anonymity in overlay networks. In: Jajodia, S., Lopez, J. (Eds.), Computer Security - ESORICS 2008. Springer Berlin Heidelberg, Berlin, Heidelberg, ISBN: 978-3-540-88313-5, pp. 238–250.

Nambiar, A., Wright, M., 2006. Salsa: A structured approach to large-scale anonymity. In: Proceedings of the 13th ACM Conference on Computer and Communications Security. CCS '06, Association for Computing Machinery, New York, NY, USA, ISBN: 1595935185, pp. 17–26. http://dx.doi.org/10.1145/1180405.1180409.

Panchenko, A., Lanze, F., Engel, T., 2012. Improving performance and anonymity in the tor network. In: 2012 IEEE 31st International Performance Computing and Communications Conference. IPCCC, pp. 1–10. http://dx.doi.org/10.1109/PCCC.2012.6407715.

Panchenko, A., Richter, S., Rache, A., 2009. NISAN: Network information service for anonymization networks. In: Proceedings of the 16th ACM Conference on Computer and Communications Security. CCS '09, Association for Computing Machinery, New York, NY, USA, ISBN: 9781605588940, pp. 141–150. http://dx.doi.org/10.1145/1653662.1653681, URL https://doi.org/10.1145/1653662.1653681.

Pfitzmann, A., Pfitzmann, B., Waidner, M., 1991a. ISDN-mixes: Untraceable communication with very small bandwidth overhead. In: Effelsberg, W., Meuer, H.W., Müller, G. (Eds.), Kommunikation in Verteilten Systemen. Springer Berlin Heidelberg, Berlin, Heidelberg, ISBN: 978-3-642-76462-2, pp. 451–463.

Pfitzmann, A., Pfitzmann, B., Waidner, M., 1991b. ISDN-mixes: Untraceable communication with very small bandwidth overhead. In: Effelsberg, W., Meuer, H.W., Müller, G. (Eds.), Kommunikation in Verteilten Systemen. Springer Berlin Heidelberg, Berlin, Heidelberg, ISBN: 978-3-642-76462-2, pp. 451–463.

Privacy-Helper, 2024. Tunnel-Mechanism: The source codes and simulation results. GitHub repository. GitHub, URL https://github.com/privacy-helper/tunnel-mechanism.

Reiter, M.K., Rubin, A.D., 1998. Crowds: Anonymity for web transactions. ACM Trans. Inf. Syst. Secur. (ISSN: 1094-9224) 1 (1), 66–92. http://dx.doi.org/10.1145/290163.290168.

Ren, J., Wu, J., 2010. Survey on anonymous communications in computer networks. Comput. Commun. (ISSN: 0140-3664) 33 (4), 420–431. http://dx.doi.org/10.1016/j.comcom.2009.11.009.

Riley, G.F., Henderson, T.R., 2010. The ns-3 network simulator. In: Wehrle, K., Güne¸s, M., Gross, J. (Eds.), Modeling and Tools for Network Simulation. Springer Berlin Heidelberg, Berlin, Heidelberg, ISBN: 978-3-642-12331-3, pp. 15–34. http://dx.doi.org/10.1007/978-3-642-12331-3_2, URL https://doi.org/10.1007/978-3-642-12331-3_2.

Safaei Pour, M., Nader, C., Friday, K., Bou-Harb, E., 2023. A comprehensive survey of recent internet measurement techniques for cyber security. Comput. Secur. (ISSN: 0167-4048) 128, 103123. http://dx.doi.org/10.1016/j.cose.2023.103123.

Sampigethaya, K., Poovendran, R., 2007. A survey on mix networks and their secure applications. Proc. IEEE 94, 2142–2181. http://dx.doi.org/10.1109/JPROC.2006.889687.

Sasy, S., Goldberg, I., 2019. ConsenSGX: Scaling anonymous communications networks with trusted execution environments. Proc. Priv. Enhanc. Technol. 2019 (3), 331–349. http://dx.doi.org/10.2478/POPETS-2019-0050.

Sayad Haghighi, M., Aziminejad, Z., 2020. Highly anonymous mobility-tolerant location-based onion routing for VANETs. IEEE Internet Things J. 7 (4), 2582–2590. http://dx.doi.org/10.1109/JIOT.2019.2948315.

Schimmer, L., 2009. Peer profiling and selection in the I2P anonymous network. In: Technische Berichte, Technische Universität Dresden, Germany, pp. 59–70.

SeyedHassani, A., Haghighi, M.S., Khonsari, A., 2019. Bayesian inference of private social network links using prior information and propagated data. J. Parallel Distrib. Comput. (ISSN: 0743-7315) 125, 72–80. http://dx.doi.org/10.1016/j.jpdc.2018.11.003.

Shahbar, K., Zincir-Heywood, A., 2018. How Far Can We Push Flow Analysis to Identify Encrypted Anonymity Network Traffic? pp. 1–6. http://dx.doi.org/10.1109/NOMS.2018.8406156.

Shannon, C.E., 1948. A mathematical theory of communication. Bell Syst. Tech. J. 27 (3), 379–423. http://dx.doi.org/10.1002/j.1538-7305.1948.tb01338.x.

Sherr, M., Blaze, M., Loo, B.T., 2009. Scalable link-based relay selection for anonymous routing. In: Goldberg, I., Atallah, M.J. (Eds.), Privacy Enhancing Technologies. Springer Berlin Heidelberg, Berlin, Heidelberg, ISBN: 978-3-642-03168-7, pp. 73–93.

Snader, R., Borisov, N., 2011. Improving security and performance in the tor network through tunable path selection. IEEE Trans. Dependable Secure Comput. 8 (5), 728–741. http://dx.doi.org/10.1109/TDSC.2010.40.

Tang, C., Goldberg, I., 2010. An improved algorithm for tor circuit scheduling. In: Proceedings of the 17th ACM Conference on Computer and Communications Security. CCS '10, Association for Computing Machinery, New York, NY, USA, ISBN: 9781450302456, pp. 329–339. http://dx.doi.org/10.1145/1866307.1866345.

Timpanaro, J.P., Chrisment, I., Festor, O., 2012. I2p's usage characterization. In: Pescapè, A., Salgarelli, L., Dimitropoulos, X.A. (Eds.), 4th International Workshop on Traffic Monitoring and Analysis. In: Lecture Notes in Computer Science, vol. 7189, Springer, pp. 48–51. http://dx.doi.org/10.1007/978-3-642-28534-9_5.

Timpanaro, J.P., Isabelle, C., Olivier, F., 2011. Monitoring the I2P Network (Ph.D. thesis). Inria.

Waidner, M., Pfitzmann, B., 1990. The dining cryptographers in the disco: Unconditional sender and recipient untraceability with computationally secure serviceability. In: Quisquater, J.-J., Vandewalle, J. (Eds.), Advances in Cryptology — EUROCRYPT '89. Springer Berlin Heidelberg, Berlin, Heidelberg, ISBN: 978-3-540-46885-1, p. 690.

Zantout, B., Haraty, R., 2011. I2P data communication system. In: Proceedings of ICN 2011, the Tenth International Conference on Networks.

Reza Mirzaei received his B.Sc. degree in Information Tecnology from Urmea University of Technology in 2014, and his M.Sc. degree in Cyber Security from the University of Isfahan in 2016. He is currently pursuing the Ph.D. degree in Information Technology at the University of Tehran in Router Lab and ANSLab. His research interests are applied cryptography and privacy in computer networks.

Nasser Yazdani is a full professor at the Department of Electrical and Computer Engineering in Tehran University. He got his B.S. degree in Computer Engineering from Sharif University of Technology, Tehran, Iran. He worked in Iran Telecommunication Research Center (ITRC) as a consultant, researcher and developer for few years. To pursue his education, he entered Case Western Reserve Univ, Cleveland, Ohio, USA, later and graduated as a Ph.D. in Computer Science and Engineering. Then, Prof. Yazdani worked in different companies and research institutes in USA. He joined the ECE Dept. of Univ. of Tehran, Tehran, Iran, in Sep. 2000. Prof. Yazdani then initiated different research projects and Labs in high speed networking and systems. His research interests include networking, packet switching, access methods, distributed systems and database systems.

Mohammad Sayad Haghighi is an Associate Professor at the School of Electrical and Computer Engineering, University of Tehran. Prior to that, he was an Assistant Professor at Iran Telecom Research Center (ITRC). He is a Senior Member of IEEE and a former research fellow of Deakin University and Macquarie University. Dr. Haghighi is also with Swinburne University of Technology. He has worked in senior positions in IT and Telecom industries for a few years and is now directing the Advanced Networking and Security research Laboratory (ANSLab.org).