mathematics

MDPI

Article

TOAR: Toward Resisting AS-Level Adversary Correlation Attacks Optimal Anonymous Routing

Hui Zhao and Xiangmei Song *

School of Computer Science and Communication Engineering, Jiangsu University, Zhenjiang 212013, China; zhaohui@ujs.edu.cn

* Correspondence: jlsxm@ujs.edu.cn

Abstract: The Onion Router (Tor), as the most widely used anonymous network, is vulnerable to traffic correlation attacks by powerful passive adversaries, such as Autonomous Systems (AS). AS-level adversaries increase their chances of executing correlation attacks by manipulating the underlying routing, thereby compromising anonymity. Furthermore, these underlying routing detours in the Tor client's routing inference introduce extra latency. To address this challenge, we propose Toward Resisting AS-level Adversary Correlation Attacks Optimal Anonymous Routing (TOAR). TOAR is a two-stage routing mechanism based on Bayesian optimization within Software Defined Networks (SDN), comprising route search and route forwarding. Specifically, it searches for routes that conform to established policies, avoiding AS that could connect traffic between clients and destinations while maintaining anonymity in the selection of routes that minimize communication costs. To evaluate the anonymity of TOAR, as well as the effectiveness of route searching and the performance of route forwarding, we conduct a detailed analysis and extensive experiments. The analysis and experimental results show that the probability of routing being compromised by correlation attacks is significantly reduced. Compared to classical enumeration-based methods, the success rate of route searching increased by close to 2.5 times, and the forwarding throughput reached 70% of that of the packet transmission. The results show that TOAR effectively improves anonymity while maintaining communication quality, minimizing anonymity loss from AS-level adversaries and reducing high latency from routing detours.

Keywords: anonymous routing; Bayesian optimization; correlation attacks; Software Defined Network

MSC: 68M25

Citation: Zhao, H.; Song, X. TOAR: Toward Resisting AS-Level Adversary Correlation Attacks Optimal Anonymous Routing. *Mathematics* **2024**, *12*, 3640. https://doi.org/10.3390/math12233640

Academic Editor: Cheng-Chi Lee

Received: 18 October 2024
Revised: 15 November 2024
Accepted: 19 November 2024
Published: 21 November 2024

1. Introduction

The Onion Router (Tor) [1] is currently the foremost anonymous communication system. It offers technical anonymity protection for internet users' privacy by concealing the connection between identity information and the IP addresses of communication entities through multi-hop overlay routing. The Tor network comprises 7500 relays and handles nearly terabytes of bandwidth traffic daily, serving more than 5 million users [2]. To ensure low-latency communication, Tor does not use traffic obfuscation techniques, which makes it challenging to defend against correlation attacks. Traffic correlation analysis is a fundamental technique used in a wide range of deanonymization attacks on Tor, leveraging external traffic features as side-channel information [3,4]. Studies have found that autonomous system adversaries are more capable of conducting traffic correlation analysis attacks than previously recognized [5]. Internet routing detours usually generate high latency and increase the likelihood of AS-level adversaries appearing at both ends of the circuit, further raising the risk of correlation attacks [6]. Additionally, there is some research aimed at improving the accuracy of these correlation attacks [7,8]. This trend is

likely to drive the growing use of correlation analysis attacks, which will ultimately reduce Tor's level of anonymity.

A significant amount of prior research has focused on developing AS-aware paths in Tor by modifying the relay selection algorithm [9–12]. These AS-aware path algorithms utilize the geographic locations of clients and relays as a criterion for weighting the relays. However, application-layer solutions are limited to passive inference techniques, leading to inaccurate routing inference and increased latency for Tor clients. A significant portion of the latency overhead can be attributed to the underlying structure of Tor relay connections. For example, Counter-RAPTOR [12] significantly increases download time [13].

To address the above challenges, we propose Toward Resisting AS-level Adversary Correlation Attacks Optimal Anonymous Routing (TOAR). Rather than modifying the relay selection algorithm in the overlay network, TOAR leverages an underlay routing scheme within Software Defined Networks (SDN). TOAR is a two-stage (i.e., route searching and route forwarding) routing mechanism based on Bayesian optimization in SDN networks. Initially, it selects routing nodes that provide strong anonymity and low communication costs, ensuring compliance with established policy guidelines while avoiding AS that could connect traffic between the source and destination. Subsequently, it queries and confirms routing policy information through programmable interfaces designed for cross-SDN domain boundary routing. To assess TOAR's anonymity, along with the effectiveness of route searching and route forwarding performance, we carry out a comprehensive analysis and extensive experiments. The results indicate that TOAR is an effective solution for addressing the reduction in anonymity caused by AS-level attackers while also mitigating the high latency associated with routing detours.

In summary, this paper makes the following contributions:

- We propose a novel two-stage (i.e., route searching and route forwarding) routing mechanism, TOAR, based on Bayesian optimization. It effectively addresses the issue of diminished anonymity in Tor resulting from AS-level correlation attacks and reduces high latency caused by routing detours.
- In TOAR, we design a software-defined programmable interface that facilitates querying routing policy information and confirming routing selections, enabling flexible end-to-end source routing choices.
- We conduct a comprehensive analysis and extensive experiments to assess TOAR's anonymity, along with the effectiveness of route searching and route forwarding performance. The results indicate that TOAR provides stronger anonymity and improves communication performance.

The remainder of this paper is organized as follows. In Section 2, we provide essential background information and discuss the motivation behind our work. In Section 3, the problem of optimal routing is defined. In Section 4, we introduce our optimal anonymous routing scheme named TOAR. In Section 5, we evaluate the performance of TOAR and discuss our experimental results. We summarize the related work in Section 6 and conclude the paper in Section 7.

2. Background and Motivation

The strong pursuit of privacy preservation motivates the emergence of anonymous communication systems. Tor is the most widely used among the deployed anonymous systems.

Latency is a major factor contributing to a positive user experience. Tor aims to provide low-latency anonymous communication, and therefore does not employ traffic obfuscation, making it difficult to resist correlation attacks. Correlation attacks utilize traffic features, such as packet timing, packet sizes, and inter-packet delays, to statistically correlate and link network flows [3]. To perform the correlation attack, an adversary must monitor the traffic entering and exiting the Tor network. An adversary can operate a large number of Tor relays to increase the likelihood of monitoring the connections at both ends of communication [14]. Alternatively, they can take control of Autonomous Systems and

manipulate the underlying network communication to place themselves on the forwarding path of Tor traffic, thereby increasing the likelihood of successfully executing a correlation attack [15–17].

Technically, Tor offers anonymity by establishing tunnels between the client and destination through cascades of onion routing relays. A significant portion of the latency overhead is due to the underlying structure of the connections between Tor relays. The Tor client randomly selects an entry relay, a middle relay, and an exit relay to establish an anonymous channel. The onion routing established by Tor is an overlay anonymous circuit network based on transport layer protocols. Creating onion routes is independent of Internet routing and not constrained by the underlying network topology, and often involves detours. This routing detour causes more than 90% of connections in the Tor network to experience delays exceeding five times that of direct Internet connections [18]. Furthermore, an Autonomous System (AS) may be traversed multiple times. The AS may be on the route between the client and the entry relay, as well as on the route between the exit relay and the destination. In this case, an adversary controlling only one AS may carry out traffic correlation attacks and degrade the anonymity of an anonymous system by monitoring traffic in both the entry and exit routing segments.

Figure 1 shows the scenarios of traffic correlation attacks by a single AS-level adversary, where S is the message source and D is the message destination; the anonymous communication between S and D is carried out using multi-hop encrypted forwarding; and the forwarding relays are R1, R2, and R3. Due to routing detours, an AS controlled by an adversary, AS2, simultaneously appears on the routing segment from the sender S to the entry relay R1, as well as on the routing segment from the exit relay R3 to the destination D. Even though the communication between the user and the entry relay is encrypted, AS2 can still observe the user's IP address by analyzing the headers of the packets sent by the user to the entry relay. Additionally, by analyzing the headers of the packets sent by the exit relay to the destination, AS2 can observe the destination's IP address. Then, through traffic correlation and statistical methods, the adversary at AS2 can infer that S is communicating with D.

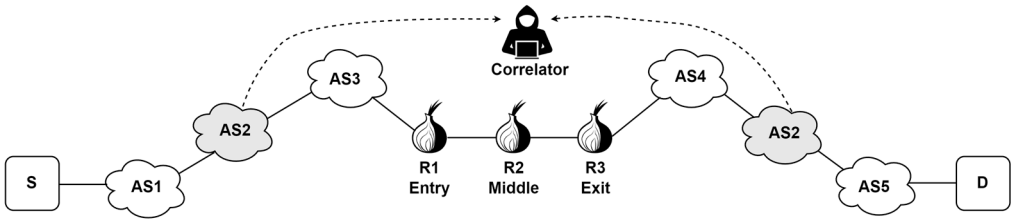

Figure 1. Scenarios of traffic correlation attacks by a single AS-level adversary.

Additionally, Internet routing is asymmetric, meaning the path from the client to the entry relay may differ from the path returning from the entry relay to the client; similarly, the path from the exit relay to the destination (e.g., a web server) may differ from the path returning from the web server to the exit relay. Specifically, an adversary may observe: (1) the data flow from the client to the entry relay and the data flow from the exit relay to the server; (2) the data flow from the client to the entry relay and the TCP acknowledgment traffic from the server to the exit relay; (3) the TCP acknowledgment traffic from the entry relay to the client and the data flow from the exit relay to the server; and (4) the TCP acknowledgment traffic from the entry relay to the client and the TCP acknowledgment traffic from the server to the exit relay. The adversary can examine the TCP headers in the observed traffic to retrieve the TCP sequence number and acknowledgment number fields and analyze the correlation between these fields over time. Thus, the asymmetry of Internet routing allows four segments of the routing paths between the two communication endpoints in Tor to be observed, increasing the chances of an AS-level adversary monitoring the traffic, which may effectively compromise Tor's anonymity.

Traditional AS-level routing generally uses the Border Gateway Protocol (BGP), which has a crucial role in inter-domain connectivity. With BGP, each AS can configure and perform its own policy to select routes, then announce the selected routes to its neighbor AS. Although BGP provides maximum network autonomy, it lacks mechanisms to prevent natural churn and routing attacks such as BGP hijacking and interception. Moreover, BGP does not offer programmable control, making it challenging to provide flexible end-to-end routing.

To address the issue of anonymity degradation due to correlation attacks by AS attackers in the Tor network, striving for an inter-domain routing protocol that balances performance and security—specifically, the efficiency of low-latency anonymous communication and the probability of paths being compromised by attackers—is a substantial breakthrough.

This consideration motivated the construction of TOAR as an underlay optimal routing scheme. This paper studies the problem of finding optimal routes with maximum disjointness and policy satisfaction at the network layer to support Tor application services, specifically addressing the situation where the same adversary appears at both ends of the Tor circuit.

3. System Model

In this section, we first present the model of inter-domain routing and then define the inter-domain optimal routing problem and analyze the complexity of the problem. The notations used and the definitions are given in Table 1.

Table 1. Notations used.

Notations	Definitions
$G(V, E)$	Global network G, a directed graph with V vertices and E edges.
S, D	Sender and destination.
$\|sym\|$	Route length, sym can be any symbol representing a route.
$\{R_i\}$	Overlay anonymous circuit $AC = \{R_i \| 1 \le i \le l\}, l = \|AC\|$.
r_i^f, r_j^b	Forward routing and backward routing $1 \le i \le l+1, 1 \le j \le l+1$.
$\left\{v_i^{dir}\right\}$	Simplified inter-routing $r = \left\{v_i^{dir} \middle\| 1 \le i \le \eta^{dir}\right\}$, $\eta = \|r\|$; dir is direction.
p	The probability that each node in the global network G is attacked.
k	Number of repetitive nodes in r.
k_{max}	Maximum number of shared nodes selected by the user.
m	Total number of nodes in routing segments r_1^f and r_{l+1}^b.
n	Total number of nodes in routing segments r_{l+1}^f and r_1^b.
$d(k)$	Simplified representation of $d(p, k, m, n)$; the degree of path safety.
$e(r)$	Exit policy of r; simplified representation of $e_v(dp, r, u)$.
$c(r)$	Price policy of r.

3.1. Basic Definitions

Using the abstraction that a routing node represents an AS domain, which is widely used in research [19], a routing model is formalized for inter-domain anonymous communication scenarios, as in Figure 2.

Global Network: Use the graph $G = (V, E)$ to represent a global network if a vertex, $v \in V$, denotes that AS_v is in the network, and $edge(u, v) \in E$ denotes that there is an inter-domain session between AS_v and AS_u.

Overlay Circuit: This represents an overlay anonymous circuit using a set of relay $(R, Relay)$ sequences $AC = [R_1, R_2, \ldots, R_l]$, with l denoting the length of the anonymous circuit; for any integer $i, j \in [1, l]$, if $i \neq j$, then $R_i \neq R_j$.

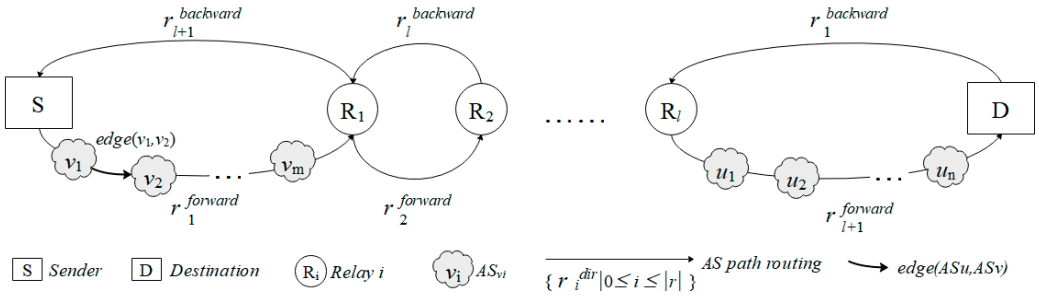

Figure 2. System model schematic.

Inter-domain Routing: An inter-domain AS route $r = [v_1, v_2, \ldots, v_\eta]$ is represented using a set of AS sequences, with η denoting the length of the underlay routing, and for any integer $i, j \in [1, \eta], i \neq j$, then $v_i \neq v_j$. For a given route r, v_1 denotes the source AS of the route and v_η denotes the destination AS of the route. For a given two-segment route $r_1 = [v_1, v_2, \ldots, v_m]$, $r_2 = [u_1, u_2, \ldots, u_n]$, if $(v_m, u_1) \in E$, and $\forall i \in [1, m], \forall j \in [1, n]$, $v_i \neq u_j$, connect the two routes $r_1 \oplus r_2 = [v_1, v_2, \ldots, v_m, u_1, u_2, \ldots, u_n]$, which is also an inter-domain route. In conjunction with overlay circuits, and considering the asymmetry of Internet routing, r^f is used to denote the forward route from the sender to the destination, with r_i^f denoting the ith segment of it, and r^b denotes the return route from the destination to the sender, with r_j^b denoting the jth segment of it.

Exit policy: The exit policy for inter-domain routing is modeled as a function $e_v(dp, r, u)$ in Equation (1).

$$e_v(dp, r, u) \triangleq \bigwedge_{i=1}^{\eta} e_{vi}(dp, [v_i, v_{i+1}, \ldots, v_\eta], v_{i-1}) \tag{1}$$

where $e_v(dp, r, u)$ denotes the exit policy of each AS node. Assuming that the routing information is propagated from the destination AS towards the source AS, if AS_v, according to its own exit policy, will advertise the route $v \oplus r$ with the destination address prefixed with dp to its neighbor AS_u, the function will return a value of 1. Otherwise, the return value is 0.

Price strategy: The price strategy for inter-domain routing is modeled as a function $c(r)$ in Equation (2). The function returns a real number representing the overhead of an inter-domain route.

$$c(r) \triangleq \bigwedge_{i=1}^{\eta} c_{vi}([v_i, v_{i+1}, \ldots, v_\eta]) \tag{2}$$

Security Policy: The security policy for inter-domain routing is modeled as a function $d(p, k, m, n)$ in Equation (3).

$$\begin{aligned} d(p,k,m,n) = 1 \quad & -(1-p)^k + (1-p)^k * \left[1-(1-p)^{m-k}\right] * \left[1-(1-p)^{n-k}\right] \\ & = 1 + (1-p)^{m+n-k} - (1-p)^m - (1-p)^n \end{aligned} \tag{3}$$

In this definition, $d(p, k, m, n)$ denotes the probability that an inter-domain route r is compromised. p denotes the probability that each vertex is attacked in the global network, $0 \leq p \leq 1$, and m and n denote the number of AS vertices in the path segments from the user AS (e.g., Src in Figure 1) to the Entry AS (e.g., R_1 in Figure 1). The number of vertices in the path segments from the exit AS (e.g., R_3 in Figure 1) to the destination AS (e.g., Dst in Figure 1) in an inter-domain route r, respectively. k denotes the number of AS vertices shared in the above two path segments, and $0 \leq k \leq min[m, n], 1 - (1-p)^k$ denotes the probability of the route being compromised when the shared AS vertices are attacked. The remaining portion denotes the probability of the route being compromised when the shared vertices are secure.

$d(p,k,m,n)$ always returns a value between 0 and 1. We can analyze the different cases of k to prove it.

Let $q = 1 - p, where\ q \in [0,1]$.

For $k = 0$, the Equation (3) simplifies to:

$$1 + q^{m+n} - q^m - q^n = (1 - q^m)\cdot(1 - q^n) \in [0,1] \tag{4}$$

For $k = \min\{m,n\}$, let $m \leq n, k = m$. Equation (3) simplifies to:

$$1 + q^n - q^m - q^n = 1 - q^m \in [0,1] \tag{5}$$

For $0 < k < \min\{m,n\}$, $q^{m+n} \leq q^{m+n-k} \leq q^n$, since the value of the Equation (3) lies between Equation (4) and Equation (5), it follows that $d(p,k,m,n) \in [0,1]$.

AS-level adversary: $AS_{Malicious}$ is used to denote an AS-level adversary, which, due to routing detours from overlay circuits and the inherent asymmetry of Internet routing, may be in possession of only a single AS autonomous system, both on routes between the client and ingress relays and between the egress relays and the destination. Thus, the traffic on the ingress and egress segments of the circuit is statistically correlated, as in Equation (6), reflecting the security aspect of the path when d = 1, indicating that the path is compromised.

$$d = 1 \text{ when } AS_{Malicious} \in \begin{cases} r_1^f \ and\ r_{l+1}^f \\ r_1^f \ and\ r_1^b \\ r_{l+1}^f \ and\ r_{l+1}^b \\ r_1^b \ and\ r_{l+1}^b \end{cases} \tag{6}$$

3.2. Optimal Anonymous Routing Problem

The inter-domain optimal routing problem is defined based on the modeling of inter-domain routing in the previous section.

Problem of inter-domain optimal routing: For an inter-domain network $G = (V,E)$, where $e(r)$, $c(r)$, and $d(r)$ denote the exit policy, price policy, and security policy, respectively; $f(r)$ is the global objective function B_{max} is the threshold to satisfy the price policy; and D_{max} is the threshold to satisfy the security of the path strategy, the optimal route r^* from the source AS S to the destination AS D, as defined in Equations (8) and (9), is the solution of the following optimization problem.

$$maxmize\ f(r) Subject\ to \tag{7}$$

$$v_1^{S \to R_1} = v_{|r_{l+1}^b|}^{R_1 \to S} = S \tag{8}$$

$$v_{|r_{l+1}^f|}^{R_l \to D} = v_1^{D \to R_l} = D \tag{9}$$

$$e(r) = 1 \tag{10}$$

$$d(r) \leq D_{max} \tag{11}$$

$$c(r) \leq B_{max} \tag{12}$$

The objective function $f(r)$ returns a real number that represents the objective of the inter-domain route r. Specifically, $f(r)$ is the inverse of the AS routing length, including the weighted exit policy, price strategy, and security polity. This is expressed by the Equation (13) as follows.

$$f(r) = \frac{e(r)}{|r|\cdot d(r)\cdot c(r)} \tag{13}$$

Finding the shortest path that satisfies the constraints over a number of segments—in this case, four segments, where none of the paths are deterministic—may result in each individual path not being the optimal path. However, the combination of these paths can result in an optimal path that satisfies the constraints. Determining the number k, which specifies how many duplicate vertices should be removed and in which segment of the path they should be removed, is a problem of combining segments to produce optimal routes when the pricing strategy is also considered. The naive algorithm involves listing a subset of all path segments of $\{V-k\}$ and examining each one, and the algorithm runs in super-polynomial time when k is close to $V/2$ in general time, so for this problem, it is difficult to find a polynomial time optimal solution.

The 3-CNF-SAT problem (3-Conjunctive Normal Form Satisfiability Problem) is known to be an NPC (Nondeterministic Polynomial Complete) problem, and if the optimal routing problem is reformulated as a determination problem, which is a simplification of the optimal solution problem, then by reducing the 3-CNF-SAT problem to the optimal routing decision problem and proving that the problem is NP-hard, we can prove that our optimal routing problem is NP-hard.

The optimal AS routing problem that satisfies the policy constraints is reformulated. Given an inter-domain network $G=(V,E)$, is it possible to find a route r from the source AS_S to the destination AS_D, such that the following conditions are met:

- The AS path length of route r does not exceed η.
- Each AS traversed by route r complies with the export policy.
- Route r adheres to the pricing policy, with a total cost less than B_{max}.
- Route r follows the security policy and does not traverse the same AS more than once.

To prove that it is NP-hard, first consider an instance of the 3-CNF SAT problem with n clauses, such as $\{C_1, C_2, \ldots, C_n\}$. Let x_{ij} represent the j-th variable in the i-th clause, where $j=1,2,3$. This is denoted as Equation (14).

$$\varphi = (x_{11} \cup x_{12} \cup x_{13}) \cap (x_{21} \cup x_{22} \cup x_{23}) \cap \ldots \cap (x_{n1} \cup x_{n2} \cup x_{n3}) \tag{14}$$

For each clause, construct a graph $G_i=(V_i,E_i)$, where $V_i=\{s_i, v_{i1}, v_{i2}, v_{i3}, t_i\}$, $E_i=\left\{(s_i, v_{ij}), (v_{ij}, t_i) \middle| j=1,2,3\right\}$.

Next, provide an instance of the shortest AS routing problem with policy constraints, where, for the graphs G_i and G_{i+1}, $i=1,\ 2,\ldots,n-1$, as shown in Figure 3. Define the exit policy such that all $AS_{v_{ij}}$ advertise routes to AS_{t_i}, all AS_{s_i} advertise routes to $AS_{v_{ij}}$, and all AS_{t_i} advertise routes to $AS_{s_{i+1}}AS_{s_{i+1}}$. Define the routing security policy such that if $AS_{v_{ij}}$ is the same as another AS in r, then no edge (v_{ij}, t_i) exists; each AS charges $B/3n$, and $\eta=3n-1$. The objective of the problem is to find a route from s_1 to t_n that satisfies the exit policy, price policy, and security policy.

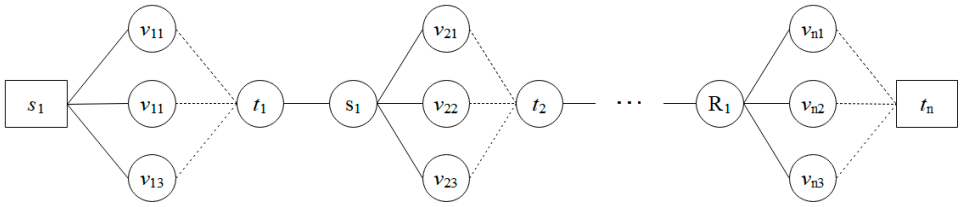

Figure 3. Instance of the optimal AS routing problem with policy constraints.

Constructing an instance of the shortest AS routing problem with policy constraints from the 3-CNF SAT problem instance is achieved in polynomial time. Once the instance is constructed, it can be seen that if the 3-CNF SAT instance is satisfiable, assuming the j-th variable in the C_i clause is true, the route $r=\left\{s_1, v_{1j}, t_1, s_2, v_{2j}, t_2, \ldots, t_n\right\}$ is a route that satisfies the policy constraints. Conversely, if a route satisfying the policy constraints is found, a set of assignments that makes the 3-CNF SAT instance satisfiable will also be determined to be TRUE.

4. Design of TOAR

Finding optimal combinatorial routes that satisfy the policy constraints is challenging. This section presents a black-box algorithm based on Bayesian optimization to find near-optimal policy-compliant end-to-end routes through two-stage route sampling. The underlying logic of the algorithm involves using a priori knowledge of the problem to guide sampling, combining global and local search to achieve trade-offs in the search space, and using a posteriori knowledge of the problem to guide iterative updating. The algorithm aims to improve security with the same cost by estimating the expected improvement of the sampled combinatorial routes and identifying paths that meet the expected improvement for model updating. This iterative process continues to approach the optimal routes.

4.1. First Stage Algorithm for Simplified Security Policy Function

The first stage searches for the shortest paths that satisfy the constraint that shared vertices are fewer than k_{max} by simplifying the security policy function in the algorithm. This is accomplished by ignoring the path lengths m and n and only considering the number of repetitive ASes k as in Equation (15).

$$d \approx 1 - (1-p)^k, 0 \leq k \leq max[m,n] \tag{15}$$

After simplification, d is bounded only by k, and the value of d grows as k grows. Specifically, a larger k means that the route segment has more shared nodes, and thus, the whole circuit is more likely to be compromised. The algorithm determines the threshold k_{max} to be used by the user and filters the routes that satisfy the security policy. A smaller k_{max} threshold means that the user chooses a stronger security policy and vice versa, and a larger k_{max} means a weaker security policy.

Specifically, define the set R_F satisfying the k_{max} constraints, which is initially empty, and use the Policy-Compliance Shortest Routing Algorithm [19] to obtain the four shortest paths that satisfy the policy: $r_1^f, r_{l+1}^f, r_1^b, r_{l+1}^b$.

Take the computed $r = r_1^f \oplus r_{l+1}^f \oplus r_1^b \oplus r_{l+1}^b$ as the a priori of the whole black-box optimization problem, and then use it as the basis to carry out path sampling in the approximate global search space. First, the two shortest paths, r_1^f and r_{l+1}^b, are taken as two segments in the final path. Then, we search for new paths of r_1^f and r_{l+1}^b segments that satisfy the requirement that the shared nodes with the paths of r_{l+1}^f and r_1^b segments are fewer than k_{max}.

Specifically, take $m - k_{max}$ vertices from the m vertices of the set $V_{r_1^f} \cup V_{r_{l+1}^b}$ each time and label the order of the set consisting of these points as V_i^M. Find the shortest path that matches the strategy, add it to the set R_F, and update R_F. Then, use the shortest paths, $V_{r_{l+1}^f}$ and $V_{r_1^b}$, as two segments in the determined final path, and search for new r_{l+1}^f and r_1^b segment paths that satisfy the requirement that the vertices shared with the r_1^f and r_{l+1}^b segments paths are fewer than k_{max}. One at a time, from the $V_{r_{l+1}^f} \cup V_{r_1^b}$ set of n nodes set to take $n - k_{max}$ vertices, label the order of the set composed of these points as V_i^N, find the shortest path that meets the policy, add it to the R_F set, and update R_F. Finally, return the shortest path in R_F, as in the search algorithm in Algorithm 1.

If providing anonymity services is considered to require path diversity properties for routes, R_F may provide pools of alternative paths. This is because information about route lengths may degrade anonymity if routing protocols always choose the shortest route.

Algorithm 1 Search for shortest routes that satisfy the constraint that shared vertices are less than k_{max} (SKCR)

Require: r_1^f, r_{l+1}^f, r_1^b, r_{l+1}^b

Ensure: $argmax_{\ r \in R_F} = f(r)$

$R_F = \emptyset$; $R_{r_1^f} = \emptyset$; $R_{r_{l+1}^f} = \emptyset$; $R_{r_1^b} = \emptyset$; $R_{r_{l+1}^b} = \emptyset$

$V_{r_1^f} = \{v_1^{S \to R_1}, v_2^{S \to R_1}, \ldots, v_{|r_1^f|}^{S \to R_1}\}$

$V_{r_{l+1}^f} = \{v_1^{R_l \to D}, v_2^{R_l \to D}, \ldots, v_{|r_{l+1}^f|}^{R_l \to D}\}$

$V_{r_1^b} = \{v_1^{D \to R_l}, v_2^{D \to R_l}, \ldots, v_{|r_1^b|}^{D \to R_l}\}$

$V_{r_{l+1}^b} = \{v_1^{R_1 \to S}, v_2^{R_1 \to S}, \ldots, v_{|r_{l+1}^b|}^{R_1 \to S}\}$

Compute k from r_1^f, r_{l+1}^f, r_1^b, r_{l+1}^b

if $k \leq k_{max}$ **then**

  return $r = r_1^f \oplus r_{l+1}^f \oplus r_1^b \oplus r_{l+1}^b$

else

  Compute $\{V_i^M | 1 \leq i \leq m\mathrm{C}(m - k_{max})\}$ from $V_{r_1^f} \cup V_{r_{l+1}^b}$

  for $i = 1, 2, \ldots, m\mathrm{C}(m - k_{max})$ **do**

    $V_G{}^* = V_G - V_i^M$

    $R_{r_{l+1}^f} = PCSR\ (V_G{}^*, v_1^{R_l \to D}, v_{|r_{l+1}^f|}^{R_l \to D})$

    $R_{r_1^b} = PCSR\ (V_G{}^*, v_1^{D \to R_l}, v_{|r_1^b|}^{D \to R_l})$

    $R_F = R_F \cup R_{r_{l+1}^f} \cup R_{r_1^b}$

  end for

  Compute $\{V_i^N | 1 \leq i \leq n\mathrm{C}(m - k_{max})\}$ from $V_{r_{l+1}^f} \cup V_{r_1^b}$

  for $i = 1, 2, \ldots, n\mathrm{C}(m - k_{max})$ **do**

    $V_G{}^* = V_G - V_i^N$

    $V_{r_1^f} = PCSR\ (V_G{}^*, v_1^{S \to R_1}, v_{|r_1^f|}^{S \to R_1})$;

    $V_{r_{l+1}^b} = PCSR\ (V_G{}^*, v_1^{R_1 \to S}, v_{|r_{l+1}^b|}^{R_1 \to S})$;

    $R_F = R_F \cup V_{r_1^f} \cup V_{r_{l+1}^b}$

  end for

end if

return $argmax_{\ r \in R_F} = f(r)$

If the autonomous domain policy is not considered, i.e., the symmetric routing scenario, Dijkstra's algorithm or Yen's algorithm can be indirectly used instead of the policy-compliance shortest routing algorithm. The symmetric routing scenario can be used as a special case of Algorithm 1. For example, in the case of LAP [20], symmetric routing is used, where each AS along the route adds the encrypted path to the packet. During the path establishment phase, as the packet is forwarded, the complete encrypted path is generated. The receiver host then uses the complete encrypted path carried by the arriving packet to return an answer.

When the number of shared vertices $k \leq k_{max}$, the algorithm can return early; otherwise, the complexity is based on the complexity of the PCSR algorithm referenced in Algorithm 1. Specifically, in the worst case, it needs to enumerate all the possible routes, and the complexity is $O(V!)$, so the iteration number is defined to limit the execution time. The complexity of Algorithm 1 is $O(mC(m - k_{max}) \times T_{PCSR} + nC(n - k_{max}) \times T_{PCSR})$, which depends on m, n, k_{max}. In other words, the complexity is determined by the route length and the user-defined k_{max}. When m and n are large, the growth of the combination count is rapid, reaching a maximum when k_{max} reaches half of the route length. However, even

in Internet topologies, the scale of path length is limited. Moreover, k_{max} is defined by the user, allowing Algorithm 1 to complete the search within a user-acceptable range.

4.2. Second Stage Algorithm for Full Security Policy Function

Based on the first phase of the algorithm, which finds the set of routes that comply with the security policy, the second phase of the optimization algorithm performs a local search in the vicinity of the path obtained in the first phase. It progressively removes shared nodes, searches for a new optimal path near the removed node, and updates the iterative basis from a priori to a posteriori knowledge. The goal is to improve security without increasing cost. If the iterative conditions are not met, a different shared node is selected for removal. The algorithm terminates when the set number of iterations is reached, all shared nodes have been removed, or no further improvement in the path can be obtained. Considering the worst-case scenario, the algorithm may enumerate every possible route. Therefore, similar to the PCSR algorithm [19], the iteration number is set as a termination condition for the search, controlling the overall runtime of the entire search process.

The second stage uses the complete security policy as in Equation (3). The security policy on which the algorithm based is jointly determined by the length of the path and the number of shared nodes, and $d(r)$ is subsequently used instead of $d(p,k,m,n)$ to simplify the expression. Define $I_d\left(r,r^*\right) = d(r) - d\left(r^*\right)$ to denote the improvement in the security of the newly sampled paths compared to the previous paths. Define $I_{path}\left(r,r^*\right) = length\left(r^*\right) - length(r)$ to denote the increase in the length of the newly sampled obtained path compared to the previous path. Define the price policy $c\left(r,r^*\right)$, as in Equation (16) to denote the unit cost of routing security enhancement.

$$c\left(r,r^*\right) = \frac{b \cdot I_{path}\left(r,r^*\right)}{I_d(r,r^*)} \tag{16}$$

The specific practice of Iteration Optimal Searching Routing, as in Algorithm 2, begins with the shortest path as r and uses the optimal path obtained in the first stage of the algorithm as r^*. The value of $c(r)$ is calculated as the maximum value of $c(r)$ that is acceptable to the user. Thereafter, in the second stage algorithm, $c(r)$ decreases as an iterative condition for route optimization. If the condition is satisfied, the route is saved as r^*, and the last r^* is used as r, continuing the iteration. If the condition is not satisfied, the path is discarded and resampled.

The complexity of Algorithm 2 is $O(k \times T_{PCSR})$. The reason for not using Algorithm 2 directly and instead completing the process in two stages is to balance the routing search space. Algorithm 1 searches for routes that meet the conditions within a larger search space, while Algorithm 2 then refines the results by searching in the nearby region of each route found.

4.3. Programmable Interface for Anonymous Routing

To create and run applications, a software defined programmable interface abstraction is designed for each participating AS, extending support for end-to-end routing while avoiding conflicts with the global routing system. Through software defined, programmable interfaces that support inter-domain communication in SDN, the negotiation and confirmation of end-to-end routing are facilitated. Each AS integrates the programmable interface along with traditional routing functions. Based on the requirements of anonymous services, the ASes publicly expose their interfaces, allowing users such as other ASes to query policies and negotiate routes. Users can select the next hop from a set of available exits, achieving end-to-end routing control.

The software defined programmable interface, as shown in Figure 4, allows users to query and confirm routing decisions. It enables SDN controllers to manage and program network paths based on application policies and routing protocols. SDN applications and routing protocols are integrated into the SDN controller, which dynamically generates

forwarding rules based on application requirements and network conditions. AS_v and AS_u are neighboring ASes, and they exchange routing information via routing protocols such as BGP, which can be enhanced by segment routing for more flexible path control.

Algorithm 2 Iterative optimal search routing algorithm

Require: $G, r_1^f, r_{l+1}^f, r_1^b, r_{l+1}^b, k_{max}$
Ensure: r

$r = r_1^f \oplus r_{l+1}^f \oplus r_1^b \oplus r_{l+1}^b$
$V_G = \{v_1, v_2, \ldots, v_{|r|}\}$
$r^* = \mathrm{SKCR}\ (G, r_1^f, r_{l+1}^f, r_1^b, r_{l+1}^b,\ k_{max})$
$R_k = \{u_1, u_2, \ldots, u_k\};$
$C = c(r, r^*) = \frac{b \cdot I_{path}(r, r^*)}{I_d(r, r^*)};$
if k_{max}>=0 **then**
  $k = k_{max}$
else
  return r
end if
for $u_i\ (i = 1, 2, \ldots, k)\ \in R_k$ **do**
  Remove u_i
  $V_G{}^* = V_G - V_r$
  $r\_add = PCSR\ (V_G{}^*, u_i\_pre, u_i\_suc)$
  $r = r^*;$
  $V_r = \{v_1, v_2, \ldots r\_add \ldots, v_{|r|}\};$
  $r^* = r\_pre\_half \oplus r\_add \oplus r\_suc\_half$
  $C^* = c(r, r^*) = (b \cdot I\_path(r, r^*))/(I\_d(r, r^*))$
  if $C^* < C$ **then**
    $r = r^*$
    $C = C^*$
  else
    $i = i + 1$
  end if
end for
return r

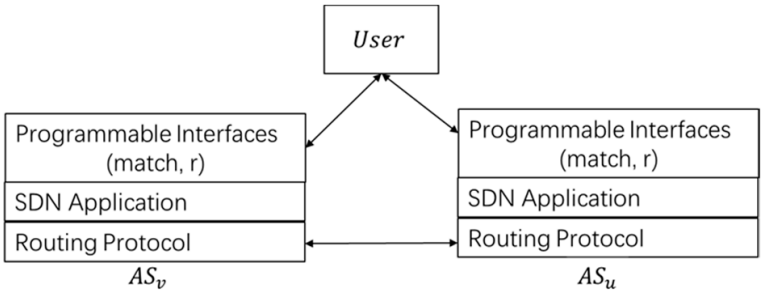

Figure 4. Software defined inter-domain programmable interface.

When it is necessary to protect privacy-sensitive information within domains, each AS domain must not disclose private information, including routing selection policies, pricing strategies, and exit policies. Therefore, in the process of establishing end-to-end routing, the user needs to check whether the route they wish to establish complies with the policies of each AS domain. The interaction process for inter-domain route selection is shown in Figure 5. For a given end-to-end route $r = [v_1, v_2, \ldots, v_n]$, the user queries the ASes along the route in reverse order. First, they query AS_n, and if an agreement is reached, the user

can query about the route AS_{n-1}. Then, they query AS_{n-2}, continuing this process until the user's own AS domain AS_1 is returned, completing the selection of an end-to-end route. The user ultimately confirms the optimal route r in the same reverse manner. Separating the route query and confirmation processes prevents the user from facing the price cost for routes not actually used.

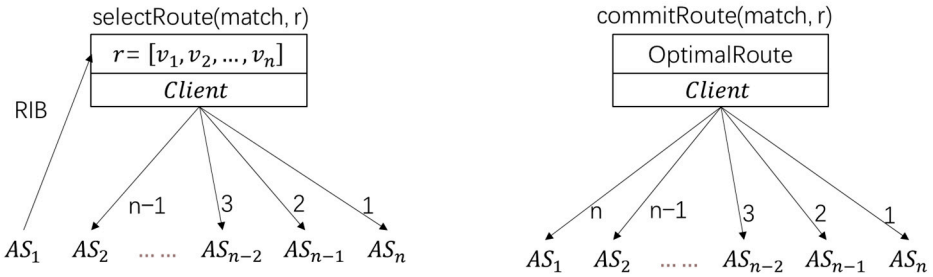

Figure 5. Process of route negotiation and route confirmation.

The route queries generated by the user are divided into remote queries and local queries. In a remote query, the user sends a query request to the AS to check whether the selected route complies with the policy, and the result is cached locally. Remote queries are a time-consuming and costly operation. If the route policy exists at the time of the query request, a local query is conducted, and the result is retrieved from the local cache instead.

5. Anonymity Analysis

In this section, we employ the anonymity analysis method proposed by Reiter and Rubin [21] for correlation attacks to assess the anonymity of TOAR.

Assuming the length of the anonymous transmission path is n, with c compromised nodes, the probability that a forwarding node is an effective node is $p = 1 - \frac{c}{n}$. The probability that the sender S and the receiver R are correctly associated by the adversary through the information collected from compromised nodes is denoted as d. According to the anonymity analysis method proposed by [21,22], d can be calculated as follows:

$$d = \frac{(1-p)^2}{\left(1-p^{np-1}\right)\left(1-p^{np}\right)} \tag{17}$$

d can serve as the anonymity metric for anonymous communication systems. The smaller the value of d, the stronger the system's anonymity, and vice versa.

For TOAR, after the execution of the two-stage routing algorithm, there is at most one compromised node controlled by the adversary on the anonymous transmission path. That is $c = 1$; then, $p = 1 - \frac{1}{n}$. Therefore, the probability d can be simplified as:

$$d = \frac{\left(\frac{1}{n}\right)^2}{\left(1-\left(1-\frac{1}{n}\right)^{n-1}\right)\left(1-\left(1-\frac{1}{n}\right)^{n}\right)} \tag{18}$$

The probability of TOAR is evaluated based on Equation (18). From Figure 6, we can see that d decreases with the increase in n, which results in stronger anonymity of TOAR. With the gradual increase in n, the probability d is decreased, and its value is far less than 1. When $n = 5$, especially, the probability d is already close to 0.1; when $n > 15$, d approaches 0, nearing the level of absolute anonymity, which proves the excellent anonymous feature of TOAR.

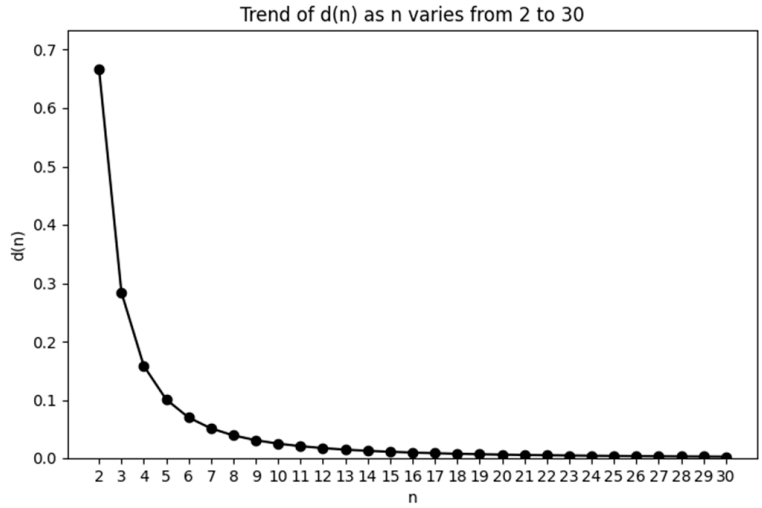

Figure 6. Impact of path length n on the probability d.

6. Evaluation

The goal of TOAR is to achieve sender anonymity as well as sender–receiver anonymity while ensuring that the scheme does not significantly affect communication performance in terms of latency, throughput, and resource overhead when deployed in SDN networks. Therefore, we evaluate the scheme in terms of anonymity analysis, scheme effectiveness analysis, and performance measurement. The experiments are set up in two ways. The first part of the experimental setup aims to evaluate the effectiveness of the two-phase route optimization algorithm, and the second part aims to evaluate the forwarding performance of the anonymous routing mechanism.

6.1. Effectiveness Analysis

The first part of the experiment involves a MATLAB code and data to evaluate the effectiveness of the algorithm. The CAIDA AS relationship dataset is used, which contains 63,361 nodes and 320,978 edges. For each AS, a routing policy is configured. Based on the topology and policy settings from the dataset, the top 10 ASes with the highest traffic are chosen as destinations. Additionally, for each selected destination AS, the top 200 ASes sending the most traffic to that destination are identified, generating 2000 end-to-end routing intents. By mapping Tor nodes to Internet AS nodes using the Tor Metric published relay node information, 1000 valid end-to-end intents are filtered out. In the following step, TOAR's effectiveness is analyzed in terms of the security policy function, the expected improvement function, and the routing search results.

- Effectiveness of Security Policy Functions:

The security policy function $d(p, k, m, n)$, as defined in Equation (3) in the system model, portrays the probability that an inter-domain route r will be compromised, and Figure 7 shows an example of this function, with the security policy function values on the vertical axis and the k values on the horizontal axis. The function grows by nearly 100% in the case of $p = 0.1, k_{max}\ 0 \rightarrow 7, max\ (m, n) < 10$. From the intuition of reducing the number of shared points and the change in the modeled security function, deploying a two-stage route optimization algorithm improves the anonymity of the system.

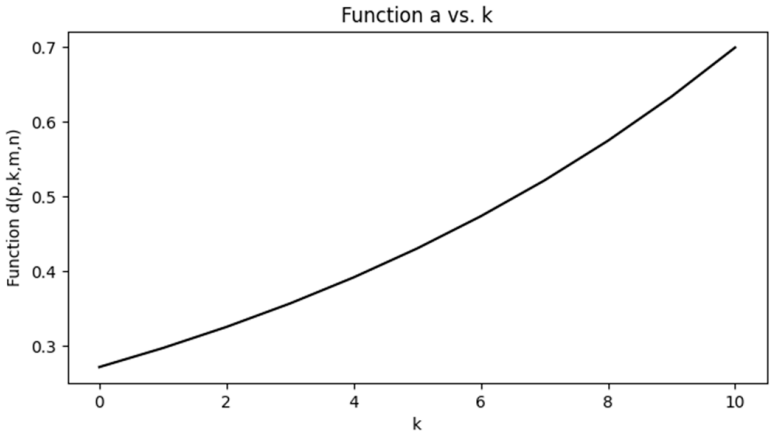

Figure 7. Trend of security policy function $d(p, k, m, n)$.

- Effectiveness of Expectation Improvement Function:

The second stage of the algorithm is based on Equation (16) which defines the unit cost of routing security improvement, $C = c(r, r^*) = \frac{b \cdot I_{path}(r, r^*)}{I_d(r, r^*)}$, to decide whether it can be further iteratively optimized or not, so the absolute magnitude of the difference between C^* and C, ΔC, can be a measure of how much the algorithm has improved the security. Therefore, the absolute value of the difference between C^* and C, ΔC, can be used to measure the improvement of the algorithm in terms of security. Other metrics can also be used; for example, the magnitude of the quotient between C^* and C can be used to indicate the degree of improvement in the second stage. Based on the experimental result data, the enhancement of the optimization algorithm in the second stage is counted, as shown in Figure 8.

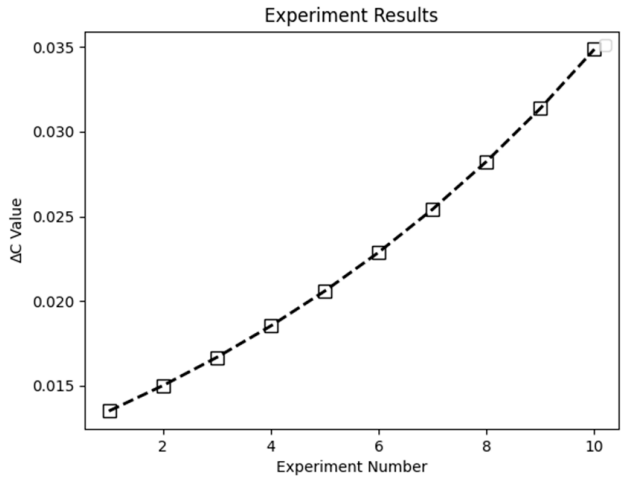

Figure 8. The ΔC statistic of the results.

- Effectiveness of search results:

Within a given number of cycles, TOAR can dramatically increase the chances of finding an end-to-end route that matches the policy by about 60%, which is nearly 2.5 times higher compared to the 25% success rate of a naive enumeration search, as shown in Figure 9.

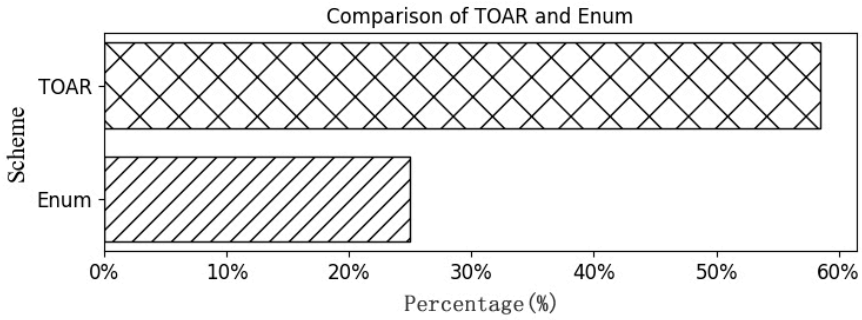

Figure 9. Percentage of search effectiveness.

6.2. Network Performance Measurement

In the second part of the experiment, the anonymous routing method described in STAR [23] is used to measure the performance of TOAR for route forwarding under different path lengths. All experiments are conducted using the SDN network simulator Mininet, and the Mininet version is 2.3.0d6. Meanwhile, the hardware consists of one server running Ubuntu 16.04 LTS operating system (Canonical Ltd., London, UK) with Intel®Core™i7-9700@ 3.00 GHz CPU (Intel Corporation, Santa Clara, CA, USA). Ryu and Open vSwitch are installed, acting as the SDN controller and virtual switch and, the network topology is the Internode topology from the Zoo dataset, which includes 66 nodes, with each node functioning as an SDN switch. An SID is assigned to each node, representing its unique identifier. As shown in Figure 10, Algorithms 1 and 2 are integrated into the Ryu controller. STAR uses SDN topology discovery as the input for shortest path computation to generate the segment list for segment routing and sends OpenFlow rules to the OVS switches, which perform matching and forwarding based on SIDs. Based on SDN topology discovery, an AS relationship table with exit policies is generated and used as input for the TOAR routing search algorithm, which then generates the segment routing list according to the computed optimal route.

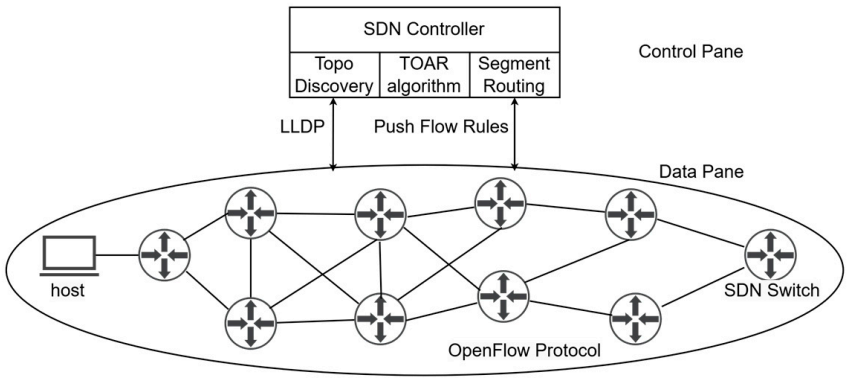

Figure 10. SDN experiment environment.

- Throughout evaluation for TOAR:

Iperf is used to assess the throughput of both TOAR-disabled (normal route forwarding) and TOAR-enabled (anonymous route forwarding) systems by varying data sizes (with the path length set to a default of 3). As illustrated in Figure 11, the throughput for both systems rises as the data size increases. However, rather than directly forwarding packets, TOAR requires additional processing time for route sampling, which results in reduced throughput, but enhances security and privacy protection.

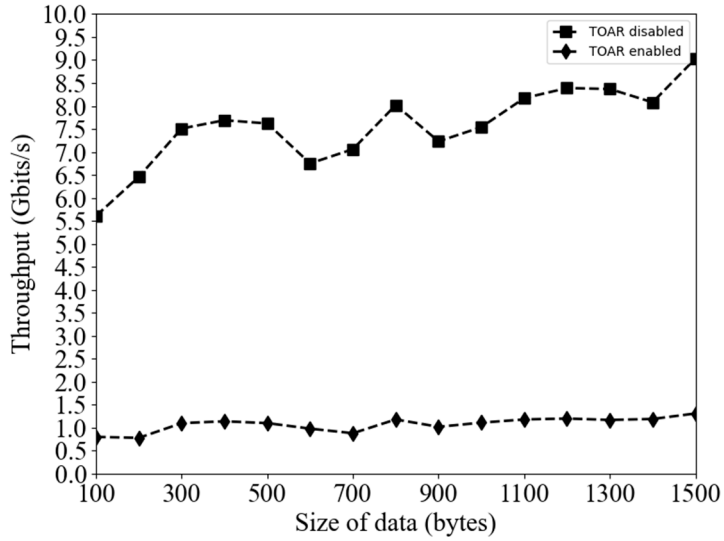

Figure 11. Impact of data size changed on throughput.

- Anonymous route effectiveness:

We further evaluate the throughput of route forwarding in order to illustrate the effectiveness of anonymous route with TOAR. IP forwarding is used as our performance baseline. For comparison, three reference systems are introduced: STAR [23], TOAR (high anonymity), and lightweight TOAR (low anonymity).

Figure 12 demonstrates the trade-off between anonymity and latency in different schemes. IP performs the best with the lowest latency, but it does not provide anonymity. Tor has the highest latency, and its latency increases the fastest as path length increases. TOAR has moderate latency, but the test results are slightly lower than STAR because STAR does not consider exit policies and calculates routes based on the shortest path.

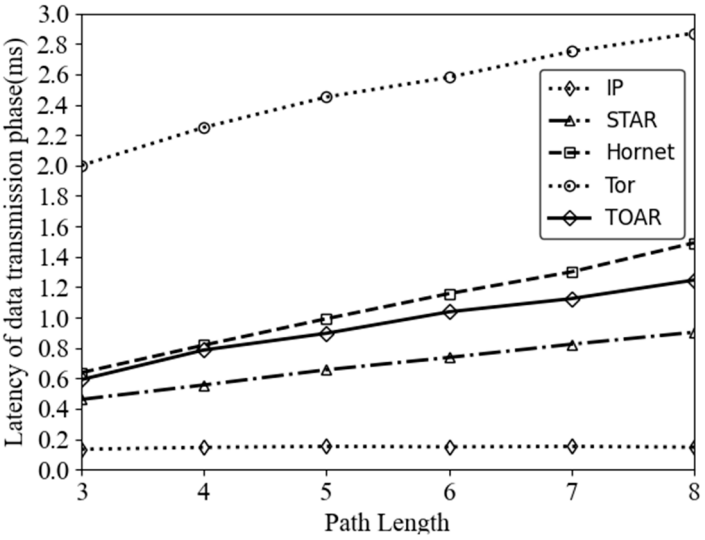

Figure 12. Communication latency.

Figure 13 demonstrates the effectiveness of all schemes when varying the packet size from 128 bytes to 1500 bytes. According to the experimental results, the throughput performance of the TOAR and STAR schemes is close, indicating that the performance of our scheme is comparable to that of STAR. In the case that only three routing nodes are selected for anonymous forwarding and other routing nodes are forwarded normally, the lightweight TOAR scheme can achieve nearly 70% of the throughput of the IP scheme, as shown in Figure 13.

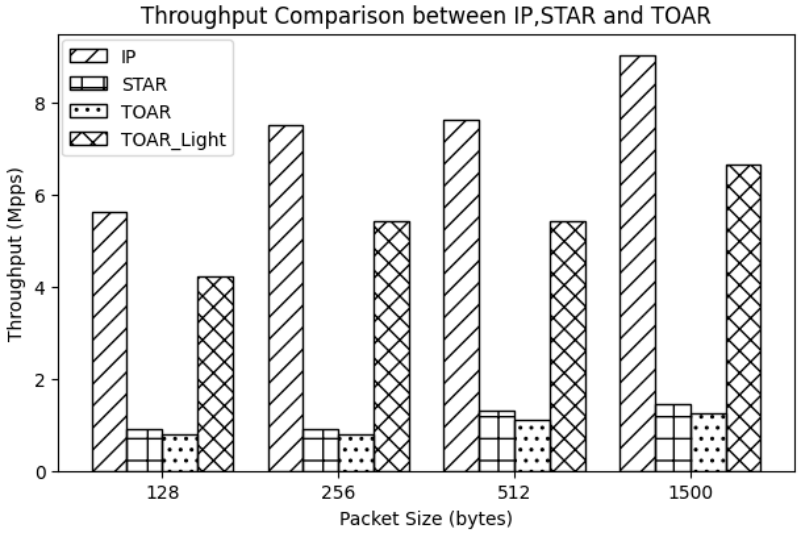

Figure 13. Throughput of route forwarding.

From above, TOAR enhances the anonymity of the communication system while maintaining the throughput within an acceptable range, ensuring the effectiveness of the proposed scheme.

7. Related Works

Research work related to this paper includes anonymous communication routing algorithms, inter-domain routing, and software-defined networks.

By manipulating the underlying routing, AS-level adversaries increase their chances of executing correlation attacks, leading to the proposal of various AS-aware path selection schemes for Tor [9,10,24–26]. Nick Feamster provided the first work [14] considering attacks from the perspective of an AS-level adversary. To counter the risk of AS-level adversary attacks, LASTor [24] modifies the Tor routing algorithm and designs and implements a new Tor client, which achieves probabilistic path selection by predicting Internet routes between onion nodes and end hosts to avoid ASes that may be subjected to an attack at the AS level. Counter-RAPTOR [12] proposed the Tor guard node selection algorithm and designed two monitoring frameworks for monitoring BGP to detect control plane attacks and monitoring trace routes to detect data plane anomalies in order to minimize and detect AS-level routing attacks. However, overlay AS-aware path selection schemes provide partial routing information to infer the underlying network path between two endpoints on the Internet. The most direct method for detecting the ASs at both ends of a Tor connection is to use tools similar to traceroute to analyze the network path from the client to the selected entry node and from the exit node to the destination. These inferences may be inaccurate.

SDN is a promising way to re-architect the Internet, and the transition from traditional networks to SDN is an important issue. RouteFlow [27] is one of the earliest methods to implement IP routing on OpenFlow switches. Another BGP-based solution is SDN-IP [28], which focuses on solving the problem of seamless interconnection between SDN domains and traditional domains. Some new routing protocols [19,29–33] have been proposed to address the Internet routing scalability and source-controlled routing to provide flexible end-to-end routing schemes, but do not focus on identity privacy in communication.

Some new network-level high-speed anonymous systems [20,23,34–39] have been proposed, but have not focused on routing computations. These studies inspire us to tackle the problem of anonymity degradation in Tor by AS adversary association attacks, focusing on AS routing in SDN networks. However, these studies assume that inter-domain shortcuts provided by the underlying network architecture already exist and adequately address routing computation issues. Both from the viewpoint of network-layer anonymous

communication and in supporting the overlay anonymous such as Tor in AS-aware relay node selection, there are inherent challenges in routing computation.

This paper takes a different approach than overlay solutions by proposing a method focused on end-to-end routing control within an SDN architecture. It aims to identify inter-domain routing paths that balance performance—specifically, the speed of low-latency anonymous communication—and security, represented by the probability of paths being compromised by adversaries. Differing from existing SDN-based network-layer anonymity solutions, this paper extends anonymous communication from intra-domain control to inter-domain control within SDN networks, ultimately supporting anonymity as a service for users on a broader scale.

8. Conclusions

In this work, we propose TOAR, an optimal anonymous routing mechanism that incorporates SDN, designed to resist correlation attacks from AS-level adversaries in anonymous communication using Tor. TOAR addresses the challenges of reduced anonymity and increased communication delays caused by AS-level attackers. It employs a two-stage optimization strategy. In the first stage, it identifies routes that adhere to policy guidelines while avoiding AS that could link traffic between the source and destination, thus ensuring anonymity and minimizing communication costs. The second stage includes a programmable interface for SDN inter-domain routing, allowing users to query routing policy information and validate their routing selections. This feature supports flexible end-to-end source routing and extends the SDN anonymity mechanism across multiple domains. Anonymity analysis and experimental results reveal that the proposed TOAR provides strong anonymity without significantly degrading communication performance.

Our security policy targets specific objectives without general comparison or discussion. Although we propose a software-defined source routing method, suggesting it could enable source routing over BGP with added programmable interfaces (e.g., route query and confirmation), the route discovery process is only simulated in code, and no dynamic routing protocol such as BGP has been deployed in an SDN test environment.

In future work, based on BGP speakers with extended programmable interfaces and segment routing, we will deploy anonymous routing across networks with multiple SDN controllers in an SDN test environment, which will allow us to discuss the scalability and efficiency of SDN AS-level routing. Additionally, we are considering a generic framework for anonymous routing, with some issues such as a consistency attack model, consistent anonymity objectives, and unified model abstraction methods as our future research direction. Furthermore, we consider a generic framework for anonymous routing, with some issues such as a consistency attack model, consistent anonymity objectives, and unified model abstraction methods as our future research direction.

Author Contributions: Conceptualization, H.Z.; methodology, H.Z.; software, H.Z.; validation, H.Z. and X.S.; formal analysis, H.Z. and X.S.; investigation, X.S.; resources, X.S.; data curation, H.Z.; writing—original draft preparation, H.Z.; writing—review and editing, H.Z. and X.S.; visualization, H.Z.; supervision, X.S.; project administration, X.S.; funding acquisition, X.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The datasets presented in this article are not readily available due to privacy, as the data are part of an ongoing study.

Conflicts of Interest: We declare that we have no financial and personal relationships with other people or organizations that can inappropriately influence our work and that there is no professional or other personal interest of any nature or kind in any product, service, and/or company.

References

1. Dingledine, R.; Mathewson, N.; Syverson, P.F. Tor: The second-generation onion router. In Proceedings of the USENIX Security Symposium, San Diego, CA, USA, 9–13 August 2004. [CrossRef]

2. Tor Metrics. Available online: https://metrics.torproject.org (accessed on 17 October 2024).
3. Nasr, M.; Bahramali, A.; Houmansadr, A. Deepcorr: Strong flow correlation attacks on tor using deep learning. In Proceedings of the ACM SIGSAC Conference on Computer and Communications Security, Toronto, ON, Canada, 15–19 October 2018. [CrossRef]
4. Xu, Y.; Yang, M.; Ling, Z.; Liu, Z.; Gu, X.; Luo, L. A De-anonymization Attack against Downloaders in Freenet. In Proceedings of the IEEE Conference on Computer Communications, New York, NY, USA, 15–19 October 2024. [CrossRef]
5. Chao, D.; Xu, D.; Gao, F.; Zhang, C.; Zhang, W.; Zhu, L. A Systematic Survey On Security in Anonymity Networks: Vulnerabilities, Attacks, Defenses, and Formalization. *IEEE Commun. Surv. Tutor.* **2024**, *26*, 1775–1829. [CrossRef]
6. Gegenhuber, G.K.; Maier, M.; Holzbauer, F.; Mayer, W.; Merzdovnik, G.; Weippl, E.; Ullrich, J. An extended view on measuring tor as-level adversaries. *Comput. Secur.* **2023**, *132*, 103302. [CrossRef]
7. Lopes, D.; Dong, J.-D.; Medeiros, P.; Castro, D.; Barradas, D.; Portela, B.; Vinagre, J.; Ferreira, B.; Christin, N.; Santos, N. Flow Correlation Attacks on Tor Onion Service Sessions with Sliding Subset Sum. In Proceedings of the Network and Distributed System Security Symposium, San Diego, CA, USA, 26 February–1 March 2024. [CrossRef]
8. Guan, Z.; Liu, C.; Gou, G.; Li, Z.; Xiong, G.; Ding, Y.; Hou, C. A blind flow fingerprinting and correlation method against disturbed anonymous traffic based on pattern reconstruction. *Comput. Netw.* **2024**, *254*, 110831. [CrossRef]
9. Hogan, K.; Servan-Schreiber, S.; Newman, Z.; Weintraub, B.; Nita-Rotaru, C.; Devadas, S. Shortor: Improving tor network latency via multi-hop overlay routing. In Proceedings of the IEEE Symposium on Security and Privacy (SP), San Francisco, CA, USA, 22–26 May 2022. [CrossRef]
10. Rochet, F.; Wails, R.; Johnson, A.; Mittal, P.; Pereira, O. CLAPS: Client-Location-Aware Path Selection in Tor. In Proceedings of the ACM SIGSAC Conference on Computer and Communications Security, Virtual Event USA, 9–13 November 2020; Association for Computing Machinery: New York, NY, USA, 2020; pp. 17–34. [CrossRef]
11. Rahimi, M. CLAM: Client-aware routing in mix networks. In Proceedings of the ACM Workshop on Information Hiding and Multimedia Security, Baiona, Spain, 24–26 June 2024. [CrossRef]
12. Sun, Y.; Edmundson, A.; Feamster, N.; Chiang, M.; Mittal, P. Counter-RAPTOR: Safeguarding Tor against active routing attacks. In Proceedings of the IEEE Symposium on Security and Privacy, San Jose, CA, USA, 22–26 May 2017. [CrossRef]
13. Mitseva, A.; Aleksandrova, M.; Engel, T.; Panchenko, A. Security and performance implications of BGP rerouting-resistant guard selection algorithms for Tor. *Comput. Secur.* **2023**, *132*, 103374. [CrossRef]
14. Mathewon, N.; Dingledine, R. Location Diversity in Anonymity Networks. In Proceedings of the ACM Workshop on Privacy in the Electronic Society, Washington, DC, USA, 28 October 2004; Association for Computing Machinery: New York, NY, USA. [CrossRef]
15. Johnson, A.; Wacek, C.; Jansen, R.; Sherr, M.; Syverson, P. Users get routed: Traffic correlation on Tor by realistic adversaries. In Proceedings of the ACM SIGSAC Conference on Computer & Communications Security, Berlin, Germany, 4–8 November 2013. [CrossRef]
16. Nithyanand, R.; Starov, O.; Zair, A.; Gill, P.; Schapira, M.J.a.p.a. Measuring and mitigating AS-level adversaries against Tor. In Proceedings of the Network and Distributed System Security Symposium, San Diego, CA, USA, 21–24 February 2016. [CrossRef]
17. Sun, Y.; Edmundson, A.; Vanbever, L.; Li, O.; Rexford, J.; Chiang, M.; Mittal, P. {RAPTOR}: Routing attacks on privacy in tor. In Proceedings of the USENIX Security Symposium, Washington, DC, USA, 12–14 August 2015; Available online: https://dl.acm.org/doi/abs/10.5555/2831143.2831161 (accessed on 17 October 2024).
18. McCoy, D.; Bauer, K.; Grunwald, D.; Kohno, T.; Sicker, D. Shining light in dark places: Understanding the Tor network. In Proceedings of the Privacy Enhancing Technologies, Leuven, Belgium, 23–25 July 2008. [CrossRef]
19. Xiang, Q.; Zhang, J.; Gao, K.; Lim, Y.-s.; Le, F.; Li, G.; Yang, Y.R. Toward optimal software-defined interdomain routing. In Proceedings of the IEEE Conference on Computer Communications, Toronto, ON, Canada, 6–9 July 2020. [CrossRef]
20. Hsiao, H.-C.; Kim, T.H.-J.; Perrig, A.; Yamada, A.; Nelson, S.C.; Gruteser, M.; Meng, W. LAP: Lightweight anonymity and privacy. In Proceedings of the IEEE Symposium on Security and Privacy, San Francisco, CA, USA, 20–23 May 2012. [CrossRef]
21. Reiter, M.K.; Rubin, A.D. Crowds: Anonymity for web transactions. *ACM Trans. Inf. Syst. Secur.* **1998**, *1*, 66–92. Available online: https://dl.acm.org/doi/pdf/10.1145/290163.290168 (accessed on 17 October 2024). [CrossRef]
22. Wu, Z.; Zhou, Y.; Ma, J. A security transmission model for Internet of things. *Chin. J. Comput.* **2011**, *34*, 1351–1364. [CrossRef]
23. Feng, L.; Ni, X.; Ling, Z.; Wang, L. Strong anonymous communication system based on segment routing over sdn. *Comput. J.* **2023**, *66*, 3092–3106. [CrossRef]
24. Akhoondi, M.; Yu, C.; Madhyastha, H.V. LASTor: A low-latency AS-aware Tor client. In Proceedings of the IEEE Symposium on Security and Privacy, Sam Francisco, CA, USA, 20–23 May 2012. [CrossRef]
25. Edman, M.; Syverson, P. AS-awareness in Tor path selection. In Proceedings of the ACM Conference on Computer and Communications Security, Chicago, IL, USA, 9–13 November 2009. [CrossRef]
26. Barton, A.; Wright, M. Denasa: Destination-naive as-awareness in anonymous communications. *Proc. Priv. Enhancing Technol.* **2016**, *2016*, 356–372. [CrossRef]
27. Lin, P.; Hart, J.; Krishnaswamy, U.; Murakami, T.; Kobayashi, M.; Al-Shabibi, A.; Wang, K.-C.; Bi, J. Seamless interworking of SDN and IP. In Proceedings of the ACM SIGCOMM, Hong Kong, China, 12–16 August 2013. [CrossRef]
28. Zhang, X.; Hsiao, H.-C.; Hasker, G.; Chan, H.; Perrig, A.; Andersen, D.G. SCION: Scalability, control, and isolation on next-generation networks. In Proceedings of the IEEE Symposium on Security and Privacy, Oakland, CA, USA, 22–25 May 2011. [CrossRef]

29. Peter, S.; Javed, U.; Zhang, Q.; Woos, D.; Anderson, T.; Krishnamurthy, A. One tunnel is (often) enough. *ACM SIGCOMM Comput. Commun. Rev.* **2014**, *44*, 99–110. [CrossRef]
30. Xu, W.; Rexford, J. MIRO: Multi-path interdomain routing. In Proceedings of the ACM SIGCOMM, Pisa, Italy, 11–15 September 2006. [CrossRef]
31. Yang, X.; Clark, D.; Berger, A.W. NIRA: A new inter-domain routing architecture. *IEEE/ACM Trans. Netw.* **2007**, *15*, 775–788. [CrossRef]
32. Gupta, A.; Vanbever, L.; Shahbaz, M.; Donovan, S.P.; Schlinker, B.; Feamster, N.; Rexford, J.; Shenker, S.; Clark, R.; Katz-Bassett, E. Sdx: A software defined internet exchange. *ACM SIGCOMM Comput. Commun. Rev.* **2014**, *44*, 551–562. [CrossRef]
33. Godfrey, P.B.; Ganichev, I.; Shenker, S.; Stoica, I. Pathlet routing. *ACM SIGCOMM Comput. Commun. Rev.* **2009**, *39*, 111–122. [CrossRef]
34. Bajic, A.; Becker, G.T. dPHI: An improved high-speed network-layer anonymity protocol. *Proc. Priv. Enhancing Technol.* **2020**, *2020*, 304–326. [CrossRef]
35. Chen, C.; Asoni, D.E.; Barrera, D.; Danezis, G.; Perrig, A. HORNET: High-speed onion routing at the network layer. In Proceedings of the ACM SIGSAC Conference on Computer and Communications Security, Denver, CO, USA, 12–16 October 2015. [CrossRef]
36. Sankey, J.; Wright, M. Dovetail: Stronger anonymity in next-generation internet routing. In Proceedings of the International Symposium on Privacy Enhancing Technologies Symposium, Amsterdam, The Netherlands, 16–18 July 2014. [CrossRef]
37. Chen, C.; Perrig, A. Phi: Path-hidden lightweight anonymity protocol at network layer. *Proc. Priv. Enhancing Technol.* **2017**, *2017*, 100–117. [CrossRef]
38. Chen, C.; Asoni, D.E.; Perrig, A.; Barrera, D.; Danezis, G.; Troncoso, C. TARANET: Traffic-analysis resistant anonymity at the network layer. In Proceedings of the IEEE European Symposium on Security and Privacy, London, UK, 24–26 April 2018. [CrossRef]
39. Zhu, T.; Feng, D.; Wang, F.; Hua, Y.; Shi, Q.; Liu, J.; Cheng, Y.; Wan, Y. Efficient anonymous communication in SDN-based data center networks. *IEEE/ACM Trans. Netw.* **2017**, *25*, 3767–3780. [CrossRef]