Break the Blackbox! Desensitize Intra-domain Information for Inter-domain Routing

Peizhuang Cong[⋆,*], Yuchao Zhang[*✉], Lei Wang[⋆,*], Hao Ni[⋆,*],
Wendong Wang[⋆,*✉], Xiangyang Gong[⋆,*], Tong Yang[†], Dan Li[‡], Ke Xu[‡]
[⋆]State Key Laboratory of Networking and Switching Technology, Beijing, China
[*]Beijing University of Posts and Telecommunications, Beijing, China
[†]Peking University, Beijing, China [‡]Tsinghua University, Beijing, China

2022 IEEE/ACM 30th International Symposium on Quality of Service (IWQoS) | 978-1-6654-6824-4/22/$31.00 ©2022 IEEE | DOI: 10.1109/IWQoS54832.2022.9812918

***Abstract*—Along with the ever-increasing amount of data generated from edge networks, cross domain (also known as Autonomous Systems, AS) transmission problem has attracted more and more attention. As mature and widely used inter-domain routing protocols, BGP-based solutions often use the number of domains (i.e. AS hops) of each path to make inter-domain routing decisions, which is simple and effective, but usually can not get the optimal routing results due to the lack of real state/information within ASes. These protocols choose the path with less AS hops as the forwarding path, even if the total latency or cost of the domains on this path is higher. While to solve this problem, directly access to intra-domain information as the assistance to make routing decisions is impractical due to data privacy.**

In this paper, we propose DIT, which makes near-optimal inter-domain routing decisions with desensitized intra-domain information. To do so, we design a homomorphic encrypted-based private number comparison scheme to export intra-domain information securely and thus assist in routing decisions. We conduct a series of experiments according to five real network topologies with nearly 900 simulated flows, and the results show that DIT reduces the number of forwarding hops by about 45% in average and reduces flow completion time by about 60%.

***Index Terms*—inter-domain transmission, routing, private number comparison**

I. Introduction

With the development of the Internet, the number of network domains is ever increasing in recent years. Internet service providers, companies or institutes are building more and more small-scaled cross-geographical edge networks, which further expands the number of domains [1]. At the same time, with the maturity of 5G and Internet of things technologies, various applications e.g., government management, scientific research, business trade, have huge demands for large-scale data transmission across domains [2]. For example, the data aggregation between multiple data collection sites in meteorological research, or data backup between different data centers in commercial companies raise an urgent requirement on large data transmission.

The work was supported in part by the National Key R&D Program of China under Grant 2019YFB1802603, the National Natural Science Foundation of China (NSFC) under Grant 62172054, the Key Project of Beijing Natural Science Foundation under M21030, the NSFC under Grant 62072047. Peizhuang Cong's work was supported in part by BUPT Excellent Ph.D. Students Foundation under Grant CX2021232. ✉Corresponding author: Yuchao Zhang (ORCID: 0000-0002-0135-8915; e-mail: yczhang@bupt.edu.cn) and Wendong Wang (e-mail: wdwang@bupt.edu.cn).

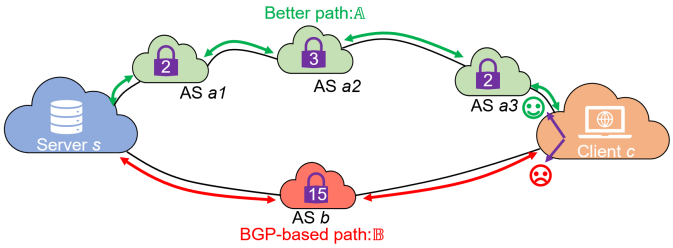


Figure 1. Example of BGP-based inter-domain routing

As a de facto inter-domain routing protocol, Border Gateway Protocol (BGP) is the most widely deployed protocol on the Internet, and there are many important enhancements to improve BGP performance in terms of refining scheduling granularity [3], accelerating convergence time [4], anomalous behavior detection [5] and so on. These BGP-like protocols follow the basic principle of taking hops - the number of Autonomous Systems (AS) on a path - as the metric for routing: the less hops, the higher the priority of the path [6]. Such strategy regards all domains as indiscriminate blackbox and thus can not achieve the optimal inter-domain routing decisions due to the lack of intra-domain information.

Take the forwarding hops as an example, Figure 1 shows two paths between server s and client c: Path $\mathbb{A}$ with 4 AS-hops ($s \rightarrow a1 \rightarrow a2 \rightarrow a3 \rightarrow c$) and Path $\mathbb{B}$ with 2 AS-hops ($s \rightarrow b \rightarrow c$). For client c, Path $\mathbb{B}$ will be selected as actually routing path according the principle of BGP, and Path $\mathbb{A}$ will be discarded. But in fact, there are additional hops in each domain, shown as the numbers in Figure 1, which makes path $\mathbb{A}$ the real better path.

To cope with the mentioned issue, there are some existing studies [7]–[9], e.g., RSA [10], SDI [11], MPC [12], and CIRCA [13], which leverage Software-Defined Networks (SDN) controller or a trusted centralized computation system to centrally schedule and route inter-domain traffic. However, these centralized strategies still require particular intra-domain information to assist in inter-domain routing decisions, which means that they can only be applied in specific scenarios with central management or with in one organization. What's

worse, such centralized strategies also suffer from poor scalability. These two drawbacks make centralized strategies fail to address the issue of intra-domain information unavailability problem. So how can we make use of intra-domain information? Can we leverage BGP to naively take the intra-domain information to diffuse to neighbor ASes? Such ideas are impractical because distributed ASes will refuse to provide their sensitive intra-domain information due to security concerns. Therefore, how to bridge the contradiction between data sharing and privacy protection is still challenging.

To this end, we propose a Desensitized Intra-domain information-aware Tactic (DIT) to assist inter-domain routing decisions, which can be embedded in BGP or applied independently as a control-plane strategy. DIT can make use of intra-domain information while protecting data privacy at the same time, thus solving the contradiction between data sharing and privacy protection. To be specific, DIT employs three tactical strategies, *abstraction*, *confusion*, and *comparison*, to guarantee security of intra-domain information exportation. DIT makes each domain as a full-mesh graph that only consists of boundary routers (nodes) and virtual connections (edges) with different weights between these nodes (*Abstraction*), which not only shields the information about intra-domain specific protocols and topologies but also retains the required intra-domain information (§IV-A-1). By adding different random numbers to the source domain information of each route (*Random Number Confusion*), DIT makes it possible to protect the data from being leaked at export and guarantee the correctness of the subsequent route computation (§IV-A-2). Then the proposed homomorphism-based encryption strategy is leveraged to compare priorities without exposing the data of each party (*Private Number Comparison*), which enables DIT to protect intra-domain information from being inferred during routing diffusion (§IV-C).

Based on five real network topologies with nearly 900 simulated flows, we incrementally deploy DIT as an independent control plane strategy to conduct a series of experiments. The results show that by making use of additional intra-domain information, DIT can enable BGP reduce about 45% end-to-end forwarding hops on average and about 60% the Flow Completion Time (FCT). To sum up, our contribution in this work can be summarized as follows:

- We disclose that traditional BGP-based routing protocols can not obtain the optimal routing decisions due to the lack of intra-domain information.
- We propose DIT, which for the first time applies homomorphic encryption to routing protocols, so as to make use of intra-domain information while still protecting data privacy.
- We demonstrate the practical benefits and improvements of DIT compared with traditional BGP-based protocols, by conducting series of emulation experiments on different scales in 5 real network topologies.

II. Background and Motivation

A. Border Gateway Protocol

BGP is the most widely deployed inter-domain routing protocol at present, which glues tens of thousands of worldwide ASes into the Internet. When routing information needs to be exchanged between ASes, each AS must designate a router running BGP, namely boundary gateway or boundary router, to be the entry and exit point for exchanging routing information with other ASes. The *segment length* value of *AS_PATH* attribute of each FIB item indicates the number of AS hops of the corresponding routing path, however, which completely neglects different transmission abilities of each AS.

BGP mainly includes four message types, `OPEN`, `UPDATE`, `NOTIFICATION`, and `KEEPALIVE`. The `UPDATE` is used to announce and withdraw route items, which contains three kinds of attributes related to routing path selection: *AS_PATH*, *MULTI_EXIT_DISC* (*MED*) and *LOCAL_PREF*. *AS_PATH* is used to keep track of which ASes a route has crossed during transmission. The router will reject all route entries that contain its own AS number, which can be used for loop-proof and also for path selection, i.e., the shorter the *AS_PATH* the better. *MED* is announced by neighbor AS to discriminate its multiple export ports. By default, for the same neighbor AS, the lower *MED*, the higher the priority of the export port. *LOCAL_PREF* is usually configured manually by the local administrator. When an AS has multiple egress routers, the router with the largest *LOCAL_PREF* value will be set as the egress.

Although manual configuration based on policies is a main way to inter-domain routing today, it still suffers from limitations, such as inflexibility of policies or misconfigurations, such as the worldwide service interruption of Meta in 2021 caused by an engineer's accidental configuration [14]. Automatic configuration based on network state has become one of the development trends. However, due to the lack of network-wide topology and global traffic view, traditional BGP can only follow the rule of minimal AS hops based on *AS_PATH* to develop routing policies, but with the explosive growth of network traffic and the diversification of transmission requirements, the shortcomings caused by the ignorance of intra-domain capabilities are becoming more and more obvious.

B. Motivation

Path	AS-path	Priority	State	Hops
[c, b, s]	2	high	selected	
[c, a3, a2, a1,s]	4	low	discarded	

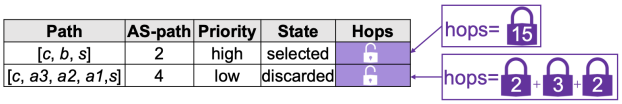


Figure 2. Route entries of Client c of Fig. 1

The routing entries for client c in Figure 1 are shown in Figure 2. Path $\mathbb{B}$ will be selected as the forwarding path due to the smaller *AS_PATH* value than that of path $\mathbb{A}$. However, Path $\mathbb{B}$ actually has 15 forwarding hops that is more than what Path $\mathbb{A}$ has, 5 ($\sum_{i=1}^{3} a_i$) forwarding hops only.

The solution to this issue is to export the intra-domain information. But can (and how?) export information without data leakage and ensure the effectiveness of final computation

results? If it is possible, then taking the accumulated intra-domain hops as an additional attribute in the local RIB would solve the above issue.

While this example is illustrative, we can see that the effect of intra-domain state on inter-domain transmission is significant, which means that the perception of intra-domain information is extremely necessary for inter-domain routing decision.

C. Homomorphic Encryption

Fortunately, homomorphic cryptography [15] can provide a potential solution to the contradiction of information exportation and privacy, it is a kind of cryptographic technique that performs arithmetic operations on the encrypted data and yields a result equivalent to the cyphertext result of some computation on the unencrypted original data. Its principle can be explained as follow:

$$De(En(a) \odot En(b)) = a \oplus b, \tag{1}$$

where $En()$ is the encryption operation, $De()$ is the decryption operation, and $\odot$ and $\oplus$ are correspond to the operations on the plaintext and cyphertext domains, respectively. When $\oplus$ represents addition, this encryption is an additive homomorphic encryption, and when $\oplus$ represents multiplication, this encryption is a multiplicative homomorphic encryption. The encryption function that satisfies both additive homomorphism and multiplicative homomorphism properties and can perform any times of additive and multiplicative operations is called fully homomorphic encryption.

Homomorphic encryption algorithms, especially full homomorphic encryption algorithms, suffer from high computational complexity. However, what we only need is to encrypt simple numbers and to satisfy the homomorphic additive property, so it is possible to circumvent the potential problems caused by high computational complexity of encryption algorithms.

Motivated by such encryption methods, we leverage Paillier [15], a classical additive homomorphic encryption algorithm, to facilitate privacy number comparison and its calculation process is presented in Section IV in detail.

III. Design Principle

To solve the contradiction between intra-domain information sharing and the information leakage risk in traditional BGP, an inter-domain routing protocol that can sense but not disclose the intra-domain information is desired. Accordingly, we clarify two requirements.

- **Information Export:** Data within a domain could be exported, mainly referring to the link performance status, e.g., delay, bandwidth, packet loss rate, hops, etc. The performance of an inter-domain transmission is jointly determined by the link performance of all passed domains, then, for different attributes, which can be summarized as bottleneck type (bandwidth) and cumulative type (delay, packet loss rate, hops), the calculation of combination will be different. In this paper, we take the number of hops as a typical example that ought to be calculated by addition.
- **Privacy Protection:** Private information of domains should not be deduced from the exported information, because information like hops may involve intra-domain topology, which requires that the information cannot be directly disclosed to other ASes.

We describe DIT in detail and explain how it meets the aforementioned requirements in the following section (§ IV).

IV. DIT Methodology

In this section, we first describe the basic version of DIT, which can be in embedded to BGP, and then we introduce its enhanced version, with integrity proof. Finally, we present the incremental deployment scheme of DIT, to further enhance its deployability.

A. DIT Overview

The judgment attributes of traditional BGP path selection do not include the impact of intra-domain performance, we propose to introduce an additional attribute, *Attr*, for BGP to accomplish data carrying and spreading. It is also possible to reuse existing properties, e.g., *Attr*, for deployment convenience.

*1) **Topology Abstraction**:*

BGP does not interfere with the intra-domain protocols running in each domain, such as RIP, OSPF, IS-IS, etc. This inherent property gives support to simplify the topology of inter-domain transmission. First, the intra-domain routing process is not interfered with by inter-domain routing protocols, which means each domain is a confidential system with complete independence and autonomy. Second, BGP runs at border routers of each domain and specifies the next hop at the border router granularity when forwarding across domains according to the RFC [6], which naturally shields intra-domain routing protocols. And, the transmission performance information between entry and exit border routers of a domain is sufficient to facilitate inter-domain routing decisions.

Therefore, we reasonably mask the intra-domain topology and abstract each domain into a characteristic topology graph with its border routers exclusively. Because of the reachability between routes within a domain, there are direct or indirect connections between all border routers (nodes) of a domain, and we abstract these connections as edges between nodes. As shown in the Figure 3. Most intra-domain protocols or SDN controllers usually maintain these point-to-point transmission performances, e.g., forwarding hops. In a domain with N boundary routers, the number of point-to-point transmission performance items to be maintained is $N(N-1)$ or $\frac{N(N-1)}{2}$, for unidirectional and bidirectional connections, respectively. Actually, N is usually a relatively small value, so maintaining this information additionally causes very limited overhead. Such abstraction not only protects the privacy of the details of the intra-domain but also preserves the necessary information.

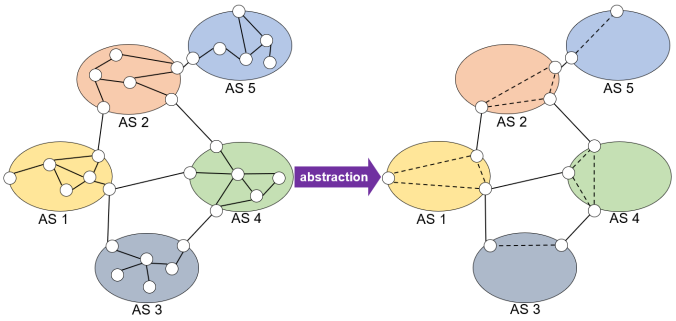


Figure 3. Abstraction illustration of domains topology

*2) **Random Number Confusion**:*

Assuming intra-domain information is directly embedded in BGP header and transmitted to the neighbors. Then, during the route convergence process, cumulative calculations (e.g., addition, $min()$ or $max()$) over multiple domains can inherently protect the privacy of all upstream domains data, i.e., mathematically speaking, on the basis of $c = a + b$, it could not infer the values of a and b when only c is known. This is one of the foundations for the privacy protection in DIT.

However, the inherent data privacy protection brought by cumulative calculations is effective only after at least one such operation has already been conducted. In other words, the cumulative calculations can only achieve non-destination domain data protection. For example, as shown in the Figure 3, for *AS 5*, the value of *AS 1* or *AS 2* cannot be inferred from the cumulative summation sent from *AS 2*. However, *AS 2* is directly connected to the destination domain of the route (*AS 1*), the value of *AS 1* is directly exposed to *AS 2* due to the lack of protection from cumulative calculation. That is, for the destination domain of each route, information leakage risk still exists, which is caused by directly connected neighbor domains, we name it the *Direct Connection* ($\leftrightarrow$) issue.

To solve *Direct Connection* ($\leftrightarrow$) issue, we propose a basic method named *Random Number Confusion*. In the path selection process, it is only necessary to select the optimal path by basic comparisons. According to the essential property of inequality, i.e., if $\exists a, \exists b \in \mathbb{R} \rightarrow a > b$, then $\forall c \in \mathbb{R} \rightarrow a+c > b + c$. Consequently, giving a random offset to all nodes in the coordinate system will not change the relative positions. So to the destination D, DIT adds a random number to the intra-domain data when initially diffusing the data to neighbor domains, i.e., D exports intra-domain data as:

$$d_i^{Exported} = d_i^{Local} + \delta_i^d, \tag{2}$$

where δ_i^d is used to confuse route i. Inconsistent δ^d can prevent inferring intra-domain information through multiple routes with different destinations of the same domain.

Therefore, DIT successfully solves *Direct Connection* ($\leftrightarrow$) issue by adopting cumulative calculation combined with *Random Number Confusion* and does not affect the comparison results of routing decisions.

*3) **Information Diffusion**:*

To carry the above intra-domain data, we add a new filed, *Attr*, to the BGP packet header, although which is not strictly required because we can also reuse existing fields, provided the re-definition of the field function is approved. And the quantified value of the destination-based cumulative path performance is embedded into this field and diffused to neighbor domains with the route update message.

DIT does not constrain the intra-domain switching policy implemented by each domain. See a typical process of inter-domain routing message diffusion is shown in Figure 4, where *AS B* runs a traditional routing protocol and *AS C* uses a central controller similar to an SDN controller. We still take the number of hops as example. Suppose *AS D* updates the route of *d0*, then based on the intra-domain topology information, *d1* sends this update message to *AS B* (*b2*) and *AS C* (*c2*), where the *Attr* value is $12 = \delta^d + 2$. The router compares the *Attr* to decide whether to update the local RIB. When the received *Attr* is smaller than the local, the route entry will be updated, otherwise it will be kept unchanged. In the intra-domain, *AS B* or *AS C* exchange update messages using the intra-domain protocol. *AS C* (*c3*) sends this update to *AS B* (*b3*), where the *Attr* value is the number of hops of the optimal path from *c3* to *c2* (*c3* $\rightarrow$ *c1* $\rightarrow$ *c2*) plus the *Attr* value received by *c2* ($21 = 3+6+12$). For *AS B* (*b3*), the received *Attr* is greater than the local, so the local RIB is not updated. Similarly, the *Attr* of the update message sent by *AS B* to *AS C* is $14 = 2 + 12$, which is smaller than local value, then updates the corresponding route entry. *AS B* (*b1*) and *AS C* (*c1*) sent update message to *AS A* (*a1*). Then, *AS A* updates the optimal path for reaching *d0* in *AS D* based on the two messages received from *AS B* and *AS C*, in which *Attr* is 19 and 17, respectively. For example, for *a1*, it will choose *c1* as next hop to *d0* because the corresponding path has a smaller *Attr* value.

For different attributes, the cumulative calculation of path performance will be different [16]. In addition to the cumulative formula given as the above cumulative-type, where we use hops as an example, there are other bottleneck-types, too. We can also use bandwidth as path selection constrain and use $max(min())$ function to calculate.

B. Delta Trap

While *Random Number Confusion* solves the destination direct connection issue, there is still a trap of information leakage. Let's start by drawing this trap from a mathematical perspective. Suppose it is known that $x_1 + x_2 = y_1$ and $x_1 + x_2 + x_3 = y_2$. Even if x_1 and x_2 are unknown, x_3 can also be calculated by using the difference value (Δ) between y_1 and y_2, i.e., $x_3 = y_2 - y_1$. Mapping this description to DIT scenario, as shown in the Figure 3, *AS 4* can receive routes with the same destination from *AS 1* and *AS 3*, respectively, and the corresponding values have been confused by random numbers in the destination AS. Assuming $V(AS_i)$ is the summation of a route that *AS i* sent to its neighbors. *AS 1* will receive $V(AS_1)$ and $V(AS_3)$, where $V(AS_1)$ is the cumulative summation of some previous ASes that from *AS 1*

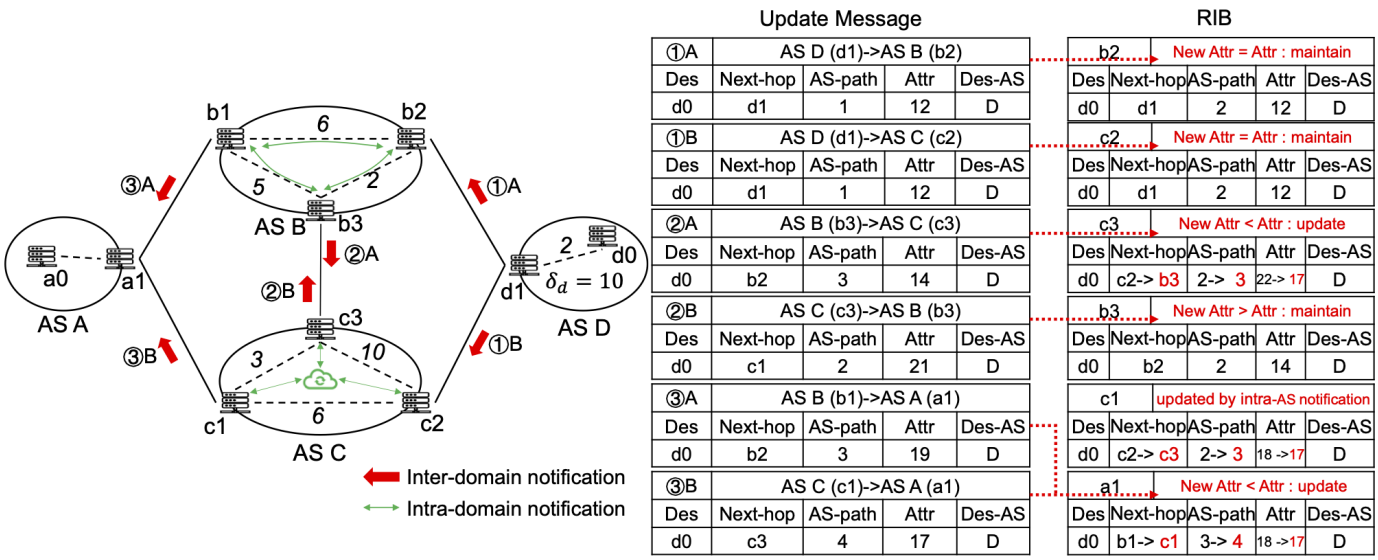


Figure 4. Diffusion example: diffusion of DIT update messages and RIB updates triggered by a new route.

to destination, and $V(AS_3)$ is the cumulative summation of same previous ASes excluding *AS 3*.

Then, for *AS 4*, the value of *AS 3* can be obtained using the aforementioned difference value (Δ) method. *AS 3* cannot use the random number confusion for some routes where *AS 3* is not the destination, because it may affect the result of the numerical comparison, i.e., $a > b \nrightarrow a > b + c$. We call this situation the *Delta Trap* (Δ).

To patch the pit of intra-domain leakage caused by *Delta Trap* ($\triangle$), we propose a homomorphic encrypted-based *Private Number Comparison* strategy to enhance DIT.

C. Enhanced DIT

Delta Trap (Δ) is triggered by one path has one more hop (itself) than the other of same destination. Illustratively, from perspective of connection topology, triangular structure is at risk of data leakage. Provided that the path selection is not affected, this trap can be avoided if the triangle in the topology can be disassembled, i.e., the edge that is not on the shortest path of triangle can be removed logically.

Based on this, we design a private number comparison algorithm leveraged by homomorphic encryption, which is capable of comparing paths in a triangle topology under guarantee of data security. The comparison result could guide the logical removal of non-shortest paths. In this subsection, we introduce the classic homomorphic encryption algorithm we use, Paillier, and detail the private number comparison algorithm.

*1) **Homomorphic Encryption**:*

Any cryptosystem includes private keys and public keys, which are used to encrypt and decrypt the data, respectively. We first describe the Paillier algorithm in terms of three processes: key generation, encryption, and decryption, and then illustrate its additive homomorphism characteristics [15].

- *Key Generation*: Randomly selecting two large prime numbers p and q that satisfy $gcd(pq, (p-1)(q-1)) = 1$, and calculating $n = pq$ and $\lambda = lcm(p-1, q-1)$. And randomly selecting integer $g \in \mathbb{Z}^*_{n^2}$, and calculating $\mu = (L(g^\lambda \bmod n^2))^{-1} \bmod n$, where $L(u) = \frac{u-1}{u}$, for $\forall u \in \{u < n^2 \mid u = 1 \bmod n\}$. Then, the public key is (n, g) and private key is (λ, μ).
- *Encryption*: For plaintext $m \in \mathbb{Z}^*_n$, its encrypted ciphertext is $c = g^m \cdot r^n \bmod n^2$.
- *Decryption*: For ciphertext $c \in \mathbb{Z}^*_{n^2}$, its decryped plaintext is $m = L(c^\lambda \bmod n^2 \cdot \mu) \bmod n$.

Assuming that $r_1, r_2 \in \mathbb{Z}^*_{n^2}$ are two random integers, for the plaintext m_1, m_2, their ciphertext are $En(m_1) \equiv g^{m_1} \cdot r_1^n \bmod n^2$ and $En(m_2) \equiv g^{m_2} \cdot r_2^n \bmod n^2$, respectively. Then,

$$\begin{aligned} En(m_1) \cdot En(m_2) &\equiv g^{m_1} \cdot r_1^n \cdot g^{m_2} \cdot r_2^n \bmod n^2 \\ &\equiv g^{m_1+m_2} \cdot (r_1 \cdot r_2)^n \bmod n^2 \\ &\equiv En(m_1 + m_2) \end{aligned}$$

As $r_1, r_2 \in \mathbb{Z}^*_{n^2}$, then $r_1 \cdot r_2 \in \mathbb{Z}^*_{n^2}$, so the Paillier cryptosystem is additive homomorphic. Based on Paillier, we propose the private number comparison to patch the leakage caused by *Delta Trap* (Δ), which is used as an independent module of DIT.

*2) **Private Number Comparison**:*

Detecting Traps. First, the triangle structures needs to be detected from the network topology. This process can be initiated and maintained by `KEEPLIVE` of BGP. It is specified that the message contains the list of neighboring domains of this domain, and after exchanging information between domains, the relevant triangle topology can be obtained according to Algorithm 1. It is noted here that since the path attribute of BGP includes the AS list, the process does not require other additional information involved in traditional BGP.

Algorithm 1: Trap detection

neighbors(): return all neighbor domains
Input: neighbors list
Output: the triangle list of S
for i *in* S*.neighbors()* **do**
  for j *in* i*.neighbors()* **do**
    $triangle\_list$.append([S, i, j]);
return $triangle\_list$

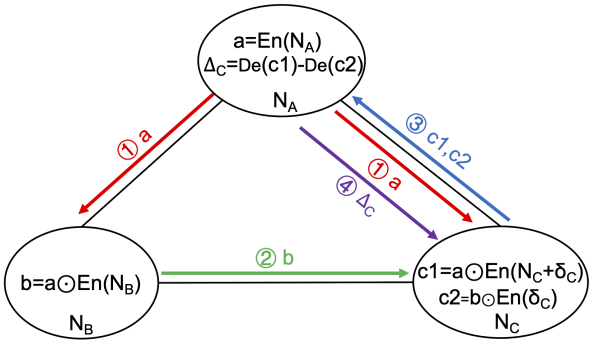


Figure 5. Comparison example: communication and computation process of homomorphic encryption-based private number comparison.

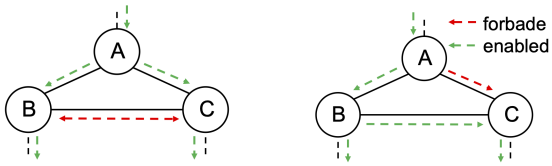
Although Algorithm 1 is based on the case of a uni-directional link between domains, i.e., $state(A \rightarrow B) \neq state(B \rightarrow A)$, it still works under bidirectional link case, i.e., $state(A \rightarrow B) \equiv state(B \rightarrow A)$, where the duplicate triangles can be directly ignored. The process is relatively stable, and this triangle relationship remains constant as long as the connectivity state is constant.

Comparing Paths. In the generic triangular topology (Figure 5), comparison and path selection would be accomplished by communicating with each other, which is described as pseudo code Algorithm 2.

Suppose *A*, *B* and *C*, each of which is responsible for local values, N_A, N_B, N_C, respectively. First, *A* sends encrypted N_A by private key of *A*, $En^A(N_A)$, to *B* and *C*. After receiving the message from *A*, *B* sends $En^A(N_A) \odot En^A(N_B)$ to *C*, where $\odot$ represents homomorphic addition calculation, which means $En(x) \odot En(y) \equiv En(x+y)$. After receiving the message from *A* and *B*, *C* sends $En^A(N_A+N_B) \odot En^A(\delta_C)$ and $En^A(N_A) \odot En^A(N_C + \delta_C)$ to *A* in the specified order. After receiving the message from *C*, *A* decrypts and subtracts the two values, $De^A(En^A(N_A + N_B + \delta_C)) - De^A(En^A(N_A + N_C + \delta_C))$, and get the signed delta value Δ_C, which will be sent back to *C*. Finally, according to Δ_C, *C* and *A* can determine the priority of the two paths, $Path_{(C \rightarrow A)}$ and $Path_{(C \rightarrow B \rightarrow A)}$.

The reason why *A* has to send N_A to *B* and *C* is that the cost of *C* or *B* through *A* to the same border router of *A* during inter-domain transmission may be different, i.e., the N_A sent by *A* to *B* and *C* is the respective corresponding cost, and this comparison algorithm is still feasible.

The confidentiality of the entire comparison process is explained here. First, the value sent by *A* to *B* and *C* is encrypted and cannot be decrypted by *B* and *C* without the private key, and likewise, the value sent by *B* to *C* cannot be deciphered. The malicious case of forcing to break the encryption algorithm is not considered here. The two values sent by *C* to *A* use the random number mix strategy to add a random number, which makes *A* cannot get the relevant data except Δ_C after decryption.



(a) (b)

Figure 6. Two types of diffusion constraint

Constraining Diffusion. When receives the BGP message which causes an `UPDATE` from upstream, *A* can specify the downstream of subsequent transmission of the `UPDATE` message. For cases that the direct connection is the optimal path, as shown in the Figure 6(a), then *A* directly diffuses the message and uses an identifier, such as $TAG_{\triangle} = 1$, to notify the downstream not to notify another domain in the triangle of this update. For the case that the direct connection, such as to *C*, is not optimal, as shown in the Figure 6(b), then *A* does not notify the update message to *C*. The downstream that gets the update, i.e., *B*, will notify this update to *C* according to $TAG_{\triangle} = 0$. Then for *C*, the path notified by *B* do be the optimal one.

D. Completeness Analysis of Privacy

The enhancement of inter-domain routing by leveraging intra-domain information requires protecting the data of each domain from being obtained by others, which can be modeled as an equation solving problem. During the convergence of a route, domain S accumulates maintained local transmission performance $cost_S$ (e.g., forwarding hops) on the *Attr* coming upstream and spreads downstream. Thus, each domain can obtain an equation $Attr_i = \sum_J^{AS\_PATH_i} cost_J$ based on the *Attr* and *AS_PATH* of route i. Privacy protection aims to prevent any domain from inferring any $cost_J$ from a series of equations generated by different routes of local RIB. In the following, we first mathematically model the problem and then prove the privacy completeness of the DIT.

*1) **Formulation**:*

We define the cumulative cost for domain S of being forwarded by its border router j to domain D is:

$$COST_S^{j \rightarrow D} \tag{3}$$

Then, based on different n border routers, S can obtain the set of equations $\mathbb{C}$:

$$COST_S^{i \rightarrow D} = y_i, i \in n, \tag{4}$$

Algorithm 2: Comparison

En(x, key): encrypt x with private key
De(x, KEY): decrypt x with public KEY
sent([x_1, x_2], [D_1, D_2]): send $[x]$ to $[D]$
Input: the triangle (A, B, C)
Output: the comparison result from triangle (A, B, C)

A **Zone:**
  ena=En(N_A, key_A)
  A.sent(ena, [B, C])// tagged message 1

B **Zone:**
  na = B.receive()// message 1
  enb = $na \odot$ En(N_B, key_A)
  B.sent(enb, C)// tagged message 2

C **Zone:**
  na = C.receive()// message 1
  nb = C.receive()// message 2
  enc = $na \odot$ En($N_C + \delta_C$, key_A)
  enb = $nb \odot$ En(δ_C, key_A)
  C.sent([enc, enb], A)// tagged message 3

A **Zone:**
  nc, nb = A.receive() // message 3
  nb = De(nb, KEY_A)
  nc = De(nc, KEY_A)
  Δ_C = nc - nb
  A.sent(Δ_C, C) // tagged message 4

C **Zone:**
  Δ_C = C.receive()// message 4

where y_i is the value of *Attr* of each related route i. Each $COST_S^{j \to D}$ can be expressed in the form of a cumulative sum of the $cost$ of the domains contained in *AS_PATH* of route i. So the set $\mathbb{C}$ can be converted as:

$$cost^D_{i^{AS_0}} + cost^D_{i^{AS_1}} + ... + cost^D = y_i, i \in n, \quad (5)$$

where $cost^D_{i^{AS_j}}$ represents the cost of the j-*th* domain of the path forwarded by the border router i to domain D. Intra-domain data leakage occurs when any $cost$ can be inferred from $\mathbb{C}$.

*2) **Mathematical Analysis**:*

Each equation in $\mathbb{C}$ is non-linearly dependent. Then, for any subset $\mathbb{C}' \subseteq \mathbb{C}$, the elements can be inferred if the number of unknown elements equals to the number of equations after elimination transformation of $\mathbb{C}'$. Due to the existing of $cost^D = y_i \in \mathbb{C}$, i.e., the *Direct Connection* can get the $cost^D$ directly. The *Random Number Confusion* implicitly extends the number of unknown elements, $cost^D + \delta^d$, which protects the $cost^D$ from being inferred.

For routes whose destinations are not in the same domain, each appended equation will introduce at least one additional unknown element to $\mathbb{C}'$ because the different domains that the route traverses. For routes whose destinations are in the same domain but different addresses, *Random Number Confusion* can guarantee to introduce at least one unknown random to $\mathbb{C}'$. That is, routes with different destinations cannot assist each other in inferring inter-domain information.

The remaining cases, i.e., routes to the same destination, can be divided into two categories. The first is that there is no intersection domain in the *AS_PATH* of routes in $\mathbb{C}'$ except for the the destination domain. At this case, for any $\mathbb{C}'$, the number of elements is greater than the number of equations, so no element can be inferred. The second case is that there are additional intersection domains. First, we assume that the paths before the intersection domain are different, i.e., there can be multiple paths to the destination domain from the intersection, which contradicts the principle that each domain will only choose one optimal path to the destination domain, i.e., the assumption is not valid. Then, it can be concluded that if some paths have intersection domains, then all domains passed before the intersection domain of these paths are the same.

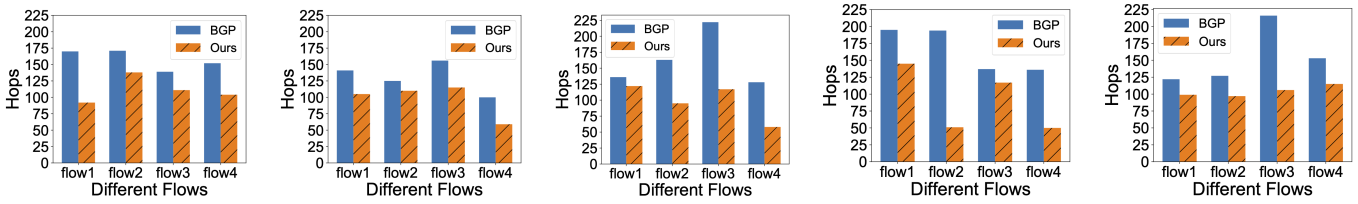
On the basis of this, we represent the domains before the intersection as $\sum_{j=1}^{D} \text{cost}_j^D$. Then, the equation corresponding to the paths with intersection, $\mathbb{K}$, can be converted as:

$$cost^D_{l^{AS_0}} + cost^D_{l^{AS_1}} + ... + \sum_{j=1}^{D} \text{cost}_j^D = y_l, \ l \in \mathbb{K}, \quad (6)$$

For equation (6), **iff** $\exists a \in \mathbb{K} \to \sum_{j=1}^{D} \text{cost}_j^D = y_t$, and $\exists b \in \mathbb{K} \to cost^D_{b^{AS_0}} + \sum_{j=1}^{D} \text{cost}_j^D = y_b$, it can uniquely infer the value of an unknown quantity. That is, $cost^D_{b^{AS_0}} = y_t - y_b$. This situation corresponding to the aforementioned *Delta Trap*, which can be solved by *Private Number Comparison*. For all other situations in this case, the number of elements is greater than the number of equations, thus no element can be inferred. In conclusion, the privacy of DIT can be guaranteed.

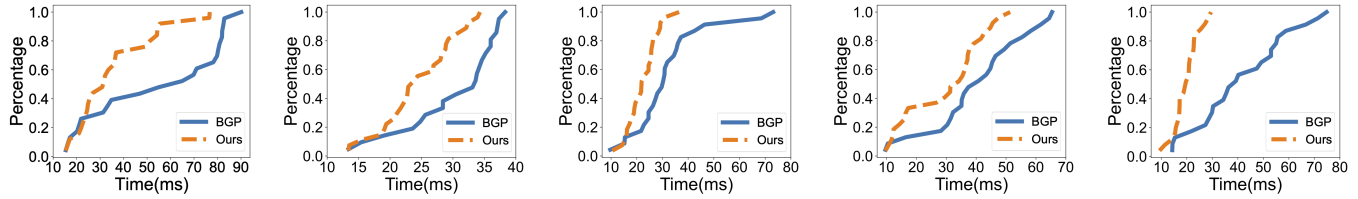
E. Incremental Deployment Discussion

Although embedding into BGP makes DIT globally optimal, it still faces the issue of coordinating all domains to run DIT, and the deployment is difficult to be done overnight at the initial stage. Therefore, we propose an incremental deployment solution. The essence of DIT is to desensitize the intra-domain transmission performance information and share it with other domains for inter-domain decision-making. So we can leverage other layer protocols for domain-to-domain data transfer to make inter-domain routing decisions in the control plane, which allows the convergence of DIT over domains that do not run DIT. It is possible to combine DIT with the *AS_PATH* attribute of the existing BGP for the domains that run DIT to make inter-domain decisions. For example, counting the percentage of domains in the *AS_PATH* that do not run DIT, the higher it is the lower the priority of the route; or using the average performance of domains running DIT in *AS_PATH* as the performance of domains not running DIT; or directly avoiding routing paths with poorly performing domains. These above strategies may lead to some traffic bypassing domains that are not running DIT, which will partly affect economic benefits of these domains. Thus, from both of



(a) ATMnet (b) Claranet (c) Compuserve (d) NSFnet (e) Peer1

Figure 7. Hops improvements under 5 topologies



(a) ATMnet (b) Claranet (c) Compuserve (d) NSFnet (e) Peer1

Figure 8. Latency improvements under 5 topologies

business perspective and network transmission performance improvement objective, DIT is worth being deployed despite the tiny deployment cost [17].

V. Evaluation

In this section, we clarify the experimental setup and analyze the comprehensive performance of DIT with BGP. As existing distributed-based policies do not utilize intra-domain information for routing, and the centralized-based strategy would involve information leakage, here we can only compare DIT with the classic BGP.

A. Experiment Setup

The network simulation is implemented by dce-ns3-dev version of NS3 on the Ubuntu 16.04.7-LTS operating system. The server is equipped with 8G of RAM, dual core Intel(R) Core(TM) i5-6300HQ 2.30GHz CPU, and 128 GB hard disk. And five real network topologies, *ATMnet, Claranet, Compuserve, NSFnet, and Peer1*, were selected from *Topology Zoo* [18] as the experimental topology.

B. Experiment Evaluation

1) ***Hops Improvement****:*

For the five topologies set up above, we randomly generate four end-to-end flows in each topology. The four flows in each topology are the ones where the source and destination nodes are farther apart, which ensures that there are more reachable paths between them and better reflects the performance difference between DIT and BGP routing decisions.

We measured the end-to-end hops all flow under traditional BGP and DIT, respectively, as shown in the Figure 7(a) to Figure 7(e), it can be seen that DIT yields better routing policies than traditional BGP whenever there are more than one diverge paths of inter-domain.

To clarify here, in this set of experiments, we did not generate the full-set of end-to-end flows in each topology. This is because DIT and BGP would have the same performance for adjacency or only one reachable path for inter-domain transmission. And it is also not the transmission scenario that DIT focuses on to optimize.

2) ***Latency Improvement****:*

In the implementation of DIT, although not the latency but the forwarding hops is used as the optimization target (which could be achieved by replacing hops information stored in *Attr* with latency information), the increment of hops during transmission could improve the total processing latency at switches, so reducing the forwarding hops can also lead to optimization in terms of latency.

For the performance improvement of DIT over BGP in terms of latency, in these five topologies, the respective FCT of DIT and BGP are measured for the same set of large number of end-to-end flows. According to the results illustrated in Figure 8(a) to Figure 8(e), it can be seen that the flow completion times of DIT is better than BGP, which means that DIT can improve the performance in terms of transmission latency while optimizing the forwarding hops.

3) ***Impact of Flows****:*

To further evaluate the performance of DIT, we generated almost 800 end-to-end inter-domain flows between any two end for the topology *NSFnet*, and measured the completion time of these flows under DIT and BGP separately.

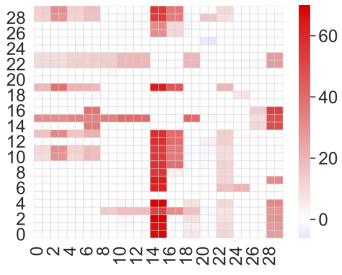
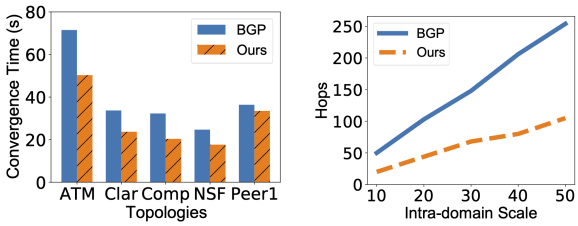


Figure 9. FCT of DIT and BGP under point-to-point inter-domain flows

It is explained here since DIT definitely outperforms BGP in terms of hops, so we further demonstrate the DIT performance improvement on latency.



(a) Convergence Time (b) Intra-domain scales

Figure 10. Protocol convergence and Impact of Domain Scales

The flow completion time difference values (BGP minus DIT) are shown in the Figure 9. It can be seen that DIT can achieve better than or at least equal to BGP for almost all flows. The uncertain fluctuations of the network, e.g., congestion, packet loss, make it reasonable that it cannot guarantee to boost all flows, so, as depicted in the Figure 9, there are only 0.5% flows that DIT performs slightly weaker than BGP. The results can indicate that DIT overall outperforms BGP.

*4) **Impact of Intra-domain Scale**:*

We measured the performance improvement of DIT over BGP for different intra-domain scales. Each domain is set to 10 to 50, respectively, for different scales scenarios. And there exist a fraction of domains with worse performance, i.e., those with more hops.

According to the transmission of the flows generated by the above conditions, the performance of DIT and BGP in terms of the forwarding hops is shown in Figure 10(b). It can be seen that the performance improvement of DIT is more obvious under the condition of large domain scale, which can reduce up to 60% end-to-end average forwarding hops. This is since the overall performance of an inter-domain transmission path is affected by the cumulative impact of each passed domain. As the overall scale of the domain becomes larger the performance gap between domains also becomes wider, ultimately leading to an increased cumulative impact.

*5) **Convergence and Cryptogram Overhead**:*

The results of measuring convergence time in the five topologies are shown in the Figure 10(a), which indicates that DIT outperforms BGP. Since DIT reuses the diffusion mechanism of BGP with adding diffusion restrictions, and additional encryption, homomorphic computation and decryption operations are the pre-determined process which independent of route diffusion and convergence.

We consider the three steps of encryption, homomorphic calculation and decryption as one operation. And by randomly selecting the corresponding numbers in the prime number array that calculated in advance, the calculation efficiency of secret&public keys generation can be improved. The *Paillier* algorithm, implemented in Python, takes about $30\,ms$ for each operation averaged over 100,000 operations. And the *Paillier* algorithm implemented with the NTL library, similarly averaged over 100,000 operations, takes less than $0.1\,ms$ per operation. Although the comparison phase includes computational overhead and communication overhead, the distributed architecture makes the comparison phase fully parallel and independent of the route diffusion.

VI. Related Work

A. Enhancement of BGP

Although BGP is the de facto inter-domain routing protocol, there still is room for further improvement of BGP in several aspects, e.g., convergence, security [4], [19], [20]. Mattia et al. propose an ASes topological position-based strategy to adjust Minimum Route Advertisement Interval (MRAI) setting to enhance the convergence of BGP, which is verified by a testbed-based experimental analysis [21]. Juan et al. exploit the asymmetric distribution of traffic and prioritize prefixes differently based on traffic prediction to reduce the traffic loss issue during BGP reconvergence [3]. Alberto et al. present a method to measure the propagation time of BGP routes with BGP route collectors and beacons, which could handle the clock offset between route source and destination [22]. There are also studies that combine the deep learning with BGP from the perspective of security, configuration, etc [23]–[25].

B. Inter-domain Routing Schemes

Many inter-domain routing schemes have been proposed to improve route control [26]–[31], and all designs can be divided into two categories. The first is third-party composition [10], [12], [13], [32]–[35]. Xiang et al. present a Software-Defined Internetworking (SDI) model, in which a network exposes a programmable interface to allow clients to define the inter-domain routes of the network [11]. Pouryousef et al. present CIRCA, which will upload the data of this region to the cloud, through the cloud's high bandwidth, sufficient computing resources and other advantages to calculate the cross-domain routing path and sent down to the ground [13]. Zhao et al. present a scalable multi-agent Reinforcement Learning method via customer-provider multi-layer structure for inter-domain routing [35]. However, the premise of this category scheme is the need for third-party cloud management agencies to ensure that they will not snoop and leak sensitive data, which is also an ideal state, the actual feasibility is controversial. The second is tunnel-based overlay [36]–[40]. And the basic idea is to let a stub AS interact with a remote AS to select routes different from the BGP route, and then build a tunnel between stub and remote ASes to utilize the negotiated routes. As such, these systems have data path overhead such as tunneling processing on each data packet. However, the feature that tunnel information is not exposed to other ASes may lead to security issues, which makes it difficult to be accepted by network organizations.

VII. Conclusion

In this paper, we disclose the effect of intra-domain state on inter-domain routing decisions, which is always ignored in the existing inter-domain transmission protocols. However, it is not easy to use intra-domain information securely. To solve this problem, we propose DIT, which can make better inter-domain routing than BGP by utilizing intra-domain information with no data leakage. In DIT, we design homomorphic encryption-based private number comparison strategies

to ensure the security of the intra-domain information, so as to avoid privacy data leakage caused by the intra-domain information export. Through a series of experiments, DIT has been shown to outperform BGP-based solutions, and it reduces about 45% end-to-end forwarding hops on average and reduces about 60% of the flow completion time. Such performance improvements will even be more prominent in large-scale networks.

REFERENCES

[1] G. Huston, "Bgp routing table analysis reports," https://bgp.potaroo.net, 2021.

[2] U. Cisco, "Cisco annual internet report (2018-2023) white paper," *www.cisco.com*, 2020.

[3] J. Brenes, A. García-Martínez, M. Bagnulo, A. Lutu, and C. Pelsser, "Power prefixes prioritization for smarter bgp reconvergence," *IEEE/ACM Transactions on Networking*, vol. 28, no. 3, pp. 1074–1087, 2020.

[4] B. Al-Musawi, P. Branch, and G. Armitage, "Bgp anomaly detection techniques: A survey," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 1, pp. 377–396, 2017.

[5] R. Fezeu and Z. L. Zhang, "Anomalous model-driven-telemetry network-stream bgp detection," in *2020 IEEE 28th International Conference on Network Protocols (ICNP)*, 2020.

[6] Y. Rekhter, T. Li, and S. Hares, "Rfc 4271: A border gateway protocol 4 (bgp-4)," *[Online] https://datatracker.ietf.org/doc/html/rfc4271*, 2006.

[7] S. Jain, A. Kumar, S. Mandal, J. Ong, L. Poutievski, A. Singh, S. Venkata, J. Wanderer, J. Zhou, M. Zhu *et al.*, "B4: Experience with a globally-deployed software defined wan," *ACM SIGCOMM Computer Communication Review*, vol. 43, no. 4, pp. 3–14, 2013.

[8] Z. Yang, Y. Cui, B. Li, Y. Liu, and Y. Xu, "Software-defined wide area network (sd-wan): Architecture, advances and opportunities," in *2019 28th International Conference on Computer Communication and Networks (ICCCN)*, 2019.

[9] P. Lin, J. Bi, S. Wolff, Y. Wang, A. Xu, Z. Chen, H. Hu, and Y. Lin, "A west-east bridge based sdn inter-domain testbed," *Communications Magazine IEEE*, vol. 53, no. 2, pp. 190–197, 2015.

[10] K. Lakshminarayanan, I. Stoica, S. Shenker, and J. Rexford, *Routing as a Service*. Citeseer, 2004.

[11] Q. Xiang, J. Zhang, K. Gao, Y.-s. Lim, F. Le, G. Li, and Y. R. Yang, "Toward optimal software-defined interdomain routing," in *IEEE INFOCOM 2020-IEEE Conference on Computer Communications*. IEEE, 2020, pp. 1529–1538.

[12] G. Asharov, D. Demmler, M. Schapira, T. Schneider, G. Segev, S. Shenker, and M. Zohner, "Privacy-preserving interdomain routing at internet scale." *Proc. Priv. Enhancing Technol.*, vol. 2017, p. 147, 2017.

[13] S. Pouryousef, L. Gao, and A. Venkataramani, "Towards logically centralized interdomain routing," in *17th USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, 2020, pp. 739–757.

[14] B. Barrett, "Why facebook, instagram, and whatsapp all went down today," *Wired*, 2020.

[15] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," in *International conference on the theory and applications of cryptographic techniques*. Springer, 1999, pp. 223–238.

[16] P. Cong, Y. Zhang, Z. Liu, T. Baker, H. Tawfik, W. Wang, K. Xu, R. Li, and F. Li, "A deep reinforcement learning-based multi-optimality routing scheme for dynamic iot networks," *Computer Networks*, vol. 192, p. 108057, 2021.

[17] D. Gupta, A. Segal, A. Panda, G. Segev, M. Schapira, J. Feigenbaum, J. Rexford, and S. Shenker, "A new approach to interdomain routing based on secure multi-party computation," in *Proceedings of the 11th ACM Workshop on Hot Topics in Networks*, 2012, pp. 37–42.

[18] Zoo, "The internet topology zoo," http://www.topology-zoo.org/, 2021.

[19] Q. Li, J. Liu, Y.-C. Hu, M. Xu, and J. Wu, "Bgp with bgpsec: Attacks and countermeasures," *IEEE Network*, vol. 33, no. 4, pp. 194–200, 2018.

[21] M. Milani, M. Nesler, M. Segata, L. Baldesi, L. Maccari, and R. L. Cigno, "Improving bgp convergence with fed4fire+ experiments," in *IEEE INFOCOM 2020-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. IEEE, 2020, pp. 816–823.

[20] M. Chiesa, A. Kamisiński, J. Rak, G. Rétvári, and S. Schmid, "A survey of fast-recovery mechanisms in packet-switched networks," *IEEE Communications Surveys & Tutorials*, vol. 23, no. 2, pp. 1253–1301, 2021.

[22] A. García-Martínez and M. Bagnulo, "Measuring bgp route propagation times," *IEEE Communications Letters*, vol. 23, no. 12, pp. 2432–2436, 2019.

[23] M. Bahnasy, F. Li, S. Xiao, and X. Cheng, "Deepbgp: a machine learning approach for bgp configuration synthesis," in *Proceedings of the Workshop on Network Meets AI & ML*, 2020, pp. 48–55.

[24] K. McGlynn, H. Acharya, and M. Kwon, "Detecting bgp route anomalies with deep learning," in *IEEE INFOCOM 2019-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. IEEE, 2019, pp. 1039–1040.

[25] A. Lutu, M. Bagnulo, C. Pelsser, O. Maennel, and J. Cid-Sueiro, "The bgp visibility toolkit: Detecting anomalous internet routing behavior," *IEEE/ACM Transactions on Networking*, vol. 24, no. 2, pp. 1237–1250, 2015.

[26] A. Gupta, R. MacDavid, R. Birkner, M. Canini, N. Feamster, J. Rexford, and L. Vanbever, "An industrial-scale software defined internet exchange point," in *13th USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, 2016, pp. 1–14.

[27] R. R. Sambasivan, D. Tran-Lam, A. Akella, and P. Steenkiste, "Bootstrapping evolvability for inter-domain routing with d-bgp," in *Proceedings of the Conference of the ACM Special Interest Group on Data Communication*, 2017, pp. 474–487.

[28] J. L. Sobrinho, D. Fialho, and P. Mateus, "Stabilizing bgp through distributed elimination of recurrent routing loops," in *2017 IEEE 25th International Conference on Network Protocols (ICNP)*. IEEE, 2017, pp. 1–10.

[29] T. Holterbach, S. Vissicchio, A. Dainotti, and L. Vanbever, "Swift: Predictive fast reroute," in *Proceedings of the Conference of the ACM Special Interest Group on Data Communication*, 2017, pp. 460–473.

[30] B. Schlinker, H. Kim, T. Cui, E. Katz-Bassett, H. V. Madhyastha, I. Cunha, J. Quinn, S. Hasan, P. Lapukhov, and H. Zeng, "Engineering egress with edge fabric: Steering oceans of content to the world," in *Proceedings of the Conference of the ACM Special Interest Group on Data Communication*, 2017, pp. 418–431.

[31] K.-K. Yap, M. Motiwala, J. Rahe, S. Padgett, M. Holliman, G. Baldus, M. Hines, T. Kim, A. Narayanan, A. Jain *et al.*, "Taking the edge off with espresso: Scale, reliability and programmability for global internet peering," in *Proceedings of the Conference of the ACM Special Interest Group on Data Communication*, 2017, pp. 432–445.

[32] A. Gupta, L. Vanbever, M. Shahbaz, S. P. Donovan, B. Schlinker, N. Feamster, J. Rexford, S. Shenker, R. Clark, and E. Katz-Bassett, "Sdx: A software defined internet exchange," *ACM SIGCOMM Computer Communication Review*, vol. 44, no. 4, pp. 551–562, 2014.

[33] R. Birkner, A. Gupta, N. Feamster, and L. Vanbever, "Sdx-based flexibility or internet correctness? pick two!" in *Proceedings of the Symposium on SDN Research*, 2017, pp. 1–7.

[34] A. Dethise, M. Chiesa, and M. Canini, "Prelude: Ensuring inter-domain loop-freedom iñ sdn-enabled networks," in *Proceedings of the 2nd Asia-Pacific Workshop on Networking*, 2018, pp. 50–56.

[35] X. Zhao, C. Wu, and F. Le, "Improving inter-domain routing through multi-agent reinforcement learning," in *IEEE INFOCOM 2020-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. IEEE, 2020, pp. 1129–1134.

[36] W. Xu and J. Rexford, "Miro: Multi-path interdomain routing," in *Proceedings of the 2006 conference on Applications, technologies, architectures, and protocols for computer communications*, 2006, pp. 171–182.

[37] X. Yang, D. Clark, and A. W. Berger, "Nira: a new inter-domain routing architecture," *IEEE/ACM transactions on networking*, vol. 15, no. 4, pp. 775–788, 2007.

[38] H. Wang, Y. R. Yang, P. H. Liu, J. Wang, A. Gerber, and A. Greenberg, "Reliability as an interdomain service," *ACM SIGCOMM Computer Communication Review*, vol. 37, no. 4, pp. 229–240, 2007.

[39] S. Peter, U. Javed, Q. Zhang, D. Woos, T. Anderson, and A. Krishnamurthy, "One tunnel is (often) enough," *ACM SIGCOMM Computer Communication Review*, vol. 44, no. 4, pp. 99–110, 2014.

[40] Y. Wang, J. Bi, and K. Zhang, "A sdn-based framework for fine-grained inter-domain routing diversity," *Mobile Networks and Applications*, vol. 22, no. 5, pp. 906–917, 2017.