EasyChair Preprint

№ 15950

SAVA Deployment for Spoofed Source Attacks

Wenjie Yang, Yong Tang and Wenyong Wang

March 28, 2025

SAVA Deployment for Spoofed Source Attacks

Wenjie Yang, Yong Tang, and Wenyong Wang(✉)

University of Electronic Science and Technology of China, Cheng du, China
wenjie4530@gmail.com , {worldgulit,wangwy}@uestc.edu.cn

Abstract. Distributed Denial of Service (DDoS) attacks are one of the main threats facing the Internet today, and a considerable number of them originate from attacks using spoofed source addresses. The Source Address Verification Architecture (SAVA) technology can effectively mitigate such attacks by verifying the legitimacy of the source address. However, the deployment of SAVA faces some practical challenges, including the complexity of real network topologies, high deployment costs, and the impracticality of full deployment. To address these issues, this paper describes the SAVA deployment model in detail and proposes an incremental deployment approximation algorithm. The algorithm can identify a set of approximately optimal SAVA deployment points in any network topology, aiming to maximize the filtering of attack traffic. Experimental results show that compared with conventional deployment methods, the deployment algorithm shows superior performance in handling spoofed source attacks while maintaining a low false negative probability.

Keywords: DDoS · Spoofed source · SAVA · Deployment

1 Introduction

With the continuous expansion of the network scale and the diversification of attack methods, network security has become a crucial issue for the Internet today. Distributed denial-of-service (DDoS) attacks, in particular, represent one of the main threats to the Internet [1]. Spoofed source attacks are a common form of DDoS attack, which mainly hides the true identity of the attacker by forging the source IP address. This attack method makes it difficult for the victim to track the source of the attack, thus increasing the difficulty of defense.

Spoofed source attacks mainly include two forms: Spoofed source flooding and reflection amplification. Spoofed source flooding attack is an attack method that floods the target server by sending a large number of SYN requests with spoofed source IP addresses. In this attack, the attacker replaces the source IP address in the data packet with a spoofed address, causing the target server to try to send a response to these spoofed IP addresses after receiving the SYN request. Since these spoofed IP addresses may not exist, the resources of the target server will be exhausted, making it unable to process requests from legitimate users. The typical characteristics of this attack are high concurrent connection requests and a large number of half-open connection states.Reflection amplification attacks

amplify initial traffic by exploiting vulnerabilities in network services, causing bandwidth consumption, resource exhaustion, and potentially leading to service interruptions, security risks, reputation loss, and financial losses [2,3]. Reflection amplification attacks in spurious sources pose unique challenges: they generate massive traffic capable of crippling a victim's server in a short time, obscure the source by using spoofed IP addresses, and are difficult to defend against as reflection servers are legitimate Internet services, making simple IP blocking an ineffective measure [4]. Given these characteristics,Spoofed source flooding and reflection amplification attacks pose a serious threat to Internet security today, highlighting the urgent need for effective countermeasures. The Source Address Validation Architecture (SAVA) [5] offers a promising solution to mitigate spoofed source attacks by verifying the legitimacy of source addresses. The next paragraph provides a brief overview of SAVA technology.

SAVA is a multi-layer IP source address validation technology designed to prevent malicious attacks based on spoofed source addresses, thereby enhancing Internet security. It enables source address validation at multiple points—within access networks, inside autonomous systems, and between autonomous systems ensuring the authenticity of source addresses. At the subnet level, SAVA dynamically binds router ports to valid source IP addresses, ensuring that network devices can only use legitimate source IP addresses. Within autonomous systems, SAVA establishes filtering tables in routers that associate each incoming interface with a valid set of source address blocks, thereby filtering out packets with spoofed source addresses. Across autonomous systems, SAVA performs route-based validation to confirm the authenticity of source addresses, ensuring that traffic between systems is traceable and legitimate [5,6]. This layered, multi-level protection mechanism verifies the legitimacy of each packet's source address throughout its transmission, effectively preventing DDoS attacks based on spoofed source address.

Despite its effectiveness in preventing attacks based on spoofed source addresses, deploying SAVA in practical faces several challenges. SAVA requires a certain deployment scale to achieve optimal effectiveness. Given the complexity of real-world Internet topologies and the multitude of potential deployment nodes, full deployment is impractical, and conventional deployment approaches often fail to yield satisfactory results [7]. Based on the above analysis, this paper mainly solves the deployment problem of SAVA. Our contributions are summarized as follows:

- We present a rigorous mathematical model for the SAVA deployment problem and prove that the deployment benefit function is submodular.
- We propose the Submodularity-based Dynamic Greedy (SDG) algorithm, and demonstrate that the SDG algorithm achieves a constant-factor approximation to the optimal solution under resource constraints.
- We analyze of the limitations with conventional deployment approaches, including near-source and near-destination deployment tactics. We compare the performance of the SDG deployment algorithm against these conventional approaches.

The remainder of this paper is structured as follows. Section 2 of this paper summarizes the relevant background and related literature. Section 3 introduces the system architecture, gives the specific modeling process, describes the algorithmic flow and gives a detailed proof of the approximation ratio of the algorithm. Section 4 presents the SDG deployment results and experimental comparisons. Finally, we conclude this paper in Section 5.

2 Related Work

Source Address Validation (SAV) mechanisms are crucial for preventing IP address spoofing, a common tactic in various network attacks. The Source Address Validation Architecture (SAVA) was introduced to ensure that every packet's source address is verifiable throughout its transmission path. Wu et al. [5] proposed SAVA as a multilayer framework capable of enforcing source address validation at access networks, within autonomous systems, and across inter-domain traffic.

Building upon SAVA, the Source Address Validation Improvement (SAVI) framework was developed to provide finer-grained, standardized IP source address validation at the level of individual IP addresses. Bi et al. [8]described the design and motivation behind SAVI methods, which prevent nodes on the same IP link from spoofing each other's IP addresses, thereby complementing ingress filtering.

SAVA and SAVI are effective in theory, their deployment in real-world networks presents challenges.An early framework by Bremler-Barr and Levy [9] discussed incremental deployment strategies for router-assisted services, emphasizing the importance of strategic placement to enhance network performance. Korczyński and Nosyk [10] conducted an Internet-wide active measurement study to assess the deployment of SAV mechanisms, revealing that a significant number of networks remain vulnerable to IP spoofing due to the lack of proper SAV implementation.

In terms of deployment,the integration of submodular optimization in deployment strategies has been investigated. Wilder [11] examined equilibrium computation and robust optimization in zero-sum games with submodular structure, providing insights relevant to security resource allocation in adversarial settings.

Despite these advancements, challenges remain in applying submodular optimization techniques to deployment in real-world network scenarios. This paper address these challenges by developing a submodularity-based deployment algorithm tailored for spoofed source attacks.

3 System Model

3.1 SAVA Deployment Architecture

The SAVA deployment architecture for preventing spoofed source attacks is illustrated in Figure 1. Attacker can control zombie hosts in botnets to spoof IP

addresses of other hosts to attack victims. We can deploy SAVA devices in place of conventional routers to prevent traffic with spoofed source addresses from passing through. Deploying SAVA devices can mitigate the impact of spoofed source attacks on victims, thereby enhancing network security and stability. The main deployment challenge lies in selecting optimal deployment points within a real network topology to maximize attack traffic filtration within resource constraints, achieving a strong defensive effect [12].

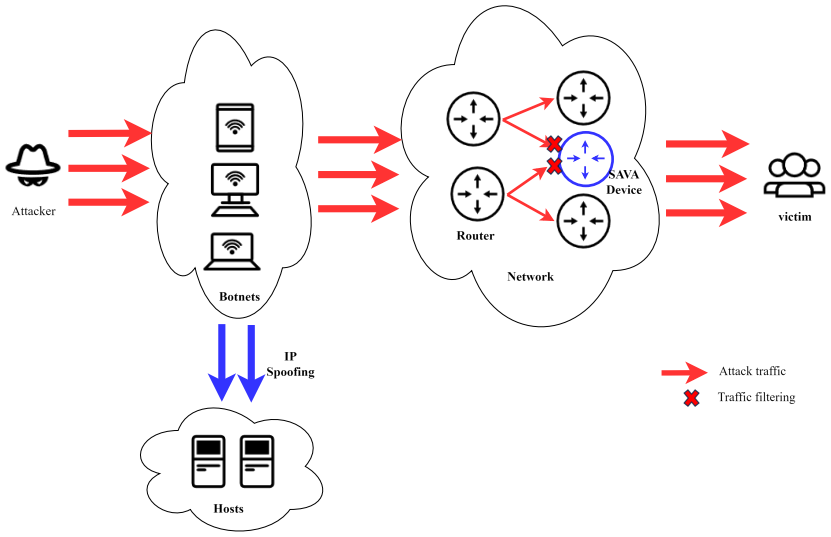

Fig. 1. Deployment Architecture

In real-world network environments, hardware and device costs vary depending on node performance requirements; high-traffic nodes often require higher-performance servers and network devices, leading to different deployment costs across nodes. Geographic location also affects costs, as remote nodes far from central infrastructure may require more resources for connection and maintenance. Additionally, nodes with high bandwidth and traffic demands necessitate more network resources, resulting in higher operational costs. Maintenance and management costs also vary, as complex network environments require more resources for node management, necessitating an effective deployment strategy.

3.2 Problem Modeling

The existing SAVA deployment problem is restated, the corresponding mathematical model is established, a SAVA device balanced deployment strategy based on reliability evaluation is proposed, and relevant indicators are defined and explained.The relevant symbol definitions are shown in Table 1.

Table 1. Symbol Definition

symbol	implication
V	The set of network nodes
E	The set of network edges (adjacencies)
G	The inter-domain topology of the network
S	The set of nodes that have deployed SAVA devices
N	The set of nodes that have not yet deployed SAVA, $N = V - S$
n	Node, $n \in V$
σ_n	Boolean value, 0 or 1, indicating whether node n is a deployment point
T_n	Attack traffic passing through node n
x_n	The cost of deploying at location n
W	Attack traffic dataset

We focus on reducing the amount of attack packets. According to the expected linear property, the expected difference is equal to the expected difference, when the deployed set is S, the number of additional packets that can be filtered by the deployment point n is:

$$reduce(S, n) = \mathrm{E}[benefit_{S\cup\{n\}}] - \mathrm{E}[benefit_S] \tag{1}$$

We use $reduce_S$ to represent the number of attack packets that can be reduced when the deployment set is S.From the definition of deployment benefit above, it can be seen that the benefit of deploying SAVA device is related to the deployed set S,that currently deploy the method and current deployment node n .

After n deploys SAVA, over a period of time, the set of legal packets identified by n is leg_n, the set of illegal packets is $ileg_n$, and the set of packets that are judged as spoofed and discarded by n is dis_n. Then the set of false negative packets is:

$$risk(S, n) = |leg_n \cap ileg_n| \tag{2}$$

An attack packet is defined as follows: $a : (s, d)$, where a is the attacker, i.e., the sender of the packet, s is the source address of the spoofed packet, and d is the target of the attack[13].The filtering method is based on the principle of path filtering. The path-based method identifies whether an packet is a spoofed packet by verifying the forwarding path of a packet. Although the attacker can write a spoofed source address in the packet, it cannot control the forwarding path of the packet because it cannot control the entire routing system. Specifically, for a packet (s, d), when the packet is sent from different locations on the network, its forwarding path should also be different.

In Figure 2, the forwarding paths of $x : (s, d)$ and $s : (s, d)$ are different. From d's point of view, the legitimate packet $s : (s, d)$ should be forwarded from d's upstream node y, not x.

Formally define the filter function after deploying the SAVA devices. A filter function completes the following work: given a topology with routing informa-

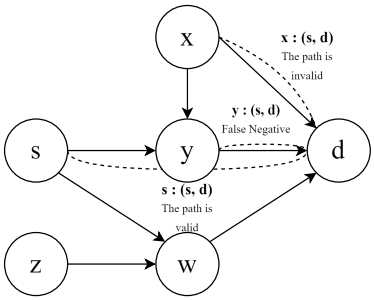

Fig. 2. Filter Method

tion, deployment nodes and an packet $a : (s, d)$ as input, given a deployment node n and assuming that n has deployed SAVA devices, if packet $a : (s, d)$ passes through n when forwarding, n can determine whether the data packet is spoofed. Its form is $filter(S, n, a : (s, d))$, and returns 1 if it is spoofed; otherwise, returns 0.

Given the filter function, the calculation method of the discard function is as follows:

$$discard(S, a : (s, d)) = \text{sgn}(\sum_{n \in S} \sum_{a:(s,d) \in W} filter(S, n, a : (s, d))) \tag{3}$$

$sgn()$[1] is a symbolic function. Its form is as follows.

$$\text{sgn}(x) = \begin{cases} 0, & x = 0 \\ 1, & x > 0 \end{cases} \tag{4}$$

If only consider the number of filtered packets, we need to subtract false negative packets:

$$\begin{aligned} \text{reduce}(S, n) = \sum_{a:(s,d) \in W} \Big(& \text{discard}(S \cup \{n\}, a : (s, d)) \\ & - \text{discard}(S, a : (s, d)) - \text{risk}(S, n) \Big) \end{aligned} \tag{5}$$

We can calculate the ratio of filtered packets to total attack packets. The set of legal packets identified by n is leg_n, the set of illegal packets is $ileg_n$, and the set of packets judged as spoofed and discarded by n is dis_n, we use γ to represent the filtering percentage:

[1] Regarding the definition, when x<0, sgn will be -1, but in formula (3), the value in () will only be >=0. If >0, the value of the discard function is 1, indicating that the message is discarded. Similarly, if =0, the value of the discard function is 0, indicating that the message is retained.

$$\gamma = |dis_n|/|ileg_n| \tag{6}$$

In real-world network deployment, deployable resources are limited. By setting budget constraints, we can ensure that the optimal deployment solution can be achieved under limited resource conditions.The upgrade budget is the total cost of upgrading SAVA devices and does not exceed $M > 0$. We have the following constraints:

$$\sum_{i \in N} x_i * \sigma_i \leq M \tag{7}$$

Our goal is to update the deployment set S to SAVA devices to maximize the deployment benefits, so our objective function is as follows:

$$\max benefit(S_G) \tag{8}$$

S_G is the set of SAVA deployments that satisfy condition (7). The target of filtering percentage can be set to $0 < P < 1$. The percentage of traffic to be filtered may not always meet the filtering requirements if the deployment cost M is too small in actual situations. Our goal is to maximize the deployment benefit, i.e., to calculate (8).

The problem of solving (7) under the condition of (8) is NP, which means that in the worst case, the time complexity of finding the best deployment solution will increase rapidly with the increase of network size.

3.3 Proof of Submodular Function

In order to optimize the deployment of SAVA devices, we use the total benefit function to represent the total traffic size that can be filtered by deploying SAVA devices on a set of nodes. Proving that $benefit$ is a submodular function is important because submodular functions have specific mathematical properties that can be used to develop efficient greedy algorithms, thereby ensuring that the algorithm strikes a good balance between computational complexity and defensive effectiveness. Submodularity ensures effectiveness in algorithm design, especially in optimization problems. Specifically, as the set grows, the additional benefit of adding nodes to the set decreases.

We can prove that the function $benefit$ is a submodular function.

For a collection function $benefit : 2^N \rightarrow S$,Assume that $A \subseteq N$ is a set, $B \subseteq N$ is a set, and $A \subseteq B$ is a set, $\exists i \in N \setminus B$

2^Nrepresents the power set of set N, i.e., the set of all subsets of N

$$\text{benefit}(\{n\} \cup B) - \text{benefit}(B) \leq \text{benefit}(\{n\} \cup A) - \text{benefit}(A) \tag{9}$$

The formula points out that as the set becomes larger, the value of n will become smaller, which is the characteristic of diminishing marginal benefits. When $A \subseteq B$, $f(A) \leq f(B)$, this submodular function is monotonic.

Proof. Select any two sets $A \subseteq B \subseteq N$, and any element $n \in N \setminus B$. N is the set of nodes in the network topology; we need to prove that inequality (9) holds

For a smaller set A, the addition of element n will bring more significant marginal benefits, because the traffic T_n carried by n is not shared or filtered by other nodes in the smaller set. For the larger set B, since it already contains more nodes, there may be redundant nodes. The traffic that these nodes can filter cannot be filtered by the nodes in the set A. Compared with the total traffic, it can The incremental number is small, and the incremental effect brought by the addition of element n is relatively small, because the nodes in set B have already taken on more traffic filtering tasks. Therefore, for any $A \subseteq B \subseteq V$ and $n \in V \setminus B$:

$$benefit(\{n\} \cup A) - benefit(A) \geq benefit(\{n\} \cup B) - benefit(B) \tag{10}$$

$$\text{benefit}(\{n\} \cup A) \geq \text{benefit}(\{n\} \cup B) \tag{11}$$

We have thus proved that the function $benefit$ is a submodular function. □

Therefore, we have proved that under the given algorithm steps and benefit function definition, the function $benefit$ is a submodular function. This means that when selecting nodes, the marginal benefit of each new node is decreasing, which is of great significance for solving optimization problems.

3.4 Algorithm Design

The pseudocode of the Algorithm 1. is as follows:

Proof of approximation ratio can provide theoretical guarantee for the performance of the algorithm, ensuring that the gap between the solution and the optimal solution is within an acceptable range. It shows the effectiveness and reliability of the algorithm in practical applications. The proof process of approximation ratio solved by SDG algorithm is as follows:

Proof. S_i is the deployment set after the ith iteration. The deployment cost of the deployment point selected by the algorithm in the ith iteration is $c(x_i)$, and S^* is the deployment solution with the largest total deployment benefit. The recursive inequality can be derived from the submodularity:

$$\begin{aligned} benefit(S^*) &\leq benefit(S_{i-1}) + benefit(S^* \backslash S_{i-1}) \\ &\leq benefit(S_{i-1}) + \frac{benefit(S_i) - benefit(S_{i-1})}{c(x_i)} \cdot \sum_{x \in S^* \setminus S_{i-1}} c(x) \\ &\leq benefit(S_{i-1}) + \frac{M}{c(x_i)} \cdot (benefit(S_i) - benefit(S_{i-1})) \end{aligned} \tag{12}$$

Algorithm 1 SDG

Step 1:
$S = \emptyset$, $F = 0$
Step 2:
while $N \neq \emptyset$ **do**
  Compute the benefit $benefit(x_i)$ for each node i
  Compute the unit cost-benefit ratio $\frac{benefit(x_i)}{x_i}$ for each node
  Step 3:
  select the node x^* with the highest unit cost-benefit ratio and add it to set S
  Update the state of node x^* to $\sigma = 1$
  if $\gamma < P$ and $\sum_{i \in S} x_i \cdot \sigma_i \leq M$ **then**
    $\mathcal{S} \leftarrow \mathcal{S} \cup \{x^*\}$
    $F = F + benefit(x^*)$
    $\mathcal{N} \leftarrow \mathcal{N} \setminus \{x^*\}$
  else
    Return to Step 2
  end if
end while
Step 4:
Output the deployment set S and total benefit F

The first inequality in Eq. (12) uses Corollary 5 from the middle literature [14], and the second inequality utilizes the greedy nature of the algorithm, i.e., $\frac{benefit(S_i)-benefit(S_{i-1})}{c(x_i)}$ is the highest current profit per unit. The last inequality states that $S^* \backslash S_{i-1}$ the cost of all items never exceeds the total cost space M [15].

Subtracting $\frac{M}{c(x_i)}$ on both sides and reordering the terms:

$$benefit(S_i) \geq benefit(S^*) + \left(1 - \frac{c(x_i)}{M}\right) \cdot (benefit(S_{i-1}) - benefit(S^*)) \tag{13}$$

By recursively applying the inequality and introducing the ratio of the cost of each step to the total cost, we get:

$$benefit(S_i) \geq \left(1 - \prod_{k=1}^{i} \left(1 - \frac{c(x_k)}{M}\right)\right) \cdot benefit(S^*) \tag{14}$$

Use the inequality $1 - x \leq e^{-x}$ to replace the accumulated terms:

$$benefit(S_i) \geq \left(1 - \exp\left(-\sum_{k=1}^{i} \frac{c(x_k)}{M}\right)\right) \cdot benefit(S^*) \tag{15}$$

$C(x^*)$ is the current highest unit profit. The last inequality depends on the cost of all items never exceeding the total deployment cost M . When $c(S) + c(x^*) > M$, replace S_i with $S \cup x^*$, we have:

$$benefit(S \cup x^*) \geq (1 - \exp\left(-\frac{c(S) + c(x^*)}{M}\right)) \cdot benefit(S^*) \tag{16}$$

That is to say, $benefit(S \cup x^*) \geq (1 - \frac{1}{e}) \cdot benefit(S^*)$
When the maximum deployment cost is reached, i.e., $c(S) + c(x^*) > M$
$benefit(S \cup x^*) \geq (1 - e^{-1}) \cdot benefit(S^*)$ holds □

The proof shows the approximation guarantee of the greedy selection algorithm for submodular functions under the cost constraints, where the algorithm tries to maximize the ratio of marginal benefit to cost at each iteration. It is proved that the algorithm can maintain a good approximation to the optimal solution when the cumulative cost is close to the cost constraints.

4 Experiments

4.1 Conventional Deployment Approaches

Conventional deployment approaches include near-source deployment and near-destination deployment. When dealing with spoofed source attacks, near-source and near-destination deployment strategies have their own advantages and disadvantages, and are applicable to different network environments and requirements. Near-source deployment deploys defense devices (SAVA devices) close to the source of the attack, with the goal of detecting and filtering malicious traffic as early as possible to mitigate the impact of the attack on the downstream network. The advantage of this approach lies in the ability to detect and filter early and reduce the amount of malicious traffic entering the network core, thereby reducing network bandwidth consumption and reducing the pressure on subsequent nodes. However, this strategy also suffers from high deployment costs, complex management, and difficulty in locating the source of the attack, especially when the source of the attack is constantly changing or dispersed, the effectiveness may be limited.

Near-destination deployment deploys defense appliances close to the target server or network node, i.e. near victims, aiming to protect the target from attacks by intercepting and filtering the attack traffic before it reaches the target. Advantages of near-destination deployment include centralized defense, cost-effectiveness, and relatively simple management. However, this strategy also faces some challenges such as high network bandwidth pressure, concentrated attack pressure and potential latency issues. Malicious traffic can still consume a large amount of network bandwidth before reaching its destination, which may lead to network congestion and performance degradation, and defense devices need to handle large-scale attack traffic, placing higher demands on device performance and stability.

Combining the above, the selected deployment strategy needs to consider the network structure and attack characteristics. If the attack sources are decentralized and change frequently, near-destination deployment may be more effective;

whereas, if the attack traffic is centralized and targeted, near-source deployment may be more suitable. In addition, cost and management factors need to be considered. Near-source deployment is suitable for network environments with sufficient resources and strong management capabilities, while near-destination deployment is suitable for scenarios with limited resources and the need for centralized management. In practice, a combination of the two deployment strategies may be the best way to achieve the optimal defense effect in order to strike a balance between cost, management and defense effect.

4.2 Experimental Results and Comparative Analysis

In this paper, we simulated the SAVA device filtering attack traffic using Python3. The network topology was generated randomly using Python's built-in random module with a fixed seed to ensure reproducibility. This approach allowed us to generate topologies that conform to the desired statistical properties. The network consists of 287 routers, 85 hosts, and 1568 links. Among the hosts, 40 were designated as zombie hosts, 5 as victim hosts, and the source addresses of the remaining 40 hosts were spoofed by the zombie hosts. The attack traffic from the 40 zombie hosts to the 5 victim hosts ranged from 100 to 1000 Mbps, and the unit deployment cost of each router node was set to 1. In this paper, we compare the near-source deployment, near-destination deployment, and the SDG deployment algorithm proposed above. The experimental parameters were set with reference to similar studies [16], and care was taken to ensure that every host maintained an active link connection.

The experimental results we obtained are as follows:

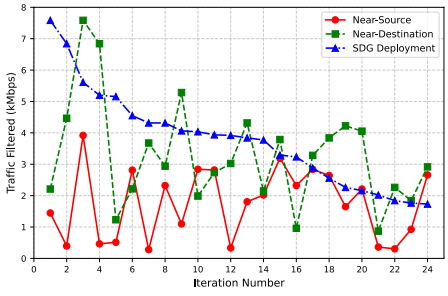

Fig. 3. Iterative Node Traffic Filtering

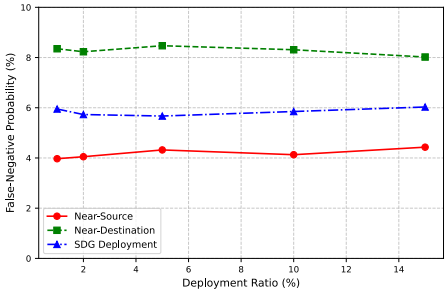

Fig. 4. False negative probability

Figure 3 shows that the nodes selected by the SDG algorithm each time meet the submodular function property, i.e., the marginal benefits of each selected node are decreasing. This means that as nodes are continuously selected, the additional filtered traffic brought by each new node is gradually reduced.

The two deployment approaches of near source and near destination show large fluctuations and instability in filtering traffic.

The false negative probability of different deployment approaches is shown in Figure 4. The near-source deployment shows the lowest false negative probability because attack traffic is intercepted early, reducing the likelihood of same path. In contrast, near-destination deployment shows the highest false negative probability. Because the attack traffic has already spread widely in the network before reaching the filtering node, the possibility that the zombie host and the host whose source address is spoofed by it take the same path increases.

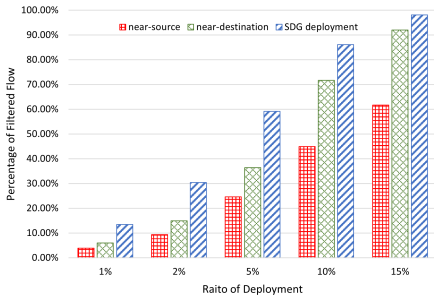

Fig. 5. Traffic Filtering Comparison

Figure 5 shows that the filtering efficiency of the deployment nodes selected by the SDG algorithm is better than that of near source deployment and near destination deployment. Although near source deployment can intercept attack traffic before it enters the network core, reducing network bandwidth occupation and subsequent node pressure, this strategy requires the deployment of defense devices at multiple source locations. But the defense effect is unstable because the attack source may change or disperse continuously. Near destination deployment concentrates defense devices near the target server or key node, but malicious traffic will still occupy a lot of bandwidth before reaching the target, resulting in network congestion and performance degradation. The SDG algorithm selects the optimal deployment node for dynamic adjustment to ensure that each new node can maximize the overall defense benefits. The SDG algorithm can significantly improve filtering efficiency at different deployment ratios[2].

The SDG deployment algorithm, proposed in this paper, strikes a balance between false negative probability and filtering efficiency.This ensures that each selected node maximizes its marginal contribution to overall defense performance. SDG achieves moderate false negative probabilities while significantly outperforming near-source and near-destination approaches in terms of filter-

[2] It is worth mentioning that when the deployment costs at different points differ too much, the SDG algorithm is likely to fail. Considering that we deploy the same SAVA devices, the actual deployment costs are similar, which can achieve good results.

ing effectiveness. This demonstrates the SDG algorithm's ability to adaptively and incrementally optimize deployment strategies, making it more effective in real-world network environments.

5 Conclusion and Future Work

This paper proposes a SAVA incremental deployment algorithm SDG for spoofed source attacks. The algorithm dynamically selects the optimal deployment node through a submodular optimization strategy, while taking into account false negative, to achieve good defense effects under limited resource conditions, and can more flexibly adapt to changes in the network environment. The article gives detailed proof of the design process and approximation ratio of the algorithm. And through experimental simulation and comparison with near-source deployment and near-destination deployment, the experimental results verify the effectiveness of the algorithm in defending against spoofed source attacks. Subsequent research can further test and optimize this algorithm in a more complex network environment, and explore its potential for application in defense against other types of DDoS network attacks.

Our study focused on the deployment of SAVA and did not make an in-depth discussion on how SAVA devices filtering attack traffic, which is related to false positive and is the focus of our future research.

Acknowledgments. This work was funded by the Science and Technology Department of Sichuan Province (Grant No. 2024ZHCG0044) and the Natural Science Foundation of Sichuan Province (Grant No. M112024NSFSC0473).

References

1. Mirkovic, J., Reiher, P.: A taxonomy of DDoS attack and DDoS defense mechanisms. ACM SIGCOMM Computer Communication Review, 34(2), pp. 39-53 (2004). https://doi.org/10.1145/997150.997156
2. Soliman, A. K., Salama, C., Mohamed, H. K.: Detecting DNS reflection amplification DDoS attack originating from the cloud. In 2018 13th International Conference on Computer Engineering and Systems (ICCES), pp. 145-150 (2018). https://ieeexplore.ieee.org/abstract/document/8639414
3. Vasques, A. T., Gondim, J. J.: Amplified reflection ddos attacks over iot mirrors: A saturation analysis. In 2019 Workshop on Communication Networks and Power Systems (WCNPS), pp. 1-6 (2019). https://ieeexplore.ieee.org/abstract/document/8896290
4. Rossow, C.: Amplification Hell: Revisiting Network Protocols for DDoS Abuse. In NDSS, pp. 1-15 (2014). https://dud.inf.tu-dresden.de/~strufe/rn_lit/rossow14amplification.pdf
5. Wu, J., Bi, J., Li, X., Ren, G., Xu, K., Williams, M.: A source address validation architecture (SAVA) testbed and deployment experience. RFC 5210 (2008). https://www.rfc-editor.org/rfc/rfc5210

6. Wu, J., Ren, G., Li, X.: Source address validation: Architecture and protocol design. In 2007 IEEE International Conference on Network Protocols, pp. 276-283 (2007). https://ieeexplore.ieee.org/abstract/document/4375858
7. Du, P., Nakao, A.: DDoS defense deployment with network egress and ingress filtering. In 2010 IEEE international conference on communications, pp. 1-6 (2010). https://ieeexplore.ieee.org/abstract/document/5502654
8. Wu, J., Bi, J., Bagnulo, M., Baker, F., Vogt, C.: Source address validation improvement (SAVI) framework. RFC 7039 (2013). https://www.rfc-editor.org/rfc/rfc7039
9. He, X., Papadopoulos, C., Radoslavov, P.: A framework for incremental deployment strategies for router-assisted services. In IEEE INFOCOM, pp. 1488-1498 (2003). https://ieeexplore.ieee.org/abstract/document/1208984/
10. Korczyński, M., Nosyk, Y.: Source Address Validation. http://dx.doi.org/10.1007/978-3-642-27739-9_1626-1.
11. Wilder, B.: Equilibrium computation and robust optimization in zero sum games with submodular structure. In Proceedings of the AAAI Conference on Artificial Intelligence (2018). https://ojs.aaai.org/index.php/AAAI/article/view/11455
12. Ryba, Fabrice J.,et al.: Amplification and DRDoS attack defense–a survey and new perspectives. (2015). https://arxiv.org/abs/1505.07892
13. Liu, B.-Y., Bi, J.: On the deployability evaluation model of internet inter-domain source address validation (In Chinese). Chinese Journal of Computers 38(3),pp. 500–514 (2015). https://doi.org/10.3724/SP.J.1016.2015.00500
14. Fujishige, Satoru.: Submodular functions and optimization. Elsevier (2005).
15. Chen, Siyang, et al: Collaborative Edge Caching and Dynamic Bitrate Adaptation for SHVC-based VR Video Streaming. In 2024 IEEE 49th Conference on Local Computer Networks (LCN) ,pp. 1-9 (2024). https://ieeexplore.ieee.org/document/10639785
16. Hui, Linbo, et al.: SAV-D: Defending DDoS with Incremental Deployment of SAV. IEEE Internet Computing, 27(3), pp. 44-49 (2023). https://ieeexplore.ieee.org/abstract/document/10122643