Design and analysis of DDoS mitigating network architecture

Swati1 · Sangita Roy2 · Jawar Singh1 · Jimson Mathew3

Published online: 30 November 2022
© The Author(s), under exclusive licence to Springer-Verlag GmbH, DE 2022

Abstract
Distributed Denial of Service (DDoS) attacks have emerged as the top security threat with the rise of e-commerce in recent years. Volumetric attacks are the most common DDoS attacks that aim to overwhelm the victim’s bandwidth. The current mitigation methods use reactive filtering techniques that are not magical and straightforward solutions. In this paper, we propose a network architecture based on the capability to address the threat of DDoS attacks. Physically Unclonable Functions (PUFs) have emerged as a promising solution in security. Motivated by the capability approach, we put forward a network architecture where the routers use Transient Effect Ring Oscillator PUF to generate and verify capabilities. This novel hardware-based solution, to address the problem, has reduced the computational overhead of capability generation. Additionally, the destination has complete control over the incoming traffic in the proposed architecture, resulting in uninterrupted communication with the legitimate clients regardless of the attacker traffic. The large-scale simulation on an open-source Network Simulator (NS-3) has shown that the proposed architecture efficiently mitigates DDoS attacks to a large extent. With our proposed architecture, the throughput was hardly affected when attacker traffic was varied from 10 to 80%.

Keywords Network security · Distributed denial of service · Capability · PUF

1 Introduction
The outbreak of the Covid-19 pandemic in 2019 has accelerated e-commerce growth, with most businesses going digital. The ubiquity and open nature of the Internet has made digital businesses more vulnerable to hackers. The recent incidences and frequent attacks launched by exploiting Internet of Things (IoT) devices’ vulnerabilities, challenging the cyber security of computer networks have established the need for valuable techniques for desktop systems and their feeble brother. The vulnerability and pervasiveness of IoT devices have attracted many hackers, mainly those launching DDoS attacks to slow down a site to gain competitive advantages or distract the security team and launch an exploit in another [1,2]. DDoS flooding attacks are launched through botnets comprising many computers and IoT devices, exploited by their vulnerabilities. Once an attacking army has been set up, an attacker can invoke a coordinated, large-scale attack against one or more targets. Botnet-based DDoS attacks on the application layer limit resources, curtail revenue, and yield customer dissatisfaction, among others [3,4].

The most crucial mechanisms supporting our civilization are those that distribute and regulate energy, like smart grids and industrial ecosystems. In the past, Supervisory Control And Data Acquisition (SCADA) networks, which are used for controlling industrial processes remotely, were isolated from other corporate networks to reduce exposure to insecure networks like the Internet. Gradually, to satisfy the real-time demand, Edge computing (EC) has evolved, improving the quality of service (QoS) in these networks. It offers the benefit of a decreased transmission latency since the compute and storage activities are moved from the distant cloud to the edge clouds. Additionally, EC may handle data from neighboring devices without much data exchange, avoiding network congestion [5]. Despite these benefits, the Internet increased attack opportunities, and all SCADA components now have a higher percentage of zero-day vulnerabilities. EC has significantly intensified security issues since a single-edge server lacks computing and storage capacities making
it highly vulnerable to DDoS attacks [6]. Their services must be continuously offered for society’s social and economic well-being. Any vulnerability or failure might have a significant cascade effect on how well other essential infrastructures function. Researchers have developed a substantial number of DDoS mitigation techniques in recent years. Conventional countermeasures for the issue include firewalls, blackhole routing, intrusion detection systems, cloud-based mitigation and fog computing. However, most are based on a prior understanding of attack characteristics. As the victims are always in a passive state, attackers will ultimately discover and exploit system flaws given enough time and reconnais-
sance efforts. Moreover, the outsourcing approaches as cloud computing are inefficient in detecting and blocking the attack near the attacking sources if the attack is initiated within the local network [7]. The literature argues that to mitigate DDoS attacks effectively, network-wide cooperated work is required and that every party, besides the victim under the DDoS attack, should contribute to the DDoS mitigation. Network-based mechanisms have been known to achieve better DDoS defense efficiency and efficacy. These mechanisms deploy the mitigation functionalities inside networks, mainly routers, and could respond to the attack more accurately.

IoT devices, the weakest link in the security chain due to a lack of computational abilities, are an easy target to penetrate a vast and secure network. Sporadic attention to IoT botnet attacks is a concern that cannot be overlooked. The overall number of cyber attacks in 2020 is more than twice as high as in 2019. According to Neustar Inc., an American internet analytics firm, the highest attack size seen this year is also the largest they have ever neutralized. The attack size was found to be 1.17 Terabits per second (Tbps), among the largest ever seen on the Internet. According to them, the most prolonged single attack they neutralized lasted five days and 18 h [8]. India is also not an exception to it. Covid-19 pandemic spurred a significant transition to online life, resulting in unprecedented levels of threat actors. In India, according to Netscout’s Atlas Security Engineering and Response Team (ASERT), advanced threat analytics and response platform, there were around 5.4 million DDoS attacks in 2021 [9]. These reports emphasize the need for more study in this area of cyber security. At present, the potential DDoS victims are dependent either on Cloud Security Service Providers (CSSP) or vendors selling on-premise hardware equipment designed for scrubbing attack traffic. The research community has proposed various approaches to deal with the detection and mitigation of DDoS attacks. These DDoS attacks through IoT botnets need to be checked by lightweight techniques without compromising security.

DDoS mitigation techniques can be broadly classified as filtering-based approaches and capability-based approaches [10]. The first approach detects malicious traffic and then filters it at the network layer to protect the victim’s resources [11]. On the other hand, only authorized traffic can reach the host in the latter approach [12,13].

A capability-based security model is one of the existing models for secure systems. In this model, an unforgeable token is issued for providing access rights. Researchers and companies have incorporated this approach in developing several secure systems for decades. Tahoe Least-Authority File Store (Tahoe-LAFS), a secure open-source distributed file system, is designed on the “Principle of Least Authority” (POLA) using capability for access control [14]. KeyKOS, a pure capability-based operating system, was designed to meet commercial computer service performance and security goals [15]. Fuchsia, Google’s open-source operating system, prioritizes security and performance and supports embedded systems to smartphones, tablets, and computers. It has introduced security mechanisms for a security-oriented design, including capability routing [16]. Our proposed network architecture is inspired by the capability-based approach to deal with DDoS attacks.

There has been a venerable tradition of authenticating or identifying objects, systems, and people using their intrinsic random physical features. Similar to the fingerprint identification of humans that dates back to the nineteenth century, PUFs emerged as the prominent solution for increased security demands in the system. The multiple-input, multiple-output, large entropy unclonable physical systems are an excellent choice for generating and storing secret keys [17]. We have used these digital fingerprints of the devices in our model to enhance security and reduce computational load.

The motivation of the research is to propose a network architecture that is:

– Scalable
– Works against even sophisticated DDoS attack
– Computationally less expensive
– Should be able to deliver victim-desired traffic (destination-driven traffic control).

1.1 Related work

Many strategies have been proposed and tested for defending the DDoS attack over the Internet. These strategies are broadly classified based on their deployment location and the point in time that defense takes place. The mechanism is generally deployed at the application level and network or transport level, which can be further categorized as source-based, destination-based, network-based, and hybrid. Moreover, on the basis of time of defense, it is further classified as before the attack, during the attack, and after the attack.

Deployment of DDoS defense mechanism at the network/transport level is always a better choice than at the application level, as responding at this level, most of the
resources have already been exhausted. Hybrid defense at the network/transport level is the best mechanism to mitigate DDoS attacks as all the nodes collaborate to deal with the attack. Some of the state-of-the-art defense mechanisms are enumerated in this section.

Yaar et al. [18] proposed a Path Identification for tracking the path traversed by the packet. A fingerprint is embedded in each packet by the routers to enable the victim to track each packet’s path and discard the packets by the attacker despite the source IP address spoofing. The drawback of this mechanism is that the limitation on the identification field’s size may result in the same path information for different paths, leading to increased false-positive rates.

Yaar et al. [12] extended their work to deal with this issue with a capability-based defense mechanism for DoS, Stateless Internet Flow Filter (SIFF), in which every router adds a precapability information based on the IP address and timestamp to the IP packet. Based on the precapability provided by the router, server generates a capability and sends it back to the sender. The legitimate sender sends the capability and precapability to the receiver every time, and the server and the routers in the path validate the info to check if the client is a legitimate sender. However, this approach is vulnerable to the DDoS attack during capability setup, and also, there is no packet flow control for the legitimate users having acquired capabilities.

Later Yang et al. [13] proposed another method based on SIFF, named Traffic Validation Architecture (TVA). They implemented a capability-based defense mechanism in the TCP/IP layer, addressing earlier proposed systems’ limitations. Nevertheless, the capability and precapability in the authenticated packet could be compromised and used to launch an attack. In this model, the receiver is supposed to distinguish the attack traffic from legitimate traffic. The main challenges to be addressed in the capability-based technique are high processing overheads, secure setup of capability channel, and efficiently choosing the unknown sources’ capabilities.

In this study, Liu et al. [19] propose the design and assessment of NetFence, an architecture that sets the network as the first line of defense against DoS attacks. They demonstrated NetFence as an effective and scalable DoS solution using a Linux implementation, ns-2 simulations, and theoretical analysis: it decreases the amount of state retained by a congested router from per-host to at most per-node (Autonomous System).

MiddlePolice, the first easily deployable and proactive DDoS prevention system was claimed and presented by Liu et al. [20] in this study. They carefully designed MiddlePolice such that it does not require any modifications to the Internet core or the client network stack, allowing it to be deployed immediately in the present Internet architecture. MiddlePolice can also impose destination-driven traffic management, ensuring that victim-desired traffic is delivered regardless of attacker techniques. They created a MiddlePolice prototype and demonstrated its viability through rigorous testing on the Internet, a hardware testbed, and a large-scale simulation on NS-3. SIFF, TVA and NetFence use cryptographic algorithms, which leads to the need for secret key management. Table 1 contains the property comparison of other research proposals.

1.2 Contribution and plan of this paper

Our main contribution in the paper are as follows:

- This paper proposes a network architecture for DDoS attack mitigation that harness the physical properties of routers.
- Our approach addresses the limitations of capability-based mechanisms, like high processing overheads and efficient algorithms to secure the capability setup channel. The architecture design is developed considering three main aspects, viz. computation, deployability, and efficiency. Primarily, we attempted to lower the computational cost significantly without compromising with the security. The computations needed to process capabilities where significantly lowered by replacing the cryptographic functions by computationally inexpensive PUFs and XOR operations. Secondly, to deploy the proposed system in the established Internet architecture, we have to upgrade the hosts. We will need to place processing boxes at points of congestion to process the packets in transition. Lastly, the maximum extent up to which the proposed system can check the DDoS attack. There is always a trade-off between performance based on cost and efficiency. We have given more emphasis on making the system more efficient performance-wise rather than considering cost-benefits.
- We propose an architectural level solution based on hardware, which can handle the capability-based defense mechanism in a faster and more secure way for limiting the DoS and DDoS attacks. Since this method alters the IP packet structure, we propose a solution for its implementation for the existing IoT devices.
- We formally prove the security of the proposed system.
- We have simulated the proposed system on an open-source simulator NS-3 to analyze its performance for volumetric attacks.

A detailed discussion on our work is done in the remainder of this paper. Section 2 discusses fundamental ideas in DDoS attacks and PUF to establish the groundwork. In Sect. 3, system and adversary models considered in formulating the problem are discussed, and Sect. 4 presents the design of the proposed architecture. The security analysis of the pro-

Springer
Table 1 Property comparison

| Proposals        | Upgrade requirements | Router support       | Other requirement |
|------------------|----------------------|----------------------|-------------------|
| TVA[13]          | Source and destination | Cryptography, O(N) states | Header update     |
| Netfence [19]    | Source and destination | Cryptography O(N) states | Header update, Passport |
| Middle Police[20] | Only destination     | None                 | mbox              |
| Our approach     | Source and destination | PUF                  | Header update     |

2 Preliminaries

This section will go over some basic terms that will be used over the paper and PUF security, which will be used to build our suggested system in the next section.

Definition 1 Trust boundaries: It is defined as the farthest group of Internet Service Providers (ISPs) with which a web server has an economic relationship. A trust barrier exists everywhere we allow user input in any form. Here, we consider each Autonomous System (AS) as a trust and fate-sharing unit as a trade-off for scalability. We believe that network-managed routers are significantly less likely to be hacked than end systems. As a result, rather than entirely relying on the end systems, we prefer to include routers for the generation of precapability.

Definition 2 PUF: These are the physical realization of a random function that is difficult to clone and shows incalculable challenge-response behavior that is ideally difficult to model but can be readily and reliably assessed in other ways. The basic properties of PUFs’s challenge-response defining its behavior used to provide cryptographic security to our proposed method [21,22].

1. Evaluatable: For given \( \tau \) and \( c \), it is easy to evaluate \( r = \tau(c) \), where \( \tau : C \rightarrow R : \tau(c) = r, c \in C, r \in R \).
2. Unique: \( \tau(c) \) contains some information about the identity of the physical unit embedding \( \tau \).
3. Reproducible: \( y = \tau(c) \) is reproducible up to a small error.
4. Unclonable: For a given \( \tau \), it is hard to construct a procedure \( \gamma \), such that \( \forall c \in C, \gamma(c) \approx \tau(c) \).
5. Unpredictable: It is hard to calculate \( r_i = \tau(c_{ni}) \), with only a set \( Q = (c_i, r_i) \) such that \( r_i = \tau(c_i) \).
6. One way function: It is hard to evaluate \( c \), for a given \( r \) and \( \tau \) without any physical access to the PUF.
7. Tamper-proof: Changing the physical entity that embeds \( \tau \) changes it into \( \tau' \).

However, because the PUF response is unexpected and unclonable, it is assumed that the PUF response cannot be predicted or calculated without physical access to the PUF. It is obvious that the hash function can be replaced with a perfect PUF.

Definition 3 Path id: Path identifiers are 32-bit tags issued by edge routers at trust boundaries that serve as an approximate source locator. These are packet markings included in each packet, as route fingerprint, allowing a victim to identify packets transiting the same Internet pathways on a per-packet basis, independent of source IP address spoofing. IP tracing is essential for detecting attack sources and putting in place Internet security measures.

3 System and adversary models

Our approach exploits the capacity to ensure that packets are sent smoothly between valid users and destinations, even during packet flooding. We consider all the routers and hosts under the trust boundaries run our protocol to achieve this. The routers are accompanied by packet processing boxes containing PUFs. These boxes generate precapability for the incoming request packets and verify regular and renewal packets. In this section, we will describe the system and adversary models.

3.1 System model

First, we will have to understand the system requirements to formulate the problem. Currently, most of the victims rely on CSSPs to mitigate DDoS. They provide reactive filters that do not assure complete success as the mitigation takes place after the detection of the attack. Only reactive filters are insufficient to prevent such attacks, but we must ensure that the network discards the unwanted packets before reaching the bottleneck link. Such an approach is effective in dealing with most adversaries’ strategies. The main properties of our proposal are enumerated below.

Unforgeable capabilities: The main issue of proactive detection of attacks can be solved using capabilities. These packets with capabilities help the router identify and discard unwanted packets before reaching a congested link. Never-
theless, these capabilities need to satisfy some requirements. Firstly, they must be unforgeable and non-transferable. So that stealing or sharing of capabilities can be checked. Secondly, the routers should verify these capabilities independently without trusting other sources. Finally, the overhead of capability generation and verification should be minimum. The source of a packet can be identified using its IP address or mac address. However, IP addresses can be easily spoofed, and we can identify the mac address only when connected to the same network. So, we have used a path id which is discussed in Sect. 4. To ensure non-transferability, we have included the source and destination IP addresses together with path ids in the precapability generation. We have also used a 16-bit timestamp to bound it with a specific time. It only allows the communication for a specific duration after that capability expires and packets are demoted to a lower priority level. We use PUFs and inexpensive XOR operations to minimize the computational overhead of precapability generation and verification. As the router knows all the inputs needed to verify the capability and we are not using cryptographic functions, there is no need for any secret to be shared or changed. PUFs provide secure and low-cost authentication. The destination authorizes the request packet with precapabilities by sending an ordered list of capabilities.

**Tackling the unwanted traffic:** In our proposed model, even the unwanted packet gets authorized at least once. In a condition when any malicious sender gets authorized access, they may flood the network for the authorized period. Such situations are tackled by rate-limiting the capabilities. When the destination issues capabilities, they authorize the traffic to transfer N bytes for the next T seconds. Any source trying to flood the network will be able to transfer the utmost allotted bytes in its authorized period and, such sources will be demoted and blacklisted by the destination.

**IP traceback:** The challenge of identifying the attack source is also addressed in this study. While identifying the device from which the attack was launched remains a difficult task, we have narrowed the problem of identifying the source of the offending packets whose addresses may be faked. Several approaches based on probabilistic packet marking (PPM) [23–25], deterministic packet marking (DPM) [29, 26–28], deterministic edge router marking (DERM) [29] and distributed star coloring (DSC) [30, 31] have been proposed. We have used a PPM approach to deal with source identification. Each router at the ingress of a trust boundary, e.g., AS edge, tags the request with a 32 bit value derived from its incoming interface that is likely to be unique across the trust boundary. Routers that are not at trust boundaries do not tag requests that have already been marked by the upstream. The tags serve as a network path’s identification.

**Victim-driven traffic control:** The destination is provided with the control of network utilization to mitigate DDoS attacks. A client has to agree upon the policies defined by the destination to communicate further. The fine-grained capabilities with rate-limited and timeout capabilities ensure uninterrupted communication between legitimate users and the destination. Capability-based systems, like CRAFT [32] that enforces per-flow fairness, NetFence enforcing per-sender fairness and SIBRA [33] enforcing per-steady-bandwidth fairness, works with a scheduling policy. We have implemented an idea as in TVA [13], where fair queuing is based on the victim’s authorization. Regular, renewal, request and demoted packets have different priority levels. Regular packets enjoy the highest priority, followed by renewal and request.

### 3.2 Adversary models and assumptions

**Adversary Model:** An attacker aims to exhaust the end-system and network resources with excessive traffic. Our proposed model focuses on mitigating such network layer flooding attacks along with application vulnerabilities. We assume that the attacker owns large botnets and can launch strategic attacks. Regarding the capacity of an attacker aiming to attack, some essential assumptions were considered.

- **Assumption 1** The attacker is outside the trust boundaries and cannot corrupt the routers. It is the basis of secure protocol to make sense.
- **Assumption 2** The routers are well connected to the packet processing boxes. These processing boxes are not compromised. Even the secure protocols running on infected devices cannot assure protection. Such integrity of the hardware device is required.
- **Assumption 3** The PUF generates the same response for the same input as a challenge. All the computations of the PUF are done in a trusted zone of the device.

### 4 Proposed network architecture

#### 4.1 Design overview

This section deals with the key components of the proposed design. The scheme aims at dealing with the flood of packets by generating capabilities, which are short-term authorization provided by the receiver/sink to the legitimate senders. The processing overheads required for generating capability/precapability are reduced without compromising the security and efficiency of the algorithm by including PUFs in the architecture design. PUFs have emerged as the most promising solution to ensure low-cost, secure authentication, access control, and traceability [34]. These PUFs will be synthesized and incorporated into the packet processing boxes of the legacy routers. We recommend TERO PUF,
which has optimized power consumption, area efficiency, and reliability, in keeping with the design consideration [35].

According to the definition of PUF, it is said to be secure if the response for two different challenges has at least $d_1$ variation, and any challenge for two different PUFs should produce distinct responses with at least $d_2$ variation where $d_1$ and $d_2$ are the error tolerance threshold for PUF. Mathematically,

$$\Pr[\text{HD}(\text{PUF}_1(C_1), \text{PUF}_1(C_2))] > d_1 = 1 - \varepsilon,$$
$$\Pr[\text{HD}(\text{PUF}_1(C_1), \text{PUF}_2(C_1))] > d_2 = 1 - \varepsilon,$$

where $\varepsilon$ is negligible small value, $\text{HD}$ is the hamming distance, and random challenges $C_1, C_2 \epsilon \{0, 1\}^k$.

### 4.1.1 Packets with capability

Considering that all the routers and hosts in the network run our protocol, the main aim is to drop the unwanted packets before reaching a congested link to prevent a DDoS attack. An early stage detection is only possible when each packet carries information that can be verified to confirm its legitimacy. The packets must be identified and dropped in an early stage. Each packet must carry information that the routers can verify to confirm its legitimacy. Such a token is known as capability. The destination issues these unforgeable capabilities, and their validity is only for a specified period. Figure 1 shows the simple flow of packet for obtaining capability.

The IP packets are categorized as request, renewal, demoted, and regular packets in the capability-based system. The packets from unknown sources without any capability/precapability are considered the request packet. Each router adds the precapability to such packets, marking its path to the server, and finally, the server computes the capability based on the precapability information. The packets already having the capability issued by the server are known as regular packets. These packets have higher priority than others as the server has already authorized them for further communication for a particular period. The router checks the capability to ensure legitimate traffic. After a specified period for regular packets expiry, the client can renew the capability using its list of capabilities. Such packets are renewal packets, and their list of capabilities is verified to generate a new set. If it fails the capability check, the priority level of such packets is lowered, and they fall in the category of demoted packets. The structure of regular and request capability packets is shown in Fig. 2. Here, the common header carries the information about the type of packet, i.e., regular, request, renewal, or demoted. The request packet carries a request header along with the common header consisting of blank capabilities and path ids.

### 4.1.2 Queue management

Request sent to the destination as a part of TCP SYN packet to obtain capabilities. The regular packets are further checked by the routers and validated for further communication. The initial request channel must, however, avoid providing a path for DoS attacks, either by flooding a destination or denying requests from authorized senders. In the proposed architecture, flooding of destination was handled by rate-limiting the requests to 5% of the total bandwidth of each link. Moreover, the possible solutions to prevent legitimate requests from the overwhelming attacker traffic with their drawbacks are tabulated in Table 2. In order to localize the impact of an attack, we have used the path identifiers to locate the upstream party, and a fair-queue analogous to TVA [13].
4.1.3 IP traceback

The intermediate border routers at the AS edge generate path ids as shown in Fig. 3. These unique tokens are generated by XORing the source IP address with the router IP address. Thus transversing the multiple hops, every path id gets XORed to the IP address of the edge router, and the subsequent path id is generated. The path id generation is given in Algorithm 1. Since XOR is its own inverse, IP tracing the packet using the path ids generated is affordable and straightforward.

Algorithm 1: An algorithm for generating path id

```
source_id ← s ⊕ edge_hopid  \(\triangleright s: \text{source IP}\)
next_pathid ← next_pathid ⊕ edge_hopid  \(\triangleright v: \text{victim IP}\)
```

The 32-bit marking information fits into the IP header as seen in Fig. 4. The IP identification field, flag bits, and fragment offset field are all overloaded by the path id. According to research, the internet fragments fewer than 0.25 percent of packets [36]. Just like all other PPM methods, we have employed the IP identification field, which is used for IP fragmentation. We have also used the reserved flag bit for packet labeling, as proposed in [37]. Because the fragment offset field loses its relevance when the IP identification field is used for something other than IP fragmentation. However, repeating the fragment offset field is not recommended because the IP datagram with a nonzero value in the fragment offset field is treated as an IP fragment by the destination. The destination host may mix up designated packets with IP fragments. To avoid this kind of perplexity, we propose to use the unused Type of Service (TOS) bits. The 8-bit ToS employs three bits for IP Precedence and four bits for ToS, with the final bit unused. Although specified, the 4-bit ToS field has never been utilized. The last two bits of TOS are used as the Don’t Fragment (DF) flag and More Fragment (MF) flag. While marking a packet, the router sets the DF to be 1 and MF to 0. Thus, the marked packets from legitimate users can be easily identified and will not be fragmented downstream from the marking router. The number of false positives in the upstream router map is reduced using this method.

4.2 Design description

We have mainly focused on three points in the proposed protocol: firstly, a secure and lightweight algorithm for the generation of precapability for the request packets by the routers; secondly, the routers’ regular packet handling and finally packet handling by the server and the client. The tokens are verified at each level, and the details of the blocklisted sources are available with the destination.

4.2.1 Precapability generation by routers

The packets are classified as regular or request based on the availability of capability. Each packet carries a capability header which is a shim layer above IP. The regular packets have capability headers piggybacking the capability information. On the other hand, the request packet carries a blank capability to initiate communication with the server. The router identifies such packets and generates the precapability using the defined algorithm. The algorithm for generating precapability is shown in Fig. 5.

We have used both the logical address and path id to generate the precapability for the request packets. The capability/precapability should be secure and bandwidth-efficient, but there is always a trade-off between these two. Small precapability makes it vulnerable to brute force attack, and large precapability will make it bandwidth inefficient. A 64-bit capability ensures security without consuming a large bandwidth.

The IP address of the sender and path id generated by the router generates the first 32-bits of precapability. They are XORed to generate a challenge for the PUF, whose the response is again XORed with the 32-bit timestamp generating the first 32 bits of precapability.
The next 32-bits of precapability are generated similarly using the IP address of the destination, path id and PUF response of timestamp. The two outputs are finally concatenated to form a 64-bit precapability.

4.2.2 Packet handling by server

The server receives both the request packets as well as the regular packets. For request packets, it has to decide whether to authorize or reject the request. Initially, every request packet is issued capability and timeout. Every capability is valid for a period of the next T seconds and has the authority to send up to N bytes. However, if a client misbehaves by flooding unexpected packets, it is blacklisted and loses its capability. For any received packet, the server first checks its capability, and if it is a request packet with an ordered list of precapability, it generates capability by hashing precapability along with N and T, as shown in Fig. 6.

4.2.3 Regular packet handling by routers

The request packets accepted by the server are issued capabilities. The routers also check such regular packets for balancing the flow of authorized traffics. Every capability generated is only valid for a period of the next T seconds and has the authority to send up to N bytes. If the regular packets fail the check by the router or the capability expires, the packets are demoted to the low priority traffic and have to get their capability reissued by the destination. Figure 7 shows the flow diagram of the regular packet handling algorithm by the routers.
4.3.2 Storage constraints

The proposed model does not impose any constraints on the routers for storing sensitive data, as they do not have to store any secret key. Moreover, the header packet only stores the path id and capabilities or precapabilities used as token for communication. On the other hand, TVA uses keyed hash function, which needs a secret key to be shared and stored.

5 Security analysis

Forgery or Tempering: The malicious end system may attempt to forge capability or manipulate the authorized access time (T) or bandwidth (N). However, capabilities cannot be spoofed if the PUFs are secure. The use of N and T in the generation of capability prevents any scope of unnoticed tampering with them. The design of PUFs is such that its output depends on the physical microstructure of the system, and it is unpredictable [38]. The PUFs are one-way functions that cannot be replicated. Finally, these functions are hard to predict but easy to construct, making them a perfect choice for lightweight devices [39].

Stealing or sharing of capability In our proposed model, the capabilities are bound with the source and destination IP addresses and the routers to reach the destination, so another sender cannot use the capability. In the case of eavesdrop, the attacker may spoof the IP address and speak for any senders. However, such a situation is also handled in our proposed model. The capabilities are rate-limited and time-bounded. Capabilities need to be renewed after a specific period, and if within authorized time sender misbehaves, they are blocklisted and demoted by the destination. Consequently, compromised senders will not be further able to send the packets. This will only hamper the compromised sender’s traffic, but other legitimate users’ communication will not get affected.

Masquerading attack An eavesdrop can launch such an attack by masquerading a receiver, say a compromised router, to send attack traffic. Here, only the queue at the particular router will be affected. As the precapability of downstream routers cannot be forged, their performance will not get affected.

Flooding with request Packets The malicious system may attempt to flood the routers with request packets to launch the attack. However, in our proposed system, the priority of different packets is set. The regular packets with capability have the highest priority and will be given preference. Then comes the renewal/demoted packets, and the least priority is given to the request packet. The request packets are rate-limited. It can use a maximum of 5% of the bandwidth. So flooding these packets will not affect the legacy packets. However, the new legitimate request packets may face some delay.

6 Simulation

This section will discuss the design parameters, simulation methodology, and its limitations. We have simulated the proposed design on network simulator NS−3.33 to check the efficiency in mitigating DDoS attacks. In real-world DDoS attacks, millions of bots flood the targeted victim. However, we cannot simulate at such a level with our limited resources. So to simulate the attack scenario, we have fixed the number of nodes and scaled-down the link capacity. We have used the dumbbell topology, which consists of 10 ASes each contain-
ing 100 clients, 1 transit AS containing the bottleneck routers, and another AS containing the victim/sink, as shown in Fig. 8. The 10 ASes are connected to the destination AS through the transit AS, which consists of the bottleneck routers. Each of the 10 ASes consists of some malicious clients and a few legitimate clients. The client data rate is 1 Mbps, and the attacker Data Rate is set to 30 Mbps. Bottleneck bandwidth is set to be 1 Gbps, and non-bottleneck bandwidth is 10 Gbps. Link delay is 5–10 ms. Request packets are rate limited to 5% of the link capacity. To flood uplink request capacity by sending requests, the attacker needs to send at the rate of 50 Mbps. We checked the performance by varying the number of attackers. The results were obtained for 10%, 20%, 40% and 80% of attackers.

### 6.1 Result and discussion

The topology and parameters used for the simulation are discussed in Sect. 6. Now, as per the set conditions, the attack was launched, and the number of attackers and the rate of packets were varied to analyze the performance of the proposed network architecture. The processing time of a request packet without attack was found to be 156 ns and, under attack, 984 ns. The processing overhead of the regular packets with capability was found to be 15 ns. Figure 9 shows the throughput of different types of packets. The variation of outgoing traffic as the incoming traffic is varied is clearly plotted in the figure. The throughput of request packets was found to be the lowest under attack as it reached its bandwidth limit. However, the legitimate users having regular packets did not suffer any loss due to increased traffic. The renewal packets did notice the slight loss of packets.

With the increased attack, the legitimate request packets also get lost. The user has to keep on trying a number of times to get success. The packet loss rate can be obtained by equation 2, where \( B_a \) is the attack bandwidth and \( B_l \) is the bottleneck bandwidth.

\[
p = \frac{B_a - B_l}{B_a}
\]  

We suppose that unprivileged traffic is clogging up the network’s last \( i \)th hops and that there is a \( \epsilon_i \) chance that it will be dropped at any one of those routers. Because inbound packets only often face congestion in routers during flooding attacks, we neglect the possibility that the server’s outgoing packets will be dropped. \( P(\text{connect after 1 try}) = (1 - \epsilon_i)^i \) is the probability that a client and server would be able to establish a privileged channel after one try (by exchanging two packets), as drop probabilities at routers are independent Bernoulli trials.

Let \( N \) be the connection attempts required by the client for successful connection. Then, the probability of being connected after \( N \) tries is as follows:

\[
P = 1 - (1 - P(\text{connect after 1 try}))^N
\]

\[
= 1 - (1 - (1 - \epsilon_i)^i)^N
\]

\[
\Rightarrow N = \frac{\log(1 - P(\text{connect}))}{\log(1 - (1 - \epsilon_i)^i)}
\]  

The logarithmic dependence of \( N \) on the connection probability implies that even for more significant \( \epsilon_i \), a new client can get the capability issued after a reasonable waiting time.

Once the packet gets capability, it does not share the same traffic with the attacker. As a result, the file transfer rate is not affected by increased attack traffic. The same phenomenon is shown in Fig. 10. There is no delay time for queuing with 10% and 20% attack traffic, and file transfer is smooth and faster. Nevertheless, when the attack traffic reaches 80%, there is
delay and some loss in getting request packet re-transmission, which is evident from the obtained results.

Figure 11 shows the comparison of throughput ratio with and without the proposed architecture when the attack traffic is varied from 10% to 80%. The performance of the network improved significantly with the proposed architecture. The throughput dropped to almost zero with an attack traffic of 60% in the network without the proposed architecture.

7 Modification for IoT

Since the above-proposed system changes the existing IP packet structure, IoT clients cannot handle the packet. In such cases, the IoT users need to process the packets at their end. We propose introducing a middleware between the IoT clients and the router to handle the capability packet’s overhead.

The idea flow graph is as shown in Fig. 12. The IoT clients send the IP packets encapsulated in an Ethernet frame to the edge device. These IP packets will get processed here and converted into capability packets without modifying their payload. Then, these packets are classified as request or regular packet as per the validity, and it is encapsulated in ethernet frame and forwarded to the internet after Network Address Translation (NAT) handling.

The edge device will first process the incoming reply packets from the server. It will store the capability if the timestamp is not expired in a hash table and convert the capability packet to an IP packet. These packets are encapsulated in an ethernet frame and sent to the IoT clients.

8 Conclusion

The proposed capability-based approach for mitigating DDoS attacks addresses three challenges. Firstly, eliminate the need to share or store any secret key for capability generation, which adds to the vulnerability. Secondly, by using PUF, we can make the whole precapability generation process faster and more reliable. Finally, it is destination-driven, as the destination sets the rate of incoming data from legitimate users having capabilities. It is evident from the results discussed in Sect. 6.1 that our motive for limiting the DDoS attack is fulfilled. Most of the data transmitted from legitimate users were received with minimum time lags. The deployability of the approach at a commercial scale remains a matter of concern. In addition to this, the network configuration’s static and homogeneity help the attackers find the vulnerabilities and exploit them. Hence, to deal with such a scenario, it would be interesting if we could reverse the situation dynamically by changing the configuration of the cyber-system.

Availability of data and materials Data sharing is not applicable to this article as no datasets were generated or analyzed during the current study.

Declarations

Conflict of interest The authors declare that they have no conflict of interest or competing interests.

Ethical approval This article does not contain any studies with human participants or animals performed by any of the authors.

Informed consent No informed consent is to be reported.
38. Gope, P., et al.: Lightweight and physically secure anonymous mutual authentication protocol for real-time data access in industrial wireless sensor networks. IEEE Trans. Ind. Inform. 15(9), 4957–4968 (2019)

39. Gope, P., Lee, J., Quek, T.Q.S.: Lightweight and practical anonymous authentication protocol for RFID systems using physically unclonable functions. IEEE Trans. Inf. For. Secur. 13(11), 2831–2843 (2018)

Publisher’s Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.