IMPLEMENTATION OF DATA PATH CREDENTIALS FOR HIGH-PERFORMANCE CAPABILITIES-BASED NETWORKS

A Thesis Presented

by

KAMLESH T VASUDEVAN

Submitted to the Graduate School of the
University of Massachusetts Amherst in partial fulfillment
of the requirements for the degree of

MASTER OF SCIENCE IN ELECTRICAL AND COMPUTER ENGINEERING

September 2009

Electrical and Computer Engineering

IMPLEMENTATION OF DATA PATH CREDENTIALS FOR HIGH-PERFORMANCE CAPABILITIES-BASED NETWORKS

A Thesis Presented

by

KAMLESH T VASUDEVAN

Approved as to style and content by:

Tilman Wolf, Chair

Weibo Gong, Member

Lixin Gao, Member

C.V.Hollot, Department Chair
Electrical and Computer Engineering

ACKNOWLEDGEMENTS

I would like to thank my advisor Prof. Tilman Wolf for his guidance, support and encouragement throughout my thesis. I would also like to thank all my friends and family whose support helped me stay focussed on this project.

This material is in part based upon work under subcontract #069153 issued by BAE Systems National Security Solutions, Inc. and supported by the Defense Advanced Research Projects Agency (DARPA) and the Space and Naval Warfare System Center (SPAWARSYSCEN), San Diego under Contract No. N66001-08-C-2013.

ABSTRACT

IMPLEMENTATION OF DATA PATH CREDENTIALS FOR
HIGH-PERFORMANCE CAPABILITIES-BASED NETWORKS

SEPTEMBER 2009

KAMLESH T VASUDEVAN, B.E., ANNA UNIVERSITY, INDIA

M.S.E.C.E., UNIVERSITY OF MASSACHUSETTS AMHERST

Directed by: Professor Tilman Wolf

Capabilities-based networks present a fundamental shift in the security design of network architectures. Instead of permitting the transmission of packets from any source to any destination, routers deny forwarding by default. For a successful transmission, packets need to positively identify themselves and their permissions to the router. A major challenge for a high performance implementation of such a network is an efficient design of the credentials that are carried in the packet and the verification procedure on the router. A network protocol that implements data path credentials based on Bloom filters is presented in this thesis. Our prototype implementation shows that there is some connection setup cost associated with this type of secure communication. However, once a connection is established, the throughput performance of a capabilities-based connection is similar to that of conventional TCP.

TABLE OF CONTENTS

Page

ACKNOWLEDGEMENTS . . . . . **iii**

ABSTRACT . . . . . **iv**

LIST OF TABLES . . . . . **vii**

LIST OF FIGURES . . . . . **viii**

CHAPTER

1. INTRODUCTION **1**

2. RELATED WORK **6**

3. DATA PATH CREDENTIALS **9**

3.1 Security Model 9

3.1.1 Security Requirements 9
3.1.2 Capabilities of an Attacker 10

3.2 Network Architecture 11
3.3 Router Architecture 13
3.4 Connection Management Scenarios 14

3.4.1 Unicast 16
3.4.2 Multicast and Network Coding 17

3.5 Authentication, Authorization and Access Control 19

4. CREDENTIALS DESIGN **21**

4.1 Requirements 21
4.2 Bloom Filter Based Credentials 22

4.2.1 Bloom Filters 22

4.2.2 Credentials Aggregation 24

4.3 Credentials Security 25
4.4 Density Limit 27
4.5 Group Credentials 28

5. PROTOCOL IMPLEMENTATION 29

5.1 Credentials Header 29
5.2 Router Processing 31
5.3 Testing 32
5.4 Bidirectional Verification 33
5.5 Implementation in Emulab 34

6. SECURITY ANALYSIS 36

6.1 Probability of Successful Attack 36

6.1.1 Unicast 37
6.1.2 Multicast/Multipath 38
6.1.3 Network Coding 39

7. PERFORMANCE RESULTS 41

7.1 Connection Setup 41
7.2 Delay Measurement and Throughput Calculation 43

8. CONCLUSION 47

BIBLIOGRAPHY 48

LIST OF TABLES

Table **Page**

5.1 Different Scenarios and Verdict on the Packets 33

LIST OF FIGURES

Figure **Page**

3.1 Connection Setup to Establish Credentials 12

3.2 Router Architecture 13

3.3 Connection Setup to Establish Credentials 16

3.4 Connection Setup using Group Credentials 17

3.5 Connection Setup to Establish Credentials 18

4.1 Empty Bloom Filter 23

4.2 Credentials Data Structure 24

5.1 Data Path Credentials Protocol Header 30

5.2 Decision Diagram for DPCP Processing on Router 32

7.1 Connection Setup time for Single Hop 42

7.2 Connection Setup Time for Different Number of Hops 42

7.3 Processing Delay for Various Packet Size 44

7.4 Throughput ratio between DPCP and TCP with iptables 46

CHAPTER 1

INTRODUCTION

The exponential growth of the current Internet has been extraordinary. The ubiquitous nature of the Internet makes it more successful in attaining worldwide connectivity between millions of computing devices, variety of networks and users. One of the main reason for such a success is due its nature of an open architecture and allowing each host to communicate with any other host or end system on the network. A major fault in such an open architecture is its failure to provide inherent security. Security services like authentication, confidentiality, integrity and availability are provided through number of add-ons or patches. These patches range from conventional cryptographic operations such as SSL, TLS and VPN tunnels to traffic monitoring tools like firewalls and Intrusion Detection Systems (IDS) to defense against denial of service (DoS) attacks through anomaly detection and rate limiting. These patches or add-ons are not considered efficient when we think about a network with inherent security features, since they are just defense against specific attacks and does not give a concrete security solution at an architectural level.

Certain communication areas that involve military communications, financial transactions, remote medical procedures, etc. requires superior level of inherent security that in-turn puts us in a position where we require total control over the entire network and the protocol stack. Using dedicated networks for such communication scenarios separates them from other existing networks that helps them to avoid denial of service attacks and intrusion. The idea of network virtualization [5] and the possibility of its deployment in the next-generation Internet testbed provides us an

opportunity to consider the clean-slate design of high-performance networks that can provide superior levels of security that operates on the same physical infrastructure. Certain important questions can be raised on how to design such a network architecture that is physically or logically isolated from the current network with inherent security features and how to protect them against insider attacks.

The design philosophy of current Internet works by the principle of "on-by-default." But recent proposals for capabilities-based networks suggested the concept of "off-by-default" in which each and every connection needs to specifically authorized to reach the end-host that is exactly opposite to the way current Internet works in which the host reaches the end-system by default. The system is authorized based on the capabilities that it provides that are similar to tokens for that particular operation. The capabilities are validated during the initial connection set up. They are also validated along the entire connection path during the data transfer. The existing designs of capabilities-based networks differ in the aspects of how they distribute, implement and verify their capabilities.

The following section explains how capabilities-based networks work with some of its key features and weakness:

1. Only Legitimate Traffic

   The capability-based communications occurs in two stages: capability setup and data transmission. Both of these stages involve the sender, the receiver and a set of verification points or nodes (e.g., routers) located on the path between the two. The capability setup stage involves the following:

   - The sender transmits a capability request to the receiver.
   - All the verification nodes (routers) along the path, marks or stamps the forwarded request with a specific mark; all of these marks finally constitute the capabilities.

- The receiver returns the capability to the sender.

In the data transmission stage, the sender includes the capabilities in all of its packets that it sends to the receiver. Each router along the path validates the packet for those special "capabilities" that has been previously issued. If the packet carries the valid capabilities, the packet will be forwarded, else it will be dropped. Capabilities are nothing but essentially tickets with an expiration date. A well-known legitimate client can get a life-long ticket, whereas a new unknown client may be given a ticket that expires in one minute. The receiver decides all these issues based on some of its own custom policies. Suppose, if the client sends unwanted malicious traffic, it will be classified as an attack source and no more new tickets will be issued. A malicious node cannot send authorized traffic, but it might attempt to flood a receiver with several capability requests. Such an attack can be prevented from interfering with authorized traffic by partitioning the receiver's downstream bandwidth. Most of the share is given to the established connections and a small share for capability requests.

The main objective of the thesis is to implement and prototype such a capabilities-based network architecture that uses a new design of capabilities called as "data path credentials." These credentials are validated along the data path of routers. They are verified at every hop along the path, which gives us overall protection against a huge variety of attacks.

2. No Per-Connection State

The main advantage of capabilities is that they do not add per-connection state to the network. In simple words, the network can filter attack traffic without keeping any end-to-end filtering state. Stateless filtering removes the need for traditional packet filters. It also removes the need for inter-ISP relations like bi-lateral filtering agreements, since no filtering state is explicitly exchanged

between any ISPs. An ISP needs to just upgrade a subset of its routers to perform marking and verification. Thus, capabilities are inexpensive and easy to deploy and they protect legitimate communications against DoS.

3. Denial of Capability

   According to Argyraki et al.,[7] there is a simplest possible attack on Capabilities-Based network. It is nothing but the attack on the capability granting mechanism itself. Such an attack is called as "Denial-of Capability" (DoC), i.e., denial of service attack on the capability granting system. An attacker can flood the receiver with a millions of capability requests and exhaust the link. Legitimate clients that had already obtained a capability before the attack has no problem connecting to the receiver. But, a new legitimate client has only a little chance of getting a capability. They also suggested a solution for DoC through datagram approach. One approach is to perform Internet-wide fair queuing of capability requests, i.e., to configure a set of routers to fair queue capability requests per incoming network interface. As a result, no interface gets to forward more than its fair share of requests. Similarly, Anderson et al.,[6] stated that if widely deployed throughout the Internet and, in particular, close to the edges, this form of policing can automatically rate-limit floods of capability requests. A proposed solution to DoC is discussed in the related work section.

   Implementing such a network architecture poses several technical challenges. Some of them are listed below:

   - Where to place the credentials in a packet?
   - Designing the credentials such that the size of the packet does not increase with the number of the hops.
   - How to test and implement such an architecture?
   - How to filter the packets based on the credentials that they carry ?

- What are the metrics used to compare it with the existing internet security architecture?
- How to design the credentials?
- Comparison between the existing Internet architecture and data path credentials architecture with respect to attack mitigation.

CHAPTER 2

RELATED WORK

Implementing such a new security architecture might sound impossible in the existing best-effort Internet, but the idea of router virtualization [5] made it possible to implement such a domain-specific architecture in parallel to the existing Internet.Various new systems like GENI and VINI are being developed that can support the data path credentials architecture that is proposed here [10, 26].

Several capabilities-based networks have been proposed focusing on security related issues. Anderson et al. [6] proposed a defense against DoS using capabilities and yang et al. [29] proposed a system design that mitigates DoS using capabilities to identify legitimate traffic. Ballani et al. [9] came up with a capabilities-based networks that provides defense against more attacks.The validation of capabilities are done at one or a small number of nodes in all of these capabilities-based networks. The design used in Data Path Credentials validates all packets at all nodes along the data path to restrict malicious traffic. An edge-to-edge filtering architecture against DoS was proposed by Felipe Huici et al. [20] that uses IP encapsulation to tunnel traffic between the edge networks. Thus, the DoS floods are identified and mitigated at a point that is located close to the traffic destination. In this design, we can identify and mitigate unwanted traffic within small number of hops from the source, which reduces the consumption of the overall network resources.The main idea of data path credentials for assurable global network was proposed by Wolf [28].

Ethane system proposed for enterprise by Casado et al. [15] is similar to the capabilities-based network that is proposed here. Ethane defines a single wide fine-grain policy and then enforces it directly. It couples simple flow-based ethernet switches with a centralized controller that manages the admittance and routing of flows. Ethane works efficiently for enterprise scenario where access is controlled by flow entries in the forwarding table of an edge switch, but fails when access permissions are issued by a node that does not have control over the forwarding table of the switch. Data path credentials architecture proposed here provides a separation between the nodes that issue the credentials and those that enforce them. A policy controller can also be assumed that basically takes care of enforcing access control in the data path. Argyraki et al. [7] identified that the capabilities-based networks are susceptible to denial of capabilities (DoC) attacks on the capability-granting system. A solution to such a DoC attack "Portcullis" was proposed by Parno et al. [21] by using proof-of-work to reduce the malicious node's capability requests.

Even though the architecture proposed here is similar to previously proposed off-by-default architectures, several new ideas are introduced by considering computationally efficient credentials and by implementing them in different network scenarios. The defense against the DoC attacks are inherently part of the design and therefore no proof-of-work scheme is required. We also show that DoC attacks are isolated to affect only a small number of routers that are close to the attack source, which reduces the effect on the overall network. It is always hard to determine the source of a DoS attack because of IP address spoofing. Packet marking was proposed as an alternative of defense against DoS attacks by tracing back the path of malicious traffic even if an attacker spoofs IP addresses. This process can be probabilistic [24] or deterministic [11]. Another idea for defense against DoS was proposed by Snoeren et al. [25] using

"Hash-based IP traceback." It can be achieved by extending routers to maintain database of packets that have been forwarded. These audit trails eventually gives the source of a packet when analyzed. Thus, once the evil nodes are identified, they can be filtered actively by the concept of "Active Internet Traffic Filtering: real-time response to DoS attacks" proposed by Argyraki [8]. The above mentioned concepts of packet marking, traffic analysis and filtering are reactive in nature rather than proactive.

The design of data path credentials are based on Bloom filters. It was introduced by Burton Bloom [13] and has several applications in network systems [14]. Some of them are IP prefix matching [16], regular expression matching for intrusion detections [17] and packet traceback [25]. We use Bloom filters for the design of credentials that are derived from hash functions such as MD5 [23] and SHA-1 [18]. The credentials data structure are further expanded to consider the density of set bits, which is called as the fill level. The fill level problem is addressed through scalable [4] Bloom filters. It is not applicable in our design since we use fixed-length credentials to limit the packet header sizes.

CHAPTER 3

DATA PATH CREDENTIALS

This section begins with a discussion about the security requirements and the capabilities of an attacker. Next, we will discuss about the overall network architecture and router architecture for data path credentials with the system design and the operation in detail.

3.1 Security Model

The security requirements discussed here is based upon a domain-specific networks such as networks used for financial transactions, military communications, etc. These networks are expected to be deployed in parallel to the existing Internet through virtualization or through use of a dedicated infrastructure. Thus this design for security requirements would be inefficient in the best-effort Internet that is extensively available right now.

3.1.1 Security Requirements

The following are the security requirements:

- Network Access: Prevention of unauthorized network access by permitting only authorized users to establish a connection in the network.
- Packet spoofing: Packet spoofing by unauthorized users should not be possible.

- Traffic Injection:An unauthorized user or an attacker should not be able to inject junk traffic into the network.
- Detecting denial-of-service attacks: The sources responsible for the denial-of-service attacks must be identified and quarantined from rest of the network.
- Prevention of Intrusion: Intrusion into the network must be prevented by allowing the connections from the outside world to the network only in specified ports to avoid attacks like port scans.
- Prevention of Extrusion: Extrusion to the outside network must be prevented by controlling the connections from the end systems to the outside network.

From the above security requirements, we can see that emphasis is put on basic security needs, i.e., authorization and availability by making sure that packets that are positively identified are forwarded in the network. The rest of the basic security requirements such as access control, confidentiality, and integrity are addressed by the existing key management and cryptographic solutions in the later sections.

3.1.2 Capabilities of an Attacker

An attacker can :

- Read any packet traversing an attacked router.
- Modify any packet traversing an attacked router.
- Send any packet from the attacked router.

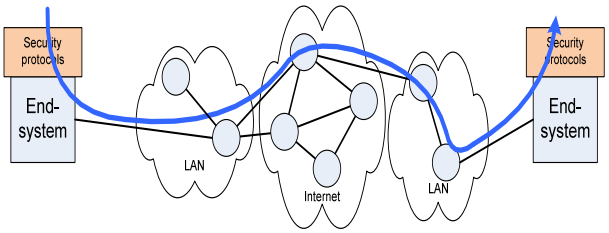
The following are the restrictions on the capabilities of the attacker. An attacker cannot:

- Spoof the identity of another entity in the network.
- Drop all or a subset of the network traffic on a router.
- Access all the nodes and links. It is limited to a subset of all network nodes.

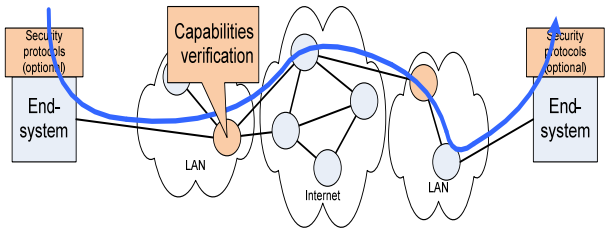
By, constraining the capabilities of an attacker we can keep the attack scenarios and security requirements within the scope.

3.2 Network Architecture

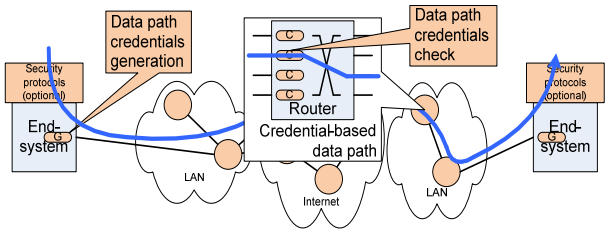
The network architecture that is proposed here is shown in Figure 3.1(c) that is compared to the traditional internet architecture shown in Figure 3.1(a) and the existing capabilities-based networks in Figure 3.1(b). The main concept is to append network traffic with credentials that can be validated in the data path on every hop. All the routers on the data path performs check for credentials and thus forwards only those packets with valid credentials. All other traffic with invalid credentials are discarded. As, we can see that all the routers along the data path participate in validating the traffic, which contrasts to the traditional Internet architecture where security protocols are constrained to the end-systems (e.g., cryptographic protocols) or isolated routers (e.g., firewalls or intrusion detection systems). In addition to credential check along the data path, end-system protocols can provide orthogonal security features of integrity and confidentiality. When comparing credential-based data path architecture to the existing capabilities-based networks, we can see that packet validation is not limited to just a few nodes along the path such as"verification points" in [6], edge routers in [20], or LAN switches in [15], instead performed at all nodes along the path. Thus, the responsiveness of the network to various attacks is also increased.



(a) Security in Existing Internet



(b) Capabilities-Based Network Architectures



(c) Credential-Based Data Path Architecture

Figure 3.1. Connection Setup to Establish Credentials

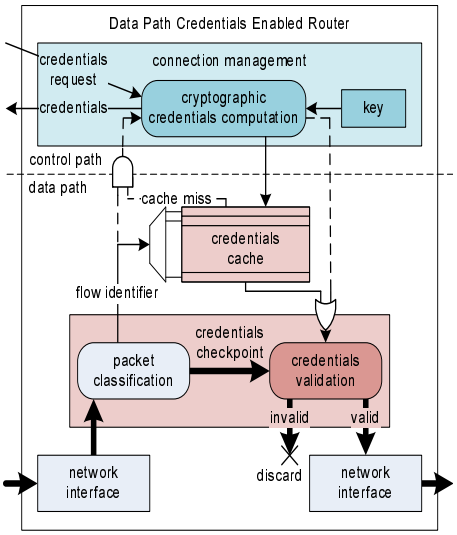


Figure 3.2. Router Architecture

3.3 Router Architecture

The architecture of a router that implements data path credentials is depicted in Figure 3.2. Connections are managed and credentials are created in the control path. An end-system can request credentials from a router for a particular flow. The credentials are then created based on the flow characteristics and router's cryptographic key. The process of generating credentials are discussed in chapter4. These credentials are then sent back to the end-system and also stored in the local credentials cache for future use. The credentials are the augmented to all the packet headers that traverse the network. When a router receives a packet for forwarding, first it classifies the packet according to the flow it belongs. Then, the router performs the credentials validation by comparing the credentials in the packet with those that it retrieves from the credentials cache. If the credentials matches , then the packet is considered to be valid and forwarded. If the credentials in the packet does not match the credentials from the cache, then the packet is considered to be invalid and discarded. If the credentials cannot be found in the credentials cache, it might be due to

the limited size of the cache or due to an invalid packet. A re-computation of credentials can be initiated before discarding the packet that is shown in Figure 3.2 as dashed lines. Re-computation of credentials might increase the system's vulnerability to denial of service (DoS) attacks since the cryptographic computation of credentials is a computationally expensive operation. Thus , it is important to maintain a credentials cache that is large and the requests for new credentials always gets the priority over credentials re-computations.

More specific decisions can also be taken on a packet rather than simply forwarding or rejecting them. If quality of service (QoS) is considered, then the QoS parameters and flow state can be stored in conjunction with the credentials information. During the validation of credentials, the packet can be scheduled according to the QoS properties, so that the desired QoS can be provided.

3.4 Connection Management Scenarios

Connection management is an important aspect of data path credentials architecture since it addresses one of the key problems that appeared in prior designs networks using capabilities[9]. The potential target for denial of service attacks are the control path of such networks [7]. Our architecture implements connection setup as an incremental process as shown in Figure **??**. The basic idea is that an end-system cannot send a credentials request to a router unless it has the valid credentials for the entire path upto that router. The main advantage of such a set-up is that any DoS attack on the control infrastructure can only target the router close to the attacker's source. Unwanted traffic cannot propagate unless the source can identify itself as an authorized source. In case, if an attacker poses an authorized end-system, the DoS attack can be traced back and mitigated by the security measures. To explain the connection management, three different communication models are considered to demonstrate the

generality of the data path credentials concept. The three scenarios explained are:

- Unicast: Communication between a single sender and a single receiver over a network.
- Multicast: Communication between a single sender and multiple receivers on a network.
- Network Coding: Packets traverse multiple paths to the destination and may be coded together with other transmissions (explained in section 4.4.2).

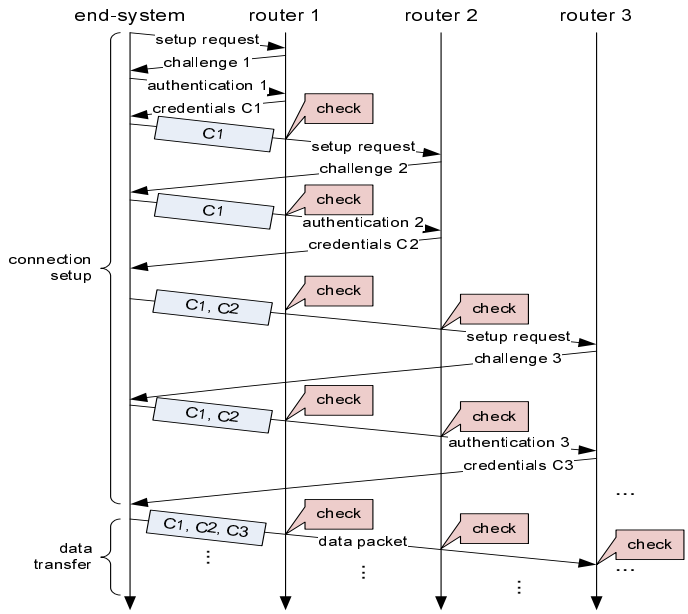


Figure 3.3. Connection Setup to Establish Credentials

3.4.1 Unicast

Figure 3.3 shows the connection establishment process involving multiple routers. For the end-system to obtain credentials from all the routers, first it needs to obtain the credentials from router1. At this point, router 1 can challenge the end-system to authenticate itself and negotiate access policies. If the router determines that the end-system is eligible to transmit data across the router, it provides the credentials C1. The credentials C1 must be included in all future transmissions through router 1 including the credentials request to router 2. This process is repeated until the end-system receives the credentials from the all the routers along the data path and finally the data transmission starts. The set of all credentials (C1, C2, and C3 shown in Figure 3.3) is then carried in each data packet. Each and every packet is validated on every router. If the packet carries the valid credentials, the packet is forwarded, else it is dropped. We can see that several messages are exchanged with every router along the

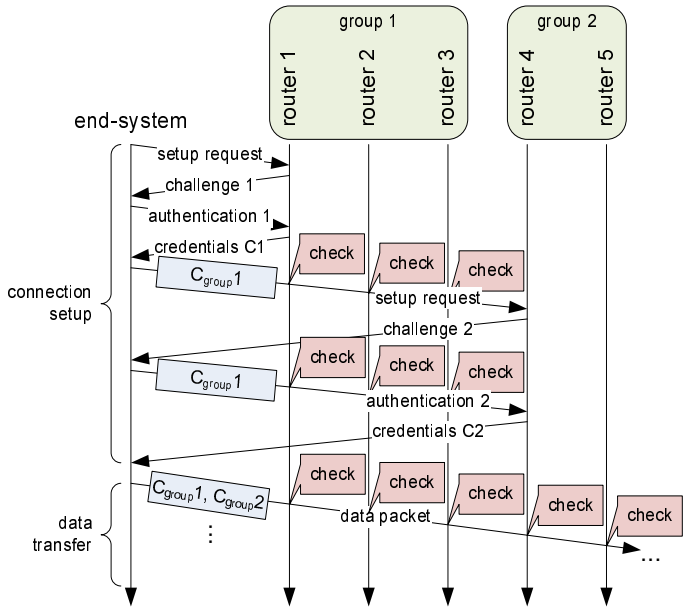


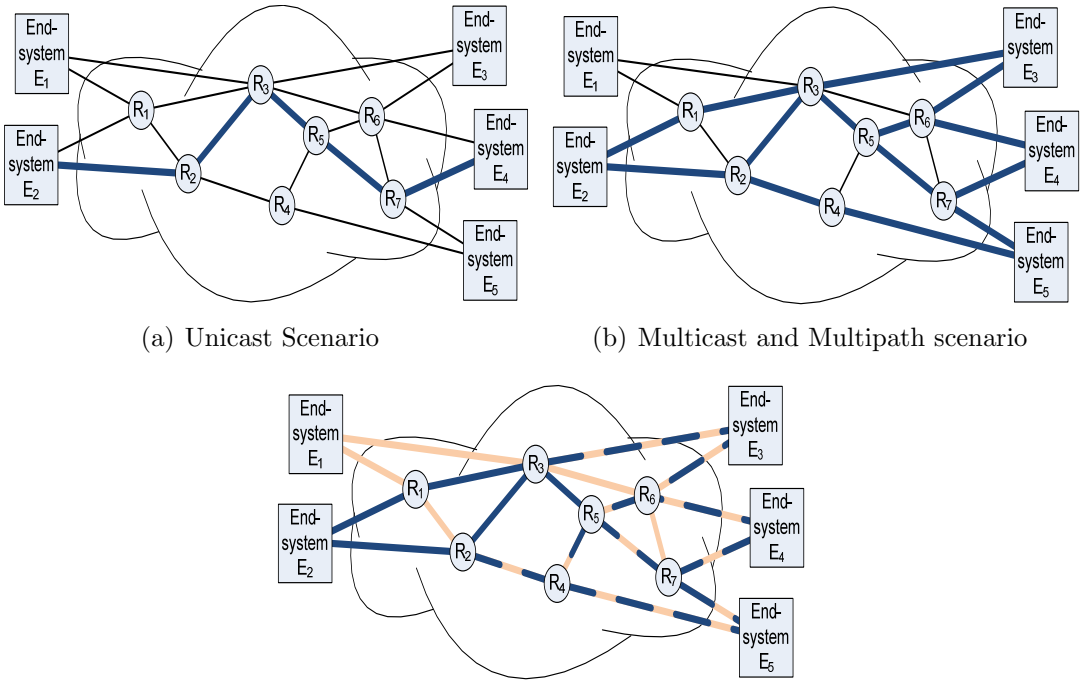
Figure 3.4. Connection Setup using Group Credentials

path to establish a single connection. To avoid this overhead, two approaches can be followed:

- Credentials Reuse: When multiple connections are established between two end-systems in parallel or within a given time window, credentials could be reused.
- Group Credentials: Routers that belong to the same autonomous system can be grouped together to create group credentials. In this case, any router can issue credentials that are valid to traverse any set of routers in that group. Thus, the number of credential requests per connection is reduced to larger extent as explained in Figure 3.4

3.4.2 Multicast and Network Coding

To explain the capabilities of the credentials-based data path design, more challenging scenarios other than unicast like multicast (and multipath) and network



(a) Unicast Scenario

(b) Multicast and Multipath scenario

(c) Network Coding Scenario

Figure 3.5. Connection Setup to Establish Credentials

coding can be considered. All these communication modes are shown in Figure 3.5. In unicast, (Figure 3.5(a)), the set of credentials includes only those along the path from source to destination. In multicast and multipath, packets get duplicated inside the network (see Figure3.5(b)). When the packets get duplicated, its credentials get duplicated, too. Thus , the end-system needs to include credentials for all routers that the packets will traverse along the multicast/multipath graph. As we can see from the Figure 3.5(b), an end-system sends the packets through all the routers in the network to reach the destinations. Thus, it needs to include the credentials for all the routers that the packet will traverse.

The third scenario is the Network coding [3]. It was recently proposed to improve ene-to-end data transmission in wireless networks. Packets traverse multiple paths to reach the destination and may be coded with other transmissions.

These operations can be reversed to obtain the original packets at the receiver. In such a communication scenario, data path credentials from both the sources need to be combined as shown as dashed lines in Figure 3.5(c). As we can see, some routers may overlap in the coded packets' paths. Since, the flow identifiers are different, the credentials are also different. Therefore, all the credentials from both sources need to be added into the set of credentials carried by the packet.

As we can see from the network coding scenario, there may be situations where a large number of data path credentials are necessary to guarantee successful forwarding of packets by all routers involved along the data path. In the later sections, it will be shown that the design of credentials only requires constant space for most practical communication scenarios. Specifically, we do not require space that increases linearly with the number of credentials that need to be provided and thus the above scenarios can be implemented successfully. The security analysis in the later sections provides quantitative performance and security tradeoffs for each of the above mentioned scenarios.

3.5 Authentication, Authorization and Access Control

There are more important processes that needs to be considered before issuing credentials such as how identities are managed, how authentication is performed, and how access control is used to authorize the network access. These issues are very important when it comes to realistic deployment of such a new architecture. For the credential-based data path architecture, we can use the existing security concepts and protocols to solve the above mentioned issues.

Identities can be provided to the end-systems or users in the network through a Public Key Infrastructure (PKI) [22] or a more complex federated trust model

[12]. These identities can be provided to end-systems or to individual users. In this architecture, the term "end-system" and "user" are used interchangeably to identify an entity that is the source or sink of a network connection. The system could also be extended to distinguish between individual users who request network access from possibly shared end-system devices. Once the end-systems and the users are identified, we can use conventional access control system like role-based access control [19] to determine the privileges of each and every user or end-system in the network.

The decisions for the security services like authentication, authorization, and access control are made by the routers during the initial connection setup. As we can see in Figure 3.3, there is only single exchange of packets for this process, it is also possible that a more comprehensive exchange takes place to establish access privileges. Once these steps are completed, data path credentials are used to enforce these network access policies by validating each packet at every hop along the data path.

CHAPTER 4

CREDENTIALS DESIGN

In this section, we will move forward and discuss how these credentials can be designed efficiently and the requirements for such credentials.

4.1 Requirements

The design of the credentials depends on several factors. Let us see the basic requirements of the credentials for the data path security concept:

(a) Security Requirement: The whole network infrastructure depends on the security of the credentials. It is always very important that the credentials are available only to authorized traffic in the network. Thus, the credentials must be difficult to duplicate.

(b) Performance Requirement: The credentials need to be validated for every packet on every router. So, it is necessary that the credentials must be validated with low computational requirements without too much of overhead.

(c) Size Requirement: According to the proposed here , credentials for every router along the data path of a connection need to be carried in each packet's header, it is very important that the total size of the credentials is limited to a fixed size.

The first requirement can be satisfied by the use of traditional cryptographic solutions. The second and the third requirements pose a new set of challenges. Power constraints becomes one of the major issue as the networks connect an increasing number of embedded devices as end-systems and as intermediate hops. Cryptographic operations require a several orders of magnitude more operations than the conventional packet processing and thus it should be limited to the initial connection setup process. In order to satisfy the third requirement, it is not practical to simply chain all the credentials in the header of the packet. A limit on the header size would constrain the maximum hop count along the path or the size of the multicast tree. Thus, we need a solution where all the credentials can be represented by a single fixed-length data structure.

4.2 Bloom Filter Based Credentials

With the above requirements for the credentials in place, we now turn to the design of the credentials data structure that is based on Bloom filters. The basic attribute of the Bloom filter structure is that, it can maintain multiple credentials at the same time. Thus, when the packet is transmitted, each router checks for its own credentials in the data structure and thus validate the packet. Bloom filter data structure is a very efficient way for the design of the credentials.

4.2.1 Bloom Filters

Let us briefly review the concept of Bloom filters that is used here to design the credentials.The Bloom filter, proposed by Burton H. Bloom in 1970, is a space-efficient probabilistic data structure that is used to test whether an element is a member of a set [13]. Since the test is of a probabilistic nature, false positives are possible (i.e., elements that are not members of the set may be reported to be members), but false negatives are not possible (i.e., elements that are

0	0	0	0	0	0	0	0	0

Figure 4.1. Empty Bloom Filter

members of the set will never be reported as not being members). One of the key property of a Bloom filter is that it is not possible to perform a reverse operation where the list of members is extracted from the Bloom filter data structure. A Bloom filter is a bit array that can store m bits. Using k different hash functions $h_1(x)...h_2(x)$, an element x is mapped to k bit position in the array. An empty Bloom filter data structure starts with all array values set to 0 as shown in Figure 4.1

To add an element x , the bits corresponding to the hash function values for element x is set to 1. As multiple elements are added, it is possible and intended that set bits overlap, which are combined with a logical OR function.To query for an element (i.e.,test whether it is in the set), feed it to each of the k hash functions to get k array positions. If any of the bits at these positions are 0, the element is not in the set. Only if all of these bits are set to 1, then the element is a member of the set. Since the data structure allows that set bits from different elements can collide in the array, an element that is not a member os the set may be reported as being a member. This occurs when the hash functions of this element map to bits that have been set by other members in the array, i.e., k collisions. This probability increases as more members are added to the set, i.e., n increases and thus more bits are set. This probability can be decreased by using larger array, i.e., larger m.

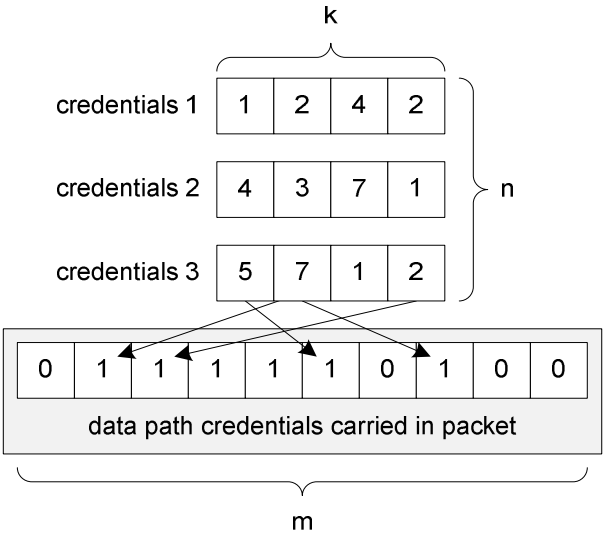


Figure 4.2. Credentials Data Structure

4.2.2 Credentials Aggregation

The Bloom filter data structure is used as data path credentials for packets that traverse the network by storing each credentials from each router along the data path. As we saw in section 4, the source node negotiates permission to transmit across a router during connection setup. If router j, $1 \leq j \leq n$ permits transmission, it provides the end-system with its credentials r_j. The router credentials are nothing but the set of indices $r_j\,[i]$, $1 \leq i \leq k$ of bits that are set in the Bloom filter array. The credentials from all the routers along the path are then superimposed using logical OR operation in the Bloom filter data structure. This gives us the aggregate credentials c that consists of a single bit array of size m that are augmented with each data packet. The process of generating the credentials is shown in Figure 4.2. Here three credentials are aggregated to a set of 1's in the Bloom filter data structure.

If router j receives a packet with aggregate credentials c, it checks the value of all bits that were provided in the router credentials r_j. If the credentials are

valid, then

$$\prod_i c[r_j[i]] = 1, \tag{4.1}$$

where the product is the equivalent of a logical AND operation. If the aggregate credentials do not contain the router credentials for the particular router, it is likely that one of the bits in the credentials c does not contain a 1 at one of the router credentials' bit positions. Thus, the validation of credentials fails. This is probabilistic in nature. A router may accept a packet with the same probability as a false positive appears in the Bloom filter. But, all packets are successfully delivered only if all the routers let them pass. Thus, a packet with invalid credentials would need to encounter a false positive on every router along the path. This probability decreases geometrically with the number of hops in the path and thus is practically very small. It will be explained later in Section 6.

4.3 Credentials Security

The security of the entire network architecture depends on the security of the credentials. It should not be practically feasible to generate duplicate credentials for malicious traffic. Thus, in the case of the Bloom filter credentials data structure, it should be difficult to guess the bits that are set by any given router. This can be achieved using cryptographically strong hash functions like MD5 [23] or SHA-1 [18] where router j uses k secret keys $s_j[i]$, $1 \leq i \leq k$. The cryptographic hash function $h_i(s_j[i], f)$ uses router j's key for bit index k to determine which bits are set in the aggregate credentials. It is also important that this hash function uses flow identifier f as an input. The flow identifier (e.g., based on a 5-tuple hash) helps in avoiding attacks where credentials from an authenticated connection are used by a different connection. Suppose, if

the router credentials are used by a different connection, the validation step (Equation 4.1) fails.

The concept of generating credentials based on the cryptographic hash functions and flow identifiers ensures the following properties:

- The data path credentials are different for different flows even though they traverse the same set of routers because of the use of flow identifier f as a parameter in the hash function.
- The data path credentials for flows that traverse different routers are different, because a different set of router credentials are superimposed in the data path credentials. The data path credentials differ because each router has a different set of secret keys s_j.
- The data path credentials are difficult to fake because the result of the cryptographic hash function h_i cannot be guessed without availability of secret keys s_j.
- Even though the generation of credentials is computationally expensive that involves n x k cryptographic hash operations, the process of credential check are very simple. The credentials can be checked by performing k lookups in the credentials c and verifying that the Equation 4.1 holds. This requires that each router remembers the router credentials r_j for a particular flow. It can be done by maintaining the credentials cache as shown in Figure 3.2. If the credentials for a flow cannot be found in the cache, the router credentials can be recalculated using s_j at a higher computational cost.
- The data path credentials are of small and constant size since all router credentials r_j can be superimposed (logical OR operation) into a single Bloom filter data structure.

- One of the important property of Bloom filter is that the credentials cannot be reversed to obtain the hash keys used by any of the routers. This is due to the reason that the cryptographic hash functions cannot be reversed. Thus, it is not possible to use valid credentials to create fake or duplicate credentials for a different flow.

From all these important properties of data path credentials, it is possible to provide security features at network architectural level as discussed in Section 4.1. A more detailed discussion on how security requirements are satisfied will be provided in later sections.

4.4 Density Limit

After the discussion of the credentials and their security properties, one important observation regarding the credentials must be mentioned here. There exists a very simple attack to circumvent the credentials check. An attacker could set all the bits in the credential data structure to 1. Such credentials would always satisfy Equation 4.1, no matter what secret keys s_j or flow identifiers f are used. This is clearly an undesirable property. To make the data path credentials to such an attack, an additional concept is introduced to the Bloom filter. A "density" metric *d(c)* that reflects the number of 1's in the credentials c as a fraction of the total size:

$$d(c) = \frac{1}{m} \sum_i c[i]. \tag{4.2}$$

To consider the credentials valid, we require that the density is equal or below a certain threshold: $d(c) \leq d_{max}$. If the density is higher, we can assume the credentials to be invalid and thus reject the packet. If the threshold is chosen

to be too low, even the valid credentials will be rejected.In the later section, an equation will be derived that will be helpful to estimate the expected number of set bits in the credentials data structure based on the number of routers involved along the data path.

4.5 Group Credentials

The group credentials was explained in Figure 3.4 and providing credentials for a groups of routers can reduce the connection setup overhead. Group credentials can be implemented by simply making those routers to share the same secret keys. Thus, router credentials issued by any router in the group sets the correct bits in the aggregate credentials c to make sure that all routers in the group let the packet pass. If the end-system is not aware of the grouping of routers, it negotiates router credentials with each router individually. Since the router credentials from all routers in the group are the same, the resulting aggregate credentials c will have the same bits set to 1.

CHAPTER 5

PROTOCOL IMPLEMENTATION

To implement the functionality of data path credentials as discussed above , we designed and prototyped a new protocol. We call this new protocol as Data Path Credentials Protocol (DPCP), which is located between the network layer and the transport layer of the Internet Protocol stack. Let us see how the header for DPCP is designed.

5.1 Credentials Header

The DPCP header is shown in Figure5.1. The overall size of the header is 28 bytes with the Bloom filter configuration as $m = 128$ bits and $k = 4$. The fields in the header are used as follows:

- Port numbers: This field is identical to the source and destination port numbers in the TCP and UDP headers. This field includes the source and destination port numbers (16 bits each) that are mere copies of the values carried in the transport layer of the packet. The transport layer protocol is assumed to either TCP or UDP to allow for packet classification. These port numbers are useful in the calculation of the five tuple hash (source IP, destination IP, source port, destination port, next protocol).
- Next protocol: This 8-bit field indicates the transport layer protocol header in the packet. It is similar to the next protocol field in the IP header. In

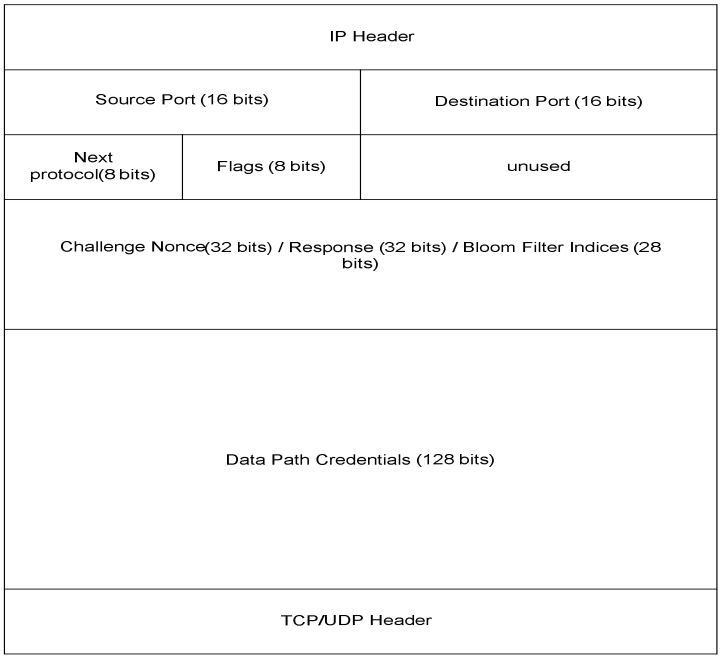


Figure 5.1. Data Path Credentials Protocol Header

this implementation, the IP header next protocol header has a value of 253 for DPCP, which is reserved for experimental protocols.

- Flags: There are four different flags used in the DPCP implementation to indicate packet types used during the connection setup. The different types are the following:
  - Setup flag(S): This flag indicates a packet containing a setup request.
  - Challenge flag(C): This flag indicates a packet containing a challenge to the end-system that requests credentials.
  - Response flag(R): This flag indicates a packet containing a response to the challenge posed by the router to the end-system.
  - Credentials flag(I): This flag indicated a packet containing the Indices generated that needs to be mapped to the Bloom filter array.

- Setup field:This field is used in multiple ways during the connection setup process. The use of this field is identified by the flags.
  - Challenge nonce(32 bits): The nonce is sent by the router to challenge the end-system.
  - Response (32 bits): The end-system sends the encrypted nonce as a response to the router's challenge.
  - Bloom filter indices(28 bits): These are the four indices generated by the router according its keys and the flow identifier(each with a size of $\log m = \log 128 = 7$ bits). The indices are generated only if the end-system proves its authentication. If the authentication fails, the packet will be dropped. The indices are also stored in the local credential cache, for future validation process. The end-system uses these indices to set the bits in the Bloom filter.
- Data path credentials: This 128-bit field is the Bloom filter that carries all the router credentials.

The Bloom filter indices are separate from the Bloom filter itself. The credential indices are the values that the end-system uses to set the bits in the Bloom filter array.

5.2 Router Processing

The decision diagram for DPCP processing on a router is shown in Figure5.2. When DPCP packets arrive on a router, it should distinguish if they belong to the connection setup request to this router or if they should be forwarded. A router also needs to forward connection setup requests directed to the other routers along the path. Thus, when the router gets a DPCP packet, it checks the carried by the packet with the credentials retrieved from the local creden-

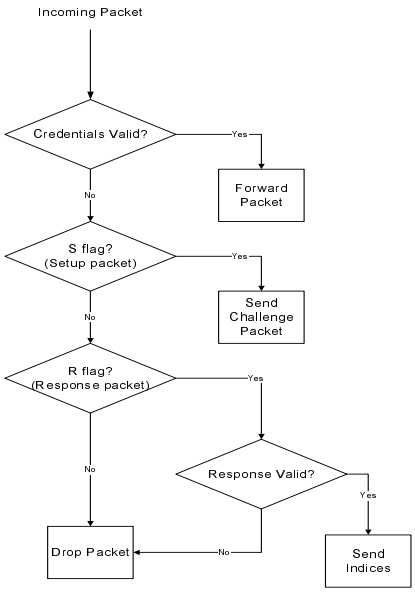


Figure 5.2. Decision Diagram for DPCP Processing on Router

tial cache. If the credentials are valid , the packet is simply forwarded to the destination. If the credentials are invalid, then the router checks whether the setup flag(S) is set in the received packet. If the condition matches, the router generates a nonce and sends it with a challenge flag(C) to the end-system . If the Response flag(R) is set, the router checks if the response received from the end-system is valid. If the response is valid, the router generates the credential indices using its secret keys and the flow identifier and send it back to the end-system with the credentials flag(I). In all the other cases, the packet will be dropped.

5.3 Testing

The implementation has been tested for various scenarios such as invalid nonce, invalid credentials, etc. For example, during the connection setup process, an end-system can send back an invalid encrypted nonce or a packet that may

contain invalid credentials. From Figure5.2, we can see that a packet might be dropped on several conditions. These essential conditions are tested and the results are shown below.

- Invalid Nonce: When an end-system sends a setup request to a router, the router replies with a random nonce. The end-system encrypts the nonce with the public key and sends it back to the router. We created a scenario where the end-system encrypts an invalid nonce and sends it to the router. The router drops the packet, since it has invalid nonce.
- Invalid Credentials: An end-system can send packets with invalid credentials with invalid bits set to 1. This condition was also tested and as expected, the router drops the packet, since the Bloom filter contains invalid credentials.
- All bits set to '1': We also created a scenario where an end-system sets all bits to '1' in the Bloom filter array. In such condition, the router must just forward the packet. But, this simple attack is defended by setting a limit on the total number of bits set in the Bloom filter. Thus, if the number of bits sets exceeds the density limit, the packets are dropped.

The tested conditions and their results are shown in the table below.

Table 5.1. Different Scenarios and Verdict on the Packets

Case	Verdict on the Packet
Invalid nonce	Packet Dropped
Invalid Credentials	Packet Dropped
Bloom filter with all bits set to '1'	Packet Dropped

5.4 Bidirectional Verification

Most of the communication in the Internet is bidirectional. Thus, it is necessary to validate the packets on the return path. During the connection setup

process in DPCP, the challenge(C) and credential packets(I) must reach the sender through number of routers. Thus, the routers need to setup and store the credentials for the return path. To avoid this process, the implementation is done in such a way that the routers use the same data path credentials for both directions of traffic. It is also assumed that the routes are symmetric. Thus, the challenge(C) and the credential packets(I) simply carry the same data path credentials that were carried in the original packet. In our implementation, the credentials are validated by matching the credentials carried by the packet with the credentials that are retrieved from the local credential cache for that particular flow. Thus,to implement this type of bidirectional verification, it is necessary to modify the flow identification process. In our system, the classification result of the 5-tuple of a connection in direction should match that of the connection in the opposite direction. This is achieved by sorting the IP addresses and the port numbers before providing them to the classifier. This sorting ensures that the classification result remains the same, since the IP addresses and the port numbers are swapped on the return path.

5.5 Implementation in Emulab

The DPCP is prototyped and implemented in Emulab[2] with a very simple topology of a chain of nodes with each hop incurring 10 ms of propagation delay. The main idea of "Deny by default" is implemented using Libipq[1]. Libipq is a development library for iptables userspace packet queuing. Netfilter provides a mechanism for passing the packets out of the stack for queuing to the userspace, then receiving the packets back into the kernel with a verdict such as ACCEPT or DROP. These packets can also be modified in the userspace before the reinjection back to the kernel. The packets are queued to the userspace using the iptable QUEUE target. Thus, all the packets are queued to the userspace

using ip_queue and the decisions are taken at the usespace. If the packet carries the valid credentials, it is simply forwarded by giving the ACCEPT verdict.

CHAPTER 6

SECURITY ANALYSIS

In the previous section, we saw the general concept of the data path credentials and the specific design of credentials based on the Bloom filters. Let us evaluate the qualitative security properties of this data path credentials architecture. It is also important to quantify these security properties to evaluate specific system configurations. In this section, we analyze the probabilistic guarantees that Bloom filters provide and present the results for specific usage scenarios. We will also discuss the data path credential architecture's inherent resistance against the denial of service attacks and how the security requirements are met.

6.1 Probability of Successful Attack

The primary goal of the data path credentials architecture is to identify valid traffic and thus not allow the transmission of attack traffic. Since the Bloom filter can yield false positives, the traffic with fake credentials may pass through the network. This false positive can be exploited by an attacker. Thus, it is very important to obtain a quantitative understanding on how likely this attack is for different system configurations. This problem can be related to the Generalized Birthday Problem (GBP) [27], but differs in that the GBP only considers a single false positive. In the case of data path credentials, we require false positives on every hop of the path for a successful end-to-end attack. The best attack from an attacker 's point of view is to send credentials with as

many bits set as possible. The more bits are set, the more likely the validation step described in Equation 4.1 satisfies. The limit to the number of bits set in credentials is given by the maximum density d_{max}. The attacker needs to decide which bits to set, since it is not possible to set all the bits in the credentials data structure. The attacker also cannot exploit any structure, since the credentials system uses cryptographic hash functions, which yield pseudo random outputs. Thus, choosing a random set of bits for the attack credentials is as good a choice as any other combination. Let us discuss the probability of successful attack for different scenarios.

6.1.1 Unicast

The attack traffic in the unicast scenario needs to traverse n hops from source to destination and encounter a false positive credentials check on each hop. When validating credentials, a router checks if all k bits determined by the router credentials indices are set. The density limit indicates that the probability of a particular bit being set in attack or fake credentials c_a is $P[\text{bit set in any } c] \leq d_{max}$. The attacker always wants to set as many bits as possible, so we can assume the probability of a particular bit being set in attack credentials $P[\text{bit set in } c_a] = d_{max}$. There is a possibility of that the hash computations yield the same index more than once due to the structure of the Bloom filter as shown in Figure 4.2. Thus, it is required to determine how many distinct bits are tested by a router or a set of routers along the path. This is nothing but the determining the number of bits set, *b(m,k,n)*, in a Bloom filter of size m with size k hash function and n stored items:

$$b(m,k,n) = m \cdot \left(1 - \left(1 - \frac{1}{m}\right)^{kn}\right). \tag{6.1}$$

The derivation of the expected value of *b(m,k,n)* is provided in the Appendix. In unicast, the number of router credentials in the aggregated credentials (i.e., h=n). The probability of a false transmission, *f(m,k,h)*, is

$$f_{unicast}(m,k,h) = (d_{max})^{b(m,k,h)}. \tag{6.2}$$

The expected number of bits set in the credentials data structure also gives an estimate on the longest path that can be supported by a particular configuration. If the expected number of bits set exceeds the density threshold, then even valid credentials may be rejected, which is nothing but a false positive. Let us consider a safety margin o with $o \geq 1$, and thus the valid configurations of m, k, and n must satisfy

$$o \cdot b(m,k,h) \leq d_{max}. \tag{6.3}$$

6.1.2 Multicast/Multipath

In a multicast/multipath scenario, the source needs to aggregate credentials from all the routers along the paths to all the destinations. In this case, to estimate the performance of the Bloom filter credentials, it is assumed that multicast is performed along a binary tree where each node corresponds to a router that duplicates the packet and send it to two more nodes. The height of the tree h assuming a balanced tree, relates to the number of leaf nodes l that are nothing but the multicast destinations as follows:

$$2^{h-1} < l \leq 2^h \quad or \quad h = [log_2 l]. \tag{6.4}$$

Let us make it simple by assuming a complete binary tree with $l = 2^h$ destinations. The number of internal nodes in such a tree corresponds to the number

of routers n that are encountered when multicasting:

$$n = 2^h - 1 \quad or \quad n = 2^{log_2 l} - 1 = l - 1. \tag{6.5}$$

Thus, 2^h - 1 router credentials have to be aggregated in the Bloom filter and the resulting number of bits set in the credentials is *b(m, k,* 2^h *- 1).* This limits the set of valid configurations to

$$o \cdot b(m, k, 2^h - 1) \leq d_{max}. \tag{6.6}$$

The exponential increase in the number of aggregated credentials require a much larger Bloom filter data structure. However, the size for this data structure grows less than exponential due to overlapping hash indices. To determine the probability of false positive transmission of attack traffic, we need to consider all $l = 2^h$ multicast paths. The false positive probability is given by:

$$f_{multicast}(m, k, h) = 1 - (1 - f_{unicast}(m, k, h))^{2^h}. \tag{6.7}$$

6.1.3 Network Coding

The analysis for network coding is similar to that of multicast. Each connection starts sending aggregate credentials 2^h - 1 router credentials and the routers in the network aggregate credentials from multiple connections when generating a network coded packet. Thus, by the time a packet reaches its destination, it may have gained credentials on every but the last hop along the path. For simplicity, let us assume that coding is done only across two packets at any node.

Therefore, there may be a total of $n = (h - 1) \cdot 2^h - 1$ credentials combined in the packet. Therefor, m and k need to be chosen properly such that

$$o \cdot b(m, k, (h-1) \cdot 2^h - 1) \leq d_{max}. \tag{6.8}$$

To determine the false positive probability, we need to consider how many packets have to be received by a node such that the network coding can be reversed. If we code packets on each of $h - 1$ hops, then $2^h - 1$ packets need to be received by the receiver for successful decoding. This corresponds to successfully achieving $h - 1$ false positive $(h - 1)$-hop multicast transmissions and a 1-hop unicast transmission. One hop cannot be multicast due to the structure of network coding. The probability of false positive is given by:

$$f_{networkcoding}(m, k, h) = (f_{multicast}(m, k, h-1) \cdot f_{unicast}(m, k, 1))^{2(h-1)} \tag{6.9}$$

CHAPTER 7

PERFORMANCE RESULTS

With the DPCP prototype implementation in Emulab, various performance results have been collected. The main functionality of the DPCP protocol was successfully implemented in which the packets are successfully forwarded when they carry the valid credentials and dropped when they do not. The performance evaluation is done with 12 nodes in Emulab. Each hop has a propagation delay of 10 ms.

7.1 Connection Setup

The connection setup of the DPCP is one of the critical and complex process that requires more time when compared to the conventional TCP setup. It is due to several complex cryptographic operations that takes place during the setup process. The measurement results for a single hop connection setup is shown in Figure7.1. We can see that, the amount of time required for sending out the response and credentials packets is more. This is due to the cryptographic operations involved in these steps. The router takes 3.9 ms to generate a nonce and send out the challenge packet to the end-system. Due to the RSA public key encryption taking place at the end-system, it takes around 40 ms to send out the response packet. The most computationally expensive is however, determining the Bloom filter indices that takes more than 1 s. This step involves RSA

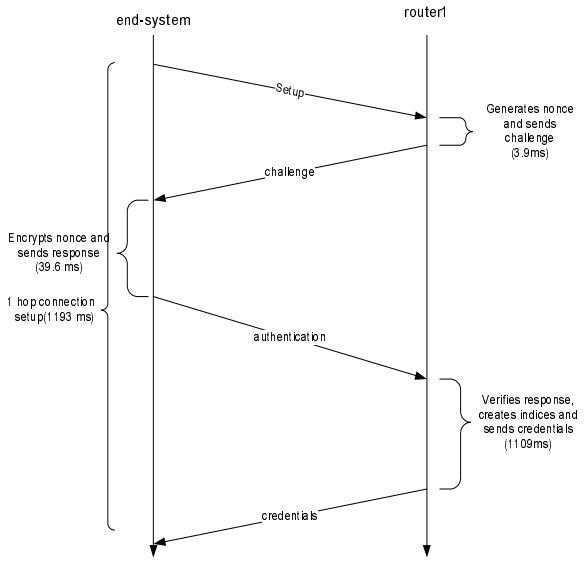


Figure 7.1. Connection Setup time for Single Hop

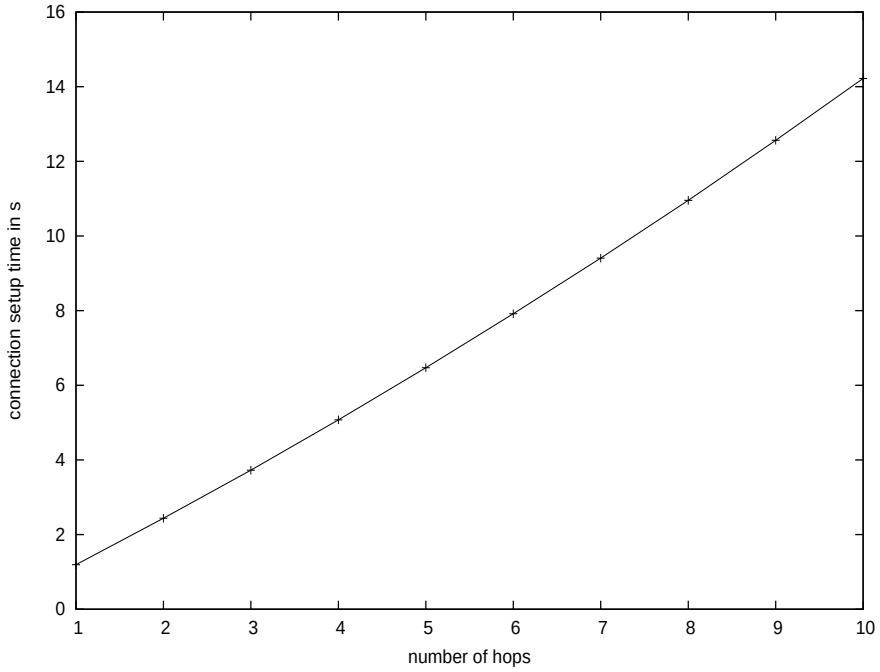


Figure 7.2. Connection Setup Time for Different Number of Hops

decryption process that verifies the end-system's response and generation of credential indices using SHA-1 cryptographic hash functions.

The growth in connection setup time for 10 hops is shown in Figure7.2. The increase in the setup time is slightly steeper than linear due to the increase in propagation time to reach the nodes that are farther away. These cryptographic operations can be performed faster with better software implementation or with high performance cryptographic co-processors. The Emulab network testbed does not provide cryptographic co-processors that can efficiently speed up these steps.

7.2 Delay Measurement and Throughput Calculation

The main objective of DPCP is to design efficient credentials that can be enforced on every hop and computationally simple for validation. Since, the credential validation process is very simple, which checks for bits that are set using the credential indices retrieved from the credentials cache, we can achieve high forwarding performance. The time delay in processing packets of different sizes are collected at a router. The data is collected at the incoming and the outgoing interface of the router for three different systems starting from the first data packet. Since, the DPCP system uses iptables to get user-space access to the packets that traverse the network, the throughput performance of the DPCP is compared with both conventional TCP and also with "TCP with iptables." The TCP with iptables is implemented to have a fair comparison for throughput as the conventional TCP runs entirely in the kernel. In the TCP with iptables implementation, the packets are moved to the user-space before performing normal forwarding.

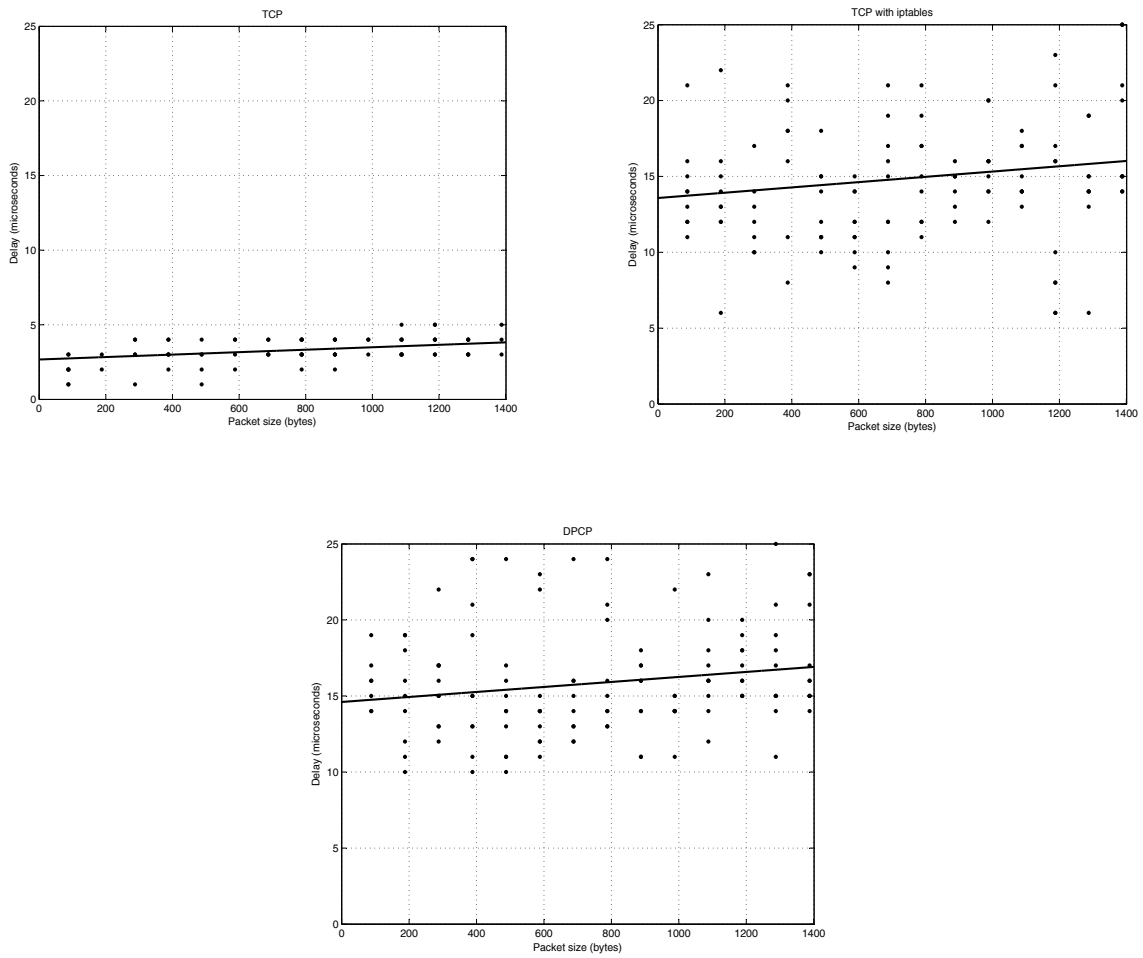
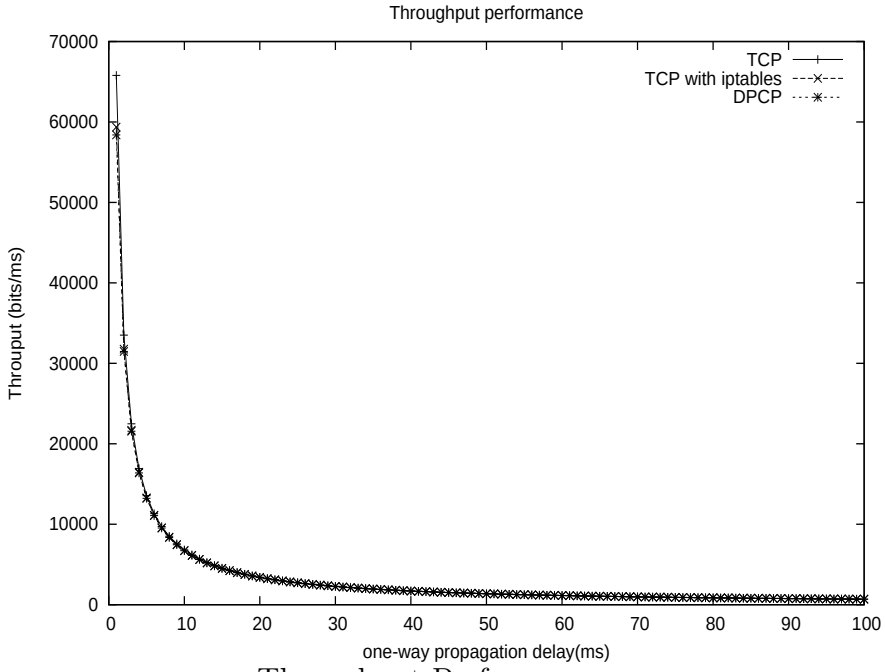


Figure 7.3. Processing Delay for Various Packet Size

The delay measurements for the three different systems are shown in Figure7.3. The processing delay for conventional TCP is very less when compared to that of TCP with iptables and DPCP as expected. The processing delays for the TCP with iptables and DPCP are nearly the same. The throughput performance can be calculated for these three systems using the RTT and MSS. We can see that the throughput of the TCP with iptables and DPCP are almost the same.

The throughput for each of the system is shown in Figure7.2. The ratio of throughput between DPCP and TCP with iptables is also calculated and plotted. See Figure7.4.



Throughput Performance

(Figure continued in next page.)

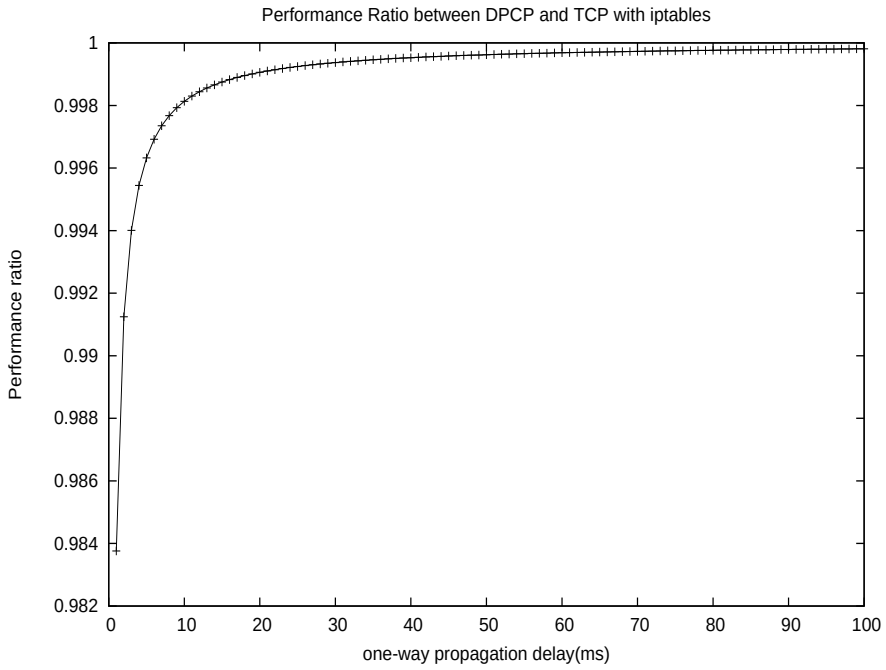


Figure 7.4. Throughput ratio between DPCP and TCP with iptables

CHAPTER 8

CONCLUSION

The Data Path Credentials Protocol prototype has been implemented successfully. We have shown that the credentials based on Bloom filter data structures can be efficiently implemented on all the routers along the path that permits only valid traffic to traverse the network. Thus, some of the main security problems are tackled by the capabilities-based networks that enforces traffic control on every hop. A malicious node cannot send traffic to the end-system unless and otherwise, it has the valid credentials for the entire path. Forging credentials is also not possible because the credentials are generated using strong cryptographic hash functions. The evaluation results shows that the cryptographic operations during the connection setup process are computationally expensive. However, once the connection setup is established, the throughput performance of DPCP is nearly equal to that of conventional TCP. The setup time can be improved by performing the cryptographic operations with better software implementation or with cryptographic co-processors. Since, the router system can be easily extended to generate and validate credentials in the data path, the Data Path Credentials Protocol provides a practical solution to provide architectural level security solution for the next-generation security architecture.

BIBLIOGRAPHY

[1]

[2] Emulab - network emulation testbed, www.emulab.com.

[3] Ahlswede, R., Cai, Ning, Li, S. Y. R., and Yeung, R. W. Network information flow. *IEEE Transactions on Information Theory 46*, 4 (July 2000), 1204–1216.

[4] Almeida, Paulo Sérgio, Baquero, Carlos, Preguiça, Nuno, and Hutchison, David. Scalable bloom filters. *Information Processing Letters 101*, 6 (Mar. 2007), 255–261.

[5] Anderson, Thomas, Peterson, Larry, Shenker, Scott, and Turner, Jonathan. Overcoming the internet impasse through virtualization. *Computer 38*, 4 (2005), 34–41.

[6] Anderson, Tom, Roscoe, Timothy, and Wetherall, David. "preventing internet denial-of-service with capabilities.". *SIGCOMM Computer Communication Review 34*, 1 (Jan. 2004), 39–44.

[7] Argyraki, Katerina, and Cheriton, David. Network capabilities: The good, the bad and the ugly. In *Proc. of Fourth Workshop on Hot Topics in Networks (Hotnets-IV)* (College Park, MD, Nov. 2005).

[8] Argyraki, Katerina J., and Cheriton, David R. Active internet traffic filtering: real-time response to denial-of-service attacks. In *ATEC'05: Proceedings of the USENIX Annual Technical Conference 2005* (Anaheim, CA, Apr. 2005), pp. 135–148.

[9] Ballani, Hitesh, Chawathe, Yatin, Ratnasamy, Sylvia, Roscoe, Timothy, and Shenker, Scott. Off by default! In *Proc. of Fourth Workshop on Hot topics in Networks (Hotnets-IV)* (College Park, MD, Nov. 2005).

[10] Bavier, Andy, Feamster, Nick, Huang, Mark, Peterson, Larry, and Rexford, Jennifer. In vini veritas: realistic and controlled network experimentation. In *SIGCOMM '06: Proceedings of the 2006 conference on Applications, technologies, architectures, and protocols for computer communications* (New York, NY, USA, Aug 2006), ACM, pp. 3–14.

[11] Belenky, Andrey, and Ansari, Nirwan. Ip traceback with deterministic packet marking. *IEEE Communications Letters 7*, 4 (Apr. 2003), 162–164.

[12] Bhatti, Rafae, Bertino, Elisa, and Ghafoor, Arif. An integrated approach to federated identity and privilege management in open systems. *Communications of the ACM 50*, 2 (Feb. 2007), 81–87.

[13] Bloom, Burton H. Space/time trade-offs in hash coding with allowable errors. *Communications of the ACM 13*, 7 (July 1970), 422–426.

[14] Broder, Andrei, and Mitzenmacher, Michael. Network applications of bloom filters: A survey. In *Proceedings of the 40th Annual Allerton Conference on Communication, Control, and Computing* (Allerton, IL, Oct. 2002), pp. 636–646.

[15] Casado, Martin, Freedman, Michael J., Pettit, Justin, Luo, Jianying, Mckeown, Nick, and Shenker, Scott. Ethane: taking control of the enterprise. In *SIGCOMM'07: Proceedings of the 2007 conference on Applications, Technologies, Architectures, and protocols for Computer Communications* (Kyoto, Japan, Aug. 2007), pp. 1–12.

[16] Dharmapurikar, S., Krishnamurthy, P., and Taylor, D. Longest prefix matching using bloom filters. *IEEE/ACM Transactions on Networking 14*, 2 (April 2006), 397–409.

[17] Dharmapurikar, Sarang, Krishnamurthy, Praveen, Sproull, Todd, and Lockwood, John. Deep packet inspection using parallel bloom filters. *IEEE Micro 24*, 1 (Jan. 2004), 52–61.

[18] Eastlake, Donald E., and Jones, Paul E. Us secure hash algorithm 1 (sha1). *RFC 1321, Networking Working Group* (April 1992).

[19] Ferraiolo, David F., and Kuhn, D. Richard. Role-based access control. In *Proc. of 15th National Computer Security Conference* (Baltimore, MD, Oct. 1992), pp. 554–563.

[20] Huici, Felipe, and Handley, Mark. An edge-to-edge filtering architecture against dos. *SIGCOMM Computer Communication Review 37*, 2 (Apr. 2004), 39–50.

[21] Parno, Bryan, Wendlandt, Dan, Shi, Elaine, Perrig, Adrian, Maggs, Bruce, and Hu, Yih-Chun. Portcullis: protecting connection setup from denial-of-capability attacks. In *SIGCOMM'07: Proceedings of the 2007 Conference on Applications, Technologies, Architectures, and protocols for Computer Communications* (Kyoto, Japan, Aug. 2007), pp. 289–300.

[22] Perlman, R. An overview of pki trust models. *IEEE Network 13*, 6 (Nov. 1999), 38–43.

[23] Rivest, Ronald L. The md5 message digest algorithm. *RFC 3174, Networking Working Group* (Sept. 2001).

[24] Savage, Stefan, Wetherall, David, Karlin, Anna, and Anderson, Tom. Network support for ip traceback. *IEEE/ACM Transactions on Networking 9*, 3 (June 2001), 226–237.

[25] Snoeren, Alex C., Partridge, Craig, Sanchez, Luis A., Jones, Christine E., Tchakountio, Fabrice, Kent, Stephen T., and Strayer, W. Timothy. Hash-based ip traceback. In *Proc. of ACM SIGCOMM 2001* (San Diego, CA, Aug. 2001), pp. 3–14.

[26] Turner, Jonathan S. A proposed architecture for the geni backbone platform. In *Proc. of ACM/IEEE Symposium on Architectures for Networking and Communication Systems (ANCS)* (San Jose, CA, Dec., 2006), pp. 1–10.

[27] Wagner, David. A generalized birthday problem. In *Proceedings of the 22nd Annual International Cryptology Conference on Advances in Cryptology* (Santa Barbara, CA, Aug. 2002), pp. 554–563.

[28] Wolf, Tilman. A credential-based data path architecture for assurable global networking. In *Military Communications Conference, 2007. MILCOM 2007. IEEE* (Florida, Oct. 2007), pp. 1–7.

[29] Yang, Xiaowei, Wetherall, David, and Anderson, Thomas. A dos-limiting network architecture. *SIGCOMM Computer Communication Review 35*, 4 (2005), 241–252.