A Credential-Based Data Path Architecture for Assurable Global Networking

Tilman Wolf
Department of Electrical and Computer Engineering
University of Massachusetts, Amherst, MA, USA
wolf@ecs.umass.edu

Abstract— **The main limitation for achieving information assurance in current data networks lies in absence of security considerations in the original Internet architecture. This shortcoming leads to the need for a new approach to achieving information assurance in networks. We propose a network architecture that uses credentials in the data path to identify, validate, monitor, and control data flows within the network. The important aspect of this approach is that credentials are tracked on the data path of the network, not just the end-systems, which implies that each and every packet can be audited. We present a credentials design that is based on Bloom filters and can achieve the desired properties to provide data path assurance.**

I. Introduction

The current Internet has been vastly successful in achieving global connectivity between a large number of different networks, devices, and users. This success is due to the openness of the system and the general "permit-by-default" design. However, this approach also presents one of the major shortcomings with respect to security and information assurance. Clearly, in some usage scenarios (e.g., military communication, financial transactions), information assurance is the top priority and a more conservative "deny-by-default" approach may be more desirable.

The main limitation for achieving information assurance in current data networks lies in the absence of security considerations of the original Internet architecture. Security protocols were added later to the protocol stacks of end systems. This leads to the following major problems:

- Current network routers are not designed to consider information assurance concerns. Thus, current security protocols are limited to operate solely on end-systems.
- Adding new information assurance features inside the network may violate the current Internet design. Thus, it is difficult to incrementally improve the capabilities of the network without causing incompatibilities.

There is a need for a new approach to achieving information assurance in networks. Information assurance encompasses more than just end-to-end security through cryptography. Assurance also addresses accountability, resource availability, end-system protection, information leakage, etc. It is important to understand that information assurance cannot be achieved by solely redesigning protocols and processes in the network's control plane. It is equally if not more important to also consider changes to the data plane so that malicious traffic can be quickly identified and blocked before it reaches its target and uses networking resources.

In this paper, we propose such an architecture that uses *credentials in the data path* to identify, validate, monitor, and control data flows within the network. The important aspect of this approach is that credentials are tracked on the data path of the network, not just the end-systems, which implies that each and every packet can be audited. This is an important step towards developing a network infrastructure that is highly sensitive and responsive to attacks. Specifically, the contributions of our work are twofold:

- Design of an architecture for data path credentials.
- Design of a specific credentials system using Bloom filter data structures.

These designs can provide the foundations for future research and for the development of prototypes assurable networks.

The remainder of this paper is structured as follows. Section II presents related work. Section III introduces the general architecture for credential-based data paths. The specific design and use of credentials is then discussed in Section IV. Section VI summarizes and concludes this paper.

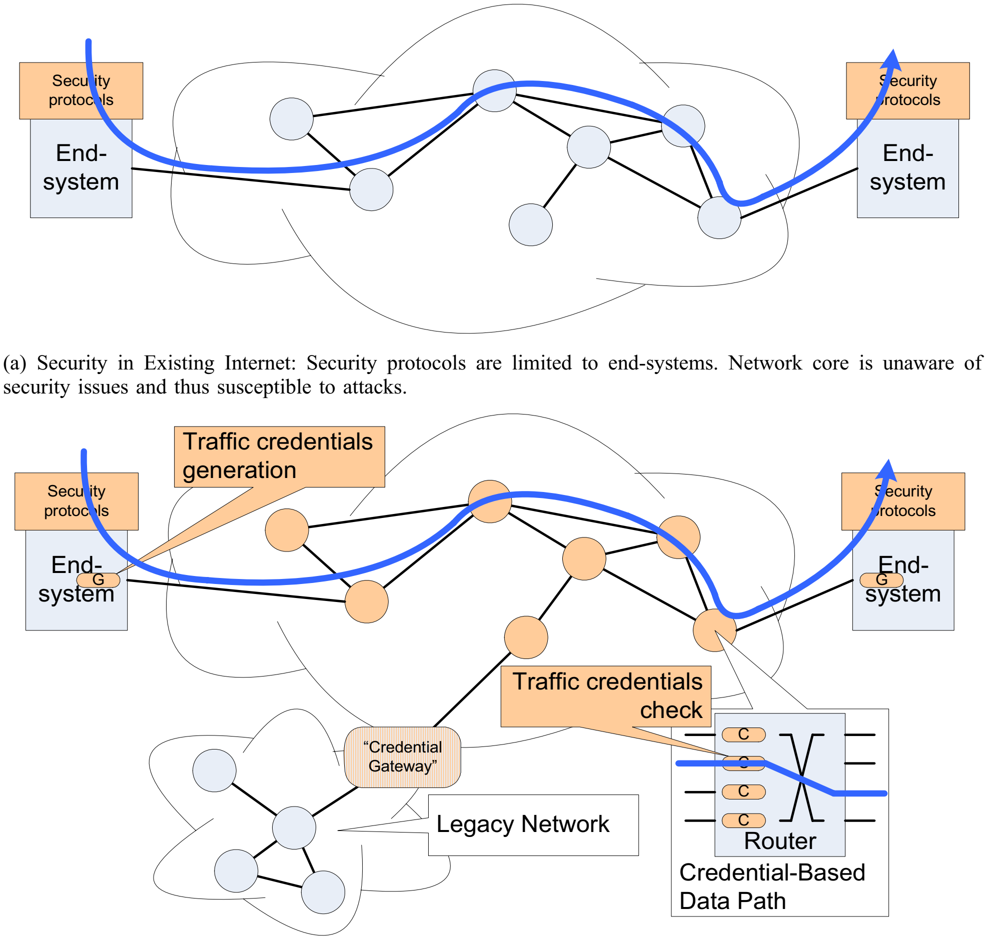
II. Related Work

The credentials that we propose in this work are based on Bloom filters. Bloom filters were introduced by Burton Bloom in 1970 [1] and found a number of applications in network systems [2] ranging from IP prefix matching [3] to regular expression matching for intrusion detection [4]. We adapt the use of Bloom filters for the use with signatures. These signatures are derived from cryptographic hash functions. Examples of such digest functions are MD5 [5] and SHA-1 [6]. We further expand the credentials data structure to consider the density of set bits in the Bloom filter (i.e., the fill level). Scalable Bloom filters have been proposed to circumvent the fill level problem [7], but are not applicable here as we need fixed-length credentials to put in packet headers.

III. A Credential-Based Data Path Architecture

We introduce the general network architecture for assurable networking in this section.

This work was supported in part by NSF grant CNS-0626690.



(a) Security in Existing Internet: Security protocols are limited to end-systems. Network core is unaware of security issues and thus susceptible to attacks.

(b) Proposed Credential-Based Data Path Architecture: Security features are built into network routers. Traffic is augmented with credentials generated by the end-systems. Routers check credentials to immediately identify violations of security policies. Legacy networks can be attached through gateways that translate between traffic credentials and conventional end-system based security.

Fig. 1. Comparison of Security Architecture in Existing Internet and Future Assurable Global Network.

A. Data-Path Credentials

The network architecture that we propose is depicted in Figure 1(b) and compared to conventional data networks in Figure 1(a). The key idea is to have traffic carry credentials with every packet. The features of credentials are as follows:

- Credentials identify a packet in terms of its source and destination (e.g., machine, user), its content (e.g., file transfer, encrypted voice communication), access privileges and transmission constraints (e.g., limited to military networks), etc.
- Credentials are obtained from intermediate routers during the connection setup phase. Router can check if the credentials match the packet as well as local and global security policies. Packets are no longer just transmitted "on good faith" as in the current Internet.
- Credentials can be validated with small amounts of processing and thus can be easily processed at high data rates. The creation of credentials may be more computationally demanding.
- Credentials can be used for identifying traffic flows in the network and monitoring their paths, performance, etc. This information can be used by the control plane to ensure correct and efficient network operation.

B. Assurance Impact

Enforcing credentials for all network traffic can be effectively used to diminish the impact of many common threats against information assurance:

- Unauthorized network access: source would have invalid credentials and thus traffic would be denied on the first hop.
- Transmission of sensitive data on unprotected network: credentials would not match local network policies and transmission could be terminated (or redirected to the appropriate network)
- Attacks on routing infrastructure or misconfiguration: since traffic is clearly identified through credentials, routers can track the path of traffic through the network and can identify routing changes, loops, etc.

- Signal intelligence attacks: credentials can help avoid the injection of traffic by the attacker (e.g., beacons or known-cleartext data) that reveals information.
- Denial of service attacks: credentials can be used for backtracking to squelch sources of distributed denial of service attacks or to identify and suppress control traffic for botnets.

Another benefit of the proposed data-path architecture is that it can be seen as a complement to other domains of information assurance. For example, improvements to control-plane operations (e.g., improved BGP routing system) can be leveraged in conjunction with a credential-based data architecture. Similarly, the deployment of MANETs is an important step for the Internet and improvements to ad-hoc networking (e.g., routing, etc.) can be used in conjunction with data-path credentials.

C. Technology Trends

One concern when designing a new data-path architecture is that of performance. Packet processing needs to be performed at data rates in the order of several Gigabits per second. There are several technological developments taking place that make the introduction of data-path level information assurance practically feasible:

- Processing capabilities on routers have significantly improved in the last decade. Chip-multiprocessor systems and FPGAs provide high-performance processing capabilities that can handle Gigabit data rates while providing programmability to adapt to new information assurance processes.
- Development of cryptographic algorithms for embedded systems is making progress. These algorithms are designed to implement secure communication with only a few thousand instructions per block. These capabilities can be utilized in high data rate router systems.

D. Example Connection Setup

Before explaining credentials in more detail in Section IV, we briefly illustrate an example connection establishment process to further illustrate our architecture. The space-time diagram shown in Figure 2 shows an end-system that establishes a connection that traverses three routers. During the connection setup, the end-system sends a connection setup request along the path of the connection. Each router responds to the end-system with a challenge. This challenge represents the negotiation process where a router authenticates the end-system, checks local and global policies, etc. The challenges may be different for each router and thus cannot be combined into a single challenge. Once the end-system has identified itself satisfactorily to a router, the router returns a signature. The set of all signatures is then combined into the credentials that are carried in each data packet. Each data packet is then checked on every router. If the credentials contain the signature of the router, the packet is forwarded. If the credentials do not match, the packet is discarded.

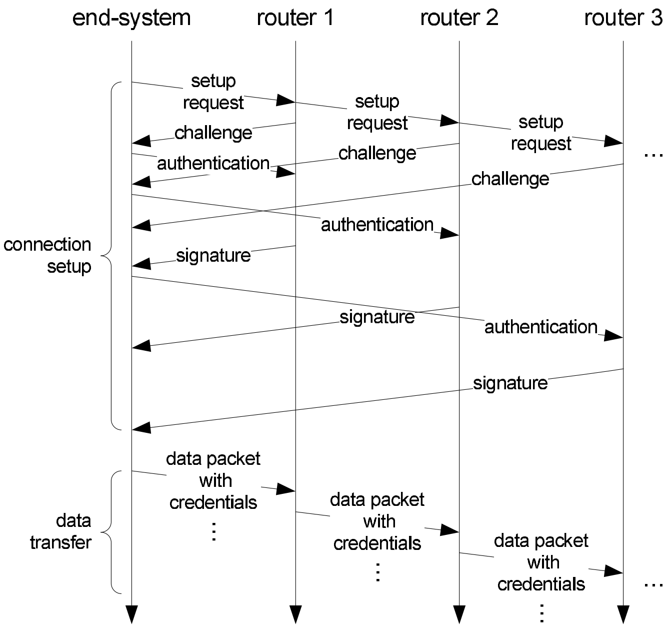


Fig. 2. Connection Setup to Establish Credentials.

IV. Data Path Credentials

With the concept of credentials in the data path introduced in the previous section, we turn to the question of what these credentials look like specifically.

A. Requirements

The requirements for credentials are driven by several conflicting needs:

1) Security Requirement: In order to provide an assurable network infrastructure, it is crucial that credentials are only available to authorized traffic in the network. Therefore, credentials should be difficult to fake.
2) Performance Requirement: Since credentials need to be validated for every packet on every router, it is necessary that credentials can be validated with low computational requirements.
3) Size Requirement: Since we assume a packet network rather than a connection-oriented network, credentials need to be carried in each packet header. Therefore, the size of credentials needs to be kept as small as feasible.

While the first requirement could be addressed by traditional cryptographic solutions, it is the second requirement that poses a novel set of challenges. As networks connect an increasing number of embedded devices (both as end-systems and as intermediate hops), power constraints are becoming increasingly important. Cryptographic operations require several orders of magnitude more operations than conventional packet processing [8] and thus need to be limited to the initial connection setup.

An implication from the third requirement is that it is not practical to set up different credentials for each hop along the path of a packet. A limit on the header size would constrain the maximum hop count along a path. Therefore, we seek a solution where a single set of credentials can authenticate a packet on all routers along the path.

B. Bloom-Filter-Based Signatures

To meet the above requirements, we introduce a data structure that is based on Bloom filters. The main idea is that this data structure can maintain multiple signatures at the same time. Thus, the signatures of all routers along the path of a packet can be placed into this data structure. These signatures are obtained during connection setup. When the packet is transmitted, each router can check if its own signature is present in the data structure and thus validate the credentials of the packet.

1) Bloom Filters: We briefly review the concept of Bloom filters to provide context for our work. A Bloom filter is a data structure that can be used to test if an element is a member of a set [1]. This test is of a probabilistic nature and 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 (i.e., elements that are members of the set will never be reported as not being members). One of the properties 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 consists of n arrays that can store m bits each. Using n different hash functions $h_1(x) \dots h_n(x)$, an element x is mapped to one position in each array. An empty Bloom filter data structure starts with all array values set to 0. When an element is added, one bit in each of the n arrays is set to 1 (as determined by the hash function for each array). As multiple elements are added, it is possible (and intended) that set bits overlap. When performing a check for membership of an element, the hash functions for each array are computed and it is checked if the according bits in all arrays are set. Only if all of these bits are set to 1, the element is reported to be a member of the set.

Since the data structure allows that set bits from different elements can "collide" in an array, it is possible that an element that is not a member of 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 all n arrays (i.e., n collisions). The probability of this occurring increases as more members are added to the set (i.e., more bits are set and thus can cause collisions). By using larger arrays (i.e., larger m) and more arrays (i.e., larger n) this probability can be decreased. In general, the probability of a false positive, $p_f(n, m, r)$ in a Bloom filter with n arrays of size m and r entered elements is

$$p_f(n,m,r) = \left(1 - \left(1 - \frac{1}{m}\right)^{nr}\right)^n \approx \left(1 - e^{-nr/m}\right)^n . \tag{1}$$

A more detailed derivation can be found in [7].

2) Signatures and Credentials: To use the Bloom filter data structure as credentials for packets that traverse the network, we store signatures from routers. During the connection setup, the source node of a connection can negotiate permission to transmit data across a router. When the router j, $\quad 1 \leq j \leq r$ permits transmission, then it provides the source with a signature s_j. A signature is the set of indices $s_j[i], 1 \leq i \leq n$ of bits that are set in all n Bloom filter arrays. The signatures from all routers along the path are then superimposed (i.e., logical OR operation) in the Bloom filter data structure. This creates credentials c (consisting of n arrays of size m) that are sent by the source node with each packet. This process of creating credentials is illustrated in Figure 3.

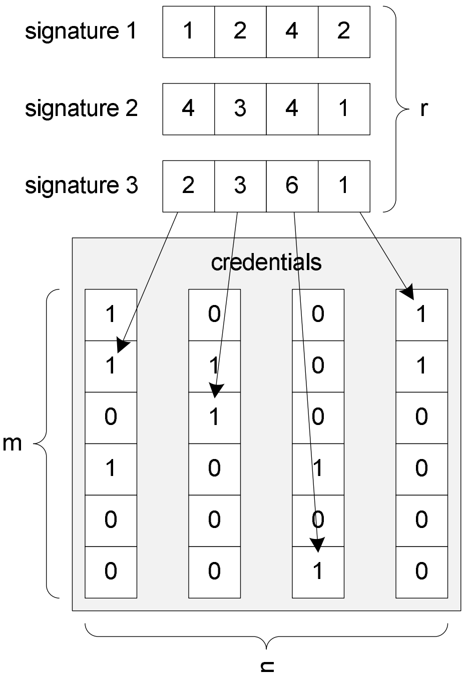


Fig. 3. Credentials Data Structure. This example shows three signatures that generated a set of 1's in the credentials data structure.

When receiving a packet, a router can then check if the credential c in the packet has all the bits that were provided as signature s_j to the source node. If the credentials are valid, then

$$c[i][s_j[i]] = 1, \quad 1 \leq i \leq n, 1 \leq j \leq r \tag{2}$$

If the credentials do not contain the signature of a router, then it is likely that one of the arrays in the credentials does not contain a 1 at one of the signature locations in the credentials and thus the validation of the credentials fail. This argument, of course, is of a probabilistic nature. A router may accept a packet that does not have correct credentials with the same probability as a false positive appears in the Bloom filter (see Equation 1). However, packets are only successfully delivered to a destination if *all* 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.

C. Credentials Security

The security of credentials depends on the quality of the signature. In order to make it difficult to create a fake signature s_j for a router j, it should be difficult to guess which bits will be set. We can achieve this by using cryptographically strong hash functions (e.g., MD5 [5] or SHA-1 [6]) where each router uses a set of n secret keys $k_j[i], 1 \leq i \leq n$. The cryptographic hash function $h_i(k_j[i], f)$ uses a key that matches that particular router j and the Bloom filter array i as secret information and flow identifier f as information that is specific to a particular flow. The use of a flow identifier f (e.g., based on a 5-tuple hash) helps to avoid attacks where signatures from an authenticated connection are used for a different connection. This process ensures that:

- Credentials for different flows are different (even if they traverse the same set of routers) because the use of f as parameter in the hash function will create different signatures.
- Credentials for flows that traverse different routers are different, because a different set of signatures are superimposed in the credentials. Signatures differ because each router has a different set of secret keys k_j.
- Credentials are difficult to fake since the result of the cryptographic hash function h_i cannot be guessed without availability of keys k_j.
- Credentials can be checked easily by performing n lookups in credentials c and checking if Equation 2 holds. Note that this requires that each router remembers the signature s_j that matches with a particular flow. This can be done by maintaining a flow cache. If the signature for a flow cannot be found in this cache, the signature can be recalculated (using k_j and f) at a higher computational cost.
- Credentials are of small size since all signatures s_j can be superimposed into a fixed-size Bloom filter data structure.
- Credentials cannot be "reversed" to obtain hash keys used by any of the routers or to create fake credentials for different flows.

It is possible that a malicious node injects traffic that uses the same flow identifier and credentials from another packet. In this scenario, all credentials checks will be valid. This, however, is only possible if the attacker injects the traffic along the path for which the credentials were generated in the first place. If the attacker is even a single hop away from that path, this traffic will likely be rejected.

D. Density Limit

One important observation regarding credentials as described above is that there exists a very simple attack to circumvent a credentials check: an attacker could set all bits in the credentials to 1. Such credentials would always satisfy Equation 2, no matter what secret keys or flow identifiers are used. This is clearly an undesirable property.

In order to address this issue, we introduce one additional concept to our Bloom filter. We define a "density" metric $d(c)$ that reflects the number of 1's in credentials c as a fraction of the total size:

$$d(c) = \frac{|\{c[i][j] = 1\}|}{nm}, \quad 1 \leq i \leq n, 1 \leq j \leq m. \tag{3}$$

To consider credentials as valid, we require that the density is equal or below a certain threshold: $d(c) \leq t_d$. If the density is higher, we assume the credentials to be invalid and thus reject the packet. If the threshold is chosen to be too low, even valid credentials may be rejected. The worst cast assumption of the number of 1's in valid credentials is a function of the number of signatures in the credentials. Assuming all signatures from r routers are placed in the credentials and do not overlap, then r bits in each array are set to 1. Thus,

$$d(c) \leq \frac{r}{m}, \quad r \leq m \tag{4}$$

holds true for all credentials.

E. Group Credentials

In some cases, it may be too complex to negotiate permissions with every single router along a path. It is possible to use credentials also for scenarios where a signature provides access to a group of routers (e.g., all routers within an autonomous system). In this case, all routers in the group share the same secret keys (and hash function). A signature that was issued by one router in the group sets the correct bits in the credentials to ensure that all other routers in the group will let the packet pass. If the end-system is not aware of the grouping of routers, it will negotiate a signature with each one. But since the signature from all routers in the group is the same, the credentials will have the same bits set.

Group credentials are particularly useful when it can be expected that routers with in a group change during the lifetime of a connection (e.g., routing changes due to OSPF updates within an autonomous system, or connectivity changes within a MANET). For the estimation of the density of the credentials (see Equation 4), the group of routers can be considered a single system. Thus the size of the arrays in the Bloom filter is a function of the number of groups traversed (rather than the number of hops traversed) if group credentials are used.

V. Security Analysis

A. Probability of False Positive Transmission

The main goal of the presented data path architecture is to identify valid traffic and thus not allow the transmission of attack traffic. Since the Bloom filter data structure used for credentials can yield false positives, it is possible that traffic with randomly forged credentials may pass through the network. To determine this probability for a given configuration of credentials size n, number of signatures r, and signature size m, we first determine the probabilities for 0's and 1's in the Bloom filter data structure.

The probability that a bit is not set by a single hash function depends on the size (i.e., n) of the Bloom filter data structure:

$$P[\text{bit not set by single hash function}] = 1 - \frac{1}{n}. \tag{5}$$

When using m hash functions in one signature, the probability that a bit is set by none of these hash functions is:

$$P[\text{bit not set by single signature}] = \left(1 - \frac{1}{n}\right)^{m}. \quad (6)$$

Note that for this analysis we assume that hash functions yield independent and uniformly distributed hashes. With r signatures combined to form credentials, a bit in c is not set with the following probability:

$$P[\text{bit not set by } r \text{ signatures}] = \left(1 - \frac{1}{n}\right)^{mr}. \quad (7)$$

Accordingly, the probability that a bit is set to 1 in a credentials data structure is:

$$P[\text{bit set by } r \text{ signatures}] = 1 - \left(1 - \frac{1}{n}\right)^{mr}. \quad (8)$$

In order to be considered valid, forged credentials need to have a bit set at all indices of the router- and flow-specific hash functions. Thus, the probability of encountering 1's at these m locations is:

$$P[\text{false positive on single router}] = \left(1 - \left(1 - \frac{1}{n}\right)^{mr}\right)^{m}. \quad (9)$$

Passing a single router without being identified as an invalid packet, is only practically useful if this is the only router along the path to the destination. However, in most network configurations, multiple routers have to be traversed. To reach a destination without valid credentials while traversing r hops (each adding one of r signatures to c), a false positive needs to occur r times in a row. The probability for this event is:

$$P[h \text{ consecutive false positives}] = \left(1 - \left(1 - \frac{1}{n}\right)^{mr}\right)^{mr}. \quad (10)$$

B. Multicast Scenario

When using credentials in a multicast environment, the source needs to signatures from all routers along all paths to all destinations in the credentials. To estimate the performance of Bloom filter credentials for this scenario, we assume that multicast is performed along a binary tree where each node corresponds to a router that duplicates the packet and sends it to two more nodes. Assuming a balanced tree, the height of the tree h relates to the number of destinations d as follows:

$$2^{h-1} < d \leq 2^{h} \quad \text{or} \quad h = \lceil \log_2 d \rceil. \quad (11)$$

For simplicity, we assume a complete binary tree with $d = 2^h$ destinations. The number of internal nodes in such a tree corresponds to the number of routers r_m that are encountered when multicasting:

$$r_m = 2^h - 1 \quad \text{or} \quad r_m = 2^{\log_2 d} - 1 = d - 1. \quad (12)$$

Thus, $2^h - 1$ signatures have to be superimposed in the Bloom filter and the resulting probability of a false positive on a single router is:

$$P[\text{false positive on single router}] = \left(1 - \left(1 - \frac{1}{n}\right)^{m(2^h-1)}\right)^{m}. \quad (13)$$

Assuming attack traffic needs to traverse the same number of hops as multicast traffic (i.e., h, the probability for a false positive end-to-end transmission is:

$$P[\text{multicast end-to-end f. p.}] = \left(1 - \left(1 - \frac{1}{n}\right)^{m(2^h-1)}\right)^{mh}. \quad (14)$$

C. Network Coding Scenario

When using network coding, then packets from multiple sources may be coded together. The extreme case would be such that there are an equal number of sources and destinations and every source sends to every destination. If we allow that any packet may be coded with any other packet on any router, then a coding step may happen on each step of the path. When coding packets, we assume that credentials are superimposed and thus, h credentials may be superimposed by the time a packet reaches its destination. (Note: For simplicity, we assume that coding is done only across two packets at any node.) Thus, the false positive rate increased to:

$$P[\text{network coding end-to-end f. p.}] = \left(1 - \left(1 - \frac{1}{n}\right)^{m(2^h-1)h}\right)^{mh}. \quad (15)$$

D. Results

Figure 4 shows a comparison of the false positive rate for different credentials configurations and usage scenarios. The size of the credentials data structure is compared for $n = 32$ bits and $n = 256$ bits. The number of hash functions is $m = 8$. The x-axis shows the length of the path r (for unicast) and h (for multicast and network coding). The upper x-axis shows the number of destination nodes for the corresponding tree height. The y-axis shows the end-to-end false positive rate, i.e., the probability that a packet with randomly forget credentials can traverse r or h hops while not being detected as invalid by any router.

The results show that for unicast, even a small credentials data structure of 32 bits provides very good detection (below 10^{-6} for $r = 3$). This probability increases for fewer hops (because there are fewer checks along the path) and for more hops (because the Bloom filter fills up). When using a larger Bloom filter of 256 bits, the probability of undetected transmission is well below 10^{-12}. This size credentials performs also very well for multicast (10^{-8} for 32 nodes) and small network coded setups. It is important to remember that these results present the absolute worst case of network coding and multicast. In a practical deployment, fewer signatures will be superimposed and thus the false positive rate is much lower. Also, larger Bloom filters can provide equal performance gains as observed between 32 bits and 256 bits in Figure 4.

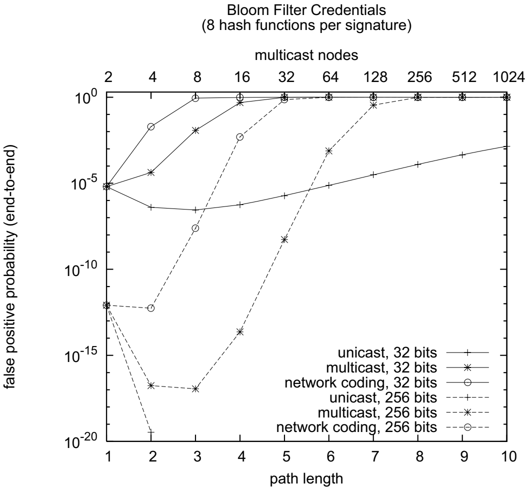


Fig. 4. Comparison of Credentials Performance under Unicast, Multicast, and Network Coding.

VI. Summary and Conclusions

In this paper, we have presented an architecture for data path credentials that allow networks to closely control traffic. We have shown that credentials that are based on Bloom filter data structures can be efficiently implement such an architecture and provide probabilistic guarantees on only permitting valid to traverse a network.

Clearly, there are several open questions that need to be answered before this design can find its way into next-generation networks. In particular, the answering the following questions are the next tasks in our future work:

- What is the optimum configuration of number of arrays (n) and size of arrays (m) in credentials for realistic networks? While larger values for m and n reduce false positives and thus provide more security, they also lead to larger space requirements in packet headers.
- What is a good design for the protocol that establishes credentials? Should the end-system directly connect to each router and request a signature, or should a centralized node in each group of routers (i.e., administrative subnet) handle connection requests? How can the credential-issuing subsystem be protected from denial-of-service attacks (as pointed out in [9] in the context of capabilities-based networking).
- Where in the packet header should credentials be placed? This question is particularly important when considering incremental deployment on an existing network.
- How does the setup and use of credentials change in wireless ad-hoc networks?

Designing networks that can provide provable guarantees on information assurance is important. The concept of credentials as presented in this paper addresses data path security issues and contributes an important step towards global assurable networking.

References

[1] B. H. Bloom, "Space/time trade-offs in hash coding with allowable errors," *Communications of the ACM*, vol. 13, no. 7, pp. 422–426, July 1970.

[2] A. Broder and M. Mitzenmacher, "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.

[3] S. Dharmapurikar, P. Krishnamurthy, and D. E. Taylor, "Longest prefix matching using Bloom filters," *IEEE/ACM Transactions on Networking*, vol. 14, no. 2, pp. 397–409, Apr. 2006.

[4] S. Dharmapurikar, P. Krishnamurthy, T. S. Sproull, and J. W. Lockwood, "Deep packet inspection using parallel Bloom filters," *IEEE Micro*, vol. 24, no. 1, pp. 52–61, Jan. 2004.

[5] R. L. Rivest, "The MD5 message digest algorithm," Network Working Group, RFC 1321, Apr. 1992.

[6] D. E. Eastlake and P. E. Jones, "US secure hash algorithm 1 (SHA1)," Network Working Group, RFC 3174, Sept. 2001.

[7] P. S. Almeida, C. Baquero, N. Preguiça, and D. Hutchison, "Scalable bloom filters," *Information Processing Letters*, vol. 101, no. 6, pp. 255–261, Mar. 2007.

[8] R. Ramaswamy, N. Weng, and T. Wolf, "Analysis of network processing workloads," in *Proc. of IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS)*, Austin, TX, Mar. 2005, pp. 226–235.

[9] K. Argyraki and D. Cheriton, "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.