Securing the Internet Routing Infrastructure

Panagiotis Papadimitratos and Zygmunt J. Haas, Cornell University

ABSTRACT

The unprecedented growth of the Internet over the last years, and the expectation of an even faster increase in the numbers of users and networked systems, resulted in the Internet assuming its position as a mass communication medium. At the same time, the emergence of an increasingly large number of application areas and the evolution of the networking technology suggest that in the near future the Internet may become the single integrated communication infrastructure. However, as the dependence on the networking infrastructure grows, its security becomes a major concern, in light of the increased attempt to compromise the infrastructure. In particular, the routing operation is a highly visible target that must be shielded against a wide range of attacks. The injection of false routing information can easily degrade network performance, or even cause denial of service for a large number of hosts and networks over a long period of time. Different approaches have been proposed to secure the routing protocols, with a variety of countermeasures, which, nonetheless, have not eradicated the vulnerability of the routing infrastructure. In this article, we survey the up-to-date secure routing schemes that appeared over the last few years. Our critical point of view and thorough review of the literature are an attempt to identify directions for future research on an indeed difficult and still largely open problem.

INTRODUCTION

Nowadays the Internet has hardly any resemblance to the early research-oriented network that was designed to operate within a single domain of trust and with practically no security mechanisms in place. Through a 20-year process, the network and its technological feats have matured, but so have the threats to its proper operation. The dramatic expansion and commercialization of the Internet rendered the network widely accessible, blurred its boundaries, and opened it up to a wide range of attacks. More important, the protocols that are collectively identified as the TCP/IP protocol suite were not designed with such a hostile environment in mind. The exploitation of their features and weaknesses, which came as no surprise, led to the vast majority of the reported security incidents.

The resultant growing security awareness of the Internet community fueled an ongoing effort to make the network a safer place for exchanging information and conducting business. In principle, the intended or already present security services fall into two categories: *data security*, which encompasses confidentiality, integrity, and origin authentication; and *access control*, which protects networked resources — hosts, servers, and other networking devices such as bridges, gateways, and routers — from unauthorized use. Such services are supported by security protocols operating at different layers of the protocol stack, and provide, to a lesser or greater extent, protection of user data. This is true even if the medium is not secure by itself.

However, the problem of securing the networking infrastructure per se is in essence orthogonal to securing the actual transfer of information. An adversary could simply target the infrastructure, instead of launching an attack against the connection between any two securely communicating ends; in fact, the latter would be onerous if strong cryptography were appropriately used. For example, by interfering with the routing protocol, data may be redirected over paths controlled by the attacker, or, even worse, the use of incorrect connectivity information can result in failure to have data delivered to a large number of destinations. This way an attack against a single element of the routing infrastructure can cause a *denial of service* (DoS) for a large portion of the user population. The routing protocol is an excellent target, not only because severe network outages can be caused, but also because this may not require a significant effort in today's absence of any form of defense: "*Abuse of the routing mechanisms and protocols is probably the simplest protocol-based attack available*" [1].

The single most important vulnerability stems

This work has been sponsored in part by the NSF grant number ANI-9980521 and the ONR contract number N00014-00-1-0564.

0163-6804/02/$17.00 © 2002 IEEE

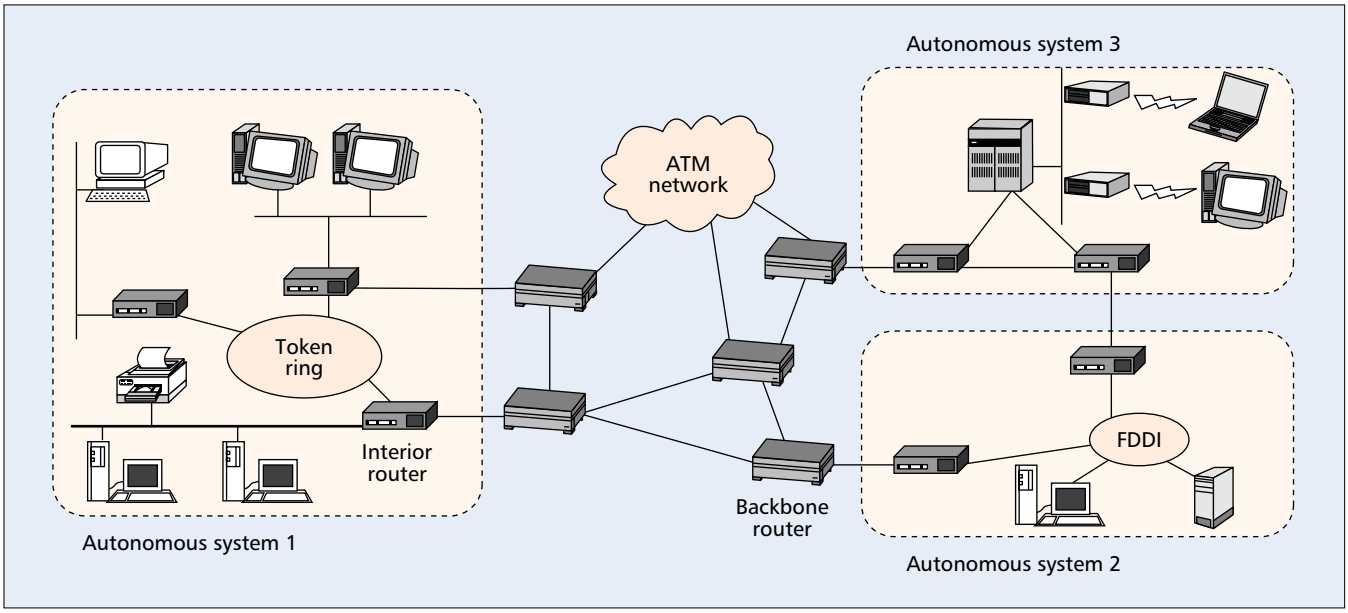

Figure 1. *Internet organization: the network is divided into interconnected autonomous systems, owned and administered by different organizations.*

from the nature of the routing operation itself. Routers (i.e., network devices dedicated to supporting the routing functionality) have to acquire global knowledge of the network topology. They periodically exchange information with their neighbors and gradually update their view (the destinations they know how to reach). However, in most cases routers are practically unable to verify the correctness of the information they receive from their peers. As a result, injected false routing information would propagate throughout the network, and it might remain in use for arbitrarily long periods of time, thus deceiving more than one routers. This way, an attack at a single point of the network would allow the adversary to exercise its control over the network. The adversary could intercept data, or, if user traffic were protected, obstruct the data flows and even disable the network operation altogether.

This article addresses exactly this issue: how to protect the routing infrastructure — in other words, how to safeguard the topology discovery in the presence of adversaries — so that *authentic* and *noncorrupted* routing information can be acquired in a *timely* manner. First, we provide a concise overview of the Internet routing mechanism. Then a classification of the possible threats and attacks and an outline of the main directions to secure the routing protocol are presented. We review the solutions proposed in the literature to *prevent* the abuse of the routing protocol, categorized according to the type of underlying protocol. The schemes that take the alternate approach to *react* to an attack against the routing protocol, by first detecting and then responding to it, are reviewed next, before our discussion and conclusions.

Overview of Internet Routing

The Internet comprises a large number of interconnected heterogeneous networks owned or administered by different organizations. Examples of such management domains are the networked resources of a university or corporate campus, or an *Internet service provider* (*ISP*). Within and between domains, which are called *autonomous systems* (ASs), routers maintain an up-to-date view of the network state. They periodically exchange *direct* or *indirect* information: direct, when a router *advertises* the addresses of networks to which it is directly connected, or indirect, when a router first processes the information received from other routers before advertising all the networks it knows how to reach. In the former case, the router has more or less a view of the entire network topology, while in the latter case such knowledge is minimal. The goal for any routing protocol is to determine the "best" route to forward data to their destinations.

These two fundamentally different design approaches lend themselves to networking contexts with different characteristics. The multitude of deployed routing protocols builds on variations of either approach, and can be classified according to the organization of the Internet, shown schematically in Fig. 1. *Intra-AS* or *intradomain* or *interior gateway protocols* are implemented by routers within an AS, while *inter-AS* or *interdomain* or *exterior gateway protocols* run on routers that exchange routing information between domains and form the so-called Internet *backbone*. The interior routers maintain a consistent map of AS connectivity and provide a summary of the reachable destinations to the outside world. Inversely, the backbone provides information on reachable exterior destinations to the interior routing protocol. Their main difference is that intradomain protocols choose the route that minimizes some metric, such as distance, delay, or load, while interdomain protocols determine routes according to a *routing policy*. A routing policy can be an arbitrary set of rules that determine, usually with fairly complex criteria, which route advertise-

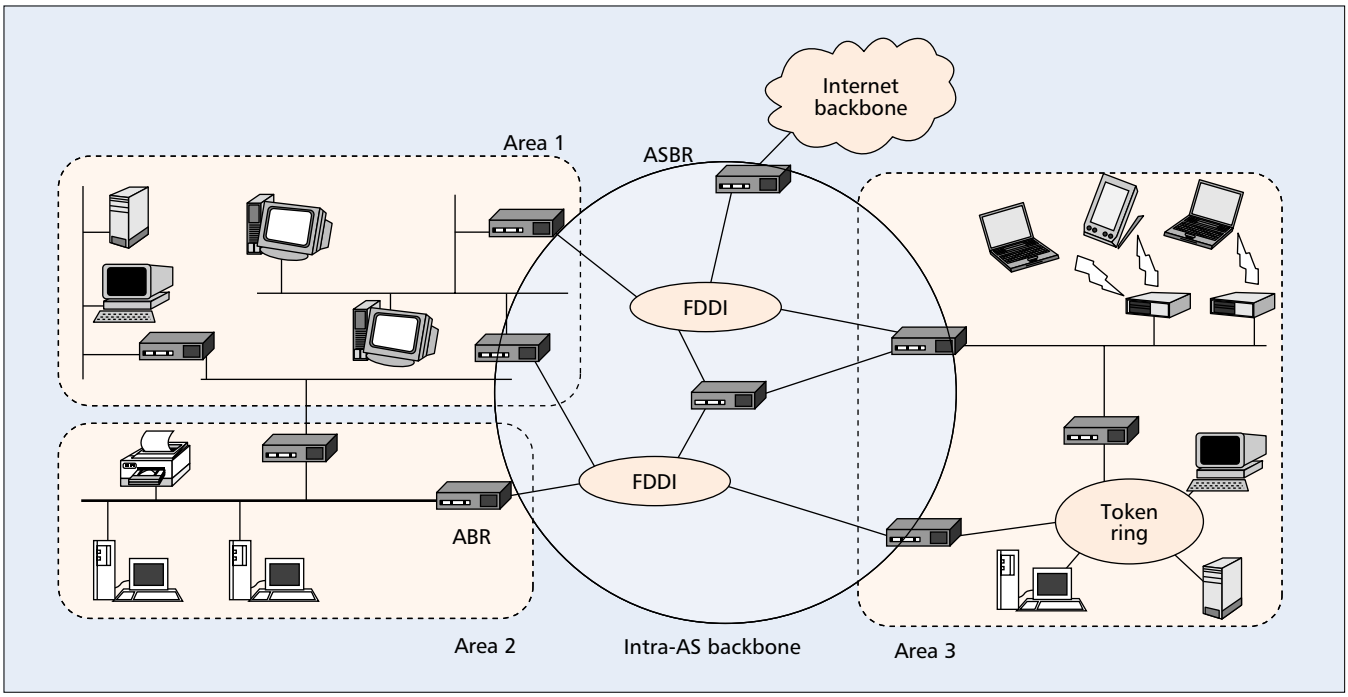

Figure 2. *Intradomain routing: ABRs connect areas to the AS backbone, and the ASBR connects the AS to the Internet.*

ments should be accepted as valid. For example, it is common to have routers configured manually with hundreds of routes, ASs, or destinations that are preferable to be included in a route or reported to certain neighbors, according to agreements among the interconnected organizations.

Among a large number of protocols that have been proposed and implemented, three protocols have been widely deployed and evolved to become the de facto standards for Internet routing: *Routing Information Protocol* (*RIP*), *Open Shortest Path First* (*OSPF*), and *Border Gateway Protocol* (*BGP*).[1] In order to facilitate the discussion in the rest of this article, we will briefly overview the operation of each of these protocols. RIP and OSPF are intradomain protocols, and BGP is an interdomain protocol. Although some security mechanisms are protocol-independent, and security enhancements have been proposed for other protocols as well, these three have indeed received the most attention. Additionally, they are representative of different types of routing algorithms, as briefly reviewed below. The presentation of the security mechanisms in a later section will follow the classification according to the algorithm type as well.

RIP is a simple and widely deployed *distance vector* protocol appropriate for small networks with relatively simple topology. Routers do not have explicit knowledge of the network topology, but simply exchange their view of the network condensed in the form of the distance vector. This vector of the minimum distances to all the networks that a router knows how to reach is periodically passed to its neighbors. As routers receive their neighbors' vectors, they update their own vectors. The used metric is the *hop count*, and a network directly connected to the router is at a one-hop distance, while an unreachable network is 16 hops away, since the maximum distance for RIP is 15 hops.

The basic idea of OSPF, a *link-state* routing protocol, is that routers first discover their neighbors and the state of their incident links, and then communicate this information in *link state advertisement* (*LSA*) messages. The LSAs are *reliably flooded* throughout the network at fixed time intervals, with the most recent LSA per originating router relayed over each link exactly once. Eventually, all routers converge to the same complete map of the network topology and calculate the shortest paths to all other routers, using metrics more complex than the hop count. The freshness of an LSA is determined by its *sequence number* and an *age field*. As routers relay LSAs, they modify the age field, so the removal of aging links is synchronized throughout the network. A domain, as shown in Fig. 2, is organized into *areas*, and the flooding of LSAs does not cross area boundaries. A set of *area border routers* (ABRs), which form an intra-AS backbone, interconnect the areas, with an *AS boundary router* (ASBR) connecting the AS to the Internet backbone.

BGP is a *path vector* protocol; that is, the routing information on reachable destinations includes the corresponding path information. The full sequence of the networks between the source and destination allows BGP to detect loops in a simple and effective manner: if an AS is listed twice, a loop has occurred. The *update* messages advertise new routes, or inform about previously advertised routes that are not usable anymore. As routers relay the updates, they prepend their own AS identifier (*AS numbers*) before passing the advertisement to their neighbors. In addition, BGP allows the ranking of routes according to preferences, including a metric-based choice, although this is not implemented in most cases. The plausibility of such sets of rules and preferences (i.e., routing policies) relies on the minimal expectation that routers advertise routes they themselves use.

[1] *The detailed specification of the protocols is provided by IETF RFCs 2453, 2328, and 1771, respectively.*

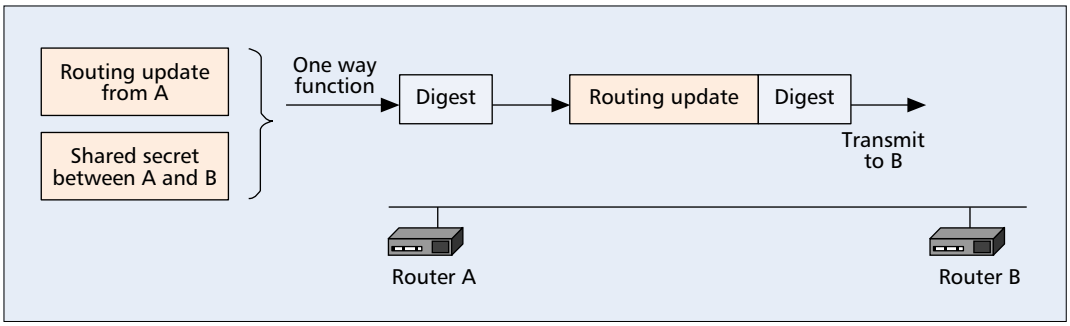

Figure 3. *Authentication of routing information exchanged between two adjacent routers.*

THREATS AND ROADMAP

Internet routing protocols may partially resist faults and detect undesirable conditions, such as "flapping" routes or links (i.e., routing information that changes at an unexpectedly high rate). In some cases, the protocol may also be capable of resuming normal operation after the removal of a faulty router. But such features do not render routing protocols robust against malicious attacks. Threats to the routing infrastructure may emerge from practically any part of the network; *external* attacks, mounted by a node that does not participate in the routing process, and *internal* attacks originating from a device "trusted" to support the routing functionality.

An internal attacker, which can be a compromised or faulty router, may not comply with the protocol by advertising false routing information and making arbitrary routing decisions. It can also tamper with the information originating from other routers, by modifying, replaying, or simply discarding it. As a result, a misbehaving router can obstruct the routing operation throughout a large part of the network.

An external attacker can acquire the topology knowledge by eavesdropping and actively falsify the routing information and the knowledge of the network state by erasing, meaningfully altering, injecting, and replaying control and routing traffic. It may also masquerade as a router and try to obstruct the traffic flow by generating floods of spurious messages or overloading ("jamming") a certain link or interface.

The impact of these attacks depends on a multitude of factors, such as the employed routing protocol, the point of attack, and the presence of security countermeasures. Protocols that pass aggregate routing information are inherently less robust against false advertisements than protocols that converge to a complete network connectivity map. For the case of distance vector protocols, it will not be long before all routing tables are updated, so all traffic is sent through the ill-behaved router that simply advertises the "best" path to all destinations. However, a misbehaving link state router can lie easily only about the victim's or its own incident links. Even if this happens, a legitimate router will have the chance to react to the false information by flooding a fresher correct update and have the false information flushed by the rest of its peers.

In practice, careful configuration can protect routers from a number of false advertisements. However, the requirement for "more administration and more manual configuration" results in a more complicated system, significantly less dynamic, and more prone to misconfiguration. More important, the routing infrastructure remains vulnerable to attacks that can bypass such defenses, when, for example, the adversary is capable of masquerading as a legitimate router or hijacking a connection and tampering with the exchanged routing information.

Cryptographic mechanisms can be a roadblock to such attacks by protecting the authenticity and integrity of the routing traffic and by providing the means to verify the authority of the participating routers. To achieve such goals, routers should be able to present their "credentials," so their peers validate the received information. The use of *public key cryptography* and digital signatures is the best way to do so, for different reasons, as will be explained below. Each router is assumed to have a unique identity and a pair of keys: a private one used to generate signatures and a public one used by the rest of the routers to validate them. As a result, the origin of routing information can be verified, message tampering can be detected, and the authorization of a node to participate in the routing process can be proven.

The cornerstone of such mechanisms is the presence of a *public key infrastructure* (*PKI*), a trusted facility that certifies and distributes the public keys of the routers throughout the network. The surveyed schemes assume an initial distribution of credentials, propose a simple protocol for the same task (e.g., each router floods its public key), or even define in detail the structure of the PKI. The use of public key cryptography, instead of the establishment of pairwise security associations, appears as an appropriate choice: the originator of a routing message may not know its possible recipients and thus be unable to select which shared keys to use. Additionally, it may be required that routing information cannot be repudiated, in order to facilitate the detection of misbehaving routers. Note that the mere disclosure of the routing information does not inflict harm on network operation, and the use of encryption is not proposed in most cases.[2]

The use of symmetric key cryptography would be useful to protect traffic exchanged between neighbors, as shown in Fig. 3, by *mes-*

Threats to the routing infrastructure may emerge from practically any part of the network; external attacks, mounted by a node that does not participate in the routing process, and internal attacks, originating from a device "trusted" to support the routing functionality.

[2] *The encryption of the BGP traffic is proposed in [2], although authenticity and integrity are the scheme's primary goals.*

The correctness of the routing operation is achieved under the assumption that a rogue router will not lie about its incident links. Nevertheless, in practice, a misbehaving router might attract traffic if it systematically corrupted the distance metrics, including its incident link lengths.

sage authentication codes (MACs) [3]. Such an approach entails more complex key management and cannot provide nonrepudiation, but it is far less computationally expensive than the public key digital signatures. It can provide authentication and integrity protection on a link basis, and complement naturally the PKI-based mechanisms. All the schemes discussed below assume, unless otherwise stated, the presence of an authorization structure and a PKI so that the authority of a router to advertise specific information and its role are determined. Unless explained otherwise, for the rest of the discussion, a certificate binds the public key of each router to its identity, and all routers are equipped with the certificates of their peers.

Preventive Security Mechanisms

In this section we survey the security enhancements that have been proposed in the literature. Before proceeding, we should note that the implementations of all current routing protocols, including RIP, OSPF, and BGP, support clear text password authentication. However, clear text passwords can easily be captured by an adversary, despite the additional assurances provided by specialized password authentication systems. The use of keyed hash functions and MACs is an apparent improvement, which has gained increased support by router vendors as an important line of defense to protect the routing traffic exchanged between adjacent routers. Both RIP and OSPF have been extended to support keyed MD5 authentication, and BGP updates can be protected by a password-based MD5 digest TCP option. However, these measures cannot stop rogue advertisements once they are injected in the network, and they do not allow the tracking of the incorrect routing information source. These additional assurances are the goal of some of the schemes we survey below.

Distance Vector Protocols

The use of digitally signed updates and significant modifications to the distance vector functionality have been proposed [4]. Routers pass the address of the *predecessor* network (i.e., the last hop before the advertised destination) and also inform their neighbors of changes in their incident links. Updates include the identity of the originating router and a sequence number so that their authenticity and freshness can be validated. To validate an update, the protocol steps backward through the recorded predecessors for all destinations, starting from the destination and the reported predecessor. If this succession of links leads back to the router that now checks the update, the advertisement is accepted as correct. However, the update signature does not cover the distance metrics, and virtually any distance other than hop count can be advertised by a misbehaving router. The correctness of the routing operation is achieved under the assumption that a rogue router will not lie about its incident links. Nevertheless, in practice, a misbehaving router might attract traffic if it systematically corrupted the distance metrics, including its incident link lengths.

Link State Protocols

The use of digital signatures differs significantly when the targeted protocol is a link state one. The originating router signs the LSA, which is validated by its peers before they relay it. According to the *robust flooding* technique [5], the sequence numbers of valid updates are compared to the ones stored locally. If the newly received number is equal to or larger than the one corresponding to the origin of the LSA, the router updates its stored value and forwards the advertisement. Otherwise, it transmits its own up-to-date value to the neighbor that relayed the obsolete advertisement. Ties, which could occur if, for example, the originating router crashed and lost its state, are broken by the numerical values of the signatures. All control traffic is acknowledged on a link basis, by copying the signature and the sequence number of the LSA. Although corrective actions such as retransmissions are taken in case of failures, this type of acknowledgment does not provide any guarantee as to which node responded. However, as long as a single route, free of subverted routers, exists for each pair of routers, the delivery of genuine LSAs will be possible.

The securing of OSPF takes a similar approach; the signature of the originating router remains attached to the LSA as it propagates throughout the network, in order to authenticate the source of the LSA, protecting the provided information and thus its timeliness [6]. The scheme proposes five new types of LSAs with modified headers carrying the signature, plus a sixth message type for distribution of the public keys throughout the AS. Its scope does not exceed the limits of the area and the AS, since misbehaving ASBRs or ABRs can still inject incorrect routing information. Another way to disrupt the protocol would be to tamper with the age field of the LSAs, which is not covered by the signature: an adversary could gradually increment the age of legitimate updates and force correct link state to be flushed as obsolete.

The large numbers of signed LSAs that need to be validated and generated is an important limiting factor for securing link state protocols. The computational overhead depends on the network size and topology, but the situation could become very difficult when handling the announcement of exterior routes within the AS. It is thus straightforward to attempt to provide a low-cost signature mechanism. The idea in [7] is to have each router commit to a *hash chain*, a long precomputed sequence of values that the router provably generated. The hash chain is calculated by the successive operation of a *hash function* [8, 9] on a random value. Then the router uses one chain element, indexed on the time of the update, per generated link state update so that receiving routers verify its validity in a computationally inexpensive manner. As illustrated in Fig. 4, they apply the same hash function on the newly received update and the current timestamp, and then compare against the initial commitment or previously validated element.

The scheme assumes that link states are

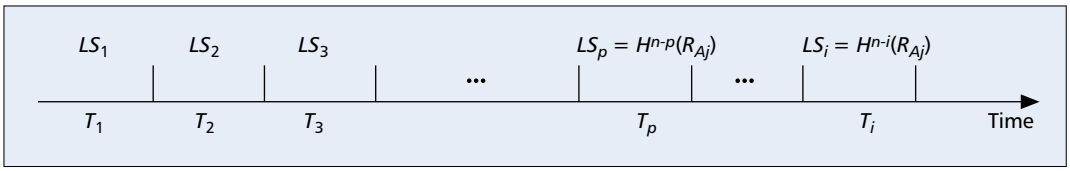

Figure 4. *Use of the hash chain to authenticate link state updates. Router A selects a random value* $R_{A,j}$ *for its* j*th incident link, and calculates the hash chain* $H^n(R_{A,j}) = H(H^{n-1}(R_{A,j}))$*, with* $H^0(R_{A,j}) = R_{A,j}$*. Its peers initially receive* $H^n(R_{A,j})$ *digitally signed by A. At time interval* T_i*,* $H^{n-i}(R_{A,j})$ *is released by router A to authenticate the advertised state of link* (A, j)*. Each receiving router has some previously validated link state* LS_p*, that is,* $H^p(R_{A,j})$ *from interval* T_p*. If a change is reported (e.g., at* T_p *the link was "down" and now, at* T_i *it is "up"), the link state is deemed authentic if* $H^i(LS_i)$ *is equal to* $H^n(R_{A,j})$*. If no change is reported,* LS_p *is compared to* $H^{i-p}(LS_i)$*.*

advertised only periodically, and relies on loosely synchronized clocks. In particular, if the maximum clock drift does not exceed two advertisement periods, the adversary cannot cause a forged message to be accepted as genuine and fresh. However, the advantage of link state protocols to use complex routing metrics would be abolished, since the scheme calculates one hash chain per up or down state for each incident link. The use of a multivalued metric requires the generation of one chain per value, thus imposing significant overhead. Most of these chain elements would remain unused, while frequent link state changes or short update periods would require frequent generation and commitment to new chains. Under these conditions, the scheme may not be plausible and, as the authors suggest, an asymmetric cryptography scheme might be more efficient.

A different approach to reduce the cost of authenticating LSAs proposes to use the elements of a hash chain not as authenticators but as keys for the generation of MACs attached to each LSA [10]. The reduction of cost is achieved by deferring the validation of LSAs until reception of the corresponding key, which is flooded by the originating router at designated times. The protocol assumes synchronized clocks and bounded maximum transmission delay over any network path. All LSAs have to be sent within a sufficiently short period before the release of the key with which they were signed. Then, no forged advertisement will be accepted during the *a posteriori* validation. However, the longer the key release period, the higher the vulnerability of the protocol. LSAs update the network map, although they cannot be immediately validated and may be forged. As a result, the use of such false routing information can degrade the protocol operation. In essence, the scheme provides low-cost authentication at the expense of restrictive timing assumptions and low assurances, since it assumes that attacks against the protocol will be infrequent and countered by disconnection of the misbehaving router.

A solution to the vulnerability of the two above-mentioned schemes to "delay and forge" attacks is given by a protocol that uses *one-time signatures* to sign routing updates [11]. The proposed protocol removes the correlation between successive keys, by "enclosing" the public key for the validation of the $(i + 1)$st message signature inside the ith message. A router chooses at random one-time secret and public key components, and initially digitally signs and distributes the hash value of the public components. Then each signed message carries the hashed value of the public key components for the following message, and it is validated according to the public key carried by the previous message. However, a message loss implies that the commitment to the initial one-time public key has to be redone with conventional public key cryptography. As a result, the protocol would retain its efficiency only when all messages are received. Additionally, the relatively large size of the signature and the resultant transmission overhead might be a limiting factor, especially because of the small size of LSA packets.

Path Vector Protocols

The securing of BGP and path vector protocols requires the protection of the path to the destination along with the advertised information. It has been proposed to secure BGP by having the advertising router identify its immediately preceding AS, as an additional attribute covered by the signature. This technique is similar to the one discussed earlier: routers maintain a predecessor database with all the validated predecessor links, and verify the authenticity of the path by stepping back from the destination and its predecessor, through the database, until the entire path is either validated of rejected [2]. However, it is restrictive to assume that an AS on the path would also be a destination; for example, it is possible that an AS is transit-only for specific endpoints. Then a valid path could be rejected because of a missing adjacency, or data would not be delivered for a certain source and destination although the path was deemed valid. Such problems occur because of the complicated BGP routing policies that appear to be the usual case, instead of a shortest-path policy. Additionally, the modification of the route of an in-transit update is possible, since the digital signature protects only the predecessor and the destination.

This can be avoided if each router passes the advertisement only after it appends its signature covering the entire path [12]. As advertisements propagate further away from their origin, nested signatures are accumulated, with each signature covering the previous one and the destination. This is necessary so that no router can detach a part of the route and the associated signatures and pass them as proof of path authenticity.

In essence, the scheme provides low cost authentication at the expense of restrictive timing assumptions and low assurances, since it assumes that attacks against the protocol will be infrequent and will be countered by the disconnection of the misbehaved router.

In order to provide an in-depth defense, security measures should be complemented by tools that could detect an attack while it is still underway, and possibly identify its source, so the misbehaving devices are isolated. This approach could be valuable for the protection of the routing infrastructure as well.

However, a router can insert itself into the path, when it propagates an advertisement and the corresponding path, without being authorized to do so.

The issue of determining the authority of routers to advertise network prefixes or routes is addressed by a scheme based on two PKIs. One certifies the ownership of the address space granted to each organization, and the second certifies the identities of the ASs and their BGP routers [13]. The authorization of an AS to advertise a block of addresses can be verified since a certificate binds address blocks to specific public keys. Moreover, the authentication of BGP routers, ASs, and the relationship between routers and ASs is possible as well. Routers can verify the ownership of an AS number by an organization, the identity of an AS, and the identity of a BGP router. With this hierarchy of certificates at hand, signatures, called *address attestations* and *route attestations*, are applied on each update message in a manner similar to the above-mentioned nested signatures. Each route attestation, added by one BGP router along the route, covers the organization identifier and the entire subpath, from the predecessor of the router that makes the attestation to the advertised destination.

Reactive Security Mechanisms

Despite the use of preventive security mechanisms, it is always possible that attackers defeat or bypass the security countermeasures, or simply target one of the unprotected components of the system. In order to provide an in-depth defense, security measures should be complemented by tools that could detect an attack while it is still underway, and possibly identify its source, so the misbehaving devices are isolated. This approach could be valuable for the protection of the routing infrastructure as well. The schemes surveyed in an earlier section defend the infrastructure against external attacks, but they leave ample space for internal attacks.

In order to be able to identify an attack, the behavior of the attacker (i.e., the rogue router) has to be noticeably different from that of a well-behaving router. This is the basis of *intrusion detection systems* (*IDSs*), which have been widely applied for the protection of computing systems and networked environments by detecting anomalies or misuse patterns.[3] In order to transcribe such techniques onto the routing context, a precise definition of what constitutes normal behavior for a router is needed, especially in terms of route discovery. Or inversely, it is important to define what an IDS-inspired system should look for before producing a misbehavior alarm.

A network IDS that aims to protect the OSPF routing protocol proposes the use of a combination of countermeasures in order to detect attacks [16]. The system tries to characterize the current network from sequences of protocol-related events it tries to correlate with known attacks. Each attack is specified by a finite state machine, and an alarm is triggered by a specific sequence of transitions caused by events related to the content of the received LSAs or other control traffic. In addition, a set of static rules used to discard erroneous control traffic and a statistical inference tool complement the system. These rules are used to increase the chances of intrusion detection. The analysis of the time-related behavior of the routing protocol, the close monitoring of the routers, and the detailed specification of known attacks are the primary weapons of this approach, whose effectiveness, especially in the case of OSPF, is aided by the fact that a successful attack should be persistent and thus is easier to identify.

A different approach uses the data forwarding operation as a proxy for identifying whether a router is faulty or compromised [17, 18]. The basic idea is that nonfaulty routers execute a protocol to test their peers and, in essence, verify whether each packet routed over the tested router is appropriately forwarded. Then the router is deemed nonfaulty. The testing router *probes* its neighbors by sending packets addressed to one of its own ports. Under the assumption that the tested router will see the same point-to-point link as the least cost path back to the testing router, a returned packet implies that the test was successful. Of course, it is also required that the tested router be unable to detect the probe packet (e.g., determine that the destination IP address corresponds to one of the interfaces of the source router). Otherwise, it could selectively respond to probes and misbehave with the rest of data and control traffic.

A more elaborate scheme proposed by the same authors relies on traffic analysis and the principle of conservation of traffic. It is assumed that the number of packets arriving at a router equals the sum of outbound traffic plus the number of packets destined for the router, or a network directly connected to the router. A router that violates this condition is considered misbehaving. To detect such misbehavior, each router continuously updates *traffic counters* for the three categories of packets per neighbor and direction of communication (incoming/outgoing): in-transit packets, packets destined to the router/neighbor, and packets originating from the neighbor/router. Moreover, routers are assumed capable of counting the number of packets their neighbors misrouted (i.e., routed over a suboptimal path). At any instance, a router can initiate the protocol by flooding a request, and wait until at least the majority of the routers respond with a request. Then, the contents of the traffic counters are exchanged among the routers within their three-hop neighborhood, and the flow conservation tests are performed. Nevertheless, these protocols may be of limited practicality, relying on very strong assumptions that are almost impossible to satisfy in practical networks, even if a link state protocol is used. For example, counting misrouted packets requires that a testing router have copies of the routing tables of all its neighbors, a well behaved router never drops packets, all routers have knowledge of exactly the same network state, and each pair of well-behaved routers is connected.

[3] *An intrusion detection system was first introduced in [14]. For an overview, see [15].*

Discussion and Conclusions

The solution space of an already difficult problem — how to secure the Internet routing infrastructure — is further constrained by the wide deployment of diverse protocols and systems. Security measures will have to be retrofitted to existing protocols, as becomes clear from many of the surveyed proposals. However, basic protocol mechanisms, or their absence, may disallow the provision of a security service; for example, BGP does not use sequence numbers to identify updates; thus, the secure version of the protocol cannot rely on such a feature to provide replay protection [13]. At the same time, protocol features, such as the real-time advertisement of exterior routes to all interior OSPF routers, may be responsible for excessive computational cost, even under normal operation conditions. Both examples indicate that as protocols evolve, it would be reasonable to expect that modifications may be driven by security considerations as well.

Nevertheless, the operational requirements of the protocol appear to be the primary factor that will determine the plausibility of deploying effective security enhancements. Although Moore's Law ensures that cryptographic operations gradually become inexpensive, the increasing network size and connectivity, and the dynamically varying network state counterbalance such benefits. The rate of cryptographic operations primarily depends on the rate of topology changes and the topology itself. The number of routers, their connectivity, the placing of a router in the network, the size of the interconnected subnets, and the speed of links are all critical factors determining the performance overhead. In order to assess the feasibility of the proposed security measures, detailed protocol-specific studies are required, as was done, for example, with secure BGP [19].

An additional important factor, which has been underrated, is the cost of reconfiguration, especially as a means to deal with accidental situations such as a router crash. The rebooted device would have to be updated with the keys of its peers, and, especially for protocols that replicate functionality and processing, re-acquire knowledge of entire network state. For example, for OSPF this state will be the entire "link database"; that is, link metrics and ages, sequence numbers, and, for BGP, the complete routing tables of all other "speakers" — the backbone routers. Upon receipt, this volume of data has to be validated before the router resumes operation.

Abnormal conditions or denial of service attacks are yet another factor that has to be considered, especially in conjunction with the expected high operational cost of security measures. It is necessary to cope with seemingly legitimate control traffic that may arrive at overwhelmingly high rates, so the apparent benefits from cryptographically powered schemes are not waived under adverse situations. Otherwise, the infrastructure would become an ideal target for denial of service attacks, which transmit at intervals comparable to the processing delay of a single update validation. The solution to such vulnerability appears to be careful configuration of the protocol suites by disabling some of their features. In addition, tools that filter out spurious traffic, bound the rate of incoming control traffic, and limit the area affected by such attacks should be used. Nevertheless, the combined effect of such an approach can only be complementary, and it may have a premium on the dynamic nature of routing.

The dynamically changing state of the network can be a major roadblock to efforts that attempt to detect intrusions by observing the packet forwarding. Apart from the previously discussed impractical aspects (e.g., the assumption that packet loss is only due to malicious behavior), intrusion detection schemes may be of limited value when the routing protocol is abused. A router can be made to appear to misbehave because of intentionally nonoptimal routing information. This is possible even when routers do not have the same view of the topology, since packets are counted as misrouted when they deviate from the optimal path. Similarly, the testable links and routers can be incorrectly determined when false link costs are injected. However, under a set of realistic assumptions, the design of protocol-specific tools appears to be a desired extension of secure routing protocols, but only when it is combined with other security enhancements. Otherwise, an attacker capable of injecting false routing information can at least mislead the IDS algorithm, especially when repudiation is possible.

This naturally brings up the issue of handling a misbehavior alarm, especially when it may be false. Apart from the obvious solution of an overlooking central management entity, it is necessary to investigate the practicality of schemes that could reach a consensus about the detection of a faulty router. However, this may increase the delays of intrusion detection while the undergoing attack affects the network. In general, the cost of relying on IDS instead of preventive security mechanisms can be significant, and the claim that IDS could be an alternative to the, in some cases, prohibitively costly security enhancements should be revisited.

Moreover, the cost of employing a mechanism cannot be considered independent of the cost (impact) of a successful attack. Since attacks may become increasingly frequent, the cost of sustaining an attack until its detection will become more significant, depending on the ability of the detection mechanism to identify the ongoing attack. The design and evaluation of such mechanisms will be more challenging when attacks are not persistent, but almost random, and thus difficult to detect. Even though the resultant outages may not be severe, in the sense that not very high percentages of packets are lost, they may have a significant impact on the quality of service of some sensitive applications. In other words, an adversary may trade off effectiveness for a less visible attack.

As the necessity to secure the routing infrastructure grows, it becomes apparent that security enhancements should be globally present. To this extent, the Internet security architecture (*IPsec*) [20] can provide authentication, integrity, replay detection, and encryption to protect

Under a set of realistic assumptions, the design of protocol-specific tools appears to be a desired extension of secure routing protocols, but only when it is combined with other security enhancements.

As the necessity to secure the routing infrastructure grows, it becomes apparent that security enhancements should be globally present.

routing protocol traffic. The global deployment of IPsec protocols will provide a set of powerful tools that would seamlessly interoperate throughout the Internet, under the assumption that the next generation of IP, IPv6, is also globally present. However, the mere provision of such services cannot secure the routing protocol itself; IPsec can deter external attacks and disallow the injection of unauthorized routing traffic by securing the point-to-point exchange of routing updates at the network layer, but cannot enforce or guarantee correct operation of the routing protocol under an internal attack. The satisfaction of the security requirements for each of the deployed protocols is a task that will be undertaken by either schemes such as those surveyed in this article or their successors. These future protocols will have to provide a solution to the most basic, but still unsolved, problem of protecting against false routing information.

REFERENCES

[1] S. M. Bellovin, "Security Problems in the TCP/IP Protocol Suite," *Comp. Commun. Rev.*, vol. 19, no. 2, Apr. 1989, pp. 32–48.
[2] B. Smith and J. Garcia-Luna Aceves, "Securing the Border Gateway Routing Protocol," *Proc. Global Internet '96*, Nov. 1996.
[3] H. Krawczyk, M. Bellare, and R. Canetti, "HMAC: Keyed Hashing for Message Authentication," RFC 2104, Feb. 1997.
[4] B. R. Smith, S. Murphy, and J. J. Garcia-Luna-Aceves, "Securing Distance-Vector Routing Protocols," *Proc. Symp. Net. Distrib. Sys. Sec.*, Feb. 1997.
[5] R. Perlman, "Network Layer Protocols with Byzantine Robustness," Ph.D. dissertation, MIT/LCS/TR-429, MIT, Oct. 1988.
[6] S. Murphy and M. Badger, "Digital Signature Protection of the OSPF Routing Protocol," *Proc. Symp. Net. Distrib. Sys. Sec.*, Feb. 1996.
[7] R. Hauser, T. Przygenda, and G. Tsudik, "Lowering Security Overhead in Link State Routing," *Comp. Net.*, vol. 31, 1999, pp. 885–94.
[8] R. Rivest, "The MD5 Message-Digest Algorithm," RFC 1321, Apr. 1992.
[9] Secure Hash Standard, NIST, Fed. Info. Proc. Standards, Pub. 180, May 1993.
[10] S. Cheung, "An Efficient Message Authentication Scheme for Link State Routing," *Proc. 13th Annual Comp. Sec. App. Conf.*, Dec. 1997.
[11] K. Zhang, "Efficient Protocols for Signing Routing Messages," *Proc. Symp. Net. Distrib. Sys. Sec.*, 1997.
[12] S. Murphy *et al.*, "Retrofitting Security into Internet Infrastructure Protocols," *Proc. DARPA Info. Surv. Conf. Expo*, 2000.
[13] S. Kent, C. Lynn, and K. Seo, "Secure Border Gateway Protocol (S-BGP)," *IEEE JSAC*, vol. 18, no. 4, Apr. 2000.
[14] D. E. Denning, "An Intrusion-detection Model," *IEEE Trans. Eng.*, vol. SE-13, no. 2, Feb. 1987, pp. 222–32.
[15] B. Mukherjee, L. T. Heberlein, and K. N. Levitt, "Network Intrusion Detection," *IEEE Net.*, May/June 1994, pp. 26–41.
[16] H. Y. Chang, S. F. Wu, and Y. F. Jou, "Real-Time Protocol Analysis for Detecting Link-State Routing Protocol Attacks," *ACM Trans. Info. Sys. Sec.*, vol. 4, no. 1, Feb. 2001, pp. 1–36.
[17] S. Cheung and K.N. Levitt, "Protecting Routing Infrastructures from Denial of Service Using Cooperative Intrusion Detection," *Proc. New Sec. Paradigms Wksp.*, Sept. 1997.
[18] K. Bradley *et al.*, "Detecting Disruptive Routers: A Distributed Network Monitoring Approach," *Proc. IEEE Symp. Sec. Privacy*, 1998.
[19] S. Kent *et al.*, "Secure Border Gateway Protocol (S-BGP) — Real World Performance and Deployment Issues," *Proc. Net. Distrib. Sys. Sec. Symp.*, San Diego, CA, Feb. 2000.
[20] S. Kent and R. Atkinson, "Security Architecture for the Internet Protocol," RFC 2401, Nov. 1998.

BIOGRAPHIES

PANAGIOTIS PAPADIMITRATOS [M] (papadp@ece.cornell.edu) is a Ph.D. candidate in the School of Electrical and Computer Engineering, at Cornell University, Ithaca, New York, affiliated with the Wireless Networks Laboratory. His research is concerned with security of computer and communication networks, design of network protocols, and wireless mobile networks. His work focuses on the security and fault tolerance of routing protocols, with an emphasis on solutions to support mobile ad hoc networking. His personal URL is: http://www.people.cornell.edu/pages/pp59/

ZYGMUNT J. HAAS [SM] (haas@ece.cornell.edu) received his B.Sc. in electrical engineering in 1979 and M.Sc. in electrical engineering in 1985. In 1988 he earned his Ph.D. from Stanford University and subsequently joined AT&T Bell Laboratories in the Network Research Department. There he pursued research on wireless communications, mobility management, fast protocols, optical networks, and optical switching. From September 1994 till July 1995 he worked for the AT&T Wireless Center of Excellence, where he investigated various aspects of wireless and mobile networking, concentrating on TCP/IP networks. In August 1995 he joined the faculty of the School of Electrical and Computer Engineering at Cornell University. He is an author of numerous technical papers and holds 15 patents in the fields of high-speed networking, wireless networks, and optical switching. He has organized several workshops, delivered tutorials at major IEEE and ACM conferences, and serves as editor of several journals and magazines, including *IEEE Transactions on Networking*. He is chair of the IEEE Technical Committee on Personal Communications (TCPC). His interests include mobile and wireless communication and networks, personal communication service, and high-speed communication and protocols. His URL is http://wnl.ece.cornell.edu