S-RIP: A Secure Distance Vector Routing Protocol [⋆]

Tao Wan Evangelos Kranakis P.C. van Oorschot

{twan, kranakis, paulv}@scs.carleton.ca
School of Computer Science, Carleton University, Ottawa, Canada

Abstract. *Distance vector routing protocols (e.g., RIP) have been widely used on the Internet, and are being adapted to emerging wireless ad hoc networks. However, it is well-known that existing distance vector routing protocols are insecure due to: 1) the lack of strong authentication and authorization mechanisms; 2) the difficulty, if not impossibility, of validating routing updates which are aggregated results of other routers. In this paper, we introduce a secure routing protocol, namely S-RIP, based on a distance vector approach. In S-RIP, a router confirms the consistency of an advertised route with those nodes that have propogated that route. A reputation-based framework is proposed for determining how many nodes should be consulted, flexibly balancing security and efficiency. Our threat analysis and simulation results show that in S-RIP, a well-behaved node can uncover inconsistent routing information in a network with many misbehaving nodes assuming (in the present work) no two of them are in collusion, with relatively low extra routing overhead.*

Keywords: Routing Security, Distance Vector, Distance Fraud, Security Analysis

1 Overview

It is well-known that today's Internet is not secure. Both Internet applications and the underlying routing infrastructures are vulnerable to a variety of attacks. Although a majority of incidents reported so far are realized by the exploitation of software vulnerabilities in client and server machines, it has been noted long ago that abusing routing protocols may be the easiest way for launching attacks [2], and a single misbehaving router can completely disrupt routing protocols and cause disaster [23]. This viewpoint has been more recently expressed by a group of network and security experts [4].

There are many factors that make today's routing infrastructures insecure. Three of them are as follows. 1) There are no strong security services built into routing protocols. Many routing protocols only provide weak authentication mechanisms, e.g., plain-text password or system-wide shared keys, for authenticating peers or routing updates. As a result, it is easy for an adversary to gain access to the routing infrastructure and manipulate routing information. 2) Software vulnerabilities and misconfigurations expose routing infrastructures to severe risks. 3) Most routing protocols assume a trustworthy environment. In the case where no authentication mechanisms are implemented, routing updates are accepted with only rudimentary validation. When authentication mechanisms are present, routing updates are verified for the correctness of data origin and integrity only. However, after a route update is verified to be "authentic", the routing information conveyed in the update is trusted and used to update the recipient's routing

[⋆] Version: April 12, 2004.

table. This is risky since data origin authentication, which includes data integrity [17], cannot guarantee the factual correctness of a message. A malicious entity or a compromised legitimate entity can send false information in a correctly signed message. A recipient can detect unauthorized alteration of the message, but cannot tell if the information conveyed in the message is factually correct unless the recipient has the perfect knowledge of what it expects to receive.

The difficulty of validating DV routing updates arises due to the fact that they are the distributed computational results of other nodes [22, 31]. Mittal and Vigna [18] propose to use intrusion detection sensors for validating routing advertisements by comparing a routing update with a master routing database that is pre-computed off-line. One disadvantage is that their approach cannot prevent fraudulent misinformation from poisoning others' routing tables, although it may be able to detect it. Hu, Perrig, and Johnson [9] propose to use hash chains and authentication trees to authenticate the distance of a route. However, their approach does not address longer distance fraud.

We present a secure DV routing protocol, namely *S-RIP*, based on RIP [15], which can prevent router and prefix impersonation, as well as shorter and longer distance fraud. In *S-RIP*, an advertised route is validated for its factual correctness before being used to update a routing table. Given the difficulty of validating the factual correctness of routing information in a DV routing protocol, we propose to use *consistency* as an approximation of *correctness*. An advertised route is treated as correct if it is consistent among those nodes that have propagated that route. Unless those nodes involved in a consistency check are in collusion, with high confidence a consistent route is correct. By this approach, we hope that nodes surrounding a misbehaving node will uncover inconsistency and prevent misinformation from further spreading.

A reputation-based framework is proposed for determining how many nodes to involve in a consistency check, providing the flexibility for balancing security and efficiency. Firstly, the notion of either trusting or distrusting a node is replaced by *node reputation* measured by a numeric value. Although in an intra-domain routing protocol (e.g., RIP), routers are under a single administrative domain and tend not to be mutually suspicious, they could be compromised due to software flaws. Malicious nodes can also manage to join a routing domain by exploiting routing vulnerabilities. Therefore, fully trusting any individual node even in an intra-domain routing protocol may introduce the vulnerability that a malicious node can call into question the legitimacy of other nodes. Node reputation provides the flexibility to relax this notion, and can be interpreted as an estimation that a node will provide correct information in the near future. Secondly, we propose an efficient method for computing the accumulated confidence in the correctness of a consistent routing update from the reputations of those nodes involved in the consistency check. Combined with confidence thresholds, this method effectively creates a *sized window* for determining how many nodes to involve in a consistency check.

The sequel is organized as follows. Section 2 analyzes RIP vulnerabilities. Section 3 presents security objectives and mechanisms of *S-RIP*. The reputation-based framework is presented in Section 4. *S-RIP* is presented and analyzed in Section 5. Section 6 presents simulation results. Section 7 reviews related work for securing routing protocols, with emphasis on securing DV routing protocols. Further comments and future work are discussed in the last section.

2 Background: RIP Vulnerabilities

RIP (we mean RIPv2) is an Internet Standard intra-domain DV routing protocol (see [15] for details). Despite certain limitations, e.g., the maximum distance between two nodes is 15 hops, it is still used by many small and medium size organizations (including some universities). RIP has several known security vulnerabilities. Five of them are discussed below.

1) An unauthorized node can easily join a routing domain and participate in routing operations. This is referred to as *router impersonation*. RIPv1 [8] does not have any authentication mechanism. RIPv2 only uses a clear-text password for authenticating peers. Since a clear-text password can be easily captured, it provides only marginal additional security in practice. Keyed MD5 has been proposed [1] to replace the password-based authentication mechanism. However, it is still vulnerable in that one compromised router discloses keying materials of every other router in the network.

In addition, RIP does not have any mechanism for preventing a *questionable node* (an unauthorized node or a compromised/malicious legitimate node) from advertising fraudulent routing information about distance or next hop.

2) A questionable node can claim a zero distance to a non-directly connected network or a nonexistent network. This is often referred as *prefix impersonation*. The proposed MD5 authentication [1] requires a system-wide shared secret key(s). This makes router impersonation harder, but cannot prevent prefix impersonation. Although prefix impersonation is a bigger issue in inter-domain routing protocol (e.g., BGP), it can also cause serious problems in intra-domain routing protocol (e.g., RIP). Figure 1 shows that a malicious node can easily launch service disruption (a type of denial of service) attacks by prefix impersonation. A similar incident (referred to as a blackhole) has occurred in the ARPANET [16].

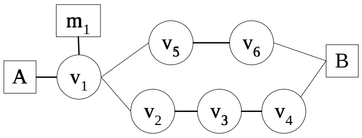

Fig. 1. m_1 advertises a zero distance route for B. As a result, v_1's routing table is poisoned by an incorrect route for B. Traffic from A to B will be forwarded by v_1 to m_1, which causes service disruption against A since m_1 does not have a route to B other than the one via v_1.

3) A questionable node may claim a distance shorter than the actual distance to a destination. This is called *shorter distance fraud*. This fraud can be used to attract traffic to launch a variety of attacks (e.g., eavesdropping, session hijacking).

4) A questionable node can claim a distance longer than the actual distance for a destination. This is called *longer distance fraud*. This fraud can be used to avoid traffic, which may lead to unfair utilization of network links and cause network congestion. Thus, it can be used to launch a denial of service attack. This fraud is different from malicious packet dropping attacks. While they both result in packet dropping, the latter can be detected by known techniques (e.g., secure traceroute [20]) while the former is more stealthy.

5) A questionable node may advertise arbitrary routing information or carefully crafted routes to poison others' routing tables, e.g., to cause routing loops or have invalid routes installed, and can also provide false information on a next hop.

3 Security Objectives and Mechanisms of S-RIP

To counter security vulnerabilities of RIP, we propose a new secure DV routing protocol, namely *S-RIP*. The security objectives of *S-RIP* include: 1) preventing router impersonation; 2) preventing prefix impersonation; and 3) preventing distance fraud (both shorter and longer). Fraud can be committed by individual nodes or colluding nodes. In this paper, we only consider uncoordinated individual fraud and leave the discussion of collusion to the future work. Our proposed mechanisms for achieving the above objectives are discussed below.

3.1 Preventing Router Impersonation

To prevent *router impersonation*, we require Assumption A1: every router shares a different key with every other router in a RIP domain. With A1 and an authentication algorithm (e.g., keyed MD5), a router can effectively detect router impersonation by validating a message authentication code (MAC) of a routing update message. Pairwise shared keys make it more difficult for an unauthorized node to impersonate a legitimate node, and ensure that the keying materials of one router will not be disclosed when another router is compromised. Of course, use of shared keys results in additional complexity; due to space limitations, we omit further discussion here.

3.2 Preventing Prefix Impersonation

To prevent *prefix impersonation*, we require Assumption A2: there is a central authority (e.g., a network administrator) with perfect knowledge of which router is physically connected to which subnets in that autonomous system (AS). Such perfect knowledge, or router-prefix mapping, is realistic for an AS since network configurations are administratively controlled by a single authority. The router-prefix mapping is then securely distributed to each router, e.g., it can be pre-configured on each router. Ongoing update (e.g., additions of subnets or routers) can then be done through a secure channel (e.g., SSH) between the central authority and each router. Although network topology may be dynamic (e.g., caused by link failures), we expect router-prefix mapping is relatively static since addition/deletion of subnets usually occurs far less frequently than link failures. Other alternatives can also be used to prevent prefix impersonation, e.g., *address attestation* in S-BGP [14], *authorization certificates* in soBGP [32], etc. However, they may require a public key infrastructure, which has its own drawbacks.

3.3 Preventing Distance Fraud

Shorter and longer distance frauds are difficult to prevent. In a distance vector routing protocol, routing updates received by a node are computational results or aggregated routes of other nodes. Unless a node has perfect knowledge of network topology and dynamics, it appears impossible to validate the factual correctness of aggregated routing updates [22, 31].

We propose to use *consistency* as an approximation of correctness. An advertised route is validated by cross checking its consistency with the routing information of those nodes from which this route is derived. If the route is consistent among those nodes, it is treated as correct. Otherwise, incorrect. For example, in Figure 2, when node v_2 advertises to v_1 a 2-hop route for v_5 with v_3 as the next hop, v_1 queries v_3's route for v_5, which is 2 hops. Since v_2's route for v_5 is supposed to be one hop longer than v_3's route for v_5 (this is specifically based on RIP, but can be easily generalized), an inconsistency is detected. Although v_1 does not know which node (v_2 or v_3) provides invalid information, v_1 knows that something is abnormal with this route. Therefore, this route is dropped. If v_2 advertises a 3-hop route for v_5, it is consistent with v_3's 2-hop route. Thus, it may be accepted. §5 presents the algorithm details for consistency checks and analyzes various threats.

To support consistency checks, we require Assumption A3: a node indicates (either voluntarily for direct neighbors or upon request otherwise) the next hop of each route in its routing table. For example, in Figure 2, v_2 should tell v_1 that v_3 is the next hop on the route for v_5. v_3 should also tell v_1 that v_4 is its next hop to v_5 upon request. Requests can be made by RIP route request or other mechanisms (e.g., SNMP MIB query [3]). If a node fails to provide information on next hops, its behavior is called into question.

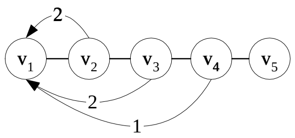

Fig. 2. Consistency Checks

One property of a DV routing protocol is that a node only communicates with its direct neighbors and does not need to maintain the network topology beyond its direct neighbors. In a link state (LS) routing protocol, a node advertises its link states to every other node in the network by flooding, and each node maintains a whole view of the network topology. A3 allows a node to query non-direct neighbors, which expands node-to-node communication boundary in a DV routing protocol to a dynamic area (by our reputation-based approach §4).

We thus note that our approach falls in between the DV and LS approaches. Pictorially, the communication range of an LS node covers the whole network (flooding), while the communication range of a traditional DV node only covers its direct neighbors (neighbor-to-neighbor). In *S-RIP*, the communication range of a node is dynamic. Although it is certainly beyond direct neighborhood and could reach the whole network, most likely, it will only cover a nearby neighborhood (e.g., within 2 or 3 hops) dependent on window size (§4.3). Therefore, additional routing overhead generated by non-neighbor querying is limited, as confirmed by our simulation results in §6. Requirement of storage space is also increased in *S-RIP*, but very slightly since an *S-RIP* node only needs to maintain the information of remote nodes when they are being or will be consulted for a consistency check.

Another question which arises is: how does a node query a remote node if it does not have a known route for that node? For example, in Figure 2, for v_1 to validate a route for v_3, v_1 may need to query v_3. However, v_1 cannot talk to v_3 if it does not have a route for v_3. This is a known problem that a secure routing protocol relies upon a routing protocol for node reachability. In *S-RIP*, a temporary routing table is maintained, which contains all routes to be validated. The temporary routing table is only used for route validation (not for routing data traffic). When a route passes a validation, it is moved to the regular routing table and can be used for routing data traffic. In the above example, v_1 first installs the route for v_3 into the temporary routing table, and sends to v_2 a routing request destined for v_3. v_2 should have a route for v_3 since it advertises such a route to v_1 (otherwise, it is misbehaving). When v_3 receives a route request from v_1, it sends back to v_1 a route response via a route either in its temporary routing table or the regular one. This route request and response process incurs additional routing overhead, but also adds another level of assurance that intermediate nodes are actually forwarding packets. If we can make a route request or response message indistinguishable from a normal data packet (e.g., by IPSec ESP [13]), this process may detect forwarding level misbehavior, (i.e., a router advertising correct routes but does not forward data packets).

To implement A3 in RIP, the next hop field in a RIP routing update message can be utilized. In RIP, the next hop field is only used for route optimization (avoiding an extra hop). For example, v_2 will not include v_3 in the next hop field (by setting it to 0) unless it believes that v_1 should forward traffic destined for v_5 directly to v_3. With A3, v_2 voluntarily includes v_3 in the next hop. This changes the meaning of a next hop from *this is your next hop* to *this is my next hop*. Thus, A3 allows a receiving node, instead of an advertising node, to decide which node should be the next hop. Despite the change of the meaning, A3 is still compatible with RIP since a receiving node will ignore the next hop field (treats it as null) if it is not directly reachable. To interoperate with an existing implementation of RIP, an *S-RIP* node may get next hop information from a RIP node by external mechanisms, e.g., SNMP MIB query.

4 Reputation-Based Framework

In this section we present a reputation-based framework, consisting of a reputation update function, an efficient method of computing accumulated confidence, localized rules for processing routing updates, and a sized window method for balancing security and efficiency.

4.1 Reputation Definition

We propose to use node reputation as an estimation of the confidence in that a node will provide correct routing information in the near future. Every node assigns an initial value as the reputation of every other node in a network. A node's reputation is then dynamically updated by Equation 1. The detail of how this equation is derived is given in [30]. Many possibilities exist for $c_i(j, t+1)$. We propose Equation 2 for its simplicity.

$$r_i(j, t+1) = \frac{r_i(j, t)}{2} + c_i(j, t+1) \tag{1}$$

$$c_i(j, t) = \begin{cases} 0.5 & \text{if } j \text{ provides consistent information at time } t \\ 0 & \text{otherwise (e.g., if } j \text{ provides conflicting information at time } t) \end{cases} \tag{2}$$

One property of Equation 1 is that if $r_i(j, t) \neq 1$, $r_i(j, t+1)$ will be always less than 1. Thus, if node i does not assign an initial value of 1 or higher as j's reputation, $r_i(j)$ will always be in the range $[0, 1)$. We propose Equation 3 for computing an accumulated confidence from node reputation in the correctness of a routing update consistent among a group of nodes.

Definition 1 (Accumulated Confidence) *Let $r_x(v_1), r_x(v_2), \ldots, r_x(v_n)$ be x's rating of the reputation of nodes $v_1, v_2, \ldots, v_n$, respectively. In the case that routing information from nodes $v_1, v_2, \ldots, v_n$, is consistent, node x's confidence in that information, denoted by $r_x(v[1..n])$, is defined as follows, where $v[1..n]$ denotes $v_1, v_2, \ldots, v_n$:*

$$r_x(v[1..n]) = \begin{cases} r_x(v_1) & \text{if } n = 1 \\ r_x(v_1) + \big(1 - r_x(v_1)\big) \cdot r_x(v_2) & \text{if } n = 2 \\ r_x(v[1..n-1]) + \Big(1 - r_x(v[1..n-1])\Big) \cdot r_x(v_n) & \text{if } n > 2 \end{cases} \quad (3)$$

Although developed independently based on our intuition, it turns out that Equation 3 is consistent with Dempster-Shafer theory (DST) of evidence reasoning [5, 27] if we assume that in our case, for all i ($1 \leq i \leq n$), v_i acquires its information from an independent source. The proof is given in [30]. The advantage of Equation 3 is that it is intuitive and computationally efficient. Although DST is more general, e.g., it can handle conflicting information, it is computationally less inefficient since it involves set operations.

4.2 Validation Rules

We propose a set of rules for determining how to treat routing advertisements based on node reputation. Two thresholds (θ_1, θ_2) are used to divide the reputation domain into three levels, namely low, medium, and high.

Rule 1 *(***Low Reputation***). If node j's reputation rated by i is in the low range ($0 \leq r_i(j) < \theta_1$), node i will ignore a routing advertisement from j without cross-checking its consistency with any other node(s).*

Rule 2 *(***Medium Reputation***). If node j's reputation rated by i is in the medium range ($\theta_1 \leq r_i(j) < \theta_2$), node i will cross check the consistency of a routing advertisements from j with other node(s).*

Rule 3 *(***High Reputation***). If node j's reputation rated by node i is in the high range ($\theta_2 \leq r_i(j) \leq 1$), node i will cross check the consistency of a routing advertisement from j with only one other node.*

4.3 Sized Windows

Since there may be multiple nodes having propagated an advertised route, a mechanism is required to decide how many nodes to involve in a consistency check. The more nodes consulted (which agree with the the advertised route), the higher the confidence acquired in the correctness of that route; but the network overhead will also be higher. We use a *sized window* as a mechanism for balancing the trade-off between security and efficiency. The size of the window is the number of the nodes consulted in a consistency check. The window size starts from 1. In other words, there is only one node in the window before the consistency check of an advertised route, which is the advertiser of that route. The window size grows by one, or an additional node is consulted, if the computed confidence using Equation 3 in the correctness of that route is less than

θ_2. The window size keeps growing for the advertised route until 1) an inconsistency occurs, i.e., a node reports conflicting information; or 2) all the nodes in the window agree upon the route, and 2.1) the computed confidence is greater than θ_2; or 2.2) all informed nodes have been involved. In case 1), the route fails the consistency check and is dropped. In case 2), the route succeeds the consistency check and is accepted.

5 Secure Routing Information Protocol (*S-RIP*)

We present the detail and analysis of *S-RIP*. For an advertised route $[dest, dist, nh]$, we use v_0, v_1, and v_n to represent the recipient, the advertiser, and the ultimate destination respectively. To be more specific, we use $dist(v_1, v_n)$ and $nh(v_1, v_n)$ to represent the distance and the next hop respectively from v_1 to v_n for this particular route.

5.1 *S-RIP*

When router v_0 receives from v_1 an advertised route $[v_n, dist(v_1, v_n), nh(v_1, v_n)]$, v_0 validates the route as required by RIP [1]. If the route passes the validation, and will be used to update v_0's routing table, *S-RIP* is triggered to perform additional validations. *S-RIP* will NOT be triggered if the advertised route does not indicate a route change or a topology change. Although the timer associated with this route will be re-initialized, there is no need to re-validate the route since such a validation should have been done when the route was first installed in v_0's routing table. Highlights of *S-RIP* on validating $[v_n, dist(v_1, v_n), nh(v_1, v_n)]$ are given immediately below. More details are presented in the remainder of this section.

1. Is the advertised route self-consistent? If not, drop the route.
2. If $dist(v_1, v_n) = 0$, v_0 performs router or prefix authentication. If the authentication succeeds, v_0 accepts the route. Otherwise, drops it.
3. If $1 \leq dist(v_1, v_n) < 15$, v_0 checks the consistency of $[v_n, dist(v_1, v_n), nh(v_1, v_n)]$. If the consistency check succeeds, v_0 accepts the route. Otherwise, drops it.
4. If $dist(v_1, v_n) \geq 15$, v_0 accepts the route without validating it.

Self-consistency Check. v_0 checks if $[v_n, dist(v_1, v_n), nh(v_1, v_n)]$ is self-consistent. 1) If v_1, v_2, or v_n is not a legitimate entity, the route is dropped. A router is legitimate to v_0 only if v_0 shares a secret key with it. 2) If $dist(v_1, v_n) = 0$, $nh(v_1, v_n)$ should be v_1 itself since the advertised route is for v_1 or a subnet directly attached to v_1. 3) If $1 \leq dist(v_1, v_n) < 15$, the next hop must not be v_0 or v_1. v_1 should not advertise a valid route back to v_0 from which it learns that route. Otherwise, the problem of counting to infinity occurs. Although RIP recognizes this problem and proposes split horizon (or with poisoned reverse) for solving it, a misbehaving node may not follow the rule and intentionally create the problem.

Router/Prefix Authentication. If $dist(v_1, v_n) = 0$, v_1 advertises to v_0 a route for itself or for a subnet directly attached to v_1. If the route is for v_1 itself, message authentication already provides data origin authentication [17]. If the route is for a subnet, the router-prefix mapping (§3.2) is used to validate if v_1 is physically connected to that subnet. If the validation succeeds, the router is accepted. Otherwise, dropped.

Consistency Check. If $1 \leq dist(v_1, v_n) < 15$, v_1 advertises to v_0 a reachable route for v_n. v_0 will check the consistency of that route with $nh(v_1, v_n)$, let's say v_2. v_0

will request from v_2 the routing information from v_2 to v_n and v_1. The message flows are given in Table 1, where * denotes a information field to be provided. The advertised route from v_1 for v_n is treated as consistent with v_2's routing information if $dist(v_2, v_1) = 1$ and $dist(v_1, v_n) = dist(v_2, v_n) + 1$ (based on RIP). Otherwise inconsistent.

$v_0 \rightarrow v_2$	$[v_n, *, *]$ $[v_1, *, *]$
$v_0 \leftarrow v_2$	$[v_n, dist(v_2, v_n), nh(v_2, v_n)]$ $[v_1, dist(v_2, v_1), nh(v_2, v_1)]$

Table 1. Routing Request and Response

If v_1 is consistent with v_2, v_0 will use Equation 3 to compute an accumulated confidence, $r_{v_0}(v_1, v_2)$. If $r_{v_0}(v_1, v_2) \geq \theta_2$, v_0 accepts the advertised route as correct. Otherwise, v_0 will consult with additional nodes based on the next hop information. Before v_0 sends a route request to node v_i, it checks if a network loop has been formed. A network loop is formed if the node (v_i) to be consulted has been consulted before. In the case that a loop is detected, v_0 drops the advertised route. Otherwise, the consistency check continues until one of the following three conditions holds: 1) $r_{v_0}(v[1..k]) \geq \theta_2$. In this case, the advertised route from v_1 is treated as correct by v_0. 2) $r_{v_0}(v[1..k-1]) < \theta_2$, and v_k disagrees with v_{k-1}, i.e., $dist(v_{k-1}, v_n) \neq dist(v_k, v_n) + dist(v_k, v_{k-1})$. In this case, v_0 treats the advertised route as inconsistent. 3) v_n has been consulted. If v_n disagrees with v_{n-1}, the advertised route from v_1 is treated as inconsistent. Otherwise, v_0 will performs router/prefix authentication with v_n. If v_n succeeds the authentication, the advertised route is treated as correct no matter what the value of $r_{v_0}(v[1..n])$ is. Otherwise, the advertised route is dropped as v_n provides incorrect information.

Infinity Route. If $dist(v_1, v_n) \geq 15$, v_1 advertises to v_0 an route for v_n which is infinite from v_0. v_0 does not validate an infinite or unreachable route since it is trivial for v_1 to make a valid route unreachable if it misbehaves, e.g., by disabling a network interface or dropping packets. The consequence of such possible misbehavior is that v_0 will drop the route and will not forward packets to v_n through v_1. If there is only one route in the network from v_0 to v_n and it goes through v_1, v_0 will not be able to communicate with v_n. It seems to be hard to force a misbehaving node forward packets for others if it is determined not to do so. Therefore, we hope a network is designed with redundancy to accommodate a single point of failure. In that case, hopefully v_0 could find an alternative route to v_n, bypassing the misbehaving node v_1.

5.2 Threat Analysis

A node may misbehave in several ways: 1) advertising false routing information; 2) providing false routing information specifically during a consistency check; 3) dropping a validation request/reply message or not responding to a validation request; 4) manipulating a validation request/reply message originated from other nodes; 5) providing correct routing information but not forwarding data traffic.

1) *Advertising false routing information*. Given a route $[v_n, dist(v_1, v_n), nh(v_1, v_n)]$ advertised by node v_1 to v_0, v_1 may provide false information about v_n, *dist*, *nh*, or any combination.

1.1) *Destination Fraud*. v_1 may advertise a route for a nonexistent destination v_n. Under our proposal, such misbehavior can be detected since v_0 does not share a secret key with v_n if it is not a legitimate entity in the network.

1.2) *Distance Fraud*. v_1 may advertise a fraudulent distance to a destination v_n, e.g., longer or shorter than the actual distance. If $dist(v_1, v_n) = 0$, but v_1 is actually one or more hops away from v_n, in our proposal, v_0 can detect this fraud by router/prefix authentication. Other shorter or longer distance fraud can be detected by cross checking consistency with those nodes which propagated the route in question. There are three scenarios in which a consistency in the corroborating group may not represent correctness: a) the nodes in the corroborating group are simultaneously misled by one or more misbehaving nodes; b) the nodes in the corroborating group are colluding; c) a subset of the corroborating group are colluding and mislead the rest of the nodes. Our idea is that by increasing the size of the corroborating group, it is increasingly unlikely that these scenarios will not be detected.

1.3) *Next Hop Fraud*. Node v_1 may provide a fraudulent next hop to support its claim of a longer or shorter distance. First, v_1 may use fictional nodes as next hops. v_1 then intercepts from v_0 the subsequent validation requests to these nodes and send back false responses on behalf of them. In our scheme, a fictional node can be detected since v_0 does not share a prior secret with it. Second, v_1 may use a remote node (i.e., a node not directly connected to v_1) as the next hop. For example, suppose v_1 is 5 hops away from v_n. If v_1 learns that v_m is one hop away from v_n, it may claim to be two hops away from v_n and use v_m as the next hop. Unless v_m is willing to provide false information (e.g., $dist(v_m, v_1) = 1$) to cover v_1's misbehavior, v_0 will be able to detect this fraud. In the case that v_m is willing to collude with v_1, we treat it as the case that v_1 establishes a virtual link (e.g., TCP connection) with v_m, and they forward packets over the virtual link to each other. This misbehavior is equivalent to the *wormhole* attack studied by Hu, Perrig, and Johnson [10]. *S-RIP* may detect such attack if a prior knowledge of node physical connections is assumed. Otherwise, the proposed *Packet Leashes* defense mechanism [10] should be used.

2) *Providing false routing information* in a consistency check. The fraud could be on distance or next hop. When the false information cause inconsistency, the consequences are: 2.1) correct routing advertisements may be disregarded by well-behaved nodes. We think it is not to the advantage of a misbehaving node to mislead another node by this type of misbehavior since it may be best to avoid a “valid” route through a misbehaving node in any case. By dropping a route involving a misbehaving node, the validation node may take an alternative good route, albeit possibly suboptimal. 2.2) the reputation of a well-behaved node may be decreased as a result of false information arising from a misbehaving node. In the worst case, if node v_0's rating of node v_1's reputation is decreased to the low range, v_0 will disregard v_1's routing advertisements for a certain period of time. Since consistency checks occur only on route changes, a misbehaving node, v_m, may only damage the reputation of v_1's reputation when there is a route change which involves both v_m and v_1 in a consistency check. v_m's own reputation may also be decreased if it provides false information. Therefore, v_m is unable to damage another node's reputation at its will. On the other hand, v_1 has other chances to increase its reputation when it advertises good routes (without going through v_m) to v_0. So the

effect of the type of misbehavior depends on the network topology and the location of the misbehaving nodes. If one or more misbehaving nodes are located on the links which can form a network-cut, they may be able to completely separate the network through collusion. It would appear no approach is resilient to such misbehavior.

3) *Dropping a validation request/reply message or not responding to a validation request*. This misbehavior can disrupt a validation process. As a result, the route being validated will be dropped. We do not consider this as a major drawback since dropping a route with misbehaving nodes en route allows an alternative route to be discovered. An adversary may launch this type of attack when it is not willing to forward packets for other nodes. As discussed before, a misbehaving node can avoid traffic by many other ways, e.g., dropping packets based on source or destination addresses, or simply disabling a network interface. We rely upon network redundancy and other mechanisms [20, 12] to counter this type of misbehavior.

4) *Manipulating a validation request/response message* originated from other nodes. If all routers are deployed with *S-RIP* and use MD5 for message authentication, validation request/response messages cannot be manipulated en route. However, communication between a secured router and a remote non-secured router is not authenticated. The consequences are: 4.1) A routing response sent back by a remote non-secured router can be modified by an adversary en route. The adversary may modify the routing response in such a way that it would confirm the consistency of a false advertised route. 4.2) An adversary may intercept routing requests sent to a non-secured router, and produce false responses on behave of that router. This vulnerability can be addressed by IP layer security. For example, if IPSec is available, an adversary would not be able to manipulate or intercept routing requests or responses between two remote nodes. It can also be mitigated if we assume that an adversary does not have the capability to launch attacks in packet level. It is easy for an adversary to manipulate a routing table to make a router to broadcast fraudulent routing information. It may not be that easy to manipulate packets transmitted through a router if the adversary does not have sufficient control over that router, e.g., modify and compile source codes, install malicious software, etc.

5) *Providing correct routing information but not forwarding data traffic*. We can make routing request and response messages indistinguishable from normal data traffic to validate forwarding level behavior of intermediate routers. Other detection techniques (e.g., probing [12]) for identifying such misbehaving routers can also be integrated into *S-RIP*, we do not address the issue in this paper.

One characteristic of *S-RIP* is that it does not guarantee that a validated route is optimal. In fact, *S-RIP* only validates route consistency, without considering the cost. *S-RIP* always accepts a consistent route and disregards an inconsistent one regardless of its cost. Therefore, optimal route involving a misbehaving node may not be used. We consider this as a good tradeoff between routing security and efficiency.

5.3 Efficiency Analysis

We consider the worst case here. The efficiency of average cases is analyzed by simulation (§6).

Suppose there are n routers and m subnets in a network. The average length of a route is $l+1$ hops. For maximum security, every router would validate every route with all other routers on that route. For a single route with a length of $l+1$ hops, the number of messages required for a consistency check, including requests and responses, is $2 \cdot l$. Each message will travel a number of hops. The first request message is sent to the node in two hops, and will travel 2 hops. The last request message is sent to the node in $l+1$ hops, and will travel $l+1$ hops. A response message will travel the same number of hops as the corresponding request message assuming they travel at the opposite direction of a same route. Therefore, the total number of hops (message transmissions) traveled by both request and response messages is $2 \cdot [2+3+\cdots+(l+1)] = (1+l) \cdot l$. Assume every router keeps a route for every subnet in the network. Each router would need $(1+l) \cdot l \cdot m$ message transmissions for validating every route. Over the whole network, the total number of message transmissions in the most secure case is $(1+l) \cdot l \cdot m \cdot n$.

We use RIP messages for route request and response. Each route request would need two route entries, one for the routing information from the recipient to the ultimate destination, and one from the recipient to its predecessor node on that route. The RIP message header is 24 bytes including authentication data, and each route entry is 20 bytes. Thus, one route request or response is 64 bytes. Plus the UDP header (8 bytes) and IP header (20 bytes), a packet carrying a route request or response is 92 bytes. The total overhead of routing validation, in addition to the overhead of regular routing updates, in the most secure case, is $92 \cdot (1+l) \cdot l \cdot m \cdot n$ bytes.

As confirmed by our simulation (§6), the validation overhead by *S-RIP* is prohibitively expensive in the maximally secured case. However, *S-RIP* provides the flexibility for balancing security and efficiency via two configurable thresholds θ_1 and θ_2 (§4.2). In practice, we expect that the maximally secured case may only be applied to a small size network (i.e., the number of nodes and network diameter are small). In other scenarios, θ_1, θ_2 can be adjusted to obtain a comfortable level of security and efficiency.

S-RIP validation overhead can also be reduced by optimized implementation (e.g., transmitting several route requests or responses in a single message). For example, if v_1 advertises to v_0 three routes with a same next hop v_2. v_0 can send a single message with 4 route entries to v_2, one for each of three advertised destinations and one for v_1. The size of the packet carrying this message is 132 bytes, considerably less than 276 bytes which are the total size of three standard packets (each has a length of 92 bytes).

5.4 Incremental Deployment

A practical challenge of securing routing protocols is how to make the secured version interoperative with the existing infrastructure. Despite their technical merits, many proposed mechanisms for securing routing protocols are not widely deployed due to the fact that they require significant modifications to existing implementations and/or do not provide backward interoperability. Since it is unrealistic to expect that an existing routing infrastructure can be replaced by a secured version in a very short period of time, ideally a secured version should be compatible with the insecure protocols. It is also desirable that security can be increased progressively as more routers are deployed with the secured protocol.

To this end, *S-RIP* supports incremental deployment. We propose that messages exchanged in *S-RIP* conform to the message format defined in RIP. *S-RIP* can be implemented as a compatible upgrade to the existing RIP, and a *S-RIP* router performs routing functions the same way as a RIP router. Therefore, deploying *S-RIP* on a router only requires a down time for the period of installation and rebooting of RIP processes. Since RIP router responds to a routing request from a non-direct neighbor (a remote node), a *S-RIP* router can successfully get information (albeit not authenticated) from a non-secured router for a consistency check. In other words, a RIP router can participate in a consistency check, but not initiate a consistency check. Thus, even before *S-RIP* is deployed on all routers, the routing table of a *S-RIP* router is partially protected as it is built from validated routing updates. The more routers deployed with *S-RIP*, the more reliable routing tables in the network become. Therefore, we can say that security can be increased incrementally.

6 Simulation

We implemented *S-RIP* in the network simulator NS2 as an as an extension to the distance vector routing protocol provided by NS2. *S-RIP* is triggered if an advertised route is used to update a recipient's routing table. In this section, we present our preliminary simulation results on how routing overhead is affected by different threshold settings and number of misbehaving nodes in *S-RIP*.

Maximally Secured	$\theta_1 = 0$	$\theta_2 = 1$
Partially Secured-1	$\theta_1 = 0.1$	$\theta_2 = 0.9$
Partially Secured-2	$\theta_1 = 0.2$	$\theta_2 = 0.8$
Partially Secured-3	$\theta_1 = 0.3$	$\theta_2 = 0.7$
Not Secured	$\theta_1 = 0$	$\theta_2 = 0$

Table 2. Simulation Scenarios

6.1 Simulation Environment

Network Topology: we simulated *S-RIP* with a number of different network topologies. In this paper, we only present the simulation results for one topology which has 50 routers and 82 network links. *Fraud*: we simulated misbehaving nodes which commit either or both shorter and longer distance fraud (§3.3). We randomly selected 5, 10, 15, 20, and 25 nodes to commit fraud in each run of the simulation. Note that 25 misbehaving nodes represent 50% of the total nodes. Each misbehaving node periodically (every 2.5 seconds) randomly selects a route from its routing table and makes its distance shorter or longer. *Simulation Scenarios*: we simulated 5 scenarios (Table 2) by adjusting the thresholds θ_1 and θ_2. Each simulation runs 180 seconds.

6.2 Routing Overhead

To determine how much network overhead is generated by *S-RIP*, we compared the *S-RIP* overhead to the total routing overhead, which is calculated as the sum of *S-RIP* overhead and regular routing update overhead in RIP. Since the distance vector routing protocol provided by NS2 is not a strict implementation of RIP RFCs, we could not obtain network overhead directly from the NS2 trace file. We use $\frac{92x}{92x+632y}$ to calculate the ratio of *S-RIP* overhead and the total routing overhead, where x is the total number of *S-RIP* message transmissions, y is the total number of rounds of regular routing updates, 92 bytes is the size of the packet carrying a *S-RIP* message (see §5.3), and 632 bytes is the overhead generated by one router in one round of regular routing updates. x and y are derived from simulation outputs, which are used to generate Figure 3.

6.3 Simulation Results

By looking at the output data from the simulation, we observed that an advertised malicious route can be successfully detected by a consistency check. This is precisely what we expected.

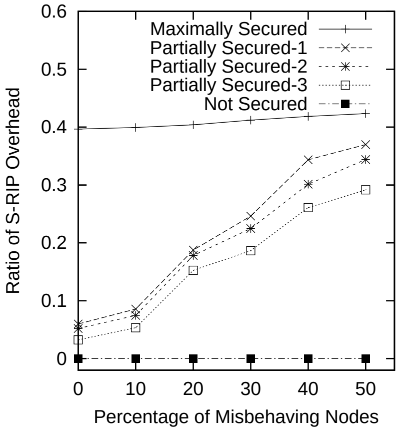

Fig. 3. *S-RIP* Routing Overhead.

Figure 3 compares the *S-RIP* overhead in different scenarios. 1) In a maximally secured network, *S-RIP* overhead is very high (about 40% of the total routing overhead). The *S-RIP* overhead stays relatively flat when the number of misbehaving nodes increases. This is because every node needs to validate every route with every other node on that route. In our implementation, a new route is not considered if the current route is being checked for consistency. Since it takes long time for a consistency check to complete, most new route changes (malicious or non-malicious) are not checked for their consistency. Therefore, overhead increased by new malicious updates is insignificant. This indicates that the speed of network convergence is significantly slowed down. We expect that it would make no difference in terms of overhead if we allow a new route to interrupt an ongoing consistency check as several uncompleted consistency checks would generate similar amount of *S-RIP* overhead as a completed one does. 2) In the three partially secured scenarios, *S-RIP* overhead is relatively low (less than 8.6%) when there are only 10% of misbehaving nodes. *S-RIP* overhead increases significantly when the number of misbehaving nodes increases. Since the number of nodes involved in a consistency check is relatively low in these scenarios, it takes less time to complete. Thus more malicious updates will trigger more consistency checks and result in more *S-RIP* overhead. *S-RIP* overhead decreases when θ_1 and θ_2 are moved toward each other because: a) the number of nodes involved in a consistency check decreases; b) the number of routes dropped without being checked for consistency increases when more than 20% of the nodes misbehave. 3) There is no *S-RIP* overhead in a non-secured network since *S-RIP* is never triggered.

7 Related Work

Significant work has been done in securing routing protocols. Perlman [22] is the first to study the problem of securing routing protocols. Perlman classified router failures into *simple failures* and *byzantine failures*, and proposed use of public key signatures, source routing, and other mechanisms, for achieving robust flooding and robust routing.

Smith et al. [29] proposed use of digital signatures, sequence numbers, and a loop-free path finding algorithm for securing DV routing protocols. One disadvantage is that it cannot prevent longer or shorter distance fraud.

Mittal and Vigna [18] proposed to use sensor-based intrusion detection for securing DV routing protocols. One notable advantage of their approach is that it does not require modifications to the routing protocol being secured. Thus, it allows incremental deployment. One disadvantage is that it cannot prevent fraudulent routing advertisements from poisoning others' routing tables, although it may be able to detect them.

Hu, Perrig and Johnson [9, 11] proposed several efficient mechanisms using one-way hash chains and authentication trees for securing DV routing protocols. Their approach is one of the first attempts to authenticate the factual correctness of DV routing updates, and can prevent shorter and same distance fraud. It can also prevent newer sequence number fraud if a sequence number is used to indicate the freshness of a routing update. However, it does not address longer distance fraud.

Pei et al. [21] proposed a triangle theorem for detecting potentially or probably invalid RIP advertisements. Probing messages based on UDP and ICMP are used to further determine the validity of a questionable route. One disadvantage is that probing messages may be manipulated. A node advertising an invalid route can convince a receiver that route is valid by: 1) manipulating the TTL value in a probing message; or 2) sending back an ICMP message (port unreachable) on behalf of the destination.

Many researchers have explored securing link state routing protocols (e.g., OSPF) [22, 19, 31] and BGP [28, 14, 7, 32]. Reputation-based systems have been used to facilitate trust in electronic commerce [25, 33].

8 Concluding Remarks

We expect our framework can be applied to other non-trustworthy environments, e.g., inter-domain routing protocols and wireless ad hoc networks. Future research includes: 1) performing detailed analysis of *S-RIP* and comparing it with other secure DV protocols (e.g., SEAD [11]); 2) applying the framework to securing BGP [24].

Acknowledgment

We thank anonymous reviewers for comments which significantly improved this paper. The first author is supported in part by Alcatel Canada, NCIT (National Capital Institute of Telecommunications), and MITACS (Mathematics of Information Technology and Complex Systems). The second author is supported in part by NSERC (Natural Sciences and Engineering Research Council of Canada) and MITACS. The third author is Canada Research Chair in Network and Software Security, and is supported in part by an NSERC Discovery Grant and the Canada Research Chairs Program.

References

[1] F. Baker and R. Atkinson. RIP-II MD5 Authentication. RFC 2082, January 1997.

[2] S.M. Bellovin. Security Problems in the TCP/IP Protocol Suite. *ACM Computer Communications Review*, 19(2): 32-48, April 1989.

[3] J. Case, M. Fedor, M. Schoffstall, and J. Davin. A Simple Network Management Protocol (SNMP). RFC 1157. May 1990.

[4] S. Deering, S. Hares, C. Perkins, and R. Perlman. Overview of the 1998 IAB Routing Workshop (RFC 2902). August, 2000.

[5] A.P. Dempster. Upper and Lower Probabilities Induced by a Multivalued Mapping. *The Annals of Statistics*, 28: pages 325-339, 1967.

[6] J.J. Garcia-Luna-Aceves and S. Murthy. A Loop-Free Algorithm Based on Predecessor Information. In *Proceedings of IEEE INFOCOM* , Boston, MA, USA. April 1995.

[7] G. Goodell, W. Aiello, T. Griffin, J. Ioannidis, P. McDaniel, and A. Rubin. Working around BGP: An Incremental Approach to Improving Security and Accuracy in Interdomain Routing. In *Proc. of NDSS'03*, San Diego, USA. Feb 2003.

[8] C. Hedrick. Routing Information Protocol. RFC 1058. June 1988.

[9] Y.C. Hu, A. Perrig, and D.B. Johnson. Efficient Security Mechanisms for Routing Protocols. In *Proc. NDSS'03*, San Diego, USA. Feb 2003.

[10] Y.C. Hu, A. Perrig, and D.B. Johnson. Packet Leashes: A Defense against Wormhole Attacks in Wireless Networks. In *Proc. of IEEE INFOCOM 2003*, San Francisco, USA. April 2003.

[11] Y.C. Hu, D.B. Johnson, and A. Perrig. SEAD: Secure Efficient Distance Vector Routing for Mobile Wireless Ad Hoc Networks. In *Ad Hoc Networks Journal*, 1 (2003):175-192.

[12] M. Just, E. Kranakis, and T. Wan. Resisting Malicious Packet Dropping in Wireless Ad Hoc Networks. In *Proc. of ADHOCNOW'03*, Montreal, Canada, Oct 2003.

[13] S. Kent and R. Atkinson. IP Encapsulating Security Payload. RFC 2406, Nov 1998.

[14] S. Kent and C. Lynn and K. Seo. Secure Border Gateway Protocol (Secure-BGP). *IEEE Journal on Selected Areas in Communications*, 18(4): 582-592, April 2000.

[15] G. Malkin. RIP Version 2. RFC 2453 (Standard). November 1998.

[16] J.M. McQuillan, G. Falk, and I. Richer. A Review of the Development and Performance of the ARPANET Routing Algorithm. *IEEE Trans. on Comm.*, 26(12): 1802-1811, Dec 1978.

[17] A.J. Menezes, P.C. van Oorschot, and S. Vanstone. *Handbook of Applied Cryptography*. CRC Press, 1996.

[18] V. Mittal and G. Vigna. Sensor-Based Intrusion Detection for Intra-Domain Distance-Vector Routing. In *Proc. of CCS'02*, Washington, D.C., USA. Nov 2002.

[19] S.L. Murphy and M.R. Badger. Digital Signature Protection of the OSPF Routing Protocol. In *Proc. of NDSS'96*, San Diego, USA. April 1996.

[20] V.N. Padmanabhan and D.R. Simon. Secure Traceroute to Detect Faulty or Malicious Routing. *ACM SIGCOMM Workshop on Hot Topic in Networks*, Princeton, NJ, USA. Oct 2002.

[21] D. Pei, D. Massey, and L. Zhang. Detection of Invalid Announcements in RIP protocols. *IEEE Globecom 2003*, San Francisco, California, USA. December 2003.

[22] R. Perlman. *Network Layer Protocols with Byzantine Robustness*. PhD thesis, MIT, 1988.

[23] R. Perlman. Interconnections: Bridges and Routers. Addison-Wesley, 1992.

[24] Y. Rekhter and T. Li. A Border Gateway Protocol 4 (BGP-4), RFC 1771, March 1995.

[25] P Resnick, R. Zeckhauser, E. Friedman, and K. Kuwabara. Reputation systems: Facilitating trust in Internet interactions. *Communications of the ACM*, 43(12): 45-48, 2000.

[26] R. Rivest. The MD5 Message-Digest Algorithm, RFC 1321, April 1992.

[27] G. Shafer. *A Mathematical Theory of Evidence*. Princeton University Press, 1976.

[28] B.R. Smith and J.J. Garcia-Luna-Aceves. Securing the Border Gateway Routing Protocol. In *Proceedings of Global Internet 1996.* London, UK. November 1996.

[29] B.R. Smith, S. Murphy, and J.J. Garcia-Luna-Aceves. Securing Distance-Vector Routing Protocols. In *Proc. of NDSS'97*, San Diego, USA. Feb 1997.

[30] T. Wan, E. Kranakis, and P.C. van Oorschot. Secure Routing Protocols Using Consistency Checks and S-RIP. Technical Report TR-03-09, School of Computer Science, Carleton University, Ottawa, Canada. Oct 2003.

[31] F.Y. Wang and F.S. Wu. On the Vulnerablity and Protection of OSPF Routing Protocol. In *Proceedings of IEEE Seventh International Conference on Computer Communications and Networks*, Lafayette, LA, USA. Oct 12-15, 1998.

[32] R. White. Securing BGP Through Secure Origin BGP. *The Internet Protocol Journal*, 6(3): 15-22, September 2003.

[33] B. Yu and M.P. Singh. Distributed Reputation Management for Electronic Commerce. In *Computational Intelligence*, 18(4): 535-549, 2002.