Formal Approach for Route Agility against Persistent Attackers

Jafar Haadi Jafarian, Ehab Al-Shaer, and Qi Duan

Department of Software and Information Systems
University of North Carolina at Charlotte
Charlotte, NC, USA
{jjafaria,ealshaer,qduan}@uncc.edu

Abstract. To proactively defend against denial of service attacks, we propose an agile multipath routing approach called random route mutation (RRM) which combines game theory and constraint satisfaction optimization to determine the optimal strategy for attack deterrence while satisfying security, performance and QoS requirements of the network. Our contribution in this paper is fourfold: (1) we model the interaction between RRM defender and DoS attacker as a game in order to determine the parameters by which the defender can maximize her benefit, (2) we model route selection as a constraint satisfaction optimization and formalize it using Satisfiability Modulo Theories (SMT) to identify efficient practical routes, (3) we provide algorithms for sound and smooth deployment of RRM on conventional as well as software-defined networks, and (4) we develop analytical and experimental models to investigate the effectiveness and limitation of RRM under different network and adversarial parameters. Our analysis and preliminary implementation show that RRM can protect up to 90% of flow packets from being attacked against persistent attackers, as compared with single-path routing schemes. Moreover, our implementation shows that RRM can be efficiently deployed on networks without causing any disruption for flows.

1 Introduction

The tragic effect of DoS attacks on networks are significantly aggravated by adoption of conventional least-cost single-path routing schemes. While such route selection simplifies reachability and manageability, it gives adversaries significant advantages to gradually learn network routes and plan DoS flooding attacks accurately. For instance, intruders can disrupt the data session simply by attacking one of the intermediate nodes along the associated route. Such a DoS attack is feasible since only one single predictable route is chosen, and this singularity enables intruders to readily discover the route and devote their resources to attacking it.

In this paper we present a random multi-route approach, called random route mutation (RRM), which protects designated flows by routing them via an optimal number of randomly-chosen routes such that each route satisfies security,

J. Crampton, S. Jajodia, and K. Mayes (Eds.): ESORICS 2013, LNCS 8134, pp. 237–254, 2013.

capacity, overlap and QoS constraints of the network. RRM significantly raises the bar for attackers because to completely compromise the flow, intruders must subvert all the routes and thus require more resources than those needed for attacking a single route. Also, nondeterministic route selection disrupts reconnaissance for attack planning and wastes attacker resources by forcing her to blindly disperse her resources across network routes. Moreover, although routes are chosen randomly, constraint-satisfying route selection guarantees that each route has the desired security and performance-related properties.

We assume a persistent adversarial model where attacker is RRM-aware and aims to defeat RRM by frequent hopping between network routes. The number of hopping (mutation) between routes determines attacker's strategy because the more routes the adversary attacks, the higher the probability of hitting the random routes which are chosen by RRM.

The first challenge of RRM is to determine the optimal number of routes for flow transmission such that the defender's benefit is maximized while making her indifferent to the attacker's strategy. We refer to this problem as *optimal strategy selection* and model it as a static game of complete information between attacker and defender, where players' strategies are defined in terms of number of mutations and their payoffs are defined based on the tradeoff between the benefit and cost of mutation.

Knowing the number of routes, the next challenge is to determine a set of qualified routes such that each route satisfies security and performance constraints of the network. In this paper, we consider the following constraints, but other constraints can be added as well:

- *Capacity constraint*: the routes should not include those nodes that are already overloaded (based on node capacity) or those nodes that do not fulfil the bandwidth requirement of the flow.
- *Overlap constraint*: to increase unpredictability and achieve fair load balancing, the overlap between the routes should be less than the tolerable overlap threshold.
- *Security constraint*: the routes should preserve security enforcement by access control policies such as firewalls; *e.g.*, if a flow must pass through a firewall, the firewall must be included in all the routes.
- *QoS constraint*: the routes should maintain the required quality, such as bounded delays or number of hops.

We refer to this problem as *optimal route selection* and model it as a constraint satisfaction problem using generalized Boolean/arithmetic format of Satisfiability Modulo Theory (SMT). We use SMT solvers to discover a random set of constraint-satisfying routes.

Knowing the set of routes, the final challenge is to design a sound mechanism for route installation and revocation such that mutating from one route to another does not cause any transient or permanent unreachability and the flow is transmitted soundly and without any packet loss. We refer to this problem as *route mutation planning*. We provide a formal algorithm for this problem and prove that it guarantees reachability throughout flow transmission.

While deployment of RRM on conventional network layer architectures is challenging, more recent application-layer architectures such as overlay networks (*e.g.*, RON [2], SOS [4], and VNET/P [17]) and emerging software-defined networking (*e.g.*, OpenFlow [11]) provide promising platforms for RRM. We implemented RRM algorithms in POX [10], a network SDN controller written in Python that communicates with OpenFlow 1.0 switches. In our implementation on SDN, mutation from one route to another is accomplished via a series of flow table updates in all the switches both along the old and new routes.

To evaluate RRM effectiveness, we introduce an analytical metric called MPE (Mutation Protection Effectiveness) which measures average effectiveness of RRM against attackers by taking into account the attacker's strategy and capability. Moreover, we used our implemented framework for extensive evaluation of RRM effectiveness in real-world scenarios. Our analytical and experimental evaluation shows that RRM is significantly effective against DoS attackers.

Previous works on multipath routing in wireless networks such as [16] propose using random forwarding to avoid jamming and blackhole attacks. These works are far from being practical for wired networks because of many topological, QoS and security constraints. Moreover, unlike previous works [16], we do not use random walk heuristic-based algorithms to identify random routes because it is infeasible to design a random walk algorithm to satisfy multiple constraints simultaneously.

The rest of the paper is organized as follows: Section 2 discusses our basic methodology. Section 3 presents implementation details of RRM. Section 4 shows the evaluation results. Section 5 presents related work. Section 6 concludes the paper.

2 Technical Approach

2.1 Adversarial Modeling

RRM effectiveness against static attackers (attackers that do not move) is obviously high. However, to accurately evaluate effectiveness for realistic scenarios, we assume a generalized *persistent* RRM-aware adversarial model. In this model, the attacker is characterized by two parameters: her capability and the number of routes she attacks. Attacker's capability, denoted as r, is defined in terms of the number of nodes that are known to the attacker. Attacker's mutation intervals, denoted as M_a, defines the attacker's strategy in the network. More specifically, at each mutation interval, the attacker uniformly chooses a route and attacks it. If the adversary by chance attacks a route that is being used by RRM, she would stay on the route for as long as RRM continues using the route; that is, until the expiration of defender's mutation interval.

The objective of RRM is to protect a flow f that is being transmitted from a source S to a destination D, such that the portion of the flow that evades the attack is maximized. To distance our model from security through obscurity, we assume that the attacker knows the flow properties including its source and destination, its size and duration, as well as the starting time of its transmission.

2.2 Overview

RRM responsibilities in a network are performed by a RRM controller with privileged accesses to network routers/switches. Alg. 1 defines the main algorithm of this controller. After each mutation interval (T seconds), the algorithm uses *ChangeRoute* to revoke the route r_k and install r_{k+1}. Note that for each route, its reverse route must also be installed. r_k^{-1} denotes the reverse route of r_k. The *ChangeRoute* algorithm is described in Section 2.5.

Algorithm 1. RRM Controller algorithm for route mutation of a flow from S to D

determine optimal defender strategy (M_d^*) by finding NE of the game	▷ Sec. 2.3
determine qualified routes $r_1, \ldots, r_{M_d^*}$ using SMT solver	▷ Sec. 2.4
upon expiration of kth defender mutation interval	
$ChangeRoute(r_k \rightarrow r_{k+1})$	▷ Sec. 2.5
$ChangeRoute(r_k^{-1} \rightarrow r_{k+1}^{-1})$	

2.3 Optimal Strategy Selection

Of fundamental significance is the problem of determining the number of routes that are used for transmitting a flow. Although it is intuitive that using more routes provides higher benefit for the defender on average, it also increases the cost associated with the routing. Therefore, choosing the optimal mutation strategy for the defender partly depends on the benefit-cost tradeoff of the mutation.

In addition to this tradeoff, the defender benefit also depends on the mutation strategy of the attacker. If the defender's mutation rate is slower than that of the attacker's, it is straightforward to see that RRM will be less effective. However, although faster hopping between routes increases the probability of hitting a flow route for the attacker, it also increases detectability of the attacker and her resources. Therefore, the defender and attacker mutation strategies can be defined as a static game of complete information, where each player aims to determine her Nash equilibrium strategy by considering other players' strategies and the cost associated with her own strategy.

The game is defined as $\Gamma = \langle I, S, U \rangle$, where $I = \{a, d\}$ is the set of players, $S = \{M_a, M_d\}$ denotes the set of strategies for the attacker and defender, and $U = \{u_a, u_d\}$ defines the payoff function for each player. Note that the attacker's strategy is defined in terms of the number of routes, M_a, that she attacks during flow transmission. Defender's strategy is defined in terms of number of routes, M_d, that are used for flow transmission.

To evaluate RRM effectiveness against attackers, we define *mutation protection effectiveness* metric (MPE) as the average percentage of the flow that is transmitted without being compromised. Suppose the defender aims to transmit a flow f between a given source and destination and the network consists of n nodes. The flow is transmitted during M_d mutation intervals such that $1/M_d$ portion of f is transmitted during each interval.

To calculate MPE, we first need to calculate node compromise probability, x:

$$x = \frac{r}{n}$$

The probability that a route is compromised is equal to the probability that *at least* one node in the route is compromised by the attacker. Assume L denotes the maximum length of routes in terms of nodes, and p_i denotes the percentage of routes with length i. Assuming *disjointness* between routes (no node is shared between routes), the route expected compromise probability, denoted as X, is:

$$X = \sum_{i=1}^{L} p_i(1 - (1 - x)^i) \tag{1}$$

If $M_a \leq M_d$, the attacker may hit the flow at each interval with probability X. Since route compromise probabilities are disjoint, the number of routes hit by the attacker follows binomial distribution $\sim B(M_a, X)$. Therefore, the average number of routes hit by the attacker is $X \cdot M_a$, and each hit compromises one $1/M_d$ portion of the flow. For this case, MPE is:

$$MPE(M_a, M_d) = 1 - \frac{M_a}{M_d}X$$

For scenarios where $M_a > M_d$, the number of routes hit by the attacker follows binomial distribution $\sim B(M_d, X)$. The average number of routes hit by the attacker is $X \cdot M_d$. However, the exact percentage of the flow hit by the attacker is more complex because the attacker is mutating faster than the defender and she may hit one defender interval (route) after a portion of the flow has been transmitted. Suppose $z = \lceil M_a/M_d \rceil$; *i.e.*, for each defender mutation, the attacker mutates z times (defender is stationary to attacker during these intervals). Based on the adversary model, if the attacker hits a route that is being used, she will remain there until the defender's mutation interval expires. During the ith defender interval, the attacker mutates z times. If the attacker hits the defender's route during the first mutation with probability X, then the whole flow is compromised. The probability that the attacker does not hit the flow during the first mutation, but during the second mutation is $(1 - X)X$ (geometric distribution), and the portion of the flow that is compromised is $\frac{(z-1)/z}{M_d}$. Generally, when $M_a > M_d$ the average percentage of the flow which is compromised during one defender interval is:

$$\sum_{k=1}^{z}(1 - X)^{k-1} \cdot X \cdot (z - k + 1)/z \cdot 1/M_d$$

Therefore, for this scenario MPE is:

$$MPE(M_a, M_d) = 1 - M_d \cdot \left(\sum_{k=1}^{z}(1 - X)^{k-1} \cdot X \cdot (z - k + 1)/z \cdot 1/M_d \right)$$

We can combine both cases into the following formula:

$$MPE(M_a, M_d) = 1 - \min(M_a, M_d) \cdot \left(\sum_{k=1}^{z} (1-X)^{k-1} \cdot X \cdot (z-k+1)/z \cdot 1/M_d \right) \tag{2}$$

where $z = \lceil M_a/M_d \rceil$. For example, for static attackers where $M_a = 1$:

$$MPE(1, M_d) = 1 - \frac{X}{M_d}$$

If both attacker and defender mutate with the same speed M:

$$MPE(M, M) = 1 - X$$

The defender's utility is defined based on the benefit from protecting the flow in terms of MPE and the cost of M_d mutations. Mutation cost emanates from updating routing tables and installing new routes in routers/switches of the network. On the other hand, the attacker's utility is defined based on the benefit from compromise (1 - MPE) and the cost of M_a mutations. The attacker mutation cost originates from the fact that as the attacker increases the number of attacked routes, her detection probability increases. Note that these benefit and cost functions are application-dependent and differ based on the properties of the flow and network. Eq. 3 and 4 denote *generic* utility functions for defender and attacker respectively, where Π denotes the benefit function, Θ denotes the cost function, and N denotes the number of disjoint routes.

$$u_d(M_a, M_d) = \Pi_d(MPE(M_a, M_d)) - \Theta_d(M_d) \tag{3}$$

$$u_a(M_a, M_d) = \Pi_a(1 - MPE(M_a, M_d)) - \Theta_a(M_a) \tag{4}$$

$$M_a, M_d \in (0, N]$$

Since the route compromise probabilities are disjoint, N is the upper bound for both players' strategies. The objective of the game is to determine the Nash equilibrium (NE) strategy profile (M_a^*, M_d^*). Note that if the cost of mutation is 0, both players tend to maximize their mutation. For such scenarios, (N, N) is the Nash equilibrium of the game. Otherwise, both players can deviate by increasing their mutation and achieving higher payoffs (Fig. 3). However, if mutation cost functions are nonzero, then the players' payoffs depend on the trade-off between benefit and cost of mutations. Numerical analysis of the game to determine the pure Nash equilibrium requires $\theta(N^2)$ payoff calculations. If no pure strategy NE exists, we either determine the mixed Nash equilibrium of the game and then randomly choose a strategy according to the distribution, or we assume that the attacker plays $M_a = N$ and determine the M_d that maximizes defender's payoff.

To determine the defender mutation interval, we simply divide flow duration T_f by M_d^*; *i.e.*, $T = T_f/M_d^*$. Flow duration is either provided as input or determined based on flow size and network bandwidth.

2.4 Optimal Route Selection

Route selection is accomplished by formalizing RRM constraints and using off-the-shelf SMT solvers to determine M_d^* qualified routes between the designated source and destination. However, for large M_d^*, the computational complexity, as well as topological limitations, does not allow SMT solvers to determine all the routes at once. Instead, we relax the problem by defining a relatively small window size w such that at each iteration, SMT solver determines w new routes until all M_d^* routes are generated. While computational limitations of SMT solvers necessitate smaller window sizes, overhead resulting from multiple model solving necessitates larger windows. In our approach, we set $w = 10$.

We can model the network as a directed graph $G = (V, E)$, where V is the set of hosts and E is the set of links. Suppose there is a flow with source S and destination D ($S, D \in V$). Also assume the network contains n nodes $v_1, \ldots, v_n$ and m edges $e_1, \ldots, e_m$. The capacity of node v_i is denoted as $C(v_i)$. Moreover, the Boolean variable b_i^k denotes inclusion of node v_i in the kth route: if $b_i^k = 1$, then node v_i is used for the flow; otherwise v_i is not used for the flow. Our objective is to use a SMT solver to find a satisfiable assignment to all the variables b_i^k. The following formalization models the problem of discovering w qualified routes between S and D:

$$b_S^k = 1, b_D^k = 1, 1 \leq k \leq w \tag{5}$$

$$b_i^k = 1 \Rightarrow \sum_{v_j \in \chi(v_i)} b_j^k = 2,\ \forall v_j \text{ except } S \text{ and } D, 1 \leq k \leq w \tag{6}$$

$$\sum_{v_j \in \chi(y)} b_j^k = 1,\ y \in \{S, D\}, 1 \leq k \leq w \tag{7}$$

$$\sum_{1 \leq i \leq n} b_i^k \leq L,\ 1 \leq k \leq w \tag{8}$$

$$b_i^k = 1,\ \forall v_i \text{ contains } \mathbb{A}, 1 \leq k \leq w \tag{9}$$

$$b_i^k = 0,\ \forall C(v_i) \leq B_f, 1 \leq k \leq w \tag{10}$$

$$((b_i^k = 1) \wedge (b_i^l = 1)) \Leftrightarrow \zeta_i^{k,l} = 1),\ \forall i, 1 \leq k, l \leq w, k \neq l \tag{11}$$

$$\eta_{k,l} = \sum_{1 \leq i \leq n} \zeta_i^{k,l},\ 1 \leq k, l \leq w, k \neq l \tag{12}$$

$$\eta_{k,l} \leq L_p,\ 1 \leq k, l \leq w, k \neq l \tag{13}$$

$$b_i^k, \zeta_i^{k,l} \in \{0, 1\},\ \forall i, k, l \tag{14}$$

Eq. 5 guarantees that the source and destination of each route are S and D. Eq. 6 guarantees that each intermediate node of each route is adjacent to exactly two nodes in the route. This also disallows inclusion of cycles in the routes. Moreover, Eq. 7 states that S and D are only adjacent to only one node in each route.

Eq. 8 (*QoS constraint*) guarantees that the length of the route does not exceed L. Note that we assume a uniform delay for each network link.

Eq. 9 (*Security constraint*) guarantees that the route must pass through the nodes that contain required access control devices (such as firewalls), which are denoted as $\mathbb{A}$.

Eq. 10 (*Capacity constraint*) guarantees that the route should avoid the nodes that do not have the capacity that is required by the flow (denoted as B_f).

Eq. 11, Eq. 12 and Eq. 13 (*Overlap constraint*) guarantee that any two routes in the w intervals will have the maximum number of overlapping nodes L_p. More specifically, Eq. 11 defines parameter $\zeta_i^{k,l}$ such that $\zeta_i^{k,l} = 1$ if node v_i is shared between kth and lth routes. Eq. 12 counts the number of overlapping nodes between the two routes and denotes it as $\eta_{k,l}$. Finally, Eq. 13 guarantees that the number of overlapping routes does not exceed the threshold L_p; *i.e.*, $\eta_{k,l} \leq L_p$. Eq. 14 specifies the value range of the variables.

If SMT solver fails to find any satisfiable assignment, we will relax the constraints (*e.g.*, increase L_p in Eq. 13, or decrease w) and solve the model again. Note that in this paper, we only consider RRM for a single flow. However, RRM for multiple flows can be defined similarly. In this case one needs to find the routes for every flow and there may be additional constraints that are related to the priority of the flows.

2.5 Route Mutation Planning

Given M_d^* routes, the objective of route mutation planning is to ensure end-to-end reachability throughout flow transmission. To achieve this objective, we must ensure that at any point during transmission all routers know how to forward the incoming flow packets toward the destination. More specifically, we must ensure that any mutation from the old route r_o to the new route r_n does not cause unreachability. Alg. 2 describes a route management algorithm that guarantees end-to-end reachability.

Theorem 1. *Alg. 2 guarantees sound and lossless flow transmission.*

Proof. Assume Alg. 2 does not guarantee lossless flow transmission. This implies that there exists a router rt that fails to forward the flow packets at some point. All network routers can be categorized into four classes based on their inclusion or exclusion in r_o and r_n.

- $rt \notin r_n \wedge rt \notin r_o$: such routers will never receive any flow packets, because no router will ever have any rule to forward flow packets to them.
- $rt \in r_n \wedge rt \notin r_o$: the router will not receive any packet before r_n entries are added because no router in r_o will forward any flow packet to them. Afterwards, the router will forward the flow packets soundly.
- $rt \in r_n \wedge rt \in r_o$: the router will forward packets soundly, either based on r_o or r_n entries.

– $rt \notin r_n \wedge rt \in r_o$: the latest time that rt may receive a packet after r_n is activated will less than the round-trip time between the source and destination. Before this time, rt will forward the packets soundly. Afterwards, the router will not receive any flow packets.

Therefore, none of the routers will fail to forward the flow packets, resulting in a contradiction.

Algorithm 2. route mutation planning algorithm

function CHANGEROUTE($r_o \rightarrow r_n$)
add entries for all routers rt s.t. $rt \in r_n \wedge rt \notin r_o$
modify entries for all routers rt s.t. $rt \in r_n \wedge rt \in r_o$
wait for one RTT
delete entries for all routers rt s.t. $rt \in r_o \wedge rt \notin r_n$

3 Implementation

Implementation of RRM on conventional networks can be done by installing static route entries in the routing tables of the corresponding routers. For example, to configure static routes in the Cisco routers, the administrator can specify the exact routing entry by using the command "ip route". The administrator can also define the priority of the static entry (also called administrative distance) to override the dynamic route entries. Route selection and mutation planning will be performed by the central controller which has privileged access to all routers in the network. Flow and network attributes are provided as input parameters to the controller via a designated interface. The controller (1) determines M_d^* and T by determining the game equilibrium, (2) uses a SMT solver such as Z3 [12] to determine the set of routes, and (4) uses its privileged access to update the routing entries for each mutation interval according to Alg. 2.

Thorough evaluation of RRM effectiveness and overhead requires its deployment in large-scale networks with random topologies. To this aim, we deployed RRM on a software-defined network (SDN). In SDN, the network controller monitors and controls the entire network from a central vantage point via an interface, such as OpenFlow [11]. Due to flexibility and programmability of network switches in software-defined networks, mutation from one route to another can be accomplished as a series of flow table updates in all the switches both along the old and new routes.

We used Mininet [5] python libraries to develop a random topology generator that constitutes large-scale software-defined networks with various edge distribution models. The network is managed by a python POX [10] controller. The POX controller acts as the central authority to manage route mutation in switches. Optimal route selection is performed using Z3 [12] binding to Python. Our prototype implementation shows that route mutation in SDN can be deployed soundly and without packet loss.

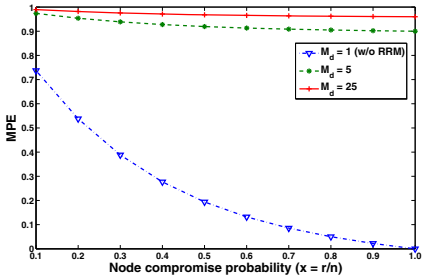

Fig. 1. MPE for static attackers for various r and M_d

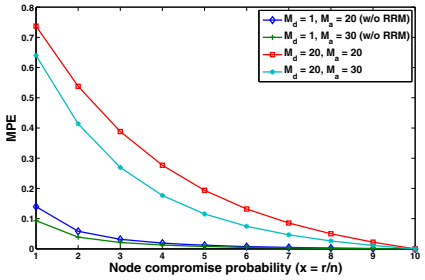

Fig. 2. MPE for static defenders (no RRM) for various r, M_a, and M_d

4 Evaluation

We evaluate effectiveness and overhead of RRM through theoretical and experimental analysis.

4.1 Effectiveness

Expected Theoretical Effectiveness. In Section 2.3 we define our analytical evaluation metric, called MPE that denotes the average theoretical effectiveness of RRM against persistent attackers in terms of the average percentage of the flow that is transmitted without being compromised. Although analytical MPE is defined based on the assumption that routes are disjoint ($L_p = 0$ in Eq. 13), it provides an accurate approximation of RRM effectiveness in random topologies.

Fig. 1 shows effectiveness of RRM against static attackers with different capabilities. Note that (1) RRM is significantly effective against static attackers, and (2) increasing M_d (the defender mutation speed) slightly improves RRM effectiveness against static attackers.

Fig. 2 compares effectiveness of RRM against persistent attackers in the non-RRM network. Non-RRM network is a network where $M_d = 1$; *i.e.*, the defender is not mutating. Note that (1) persistent attacks on non-RRM networks are very

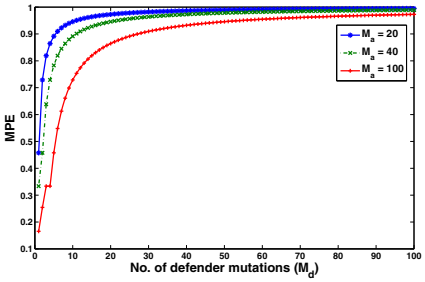

Fig. 3. MPE for various M_d and M_a

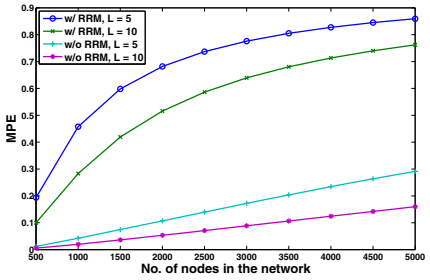

Fig. 4. MPE for various n and L

disruptive, and (2) as the number of defender's mutation intervals approaches that of attacker's, RRM effectiveness is improved.

Fig. 3 compares RRM effectiveness for various attacker and defender mutation interval lengths. Note that as the ratio of M_d over M_a increases, MPE approaches 1. However, both M_d and M_a cannot theoretically exceed the number of node-disjoint routes, which is limited for practical networks [18].

Fig. 4 compares the effect of network size (n), and the route length L on MPE in non-RRM and RRM ($M_d = M_a$) networks with the fixed attacker capability $r = 250$. Note that as the network size increases, the node compromise probability decreases which improves MPE. Also the advantage of RRM over non-RRM gradually decreases with the increase of n. This is because for large non-RRM networks, the attacker needs longer time to hit the route.

Theoretical Effectiveness for Threshold-Critical Flows. Certain classes of flows such as Shamir's threshold k-out-of-n secret sharing scheme [15] require threshold-critical effectiveness; *i.e.*, the flow transmission is successful as long as less than a certain percentage of flow packets are compromised.

For a flow that can tolerate up to l route (*i.e.*, interval) compromises, MPE^l denotes the probability that *at most* l intervals are compromised by the attacker. Note that l is an application-dependent input parameter, which is determined based on sensitivity and criticality of the flow. If each route is compromised

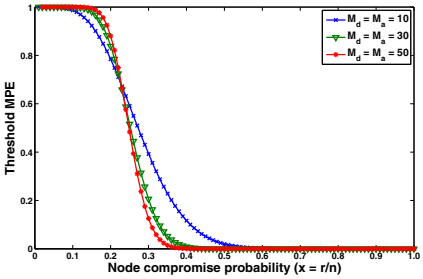

Fig. 5. Threshold MPE ($l = M_d/4$) for various r

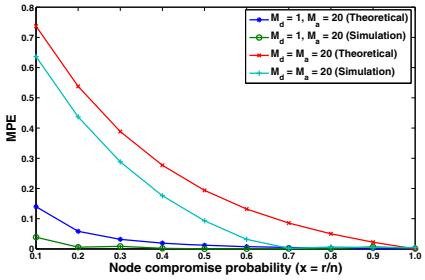

Fig. 6. Comparison of analytical and experimental MPE for various r and M_d

independently of other routes (routes are disjoint) and both players are mutating with the same rate (*i.e.*, $M_a = M_d$), the probability that exactly i mutation intervals are hit is denoted as the random variable Z and follows binomial distribution $Z \sim B(M_d, X)$ [16]. Accordingly, MPE^l can be defined as:

$$MPE^l = P(Z \leq l) = \sum_{i=0}^{l} \binom{M_d}{i} \cdot (X)^i \cdot (1-X)^{M_d - i} \tag{15}$$

Also, it is straightforward to show that $E(MPE^l) = 1 - X$:

$$\begin{aligned} E(MPE^l) &= \frac{1}{M_d} \sum_{i=0}^{M_d} P(Z \leq i) = \\ &= \frac{1}{M_d}(M_d - P(Z=1) - \ldots iP(Z=i) - \ldots - M_d P(Z=M_d) \\ &= \frac{1}{M_d}(M_d - E(Z)) = \\ &= 1 - X \end{aligned} \tag{16}$$

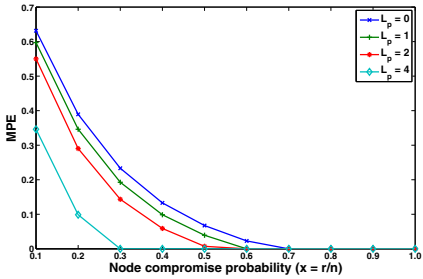

Fig. 7. Experimental MPE for various L_p and r

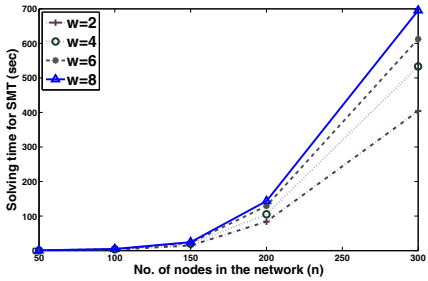

Fig. 8. SMT solving time for different w

which is consistent with Eq. 2. Fig. 5 shows the effect of mutation intervals on threshold MPE. Note that all lines intersect at the point where $l/M = X$; *i.e.*, where the route compromise probability is equal to the tolerable threshold. Moreover, contrary to the average MPE, in cases where $l/M > X$, increasing M_d has a negative effect on the threshold MPE.

Experimental Effectiveness. In practical networks, very few node-disjoint routes can be found for a fixed source and destination [18]. For overlapping routes, the assumption that the compromise probabilities of routes are independent is not valid. Therefore, for random topologies we calculate MPE via experimentation. In order to generate required topologies we developed a random topology generator for Mininet that allows generation of random Mininet networks with n switches and average node degree d according to one of the Erdos-Rnyi (random graph), Barabsi-Albert (scale-free), or Watts and Strogatz (small-world) models.

To generate the ith simulation scenario, n and d are provided to the generator. The generator creates a network by uniformly choosing one of the random graph, scale-free or small-world models. This ensures that the calculated MPE demonstrates the average effectiveness of RRM for various real-world network models. Then, given r the controller determines M_d^* and uses the SMT solver

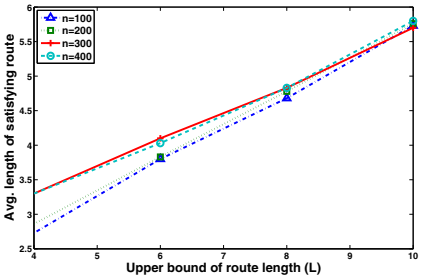

Fig. 9. Average route length for different L and n

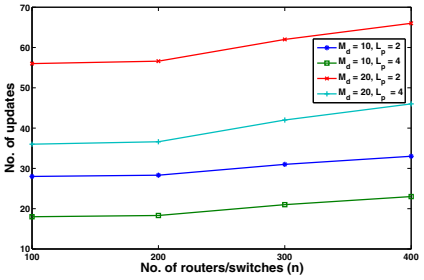

Fig. 10. Average no. of routing table updates for various M_d and L_p

to determine the set of routes. Next, at each mutation interval the controller uses one of these routes for flow transmission. The attacker is simulated in the following way: given r, for each simulation we randomly choose r nodes as the set of nodes known to the attacker. We also assume that the attacker is rational and plays her best strategy M_a^*. At each interval, the attacker uniformly selects one node and attacks it. If this node belongs to any route that is currently being used by RRM, we mark the portion as compromised.

To approximate expected MPE with acceptable accuracy, we use the Monte Carlo method [14]. Suppose random variable Y_i denotes MPE of the ith simulation for $i = 1, \ldots, l$ as iid sequence of samples of MPE. Using the law of large numbers, the approximation of expected MPE is:

$$\widehat{E}(MPE) = \frac{1}{l} \sum_{i=1}^{l} Y_i$$

The estimated magnitude of error for $\widehat{E}(MPE)$ is of order $\sigma_{\hat{MPE}}$. For each simulation scenario, the expected MPE is approximated by repeating the simulation until the error falls below the threshold.

Fig. 6 compares analytical and experimental MPE for random networks with $n = 1000$, $d = 5$ and $L = 5$. Note that analytical MPE serves as an upper bound

for RRM effectiveness. Also note that expected MPE of the flow approaches 0 when node compromise probability approaches 1. This means that if an adversary is highly persistent and highly capable, RRM will lose its effectiveness.

Fig. 7 compares the experimental MPEs for various overlap constraints. Note that for $L_p = 0$, the experimental MPE is consistent with analytical MPE. Moreover, as L_p increases, MPE decreases significantly. This is because as the overlap between the routes increases, attack on a node has higher probability of compromising more than one route.

4.2 Overhead Evaluation and Limitations

Alg. 1 describes the general outline of the RRM controller algorithm. The complexity and computational overhead of each step is as follows:

- Optimal strategy selection: numerical calculation of pure Nash equilibrium requires at most $\theta(N^2)$ steps.
- Optimal route selection: Satisfiability problem is NP-complete in general. However, recent advances in SMT solvers have made them scalable to satisfiability problems with thousands of variables. Fig. 8 shows the time of SMT solving for optimal route mutation on a machine with Quad Core processor (3.3GHz, 6M cache) and 4 GB DDR3 RAM. We can see that the SMT solving time increases with the network size n, especially when the number of switches/routers in the network reaches 300. This is also because the number of possible routes increases exponentially with the size of the network. This has a negative effect on scalability of RRM. However, (1) RRM is used to protect the designated flows, and normal traffic is routed via conventional protocols, and (2) instead of using one centralized controller, the RRM responsibilities can be distributed among several cooperating controllers.
- Route mutation planning: the *RouteChange* algorithm installs a new route in $O(n)$. The upper bound for the number of routing table updates is $O(M_d^* L)$. However, the accurate number of updates depends on the average route lengths and the average number of overlaps between routes. Fig. 9 shows the average length of the route found by the SMT formalization for random networks with different sizes and different length upper bounds. We can see that the average route length of the RRM algorithm converges to some value with the increase of network size. Fig. 10 shows the experimentation results for the average number of routing table updates (flow entries in SDN) in networks with different number of mutation intervals and different overlap upper bounds. In this figure, $L = 6$. Note that (1) higher mutation speeds requires higher number of updates, (2) higher overlaps between routes reduces the number of updates, and (3) although the number of updates increases linearly with the network size, but since route lengths are upper bounded the linear line has a mild upward slope.

5 Related Works

Applying multipath routing in computer networks had been proposed as early as 1970s, but the original purpose is mainly for load balancing. The protocols such as Split Multiple Routing (SMR) [1], multipath DSR [3], AOMDV [8], and AODVM [18] try to find disjoint paths in routing. However, in practical networks, the number of disjoint paths is usually very small [18].

Other protocols try to improve security through multipath routing such as SPREAD [7], SRP [13], SecMR [9], DSM [6]. The route selection in these protocols is deterministic. This means if the attacker knows the algorithm, the routes can be predicted.

The multipath algorithm in [16] generates randomized multipath routes that are also highly dispersive and energy efficient in wireless sensor networks. The algorithm is also based on random walk and its variants and the generated multipath routes are highly resilient to black hole attacks.

Unlike previous approaches, our work provides an automated, nondeterministic, and optimal approach to route mutation problem by formalizing the strategy selection based on game-theoretic concepts, and formalizing route selection as a constraint satisfaction problem with various operational, QoS and security constraints. Moreover, in our approach the route selection is random and designed to counter persistent and informed adversaries.

6 Conclusion

In this paper, we present RRM as a proactive defense strategy against DoS attackers. To the best of our knowledge, RRM is the first proposed technique that offers an efficient practical random route mutation which considers flow, network and security constraints as well as attacker's capabilities and strategies. Our analysis and preliminary implementation show that RRM is feasible and flexible, guarantees end-to-end reachability and can decrease the percentage of disrupted packets to less than 10% of the case without RRM.

One drawback of RRM is its limited scalability due to the centralized control as well as the overhead raised from solving the SMT model for large networks. For future work, we plan to investigate how several controllers can interact to improve the scalability of RRM. Solutions include separating the route selection and the route planning, or dividing the network into several segments each managed by a separate controller.

References

1. Lee, S.-J., Gerla, M.: Split multipath routing with maximally disjoint paths in ad hoc networks. In: IEEE International Conference on Communications, ICC 2001, vol. 10, pp. 3201–3205 (2001)
2. Andersen, D., Balakrishnan, H., Kaashoek, F., Morris, R.: Resilient overlay networks. In: Proceedings of the Eighteenth ACM Symposium on Operating Systems Principles, SOSP 2001, pp. 131–145. ACM, New York (2001)

3. Johnson, D.B., Maltz, D.A., Broch, J.: DSR: the dynamic source routing protocol for multihop wireless ad hoc networks. In: Ad Hoc Networking, pp. 139–172. Addison-Wesley, Boston (2001)
4. Keromytis, A.D., Misra, V., Rubenstein, D.: SOS: an architecture for mitigating ddos attacks. IEEE Journal on Selected Areas in Communications 22(1), 176–188 (2004)
5. Lantz, B., Heller, B., McKeown, N.: A network in a laptop: rapid prototyping for software-defined networks. In: Proceedings of the Ninth ACM SIGCOMM Workshop on Hot Topics in Networks, Hotnets 2010, pp. 19:1–19:6. ACM, New York (2010)
6. Lee, P., Misra, V., Rubenstein, D.: Distributed algorithms for secure multipath routing in attack-resistant networks. IEEE/ACM Transactions on Networking 15(6), 1490–1501 (2007)
7. Lou, W., Liu, W., Fang, Y.: SPREAD: enhancing data confidentiality in mobile ad hoc networks. In: IEEE INFOCOM, pp. 2404–2413 (2004)
8. Marina, M., Das, S.: On-demand multipath distance vector routing in ad hoc networks. In: Proceedings of IEEE International Conference on Network Protocols, ICNP, pp. 14–23 (2001)
9. Mavropodi, R., Kotzanikolaou, P., Douligeris, C.: SecMR - a secure multipath routing protocol for ad hoc networks. Ad Hoc Networks 5(1), 87–99 (2007)
10. OpenFlow group at Stanford University: POX Wiki (2013), `https://openflow.stanford.edu/display/ONL/POX+Wiki`
11. McKeown, N., Anderson, T., Balakrishnan, H., Parulkar, G., Peterson, L., Rexford, J., Shenker, S., Turner, J.: Openflow: enabling innovation in campus networks. ACM SIGCOMM Computer Communication Review 38(2), 69–74 (2008)
12. Microsoft: Z3: An Efficient Theorem Prover (2012), `http://research.microsoft.com/en-us/um/redmond/projects/z3/`
13. Papadimitratos, P., Haas, Z.J.: Secure routing for mobile ad hoc networks. In: SCS Communication Networks and Distributed Systems Modeling and Simulation Conference, San Antonio, TX, USA, pp. 193–204 (2002)
14. Robert, C.P., Casella, G.: Monte Carlo Statistical Methods, 1st edn. Springer (1999)
15. Shamir, A.: How to share a secret. Commun. ACM 22(11), 612–613 (1979)
16. Shu, T., Krunz, M., Liu, S.: Secure data collection in wireless sensor networks using randomized dispersive routes. IEEE Transactions on Mobile Computing 9(7), 941–954 (2010)
17. Xia, L., Cui, Z., Lange, J.R., Tang, Y., Dinda, P.A., Bridges, P.G.: VNET/P: bridging the cloud and high performance computing through fast overlay networking. In: Proceedings of the 21st international symposium on High-Performance Parallel and Distributed Computing, pp. 259–270. ACM Press, New York (2012)
18. Ye, Z., Krishnamurthy, S.V., Tripathi, S.K.: A framework for reliable routing in mobile ad hoc networks. In: IEEE INFOCOM, pp. 270–280 (2003)

Appendix: Table of Parameters

Table 1. Description of main parameters

B_f	capacity required by the flow
b_i^k	variable denoting whether node v_i belongs to the kth route
$C(v_i)$	capacity of the node i
f	flow
L	maximum route length
L_p	upper bound for number of overlapping nodes between the routes
M_a	attacker strategy: no. of attacker's mutations
M_d	defender strategy: no. of defender's mutations
N	average no. of routes between a given source and destination
n	no. of nodes in the network
r	no. of network nodes known to attacker
S	source or sender of the flow
D	destination or receiver of the flow
p_i	percentage of routes with length i
T_f	duration of the flow f
x	node compromise probability ($x = r/n$)
X	route compromise probability
z	ratio of attacker to defender mutations $z = \lceil M_a/M_d \rceil$
v_i	network node
u_a	attacker's payoff function
u_d	defender's payoff function
$\eta_{k,l}$	variable denoting number of shared nodes between kth and lth routes
$\mathbb{A}$	nodes that include access control devices
$\chi(v_i)$	the set of neighbors of node v_i
$\mathit{\Pi}$	the benefit function
Θ	the cost function