Hindawi Publishing Corporation
EURASIP Journal on Wireless Communications and Networking
Volume 2009, Article ID 946493, 13 pages
doi:10.1155/2009/946493

Research Article

On Multipath Routing in Multihop Wireless Networks: Security, Performance, and Their Tradeoff

Lin Chen and Jean Leneutre

Department of Computer Science and Networking, LTCI-UMR 5141 laboratory, CNRS-Telecom Paris Tech, 46 Rue Barrault, 75013 Paris, France

Correspondence should be addressed to Lin Chen, lchen@enst.fr

Received 29 January 2009; Accepted 1 June 2009

Recommended by Hui Chen

Routing amid malicious attackers in multihop wireless networks with unreliable links is a challenging task. In this paper, we address the fundamental problem of how to choose secure and reliable paths in such environments. We formulate the multipath routing problem as optimization problems and propose algorithms with polynomial complexity to solve them. Game theory is employed to solve and analyze the formulated multipath routing problem. We first propose the multipath routing solution minimizing the worst-case security risk (i.e., the percentage of packets captured by attackers in the worst case). While the obtained solution provides the most security routes, it may perform poorly given the unreliability of wireless links. Hence we then investigate the multipath routing solution maximizing the worst-case packet delivery ratio. As a natural extension, to achieve a tradeoff between the routing security and performance, we derive the multipath routing protocol maximizing the worst-case packet delivery ratio while limiting the worst-case security risk under given threshold. As another contribution, we establish the relationship between the worst-case security risk and packet delivery ratio, which gives the theoretical limit on the security-performance tradeoff of node-disjoint multipath routing in multihop wireless networks.

1. Introduction

It is widely recognized that the intrinsic nature of wireless networks, such as the broadcast nature of the wireless channel and the limited resources of network nodes, makes them extremely attractive and vulnerable to attackers. Routing amid malicious attackers in such environments is a challenging task. On one hand, the most secure route(s) should be chosen such that the percentage of packet captured by attackers is as small as possible. On the other hand, given the unreliability of wireless links, the most reliable route(s) should be selected such that the packet delivery ratio at destination is as high as possible.

A natural approach is to use multiple paths to increase the fault tolerance and the resilience to attackers. However, how to choose the secure and reliable paths among exponentially many candidates and how to allocate traffic among them remain a difficult but crucial problem.

1.1. Paper Overview. In this paper, we address the above fundamental routing problem by focusing on two metrics: route security and performance. We start with the single-attacker case and extend our work to the multiple-attacker case in Section 7.

We first study the multipath routing solution minimizing the worst-case security risk; that is, the percentage of packets captured by the attacker under the condition that the attacker makes all its efforts to maximize this percentage. We model such multipath routing problem as a minimaximization problem and formulate it as the maximum flow problem in lossy networks based on which a routing algorithm with polynomial time complexity being derived to solve it.

While the obtained solution provides the most security routes, which is crucial for security sensitive applications, performance is another important issue that definitively cannot be ignored, especially in wireless networks with unreliable links. To this end, we investigate the multipath routing solution maximizing the packet delivery ratio under the condition that the attacker makes all its efforts to minimize this ratio. Noticing that solving this problem requires exponential time complexity, we propose a heuristic algorithm computing the optimal path set with polynomial time complexity. In our study, we also apply game theory as a systematic tool to solve and analyze the formulated multipath routing problems.

Next, we extend our efforts to study a natural problem: how to achieve a tradeoff between the route security and performance. In this perspective, we derive the routing solution maximizing the worst-case packet delivery ratio while limiting the worst-case security risk under given threshold. Furthermore, as a theoretical limit on the security-performance tradeoff of node-disjoint multipath routing, we establish the relationship between the worst-case packet delivery ratio a^* and the security risk r^*:

$$a^* \leq r^* \left(\left| \mathcal{P}^{\mathrm{nd}} \right| - 1 \right), \tag{1}$$

where $|\mathcal{P}^{\mathrm{nd}}|$ is the maximum number of node-disjoint paths in the network.

By simulation, we evaluate the performance of the proposed multipath routing protocols. The results show that our solutions show the best worst-case security and performance among the simulated multipath routing protocols.

1.2. Background and Motivation. Multipath routing, as mentioned above, is a promising way to improve route reliability and security. Past work on multipath routing in wireless networks mainly consists of evaluating the possible paths via reputation metrics based on security or reliability and distributing traffic among the routes with the highest reputation ratings.

In [1], Papadimitratos et al. proposed an algorithm, called Disjoint Path-set Selection Protocol (DPSP), to find the maximum number of paths between a source and destination with the highest reliability. DPSP tries to find maximum number of node-disjoint paths based on the reliability metric to improve the reliability of communication by increasing the number of used paths.

In [2], Lou et al. proposed another solution for calculating the maximum number of the most secure paths called Security Protocol for REliable dAta Delivery (SPREAD). Their solution relies on previous knowledge of security level of each node and calculates the link costs according to them. It also exploits secret sharing to spread data over multiple paths and proposes a security-optimized share allocation method.

In [3], Papadimitratos and Haas proposed and analyzed a routing protocol named Secure Message Transmission Protocol (SMT) which improves security and reliability of data transmission through diversity coding of data into multiple symbols and transmitting each symbol over one path by uniform loading. SMT employs a rating mechanism to select the most reliable paths based on end-to-end feedback.

Our work in this paper differs with existing work in that we base our work on the worst-case scenarios and provide multipath routing solutions with guaranteed security and performance properties. Our motivation is twofold: first, in most of the proposed solutions, each path is rated according to its past performance, and the paths with high rate are selected to carry traffic. In such reputation-based mechanism, the computation of the reputation rates is not trivial at all; furthermore, this mechanism may fail to provide good paths when facing strategic attackers. For example, assume that three paths are available and each time the two paths with the highest rates are selected. A strategic attacker can itself do the same rating estimation and attack the two paths with the highest rate. The problem is that the rating mechanism implicitly assumes that there exists correlation between the history and future performance. With this correlation, one can predict the attacker's action to some extent. Unfortunately, a strategic attacker will certainly not take predictable actions. Instead, in some cases it can even take the advantage of the rating mechanism to cause more severe damage to the networks. Motivated by the above observation, we believe that it is crucial to study multipath routing solutions with guaranteed worst-case security and performance properties, which is the focus of our work.

In terms of the underlying methodology, our work is also related to the min-max optimization and routing games [4–7]. In fact, our work can be seen as the application of this tools in hostile wireless networks with unreliable/lossy links absent in classical context which pose significant difficulties in solving the problem, as shown in later sections.

2. System Model and Assumptions

In our work, we consider a multihop wireless network, modeled as a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with n nodes and m edges. For the wireless links, we consider a model in which any link is either "good" (i.e., error-free) or "bad" otherwise. We refer to the probability that link $e \in \mathcal{E}$ is "good" as the reliability factor of e, denoted by r_e. We assume that different links are independent. (This assumption holds in the case where different wireless links use channels that are well separated in time and frequency via the MAC protocol or some channel coordination mechanism. The extension of our analysis to alleviate this assumption to consider the correlated-link case (the correlation between wireless links highly depends on the underlying MAC protocol) is left for future work.)

We consider a data session between a single source S and destination T. S routes its packets along path $P_i \in \mathcal{P}$ (let $\mathcal{P}$ be the set of paths between S and T) with probability q_i. An attacker M attacks the node $v \in \mathcal{V} \setminus \{S, T\}$ with probability p_v to disrupt the communication between S and T. (We assume that S and T are not attacked by M during the communication. Multiple-attacker case is discussed in Section 7.) If node v is attacked, all the traffic passing by it is captured by M during the attack period.

In this paper, we assume that each node knows the link reliability factors $\{r_e\}$. References [8, 9] address the issue of how to estimate and collect this information. We also assume that each node has the knowledge of network topology.

This information can be acquired from any secure link-state routing protocol, for example, [10]. These assumptions allow us to concentrate on the essential theoretical properties of the multipath routing problem and the resulting solutions. In the case where link reliability factors and network topology change frequently, the update of the multipath set should be performed periodically or triggered by the change.

3. Multipath Routing with Minimum Worst-Case Security Risk

In this section, we study the multipath routing solution minimizing the worst-case security risk. We quantify the worst-case security risk by the percentage of packets captured by the attackers under the condition that the attackers make all their efforts to maximize this percentage (or equivalently, the probability that a packet is captured by the attackers under the condition that the attackers make all their efforts to maximize this probability). We start with the case of single attacker M. In such a routing problem, the objective of S is to calculate $\mathbf{q} = \{q_i\}$ to minimize the maximum security risk caused by M. Mathematically, the multipath routing problem can be formulated as the following minimaximization problem $\mathbf{MP_1}$:

$$\begin{gathered} r^* = \min_{\mathbf{q}} \max_{\mathbf{p}} \sum_{v\in\mathcal{V}} \left[\sum_{v\in P, P\in\mathcal{P}} q(P)\tau(P,v)\varphi(P,v) \right] p_v \\ \text{Subject to} \sum_{v\in\mathcal{V}} p_v \le 1, \quad p_v \ge 0, \ \forall v \in V \\ \sum_{P\in\mathcal{P}} q(P) = 1, \quad q(P) \ge 0, \ \forall P \in \mathcal{P}, \end{gathered} \tag{2}$$

where $\tau(P,v) = \prod_{e\in P, e\succ v} r_e$, $\varphi(P,v) = \prod_{b\in P, b\succ v}(1-p_b)$. $a \succ b$ denotes that packets encounter node/edge a before node/edge b when routed along P. $r = \sum_{v\in\mathcal{V}}[\sum_{v\in P,P\in\mathcal{P}} q(P)\tau(P,v)\varphi(P,v)]p_v$ is the expected probability that the packet is captured by M. Let $r' = \sum_{v\in\mathcal{V}}[\sum_{v\in P,P\in\mathcal{P}} q(P)\tau(P,v)]p_v$. If M attacks at most one node per path, then $r = r'$. In general case, it always holds that $r \le r'$. Noticing that $\mathbf{MP_1}$ is a nonlinear optimization problem, we focus on solving $\mathbf{MP_1'}$:

$$(r')^* = \min_{\mathbf{q}} \max_{\mathbf{p}} r', \tag{3}$$

which is a linear optimization problem. Later in Section 3.2 we will show that $r^* = (r')^*$.

Consider the inner maximization problem of $\mathbf{MP_1'}$ for fixed $\mathbf{q}$:

$$\begin{gathered} \max_{\mathbf{p}} \sum_{v\in\mathcal{V}} \left[\sum_{v\in P, P\in\mathcal{P}} \tau(P,v)q(P) \right] p_v \\ \text{Subject to} \sum_{v\in\mathcal{V}} p_v \le 1, \quad p_v \ge 0, \ \forall v \in \mathcal{V}. \end{gathered} \tag{4}$$

Associating a dual variable y, we obtain the following dual optimization problem:

$$\begin{gathered} \min \ y \\ \text{Subject to } y \ge \sum_{v\in P, P\in\mathcal{P}} \tau(P,v)q(P), \quad \forall v \in \mathcal{V}. \end{gathered} \tag{5}$$

Substituting this minimization problem in $\mathbf{MP_1'}$ leads to the following linear optimization problem $\mathbf{LP_1'}$:

$$\begin{gathered} \min \ y \\ \text{Subject to} \sum_{v\in P, P\in\mathcal{P}} \tau(P,v)q(P) \le y, \quad \forall v \in \mathcal{V}, \\ \sum_{P\in\mathcal{P}} q(P) = 1, \quad q(P) \ge 0, \ \forall P \in \mathcal{P}. \end{gathered} \tag{6}$$

The size of $\mathbf{LP_1'}$ grows with the number of possible paths between S and T and can be exponentially large. For this reason we reformulate $\mathbf{LP_1'}$ as the maximum flow problem in lossy networks which can be solved in a polynomial number of steps.

In $\mathbf{LP_1'}$, we can interpret $q(P)$ as a flow on P and y as the capacity of node v. Thus the constraint $\sum_{v\in P,P\in\mathcal{P}} \tau(P,v)q(P) \le y$ restricts the flow on node v. The constraint $\sum_{P\in\mathcal{P}} q(P) = 1$ states that one unit of flow is sent from S to T. Assume that the capacity of each node v in the network is 1. $\mathbf{LP_1'}$ equals to determine the smallest scaling factor y on the network nodes such that one unit of flow can be sent from S to T. In this way $\mathbf{LP_1'}$ can be mapped to the *maximum flow* problem.

Here we would like to emphasize that the maximum flow problem in our context differs from the classical maximum flow problem due to the packet loss factor $\tau(P,v)$. Indeed our problem can be seen as the maximum flow problem in lossy networks [11]. Each link has unlimited capacity $+\infty$, but has a reliable factor r_e. If $r_e = 1, \text{for all}\, e \in \mathcal{V}$, our problem degenerates to the standard maximum flow problem with node capacity constraint.

3.1. Solving the Multipath Routing Problem.

3.1. Solving the Multipath Routing Problem. We first give the stretch of the solution.

(i) Perform *node splitting* to transform the maximum flow problem with node capacity constraint into the maximum flow problem with link capacity constraint.

(ii) Calculate the maximum flow f^* in the transformed network after the node splitting procedure. Decompose the maximum flow into subflow on paths P_1, P_2,..., P_l from S to T with flow f_i on P_i, respectively.

(iii) S should route its packets along path P_i with probability $q_i = f_i/f^*$ to minimize the security risk. The minimum security risk r^* is $1/f^*$.

(iv) Perform the inverse procedure of node splitting. Map the paths and flows in transformed graph into the correspondent paths and flows in the original graph.

In the following, we detail the core part of the solution.

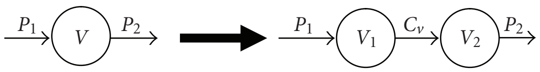

Figure 1: Node splitting.

3.1.1. Node Splitting. The objective of *node splitting* is to transform the maximum flow problem with node capacity constraint into the standard maximum flow problem with link capacity constraint. The key idea is to replace a node with capacity c with two virtual nodes with a link of capacity c between them. The detailed transformation procedure is as follows.

(i) Split each node $v \in \mathcal{V}$ of capacity c_v into two virtual nodes v_1 and v_2. Add a link (v_1, v_2) with the same capacity c_v and the reliable factor 1.

(ii) For each link $(v, v') \in \mathcal{E}$ of reliability p, replace (v, v') by a link (v_2, v') with the same reliability p and the capacity $+\infty$. For each link $(v'', v) \in \mathcal{E}$ of reliability p, replace (v'', v) by a link (v, v_1) with the same reliability p and the capacity $+\infty$.

Figure 1 illustrates the node splitting procedure. After the procedure, node v_1 receives all the input flows of node v; the output flows of node v are sent by the node v_2; the added virtual link (v_1, v_2) carries the flow from input to the output which is restricted by its capacity c_v. Let $\mathcal{G}'$ denote the resulting network after applying the node splitting process on the original network $\mathcal{G}$. It is clear that each flow in $\mathcal{G}$ is one-to-one mapped into a flow with the same quantity in $\mathcal{G}'$. Hence it holds that f^* is the maximum flow in $\mathcal{G}$ if and only if f^* is the maximum flow in $\mathcal{G}'$.

3.1.2. Finding Maximum Flow. Our discussion in this subsection relies on the maximum flow problem in lossy networks. Given a lossy network, the maximum flow problem is to determine the maximum flow that can be sent from a source node S to a sink node T subject to the capacity constraints (i.e., each link has flow bounded by the link capacity) [11].

Such maximum flow problem in lossy networks is a generalized case of the classical maximum flow problem. To solve this generalized problem, we run the most improving augmenting path algorithm described in [11], which generalizes the maximum capacity augmenting path algorithm for the traditional maximum flow problem [12].

In Algorithm 1, the augmenting path has a value, defined as the maximum amount of flow that can reach the sink, while respecting the capacity limits, by sending excess from the first node of the path to the sink. A most improving augmenting path is an augmenting path with the highest value. The algorithm repeatedly sends flow along the most improving augmenting paths. Since these may not be the highest gain augmenting paths, this may creates residual flow-generating cycles. After each augmentation, the algorithm cancels all residual flow-generating cycles in CancelCycles(), so that computing the next most improving path can be done efficiently. Intuitively, canceling flow-generating cycles can be interpreted as rerouting flow from its current paths to the highest-gain paths.

```
Input: transformed network $\mathcal{G}'$
Output: maximum flow $f^*$
repeat
    $f \leftarrow$ CancelCycles($\mathcal{G}'$)
    $f^* \leftarrow f^* + f$
    Find a most improving augmenting path $P$ in $\mathcal{G}'$
    Augment flow along $P$ and update $f^*$
until $f^*$ is maximum
```

Algorithm 1: Max-flow: most Improving Augmenting Path.

An efficient algorithm for computing a most improving augmenting path based on Dijkstra's shortest path algorithm is proposed in [12] with time complexity $O(m+n \log n)$ when implemented using Fibonacci heaps. We refer readers to [11] for detailed algorithm and [13] for a completed survey on the generalized maximum flow problem in lossy networks.

3.2. A Game Theoretic Interpretation. In this subsection, to gain a more in-depth insight of the internal structure of the obtained multipath routing solution, we study the multipath routing problem from a game theoretic perspective by modelling it as a noncooperative game between S and M, denoted as G_1. The strategy of S and M is $\mathbf{q}$ and $\mathbf{p}$, respectively. The objective of S is to determine $\mathbf{q}$ to minimize its utility function $U_s = r$, which is the security risk. The objective of M, on the other hand, is to determine $\mathbf{p}$ to maximize its utility function $U_a = r$.

G_1 is a classical two-person zero-sum game with finite strategy set. Following [14, Proposition 33.1], a Nash equilibrium (mixed strategy) is guaranteed to exist. Based on the result on the two-person zero-sum game [14, Proposition 22.2], we have the following theorem on the NE (Nash equilibrium) of the multipath routing game G_1.

Theorem 1. *At the NE of $G_1(\mathbf{p}^*, \mathbf{q}^*)$, it holds that*

$$\begin{aligned} U_s(\mathbf{p}^*, \mathbf{q}^*) &= U_a(\mathbf{p}^*, \mathbf{q}^*) \\ &= \min_{\mathbf{q}} \max_{\mathbf{p}} r = \max_{\mathbf{p}} \min_{\mathbf{q}} r \end{aligned} \tag{7}$$

Theorem 1 shows that the solution of $\mathbf{MP_1}$ is the most secure routing strategy minimizing the security risk. The minimized security risk from S's point is, on the other hand, the upper bound of the payoff that M can get. Hence, at the NE, the two players reach a compromise through self-optimization such that neither has incentive to deviate.

We now investigate the attacker's strategy at the NE. We consider the maximum flow f^* on the lossy network $\mathcal{G}'$ which is obtained from $\mathcal{G}$ applying the node splitting. Let f_e^* be the flow of f^* on the edge e. It follows from [15] that there exists a cut $\mathcal{C}$ separating S and T such that $\sum_{e \in S} f_e^* = \sum_{e \in S} C_e$. In our case, $\mathcal{C}$ consists of a subset of virtual links added in the node splitting process with capacity 1. This can be shown by the fact that the capacity of all other links is $+\infty$. These virtual links correspond to a set of nodes in the original network, denoted as $\mathcal{V}^{\mathcal{C}}$. As a dual part of the maximum flow problem, at the NE, M attacks every node $v \in \mathcal{V}^{\mathcal{C}}$ with probability $1/|\mathcal{V}^{\mathcal{C}}|$ where $|\mathcal{V}^{\mathcal{C}}|$ denotes the cardinality of $\mathcal{V}^{\mathcal{C}}$. At the NE, the probability that a packet passes the node $v \in \mathcal{V}^{\mathcal{C}}$ is $1/f^*$; thus the probability of the packet captured can be computed as

$$r^* = \frac{1}{f^*} \times \frac{1}{|\mathcal{V}^{\mathcal{C}}|} \times \left|\mathcal{V}^{\mathcal{C}}\right| = \frac{1}{f^*}, \tag{8}$$

which confirms the previous analytical results. Furthermore, it follows that at such NE, M attacks at most one node per path. This leads to $r^* = (r')^*$, which justifies our operation of solving $\mathbf{MP}_1^{'}$ instead of $\mathbf{MP}_1$.

3.3. Complexity Analysis. In the solution of the previous multipath routing problem, the complexity of the node splitting and the inverse procedure is $O(n)$. We now investigate the complexity of Algorithm 1 in the following theorem.

Theorem 2. *Let ϵ_0 be the smallest positive number describing all possible values in Algorithm 1; Algorithm 1 terminates within at most $\lfloor \log_{m/(m-1)}(f^*/\epsilon_0)\rfloor + 1$ iterations, where $\lfloor n \rfloor$ denotes the largest integer not larger than n.*

Proof. The key idea of the proof is to notice that the maximum flow in lossy networks can be decomposed into at most m augmenting paths. Algorithm 1 selects the path that generates the maximum amount of excess at the sink. Thus, each iteration captures at least a $1/m$ fraction of the remaining flow. Please refer to appendix for the detail of the proof. □

Note that in Algorithm 1, the time complexity of the CancelCycles subroutine is $O(mn^2 \log(1/\epsilon_0))$ and that of finding the most augmenting path is $O(m + n \log n)$. Generally, ϵ_0 is sufficiently small. The total time complexity of the algorithm is thus $O(mn^2 \log(1/\epsilon_0) \log(f^*/\epsilon_0))$.

In reality, it is often more practical for S to find the quasioptimal solution of $\mathbf{MP}_1$, that is, the flow $\widetilde{f^*} = (1-\epsilon)f^*$ where ϵ is sufficiently small. In such cases, the time complexity of finding $\widetilde{f^*}$ is $O(mn^2 \log(1/\epsilon) \log(f^*/\epsilon))$ applying the proof of Theorem 2. As a result, the proposed solution offers the flexibility for the source node to balance between the time complexity of the algorithm and the optimality of the result by tuning the parameter ϵ.

3.4. Discussion. The multipath routing problem investigated in this section is related to the work of inspection point deployment in [16] and intrusion detection via sampling in [17] which root from the drug interdiction problem. Our work differs from theirs in the following. Firstly, in [16, 17], the strategy of the police and the service provider is to inspect and sample the edges, while in our problem, the attack is on the nodes, which is more efficient from the attacker's point of view. Secondly, in [16, 17], the network is lossless, while we work on the lossy network, which is more adapted for wireless networks where packet loss and link instability is one of the major concerns. Thirdly, since finding the maximum flow in lossy networks is by nature much more complex to solve than in classical lossless networks, we choose a solution providing the flexibility for the source node to balance between the time complexity of the algorithm and the optimality of the result by tuning the parameter ϵ.

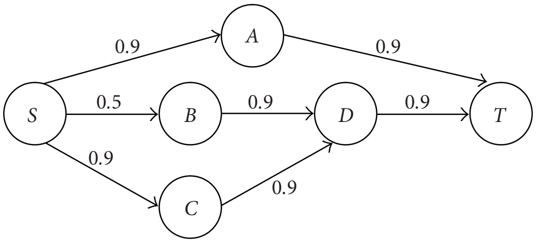

Figure 2: Limitation.

One limitation of the obtained multipath routing solution is that it minimizes the security risk by choosing appropriate multipaths without taking into account the performance of the selected path set. Figure 2 (the number beside the edge is the reliability of the link) provides an illustrative example. Based on the proposed solution, S should select the path SAT and $SBDT$, but it is clear that the path $SCDT$ is more efficient than $SBDT$. The problem is that in previous solution, in some cases, the security is obtained at the price of performance (characterized by the packet delivery ratio). This limitation may pose problem for the applications where the performance of the paths is as important as the security or even more, such as ad hoc networks for emergency rescue. In such scenarios, it is more important for S to find the paths of which the packet delivery ratio at T is maximized even at the presence of M. This motivates us to investigate the multipath routing solution maximizing the worst-case packet delivery ratio. In Section 6, we extend our work to derive the multipath routing solution to achieve a tradeoff between route security and performance.

4. Multipath Routing with Maximum Worst-Case Packet Delivery Ratio

In this section, we study the multipath routing solution to maximize the worst-case packet delivery ratio (or equivalently, the probability that a packet arrives at T under the condition that the attacker makes all its efforts to minimize this probability). In such context, S solves the following maximinimization problem $\mathbf{MP}_2$:

$$\begin{aligned} a^* = \max_{\mathbf{q}} \min_{\mathbf{p}} & \sum_{P \in \mathcal{P}} q(P)\tau(P,T) \prod_{v \in P} (1 - p_v) \\ \text{Subject to } & \sum_{v \in \mathcal{V}} p_v \le 1, \quad p_v \ge 0, \ \forall v \in \mathcal{V}, \\ & \sum_{P \in \mathcal{P}} q(P) = 1, \quad q(P) \ge 0, \ \forall P \in \mathcal{P}, \end{aligned} \tag{9}$$

where $a = \sum_{P\in\mathcal{P}} q(P)\tau(P,T)\prod_{v\in P}(1 - p_v)$ is the expected probability that a packet arrives at T.

4.1. Solving the Maximinimization Problem $\mathbf{MP_2}$.

The maximinimization problems such as $\mathbf{MP_2}$ are usually hard to solve directly. In our study, in order to make the problem more tractable, we apply game theory by modelling the multipath routing problem $\mathbf{MP_2}$ as a game G_2 by following the similar way as in Section 3.2. What differs here is that the objective of S is to maximize its utility function defined as $U_s = a$ and that the objective of M is to minimize $U_a = a$. Following the same argument, the following theorem is immediate.

Theorem 3. *G_2 admits at least one NE $(\mathbf{p}^*, \mathbf{q}^*)$, at which it holds that*

$$\begin{aligned} U_s(\mathbf{p}^*, \mathbf{q}^*) &= U_a(\mathbf{p}^*, \mathbf{q}^*) \\ &= \max_{\mathbf{q}} \min_{\mathbf{p}} a = \min_{\mathbf{p}} \max_{\mathbf{q}} a. \end{aligned} \tag{10}$$

Under the game theoretic formulation, solving $\mathbf{MP_2}$ consists of solving the multipath routing game G_2, more specifically, finding the NE of G_2.

Before delving into the solution, we prove the following useful theorems on the choice of strategy at the NE for the players S and M.

Theorem 4. *There exists an NE where the source node S chooses only node-disjoint paths between S and T.*

Proof. The proof consists of showing that if there exists an NE where S routes its traffic on the paths with common nodes, we can always construct an NE where the source node S chooses only node-disjoint paths. Please refer to appendix for the detailed proof. □

In the following, we focus ourselves on finding the NE with node-disjoint paths.

Theorem 5. *At the NE with only node-disjoint paths, the attacker M attacks at most one node per path.*

Proof. If at such NE, M attacks node $V_1, \dots, V_n$ on the same path P with probability $p_1, \dots, p_n$, then the payoff M gets on the path P is

$$U_P = \tau(P,T)(1 - p_1) \cdots (1 - p_n). \tag{11}$$

If M uses the same resource to attack only one node on P, say V_1, then the payoff it gets on P is

$$U'_P = \tau(P,T)(1 - p_1 - \cdots - p_n) < U_P \tag{12}$$

which implies that the strategy of attacking more than one node on the same path cannot be an NE. □

Now we are ready to solve the NE. We cite the following well-known lemma [14] to conduct further analysis.

Lemma 1. *Every action in the support of any player's mixed strategy NE yields that player the same payoff.*

Let $\mathcal{P}^*$ denote the multipath set chosen by S at the NE, and q_i the probability that S chooses path $P_i \in \mathcal{P}^*$ to route its traffic at the NE, p_i the probability that M attacks P_i at the NE, $\tau_i = \tau(P_i, T) = \prod_{e\in P_i} r_e$. Applying Lemma 1 , we have

$$\begin{aligned} \tau_i(1 - p_i) &= \tau_j\left(1 - p_j\right), \\ & \qquad\qquad \forall P_i, P_j \in P, \\ q_i\tau_i &= q_j\tau_j. \end{aligned} \tag{13}$$

The packet delivery ratio $a = \sum_{P_i\in\mathcal{P}^*} q_i\tau_i(1 - p_i)$. Noticing $\sum_{P_i\in\mathcal{P}^*} p_i = 1$, we have $a = (|\mathcal{P}^*| - 1)/\sum_{P_i\in\mathcal{P}^*}(1/\tau_i)$, where $|\mathcal{P}^*|$ is the number of paths in $\mathcal{P}^*$. Noticing that a is the packet delivery ratio that S wants to maximize, solving the NE consists of finding the multipath set $\mathcal{P}^*$ such that $(|\mathcal{P}^*|-1)/\sum_{P_i\in\mathcal{P}^*}(1/\tau_i)$ is maximized. The maximized value is the solution of $\mathbf{MP_2}$. The strategy of S and M at the NE can be solved as follows.

(i) S's strategy: route the packet along path P_i with probability $q_i^* = 1/\tau_i \sum_{P_j\in\mathcal{P}^*}(1/\tau_j)$.

(i) A's strategy: attack path P_i with probability $p_i^* = 1 - ((|\mathcal{P}^*| - 1)/\tau_i \sum_{P_j\in\mathcal{P}^*}(1/\tau_j))$.

It follows from $p_i^* \leq 1$, for all $P_i \in \mathcal{P}^*$ that $\tau_i \geq (|\mathcal{P}^*| - 1)/(\sum_{P_j\in\mathcal{P}^*}(1/\tau_j))$. This implicates that M only focuses on a subset of routes to minimize a. Interestingly, S also has incentive to only route its packets on these paths even though other paths are attack free due to the fact that the attack-free paths are very poor in terms of performance. In summary, S should solve the following optimization problem $\mathbf{MP_2'}$ to find the NE:

$$\begin{aligned} a^* &= \max_{\mathcal{P}^*} \frac{|\mathcal{P}^*| - 1}{\sum_{P_i\in\mathcal{P}^*}(1/\tau_i)} \\ &\text{Subject to } \tau_i \geq \frac{|\mathcal{P}^*| - 1}{\sum_{P_j\in\mathcal{P}^*}\left(1/\tau_j\right)} \quad \forall P_i \in \mathcal{P}^*. \end{aligned} \tag{C_1}$$

4.2. Heuristic Path Set Computation Algorithm.

Although solving $\mathbf{MP_2'}$ is more tractable than solving $\mathbf{MP_2}$, yet it requires searching all possible node-disjoint paths between S and T, which leads to exponential time complexity. In the following, we propose a heuristic algorithm computing $\mathcal{P}^*$ with polynomial time complexity.

The goal of the heuristic algorithm is to find the optimal multipath set $\mathcal{P}^*$ such that $a = (|\mathcal{P}^*| - 1)/\sum_{P_i\in\mathcal{P}^*}(1/\tau_i)$ is maximized. We first introduce the two intuitions of the algorithm. Firstly, if we define τ_i as the reliability of path P_i, then choosing more reliable paths leads to higher global packet delivery ratio. Secondly, if we include more paths in $\mathcal{P}^*$, then $|\mathcal{P}^*|$ increases. However, the denominator of a also increases, especially when τ_i is small. Thus, the key point of our heuristic path set computation algorithm is to find as many node-disjoint paths as possible while at the same time as reliable as possible under the condition that the paths in the multipath set satisfy the constraint (C_1) such that the global packet delivery ratio a is maximized.

In order to change the path reliability from a multiplicative to an additive form, each edge $e \in \mathcal{E}$ is assigned

```
1: Input: network $\mathcal{G}$.
2: Output: multipath set $\mathcal{P}^*$ maximizing $a = (|\mathcal{P}^*| - 1)/\sum_{P_i \in \mathcal{P}^*}(1/\tau_i)$
3: Find the most reliable path $P_1$ by Dijkstra algorithm, select $P_1$; Set $\mathcal{P}^*(1) = \{P_1\}$, $k = 1$, $a = 0$.
4: for each path $P_i \in \mathcal{P}^*(k)$ do
5:     Inverse the direction of each edge on $P_i$, and make its length negative of the original link cost.
6:     Split each node $v$ on $P_i$ (except $S$ and $T$) into two nodes $v_1$ and $v_2$; Add an edge $(v_2, v_1)$ of cost 0. Replace each edge $(v', v) \in \mathcal{E}$
        by the edge $(v', v_1)$ without changing its reliability, replace each edge $(v, v'') \in \mathcal{E}$ by the edge $(v_2, v'')$ without changing
        its reliability.
7: end for
8: Run the Dijkstra algorithm, find the most reliable path $P'$ with reliability $\tau'$ in the transformed graph.
9: If $\tau' < |\mathcal{P}^*(k)|/(1/\tau') + \sum_{P_j \in \mathcal{P}^*(k)}(1/\tau_j)$, halt by returning $\mathcal{P}^*$.
10: Transform back to the original graph; erase any interlacing edges; group the remaining edges to form the new path set $\mathcal{P}^*(k+1)$.
11: If $a < (|\mathcal{P}^*(k+1)| - 1)/\sum_{P_i \in \mathcal{P}^*(k+1)}(1/\tau_i)$, then $\mathcal{P}^* = \mathcal{P}^*(k+1)$, $a = (|\mathcal{P}^*(k+1)| - 1)/\sum_{P_i \in \mathcal{P}^*(k+1)}(1/\tau_i)$.
12: If no more path can be found in the transformed graph, halt by returning $\mathcal{P}^*$, else $k = k + 1$ and go to 2.
```

Algorithm 2: Heuristic path set computation algorithm.

a weight $w_e = -\log p_e$. Then the conventional shortest path algorithm such as Dijkstra algorithm can be applied to find the most reliable path.

The heuristic path set computation algorithm, shown as above, is based on the K-node-disjoint shortest path algorithm [18]. The basic idea of the K-node-disjoint shortest path algorithm is to add a path in each iteration using graph transformation and link interlacing removal such that the total cost is minimized. We refer readers to [18] for a detailed description of the algorithm.

Algorithm 2 is a greedy approach finding the most reliable path at each iteration. The iteration continues as long as: (1) there exist paths in the transformed graph, implying that there exist node-disjoint paths in the original graph; (2) the constraint (C_1) is satisfied. At the end of the algorithm, the multipath set $\mathcal{P}^*$ maximizing a is returned. Once $\mathcal{P}^*$ is found, S routes its traffic along P_i with probability q_i^*.

One point concerning the correctness of the heuristic algorithm is that if the most reliable path found in the transformed graph satisfies the constraint (C_1) (in the transformed graph), then after erasing the interlacing edges, all the paths in the newly formed multipath set $\mathcal{P}^*(k+1)$ satisfy (C_1). This can be shown by recursively applying the following lemma.

Lemma 2. *If P_2 is the most reliable path in the transformed graph that satisfies the constraint (C_1) (in the transformed graph), then after erasing an interlacing edge with another path $P_1 \in \mathcal{P}^*$, the resulting path P_1' and P_2' satisfy (C_1).*

Proof. Please refer to appendix for the detailed proof. □

We conclude this subsection by addressing the complexity of Algorithm 2. The worst-case complexity of the heuristic algorithm is $O(n^3)$ in that there are at most d_s node-disjoint paths between S and T, where d_s is the number of outgoing edges from S. Since $d_s \leq n-1$, the algorithm iterates $n - 1$ times in the worst case (S can reach all nodes in the graph in one hop). In each iteration we run a minimum weight node-disjoint paths algorithm whose complexity is $O(n^2)$. The result is an overall worst-case complexity of $O(n^3)$.

5. Achieving Security-Performance Tradeoff

In Sections 3 and 4, we focus on the multipath routing solution minimizing the worst-case security risk and maximizing the worst-case packet delivery ratio. In fact, security and performance are two important aspects, of which neither should be ignored. Unfortunately, these two aspects sometimes lead to divergent routing solutions. Hence a natural next step is to investigate the multipath routing solution for multihop wireless networks that achieves a good tradeoff between the route security and performance. We formulated the routing problem in such context as the following maximinimization problem $\mathbf{MP_3}$:

$$\begin{gathered}
\max_{\mathbf{q}} \min_{\mathbf{p}} \sum_{P \in \mathcal{P}} \sum_{v \in P} q(P)\tau(P,T) \prod_{v \in P} (1 - p_v) \\
\text{Subject to} \sum_{v \in \mathcal{V}} \left[\sum_{v \in P, P \in \mathcal{P}} q(P)\tau(P,v)\varphi(P,v) \right] p_v \leq r_0, \\
\sum_{v \in \mathcal{V}} p_v \leq 1, \quad p_v \geq 0, \ \forall v \in \mathcal{V}, \\
\sum_{P \in \mathcal{P}} q(P) = 1, \quad q(P) \geq 0, \ \forall P \in \mathcal{P}.
\end{gathered} \tag{14}$$

In $\mathbf{MP_3}$, S wants to maximize the worst-case packet delivery ratio in the presence of attacker M, while limiting the worst-case security risk at most r_0. Directly solving $\mathbf{MP_3}$ needs an algorithm of exponential time complexity. In this section, we propose a heuristic solution based on Algorithm 2 to solve $\mathbf{MP_3}$. As discussed in Section 4, maximizing the worst-case packet delivery ratio equals to solve $\max_{\mathcal{P}^*}(|\mathbf{P}^*| - 1)/\sum_{P_i \in \mathcal{P}^*}(1/\tau_i)$ under the constraint (C_1). The routing strategy for S is to route the packets along path P_i with probability $q_i^* = 1/\tau_i \sum_{P_j \in \mathcal{P}^*}(1/\tau_j)$. In such context, it is easy to compute the worst-case security risk as $r = \max_{P_i \in \mathcal{P}^*}(r_{e_1^i}/\tau_i \sum_{P_j \in P}(1/\tau_j))$ where $r_{e_1^i}$ is the reliability of the first edge of P_i, since $\max_{\mathbf{p}}\min_{\mathbf{q}} r = \min_{\mathbf{q}}\max_{\mathbf{p}} r$, and the first constraint of **MP** on the security risk can be transformed into

$$\tau_i \geq \frac{r_{e_1^i}}{r_0 \sum_{P_j \in \mathcal{P}^*}\left(1/\tau_j\right)}, \quad \forall P_i \in \mathcal{P}^*. \qquad (C_2)$$

Our heuristic solution is extended form Algorithm 2. The key idea is to include enough number of reliable paths in $\mathcal{P}^*$ to limit the security risk. The intuition behind is that distributing the traffic among more paths helps limit the security risk. With this in mind, we modify Algorithm 2 such that the iteration stops until the constraints (C_1) and (C_2) are both satisfied or there is no more node-disjoint path available. In the latter case, the heuristic algorithm fails to find the multipath routing solution to $\mathbf{MP_3}$. This failure may due to the fact that the constraint on the security risk is too stringent such that no possible multipath set can meet the constraint, or alternatively, the heuristic algorithm itself cannot find the solution though it does exist. In such cases, possible solutions include secret sharing and information dispersion in which the key idea is to divide the packet to N parts, and the recovery of the packet is possible only with at least T parts. These techniques can further decrease the security risk and improve the performance. We refer readers to [3, 19] since they are out of the scope of our work.

6. Theoretical Security-Performance Limit of Node-Disjoint Multipath Routing

In this section, we establish the relationship between the worst-case packet delivery ratio a^* and the worst-case security risk r^* in node-disjoint multipath routing. The relationship gives one important security-performance limit of the node-disjoint multipath routing with the presence of an attacker in the sense that we cannot find better routing solutions with node-disjoint paths whose security and performance can go beyond the limit.

Let $\mathcal{P}^{\text{nd}}$ be the node-disjoint multipath set selected by S to route traffic; we have shown in Section 4 that

$$a^* = \frac{\left|\mathcal{P}^{\text{nd}}\right| - 1}{\sum_{P_i \in \mathcal{P}^{\text{nd}}}(1/\tau_i)}. \qquad (15)$$

On the other hand, let $q_k^0 = 1/\tau_k \sum_{P_j \in \mathcal{P}^{\text{nd}}}(1/P_j)$. We have $\sum_{P_k \in \mathcal{P}^{\text{nd}}} q_k^0 = 1 = \sum_{P_k \in \mathcal{P}^{\text{nd}}} q_k$, where q_k is the probability of routing packets along P_k. From the Pigeon Hole Principle, there exists at least one path $P_m \in \mathcal{P}^{\text{nd}}$ such that $q_m \geq q_m^0$. It follows that

$$\begin{aligned} r^* &= \min_{\mathbf{q}}\max_{\mathbf{p}} = \max_{\mathbf{p}}\min_{\mathbf{q}} \\ &\geq q_m r_{e_1^m} = \frac{r_{e_1^m}}{\tau_m \sum_{P_j \in \mathcal{P}^{\text{nd}}}\left(1/\tau_j\right)}, \end{aligned} \qquad (16)$$

where $r_{e_1^m}$ is the reliability of the first edge on P_m.

As a result, we get

$$\frac{a^*}{r^*} = \left(\left|\mathcal{P}^{\text{nd}}\right| - 1\right)\frac{\tau_m}{r_{e_1^m}} \leq \left|\mathcal{P}^{\text{nd}}\right| - 1 \leq \left|\mathcal{P}^{\text{nd}}\right|_{\max} - 1, \qquad (17)$$

where $|\mathcal{P}^{\text{nd}}|_{\max}$ is the maximum number of node-disjoint path between S and T.

As a limit of node-disjoint multipath routing, the above relationship shows the intrinsic constraint of minimizing r and maximizing a at the same time. More specifically, if we want to limit the worst-case security risk as low as r, it is impossible to achieve $a > (|\mathcal{P}^{\text{nd}}|_{\max} - 1)r$; if we want to guarantee the worst-case packet delivery ratio as high as a, then we should expect the worst-case security risk of at least $r/(|\mathcal{P}^{\text{nd}}|_{\max} - 1)$. Moreover, given the requirement on the route security and performance, one can check if it is realizable or too stringent by using the above formula before searching for the routing solution.

7. Multipath Routing with Multiple Attackers

In this section, we extend our efforts to investigate the case where there are n $(n > 1)$ attackers in the network.

7.1. Minimizing Worst-Case Security Risk. There are various formulations of the multipath routing problem under n attackers to minimize the worst-case security risk, among which we are interested in two typical formulations. In the first formulation, let r_i be the probability that a packet is captured by attacker i, and S wants to minimize $\sum r_i$. This case can be regarded as the case where S plays the multipath routing game G_1 with each of the attackers. Hence, the solution of $\mathbf{MP_1}$ can be applied here. The only difference is that the resulting minimum worst-case security risk is nr^*. However, this does not influence routing strategy of S; in other words, no matter how many attackers are there, the routing strategy of $\mathbf{MP_1}$ provides the most secure routing strategy minimizing the worst-case security risk in this case.

In the second formulation, the security risk is defined as the probability that a packet is captured by at least one attacker. In this context, the attackers will arrange their attacks such that no more than one attacker will attack the same node simultaneously; that is, they try to coverage the most nodes possible to maximize the probability of capturing the packet. Similar as in Section 3.2, we can show that the attackers attack at most one node per path to maximize the security risk. For S, to minimize the worst-case security risk is to solve the following optimization problem $\mathbf{MP_4}$:

$$\begin{aligned} &\min_{\mathbf{q}}\max_{\mathbf{p}} \sum_{v \in \mathcal{V}} \left[\sum_{v \in P, P \in \mathcal{P}} q(P)\tau(P, v)\right] p_v \\ &\text{Subject to} \sum_{v \in \mathcal{V}} p_v \leq n, \quad 0 \leq p_v \leq 1, \ \forall v \in \mathcal{V}, \qquad (18) \\ &\qquad \sum_{P \in \mathcal{P}} q(P) = 1, \quad q(P) \geq 0, \ \forall P \in \mathcal{P}, \end{aligned}$$

where p_v is the probability that a node v is attacked by any of the n attackers.

$\mathbf{MP_4}$ is a linear optimization problem and can be solved by classical linear programming techniques. However, due to additional constraints $p_v \leq 1$, $\mathbf{MP_4}$ cannot be transformed into maximum flow problem in lossy networks as $\mathbf{MP_1}$ that can be solved in polynomial time. As a result, solving $\mathbf{MP_4}$ may require an algorithm with exponential time complexity.

In the following, we give the upper bound of the worst-case security risk under n attackers. To this end, we relax the constraint $p_v \leq 1$ and perform variable transformation by letting $p'_v = p_v/n$. $\mathbf{MP_4}$ after the transformation becomes $\mathbf{MP'_4}$:

$$\min_{\mathbf{q}} \max_{\mathbf{p}} n \sum_{v \in \mathcal{V}} \left[\sum_{v \in P, P \in \mathcal{P}} q(P)\tau(P, v) \right] p'_v$$

$$\text{Subject to} \sum_{v \in \mathcal{V}} p'_v \leq 1, \quad 0 \leq p'_v \leq 1, \ \forall v \in \mathcal{V} \qquad (19)$$

$$\sum_{P \in \mathcal{P}} q(P) = 1, \quad q(P) \geq 0, \ \forall P \in \mathcal{P}.$$

$\mathbf{MP'_4}$ is identical to $\mathbf{MP'_1}$ except for a constant coefficient n. It follows immediately that its solution is n/f^* where $1/f^*$ is the maximum flow in $\mathbf{MP'_1}$. Let $r\prime$ be the worst-case security risk under n attackers; following the fact that $\mathbf{MP'_4}$ is obtained by relaxing the constraint $p_v \leq 1$ in $\mathbf{MP_4}$, it holds that $r' \leq n/f^*$. In summary, by increasing the number of attackers from 1 to n, the worst-case security risk increases at most n times.

7.2. Maximizing Worst-Case Packet Delivery Ratio. We consider the multipath routing game between S and the attacker side consisting of n attackers. S tries to maximize the packet delivery ratio and the attacker side tries to minimize it. It can be shown that at the NE of the game, no more than one attacker attacks the same node at the same time. This is because attacking the same node at the same time gives the attacker side the same payoff as the case where only one attacker attacks the node, which gives the attacker side less payoff than the case where the attacker side arranges the attack to cover the most number of nodes possible. With this in mind, by conducting the similar analysis as in Section 4.1, the optimization problem S should solve in multiple-attacker case $\mathbf{MP_5}$

$$\max_{\mathcal{P}^*} \frac{|\mathcal{P}^*| - n}{\sum_{P_i \in \mathcal{P}^*} (1/\tau_i)}$$

$$\text{Subject to } \tau_i \geq \frac{|\mathcal{P}^*| - n}{\sum_{P_j \in \mathcal{P}^*} \left(1/\tau_j\right)} \qquad \forall P_i \in \mathcal{P}^*, \qquad (C_3)$$

where $\mathcal{P}^*$ consists of node-disjoint paths. The extension of Algorithm 2 to solve $\mathbf{MP_5}$ is straightforward.

We now investigate the case where S also wants to limit the worst-case security risk as low as r_0 at the same time, as in Section 5. Recall that $r_{e_1^i}$ denotes the reliability of the first edge of P_i, and we sort the path by $r_{e_1^i}/\tau_i$, that is, $r_{e_1^i}/\tau_i \leq r_{e_j^1}/\tau_j \Leftrightarrow i \leq j$. The worst-case security risk in multiple-attacker case is $\sum_{i=1}^{n}(r_{e_i^1}/\tau_i \sum_{P_j \in P}(1/\tau_j))$, which is achieved when the n attackers attack the n most profitable paths. To limit the worst-case security risk, the constraint $\sum_{i=1}^{n}(r_{e_i^1}/\tau_i \sum_{P_j \in P}(1/\tau_j)) \leq r_0$ should be added to $\mathbf{MP_5}$. Algorithm 2 can be extended in a similar way as Section 5 solves it. In the multiple-attacker case, if $|\mathcal{P}^{\text{nd}}|_{\max} \leq n$, the communication between S and T is paralyzed by the attackers.

Table 1: Simulation parameters.

Simulation time	1000 s
Number of nodes	100, randomly distributed
Network dimension	1000 m × 1000 m
Transmission range	200 m
Node speed	4 m/s, Random waypoint model
Data traffic	CBR 4 pkt/s 64 bytes per pkt

Table 2: Simulation results: single-attacker case.

	Scenario 1		Scenario 2	
	r	p_s	r	p_s
MinSR	15.2%	54.2%	13.1%	50.3%
MaxDR	19.1%	62.2%	16.8%	59.0%
MaxDR-SR	15.8%	58.2%	15.3%	54.4%
SMT	32.3%	48.5%	39.8%	36.5%
DPSP	24.1%	49.7%	22.8%	45.3%

8. Performance Evaluation

In this section, we evaluate the performance of proposed multipath routing solutions through simulation using Network Simulator (NS 2). Table 1 shows the simulation setting. The link reliability of each link is generated from a normal distribution $\sigma(0.7, 0.2)$ trunked in $[0, 1]$ interval.

8.1. Single-Attacker Case. We start with single-attacker case. Two scenarios are simulated: the attacker launches its attack to maximize the packet capture probability (scenario 1) or minimize the packet delivery ratio (scenario 2). In both scenarios, we assume that the attacker knows the routing strategy of S.

We compare our solutions with SMT [3] and DPSP [1]. To focus on the multipath routing solution itself and perform a fair comparison, we do not implement the message dispersion in SMT. Since SMT and DPSP do not specify how to balance traffic among the paths, we let S chose randomly in the multipath set when having a packet to send.

Let MinSR denote the multipath routing algorithm minimizing the worst-case security risk, MaxDR denote the heuristic multipath routing algorithm maximizing the worst-case packet delivery ratio, and MaxDR-SR denote the heuristic multipath routing algorithm maximizing the worst-case packet delivery ratio while limiting the worst-case security risk under certain threshold (the threshold is set to 16% in out simulation). In MinSR, to balance the complexity of the algorithm and the solution optimality, we set $\epsilon = 0.05$. Table 2 shows the simulation results.

The simulation results show that SMT performs poorly in both scenarios. This is due to the fact that in our simulation, different from the scenarios simulated in literatures [3, 20], we simulate the worst-case scenarios where the attacker

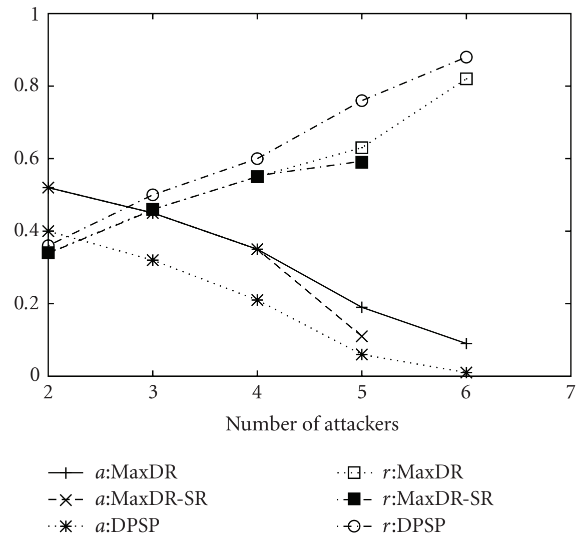

Figure 3: Multiple-attacker case: scenario 1.

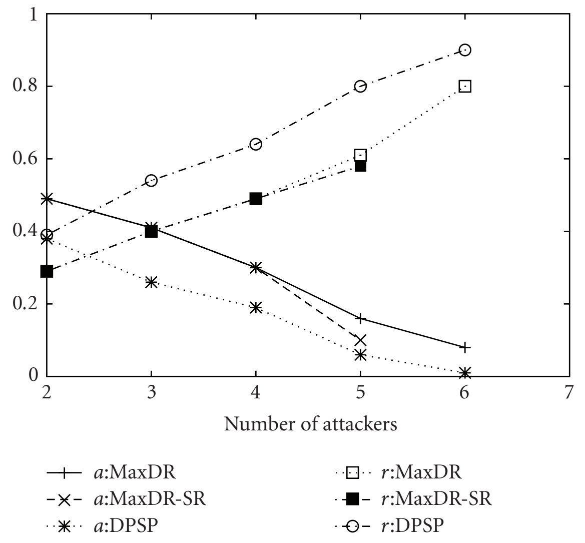

Figure 4: Multiple-attacker case: scenario 2.

launches its attack in the unpredictable way which is not correlated with the history rating. In such context, the attacker can actually take the advantage of the path rating mechanism to cause more severe damage. DSDP performs almost the same in two scenarios in that it selects the most reliable multipath set without taking into consideration of attackers. The resilience to attacks of DPSP is purely due to its multipath nature.

For our solution MinSR, it achieves the minimum security risk in scenario 2, which confirms the analytical result in that the upper bound of the security risk r^* is achieved in scenario 1. However, the packet delivery ratio in MinSR is less than that in MaxDR. This is due to the limitation of MinSR discussed in Section 3.4. From the simulation, we can see that the suboptimality of MinSR in terms of performance can be rather important compared to MaxDR, which achieves the best performance among all the simulated multipath routing solutions. MaxDR-SR, on the other hand, achieves a tradeoff between the route security and performance, which is shown by the simulation results that MaxDR-SR lies between MinSR and MaxDR in terms of route security and performance. Furthermore, we observe the fact that the number of maximum node-disjoint paths in our simulation is around 6. From this observation, we can verify the relation between the route security and performance using the formula derived in Section 6 on the theoretical limit of node-disjoint multipath routing.

8.2. Multiple-Attacker Case. We then evaluate the performance of MaxDR and MaxDR-SR (the security risk threshold r_0 is set to 0.55) in cooperative multiple-attacker case where the attacker side arranges their attacks on a subset of paths so as to maximize the security risk in scenario 1 and to minimize the packet delivery ratio in scenario 2. Figures 3 and 4 plot a and r as a function of the number of attackers. SMT is not plotted here since the worst-case packet delivery ratio of SMT drops below 20% even with 2 attackers. MinSR is not simulated here in that according to our analysis in Section 7.1, the first formulation is simply the aggregated case of the single-attacker case; in the second formulation, no polynomial routing algorithm exists minimizing the worst-case security risk.

The results show that the performance degrades significantly with the increase of the number of attackers. The communication is almost paralyzed with 5 attackers. At the presence of 6 attackers, MaxDR-SR cannot find routing solution whose security risk is not more than 0.55. Once again, our results seem very different from those obtained from literatures. This is because we focus on the worst-case scenarios throughout this paper. Unlike the traditional simulation where a percentage of nodes is assumed to be compromised, we implement much more powerful attackers with perfect knowledge of the network and the routing strategies. These attackers are able to launch the most severe attacks which are not predictable nor correlated in time or space. In such context, our results reflect the lower bound of performance of the simulated routing solutions. We argue that maximizing this lower bound, as discussed in our work, is of great importance since the attackers cannot be underestimated in any case. Meanwhile, we can see from the results that our solutions perform substantially better than DPSP in terms of both route security and performance.

In summary, the simulations show that the proposed multipath routing solutions achieve the design objective of providing the best security and/or performance in the worst-case scenarios.

9. Conclusion

In this paper, we address the fundamental problem of how to choose secure and reliable paths in wireless networks. We formulate the multipath routing problem as optimization problems and propose algorithms with polynomial complexity to solve them. Three multipath routing solutions are

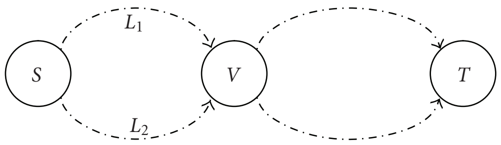

Figure 5: Two paths forms a cycle.

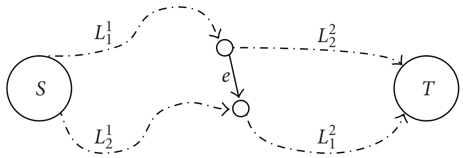

Figure 6: P_1, P_2 shares the edge e.

proposed: MinSR minimizes the worst-case security risk, MaxDR maximizes the worst-case packet delivery ratio, and MaxDR-SR achieves a tradeoff between them by maximizing the worst-case packet delivery ratio while limiting the worst-case security risk under given threshold. We also establish the relationship between the worst-case security risk and packet delivery ratio, which gives the theoretical security-performance limit of node-disjoint multipath routing.

The analytical and simulation results in the paper lead us to the following conclusion.

(i) Solutions based on path rating which work well in the presence of time or location correlated attacks may fail to provide secure and reliable paths facing strategic attackers with unpredictable attack patterns.

(ii) Two issues are crucial in multipath routing. Firstly, both the security and performance should be taken into account when choosing the optimal paths, as in [2] and our work. Secondly, the traffic should be balanced among paths such that they are equally "attractive" to attackers.

(iii) Among the proposed multipath solutions, MaxDR-SR achieves good security-performance tradeoff by choosing sufficient number of mutually disjoint paths with high reliability and balancing the traffic in the optimal way.

Appendix

A. Proof of Theorem 2

By [11, Corollary 2.3.4], the maximum flow in lossy networks can be decomposed into at most m augmenting paths. Algorithm 1 selects the path that generates the maximum amount of excess at the sink. Thus, each iteration captures at least a $1/m$ fraction of the remaining flow. Let f_k be the flow after iteration k, and we have

$$\begin{aligned} f_1 &\geq \frac{1}{m} f^*, \\ f_2 &\geq f_1 + \frac{1}{m}(f^* - f_1), \\ &\cdots \\ f_k &\geq f_{k-1} + \frac{1}{m}(f^* - f_{k-1}). \end{aligned} \tag{A.1}$$

Injecting $f_{k-1}, \ldots, f_2, f_1$ into f_k, we have

$$\begin{aligned} f_k &\geq f_{k-1} + \frac{1}{m}(f^* - f_{k-1}) \\ &= \frac{1}{m} f^* + \frac{m-1}{m} f_{k-1} \\ &\geq \frac{1}{m} f^* + \frac{m-1}{m}\left(\frac{1}{m} f^* + \frac{m-1}{m} f_{k-2}\right) \\ &= \frac{1}{m}\left(1 + \frac{m-1}{m}\right) f^* + \left(\frac{m-1}{m}\right)^2 f_{k-2} \\ &\geq \frac{1}{m}\left(1 + \frac{m-1}{m}\right) f^* + \left(\frac{m-1}{m}\right)^2 \left(\frac{f^*}{m} + \frac{m-1}{m} f_{k-3}\right) \\ &= \frac{1}{m}\left(1 + \frac{m-1}{m} + \left(\frac{m-1}{m}\right)^2\right) f^* + \left(\frac{m-1}{m}\right)^3 f_{k-3} \\ &\geq \cdots \\ &\geq \frac{1}{m}\left[\sum_{i=0}^{k-2}\left(\frac{m-1}{m}\right)^i\right] f^* + \left(\frac{m-1}{m}\right)^{k-1} f_1 \\ &\geq \left[1 - \left(\frac{m-1}{m}\right)^{k-1}\right] f^* + \left(\frac{m-1}{m}\right)^{k-1} \frac{1}{m} f^* \\ &= \left[1 - \left(\frac{m-1}{m}\right)^{k}\right] f^*. \end{aligned} \tag{A.2}$$

Algorithm 1 terminates if $f^* - [1 - ((m-1)/m)^k] f^* < \epsilon_o$, that is, $k > \log_{m/(m-1)}(f^*/\epsilon_0)$.

B. Proof of Theorem 4

We have shown that there exists at least one NE in G_2. We now show that if the NE consists of overlapped paths with common nodes, we can construct another NE with node-disjoint paths.

We first give some definitions. For two paths sharing nodes A, B with $(A, B) \neq (S, T)$, let Q_1 and Q_2 be the node sequence of the two paths between A and B. Q_1, Q_2 can be empty, but they cannot both be empty. Let $l(Q)$ denote the number of nodes in the sequence Q, we call the node sequence AQ_1BQ_2A a *cycle*, and define the *diameter* of the cycle AQ_1BQ_2A as $\min\{l(Q_1), l(Q_2)\}$.

Assume that at the NE, there exists paths with common nodes. We now study the cycle containing S with the common nodes S and V with the smallest diameter. Suppose that this cycle is formed by paths P_1 and P_2 with the node sequence $L_1 \in P_1$ and $L_2 \in P_2$ between S and V, as shown in Figure 5 . Without loss of generality, we assume that $l(L_1) \leq l(L_2)$. It follows that at the NE, any node $V_n \in L_1$ does not belong to the multipath set chosen by the source except P_1; otherwise we find a cycle with smaller diameter, which contradicts our assumption. It then holds that, at the NE, the attacker has no incentive to attack any nodes on L_1 because if it attacks any node on L_1 with probability p, it gets less payoff if it uses the same resource attacking V. From the definition of NE, routing the packets on L_1 gives S the same payoff as routing them on L_2. Hence, we can switch all the traffic from L_1 to L_2 without changing the payoff of S. Moreover, since the attacker does not attack any node on L_1 at the NE, this operation does not change the payoff of the attacker, either. Therefore, it is easy to verify that the multipath set after the above operation is also an NE of G_2. However, the number of cycles decreases by one. As a result, by recursively repeating the above process, we can transfer any NE to an NE where the number of cycles is 0. Such NE consists of only node-disjoint paths between S and T.

C. Proof of Lemma 2

The lemma holds evidently if P_2 does not intercross P_1. In the following we prove the case where P_2 intercrosses with P_1. As illustrated in Figure 6 , P_1 is composed of L_1^1, e, L_1^2, and P_2 is composed of L_2^1, e, L_2^2 before erasing the interlacing edge e. Here L_i^j $(i, j = 1, 2)$ denotes a sequence of edges. Since P_2 satisfies the constraint (C_1), we have

$$r_2^1 \frac{1}{r_e} r_2^2 \geq \frac{|\mathcal{P}^*(k)|}{1/r_1^1 r_e r_1^2 + r_e/r_2^1 r_2^2 + \Gamma}, \qquad \text{(C.1)}$$

where $\Gamma = \sum_{P_j \in \mathcal{P}^*(k), P_j \neq P_1} (1/\tau_j)$ and $r_i^j = \prod_{e \in L_i^j} r_e$ $(i, j = 1, 2)$. At this moment, P_2 has not been added into $\mathcal{P}^*(k)$ yet, and so the numerator of the above inequality and that in step 7 in Algorithm 2 is $|\mathcal{P}^*(k)|$, not $|\mathcal{P}^*(k)| - 1$. Note that the cost of e is $-\log(r_e)$ in P_1 and $\log(r_e)$ in P_2 in the transformed graph.

Since the Dijkstra algorithm is applied on the graph with link cost $w_e = -\log r_e$, it follows that $r_1^1 r_e \geq r_2^1$ and $r_e r_1^2 \geq r_2^2$. Hence, we have

$$\begin{aligned}
&\frac{1}{r_2^1 r_1^2} \geq \frac{1}{r_1^1 r_e r_1^2}, \quad r_1^1 r_2^2 \geq \frac{r_2^1 r_2^2}{r_e} \\
&\quad \Longrightarrow 1 + \frac{r_1^1 r_2^2}{r_2^1 r_1^2} + r_1^1 r_2^2 \Gamma \\
&\quad \geq 1 + \frac{r_2^1 r_2^2}{r_1^1 (r_e)^2 r_1^2} + \frac{r_2^1 r_2^2}{r_e} \Gamma \\
&\quad \Longrightarrow r_1^1 r_2^2 \left(\frac{1}{r_1^1 r_2^2} + \frac{1}{r_2^1 r_1^2} + \Gamma \right) \\
&\quad \geq \frac{r_2^1 r_2^2}{r_e} \left(\frac{1}{r_1^1 r_e r_1^2} + \frac{r_e}{r_2^1 r_2^2} + \Gamma \right) \\
&\quad \Longrightarrow r_1^1 r_2^2 \left(\frac{1}{r_1^1 r_2^2} + \frac{1}{r_2^1 r_1^2} + \Gamma \right) \geq |\mathcal{P}^*(k)| \\
&\quad \Longrightarrow \tau_1' = r_1^1 r_2^2 \geq \frac{|\mathcal{P}^*(k)|}{1/r_1^1 r_2^2 + 1/r_2^1 r_1^2 + \Gamma}.
\end{aligned} \qquad \text{(C.2)}$$

In the same way, we can show that $\tau_2' = r_2^1 r_1^2 \geq |\mathcal{P}^*(k)|/(1/r_1^1 r_2^2 + 1/r_2^1 r_1^2 + \Gamma)$. Noticing that P_1', P_2' consist of $r_1^1 r_2^2$ and $r_2^1 r_1^2$, respectively, it follows that both P_1' and P_2' satisfy (C_1), which concludes our proof.

References

[1] P. Papadimitratos, Z. J. Haas, and E. G. Sirer, "Path set selection in mobile ad hoc networks," in *Proceedings of the International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc '02)*, pp. 1–11, Lausanne, Switzerland, June 2002.

[2] W. Lou, W. Liu, and Y. Fang, "SPREAD: enhancing data confidentiality in mobile ad hoc networks," in *Proceedings of the Conference on IEEE Computer and Communications Societies (INFOCOM '04)*, vol. 4, pp. 2404–2413, Hong Kong, April 2004.

[3] P. Papadimitratos and Z. J. Haas, "Secure data communication in mobile ad hoc networks," *IEEE Journal on Selected Areas in Communications*, vol. 24, no. 2, pp. 343–356, 2006.

[4] J. P. Brumbaugh-Smith and D. R. Shier, "Minimax models for diverse routing," *INFORMS Journal on Computing*, vol. 14, no. 1, p. 8195, 2002.

[5] J. P. Hespanha and S. Bohacek, "Preliminary results in routing games," in *Proceedings of the American Control Conference (ACC '01)*, vol. 3, pp. 1904–1909, Arlington, Va, USA, June 2001.

[6] P. P. C. Lee, V. Misra, and D. Rubenstein, "Distributed algorithms for secure multipath routing," in *Proceedings of the Conference on IEEE Computer and Communications Societies (INFOCOM '05)*, vol. 3, pp. 1952–1963, Miami, Fla, USA, April 2005.

[7] S. Bohacek, J. Hespanha, J. Lee, C. Lim, and K. Obraczka, "Enhancing security via stochastic routing," in *Proceedings of the International Conference on Computer Communications and Networks (ICCCN '02)*, Miami, Fla, USA, October 2002.

[8] Y. Wang, M. Martonosi, and L. Peh, "A new scheme on link quality prediction and its applications to metric-based routing," in *Proceedings of the ACM Workshop on Security of Ad Hoc and Sensor Networks (SENSYS '05)*, San Diego, Calif, USA, November 2005.

[9] S. Zhong, L. Li, Y. G. Liu, and Y. R. Yang, "On designing incentive-compatible routing and forwarding protocols in wireless ad-hoc networks—an integrated approach using game theoretical and cryptographic techniques," in *Proceedings of the ACM Annual International Conference on Mobile Computing and Networking (MobiCom '05)*, pp. 117–131, Cologne, Germany, August 2005.

[10] P. Papadimitratos and Z. J. Haas, "Secure link state routing for mobile ad hoc networks," in *Proceedings of the IEEE Workshop on Security and Assurance in Ad Hoc Networks*, 2003.

[11] K. D. Wayne, *Generalized maximum flow algorithms*, Ph.D dissertation, Cornell University, 1999.

[12] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin, *Network Flows: Theory, Algorithms, and Applications*, Prentice-Hall, Englewood Cliffs, NJ, USA, 1993.

[13] M. Shigeno, “A survey of combinatorial maximum flow algorithms on a network with gains,” *Journal of the Operations Research Society of Japan*, vol. 47, no. 4, pp. 244–264, 2004.

[14] M. J. Osborne and A. Rubinstein, *A Course in Game Theory*, MIT Press, Cambridge, Mass, USA.

[15] W. Mayeda and M. Van Valkenburg, “Properties of lossy communication nets,” *IEEE Transactions on Circuits and Systems*, vol. 12, no. 3, pp. 334–338, 1965.

[16] A. Washburn and K. Wood, “Two-person sum games for network interdiction,” *Operations Research*, vol. 43, pp. 243–251, 1995.

[17] M. Kodialam and T. V. Lakshman, “Detecting network intrusions via sampling: a game theoretic approach,” in *Proceedings of the Conference on IEEE Computer and Communications Societies (INFOCOM ’03)*, vol. 3, pp. 1880–1889, San Francisco, Calif, USA, April 2003.

[18] R. Bhandari, “Optimal physical diversity algorithms and survivable networks,” in *Proceedings of the IEEE Symposium on Computers and Communications*, pp. 433–441, Alexandria, Egypt, July 1997.

[19] J. Yang and S. Papavassiliou, “Improving network security by multipath traffic dispersion,” in *Proceedings of IEEE Military Communications Conference on Communications for Network-Centric Operations: Creating the Information Force (MILCOM ’01)*, Washington, DC, USA, October 2001.

[20] M. Kefayati, H. R. Rabiee, S. G. Miremadi, and A. Khonsari, “Misbehavior resilient multi-path data transmission in mobile ad-hoc networks,” in *Proceedings of the 4th ACM Workshop on Security of ad hoc and Sensor Networks (SASN ’06)*, pp. 91–100, Alexandria, Va, USA, October 2006.