Barrier Penetration Routing Against Wireless Spy Sensors

Kazuya Sakai, *Member, IEEE,* Min-Te Sun, *Member, IEEE,* Wei-Shinn Ku, *Senior Member, IEEE,* and Jie Wu, *Fellow, IEEE,*

Abstract—We consider a potential communication problem in national security, where wireless spy sensors with eavesdropping capability are strategically deployed around an area of interest. For counterintelligence, achieving secure communication by penetrating such a spy barrier is of great importance. In this paper, we first formulate the problem of barrier penetration routing against spy barriers consisting of strategically deployed wireless sensors. We point out that existing multi-path avoidance routing protocols cannot efficiently counteract collusion attacks, where connected adversaries collaborate with each other to compromise data packets. We propose a barrier penetration routing (BPR) protocol to securely penetrate the barrier of adversaries. In the protocol, a set of physically distanced paths are identified based on distance vectors as well as network-wide flooding. Then, each data packet encoded by XOR coding is routed via a different path. Unlike existing avoidance routing, the proposed scheme does not rely on the assumption that the adversary's locations are known. The simulation results demonstrate that the proposed BPR outperforms the baseline protocol as well as existing routing protocols in terms of secure delivery rate.

Index Terms—Barrier penetration routing, avoidance routing, wireless sensor networks.

1 Introduction

The concept of *barrier coverage*, in which stealthy wireless sensors with sensing capabilities are deployed in a national border to discourage, detect, and thwart intrusion attempts, was introduced by Kumar et al. [1] to protect a national border by wireless sensors more than ten years ago. Since then, a number of research works have been proposed, including camera barrier coverage [2], one-way barrier coverage [3], and target barrier coverage [4], primarily to protect a homeland. However, to the best of our knowledge, the critical scenario, in which the concept of barrier coverage is exploited by a hypothetical enemy, has never been considered. For example, a military camp in a hypothetical enemy territory could be surrounded by wireless spy sensors with eavesdropping capability. Then, all the traffic from the camp to the military post or garrison would be compromised by these spy sensors. Another potential scenario is that malicious wireless sensors are deployed around IoT smarthome and/or smart factories [5] to compromise data communications among IoT devices. In this paper, we seek to design a secure sensor network routing protocol that penetrates a *spy barrier* of wireless sensors in such scenarios.

Figure 1 shows an example of barrier penetration routing, where the shaded circles represent the source and destination nodes, the shaded squares represent adversaries, and the large circles represent the eavesdropping areas of adversaries. There are three routes, paths 1, 2, and 3, from the source to the destination, and all of them must go through the barrier of adversaries. While the use of cryptography protects data privacy against polynomial time adversaries, the computational power of the adversaries in this paper is assumed to be *unbounded*, i.e., once an adversary eavesdrops on an encrypted message, the adversary can quickly compromise the message. In addition, we assume that a group of adversaries can perform a *collusion attack* to compromise messages, where connected components of adversaries can collaborate with each other.

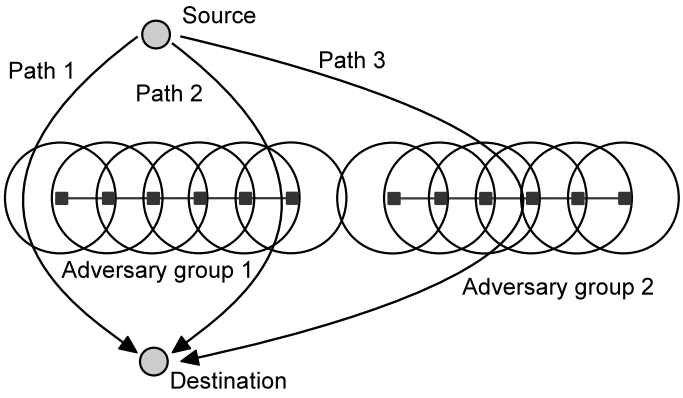


Fig. 1. Multi-path avoidance routing with the XOR coding.

At first glance, one may think that avoidance routing alone, which avoids insecure areas instead of counteracting the security threats, can securely deliver messages under such an attack model. However, while a number of avoidance routing protocols in [6]–[10] for wireless sensor networks (WSNs) have been proposed, none of them works in our critical scenario. The single-path based protocols in [6] and [7] successfully discover a safe path along which no adversary eavesdrops on the channel when adversaries are randomly deployed. However, there may exist no safe path

- *Kazuya Sakai is with the Department of Electrical Engineering and Computer Science, Tokyo Metropolitan University, 6-6 Asahigaoka, Hino, Tokyo 191-0065, Japan. E-mail: ksakai@tmu.ac.jp*
- *Min-Te Sun is with the Department of Computer Science and Information Engineering, National Central University, Taoyuan 320, Taiwan. E-mail: msun@csie.ncu.edu.tw*
- *Wei-Shinn Ku is with the Department of Computer Science and Software Engineering, Auburn University, Auburn, AL 36849, USA. E-mail: weishinn@auburn.edu*
- *Jie Wu is with the Department of Computer and Information Science, Temple University, 1925 N. 12th St. Philadelphia, PA 19122. E-mail:jiewu@temple.edu*

Manuscript received June xx, 2022; revised August xx, 2022.

between the source and destination, if adversaries are strategically deployed as shown in Figure 1. To overcome the performance limit, multi-path-based avoidance routing (MPAR) [8] with XOR coding and its variants [9], [10] are proposed. In MPAR, message m is encoded into k messages, $m_1, m_2, ..., m_k$, where $m = m_1 \oplus m_2 \oplus ... \oplus m_k$. Each m_i $(1 \leq i \leq k)$ is sent via a different path and the destination node assembles the original m after receiving all pieces. Since XOR coding serves as a one-time pad, perfect secrecy is guaranteed, unless an adversary obtains all the pieces. Nevertheless, the existing MPAR protocol family [8], [9] does not accommodate a collusion attack. For example, in Figure 1, there are two connected components of adversaries, groups 1 and 2. Assume that message m is encoded into two pieces m_1 and m_2, where $m = m_1 \oplus m_2$, and each of them is forwarded via a different path. The set of path 1 and path 2 is not safe, since the adversaries in group 1 can obtain both m_1 and m_2 routed via paths 1 and 2 by collaborating with each other. On the other hand, the set of paths 1 and 3 is safe, as the adversaries in neither group 1 nor 2 are able to obtain both pieces.

Therefore, the MPAR protocol family may succeed in delivering messages across a barrier. However, there are three key differences between barrier penetration routing and existing avoidance routing problems [6]–[11], [13]–[15]. First, avoidance routing solutions rely on the assumption of known adversaries' locations, which does not reflect real scenarios in which adversaries clandestinely listen to the channel. Second, adversaries are assumed to be randomly deployed over the entire network, i.e., strategic deployment along a belt region is not considered. Third, many existing works are not intended to alleviate a collusion attack. While MCAR proposed in [10] explicitly avoids collusion attacks, the location of adversaries as well as the hop distance among them are required. Another approach to countermeasure collusion attacks is a jamming-based solution. The work in [12] derives the secure throughput of cooperative jamming-based secure routing protocols without location information of adversaries. Although cooperative jamming defends data against collusion attacks, the solutions of this kind require additional hardware capabilities, and thus, they are considered too expensive.

A summary of comparisons among related works is presented in Table 1. The aforementioned factors impose a totally different protocol design issue in how to securely deliver messages across a barrier of adversaries without the information of the adversaries' locations under the collusion attack, which cannot be addressed by a simple extension of MPAR [8]. Specifically, the contributions of this paper are as follows:

- First, we introduce a new secure routing problem to securely penetrate a spy barrier of wireless sensors, including the attack models as well as the definition of a barrier of adversaries along a belt region. In addition, the notion of a strong barrier, i.e., there is no set of safe paths from source to destination with a high probability, is introduced.
- Second, we derive the necessary and sufficient conditions for a set of adversaries to form a strong barrier. These critical conditions can be used to understand whether or not the multi-path protocol family with XOR coding can securely penetrate the barrier with a high probability.
- Third, we propose a barrier penetration routing (BPR) protocol, where a small subset of legitimate nodes is randomly selected as reference points, called *beacons*. Through the distance vectors among the beacon nodes, a set of physically distanced paths between source and destination nodes is discovered by the network-wide flooding.
- Fourth, we conduct the computer simulations to compare the performance of the proposed BPR and a modified version of MPAR (M-MPAR) protocols with existing avoidance routing protocols, e.g., MPAR [8]. The simulation results demonstrate that the proposed scheme outperforms MPAR and its modified version in terms of secure delivery rate.

The rest of this paper is organized as follows: Related works are reviewed in Section 2. The problem of barrier penetration routing is formulated in Section 3. We derive critical conditions of a strong barrier in Section 4. The design challenges are discussed in Section 5. The baseline protocol, as well as the BPR protocol, are proposed in Sections 6 and 7, respectively. In Section 8, the performance of the proposed protocols is evaluated. Section 9 concludes this paper.

2 Related Work

2.1 Multi-Path Routing Protocols

Multi-path routing protocols, such as [16]–[19], discover a set of node/link disjoint paths for different design goals. The primary goal of multi-path routing is to improve the throughput by simultaneous message forwarding via multiple paths [16]. Another reason for setting up multiple paths is fault tolerance [17], [18], in which an additional path is used as the backup path when some nodes on the primary path go down. The optimal node/link disjoint path discovering is discussed in [19]. However, none of them provides security mechanisms.

2.2 Secure Routing Protocols

Secure routing protocols protect data privacy against eavesdropping by applying the secret sharing scheme [20] or the network coding [21]. However, their path discovery is not dedicated to identifying a set of safe paths for point-to-point communication. The theoretical works [22], [23] for secure ad hoc and sensor network routing have been studied to derive the secrecy capacity in a static network with eavesdroppers of known and unknown locations. The secure throughput when multi-path routing and the corporative jamming are applied is explored in [12]. However, an additional operational cost for transmission scheduling and energy consumption is introduced in the cooperative jamming, and such an approach is out of the scope of this research.

2.3 Avoidance Routing Protocols

Avoidance routing has been studied for different types of networks, such as the Internet, opportunistic networks, and WSNs. The protocols designed for the Internet [13]–[15] protect data privacy by avoiding the routers in opponent nations and/or internet service providers (ISPs). The idea of multi-path routing is applied for securing software defined networks [24]. Contact avoidance routing (CAR) in [11] is designed for contact-based opportunistic networks to avoid contact with malicious nodes. In addition, the concept of avoidance routing is applied to anonymous communications in [25], [26] and to privacy-preserving routing [27]. The most related work to this paper is the avoidance routing protocols for WSNs [6]–[10], which prevent adversaries with unbounded computational power from eavesdropping on messages. The single-path-based protocols, e.g., virtual positioning

TABLE 1
Summary of comparisons among related works.

Protocols	Multi-Path	Adversary deployment	Collusion Attacks	Adversaries' locations	Jamming
Proposed BPR	Yes	Strategic	Yes	No	No
VPSR [6]	No	Random	No	Required	No
Area Avoidance [7]	No	Random	No	Required	No
MPAR [8]	Yes	Random	No	Required	No
TMPAR [9]	Yes	Random	No	Required	No
MCAR [10]	Yes	Random	Yes	Required	No
CAR [11]	No	Random	No	Required	No
Jamming-based [12]	No	Random	Yes	No	Yes

source routing (VPSR) [6] and area avoidance routing [7], try to discover a safe path so that no adversary eavesdrops on wireless channels, under the assumption that the adversaries' locations are available at each node. However, there may exist no safe path, when too many adversaries are deployed in a network. Multi-path-based avoidance routing (MPAR) [8] and its variants, e.g., timer-based multi-path avoidance routing (TMPAR) [9] and multi-path-based collusion avoidance (MCAR) [10], significantly alleviate the condition of secure message delivery by combining multi-path forwarding and XOR coding. However, MPAR relies on the assumption of known adversaries' locations, does not consider strategic deployment of adversaries along a belt region, and cannot accommodate a collusion attack.

3 Problem Formulation

3.1 The Network Model

A WSN consists of a set of legitimate nodes and malicious nodes, called *adversaries*. Let v_i be node i, and the open neighbor set of v_i is a set of nodes, denoted by $N(v_i)$. Node v_j is in $N(v_i)$ if and only if v_i and v_j are within the communication range, denoted by r. A disk model is applied for our scenario, i.e., all the nodes have the same communication range and the communication area is circular. This is because the assumption of a disk model is essential to strengthen the theoretical aspect provided in Section 4. On the other hand, in reality, each node may have different transmission power, and the communication area could be oriented. Thus, the link layer assumption will be relaxed in our protocol designs, and we claim that the proposed BPR itself works under a bidirectional link model, where $v_i \in N(v_j)$ if and only if $v_j \in N(v_i)$ for all nodes v_i, v_j $(i \neq j)$.

The total number of nodes in a network is n. Among them, np nodes are adversaries. That is, p denotes the percentage of adversaries. For simplicity, we assume that p is set so that np is always an integer. Let $\mathcal{A}_j$ be adversary j. Adversary $\mathcal{A}_j$ can eavesdrop on the messages transmitted from any node in $N(\mathcal{A}_i)$. Node v_i is said to be in the *proximity* of $\mathcal{A}_j$, if $\mathcal{A}_j \in N(v_i)$; the set of adversaries in $N(v_i)$ is denoted by $N_{\mathcal{A}}(v_i)$, and $N_{\mathcal{A}}(v_i) \subseteq N(v_i)$ always holds. Note that the percentage of adversaries and the communication range are the function of n, i.e., they are formally denoted by $p(n)$ and $r(n)$. For simplicity, we write p and r as defined above. The notations used in this paper are listed in Table 2.

Similar to [1], [28], [29], asymptotic network models and asymptotic analyses are considered. We define *the target area* as a $\sqrt{s} \times \sqrt{s}$ square region and *a belt region*, which is formed by two parallel curves with orthogonal lines on two sides, which are defined by Definitions 1 and 2.

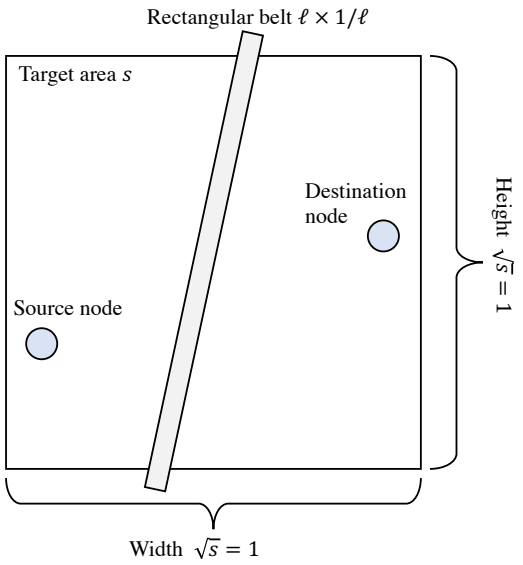


Fig. 2. An example of rectangular belts.

Definition 1 (Parallel curves) *Two curves, denoted by c_1 and c_2, are said to be parallel with separation $1/w$ if $d(x, c_2) = d(y, c_1) = 1/w$ for all points $x \in c_1$ and $y \in c_2$, where $d(\cdot, \cdot)$ indicates the distance between a point and a curve.*

Definition 2 (A belt region) *A belt region is defined by the area bounded by two parallel curves c_1 and c_2 with separation $1/w$.*

In this paper, we are particularly interested in a special case of a belt region, called *a rectangular belt* with dimension $\ell \times 1/\ell$. The definition of a rectangular belt is given in Definition 3.

Definition 3 (A rectangular belt) *A belt region of dimension $\ell \times 1/\ell$ is said to be a rectangular belt, where ℓ is the height and $1/\ell$ is the width of the region.*

A belt region is strategically located and cuts a given target area into two subareas, so that all the communications between nodes in different sides must go across the belt region, as shown in Figure 2. In our scenario, $n - np$ legitimate nodes and np adversaries are randomly deployed within a target area and a rectangular belt with dimension $\ell \times 1/\ell$, respectively, by a uniform distribution. Note that our analyses in Section 4 can be easily extended to the cases of the grid and the Poisson distributions of adversaries in a belt region. Furthermore, the proposed protocol works not only under a rectangular belt but also under a general belt region.

TABLE 2
Definition of notations.

Symbols	Definition
v_i, $\mathcal{A}_i$	Node i and adversary $\mathcal{A}_i$
n	The number of nodes
p	The adversary rate
$S_{s,d}$	A set of paths from v_s to v_d
$P_{s,d}^{(i)} \in S_{s,d}$	A path i between v_s and v_d
$N(v_i)$	The open neighbor set of v_i
G	A connected component of adversaries
r	The communication range
s, ℓ	Dimensions
m	A message
$Gen_u(.)$	A pseudorandom generator
$\oplus$	The XOR operator
q	The virtual adversary rate
η	The length of connected virtual adversaries
β	The percentage of beacon nodes
α	The number of anchor nodes
δ	The minimum distance between two anchors

3.2 The Adversary Model

In this paper, the encryption is assumed not to be a perfect solution. That is, an adversary can compromise the privacy of an encrypted message by eavesdropping. This assumption abstracts many real-world scenarios. For instance, a nation may spend a large amount of human and computing resources in a war - a typical example is the British intelligence in World War II [30]. Another possibility is that the implementation of cryptographic operations may have flaws [31]. In fact, many Secure Socket Layer (SSL) sessions on the Internet can be compromised due to invalid ways of generating prime numbers [32]. Furthermore, the longest key length of the advance encryption standard (AES) [33] that can be used in the United States is limited to 256-bit, so that law enforcement e.g., the Federal Bureau of Investigation (FBI) and the National Security Agency (NSA), can monitor encrypted data in the Internet.

The goal of adversaries in this paper is to compromise a message, denoted by m, which is initiated at source node v_s and destined to its destination node v_d. Clearly, any routing path must not contain any adversary as an intermediate node for secure delivery. Otherwise, an adversary not only obtains message m, but also may deny forwarding m to its destination. In addition, a routing path must avoid insecure areas, where adversaries are clandestinely hidden to eavesdrop on wireless channels. We define eavesdropping and the privacy of message m in Attack 1 and Definition 4, respectively.

Attack 1 (Eavesdropping) *Adversary $\mathcal{A}$ can eavesdrop on messages transmitted by any node v_i in $N(\mathcal{A})$.*

Definition 4 (Privacy of message m) *The privacy of message m is said to be compromised, if adversary $\mathcal{A}$ with unbounded computational power eavesdrops on m.*

In this paper, we consider *a collusion attack*, in which a group of adversaries collaborates with each other to compromise message m. Let G be a connected component of ζ adversaries, $\{\mathcal{A}_1, \mathcal{A}_2, ..., \mathcal{A}_\zeta\}$. The formal definition of the collusion attack is defined by Attack 2.

Attack 2 (Collusion attacks) *A group of adversaries, denoted by G, can collaborate with each other to compromise message m, if they are connected, i.e., for all $\mathcal{A}_i, \mathcal{A}_j \in G$ ($\mathcal{A}_i \neq \mathcal{A}_j$) there exists at least one path on the subgraph constructed from adversaries in G and the links among them. Such an attack is said to be a collusion attack.*

We assume that locally connected adversaries are able to collude each other, but they cannot use global information to form a barrier. If so, legitimate nodes can detect such active communications.

In [8], an adversary cannot compromise the original m unless she obtains all the k pieces, $m_1, m_2, \cdots, m_k$. This is because XOR coding behaves as Vernam's one-time pad, which achieves perfect secrecy as long as a secret key is used once, and the key, message, and ciphertext spaces are of the same size. However, the encoded messages are still susceptible to collusion attacks. Thus, we formally define the privacy of a set of encoded messages of m by XOR coding in Definition 5.

Definition 5 (Privacy of XORed message m) *The privacy of message m, where $m = m_1 \oplus m_2 \oplus ... \oplus m_k$, encoded by XOR coding, is said to be compromised, only if adversary $\mathcal{A}$ or a group of adversaries G eavesdrop on all of the m_1, m_2, ..., and m_k.*

3.3 The Barrier Penetration Routing Problem

Let v_s be the node which wishes to deliver message m to node v_d. The goal of routing is to securely route m from v_s to v_d across a belt region. Node v_i is said to be *safe*, if and only if $N_{\mathcal{A}}(v_i) = \emptyset$, where $\emptyset$ denotes an empty set. Otherwise, v_i's transmission of a message can be eavesdropped by one of the adversaries in $N_{\mathcal{A}}(v_i)$. In a single-path-based protocol, only safe nodes can be used as intermediate nodes. A routing path from v_s to v_d, denoted by $P_{s,d}$, consists of a list of legitimate nodes. The definition of a safe path is given in Definition 6.

Definition 6 (A safe path) *Path $P_{s,d}$ is said to be safe, if and only if $\bigcup_{\forall v_i \in P_{s,d}} N_{\mathcal{A}}(v_i) = \emptyset$.*

Even if there exists no safe path between v_s and v_d, the multi-path protocol family with XOR coding can securely deliver a message from v_s to v_d. To achieve this, a set of k paths must be *adversary disjoint*, i.e., there is no common adversary which can eavesdrop on all of the k paths. Let $S_{s,d}$ be a set of k paths from v_s to v_d and $P_{s,d}^{(i)}$ be the i-th path in $S_{s,d}$. The definition of a set of adversary disjoint paths is provided in Definition 7.

Definition 7 (A set of adversary disjoint paths) *A set of k paths, $S_{s,d}$, is said to be adversary disjoint, if and only if $\bigcap_{\forall P_{s,d}^{(i)} \in S_{s,d}} \{\bigcup_{\forall v_j \in P_{s,d}^{(i)}} N_{\mathcal{A}}(v_j)\} = \emptyset$.*

If there exists a set of adversary disjoint paths, then MPAR protocol family succeeds in securely delivering a message under the independent adversary model. However, this definition does not cover a collusion attack (Attack 2). Thus, we further introduce the notion of a set of *adversary group disjoint paths*. Let $N_G(v_i)$ be a set of adversary groups in $N(v_i)$, where $G_j \in N_G(v_i)$, if and only if $\exists \mathcal{A} \in N_{\mathcal{A}}(v_i)$ s.t. $\mathcal{A} \in G_j$.

Definition 8 (A set of adversary group disjoint paths) *A set of k paths, $S_{s,d}$, is said to be adversary group disjoint, if and only if $\bigcap_{\forall P_{s,d}^{(i)} \in S_{s,d}} \{\bigcup_{\forall v_j \in P_{s,d}^{(i)}} N_G(v_j)\} = \emptyset$.*

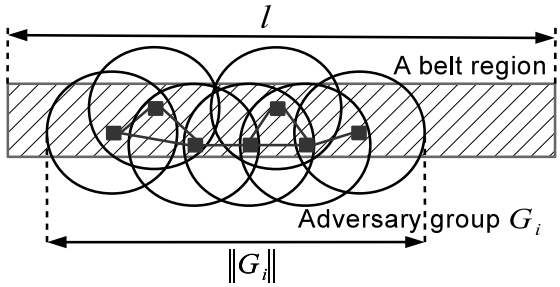


Fig. 3. The length of a connected component.

Note that independent adversaries can be seen as a special case of adversary groups. For instance, np independent adversaries can be mapped to a set of np adversary groups with each group G_i containing one adversary, i.e., $G_i = \{\mathcal{A}_i\}$. In addition, a safe path can be seen as a special case of a set of adversary disjoint paths, and we may write a set of paths with one element as $\{P_{s,d}^{(1)}\}$. Therefore, our problem formulation is of a more general form of the multi-path avoidance problem presented in [8].

Now, we can formally define the secure message delivery in the barrier penetration routing problem as follows.

Definition 9 (Secure message delivery) *With the one-time pad property, message m is said to be securely delivered from v_s to v_d, if m is encoded into m_1, m_2, ..., and m_k, where $m = m_1 \oplus m_2 \oplus ... \oplus m_k$, and each m_i ($1 \leq i \leq k$) is routed through a different adversary group disjoint path (Definition 8).*

3.4 The Notion of A Strong Barrier

Let G_i be connected component i consisting of adversaries in a belt region. Depending on the connectivity among adversaries, the number of adversary groups ranges from one to np. When there is only one connected component of adversaries, and the union of eavesdropping areas of the adversaries in the group covers the entire belt region, no adversary group disjoint path will be found. To formally define such a condition, we introduce the notion of a *strong barrier*. Let $\|G_i\|$ be the length of the barrier formed by adversary group G_i, i.e., the length of the union of the eavesdropping areas of the adversaries in the group, as shown in Figure 3. The barrier is strong if and only if there exists an adversary group whose length is greater than ℓ. If not, the barrier is said to be *weak*. The formal definition of a strong barrier is provided in Definition 10.

Definition 10 (A strong barrier) *A set of np adversaries deployed within a belt of dimension $\ell \times 1/\ell$ is said to form a strong barrier, if and only if there exists G_i such that $\|G_i\|$ is greater than or equal to ℓ.*

When a barrier is strong, the adversaries in the group eavesdrop on all the traffic across the barrier by colluding with each other. Such a condition indicates the performance bound of the barrier penetration routing. The critical conditions of a strong barrier are quantitatively analyzed in Section 4.

4 Critical Conditions

In this section, we derive two critical conditions, the necessary and sufficient conditions that a set of np adversaries in a belt of dimension $\ell \times 1/\ell$ forms a strong barrier that cuts a target area of dimension $\sqrt{s} \times \sqrt{s}$. These conditions illuminate whether the barrier of adversaries can be penetrated by a multi-path routing protocol with XOR coding. That is, if the necessary condition is not satisfied, then there exists at least one set of adversary disjoint paths with a high probability; if the sufficient condition is met, then there is a high probability that no adversary disjoint path exists. Otherwise, a set of adversary disjoint paths may or may not exist depending on how adversaries are deployed in the belt region. In addition, no routing protocol guarantees secure message delivery when adversaries locations are unknown, and thus, the critical conditions are particularly important for understanding the performance bound of the secure delivery rate of secure routing protocols.

4.1 The Necessary Condition

The probability of a barrier being strong is denoted by $\Pr[\exists G_i \text{ s.t. } \|G_i\| \geq \ell]$. The necessary condition can be obtained by deriving the condition that $\lim_{n\to\infty} \Pr[\exists G_i \text{ s.t. } \|G_i\| \geq \ell] = 1$ as presented in Theorem 1.

Theorem 1 *The necessary condition that a set of np adversaries forms a strong barrier in a belt of dimension $\ell \times 1/\ell$ is given by*

$$np \geq \frac{\ell}{r} \ln\left(\frac{\ell}{r}\right) + 1. \tag{1}$$

Proof: As shown in Figure 4, the belt of dimension $\ell \times 1/\ell$ is divided into a set of ℓ/r rectangles, denoted by R_1, R_2, ..., and $R_{\ell/r}$. Each R_j has dimension $r \times 1/\ell$. For the necessary condition to be satisfied, at least one adversary must be inside of R_j for each j ($1 \leq j \leq \ell/r$). Otherwise, any pair of two adversaries in R_{j-1} and R_{j+1} are disconnected. As a result, the length of the barrier L will be smaller than ℓ.

Let $E(R_j)$ be the event that at least one adversary is located in the j-th rectangle, R_j. Then, we have

$$\Pr[\exists G_i \text{ s.t. } \|G_i\| \geq \ell] \leq \Pr[\forall j, E(R_j)]. \tag{2}$$

Here, $1 \leq j \leq \ell/r$. By using the fact $(1-a)^b \leq e^{-ab}$, the probability that at least one adversary is inside of R_j for all j is computed as follows.

$$\begin{aligned}
\Pr[\forall j, E(R_j)] &= \bigcap_{j=1}^{\frac{\ell}{r}} \Pr[E(R_j)] && (3)\\
&= \left(\Pr[E(R_1)]\right)^{\frac{\ell}{r}} && (4)\\
&= \left[1 - \left(1 - \frac{r}{\ell}\right)^{np}\right]^{\frac{\ell}{r}} && (5)\\
&\leq \left[1 - e^{-\frac{rnp}{\ell}}\right]^{\frac{\ell}{r}} && (6)\\
&\leq \exp\left[-e^{-\frac{rnp}{\ell}} \cdot \frac{\ell}{r}\right] && (7)
\end{aligned}$$

For the left-hand side of Equation 2 to be 1 with a high probability, $\lim_{n\to\infty} \frac{r}{\ell} \cdot e^{\frac{rnp}{\ell}} = \infty$ must hold. Therefore, the necessary condition for a strong barrier is that the number of adversaries np is equal to or greater than $\frac{\ell}{r}\ln\left(\frac{\ell}{r}\right) + 1$. This concludes the proof. ∎

4.2 The Sufficient Condition

Next, we will derive the sufficient condition that a set of np adversaries forms a strong barrier with a high probability. Thus, we may derive the sufficient condition as follows.

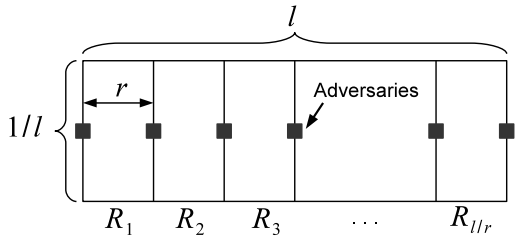


Fig. 4. The sub-regions of a belt 1.

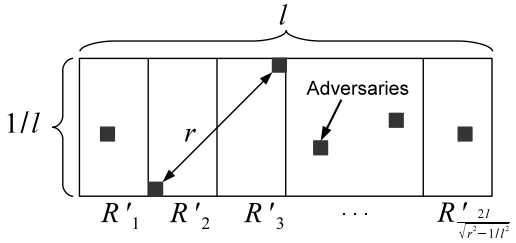


Fig. 5. The sub-regions of a belt 2.

Theorem 2 *The sufficient condition that a set of np adversaries forms a strong barrier in a belt of dimension $\ell \times 1/\ell$ is given by*

$$np \geq \frac{2\ell}{\sqrt{r^2 - 1/\ell^2}} \ln\left(\frac{2\ell}{\sqrt{r^2 - 1/\ell^2}}\right) + 1. \quad (8)$$

Proof: As shown in Figure 5, a belt of dimension $\ell \times 1/\ell$ is divided into $\frac{2\ell}{\sqrt{r^2-1/\ell^2}}$ rectangles of dimension $\frac{\sqrt{r^2-1/\ell^2}}{2} \times 1/\ell$, and the j-th rectangle is denoted by R'_j. Let $E(R'_j)$ be the event that at least one adversary exists inside of rectangle R'_j. We define $\Pr[\forall j, E(R'_j)]$ as the probability that at least one adversary is placed in each R'_j ($1 \leq j \leq \frac{2\ell}{\sqrt{r^2-1/\ell^2}}$). Assume that adversaries $\mathcal{A}_j$ and $\mathcal{A}_{j+1}$ are located within R'_j and R'_{j+1}, respectively. Since the distance between the left bottom point of R'_j and the right top point R'_{j+1} equals r, adversaries $\mathcal{A}_j$ and $\mathcal{A}'_{j+1}$ are connected no matter where they are located within R'_j and R'_{j+1}. Thus, we will have

$$\Pr[\forall j, E(R'_j)] \leq \Pr[\exists G_i \text{ s.t. } \|G_i\| \geq \ell]. \quad (9)$$

We seek to find the condition that satisfies $\Pr[\forall j, E(R'_j)] = 1$ for sufficiently large n. Similar to the argument in Theorem 1, $\Pr[\forall j, E(R'_j)]$ can be derived as follows.

$$\Pr[\forall j, E(R'_j)] \leq \exp\left[-e^{-\frac{np\sqrt{r^2-1/\ell^2}}{2\ell}} \cdot \frac{2\ell}{\sqrt{r^2 - 1/\ell^2}}\right] \quad (10)$$

For the left-hand side of Equation 9 to be 1 with a high probability, $\lim_{n\to\infty} \frac{\sqrt{r^2-1/\ell^2}}{2\ell} \cdot e^{\frac{np\sqrt{r^2-1/\ell^2}}{2\ell}} = \infty$ must hold. Therefore, the necessary condition for a strong barrier is that the number of adversaries np is equal to or greater than $\frac{2\ell}{\sqrt{r^2-1/\ell^2}} \ln\left(\frac{2\ell}{\sqrt{r^2-1/\ell^2}}\right) + 1$. This completes the proof. ∎

4.3 Analysis

The correctness of the critical conditions can be validated by simulations. Figure 6 illustrates the probability of a barrier being strong with respect to the number of adversaries np, where $\ell = 100$ and $r = 1$. The adversaries are randomly deployed within a belt of dimension $\ell \times 1/\ell$ by the uniform distribution. In this setting, the necessary condition is $np \geq 461$ and the sufficient condition is $np \geq 1060$. In fact, the probability of a barrier being strong is 0 and 1 with a high probability, when the necessary condition does not hold and when the sufficient condition holds, respectively.

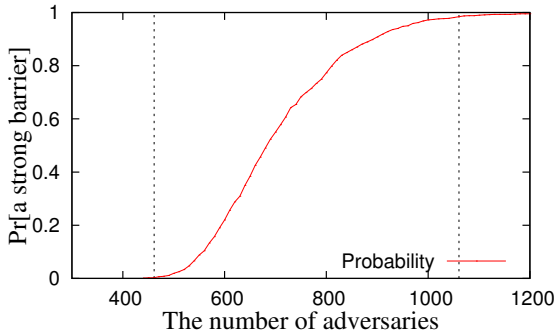


Fig. 6. The probability of a barrier being strong in a belt of dimension $100 \times 1/100$ with $r = 1$.

5 Design Challenges

The goal of this paper is to design a routing protocol that is able to securely penetrate a spy barrier formed by adversaries within a belt region. The design challenges that must be addressed are listed as follows.

- **Challenge 1:** To the best of our knowledge, in the existing avoidance routing [6]–[11], [13]–[15], either the physical location or topology information of adversaries is assumed to be known. That is, each node knows if there exist adversaries in its proximity. This assumption is too strong to accommodate in a real world scenario, where adversaries clandestinely listen to the wireless channel. Thus, the first challenge is how to discover safe paths without knowledge of the adversaries' locations.
- **Challenge 2:** In existing works, adversaries are assumed to be randomly deployed within the entire target area. In our scenario, adversaries are strategically deployed within a belt region, and all the communication must go across the belt region. This implies that there is no safe path between the source and destination nodes with a high probability. As a result, the existing single path-based avoidance protocols [6], [7] do not work at all. Hence, more than one path must be utilized in data forwarding.
- **Challenge 3:** The existing avoidance routing protocols are primarily designed to avoid individual adversaries. Thus, the paths discovered by the existing multi-path-based protocols with XOR coding, e.g., MPAR [8], do not attempt to physically diverge, and thus discovered paths are susceptible to the collusion attack. Therefore, the third challenge is how to discover a set of paths physically distanced from each other.

To address the aforementioned challenges, we first design a baseline protocol by modifying the existing MPAR [8] in Section 6 and then propose the barrier penetration routing (BPR) protocol in Section 7.

6 The Baseline Protocol

In this section, we propose the modified multi-path avoidance routing (M-MPAR) protocol, which tries to discover a set of paths

with no information about adversaries' locations, based on the existing MPAR [8]. The basic idea of M-MPAR is that a subset of legitimate nodes in a network are randomly selected as virtual adversaries, and the route discovery phase identifies a set of virtual adversary disjoint paths. By doing this, a path is intentionally split into multiple paths without knowing adversaries' locations.

The protocol parameters of M-MPAR include the maximal number of paths, the virtual adversary rate, and the length of the connected component of adversaries, which are denoted as k_{max}, q, and η, respectively. Each node, say v_i, switches its status to a virtual adversary with probability q, and η nodes are additionally selected by broadcasting an announce message from v_i. Thus, v_i is called the root. Announce message ANN is generated, including the node ID of the root, a set of virtual adversary IDs, and TTL. Each field is initialized by $ANN.root = v_i$, $ANN.SA = \{v_i\}$, and $ANN.ttl = \eta - 1$. Let v_j be the node who receives ANN from v_i. If $|ANN.SA \cap N(v_j)| \leq 1$, i.e., node v_j does not have any virtual adversary in its neighborhood except v_i, then v_j plays as a virtual adversary. When the announced message still has a positive TTL (time to live), i.e., $ANN.ttl \geq 1$, then v_j generates a new announce message, say ANN', with each field being $ANN'.root = ANN.root$, $ANN'.SA = \{v_j\} \cup ANN.SA$, and $ANN'.ttl = ANN.ttl - 1$. The announce message ANN' is sent to v_j's neighbors. This process continues until the TTL reaches zero.

M-MPAR tries to find a set of paths by avoiding virtual adversaries. The route discovery and data forwarding phases of M-MPAR are basically the same as the ones of the original MPAR. The source node v_s floods the network with a route request packet, $RREQ_1$. In the route reply phase, starting from the destination node v_d, the first path is set up. During the path construction, the virtual adversaries' IDs are included in the reply packet, $RPLY_1$. In addition, the intermediate nodes' IDs are included to $RPLY_1$ to make sure that at least two paths are initialized. Thus, M-MPAR differs from the original MPAR in that the ID of the root of connected virtual adversaries and the intermediate nodes' IDs are included in the first reply packet, instead of the real adversaries' IDs. When v_s receives $RPLY_1$, it tries to find another path, which shall be node disjoint. The second route request packet, $RREQ_2$ contains the set of virtual adversaries' IDs collected in the first reply packet. Thus, the network is flooded with $RREQ_2$ toward the destination by avoiding the virtual adversaries in $RREQ_1$ as well as the intermediate nodes in the first path. This process continues until v_s discovers a set of virtual adversary disjoint paths or the number of discovery processes reaches k_{max}. The pieces of information included in the request and reply packets are basically the same as the original MPAR [8].

By doing this, the design challenges are addressed as follows. Clearly, M-MPAR never uses the information about adversaries' locations (Challenge 1). The introduction of virtual adversaries forces M-MPAR to set up multiple paths away from each other (Challenge 2). By selecting a set of nodes to form a connected component of virtual adversaries of length η, the discovered paths shall be physically apart from each other (Challenge 3).

7 Barrier Penetration Routing

7.1 The Basic Idea

The proposed barrier penetration routing (BPR) protocol relies on both the distance vectors and the flooding-based path discovery. The BPR protocol consists of four phases, the beacon selection, anchor nodes selection, route discovery, and message forwarding phases as follows.

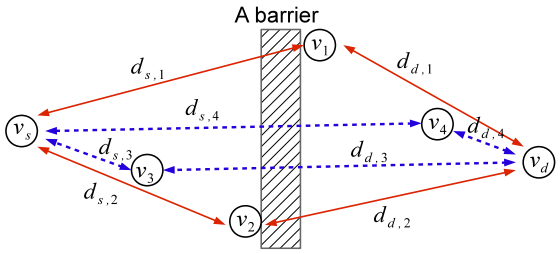


Fig. 7. The idea of BPR.

In the beacon selection phase, a small set of nodes in a network are randomly selected as reference points, called *beacon nodes*. Then, each beacon exchanges their distance vectors, and the other nodes compute its distance to each beacon. Here, the distance between two nodes is defined by the smallest number of hops. In the anchor nodes selection phase, α ($\alpha \geq 2$) nodes among the beacons are selected as anchor nodes. Note that the number of paths discovered by BPR will be α. To this end, a source node first identifies two beacons such that the distance between them is at least δ hops away as shown in Figure 7. When $\alpha > 2$, additional anchor nodes are selected so that the distance between each pair of two anchors is at least $\lfloor \frac{\delta}{\alpha - 1} \rfloor$ hops away. Note that this beacon setup phase is performed in a proactive fashion and will not be executed for individual message transmissions.

The route discovery of BPR consists of the route request and reply stages, which is performed in an on-demand fashion. The source node floods the network with a route request packet to discover a path to the anchor nodes, and then, a path from each anchor node to the destination. The routing table at intermediate nodes is set up in the route reply stage in the reverse order from the destination. When the source node receives the reply packets from the destination, message m is encoded into α messages, m_1, m_2, ..., and m_α by XOR coding. Each m_i ($1 \leq i \leq \alpha$) is forwarded via a different path toward the corresponding anchor node and then toward the destination. The destination node assembles the original m by collecting all the encoded pieces. By doing this, colluding adversaries obtain no information about m unless they eavesdrop on all pieces.

7.2 The Beacon Selection

Let v_s be the source node who wishes to securely deliver message m to node v_d. The protocol parameters include the percentage of beacon nodes β, the number of anchors α, and the minimum distance between two anchors δ. While the values of α and δ can be optimized with knowledge of the number of adversaries, we assume that neither the number of adversaries nor their locations are available. Thus, α and δ are considered as protocol parameters. The initialization of beacon nodes as well as the distance vector among nodes are completed by the beacon vector routing proposed in [34]. The distance from v_i to v_j is denoted by $d_{i,j}$. In addition to distance vectors, each node maintains its routing table.

7.3 The Anchor Nodes Selection

For given protocol parameters, v_s initializes the anchor nodes as follows. We introduce the distance range that orders pairs of two integers in order to define a better pair of anchor nodes. For example, in Figure 7, v_1 and v_2 are better anchor nodes than v_3 and v_4, since the paths via v_1 and v_2 are physically

distanced with respect to the barrier. Let (v_x, v_y) and (v_w, v_z) be two pairs of beacons, both of which satisfy $d_{x,y} \geq \delta$ and $d_{w,z} \geq \delta$. The inequality $(|d_{s,x} - d_{d,x}|, |d_{s,y} - d_{d,y}|) \leq (|d_{s,w} - d_{d,w}|, |d_{s,z} - d_{d,z}|)$ holds, i.e., (v_x, v_y) is a better beacon pair than (v_w, v_z) with respect to (v_s, v_d), if and only if either $|d_{s,x} - d_{d,x}| \leq |d_{s,w} - d_{d,w}| \wedge |d_{s,y} - d_{d,y}| \leq |d_{s,z} - d_{d,z}|$ or $\neg(|d_{s,x} - d_{d,x}| \leq |d_{s,w} - d_{d,w}| \wedge |d_{s,y} - d_{d,y}| \leq |d_{s,z} - d_{d,z}|) \wedge (|d_{s,x} - d_{d,x}| + |d_{s,y} - d_{d,y}| \leq |d_{s,w} - d_{d,w}| + |d_{s,z} - d_{d,z}|)$. Here, $\neg$ indicates the negation. The tie can be broken by the minimum ID among v_x, v_y, v_w, and v_z.

All pairs of two beacon nodes with the distance greater than or equal to δ are identified, and such a set is denoted by V_δ. Among V_δ, the two beacons, say v_i and v_j, with the smallest distance range, i.e., $\min_{\forall v_i, v_j \in V_\delta} (|d_{s,i} - d_{d,i}|, |d_{s,j} - d_{d,j}|)$ are selected as the anchor nodes. Let v_1 and v_2 be the first two selected anchor nodes. When $\alpha > 2$, additional $\alpha - 2$ anchor nodes labeled by v_3, v_4, ..., v_α are selected as follows. For each i, a set of beacons V'_δ such that $d_{s,i} \geq (i-2) \times \lfloor \frac{\delta}{\alpha - 1} \rfloor$ and $d_{d,i} \geq \delta - (i-2) \times \lfloor \frac{\delta}{\alpha - 1} \rfloor$ are identified. Then, the beacon with the minimum distance range, $\min_{\forall v_i, \in V'_\delta} (|d_{s,i} - d_{d,i}|, |d_{s,j} - d_{d,j}|)$, is added to the anchor list.

7.4 The Route Request Phase

After the beacon nodes which serve as anchors $L_{anch} := \{v_1, v_2, .., v_\alpha\}$ are determined, the route discovery process is started. A routing entry consists of five fields, the path ID $rt.pid$, the source ID $rt.src$, the destination ID $rt.dst$, the predecessor node ID $rt.prev$, and the descendant node ID $rt.next$. Each field of the routing entry is set during the request and reply phases.

The request packet, denoted as $RREQ$, contains the mode, the path ID, the source ID, the destination ID, and a list of the anchor node IDs, which are denoted by $RREQ.mode$, $RREQ.pid$, $RREQ.src$, $RREQ.\ dst$, and $RREQ.L_{bcn}$, respectively. The mode can be either the toward-anchor mode *ANCH* or toward-destination mode *DST*. At v_s, each field is set to be $RREQ.mode \leftarrow$ *ANCH*, $RERQ.src \leftarrow v_s$, $RREQ.pid \leftarrow 0$, $RREQ.dst \leftarrow v_d$, and $RREQ.L_{anch} \leftarrow \{v_1, v_2, .., v_\alpha\}$, respectively. Then, v_s floods the network with $RREQ$.

The route request phase starts with the toward-anchor mode. Let v_i be the node which receives $RREQ$ from v_j. If v_i has already seen $RREQ$ before and $RREQ.pid = 0$, it discards $RREQ$. First, v_i checks if its ID is included in $RREQ.L_{anch}$, and if not, v_i is not an anchor node and broadcasts $RREQ$. At this time, v_i creates a new entry in its routing table and initializes each field of the entry (i.e., $rt.pid$, $rt.src$, and $rt.dst$) based on the corresponding field of $RREQ$. Note that $rt.prev$ is set to be the sender's ID, i.e., v_j, but $rt.next$ is null at this moment. The descendant ID is determined in the route reply phase. In addition, $rt.pid$ is equal to 0, since the path ID has not been determined yet. In the case that v_i is an anchor, i.e., $v_i \in RREQ.L_{bcn}$, the mode of the request packet switches to *DST*. Note that changing the mode from *ANCH* to *DST* guarantees that no more than two anchor nodes are used as the intermediate nodes on the same path. The path ID is set to be the index of v_i in $RREQ.L_{bcn}$. Thus, $1 \leq RREQ.pid \leq \alpha$. Then, the anchor node, v_i, floods the entire network with $RREQ$ whose mode is *DST*. At this time, v_i creates a new entry rt in its routing table by setting each field based on $RREQ$.

The toward-destination mode of the route request phase works as follows. When v_i receives $RREQ$ from v_j, v_i checks if it has already seen $RREQ$ with the corresponding $RREQ.pid$. If not, $RREQ$ is discarded. Note that the request packet is uniquely identified by the three fields, pid, src, and dst. Hence, if two request packets have different path IDs, then they are considered to be different request packets. Otherwise, v_i creates a new routing entry. Each entry is initialized based on $RREQ$. If v_i is not the destination, then v_i continues the request phase by broadcasting $RREQ$. Since there will be α anchor nodes, the destination eventually receives α request packets with the path ID being $1 \leq pid \leq \alpha$. Note that if the destination receives $RREQ$ with $RREQ.pid = 0$, the packet will be discarded as it has not reached any of the anchor nodes. When $RREQ$ reaches v_d, the route reply phase will start.

In the proposed BPR, the source node, as well as each anchor node, floods the network with a request packet with different pid being 0, 1, ..., or α. Thus, the number of request packets introduced in BPR is bounded by $O((\alpha + 1)(n - np))$.

Algorithm 1 BPR(v_s, v_d, m, α, δ)

/* Initialization: the source node v_s does the following. */
v_s selects anchor nodes, $L_{anch} := \{v_1, v_2, ..., v_\alpha\}$, among V_δ.
v_s creates $RREQ := (\textit{ANCH}, 0, v_s, v_d, L_{anch})$.
v_s broadcast $RREQ$.
/* On receiving $RREQ$ from v_j, v_i does following. */
if (v_i has seen $RREQ$ before) **then**
  v_i drops $RREQ$.
if ($RREQ.mode =$ *ANCH*) **then**
  v_i creates $rt := (0, RREQ.src, RREQ.dst, v_j, -1)$.
  if ($v_i \in RREQ.L_{bcn}$) **then**
    $k \leftarrow$ the index of v_j in $RREQ.L_{bcn}$.
    $rt.pid \leftarrow k$
    $RREQ.pid \leftarrow k$, $RREQ.mode \leftarrow$ *DST*.
    v_i broadcasts $RREQ$.
else if ($RREQ.mode =$ *DST*) **then**
  v_i creates $rt := (RREQ.pid, RREQ.src, RREQ.dst, v_j, -1)$,
  if ($v_i = RREQ.v_d$) **then**
    $v_k \leftarrow$ the $RREQ.pid$-th node in $RREQ.L_{bcn}$.
    v_d creates $RPLY(\textit{ANCH}, RREQ.pid,, RREQ.src, RREQ.dst, v_k)$.
    v_d sends $RPLY$ to $rt.prev$.
  else
    v_i broadcasts $RREQ$.
/* On receiving $RPLY$ from v_j, v_i does following. */
if (v_i has seen $RPLY$ before) **then**
  v_i drops $RPLY$.
if ($RPLY.mode =$ *ANCH*) **then**
  if (there exits rt s.t. $rt.pid = RPLY.pid$, $rt.src = RPLY.src$, and $rt.dst = RPLY.dst$) **then**
    $rt.next \leftarrow v_j$.
    if ($v_i = RPLY.anchor$) **then**
      $RPLY.mode \leftarrow$ *SRC*.
    v_i broadcasts $RPLY$.
  else if ($RPLY.mode =$ *SRC*) **then**
    if (there exits rt s.t. $rt.pid = 0$, $rt.src = RPLY.src$, and $rt.dst = RPLY.dst$) **then**
      v_i duplicates rt and let rt' be the copy.
      $rt'.pid \leftarrow RPLY.pid$, $rt.next \leftarrow v_j$.
      v_i sends $RPLY$ to $rt'.prev$.
      if ($v_i = RPLY.src$ and has α routing entries) **then**
        The path discovery succeeds.

7.5 The Route Reply Phase

The reply packet, denoted as $RPLY$, contains the mode, the path ID, the source ID, the destination ID, and the corresponding anchor node ID (denoted by $RREQ.anch$). In the reply phase, the mode of a reply packet can be either the toward-anchor mode *ANCH* or the toward-source mode *SRC*. At destination node v_d, $RPLY.mode$ is set to be *ANCH*. The other fields, $RPLY.pid$, $RPLY.src$, $RPLY.dst$, $RPLY.anch$, are initialized based on the corresponding request packet. Note that $RPLY.anch$ is found in the $RREQ.pid$-th node ID in $RREQ.L_{anch}$. The reply packet is forwarded toward the corresponding anchor node and then toward the source node along the predecessor node, whose ID is stored at $rt.prev$. Let v_i be the node who receives $RPLY$ from node v_j. Let k be the path ID of $RPLY$. Node v_i shall have a routing entry with $rt.pid = 0$ or $rt.pid = k$. If $rt.pid = k$, then v_i is an intermediate node between anchor v_k and destination v_d. The descendant node ID kept at $rt.next$ in the v_i's routing table is set to be the sender of $RPLY$, i.e., $rt.next \leftarrow v_j$. Then, v_i forwards $RPLY$ to the node with its ID being $rt.prev$.

If the v_i's ID equals $RPLY.anch$, i.e., v_i is the corresponding anchor node for $RPLY$, $RPLY.mode$ is switched to the *SRC* mode. The descendant node ID in the v_i's routing table is set to be v_j. Then, $RPLY$ is forwarded toward the source node via $rt.prev$. If v_i receives $RPLY$ with the mode being *SRC*, v_i is an intermediate node between the anchor and the source nodes. In this case, v_i should have the routing entry with $rt.pid = 0$, since the path ID is initialized at one of the anchor nodes in the route request phase. The corresponding routing entry is duplicated and $rt.pid$ is set to be $RPLY.pid$. In addition, the descendant node ID $rt.next$ is set to be v_j. Then, v_i forwards $RPLY$ toward v_s via $rt.prev$. If v_i is the source node, it updates the routing entry in the same way as the intermediate nodes. This process continues until the source node receives α reply packets.

7.6 The Data Forwarding Phase

At the end of the route request and reply phases, the source node v_s has α routing entries with each having different path IDs. Message m is encoded into α pieces, m_1, m_2, ..., m_α, where $\forall i, |m_i| = |m|$. First, v_s computes $\alpha - 1$ random strings, m_1, m_2, ..., and $m_{\alpha-1}$, by $Gen_u(.)$. Then, the last piece is obtained by $m_\alpha \leftarrow m \oplus m_1 \oplus m_2 \oplus ... \oplus m_{\alpha-1}$. Each encoded message, m_i, is sent toward v_d based on the routing table with $rt.pid = i$. When v_d receives all the pieces, v_d assembles them and obtains the original m.

7.7 Complexity

The time complexity of BPR with respect to the number of legitimate nodes, $n - np$, is quantified as follows. Let b be the number of beacon nodes, which is computed by $b = \beta(n - np)$. Then, the set up overhead will be $O\left(b(n - np)\right)$. The route discovery consists of anchor selection, request, and reply phases, each of which takes $O(1)$, $O\left((\alpha + 1)(n - np)\right)$, and $O(\alpha h)$, where h is the network diameter. Thus, the time complexity of the route discovery is bounded by $O\left((\alpha + 1)(n - np)\right)$.

Let $\tilde{N}$ be the network density, i.e., the average number of neighbors, $\tilde{N} = \frac{1}{n-np} \sum_{\forall v_i} N(v_i)$. Since each node maintains the information about the beacon nodes and its neighbors, the per-node state complexity is $O(\tilde{N} + b)$.

The message complexity can be derived as follows. The beacon selection phase introduces $O(bn)$ control messages. The route discovery phase incurs $O(n^2)$ message overhead, in which request packets are flooded in the entire network, i.e., $O(n^2)$ message complexity, and the reply packets are introduced for each path, i.e., $O(kn) = O(n)$ message complexity. The message complexity of the data forwarding phase is $O(kn) = O(n)$. Therefore, the message complexity of the proposed BPR will be $O(n^2)$.

Remark (Cost vs. security trade-off): One may be concerned about the trade-off between the cost (the number of adversary disjoint paths) and the secure delivery rate. As illuminated by the existing works [8], [9] as well as the simulation results in Section 8.2, having two paths is enough for secure delivery in most cases and introducing more than three paths does not increase secure delivery rate much.

8 Performance Evaluation

The BPR and M-MPAR protocols are implemented for performance evaluation. Since there is no avoidance routing protocol without location information of adversaries, we compared the proposed schemes with the original MPAR [9].

8.1 Simulation Configuration

Unless specified, a sensor network consisting of 1000 nodes is generated with the number of adversaries ranging from 5 to 40. Thus, $n = 1000$, $p \in [0.5\%, 4\%]$, and $np \in [5, 40]$. When specified, the number of nodes ranges from 250 to 2000 with the number of adversaries being 10. The legitimate nodes are randomly placed within a 1000 by 1000 square region, and the adversaries are randomly deployed within an open belt of dimension 10 by 1000. Note that in Section 4, we define the dimension of a belt as of form $\ell \times 1/\ell$ for asymptotic analysis. In our simulations, the width of a belt is set to be a much larger value than $1/\ell$. In addition, the mobility of adversaries is not considered, which is a typical case of wireless sensor networks. The communication range r is set to be 100 so that the network scale is sufficiently large. A pair of source and destination nodes are randomly located at the left and right edges of the target region, and the belt orthogonally crosses the target area so that all the packets between source and destination pair must travel the belt.

The protocol-dependent parameters are set to be as follows. Among the legitimate nodes, 5% to 10% of the nodes are selected as beacons. The number of anchor nodes α for BPR is set to be either 2, 3, 4, or 5. The minimum distance between/among anchors δ ranges from 5 to 12. For M-MPAR, the number of virtual adversaries is set to be five, and the length of connected virtual adversaries η ranges from 5 to 12. Unless specified otherwise, these parameters are set to be $\beta = 0.05$, $\alpha = 2$, $\delta = 10$, and $\eta = 10$, respectively. For both the original MPAR and M-MPAR protocols, the value of k_{max} is set to be five, i.e., their route discovery phase tries to find up to five adversary disjoint paths.

To evaluate the performance of protocols, four metrics are considered as follows: First, the delivery rate is defined as the ratio of data packets securely delivered to a destination, in the sense that the privacy of data under collusion attacks (Attack 2) is preserved as defined in Definition 5, and the total number of data packets transmitted from a source node. Second, hop stretch is defined by the ratio between the number of hops of the longest path among discovered paths and the shortest hop count. Denoting the shortest path between v_s and v_d by $SP_{s,d}$, the hop stretch is obtained by $\frac{\max_{\forall P^{(i)}_{s,d} \in S_{s,d}} |P^{(i)}_{s,d}|}{|SP_{s,d}|}$. Third, the control overhead of

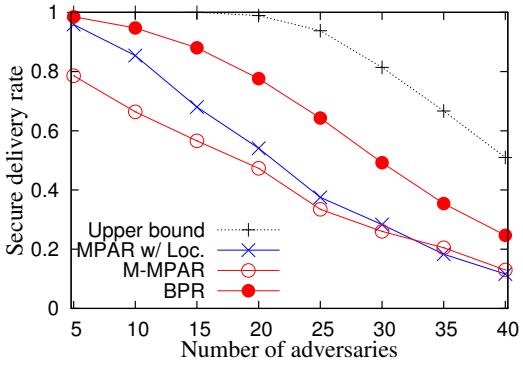


Fig. 8. The delivery rate vs. the number of adversaries.

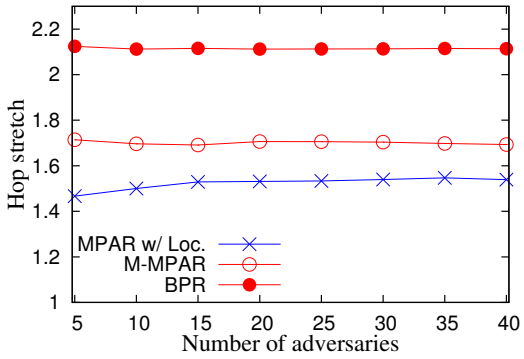


Fig. 9. The hop stretch vs. the number of adversaries.

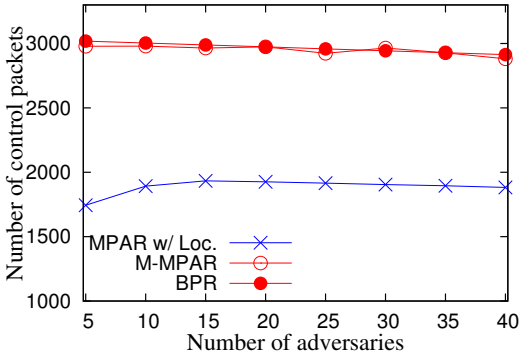


Fig. 10. The number of control packets vs. the number of adversaries.

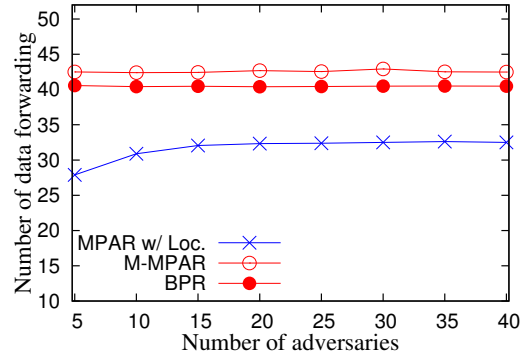


Fig. 11. The number of data forwarding vs. the number of adversaries.

a protocol is defined by the number of route request and reply packets to discover a set of paths. Fourth, the amount of traffic incurred by a routing protocol is quantified by the number of data packet forwardings, which can be computed by the summation of the hop counts of a set of k paths, i.e., $\sum_i^k |P_{s,d}^{(i)}|$.

8.2 Simulation Results

Figure 8 shows the delivery rate of different protocols with respect to the number of adversaries. The upper bound indicates the theoretically achievable performance bound. In other words, if there exists at least one set of adversary disjoint paths, a multi-path-based protocol with XOR coding possibly penetrates the barrier. Note that there may not exist a set of adversary disjoint paths, even when the necessary condition does not hold. This is because the necessary condition holds for $n \to \infty$, but there is a bounded number of nodes, $n = 1000$, in our simulations. For instance, in our simulation, the necessary condition is $np \geq 23$, but the delivery rate of the upper bound is slightly smaller than 1.0 when $np = 20$. As can be seen in the figure, the performance bound gradually decreases as the number of adversaries increases. BPR and M-MPAR do not rely on the assumption of known adversaries' locations, while MPAR is executed using the information of adversaries' locations. Nevertheless, BPR presents the highest delivery rate. To be specific, the delivery rate of BPR is higher than that of MPAR by at least 30%, when the number of adversaries is greater than or equal to 15. Since MPAR will discover a set of adversary disjoint paths under the independent adversary attack model, a set of discovered paths is likely to be susceptible to a collusion attack. M-MPAR, unfortunately, results in the lowest delivery rate when the number of adversaries is smaller than or equal to 20. However, we would like to claim that the M-MPAR protocol works with a much weaker assumption than the original MPAR protocol does in the sense that the adversaries' locations are assumed to be unknown.

Figure 9 illustrates the hop stretch of different protocols with respect to the number of adversaries. M-MPAR incurs slightly higher hop stretch than MPAR does, but the difference is not significant. On the other hand, BPR requires much higher hop stretch for any number of adversaries. The primary reason is that the anchor nodes must be physically apart from each other to securely penetrate the barrier. As a result, the paths which BPR discovers are likely to be longer than the shortest path. Considering the higher delivery rate of BPR, the additional number of hops is not a primary issue for secure data communications.

Figure 10 presents the number of control packets of different protocols with respect to the number of adversaries. The control overheads of each packet for BPR, M-MPAR, and MPAR are roughly $O\left((\alpha+1)(n-np)\right)$, $O\left(k(n-np)\right)$, and $O\left(k(n-np)\right)$, respectively. Thus, most of the cases of MPAR perform network-wide flooding two times. On the other hand, BPR performs exactly three network-wide floodings when two anchors are used. This is the primary reason why BPR introduces more control overhead than MPAR does. However, the number of anchors α is given as a protocol parameter, and thus, BPR always floods the network $(\alpha+1)$ times. On the other hand, the number of paths k that M-MPAR and MPAR discover can be up to k_{max}, which is set to be five in our simulations. Therefore, in the worst case scenario, M-MPAR and MPAR incur more control overhead than BPR.

Figure 11 depicts the number of data forwarding of different protocols with respect to the number of adversaries. As can be seen in the figure, BPR and M-MPAR introduce a larger number of data forwarding than the original MPAR does. As a rule of thumb, a safer set of paths tends to be longer, which results in a greater number of data forwarding. Thus, BPR achieves a higher delivery rate by introducing a larger amount of overhead.

Figure 12 illuminates the delivery rate of different protocols with respect to the number of nodes. The delivery rate of all

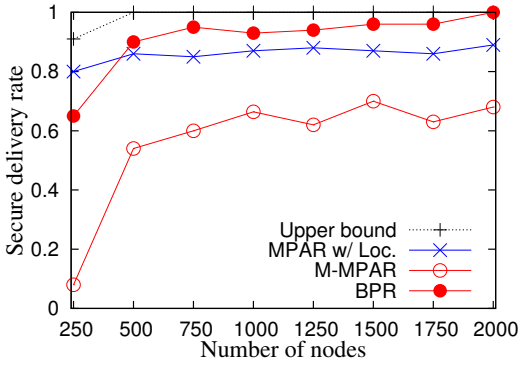


Fig. 12. The delivery rate vs. the number of nodes.

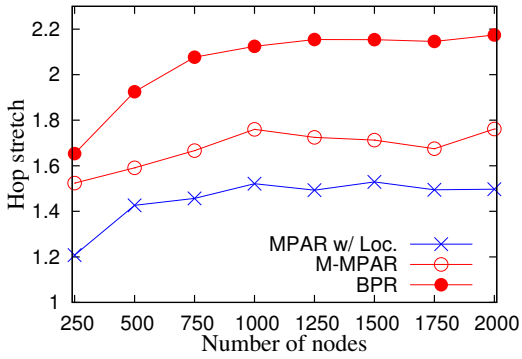


Fig. 13. The hop stretch vs. the number of nodes.

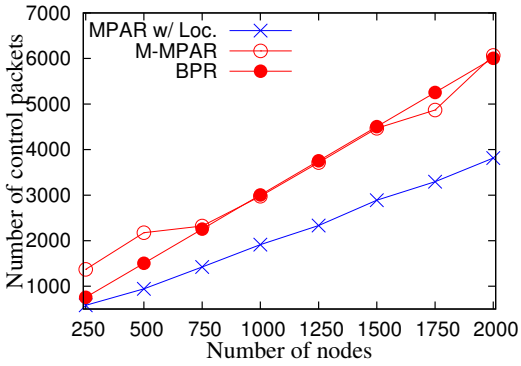


Fig. 14. The number of control packets vs. the number of nodes.

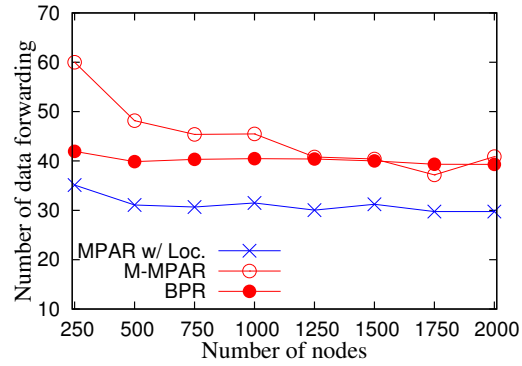


Fig. 15. The number of data forwarding vs. the number of nodes.

the protocols increases as the number of nodes increases. This is because more legitimate nodes in a network leads to more relay nodes that can avoid adversaries. When the number of nodes equals 250, the proposed BPR results in a small delivery rate and even the upperbound is less than 0.9. This indicates a set of adversaries disconnect source and destination nodes. However, the delivery rates of BPR are more than 0.9 for $n \geq 500$ and higher than those of M-MPAR and MPAR. Hence, BPR works well in dense networks.

Figure 13 shows the hop stretch of different protocols with respect to the number of nodes. Note that only the successful routing processes are considered when computing the hop stretch. For $n = 250$, routing succeeds when a source node and a destination node are close to each other, i.e., a discovered path tends to be short. On the contrary, a longer safe path and/or a set of longer safe paths can be found when $n \geq 500$. As a result, the hop stretch of BPR increases when the number of nodes increases from 250 to 500 and then converges when the number of nodes becomes sufficiently large. While our BPR incurs the largest hop stretch, considering the achievable secure delivery rate presented in Figure 12, the introduction of additional number of hops is worth it.

Figure 14 illustrates the number of control packets of different protocols with respect to the number of nodes. The number of control packets of all the protocols linearly increases as the number of nodes increases. This is because a network is flooded with request packets in their route discovery phase. BPR and M-MPAR incur a similar amount of control overhead. On the contrary, MPAR introduces lower control overhead, since an additional k-path discovery process kicks in only when a single safe path is not found. However, we claim that MPAR relies on the assumption of known adversary locations, while BPR and M-MPAR do not.

Figure 15 presents the amount of data forwarding with respect to the number of nodes. In general, there will be more safe paths and/or a set of safe paths in a network when there exist a large number nodes. As a result, the amount of data forwarding of all the nodes slightly decreases when the number of nodes increases. Our BPR incurs more data forwarding than MPAR does, but the difference is not significant. In addition, M-MPAR incurs more data forwarding than BPR when the number of nodes is fewer than or equal to 1000. This is because, in MPAR, some nodes must play as virtual adversaries, and thus fewer nodes can be used for data forwarding. As a result, a discovered path tends to be detour routes. From Figures 12 to 15, we conclude that the proposed BPR can accommodate sufficiently large networks.

Figure 16 gives the delivery rate of the BPR protocol with respect to the beacon rate. In general, the more beacon nodes that exist in a network, the better opportunities to find a better set of anchors, but at the same time, a larger set up cost will be introduced for computing distance vectors. From the figure, the delivery rate remains stable for any number of adversaries, even when the beacon rate increases. This implies that selecting 5 percent of nodes as beacons is sufficient for constructing distance vectors.

Figure 17 shows the minimum distance between two anchor nodes of the BPR protocol with respect to the number of adversaries. Essentially, having the longer distance between two anchors accommodates the longer length of a spy barrier. Therefore, increasing the minimum distance between two anchors leads to a higher delivery probability. On the other hand, if the distance is too long, the route discovery phase may fail to find anchors due to the limited dimension of a simulation area. The figure clearly shows this intuition. The delivery rate gradually increases when the minimum distance increases from 5 to 10 hops. Then, the delivery rate becomes stable, when the minimum distance is greater than or equal to 10 hops. We conclude that the minimum distance between two anchors being 10 is sufficient.

Figure 18 illustrates the delivery rate of the BPR protocol

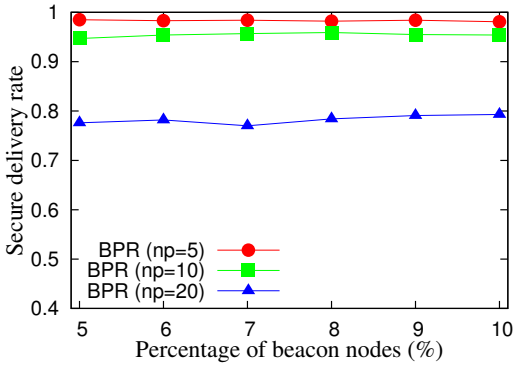


Fig. 16. The delivery rate vs. the beacon rate.

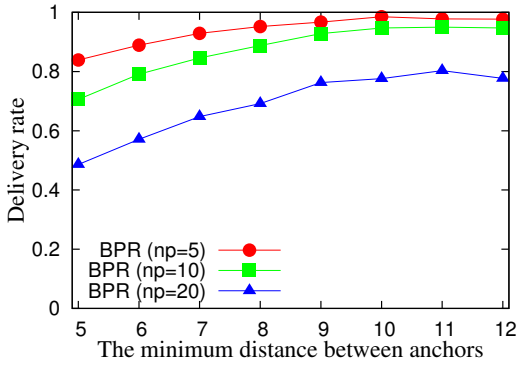


Fig. 17. The delivery rate vs. the minimum distance between anchors.

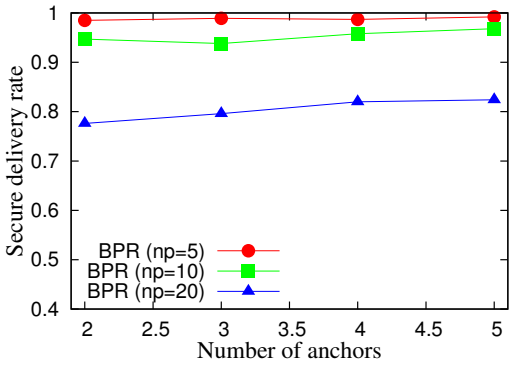


Fig. 18. The delivery rate vs. the number of anchors.

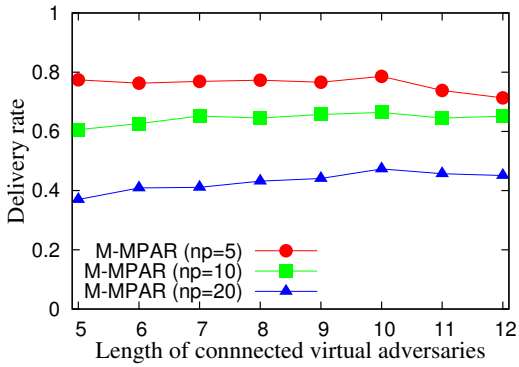


Fig. 19. The delivery rate vs. the length of connected virtual adversaries.

with respect to the number of anchors. The figure indicates that the delivery rate gradually increases, as the number of anchors increases. A set of colluding adversaries must collect all the pieces from all the paths between the source and destination nodes, and thus, the probability that all encoded data packets are compromised decreases by using a greater number of anchors. However, significant improvement is not observed, even when five anchors (i.e., five paths) are used. On the other hand, the asymptotic analysis of control overhead of BPR, $O\left((\alpha+1)(n-np)\right)$, implies that the number of control packets introduced in the route discovery phase increases when the number of anchors increases. Thus, considering the additional cost for increasing the number of anchors, the use of two anchors is reasonable for penetrating the barrier.

Figure 19 presents the length of connected virtual adversaries of the M-MPAR protocol with respect to the number of adversaries. Each component of virtual adversaries behaves as a barrier, and M-MPAR tries to make multiple paths as far away from each other as possible. The intuition is that the longer the length of connected virtual adversaries, the higher the delivery rate is. However, if the length is too long, the route discovery phase of M-MPAR may fail to find adversary disjoint paths. This intuition is clearly observed in the figure, where the delivery rate increases when the length of connected virtual adversaries increases from 3 to 6. Then, the delivery rate slightly decreases when the length of connected virtual adversaries increases from 10 to 12.

9 Conclusion

In this paper, we first introduce a new secure routing problem in WSNs, namely barrier penetration routing, against strategically deployed wireless sensors with eavesdropping capability. Then, the necessary and sufficient conditions of a strong barrier of adversaries are derived. These conditions can be used to understand whether or not multi-path avoidance routing with XOR coding succeeds with a high probability. Our routing protocol design differs from the existing avoidance routing problem in the sense that the adversaries' locations are assumed to be unknown, that adversaries are strategically deployed along a belt region, and that a set of paths should be distanced to avoid a possible collusion attack. For secure communication in such a critical scenario, we propose the barrier penetration routing (BPR) protocol that uses beacon vectors as well as flooding for path discovery. The simulation results demonstrate that BPR achieves its design goals.

References

[1] S. Kumar, T. H. Lai, and A. Arora, "Barrier Coverage With Wireless Sensors," in *ACM MobiCom*, Gologne, Germany, Aug. 28.–Sep. 2, 2005, pp. 284–298.

[2] Y. Wang and G. Cao, "Barrier Coverage in Camera Sensor Networks," in *ACM MobiHoc*, Paris, France, May 15-19, 2011, pp. 12:1–12:10.

[3] A. Chen, Z. Li, T. Lai, and C. Liu, "One-Way Barrier Coverage with Wireless Sensors," in *IEEE INFOCOM*, Shanghai, China, Apr. 11-15, 2011, pp. 626–630.

[4] C.-F. Cheng and C.-W. Wang, "The Target-Barrier Coverage Problem in Wireless Sensor Networks," *IEEE Trans. Mob. Comput.*, vol. 17, no. 5, pp. 1216–1232, May 2017.

[5] S. Roy, N. Ghosh, and S. K. Das, "bioSmartSense: A Bio-inspired Data Collection Framework for Energy-efficient, QoI-aware Smart City Applications," in *IEEE PerCom*, Kyoto, Japan, Mar. 11-15, 2019, pp. 1–10.

[6] H. Zlatokrilov and H. Levy, "Navigation in Distance Vector Spaces and Its Use for Node Avoidance Routing," in *IEEE INFOCOM*, Anchorage, AL, USA, May 6-12, 2007, pp. 1253–1261.

[7] ——, "Area Avoidance Routing in Distance-Vector Networks," in *IEEE INFOCOM*, Phoenix, AZ, USA, April 13-17, 2008, pp. 475–483.

[8] K. Sakai, M.-T. Sun, W.-S. Ku, J. Wu, and T. H. Lai, "Multi-Path-Based Avoidance Routing in Wireless Networks," in *IEEE ICDCS*, Columbus, OH, USA, Jun. 29 – July 2, 2015, pp. 706–715.

[9] ——, "Secure Data Communications in Wireless Networks Using Multi-Path Avoidance Routing," *IEEE Trans. Wireless Commun.*, vol. 18, no. 10, Oct. 2019.

[10] Z. Zhang and J. Wang, "MCAR: Multi-Path-Based Collusion Avoidance Routing for Wireless Ad-Hoc Networks," in *ICNCC*, Taipei, Taiwan, Dec. 13-16, 2018, pp. 177–181.

[11] T. Osuki, K. Sakai, and S. Fukumoto, "Contact Avoidance Routing in Delay Tolerant Networks," in *IEEE INFOCOM*, Atlanta, GA, USA, May 1-4, 2017, pp. 1–9.

[12] C. Capar, D. Goeckel, B. Liu, and D. Towsley, "Secret Communication in Large Wireless Networks Without Eavesdropper Location Information," in *IEEE INFOCOM*, Orlando, FL, USA, Mar. 25–30, 2012, pp. 1152–1160.

[13] E. Kline and P. Reiher, "Securing Data Through Avoidance Routing," in *NSPW*, Oxford, United Kingdom, Sep. 8-11, 2009, pp. 115–124.

[14] D. Levin, Y. Lee, L. Valenta, Z. Li, V. Lai, C. Lumezanu, N. Spring, and B. Bhattacharjee, "Alibi Routing," in *ACM SIGCOMM*, London, United Kingdom, Aug. 17-21, 2015, pp. 611–624.

[15] A. Edmundson, R. Ensafi, N. Feamster, and J. Rexford, "A First Look into Transnational Routing Detours," in *ACM SIGCOMM*, Florianópolis, Brazil, 2016, pp. 567–568.

[16] A. Tsirigos and Z. J. Haas, "Analysis of Multipath Routing-Part I: The Effect on The Packet Delivery Ratio." *IEEE Trans. Wireless Commun.*, vol. 3, no. 1, pp. 138–146, Jun. 2004.

[17] X. Zhang, X. Dong, J. Wu, X. Li, and N. Xiong, "Fault-Aware Flow Control and Multi-Path Routing in Wireless Sensor Networks," in *IEEE ICDCS Workshops*, Philadelphia, PA, USA, Aug. 8-11, 2013, pp. 27–32.

[18] X. Zhang, X. Dong, N. Xiong, J. Wu, and X. Li, "Fault-Aware Flow Control and Multi-Path Routing in VANETs," *Peer Peer Netw. Appl.*, vol. 8, no. 6, pp. 1090–1107, Sep. 2015.

[19] J. Yallouz, O. Rottenstreich, P. Babarczi, A. Mendelson, and A. Orda, "Optimal Link-Disjoint Node-"Somewhat Disjoint" Paths," in *IEEE ICNP*, Singapore, Nov. 8-11, 2016, pp. 1–10.

[20] W. Lou, W. Liu, Y. Zhang, and Y. Fang, "SPREAD: Improving Network Security by Multipath Routing in Mobile Ad Hoc Networks," *Wireless Netw.*, vol. 15, no. 3, pp. 279–294, Apr. 2009.

[21] N. Cai and R. W. Yeung, "Secure Network Coding on a Wiretap Network," *IEEE Trans. Inf. Theory*, vol. 57, no. 1, pp. 424–435, Jan. 2011.

[22] O. O. Koyluoglu, C. E. Koksal, and H. E. Gamal, "On Secrecy Capacity Scaling in Wireless Networks," *IEEE Trans. Inf. Theory*, vol. 58, no. 5, pp. 3000–3015, May 2012.

[23] S. Vasudevan, D. Goeckel, and D. F. Towsley, "Security-Capacity Trade-off in Large Wireless Networks Using Keyless Secrecy," in *ACM MobiHoc*, Chicago, IL, USA, Sep. 20–24, 2010, pp. 21–30.

[24] G. Liu, W. Quan, N. Cheng, N. Lu, H. Zhang, and X. Shen, "P4NIS: Improving Network Immunity Against Eavesdropping with Programmable Data Planes," in *IEEE INFOCOM Workshop*, Virtual Event, Jul. 6–9, 2020, pp. 91–96.

[25] Z. Li, S. Herwig, and D. Levin, "DeTor: Provably Avoiding Geographic Regions in Tor," in *USENIX Security*, Vancouver, BC, Canada, Aug. 16–18, 2017, pp. 343–359.

[26] K. Kohls, K. Jansen, D. Rupprecht, T. Holz, and C. Pöpper, "On the Challenges of Geographical Avoidance for Tor," in *USENIX NDSS*, San Diego, CA, USA, Feb. 24–27, 2019.

[27] B. Sengupta, Y. Li, K. Bu, and R. H. Deng, "Privacy-Preserving Network Path Validation," *ACM Trans. Internet Technol.*, vol. 20, no. 1, pp. 5:1–5:27, Feb. 2020.

[28] P. Gupta and P. R. Kumar, "Critical Power for Asymptotic Connectivity in Wireless Networks," *Stochastic Analysis, Control, Optimization and Applications: A Volume in Honor of W.H. Fleming*, pp. 547–566, 1998.

[29] S. Kumar, T. Lai, and J. Balogh, "On k-Coverage in A Mostly Sleeping Sensor Network," in *ACM MobiCom*, Philadelphia, PA, USA, Sep. 26 – Oct. 1, 2004, pp. 144–158.

[30] S. Singh, *The Code Book: The Evolution of Secrecy from Mary, Queen of Scots, to Quantum Cryptography*. Doubleday, 1999.

[31] P. Gutmann, "Lessons Learned in Implementing and Deploying Crypto Software," in *USENIX Security Symposium*, San Francisco, CA, USA, Aug. 5-9, 2002.

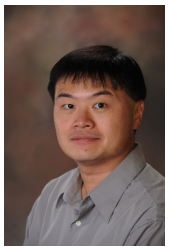
[32] R. Nojima, T. Kurokawa, and S. Moriai, "XPIA, X.509 Certificate Public-key Investigation and Analysis System," *NICT News, Dec. 2013*, no. 435, pp. 5–6, 2013.

[33] "Advanced Encryption Standard (AES)," https://csrc.nist.gov/csrc/media/publications/fips/197/final/documents/fips-197.pdf, Accessed: July 28, 2018.

[34] R. Fonseca, S. Ratnasamy, J. Zhao, C. T. Ee, D. Culler, S. Shenker, and I. Stoica, "Beacon Vector Routing: Scalable Point-to-Point Routing in Wireless Sensornets," in *USENIX NSDI*, Boston, MA, USA, May 2-5, 2005, pp. 329–342.

Kazuya Sakai (S'09-M'14) received his Ph.D. degree in Computer Science and Engineering from The Ohio State University in 2013. He is currently an associate professor at the Department of Electrical Engineering and Computer Science, Tokyo Metropolitan University. His research interests are in the area of information and network security, wireless and mobile computing, and distributed algorithms. He received the IEEE Computer Society Japan Chapter Young Author Award 2016. He is a member of the IEEE and ACM.

Min-Te Sun (S'99-M'02) is a professor in the Department of Computer Science and Information Engineering, National Central University, Taiwan. He received the BSc degree from National Taiwan University, the MSc degree from Indiana University, Bloomington, and the PhD degree in Computer and Information Science from The Ohio State University. His research interests include distributed computing and IoT. He is a member of the IEEE and ACM.



Wei-Shinn Ku (S'02-M'07-SM'12) received his Ph.D. degree in computer science from the University of Southern California (USC) in 2007. He also obtained both the M.S. degree in computer science and the M.S. degree in electrical engineering from USC in 2003 and 2006, respectively. He is a professor with the Department of Computer Science and Software Engineering at Auburn University. He was a Program Director with the National Science Foundation between 2019 and 2022. His research interests include databases, data science, mobile computing, and cybersecurity. He has published more than 160 research papers in refereed international journals and conference proceedings. He is a senior member of the IEEE and a member of the ACM.

Jie Wu is Laura H. Carnell Professor at Temple University and the Director of the Center for Networked Computing (CNC). He served as Chair of the Department of Computer and Information Sciences from the summer of 2009 to the summer of 2016 and Associate Vice Provost for International Affairs from the fall of 2015 to the summer of 2017. Prior to joining Temple University, he was a program director at the National Science Foundation and was a distinguished professor at Florida Atlantic University, where he received his Ph.D. in 1989. His current research interests include mobile computing and wireless networks, routing protocols, network trust and security, distributed algorithms, applied machine learning, and cloud computing. Dr. Wu regularly published in scholarly journals, conference proceedings, and books. He serves on several editorial boards, including IEEE Transactions on Service Computing and Journal of Computer Science and Technology. Dr. Wu is/was general chair/co-chair for IEEE DCOSS09, IEEE ICDCS13, ICPP16, IEEE CNS16, WiOpt21, ICDCN22, IEEE IPDPS'23, and ACM MobiHoc'23 as well as program chair/cochair for IEEE MASS04, IEEE INFORCOM11, CCF CNCC13, and ICCCN20. He was an IEEE Computer Society Distinguished Visitor, ACM Distinguished Speaker, and chair for the IEEE Technical Committee on Distributed Processing (TCDP). Dr. Wu is a Fellow of the AAAS and a Fellow of the IEEE. He is the recipient of the 2011 China Computer Federation (CCF) Overseas Outstanding Achievement Award. He is a Member of the Academia Europaea (MAE).