A Graph-theoretic Framework for Identifying Trigger Nodes against Probabilistic Reactive Jamming Attacks

Ying Xuan, Yilin Shen, Incheol Shin, My T. Thai

Abstract—During the last decade, *Reactive* Jamming Attack has emerged as a greatest security threat to wireless sensor networks, due to its mass destruction to legitimate sensor communications and difficulty to be disclosed and defended. Considering the specific characteristics of reactive jammer nodes, a new scheme to deactivate them by efficiently identifying all *trigger nodes*, whose transmissions invoke the jammer nodes, has been proposed and developed. Such a trigger identification procedure could serve as a leverage subroutine for a jamming-resistent routing scheme and exhibit great potentials in enhancing the defense efficiency. By modeling this procedure as a graph optimization problem, in this paper, on the one hand, we employed an advanced *randomized error-tolerant non-adaptive group testing* technique and a classic *clique-independent set* problem to further speed up the identification process, compared to our previous work. On the other hand, by investigating two sophisticated jamming behavior models, we proposed an efficient algorithm which limits the identification false rates to desirable low levels. The theoretical analysis and simulation results illustrate the robustness and efficiency of the proposed solution.

Index Terms—Trigger Identification, Clique-Independent Set, Error-tolerant Nonadaptive Group Testing, Graph Theory, Optimization, NP-Hardness.

1 Introduction

Since the last decade, the security of wireless sensor networks (WSNs) has attracted numerous attentions, due to its wide applications in various monitoring systems and invulnerability toward sophisticated wireless attacks. Among these attacks, jamming attack where a jammer node disrupts the message delivery of its neighboring sensor nodes with interference signals, has become the most critical threat to WSNs. Thanks to the efforts of researchers toward this issue, as summarized in [11], various efficient defense strategies have been proposed and developed. However, a reactive variant of this attack, where jammer nodes stay quite until an ongoing legitimate transmission (even has a single bit) is sensed over the channel, emerged recently and called for stronger defending system and more efficient detection schemes.

Existing countermeasures against Reactive Jamming attacks consist of jamming (signal) detection and jamming mitigation. On the one hand, detection of interference signals from jammer nodes is non-trivial due to the discrimination between normal noises and adversarial signals over unstable wireless channels. Numerous attempts to this end monitored critical communication related objects, such as *Receiver Signal Strength* (RSS), *Carrier Sensing Time* (CST), *Packet Delivery Ratio* (PDR), compared the results with specific thresholds, which were established from basic statistical methods and multi-modal strategies [8][11]. By such schemes, jamming signals could be discovered, however, how to locate and catch the jammer nodes based on these signals is much more complicated and has not been settled. On the other hand, in order to mitigate these attacks, two strategies were adopted at sensor nodes to escape from the detected interferences, namely, channel surfing and spatial retreats [11]. The former one employs frequency hopping techniques at both communication ends [5][7][9], in which case jammer nodes are unable to find the current channel that is used for the communication, so that the attack efficiency is greatly decreased. The latter one requires sensor nodes to retreat from the possible jammed areas, then no sensor nodes will be effected by the jamming signals [10][12]. However, owing to the limited power and spectral diversity [8] of wireless sensors, these mitigation schemes are inefficient due to their considerable computation and communication overheads.

Instead of discovering the jammed areas, which may be inaccurate and unnecessarily large, we proposed a new solution to mitigate the attacks by identifying the *trigger nodes* in [6], whose transmissions invoke the jammer nodes, and preventing these trigger nodes from transmitting messages. Specifically, we provided a novel jamming-resistent routing scheme with regulating all identified trigger nodes as terminals, therefore no messages will be transmitted from the trigger nodes and

• *Y. Xuan, Y. Shen, I. Shin and My T. Thai are with the Department of Computer Information Science and Engineering.*
E-mail: {yxuan, yshen, ishin, mythai}@cise.ufl.edu

This is an extended version of "Y. Xuan, Y. Shen, I. Shin, M. T. Thai, "On Trigger Detection Against Reactive Jamming Attacks: A Clique-Independent Set Based Approach", IPCCC, Phoenix, Arizona, 2009."

all jammer nodes will stay quiet. The motivation of studying this trigger identification problem is not limited in this reactive jamming scenario, but to provide a solution framework to the mitigations against the reactive variant of a general scope of attacks.

Since the performance of the trigger identification is critical for the routing scheme and other applications that make benefit from it, in this paper, we develop a solution for real sophisticated attack scenarios. Specifically, as many attackers play tricks to evade detections, the feedbacks of jammer nodes toward sensed message transmissions can be non-deterministic or along with randomized time delays. To handle these unsure factors, we introduce a novel *randomized error-tolerant group testing scheme*, which is combined with a clique-independent set model, and speeds up the identification procedure with low error rates under unreliable enviroments. The basic idea of our solution is to partition the victim nodes (which are interfered by jamming signals) into multiple *testing teams*, and then conduct *group testing* based on the constructed randomized (d, z)-disjunct matrix, by letting each victim node broadcast test messages simultaneously and some *leader* nodes gather the feedbacks (interference signals) from the jammers. By generating *test outcomes* from these gathered feedbacks, all the trigger nodes are identified via a prompt decoding process. Compared with our previous work [6], more sophisticated jammer behaviors are considered and handled in this paper by the new group testing scheme, with lower time and communication complexity, as well as accuracy guarantees. Moreover, we model the partitioning phase of victim nodes as a *clique-independent set* problem, whose NP-Hardness on UDG (unit disk graph) is shown.

In the remainder of this paper, we first present the problem definition in Section 2, where the network model, victim model and attacker models are included. Then we introduce two kernel techniques for our scheme, *clique-independent set* and *randomized error-tolerant non-adaptive group testing* in Section 3. The core of this paper: *trigger identification procedure* and its error-tolerant extension toward sophisticated jammer behaviors are presented respectively in Section 4 and 5. A series of simulation results for evaluating the system performance and validating the theoretical results are included in Section 6. We also present some related works in Section 7 and summarize the whole paper in Section 8.

2 Problem Models and Definition

2.1 Network Model

We consider a wireless sensor network consisting of n sensor nodes and one base station (larger networks with multiple base stations can be split into small ones to satisfy the model). Each sensor node has a uniform transmission radius r and is equipped with m radios for in total k channels throughout the network, where $k > m$. The network can abstracted as a *unit disk graph* (UDG) $G = (V, E)$, where any node pair i, j is connected iff the Euclidean distance between i, j: $\delta(i, j) \leq r$.

2.2 Victim Model

Victim nodes refer to those sensor nodes whose transmissions are disturbed by jamming signals, i.e., node v is a victim node *iff* $\delta(J, v) \leq R$ for some *activated* jammer J. In this paper, we assume that each sensor can identify received jamming signals and justify whether itself is a victim node. Furthermore, the results of these self-identifications are reported to the base station by means of the existing message forwarding schemes periodically, therefore the set of victim nodes is maintained at the base station. Since the detection of jamming signals have been well developed with multi-modal statistical methods, the above assumptions are feasible even in unreliable environments.

As a subset of the victim nodes, *trigger nodes* refer to a subset of victim nodes, whose transmissions activate the jammer nodes. In another word, node v is a trigger node *iff* $\delta(J, v) \leq r$ for some activated jammer J. Therefore the problem studied in this paper is to identify all the trigger nodes from a given set of victim nodes.

2.3 Attacker Model

We consider both a basic attacker model and several advanced attacker models in this paper. In the next sections, we will first illustrate our framework solution toward the basic attacker model, and then validate its performance toward multiple advanced attacker models theoretically and experimentally.

2.3.1 Basic Attacker Model

The basic attacker model is defined as follows: there exists at most $J \ll n$ reactive jammer nodes in the network, whose transmission radiuses are $R = \alpha r$ with $\alpha > 1$. These jammer nodes keep idle until they sense any ongoing legitimate transmissions and broadcast interference signals to jam all the sensors in distance R on this specific channel. The maximum damages caused by the jammer nodes are limited to the interferences toward specific sensor nodes on specific transmission channels for a short period, instead of long-term disabling the sensors. The motivation behind this assumption arises from the basic goal of reactive jamming: disrupt the message delivery with minimum energy cost. As soon as the sensors detect any jamming signals, the transmissions will be terminated, or continue on some other channels. Thus it is unnecessary for the jammer nodes to keep sending interference signals on this channel for a long time, or either disrupt all the channels with large energy overheads as an active jammer does. Moreover, from the standpoint of the attacker, it will be a waste to deploy two jammer nodes too close to each other, thus we assume that for any two jammer nodes J_1 and J_2, $\delta(J_1, J_2) \geq 2R - R'$ with a small overlap R' such that $R' \leq R - r$ (see Fig. 3 in the next section).

2.3.2 Advanced Attacker Models

Considering possible adjustments at the jammer nodes to evade the detection, we take into account two probabilistic

attacker models: *probabilistic attack* and *variant response time delay*. In the first one, the jammer responds each sensed transmission with a probability η independently. Practically, η is approaching 1, to guarantee the attack efficiency. However, in order to validate the accuracy of our solution toward extreme cases, we also consider small η in the theoretical analysis and simulations. In the other model, the jammer delays each of its jamming signals with an independently randomized time interval. Similarly, too large delays do not make sense for practical attacks, but our solution is also satisfiable under these extreme cases.

It is evident that most tricks in reactive jamming attack can be abstracted into either of these two models. Therefore, showing the efficiency of our identification toward such models suffices validating its applicability to practical defense systems and unreliable WSN environments.

3 Two Kernel Techniques

This section includes two advanced techniques which benefit our identification procedure. We first provide the NP-hardness proof of the *Clique-Independent Set* problem along with a simple approximation algorithm, then introduce the *randomized error-tolerant group testing* by providing our novel design for randomized (d, z)-disjunct matrix.

3.1 Clique-Independent Set

Cliques-Independent Set is the problem to find a set of maximum number of pairwise vertex-disjoint maximal cliques, which is referred as a *maximum clique-independent set* (MCIS) [4]. Since this problem serves as the abstracted model of the *grouping* phase of our identification, its hardness is of great interest in this scope. To our best knowledge, it has already been proved to be NP-hard for cocomparability, planar, line and total graphs, however its hardness on UDG is still an open issue.

3.1.1 NP-hardness

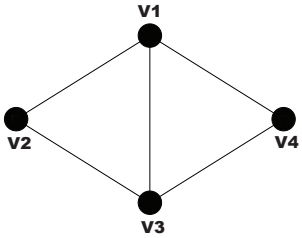
In this section, we prove the NP-hardness of this problem on UDG via a polynomial-time reduction from the *Maximum Independent Set* problem on planar graph with maximum node degree 3 to it.

From [20], the *Maximum Independent Set* problem is NP-hard on planar graph with maximum degree 3, and from [21], any planar graph G with maximum degree 4 can be embedded in the plane using $O(|V|^2)$ area units such that its vertices are at integer coordinates and its edges consist of line segments of the form $x = i$ or $y = j$, for any integers i and j.

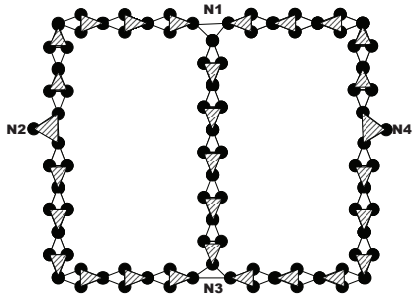
Theorem 3.1: Clique-Independent Set problem is NP-hard on Unit Disk Graph.

Proof: Given an instance $G' = (V', E')$ of such a *MIS* problem, whose optimal value is denoted as $MIS(G')$, we construct an instance $G = (V, E)$ of the *CIS* problem as follows:

- Embed G' in the plane in the way mentioned above [21].
- For each node $v_i \in V'$, attach two new nodes v_{i1} and v_{i2} to it and form a triangle $N_i = \{v_{i1}, v_{i2}, v_{i3}\}$, where each edge of this triangle N_i is of a unit length $r = \frac{\sqrt{3}}{3}$.
- Since each nodes v_i is incident to at most three edges, for all edges $(v_i, u), \cdots, (v_i, v)$, move their endpoint from v_i to different v_{ij}s, e.g., (v_1, u) changes to (v_{11}, u) and (v_1, v) to (v_{12}, v). Afterwards, for each of such edges $e = (u, v)$, assume that it is of length t, we divide it into t pieces and replace each piece with a concatenation of 2 triangles (not necessarily equilateral), as shown in Fig. 1(b). Therefore, any edge $e_{ij} = (v_i, v_j) \in E'$ of length $|e_{ij}|$ becomes a concatenation of $2|e_{ij}|$ 3-cliques, denoted as $\{c_{ij}^{1,1}, c_{ij}^{1,2}, c_{ij}^{2,1}, \cdots c_{ij}^{|e_{ij}|,1}, c_{ij}^{|e_{ij}|,2}\}$. Because of the triangles N_is, the two triangles at each corner of Fig. 1(b) may need slight stenches, which can be done in polynomial time.
- The resulting graph G is then a unit disk graph with radius $r = \frac{\sqrt{3}}{3}$.



(a) Instance $G' = (V', E')$ of MIS problem on planar graph with maximum degree 3



(b) Instance $G = (V, E)$ with radius $\frac{\sqrt{3}}{3}$ of Clique-Independent Set problem on UDG

Fig. 1. Polynomial Time Reduction

The reduction is as follows:
($\Rightarrow$): if G' has a maximum independent set M, for each $u_i \in M$, we choose cliques of two kinds in the corresponding instance G: (1) the clique N_i at u_i; (2) for each incident edge $e_{ij} = (u_i, u_j)$, choose cliques $\{c_{ij}^{1,2}, c_{ij}^{2,2}, c_{ij}^{3,2}, \cdots, c_{ij}^{|e_{ij}|,2}\}$. Since the clique N_j at u_j shares a vertex with $c_{ij}^{|e_{ij}|,2}$, it cannot be selected. For any edge $e_{jk} = (u_j, u_k)$ where $u_j \notin M$ and $u_k \notin M$, choose cliques $\{c_{jk}^{1,2}, c_{jk}^{2,2}, \cdots c_{jk}^{|e_{jk}|,2}\}$. It is easy to

verify that all the cliques selected are vertex-disjoint from each other.

Assume that after embedding G' into the plane, each node $v_i \in V'$ has coordinate (x_i, y_i), then edge length $|e_{ij}| = \| v_i, v_j \|_1 = |x_i - x_j| + |y_i - y_j|$. Therefore if we have an independent set of size $|M| = k$ for G', we then have a clique independent set of size $k' = k + \sum_{(i,j)\in E'} |e_{ij}|$.

($\Leftarrow$): if G has a clique independent set of size k', since the lengths of the embedded edges are constant, then G' has exactly an independent set of size $k = k' - \sum_{(i,j)\in E'} |e_{ij}|$. The proof is complete. □

3.1.2 Algorithms

There have been numerous polynomial exact algorithms for solving this problem on graphs with specific topology, e.g., Helly circular-arc graph and strongly chordal graph [4], but none of these algorithms gives the solution on UDG. In this paper, we employ the *scanning disk approach* in [3] to find all maximal cliques on UDG, and then find all the *MCIS* using a greedy algorithm. In fact, by abstracting this problem as a *Set Packing* problem, we can obtain a $\sqrt{n}$-approximation algorithm, however, it exhibits worse performance than the greedy algorithm proposed in our trigger identification procedure.

3.2 Error-tolerant Randomized Non-Adaptive Group Testing

Group Testing was proposed since WWII to speed up the identification of affected blood samples from a large sample population. This scheme has been developed with a complete theoretical system and widely applied to medical testing and molecular biology during the past several decades [1]. Notice that the nature of our work is to identify all triggers out of a large pool of victim nodes, so this technique intuitively matches our problem.

3.3 Traditional Non-adaptive Group Testing

The key idea of group testing is to test items in multiple designated groups, instead of testing them one by one. The traditional method of grouping items is based on a designated 0-1 matrix $M_{t\times n}$ where the matrix rows represent the testing group and each column refers to an item, as Fig. 2 shows. $M[i, j] = 1$ implies that the j^{th} item appears in the i^{th} testing group, and 0 otherwise. Therefore, the number of rows of the matrix denotes the number of groups tested in parallel and each entry of the result vector V refers to the test outcome of the corresponding group (row), where 1 denotes positive outcome and 0 denotes negative outcome.

Given that there are at most $d < n$ positive items among in total n ones, all the d positive items can be efficiently and correctly identified on the condition that the testing matrix M is d-disjunct: any single column is not *contained* by the union of any other d columns. Owing to this property, each negative item will appear in at least one row (group) where all the positive items do not show up, therefore, *by filtering all the items appearing in groups with negative outcomes, all the left ones are positive*. Although providing such simple decoding method, d-disjunct matrix is non-trivial to construct [1][2] which may involve with complicated computations with high overhead, e.g., calculation of irreducible polynomials on Galois Field. In order to alleviate this testing overhead, we advanced the deterministic d-disjunct matrix used in [6] to randomized error-tolerant d-disjunct matrix, i.e., a matrix with less rows but remains d-disjunct w.h.p. Moreover, by introducing this matrix, our identification is able to handle test errors under sophisticated jamming environments.

$$M = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \end{bmatrix} \overset{\text{testing}}{\Longrightarrow} V = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

Fig. 2. Binary testing matrix M and testing outcome vector V. Assumed that item 1 (1^{st} column) and item 2 (2^{nd} column) are positive, then only the first two groups return negative outcomes, because they do not contain these two positive items. On the contrary, all the other four groups return positive outcomes.

3.4 Error-tolerant Randomized Designs

In order to handle errors in the testing outcomes, the *error-tolerant non-adaptive* group testing has been developed using (d, z)-disjunct matrix, where in any $d+1$ columns, each column has 1 in at least z rows where all the other d columns are 0. Therefore, a $(d, 1)$-disjunct matrix is exactly d-disjunct. By use of (d, z)-disjunct matrix, we can still correctly identify d positive items, even in the presence of at most $z-1$ test errors. A prompt decoding scheme by differentiating positive and negative items based on their number of appearances in groups with negative outcomes is summarized in [1]: considering any single positive item i and negative item j. Suppose there are c negative groups containing i, then these c groups (tests) have errors, hence there are at most $z - 1 - c$ other groups turning negative outcomes to positive outcomes. Due to the definition of (d, z)-disjunct matrix, column j appears in at least z negative groups where none of the d positive items exist, so even $z-1-c$ of these groups are turned into positive ones, the number of negative groups containing j is at least $z - (z - 1 - c) = c + 1 > c$. It is evident that by sorting all the suspected items by their number of appearances in negative groups, those d items with smallest number of appearances are positive.

In the literature, one the one hand, numerous deterministic designs for (d, z)-disjunct matrix have been provided [1],

however, these constructions often suffer from high computational complexity, thus are not efficient for practical use and distributed implementation. On the other hand, to our best knowledge, the only randomized construction for (d, z)-disjunct matrix dues to Cheng's work via q-nary matrix [19], which results in a (d, z)-disjunct matrix of size $t_1 \times n$ with probability p', where

$$t_1 = 4.28d^2 \log \frac{2}{1-p'} + 4.28d^2 \log n + 9.84dz + 3.92z^2 \ln \frac{2n-1}{1-p'}$$

with time complexity $O(n^2 \log n)$. Compared with this work, we advance a classic randomized construction for d-disjunct matrix, namely, random incidence construction [1][2], to generate a (d, z)-disjunct matrix which can not only generate comparably smaller $t \times n$ matrix, but also handle the case where z is not known beforehand, instead, only the error probability of each test is bounded by some constant γ. Although z can be quite loosely upperbounded by γt, yet t is not an input. The motivation of this construction lies in the real test scenarios, the error probability of each test is unknown and asymmetric, hence it is impossible to evaluate z before knowing the number of pools.

We only show the performance of this new construction, namely, ETG algorithm in this section. For the review purpose, we include the details of the construction and proofs in the Appendix.

Theorem 3.2: ETG algorithm produces a (d, z)-disjunct matrix with

$$t = 2\left(\frac{(d+1)^{d+1}}{d^d}\right)\left(z - 1 + \ln(\frac{1}{1-p'}) + (d+1)\ln n\right)$$

rows with probability p' for an arbitrarily large constant p'.

Corollary 3.1: The (d, z)-disjunct matrix is asymptotically smaller than the one constructed by Cheng [19].

Corollary 3.2: The time complexity of *ETG* algorithm is asymptotically smaller than that of Cheng's algorithm, given that $d < \sqrt{n}$.

Corollary 3.3: Given that each test has an independent error probability γ, ETG algorithm produces a (d, z)-disjunct matrix with $t = \frac{\tau \ln n(d+1)^2 - 2\tau(d+1)\ln(1-p')}{(\tau - \gamma(d+1))^2}$ with probability p', where $\tau = (d/(d+1))^d$.

4 Trigger Identification Procedure for Basic Attacker Model

In this section, we present the trigger identification procedure for the basic attacker model, where the jammers *deterministically* and *immediately* broadcasts jamming signals on the particular channel which carries the sensed message transmissions between sensor nodes. Therefore, as long as some jamming signals are received, at least one of the broadcasting victim nodes is a trigger. In the next section, we will further investigate the performance of our solution towards some sophisticated attack models, in order to show the robustness of this scheme in real scenarios.

4.1 Identification Overview

The trigger identification can be sketched as follows (Fig. 3):

Assume that at the beginning of the identification phase, all jammer nodes are idle and all the victim nodes in grey and blue have been discovered beforehand. The set of victims are divided into interference-free teams, where the transmissions of victim nodes within one team will not invoke a jammer node, whose interference signals will disrupt the communications within another team, as shown in Fig. 3. We call these teams *testing teams* in the remainder of the paper.

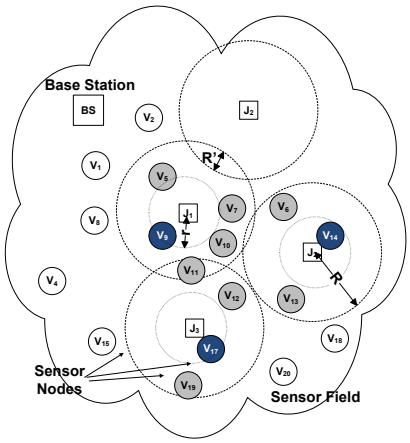


Fig. 3. Nodes in grey and blue are victim nodes around jammer nodes, where blue nodes are also trigger nodes, which invoke the jammer nodes.

The identification of trigger nodes involves two parallel testing types: (1) Denote the set of victim nodes within each *testing team* as W, and the number of trigger nodes (to be estimated) as d, then a group testing procedure will run simultaneously over each *testing team*, to identify the d trigger nodes from $|W|$ victim ones; (2) Victim nodes within each *testing team* will be divided into the multiple *group*s, according to a randomized $(d, 1)$-disjunct matrix, as mentioned in Section 3.2. Each group of victim nodes will be tested on a different channel, to avoid interference among groups.

The testing procedure within each pool is two-fold: (1) Each group i is corresponding to a row in the testing matrix M, and assigned with a different channel frequency f from that of other groups. Let a victim node j broadcast a single bit on f, *iff* $M[i, j] = 1$, to activate possible jammer nodes nearby. Assume M has t rows and each sensor has m radios, then only m groups can be tested at a time, and all t groups can be tested within $\lceil \frac{t}{m} \rceil$ rounds. This is because one victim node

can exist in all the t groups, and it cannot broadcast on $k > m$ different channels in parallel. (2) A minimum dominating set (DS) of the induced subgraph within each *testing team* is also discovered. Upon detecting any jamming signals, a victim node will generate a jamming alarm and send it to the nearest DS node (one-hop transmission due to the definition of DS). Among the multiple DS nodes within a *testing team*, the one which has the shortest distance from the base station will be elected as a *leader*. All the jamming alarms collected by the DS nodes will be transmitted to this leader, using channel surfing schemes. The leader nodes of all the *testing team*s will send all the testing results (jamming alarms) along with the grouping information ($(d, 1)$-disjunct matrix used) to the base station, where the identifications are completed by decoding.

In principle, the flow of the identification algorithm is: (1) Find a maximum number of disjoint interference-free *testing team*s of untested victim nodes. Nodes that are not covered by these teams will be left for the next iteration; (2) Conduct nonadaptive group testing within each *testing team*, by dividing the team members into multiple groups following a $(d, 1)$-disjunct matrix, and testing each group of victim nodes on different channels; (3) Within each *testing team*, let a set of dominating nodes (DS) collect all the testing results and send them to the DS leader, who then transmit the results to the base station; (4) The base station decodes the results and identifies all the trigger nodes within the current set of tested victim nodes; (5) Iterate the above four steps until all the victim nodes are tested.

Take Fig. 3 as an example to show the procedure: the grey nodes and blue nodes (known as grey before identified as a trigger) are all victim nodes, thus divided into three testing teams: $\{v_5, v_7, v_9, v_{11}, v_{16}\}$, $\{v_{11}, v_{12}, v_{17}, v_{19}\}$, $\{v_6, v_{14}, v_{13}\}$. Each of the testing teams are further divided into small groups and tested based on a $(d, 1)$-disjunct matrix. The test outcomes will be sent to the base station for decoding.

Two critical issues of this identification are **how to partition the victim set into maximal interference-free *testing team*s**, and **how to estimate the number of trigger nodes d to determine the testing matrix**, which are illustrated along with theoretical performance analysis in the following sections.

4.2 Discovery of Interference-free *Testing Team*s

As stated above, two disjoint sets of victim nodes are interference-free *testing team*s *iff* the transmission within one set will not invoke a jammer node, whose jamming signals will interfere the communications within the other set. Although the positions of jammer nodes cannot be precisely anticipated, it is possible to discover the set of victim nodes within the same jammed area, i.e. with a distance R from the same jammer node. Any two nodes within the same jammed area should be at most $2R$ far from each other. Consequently, if we induce a new subgraph with all the victim nodes by connecting each node pair with a distance less than $2R$, then the ones jammed by the same jammer node should form a clique. Based on this motivation, we discover all the interference-free *testing team*s in two steps: (1) Find a set of maximum number of vertex-disjoint maximal cliques (clique-independent set); (2) Identify the interferences between these maximal cliques, and decide interference-free *testing team*s. With a subgraph $G' = (W, E')$ where W refers to the set of victim nodes in the network, and $E' = \{(u, v) | \delta(u, v) \leq 2R\}$, it is likely that cliques in G' correspond to the victim nodes jammed by the same jammer. However, maximal cliques which intersect with each other at some victim nodes can cause interferences when testing each of these cliques as a *testing team* in parallel. An example in this case is shown in Fig. 4. To this end, we find all the clique-independent set by adapting Gupta's MCE algorithm [3], as shown in Algorithm 1.

Algorithm 1 Finding Clique-Independent Set (FCIS)

1: **Input**: Induced Subgraph $G' = (W, E')$.
2: **Output**: The set $\mathcal{C}$ of maximum number of disjoint maximal cliques.
3: Find out the set S of all maximal (not disjoint) cliques by using Gupta's *MCE* algorithm [3].
4: **while** $S \neq \emptyset$ **do**
5: Choose clique $C \in S$ which intersects with the minimum number of other cliques in S;
6: $\mathcal{C} \leftarrow \mathcal{C} \cup \{C\}$
7: Remove all the maximal cliques intersecting with C;
8: $S \leftarrow S \setminus \{C\}$
9: **end while**
10: **return** $\mathcal{C}$

Denote the number of cliques returned by Algorithm 1 as $\mathcal{Q}$. It is possible that any two such maximal disjoint cliques can still interfere each other, which is also shown by Fig. 4. Therefore, we study on the minimum distance between any two maximal disjoint clique pair to guarantee interference-free for further group testing.

Definition 4.1: The *shortest clique-path* (*SCP*) between any two maximal disjoint cliques, is defined as the path between the nearest two nodes of these two cliques, which goes through the least number of maximal cliques. As in Fig. 4, the *SCP* between clique C_1 and C_3 is of length $1(C_2)$.

Lemma 4.1: Two *testing team*s are interference-free *iff* the length of *SCP* between them is at least 2.

Proof: Given two disjoint maximal cliques C_i and C_j, denote the two nearest nodes of them as v_i and v_j. Since *SCP* ≥ 2, there is no such a path between C_i and C_j that consists of the edges in only one maximal clique, then the shortest distance between v_i and v_j is larger than $2R$, according to the construction of the subgraph G'. Therefore, C_i and C_j have no nodes that are jammed by or activating the same jammer node. Consequently, no transmissions within C_i can interfere that within C_j and vice versa. □

Lemma 4.2: For any single maximal clique C, it has at most 12 other disjoint maximal cliques, each of which has *SCP* of

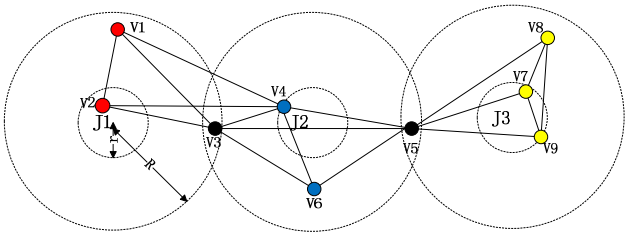


Fig. 4. 3 maximal cliques $C_1 = \{v_1, v_2, v_3, v_4\}$, $C_2 = \{v_3, v_4, v_5, v_6\}$, $C_3 = \{v_5, v_7, v_8, v_9\}$ can be found within 3 jammed areas. C_1 and C_2 are overlapped and if they are tested on the same channel simultaneously, when J_2 is activated by v_4, jamming alarm reported by v_3 might interfere the testing in C_1. Meanwhile, C_1 and C_3 are disjointed, but interference still exists in that, J_2 activated by v_4 in C_1 can disrupt the transmission at v_5, which causes error tests in C_3.

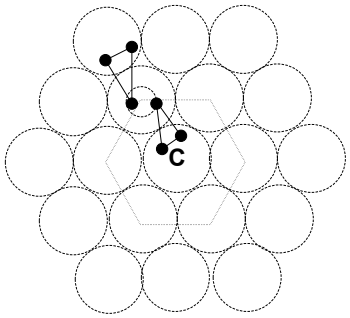


Fig. 5. In the worst case, all the 12 cliques that are 2-hops away from C will have interference with C, similar to the two triangles.

length 1 from C.

Proof: This is straightforward and shown in Fig. 5. □

Based on Lemma 4.1 and Lemma 4.2, we can determine the interference-free *testing team*s by Algorithm 2 as follows: (1) For *testing team*s which have 2-length pairwise *SCP* are interference-free, therefore each team can use up to k channels for testing simultaneously, meanwhile, since each sensor has at most m radios, only m groups can be tested within each *testing team*. (2) For any two *testing team*s which have only 1-length *SCP* from each other, test them using different channels at a time. According to Lemma 4.2, each team can have up to 12 neighbors with 1-length *SCP* from it, therefore up to 13 channels are required to test them simultaneously. If $k \geq 13$, then each team can have $\min\{\lceil \frac{k}{13} \rceil, m\}$ groups tested in parallel.

4.3 Estimation of Trigger Upperbound

Given the number of victim nodes $|W_i|$ in *testing team* i, we first find a deterministic upperbound d_i on the number of trigger nodes within this team in Theorem 4.1, and then decrease d_i by statistically analyzing the number of jammer nodes activated by this team. *Notice that we avoid assuming the deployment of jammer nodes as [6] did and all the results shown are in the worst case.*

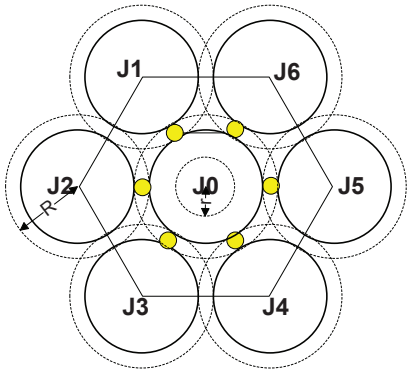


Fig. 6. Maximum number of jammer nodes that can be activated by one *testing team*.

Algorithm 2 Testing on Interference-Free *Testing Team*s

1: **Input**: A set S of disjoint maximal cliques returned by Algorithm 1.
2: **Output**: Interference-free *testing team*s.
3: **if** the number of channels $k \geq 13$ **then**
4: use all the disjoint maximal cliques in S as *testing team*s, however, any two *testing team*s with 1-length *SCP*, are tested on different channels.
5: **else**
6: construct an auxiliary graph $H = (C, E)$ where each maximal clique in S is mapped into a node $v \in C$. Two nodes are connected *iff* their corresponding cliques have 1-length *SCP* from each other.
7: find a maximal independent set (MIS) in H and test the cliques corresponding to this *MIS* using arbitrary channels, since they are interference-free.
8: update H by removing all tested cliques, and iterate step 6–7 until all cliques are tested.
9: **end if**

Lemma 4.3: The triggers in each *testing team* can activate at most 4 jammer nodes.

Proof: As shown in Fig. 6, suppose that there are 6 jammer nodes that can be activated by this clique. Notice that this only happens when all the jammer nodes form a hexagon around the clique. Since the largest distance between any 2 jammer nodes (J_2 and J_5) in this case is at least $2(2R - R')$ (because the distance between two adjacent jammer nodes in the hexagon (J_1 and J_6) is at least $2R - R'$, we have the distance between the two trigger nodes v_1 and v_2, which activate J_2 and J_5 respectively, is at least $2(2R - R') - 2r$. According to our assumption $R - R' > r$, this distance should be larger than $2R$,

TABLE 1
The Probabilities of i jammer nodes activated by one team

i	Probability p_i
2	$[(2c_1^2 \arccos(c_2/4c_1) - 2c_2\sqrt{c_1^2-(c_2/4)^2})/(\pi c_1^2)]^2$
3	$[(2c_1^2 \arccos(c_2/2\sqrt{3}c_1) - 2c_2\sqrt{c_1^2-(c_2/2\sqrt{3})^2})/(\pi c_1^2)]^3$
4	$[(2c_1^2 \arccos(\sqrt{2}c_2/4c_1) - 2c_2\sqrt{c_1^2-(\sqrt{2}c_2/4)^2})/(\pi c_1^2)]^4$

which contradicts with the definition of *testing team*. Therefore, only one jammer of such pair (J_1, J_4), (J_2, J_5), (J_3, J_6) will be activated by this team, so the upperbound of the activated jammer nodes is 4, by considering the center J_0. □

Theorem 4.1: The upper bound d_i on the number of trigger nodes is $\min\{\sum_{s=1}^{4}|c_s(G_i)|, |W_i|\}$, where $c_s(G_i)$ is the s^{th} largest clique over an induced unit disk subgraph $G_i = (W_i, E_i, 2r)$ in the *testing team* i.

Proof: It is straightforward that the maximum distance between two trigger nodes that invoke the same jammer node is $2r$. According to Lemma 4.3, each *testing team* can activate at most 4 jammer nodes. In light of this fact, we can construct another unit disk graph $G_i = (W_i, E_i, 2r)$ on *testing team* i by connecting each victim node pair with distance less than $2r$. In this sense, the trigger nodes which invoke the same jammer nodes are supposed to be in the same clique.

By discovering all the maximal cliques on this new subgraph G_i using Gupta's algorithm [3], we can find the 4 largest maximal cliques and their aggregate size, i.e., $\sum_{s=1}^{4}|c_s(G_i)|$ is thus the maximum number of trigger nodes. Since the upperbound should not exceed the size of the *testing team*, so $d_i = \min\{\sum_{s=1}^{4}|c_s(G_i)|, |W_i|\}$. □

Lemma 4.4: The upperbounds on probabilities of different number of jammer nodes that can be activated by one *testing team* are shown in Table 1, where $c_1 = R - R'$, $c_2 = 2R - R'$, $c_3 = \sin(\pi/10)$.

Proof: Due to the page limit, we just show the proof for the probability of the case that only 2 jammer nodes are activated. Other results can be calculated similarly.

Assume that the 2 jammer nodes are deployed as shown in Fig. 7. Their positions are unnecessarily like this, but we require the two jammed areas to have no overlapping, in order to maximize the probability for the team to activate two jammer nodes. Without loss of generality, we assume that the sensor nodes are deployed randomly over the disk F with radius $R - R'$, therefore by denoting the two areas in shadow as S_1 and S_2 respectively, we can have the probability of this case is upperbounded by $S_1 \cdot S_2/S^2$, where S refers to the area of the disk with radius $R - R'$ (same as the jammed area). To maximize this probability, we assume $S_1 = S_2$ in the following calculations.

Since $|AF| = R - R'$ and $|EF| = R'/2$, we have $|DF| = |AF| - (|AF| - |EF|)/2 = c_2/4$. As a result, $S_{FBAC} = [2\arccos(|DF|/|BF|)/2\pi]\pi|BF|^2 = c_1^2\arccos(c_2/4c_1)$. For the triangle FBC, $|BD| = \sqrt{|BF|^2 - |DF|^2} = \sqrt{c_1^2 - (c_2/4)^2}$, we have $S_{\Delta FBC} = 1/2BC \cdot DF = BD \cdot DF = c_2\sqrt{c_1^2-(c_2/4)^2}$. Hence $S_{BAEC} = 2(S_{FBAC} - S_{\Delta FBC})$ and the probability is equal to $(S_{BAEC}/S)^2 = [(2c_1^2\arccos(c_2/4c_1) - 2c_2\sqrt{c_1^2-(c_2/4)^2})/(\pi c_1^2)]^2$, which completes the proof. □

Therefore, the conclusion from Theorem 4.1 can be improved by replacing the deterministic maximum number of jammer nodes with its expected value derived from the above results. Since the calculation of this part is quite complicated, but not the core of our solution, it is not included in this paper.

4.4 Analysis of Time and Message Complexity

Time complexity: The time overhead of this trigger identification procedure is three-fold: (1) the discovery of maximum number of disjoint maximal cliques; (2) the iterative tests on multiple *testing team*s. As mentioned above, the algorithm in [3] finds $O(l\Delta)$ maximal cliques on UDG, within $O(l\Delta^2)$ time, where $l = |E|$ and Δ refers to the maximum degree. We used a greedy algorithm to find a *MCIS* from these $O(l\Delta)$ cliques with $O(l^3\Delta^3\mathcal{Q})$ time: $O(l\Delta)$-time for each clique to check the overlapping with other cliques, $O(l\Delta)$-time to find a clique overlapping with minimum other cliques, and $\mathcal{Q}$ denotes the number of *testing team*s. Notice that in practice, sensor networks are not quite dense, so the number of edges l and maximum degree Δ are actually limited to small values. Since the efficiency of this phase depends on the *MCIS* algorithms, which hopefully can be further improved, this section therefore only focuses on (2), which is also the kernel of our scheme.

Since the group testing procedures are conducted within each *testing team*, which is a clique that can be covered by a disk with radius $2R$, therefore the transmission latency between nodes within each *testing team* is quite low and thus negligible. Moreover, no new testing rounds can start until all the activated jammer nodes *hibernate* again. Therefore, the length of each testing round could be set to a predefined constant, which does not rely on the size of each testing group. *To this end, we count the time complexity of this phase in terms of the total number of* testing round*s needed*. Specifically, each *testing round* is counted since the victim nodes broadcasting testing signals to activate jammer nodes nearby, till the DS leader node finishing collecting the testing results. Under this basic jamming environment, where jammer deterministically reply to any sensed legitimate transmissions with interference signals, we conduct all the tests within the same *testing team* in a synchronized manner, i.e. set the length of each testing round as a predefined value, therefore by denoting the number of total testing rounds as $\mathcal{L}$, the length of identification period is $O(\mathcal{L})$. However, under some sophisticated jamming environments to be discussed later, we will relax the synchronized constraint, advance the tests to asynchronous ones, and make benefit from this change to alleviate test errors.

Lemma 4.5: Based on the ETG algorithm, the minimum number of tests to identify d trigger nodes from $|W|$ vic-

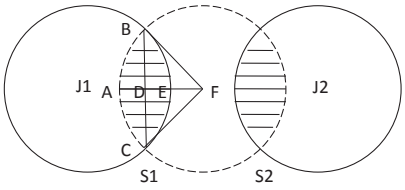


Fig. 7. The maximum probability of 2 jammer nodes being activated by one *testing team*

tim nodes can be loosely upperbounded (for simplicity) by $t(|W|, d) = O(d_i^2 \lceil \ln |W| \rceil)$ w.h.p.

Theorem 4.2: **(Main)** The total number of testing rounds is upper bounded by

$$O(\max_{i=1}^{\mathcal{Q}}\{\frac{13\min\{d_i^2\lceil \ln |W_i|\rceil, |W_i|\}}{m}\})$$

w.h.p, with $d_i = \min\{\sum_{s=1}^{4}|c_s(G_i)|, |W_i|\}$ and $c_s(G_i)$ is the s^{th} largest clique over an induced unit disk subgraph $G_i = (W_i, E_i, 2r)$ in the *testing team* i.

Proof: First, from Lemma 4.5, at most $\frac{t(|W|,d)}{m} = \frac{d_i^2\lceil \ln |W|\rceil}{m}$ testing rounds are needed to identify all nodes in *testing team* i. Second, the set of *testing team*s that can be tested in parallel is corresponding to a *MIS* on H constructed in Algorithm 2, according to Lemma 4.2, nodes in H has a maximum degree 12. It is straightforward to see the maximum degree of the resulting graph H' by iteratively removing a *MIS* from it decreases by at least 1 in every iteration, therefore all the nodes can be covered by up to 13 *MIS*. Therefore, from Theorem 4.1, the proof is completed. □

Notice that the computation overhead of the randomized $(d, 1)$-disjunct matrix is quite small, and thus not included in the analysis. *Compared with the number of testing rounds*

$$O(\sum_{i=1}^{\Delta(H)}\max_j\lceil\min\{(2+o(1))\frac{d_j^2\log_2^2|W_j|}{\log_2^2(d_j\log_2|W_j|)}, |W_j|\}/m\rceil)$$

in [6] where $\Delta(H)$ refers to the maximum degree of the an induced graph H, our result is asymptotically better since no maximum degree is involved.

Message Complexity: Based on our assumptions above, the jammer nodes will be activated upon receiving a single message from the trigger nodes. Considering that there are approximately $\frac{|W_i|}{d+1}$ victim nodes in each testing group of team W_i (mentioned in the construction of randomized (d, z)-disjunct matrix in Appendix), the communication overhead of each testing group in a testing round is three-fold: (1) $\frac{|W_i|}{d+1}$ *testing message* broadcasted by all victim nodes in each group of team W_i; (2) $\frac{|W_i|}{d+1}$ *jamming alarm message* sent to some *DS* nodes by victim nodes that senses the jamming signals; (3) 1 *result report message*s from each *DS* node to the *DS* leader node and 1 *testing result message* sent to the base station by each *DS* header node. Since the size of *DS* for each group W_i is at most $|W_i|$, the overall communication complexity is

$$O(\sum_i^{\mathcal{Q}}|W_i|\max_{i=1}^{\mathcal{Q}}\{d_i^2\lceil\ln|W_i|\rceil, |W_i|\}m)$$

according to Theorem 4.2. Notice that this is slightly larger than that of [6], however, we did not take into account the amount of *result report message*, which contributes the most in the message complexity above.

TABLE 2
Notations

Notation	Content
T^+	The number of false positive outcomes
T^-	The number of false negative outcomes
$u(i)$	The number of trigger nodes in test i
$x(i)$	The reaction time of jammer toward test i
$g(i)$	The outcome of test i

5 Advanced Solutions Toward Sophisticated Attack Models

In this section, we consider two sophisticated attacker models: *probabilistic attack* and *variant response time delay*, where the jammers rely each sensed transmission with different probabilities, instead of deterministically, or delay the jamming signals with a random time interval, instead of immediately. Since our scheme is robust and accurate in the steps of grouping, generating disjunct matrix and decoding the testing results, the only possible test errors arise from the generation of testing outcomes. Nevertheless, by using the error-tolerant disjunct matrix and relaxing the identification procedures to asynchronous manner, our scheme will provide small false rates in these cases. Some notations can be found in Table 2. In this section, the terms *test* and *group*, the terms *column* and *nodes* are interchangeable.

5.1 Upperbound on the Expected Value of z

First, we investigate the properties of both jamming behaviors and obtain the expected number of error tests in both cases through the following analysis. Since in practice, it is not trivial to establish accurate jamming models, we derive an upperbound of the error probability which does not require the beforehand knowledge of the objective jamming models, which is therefore feasible for real-time identifications. Since it is a relaxed bound, it could be further strengthened via learning the jamming history.

5.1.1 Probabilistic Jamming Response

A clever jammer can choose not to respond to some sensed ongoing transmissions, in order to evade the detection. Assume that each ongoing transmission has an independent probability

η to be responded. In our construction algorithm ETG, where each matrix entry is IID and has a probability p to be 1, therefore for any single test i with $i \in [1, t]$:

$$\mathbf{Pr}[u(i) = x] = \binom{d}{x} p^x (1-p)^{d-x} \quad (1)$$

Hence for each test i, the event that it contains no trigger nodes but returns a positive result, has a probability at most:

$$\begin{aligned} & \mathbf{Pr}[g(i) = 0 \ \& \ u(i) \geq 1] \\ = & \sum_{x=1}^{d} (1-\eta)^x \binom{d}{x} p^x (1-p)^{d-x} \\ = & [(1-\eta)p + 1 - p]^d - (1-p)^d \\ = & (1-\eta p)^d - (1-p)^d < (1-\eta)p \end{aligned}$$

Meanwhile, the event that it contains at least one trigger but returns a negative result, has a probability:

$$\mathbf{Pr}[g(i) = 1 \ \& \ u(i) = 0] = 0 \quad (2)$$

Since in practical $\eta \geq \frac{1}{2}$, we therefore have the expected number of false positive and negative tests is respectively at most $pt/2$ and 0.

5.1.2 Variant Reaction Time

The introduction of group testing techniques aims to decrease the identification latency to the minimum, therefore, if the jammer would not respond intermediately after sensing the ongoing transmissions, but instead wait for a randomized time delay, the test outcomes would be messed up. Since it is expensive to synchronize the tests among sensors, we use a predefined testing length as $\mathcal{L}$, thus the test outcome of test $i \in [1, t]$ is generated within time interval $[(\lceil \frac{i}{m} \rceil - 1)\mathcal{L}, \lceil \frac{i}{m} \rceil \mathcal{L}]$. There are two possible error events regarding any test i.

- $Fp(i)$: test i is negative, but some jamming signals are delayed from previous tests and interfere this test, where we have a false positive event;
- $Fn(i)$: test i is positive, but the jammer activated in this test delayed its jamming signals to some subsequent tests, meanwhile, no delayed jamming signals from previous tests exists, where we have a false negative event.

Since the jammers in this paper are assumed to block communications only on the channels where transmissions are sensed, for the following analysis, we claim that the interferences can only happen between any two tests i, j with $i \equiv j (\mathbf{mod}\ m)$. Denote the delay of jamming signals as a random variable $X = \{x(1), x(2), x(3), \cdots x(t)\}$ where $x(i)$ is the delay for possible jamming signals arisen from test i.

(1) For event $Fp(i)$, consider the test $i - m$, in order to have its jamming signals delayed to test i, we have a bound on $x(i-m) \in (0, 2\mathcal{L})$. Similarly, in order to have the signals of any test j delayed to i, we have $x(j) \in [(\frac{i-j}{m} - 1)\mathcal{L}, (\frac{i-j}{m} + 1)\mathcal{L}]$. Further assume the probability density function of X is $\mathcal{P}(i) = \mathbf{Pr}[X = x(i)]$. Consider all the tests prior to i, which are $i\%m, 1 + i\%m, \cdots, i - m$, we then have the probability for $Fp(i)$:

$$(1-p)^d \sum_{j=i\%m}^{i-m} \int_{(\frac{i-j}{m}-1)\mathcal{L}}^{(\frac{i-j}{m}+1)\mathcal{L}} \mathcal{P}(w)\mathbf{d}w(1-(1-p)^d) \quad (3)$$

To simplify this expression, we assume that $X/\mathcal{L}$ follows a uniform distribution within the range $[0, \beta]$ with a small β, which is reasonable and efficient for attackers in practice. Since the nature of jamming attacks lies in adapting the attack frequency due to the sensed transmissions, too large delay does not make sense to tackle the ongoing transmissions. Under a uniform distribution, the probability of $Fp(i)$ becomes:

$$\begin{aligned} & (1-(1-p)^d)(1-p)^d \sum_{j=\max i\%m, i-m-\beta-1}^{i-m} \frac{2}{\beta} \\ = & (1-(1-p)^d)(1-p)^d (\lceil \frac{i}{m} \rceil - 1)\frac{2}{\beta} \end{aligned}$$

Therefore, the expected number of false positive tests is at most

$$\begin{aligned} T^+ & \leq \sum_{i=1}^{t} (1-(1-p)^d)(1-p)^d(\beta)\frac{2}{\beta} \\ & \leq 2\sum_{i=1}^{t} (1-(1-p)^d)(1-p)^d \\ & \leq 2(1-(1-p)^d)(1-p)^d t \end{aligned}$$

(2) For event $Fn(i)$, following the similar arguments above, we have an upperbound of the probability for $Fn(i)$ (assume that any delays larger than l at test i will interfere the tests j following i where $j \in [\max(i\%m, i-m-\beta-1), i-m]$):

$$\begin{aligned} & (1-(1-p)^d) \int_{l}^{+\infty} \mathcal{P}(w)\mathbf{d}w \\ & \cdot \left(1 - \sum_{j} \int_{(\frac{i-j}{m}-1)\mathcal{L}}^{(\frac{i-j}{m}+1)\mathcal{L}} \mathcal{P}(w)\mathbf{d}w(1-(1-p)^d)\right) \\ \leq & (1-(1-p)^d)(1-2(1-(1-p)^d))(\beta - l)/\beta \\ \leq & (1-(1-p)^d)(1-2(1-(1-p)^d)) \end{aligned}$$

So the expected number of false negative tests is at most

$$T^- \leq (1-(1-p)^d)(1-2(1-(1-p)^d))t \quad (4)$$

Therefore, we could use a union bound and obtain a worst-case error rate of each test:

$$\begin{aligned} \gamma & = \frac{p}{2} + 2(1-(1-p)^d)(1-p)^d \\ & \quad + (1-(1-p)^d)(1-2(1-(1-p)^d)) \\ & = (10\tau - 8\tau^2 - \tau^{-d} - 1)/2 \end{aligned}$$

where $\tau = (d/(d+1))^d$. Intuitively, we can have an upper-bound on the number of error tests as $z = \gamma t = (10\tau - 8\tau^2 -$

$\tau^{-d} - 1)/2$, and take it as an input to construct the (d, z)-disjunct matrix. However, notice that z depends on t, i.e., the number of rows of the constructed matrix, we therefore derive another bound of t related to γ, as shown by Corollary 3.3 in the appendix.

5.2 Error-tolerant Asynchronous Testing within each testing team

By applying the derived worst-cast number of error tests into the ETG construction, we can obtain the following algorithm where tests are conducted in asynchronous manner to enhance the efficiency.

Algorithm 3 Asynchronous Testing

1: **Input**: n victim nodes in a *testing team*.
2: **Output**: all trigger nodes within these victim nodes.
3: Estimate d as mentioned.
4: Set $\gamma = (10\tau - 8\tau^2 - \tau^{-d} - 1)/2$. //upper bound of error probability for each test.
5: Set $t = \frac{\tau \ln n(d+1)^2}{(\tau - \gamma(d+1))^2}$. //number of rows.
6: Construct a (d, z)-disjunct matrix using ETG algorithm with t rows, and divide all the n victim nodes into t groups accordingly $\{g_1, g_2, \cdots, g_t\}$.
7:
8: /* For each round, conduct group testing on m groups using m different channels (radios). The testing is asynchronous in that, the m groups tested in parallel do not wait for each other to finish the testing, instead, any finished test j will trigger the test $j + m$, i.e., the tests are conducted in m pipelines. */
9: **for** $i = 1$ to $\lceil t/m \rceil$ **do**
10: Conduct group testing in groups g_{im+1}, g_{im+2}, g_{im+m} in parallel;
11: If any nodes in group g_j with $j \in [im + 1, im + m]$ detects jamming noises, the testing in this group finishes and start testing on g_{j+m}.
12: If no nodes in group g_j detect jamming noises, while at least one other test in parallel detects jamming noises, let all the nodes in group g_j resend 3 more messages to activate possible hidden jammers. If no jamming signals are detected till the end of the predefined round length ($\mathcal{L}$), return a negative outcome for this group and start testing on g_{j+m}.
13: **end for**

As shown in Algorithm 3, after all the groups are decided, conduct group testing on them in m pipelines, where in each pipeline any detected jamming signals will end the current test and trigger the next tests while groups receiving no jamming signals will be required to resend triggering messages and wait till the predefined round time has passed. These changes over the original algorithm, especially the asynchronous testing are located in each testing team, thus will not introduce significant overheads, however, the resulted error rates are limited to a quite low level. The corresponding simulation results will be shown in the next section.

6 Simulation Results

For the sake of validating the theoretical results obtained and showing the applicability of this approach to real-time identification, we simulated the proposed trigger identification procedure on a 1000×1000 square sensor field with uniformly distributed n sensor nodes, one base station and J randomly distributed jammer nodes. We did not investigate more sophisticated node deployments, since our solution is orthogonal and robust with them, and the simulation results suffice reflecting the efficiency of this scheme.

In detail, we set the transmission radius $r = 50$ for the sensor nodes, and $R = 2r$ for the jammer nodes. $m = 3$ radios with $k > m$ channels are implemented with no packet-loss or external noise (except jamming signals), to guarantee the accuracy of jamming detection and test results generation. The reason why we limit R to only $2r$ is, jammer nodes with extremely large transmission range can be favored by attackers, but this has huge energy cost and risk to be disclosed. Notice from the above analysis, the performance of our solution does not rely on this assumption.

The performance assessments are two-fold: (1) since the only existing trigger identification work is our previous result in [6], we compare our new approach (referred as *Clique-Based* below) with that one (referred as *Disk-Based* below) through two benchmarks: average number of the *testing rounds* and the *communication messages* per victim node, with different environment settings. (2) besides the time and message complexity, we investigate the precision of this new solution in terms of false positive/negative rate, in the presence of a different jamming behaviors. Although changes of network size, number of jammers, transmission radius will probably affect the false rate of our identification, since in principle, we divide the victim nodes into small testing teams and further tested in even smaller groups, the influences of the jammer behaviors are intuitively much more significant than those parameters.

6.1 Time and Message Complexity

We range n ranges from 450 to 550 with step 2, r from 50 to 60 with step 0.2 and J from 3 to 10 with step 1 to show the robustness of our solution in time and message complexity. Parameter values lower than these intervals would make the sensor network less connected and jamming attack less severe, while higher values would lead to impractical dense scenarios and unnecessary energy waste.

As shown in Fig. 8(a) and 8(b), this clique-based scheme completes the identification with steadily less than 10 rounds, compared to the increasing time overhead with more than 15 rounds of the disk-based solution, as the network grows denser with more sensor nodes. Meanwhile, its amortized

communication overheads are only slightly higher than that of the other solution, whereas both are below 10 messages per victim node. Therefore, the new scheme is even more efficient and robust to large-scale network scenarios.

With the sensor transmission radius growing up, the time complexity of the disk-based solution gradually ascends (Fig. 8(d) and 8(c)) due to the increased maximum degree $\Delta(H)$ mentioned in the analysis above. Comparatively, the time cost of clique-based solution remains below 10 rounds, while the message complexity still approximates the other one.

Since sensor nodes are uniformly distributed, the more jammer nodes placed in the networks, the more victim nodes are expected to be tested, the identification complexity will therewith raises, as the performance of disk-based scheme shows in Fig. 8(f) and 8(e). Encouragingly, the proposed scheme can still finish the identification promptly with less than 10 rounds, which grows up much slower than the other. It has slightly more communication overheads (10 messages per victim nodes) but is still affordable to power-limited sensor nodes.

6.2 False Positive/ Negative Rate

In order to show the precision of our proposed solution under different jamming environments, we vary the two parameters of the jammer behaviors above: *Jammer Response Probability* α and *Testing Round Length/Maximum Jamming Delay* $\mathcal{L}/X$ and illustrate the resulted false rates in Fig. 8(g) and 8(h). To simulate the most dangerous case, we assume a hybrid behavior for all the jammers, for example, the jammers in the simulation of Fig. 8(g) not only launch the jamming signals probabilistically, but also delay the jamming messages with a random period of time up to $2\mathcal{L}$. On the other hand, the jammers in the simulation of Fig. 8(h) respond each sensed transmission with probability 0.5 as well. All the simulation results are derived by averaging 10 instances for each parameter team.

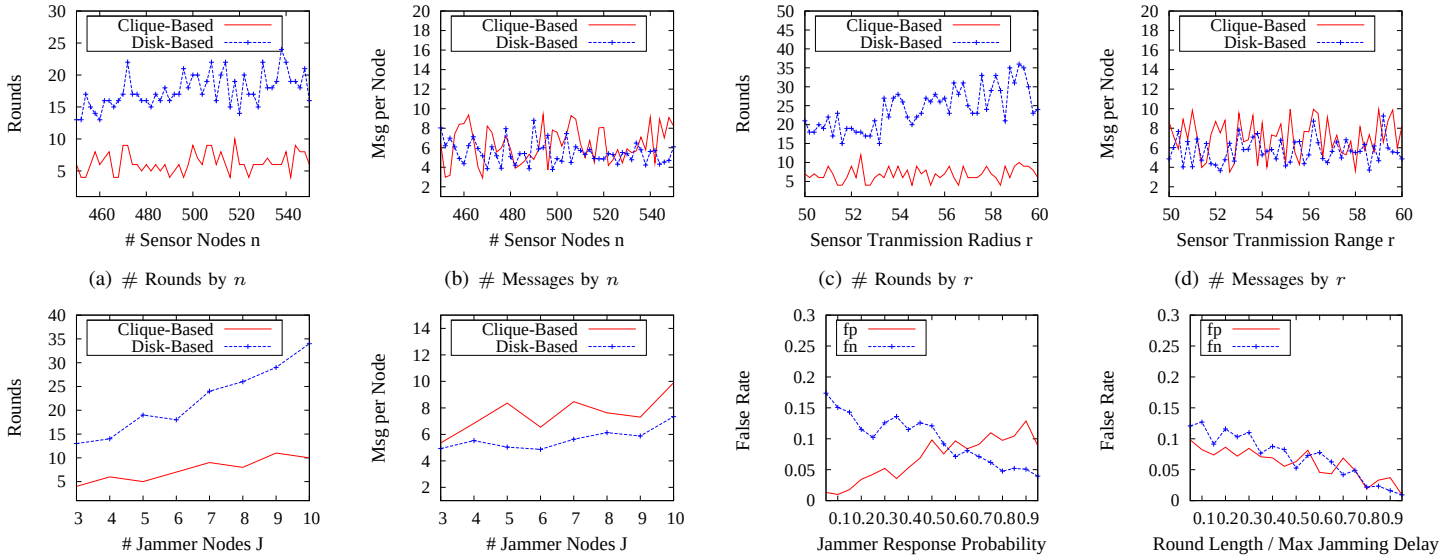
As shown in both figures, we consider the *extreme* cases where jammers respond transmission signals with a probability as small as 0.1, or delay the signals to up to 10 testing rounds later. This actually contradicts with the nature of reactive jamming attacks, which aim at disrupting the network communication as soon as any legitimate transmission starts. The motivation of such parameter setting is to show the robustness of this scheme even if the attackers sense the detection and intentionally slow down the attacks. The overall false rates are below 20% for any parameter values.

In Fig. 8(g), when $\alpha > 1/2$ which corresponds to practical cases, we find that the false negative rates generally decrease from 10% to 5% as α increases. Meanwhile the false positive rate grows gently, but is still below 14%, this is because as more and more jamming signals are sent, due to their randomized time delays, more and more following tests will be influenced and become false positive. In Fig. 8(h), considering the practical cases where $\mathcal{L}/X > 1/2$, both rates are going down from around 10% to 1%, since the maximum jamming delay becomes shorter and shorter compared to the testing round length $\mathcal{L}$, in which case, the number of interferences between consecutive tests is decreasing.

Overall, the new solution not only improves the efficiency of the trigger-identification procedure with much smaller time complexity and acceptable message overhead, but also limits the identification false rate to the desirable low levels in the presence of various jamming behaviors. Considering the prompt testing procedure could be iteratively conducted, the false rate would be further decreased to enhance the identification accuracy. With these performance guarantee, this procedure is promising to be developed into a jamming-resilient routing scheme.

7 Related Works

Existing countermeasures against jamming attacks in WSN can be categorized into two facets: signal detection and mitigation, both of which have been well studied and developed with various defense schemes. On the one hand, a majority of detection methods focus on analyzing specific object values to discover abnormal events, e.g., Xu et. al [15] studied a multi-model (*PDR*, *RSS*) to consistently monitor jamming signals. Work based on similar ideas [16][14][13] improved the detection accuracy by investigating sophisticated decision criteria and thresholds. However, reactive jamming attacks, where the jammer node are not continuously active and thus unnecessary to cause huge deviations of these variables from normal legitimate profiles, cannot be efficiently tackled by these methods. In addition, some recent works proposed methods for detecting jammed areas [10] and directing normal communications bypass possible jammed area using wormhole [17]. These solutions can effectively mitigate jamming attacks, but their performances rely on the accuracy of detection on jammed areas, i.e. the transmission overhead would be unnecessarily brought up if the jammed area is much larger than its actual size. On the other hand, mitigation schemes which benefit from channel surfing [12], frequency hopping and spatial retreats[11], reactively help legitimate nodes escape from the jammed area or frequency. Unfortunately, being lack of pre-knowledge over possible positions of hidden reactive jammer nodes, legitimate nodes cannot efficiently evade jamming signals, especially in dense sensor network when multiple mobile nodes can easily activate reactive jammer nodes and cause the interference. For the sake of overcoming these limitations above, in [6] we studied on the problem of identification trigger nodes with a short period of time, whose results can be employed by jamming-resistent routing schemes, to avoid the transmissions of these trigger nodes and deactivate the reactive jammer nodes. In this paper, we improve this scheme by introducing a novel randomized error-tolerant group testing technique, which enhances the identification speed, and handles error tests under unreliable network environments as well.



(a) # Rounds by n (b) # Messages by n (c) # Rounds by r (d) # Messages by r

(e) # Rounds by J (f) # Messages by J (g) False Rate by different Jammer Response Probability (h) False Rate by different Ratios of Testing Round Length/Maximum Jamming Delay

Fig. 8. Performance Under Different Circumstances

8 Conclusions

We proposed the idea of identifying trigger nodes using group testing schemes in previous work. In this paper, we improve this mitigation method by optimizing the grouping scheme and testing matrix by employing an advanced clique-independent set algorithm and a randomized (d, z)-disjunct matrix construction. Through theoretical proof and analysis, the identification procedure has a lower time complexity than the previous solution. We also provide preliminary simulation results to elaborate the applicability and robustness of this scheme to wireless sensor network with various settings.

Moreover, we analyze two representative jamming behavior models and propose an error-tolerant asynchronous identification algorithm which allows very few test errors under unreliable jamming environments. Since both our new construction of (d, z)-disjunct matrix and the discovery of clique-independent set have the potential to be implemented distributively (Gupta *et al.* already proposed a distributed algorithm for finding maximal cliques with unit radius on UDG in[3]), the distributed version of this trigger identification is quite promising and will become a part of our future work.

Appendix A
Construction of Randomized Error-Tolerant d-disjunct Matrix

We include the proofs of several theorems mentioned in Section 3 regarding the performance of this algorithm.

Proof of Theorem 3.2.

Proof:

Algorithm 4 ETG construction

1: **Input**: n, d, z, p';
2: **Output**: (d, z)-disjunct matrix with probability p'
3: Set $p = \frac{1}{d+1}$,
4: Set $t = 2\left(\frac{(d+1)^{d+1}}{d^d}\right)\left(z - 1 + \ln\frac{1}{1-p'} + (d+1)\ln n\right)$
5: Construct a $t \times n$ matrix M by letting each entry to be 1 with probability p.
6: **return** M

M is not (d, z)-disjunct matrix if for any single column c_0 and any other d columns $c_1, \cdots c_d$, there are at most $z-1$ rows where c_0 has 1 and all $c_1, \cdots c_d$ have 0. By denoting $p = (\frac{1}{2})^l$, considering a **particular** column and d other columns in the matrix, the probability of such failure pattern is:

$$\sum_{i=0}^{z-1}\binom{t}{i}[p(1-p)^d]^i[1-p(1-p)^d]^{t-i}$$

So use the union bound for all possible combinations and permutations of $(d+1)$ columns, we have the failure possibility bounded by

$$P_1 \le (d+1)\binom{n}{d+1}\sum_{i=0}^{z-1}\binom{t}{i}[p(1-p)^d]^i[1-p(1-p)^d]^{t-i}$$

Here consider the CDF of binomial series and assume that $z - 1 \le tp(1-p)^d$(**assert 1**), we then have

$$P_1 \le n^{d+1}exp(-\frac{(tp(1-p)^d - z + 1)^2}{2tp(1-p)^d})$$

by *Chernoff bound*. To bound this by $1-p'$, i.e.,

$$P_1 \leq n^{d+1} exp(-\frac{(tp(1-p)^d - z + 1)^2}{2tp(1-p)^d}) \leq 1 - p'$$

we can derive that (**assert 2**)

$$\begin{aligned} p(1-p)^d &\leq \frac{z - 1 + \ln \frac{1}{1-p'} + (d+1)\ln n}{t} \\ &\quad - \frac{\sqrt{\ln^2(\frac{1}{1-p'} n^{d+1}) + 2(z-1)\ln \frac{1}{1-p'} n^{d+1}}}{t} \end{aligned}$$

(infeasible by assert 1)
or

$$\begin{aligned} p(1-p)^d &\geq \frac{z - 1 + \ln \frac{1}{1-p'} + (d+1)\ln n}{t} \\ &\quad + \frac{\sqrt{\ln^2(\frac{1}{1-p'} n^{d+1}) + 2(z-1)\ln \frac{1}{1-p'} n^{d+1}}}{t} \end{aligned}$$

Therefore, we can derive the lower bound

$$t \geq 2\left(\frac{(d+1)^{d+1}}{d^d}\right)\left(z - 1 + \ln(\frac{1}{1-p'}) + (d+1)\ln n\right)$$

□

Proof of Corollary 3.1.

Proof:

From Theorem 3.2, we have

$$\begin{aligned} t &\leq 2(\frac{d+1}{d})^d (d+1)[z - 1 + \ln(\frac{1}{1-p'}) + (d+1)\ln n] \\ &\leq 2e(d+1)[z - 1 + \ln(\frac{1}{1-p'}) + (d+1)\ln n] \end{aligned}$$

Since $\lim_{d\to\infty}(1+\frac{1}{d})^d = e$ and $(1+\frac{1}{d})^d$ monotonically increases, we have

$$\begin{aligned} t &\leq 3.78(d+1)^2 \log n + 3.78(d+1)\log(\frac{2}{1-p'}) \\ &\quad -3.78(d+1) + 5.44(d+1)(z-1) \end{aligned}$$

Compared with the number of rows of the matrix constructed by Cheng [19], which is denoted as t_1 as mentioned:

$$\begin{aligned} t_1 &= 4.28d^2 \log \frac{2}{1-p'} + 4.28d^2 \log n + 9.84dz \\ &\quad +3.92z^2 \ln \frac{2n-1}{1-p'} \end{aligned}$$

with $d \geq 2$, $z \geq d+1$, $p' < \frac{1}{2}$ and $n > d$, it is evident that $t < t_1$.

□

Proof of Corollary 3.2.

Proof:

Since ETG algorithm only contains one probability calculation for each entry, the overall time complexity is loosely upper-bounded by $O(d^2 n \log n)$.
Given that $d < \sqrt{n}$ which is practical, this is smaller than that of Chang's algorithm, $O(n^2 \log n)$.

□

Proof of Corollary 3.3.

Proof:

Substituting z by γt in the proof of Theorem 3.2 completes this proof. □

REFERENCES

[1] D. Z. Du and F. Hwang, *Pooling Designs: Group Testing in Molecular Biology*, World Scientific, Singapore, 2006.

[2] M. Goodrich, M. Atallah, and R. Tamassia. "Indexing information for data forensics." *3rd ACNS, Lecture Notes in Computer Science 3531, Springer*, 2005.

[3] R. Gupta, J. Walrand, and O. Goldschmidt, "Maximal cliques in unit disk graphs: Polynomial approximation." *INOC '05*, Portugal, March 2005.

[4] V. Guruswami and C. P. Rangan, "Algorithmic aspects of clique-transversal and clique-independent sets." *Discrete Applied Mathematics*, 100:183–202, 2000.

[5] W. Hang, W. Zanji, and G. Jingbo, "Performance of dsss against repeater jamming." *Electronics, Circuits and Systems, ICECS '06*, Dec. 2006.

[6] I. Shin, Y. Shen, Y. Xuan, M. T. Thai, and T. Znati, "Reactive jamming attacks in multi-radio wireless sensor networks: an efficient mitigating measure by identifying trigger nodes." *FOWANC, in conjunction with MobiHoc*, 2009.

[7] O. Sidek and A. Yahya, "Reed solomon coding for frequency hopping spread spectrum in jamming environment." *American Journal of Applied Sciences*, 5(10):1281–1284.

[8] M. Strasser, B. Danev, and S. Capkun. "Detection of reactive jamming in sensor networks." *ETH Zurich D-INFK Technical Report*, August 2009.

[9] H. Wang, J. Guo, and Z. Wang. "Feasibility assessment of repeater jamming technique for dsss." *WCNC2007. IEEE*, pages 2322–2327, March 2007.

[10] A. D. Wood, J. Stankovic, and S. Son. "A jammed-area mapping service for sensor networks." *RTSS '03*, pages 286–297, 2003.

[11] W. Xu, K. Ma, W. Trappe, and Y. Zhang. "Jamming sensor networks: Attack and defense strategies." *IEEE Network*, 20:41–47, 2006.

[12] W. Xu, T. Wood, W. Trappe, and Y. Zhang. "Channel surfing and spatial retreats: Defenses against wireless denial of service." *2004 ACM workshop on Wireless security*, pages 80–89, 2004.

[13] Mingyan Li, I. Koutsopoulos, and R. Poovendran. "Optimal Jamming Attacks and Network Defense Policies in Wireless Sensor Networks". *INFOCOM '07*, May 2007.

[14] R. A. Poisel. "Modern Communications Jamming Principles and Techniques". *Artech House*, 2004.

[15] W. Xu, W. Trappe, Y. Zhang, and T. Wood. "The feasibility of launching and detecting jamming attacks in wireless networks". *MobiHoc '05*, pages 46–57, New York, NY, USA, 2005.

[16] M. Cakiroglu and A. T. Ozcerit. "Jamming Detection Mechanisms for Wireless Sensor Networks." *3rd InfoScale*, Brussels, Belgium, 2008.

[17] M. Cagalj, S. Capkun, and J. P. Hubaux. "Wormhole- Based Antijamming Techniques in Sensor Networks."*IEEE Transactions on Mobile Computing*, 2007.

[18] I. Shin, R. Tiwar, T. N. Dinh, M. T. Thai and T. Znati, "A localized algorithm to locate reactive jammers with trigger nodes in wireless sensor networks". *Manuscript, 2009*.

[19] Y.-X. Chen and D.-Z. Du, "New Constructions of One- and Two- Stage Pooling Designs", *Journal of Computational Biology*, 2008

[20] Garey, M.G., Johnson, D.S, "The Rectilinear Steiner Tree Problem is NP-Complete", *SIAM J. Appl. Math.* 32, 826C834 (1977)

[21] L. G. Valiant, "Universality considerations in VLSI circuits", *IEEE Transactions on Computers 30 (1981)*, 135C140.