Trust-enabled Link Spoofing Detection in MANET

Mouhannad Alattar, Françoise Sailhan, Julien Bourgeois

Trust-enabled Link Spoofing Detection in MANET

Mouhannad ALATTAR
Femto-st Laboratory,
University of Franche-Comté,
Montbéliard, France.
firstName.lastName@univ-fcomte.fr

Françoise SAILHAN
Cédric Laboratory,
CNAM,
Paris, France.
firstName.lastName@cnam.fr

Julien BOURGEOIS
Femto-st Laboratory,
University of Franche-Comté,
Montbéliard, France.
firstName.lastName@univ-fcomte.fr

***Abstract*—Ad hoc networks operate over open environments and are hence vulnerable to a large body of threats. To tackle this issue, we propose a distributed, signature-based anomaly detector that evaluates the trustworthiness of others so as to secure such a distributed detection. Contrary to existing detectors that passively observe packets, our detector analyses logs so as to identify patterns of misuse and proactively collaborate with others to gather additional evidences. As a result, no change is requested in the implementation of the node. The main challenge stems from difficulty involved in stating the occurrence of an attack based on second-hands evidences that may come from colluding attacker(s). To tackle this issue, we propose an entropy-based trust system that evaluates the trustworthiness of the nodes that provide the evidences. We further introduce a novel indicator which measures the level of confidence in the detection. Preliminary evaluations of the trust system along with the confidence measure have been conducted.**

***Index Terms*—Intrusion detection; Trust; MANETs; routing protocols.**

I. Introduction

Detecting attacks in *ad hoc* networks is challenging because these networks are cooperative and hence lack of centralized security enforcement points from which preventive strategies are launched. Recent works showed that attacks may be identified as a deviation of the correct behavior (anomaly detection); this correct behavior is either hand-specified relying on a protocol description, e.g., [1] or automatically built/analyzed using machine learning or data mining techniques, e.g., [2]. The difficulty inherent to the automatic modeling of the behavior of dynamic routing protocols leads to many false positives that are reduced by coupling automatic and specification-based anomaly detection. An alternative, which reduces the number of false positives, consists in describing the way the intruder penetrates the system (by establishing an intrusion signature) and detecting any behavior that is close to that signature. Comparatively, little attention - to the best of our knowledge, only couples of works [3], [4] - focuses on signature-based detection in *ad hoc* networks. We propose a signature-based intrusion detection system dedicated to the ad hoc routing protocols. We exemplify our system on the Optimized Link State Routing (OLSR) protocol [5] focusing on a link spoofing attack. The general idea lies in monitoring locally the routing activity so as to detect any preliminary sign of suspicious activity, which is materialized as a set of events that match, possibly partially, an attack signature. Then, if the local observations are not sufficient, additional evidences are gathered from other nodes. Rather than sniffing or inspecting the incoming or outcoming traffic, as it is the case with any of the detection systems that have been proposed in order to operate in *ad hoc* networks, we take advantage of the audit logs that are generated by the routing protocol. While not requiring changes in the implementation of the routing protocol, this approach permits to take advantage and leverage the work that is already achieved by the routing protocol. Logs are parsed so that patterns of events that characterize intrusion attempts are identified. In order to minimize the number of investigations, events are categorized, and depending on their level of criticality, distributed and cooperative investigation is whether conducted to glean additional observations (and possibly attack evidences). The main challenge stems from the difficulty involved in stating the occurrence of an attack based on second-hand evidences that may come from colluding attacker(s). To tackle this issue, we introduce an entropy-based trust system that evaluates the trustworthiness of the nodes that provide the evidences. As a result, the robustness of the detection is increased. In addition, we introduce the notion of *confidence interval*, which is intended to measure the amount of confidence in a detection-related decision given an uncertain environment. Whereas the evaluation of the intrusion system is found in [6], preliminary evaluations of the trust system are herein provided.

The reminder of this paper is organized as follows. We first survey attacks on *ad hoc* routing protocols (§II). Grounded upon the defined intrusion signatures, we present our intrusion detection system (§III). Then, we illustrate the trust system (§IV) and evaluate its performance (§V). Finally, we conclude with a summary of our results along with directions for future works (§VII).

II. Vulnerabilities

In *ad hoc* networks, many attacks threaten the routing functionality. This comes from the fact that (i) no security countermeasure has been decided as part of the RFCs that specify the behavior of routing protocols, (ii) the absence of a centralized infrastructure complicates the deployment of preventive measures e.g., firewalls, and (iii) devices operate as routers, which facilitates the manipulation of messages and more generally the compromising of the routing. Thus, routing protocols constitute a key target for the intruders. In order to illustrate our presentation, let us exemplify attacks on a

specific protocol: the Optimized Link State Routing (OLSR) [5].

A. Background on OLSR

OLSR aims at maintaining a constantly updated view of the network topology on each device. One fundamental is the notion of multipoint relay (MPR): each device selects a subset of 1-hop neighbors, the MPRs, that are responsible for forwarding the control traffic. The idea is to select the minimum number[1] of MPRs that cover 2-hops neighbors so as to reduce the number of nodes retransmitting messages and hence keep to a minimum the bandwidth overload. In practice, a node N selects MPRs among the 1-hop neighbors that are announced in periodic heartbeat messages, termed $hello$ messages. Then, a *Topology Control* (TC) message intended to be diffused in the entire network, is created by the selected MPR(s). In this message, a MPR declares the nodes (including N) that selected itself to act as a MPR. Then, any device can compute the shortest path, represented as a sequence of MPRs, to any destination. In addition, last versions of the specification support a node holding several network interfaces which are declared (if many) in a so-called MID (Multiple Interface Declaration) message. This message is broadcasted regularly by MPRs so that one another maps multiple interfaces with a main address, hence permitting a unique identification. Additional extensions have been devised in compliance with the above-summarized core functions. Examples include (i) dealing with the nodes that commit (or not) to carry the traffic for others, and (ii) supporting interconnection of an OLSR MANET with another routing domain. Overall, these core and auxiliary functionalities are together subject to various attacks.

B. Attacks Targeting OLSR

OLSR is vulnerable to several attacks, which are hereafter classified [7] according to the action which is undertaken on the routing:

- *Drop attacks* consist in dropping routing message(s) rather than relaying it.
- *Active forge attacks* generate novel and deceptive routing message(s).
- *Modify and forward attacks* modify a received routing message(s) before forwarding it.

Drop attack is characterized by a node I, potentially an attacker or a selfish node, which drops a message instead of relaying it (i.e., I does not forward it within the maximum allowed period). Threatened messages are restricted to the messages that are created and relayed by MPR(s); an attempt to drop all these packets is termed *black hole* whereas selective dropping is named *gray hole*. Rather than dropping traffic, an opposite behavior consists in introducing (falsified) routing information.

Active forge comes from a node that injects novel and deceptive routing messages. Among others, the (broadcast) storm aims at exhausting resources (e.g., energy). For this purpose, an intruder I forges a large number of control messages CM within a short period of time. This attack may be conducted in a distributed manner with several nodes colluding so as to emit (a large number of) messages. Such an attack is characterized by a high visibility and is hence typically coupled with a masquerade: I spoofs[2] the identity of another node. Identity spoofing is not limited to masquerading: it may be intended to create conflicting route(s) and loop(s). In addition, it may be coupled with a modification of the willingness field so as to impact the selection of MPR. Recall that MPRs are selected among the nodes with the highest willingness and in case of multiple choices, the node providing a reachability to the maximum number of 2-hops nodes is primarily selected, then an intruder setting the willingness attribute to $will\_never$ (resp. $will\_always$), ensures that the target is never (resp. always) selected as MPR.

In addition, active forges cover the introduction of tampered messages. This tampering typically keeps the message syntactically correct and focuses on the routing information that are central to the establishment of the shortest paths, i.e., the link state information included in the Hello message, the topology declaration provided in the TC message, the external route(s) in the HNA message and the multiple network interface(s) declared in the MID message. For instance, the neighboring adjacency may be perverted by a node I, which advertises falsified neighboring adjacency in the hello message so as to impact the selection of MPR(s). Upon the reception of a falsified message, local routing tables are corrupted and may contaminate the network itself as well as any interconnected routing domain if a gateway exports those OLSR routes. Note that the gateway may also forge itself wrong routes. Generally speaking, similar tampering may be performed by a MPR relaying control messages (and is henceforth omitted hereafter due to the lack of space).

Modify and forward attacks are characterized by an intermediate that captures the control message and replays or/and modifies this message before forwarding it. Replaying a message includes delaying (i.e., forwarding latter potentially in another area) and repeating this message. As a result, routing tables are updated with obsolete information. Both attacks can be performed in a distributed manner with two intruders: one recording the message from one region so as to replay it in another region (i.e., the one of the colluding intruder); this leads to the creation of a *wormhole*. In order to stay invisible, both intruders may keep the identification field unchanged: the source is still S. Note that sequence numbers constitute a standard mechanism that provides protection against replay attacks. Based on those numbers, a node identifies freshest information, prevents duplicates and replaying while indicating insertion/deletion. In counterpart, there usage may be hijacked. For instance, an intruder I may forward the message including an increased sequence number. Thus, the source assumes that I provides the freshest route.

[1] Redundant MPRs may be selected to increase the availability.

[2] This case should be distinguished from a node that holds several interfaces (and hence several identities) and advertises these latter in the dedicated MID message.

III. Intrusion Detection

We propose a distributed, log- and signature-based intrusion detection system that periodically analyses the OLSR logs. These logs characterize the activities of OLSR (e.g., packet reception, MPR selection). From a practical perspective, the advantage of a log-based detection is threefold. First, additional logs, e.g., system-, security-related logs, could be integrated and correlated. Second, no change in the implementation of the OLSR protocol is required. Third, no in-depth analysis of the packet that are changed is required. Once parsed, a log is used so as to detect a sign of suspicious activity. This consists in matching the log against predefined signatures; a signature is thought as a partially ordered sequence of events that characterizes a misbehaving activity. Let first exemplify the establishment of a signature based on the link spoofing attack, which constitutes an active forge attack, we purposely developed.

A. Signature of a Link Spoofing Attack

This attack attempts to compromise the routing protocol and in particular the neighboring topology (i.e., adjacent links) that is perceived by nodes. It corresponds to a spoofing attack wherein messages are tampered with incorrect information relating to adjacent links. Generally speaking, this attacks constitutes an illustration of a spoofing attack and shows many similarities with the spoofing attacks wherein topology information (TC messages), network interfaces (MID) or routes (HNA messages) are tempered. It follows that the related attack signature and detection strategy are quite share identical.

More particularly, in order to perform a link spoofing attack, an intruder I forges at t' a *hello* message, which declares 1-hop and symmetric neighbors NS'_I differing from the real ones NS_I: $S \xleftarrow{hello(NS'_I)_{t'}} I, NS'_I \neq NS_I \Rightarrow I \in \mathcal{I}$. When forging NS'_I, the attacker holds 3 options:

- declaring a non-existing node as a symmetric neighbor, implies that I (or another misbehaving node) is further selected as a MPR (Expression 1): if I advertises a non-existing node N ($N \notin \mathcal{N}$ with $\mathcal{N}$ defining the set of nodes composing the OLSR network), I ensures that no other (well-behaving) MPR claims being a 1-hop symmetric neighbor of N. Recall that MPRs are selected so that all the 2-hops and symmetric neighbors are covered, I is selected as a MPR.

$$\begin{gathered} \xleftarrow{hello(NS_S)_t} S, S \xleftarrow{hello(NS'_I)_{t'}} I, |t'-t| < \triangle t, \\ \exists N \in NS'_I \ni: N \notin \mathcal{N} \cap NS_I \\ \Downarrow \\ I \in \mathcal{I}, \exists I' \in \mathcal{I} \cap NS_S \ni: I' \in MPR_S, \\ Card(NS'_I \backslash NS'_I \cap \mathcal{N}) > 0. \end{gathered} \tag{1}$$

  This is verified as long as no other misbehaving neighbor of S claims the same. Overall, inserting at least one non-existing neighbor ($\exists N \in NS'_I \ni: N \notin \mathcal{N} \cap NS_I$) guaranties that a misbehaving node I' (with $I' \in \mathcal{I}$) is selected to act as a MPR of S ($\exists I' \in \mathcal{I} \cap NS_S \ni: I' \in MPR_S$). In addition to the above, the connectivity of I increases.
- declaring that an existing node is a symmetric 1-hop neighbor whereas it is not the case ($\exists X \in NS'_I \cap \mathcal{N} \ni: X \notin NS_I$). This claiming increases artificially the connectivity of I, i.e., $Card((NS'_I \backslash [NS'_I \cap NS_I]) \cap \mathcal{N}) > 0$. If no (well-behaving) MPR covers S ($\nexists M \in \mathcal{N} \backslash \mathcal{I} \ni: M \in NS_S \wedge X \in NS_M$), then at least one misbehaving node is selected as a MPR of S ($\exists I' \in \mathcal{I} \ni: I' \in MPR_S$). This typically characterizes an attempt to create a blackhole: I introduces a novel path that provisions the blackhole.

$$\begin{gathered} \xleftarrow{hello(NS_S)_t} S, S \xleftarrow{hello(NS'_I)_{t'}} I, |t'-t| < \triangle t, \\ \exists X \in NS'_I \cap \mathcal{N} \ni: X \notin NS_I \\ \Downarrow \\ I \in \mathcal{I}, \\ Card((NS'_I \backslash [NS'_I \cap NS_I]) \cap \mathcal{N}) > 0, \\ \nexists M \in \mathcal{N} \backslash \mathcal{I} \ni: M \in N_S \wedge X \in N_M \\ \Downarrow \\ \exists I' \in \mathcal{I} \ni: I' \in MPR_S. \end{gathered} \tag{2}$$

- omitting an existing 1-hop symmetric neighbor P ($\exists P \in NS_I \ni: P \notin NS'_I$), decreases artificially the connectivity of both P and I ($NS_I \nsubseteq NS'_I$):

$$\begin{gathered} \xleftarrow{hello(NS_S)_t} S, S \xleftarrow{hello(NS'_I)_{t'}} I, \\ |t'-t| < \triangle t, \exists P \in NS_I \ni: P \notin NS'_I \\ \Downarrow \\ I \in \mathcal{I}, \exists I' \in \mathcal{I} \cap NS_S, NS_I \nsubseteq NS'_I. \end{gathered} \tag{3}$$

Overall, such a falsification of the neighboring adjacency perverts the topology seen by S and may impact the selection of MPR(s) of S.

B. Link Spoofing Detection

Obtaining a complete, accurate and timeliness detection of a link spoofing attack is especially memory and bandwidth-consuming: it involves the examination of local logs, the requesting of others so as to collect/correlate additional intrusion evidences, and the matching of the agreed evidences against an intrusion signature. Indeed, in the worst case, a node S should continuously exchanges information about the link states between any 1-hop neighbors and their respective adjacent neighbors. Rather than launching a cooperative investigation upon any changes, we minimize the number of investigations by initiating it only when the event that occurs is relevant to a link spoofing attack. Changes in the 1-hop neighborhoods (e.g., node apparition) are observed by the node itself. In practice, those changes are obtained by analyzing the local logs and they do not require the requesting of other nodes to be established. In addition, changes in the 2-hops neighborhood are restricted to the following:

- A MPR is replaced (Evidence 1, $E1$ for short), which means that a change in the covering of 1-hop neighbors leads to this replacement. This comes from 1-hop neighbor(s), possibly the replacing MPR, that increase(s)/decreases it/their coverage(s) to the detriment of the replaced MPR.
- No MPR replacement takes place but a previously-selected MPR is detected as misbehaving. For instance, messages are dropped, forged or misrelayed by that MPR ($E2$). Overall, a spoofing link also covers the case wherein an intruder continues to advertise identical 1-hops neighbors despite recent changes.
- a MPR is the only one that provides the connectivity to node(s) ($E3$).
- a MPR does not cover its adjacent neighbor(s) ($E4$).
- a MPR provides connectivity to a non-neighbor ($E5$).

$$\begin{array}{cc} (E1 \vee E2)\ , & \text{optional}(E3) \\ \Downarrow & \Downarrow \\ E4 \vee E5 & (!E4 \wedge !E5) \\ \Downarrow & \Downarrow \\ \text{The suspicious MPR} & \text{The suspicious MPR} \\ \text{is an intruder.} & \text{is well-behaving.} \end{array} \quad (4)$$

Note that contrary to case $E1$, others cases are not necessarily event-driven and may be handled by launching periodical/random checks. The occurrence of ($E1$) or ($E2$) is the starting point for further investigation. Note that a MPR that is the only one that provides the connectivity to node(s) ($E3$) is suspicious but this condition is not sufficient to launch an investigation: (i) this situation is typical in a sparse network and (ii) 2 nodes within communication range often fail in communicating due to the unpredictable nature of wireless transmission resulting from, e.g., obstacles, noises. Thus, diagnosing $E3$ is especially difficult under no specific assumption. Overall, the occurrence of either $E1$ or $E2$ and optionally $E3$ leads to an in-depth investigation. In practice, the investigator interrogates the 1-hops neighbor(s) of the suspicious MPR so as to discover whether the suspicious MPR does not cover its neighbors ($E4$) or advertises a distant node (E5). If all the requested nodes confirm (resp. infirm) $E4$ or $E5$, then the MPR is suspected (resp. well-behaving). Note that, if part of those requested nodes express a different opinion, the number of these nodes and their reputations is taken into account (as established in §IV).

C. Cooperative Investigation

The cooperative investigation is intended to gather intrusion evidences that are matched against an intrusion signature. This investigation (Algorithm 1) is conducted as follows. First, the 2-hops neighbours that have shown their MPR(s) changed, are established. For this purpose, replacing MPR(s) and replaced MPR(s) are computed (lines 2, 3); the 2-hop neighbors that are covered by both are established (line 4). In practice, this interrogation of a 2-hops neighbor Si consists in sending a request to Si so that this latter provides in return its local topology (including the status of the link between I and Si). The request and the related answer together should not go through both the suspicious MPR I or a colliding intruder I'_j. This avoidance is required to prevent I and I'_j from dropping the request and/or simply forging a defective answer. But, this cannot be guaranteed: it depends on the network topology. Note that if no alternative path is available, then one fall into case $E3$. In order to avoid I and I'_j, another MPR that also covers the requested 2-hops neighbors is provided the request. If no answer is obtained (i.e., when the related time-out elapses), then the demand is sequentially transferred through the rest of the covering MPR, and if no MPR is left, then a (multi-hops) alternative path is researched in the routing table to reach Si. Note that this interrogation is performed within a thread so that the investigation of one node (and the result waiting) is not blocking for others. If no answer is provided at all about the suspicious MPR, then the suspicious MPR is tagged as not verified. If no deny is provided, the suspicious MPR I is defined as well behaving. Otherwise, if Si denies the fact that the suspicious MPR I is a 1-hop symmetric neighbor then that link between Si and I is controversial (Si may have returned an incorrect answer and/or I initiated a link spoofing). To takle the issue of distinguishing, we propose a trust system.

Algorithm 1 : Advanced Investigation

```
1: SuspiciousMPRs = new (MPR)
2: OldMPRs = getReplaced-MPR();
3: for (suspiciousMPR ∈ SuspiciousMPRs) do
4:    Common2HopsNeighbors = getCommon2HopsNeighors(suspiciousMPR, OldMPRs)
5:    for (2HopsNeighbor ∈ Common2hopsNeighbors) do
6:       if (verifyLink(2HopsNeighbor, suspiciousMPR)== false) then
7:          Desagreeing2HopsNeighbor (suspiciousMPR) += suspiciousMPR;
8:       else
9:          Agreeing2HopsNeighbor(suspiciousMPR) += suspiciousMPR;
10:      end if
11:   end for
12:   SuspiciousMPRs = SuspiciousMPRs - suspiciousMPR;
13: end for
```

IV. TRUST SYSTEM

A trust system is useful when there exists an uncertainty which prevents from accurately establishing a judgment. A trust system entails two main activities: the establishment of a trust relationship (§IV-A) and the dynamic update of this existing relationship. For this purpose, the system makes use of the observations provided during the investigation.

A. Trust Establishment

A trust relation $T_{A,I}$ established between two nodes A and I represents how much A beliefs that I acts as expected. This belief is built according to I 's previous activities [8]. Indeed, based on the evidences that are collected, the system evaluates the trustworthiness. Generally speaking, several properties should be taken into account during the establishment of the trustworthiness:

- **Properties 1**: the beneficial activity that is performed by a node increases the trustworthiness of that node, whereas

a harmful activity decreases the trustworthiness. Examples of beneficial activity includes the normal relaying of the traffic. In contrast, an harmful activity related to e.g., an intrusion or the supplying of an incorrect answer/feedback to an investigation request.

- **Properties 2**: the degree of gravity (versus reputability) of a harmful (versus beneficial) activity influences the risk for other nodes and hence should be reflected in the establishment of the trust value.
- **Properties 3**: the risk characterized by the imminence of the intrusion decreases drastically the trustworthiness. This risk is established based on the sequence of evidences characterizing the evolving of the attack
- **Properties 4**: fresh activities should be privileged in opposition to stale activities.
- **Properties 5**: first hand evidences (i.e., evidences that are gleaned by the node itself) are privileged comparing to the second hand information which are subject to controversial.

The above properties are enforced as follows. A node A calculates the trust $T_{\Delta t}(A, I)$ of a node I based on the n evidences $e_1^{A,I}, ..., e_i^{A,I}, ..., e_n^{A,I}$ about I that are collected by A during a time slot Δt:

$$T_{\Delta t}^{A,I} = \sum_{j=0}^{n} \alpha_j e_j^{A,I} + \beta T_{\Delta(t-1)}^{A,I} \tag{5}$$

Beneficial and harmful evidences $e_j^{A,I}$ take respectively positive and negative values (property 1). A weighting factor α_j pondered $e_j^{A,I}$ so as to reflect the degree of gravity/reputability of this evidence (property 2) as well as the risk (property 3). Meanwhile, a forgetting factor β permits to privilege fresh evidences rather than the stale evidences that were computed at the previous time slot $\Delta(t-1)$ (property 5) .

When the observations of A are not sufficient, additional evidences provided by other nodes are gleaned. These evidences are less reliable than the local observations. Thus, an uncertainty is involved. To compute such an uncertainty, we rely on the *entropy*, a measure of uncertainty stated in information theory [9]. As in [10], [11], we establish trust through a third party (termed *concatenated propagation*) and through recommendations provided by multiple sources (called *multipath propagation*). Let a *recommendation* $R_{A,S}$ represent how much A trusts the recommendations generated by S. A builds its belief about I according to a third party S's recommendation as follows:

$$Tc_{\Delta t}^{A,I} = R_{\Delta t}^{A,S} T_{\Delta t}^{S,I} \tag{6}$$

When multiple nodes $S_1, S_2, ..., S_m$ generate a recommendation, A builds its belief about I as follows:

$$Tm_{\Delta t}^{A,I} = \sum_{i=1}^{m} w_i . R_{\Delta t}^{A,S_i} . T_{\Delta t}^{S_i,I} \tag{7}$$

where $w_i = \frac{1}{\sum_{j=0}^{m} R_{A,S_j}^{\Delta t}}$

Overall, the trust relationship is used to secure the intrusion detection and guide the decision making.

B. Trusted Intrusion Detection

The trustworthiness of the node(s) that provide second-hand observations is taken into account so as to prevent as much as possible misbehaving nodes from foiling the detection. The objective is to favor the observation provided by trustworthy nodes while being detrimental to misbehaving nodes. A misbehaving node is disserviced unless this latter ameliorates its trustworthiness.

In practice, let consider an adjacent link that is established between a suspicious node I and its neighbor S_i and that is potentially subject to a link spoofing attack. The investigation results in a contestation between I and S_i about this link. In order to establish whether I is launching a link spoofing, the second-hand evidences $e^{S_1,I}, ..., e^{S_i,I}, ..., e^{S_m,I}$ that are provided by the m 1-hops neighbors of I $(S_1, ..., S_m)$ are aggregated (Formula 8): each evidence $e_i^{S_i,I}$ is pondered with a weighting factor along with the trust T_{A,S_i} that the investigator A shares with the requested S_i.

$$Detect_{\Delta t}^{A,I} = \sum_{i=1}^{m} w_i T_{A,S_i} e_i^{S_i,I} \tag{8}$$

with $w_i = \frac{1}{\sum_{j=0}^{m} T_{A,S_j}}$.

An evidence $e_i^{S_i,I}$ is either equal to 1 (meaning that link that is advertised by I is correct and there is no spoofing attack lead by I) or in the contrary -1 (meaning that the advertised link is wrong). Note that if a requested node S_i does not return an answer (before the waiting time expires) then $e_{S_i} = 0$. As a result, an attack that is carried by I by falsifying a specific link $I - S_i$ if $Detect_{\Delta t}^{A,I}$ is nearly equal to -1. Once stated, this result is used to update the trust related to I and $S_1, ..., S_m$.

C. Level of Confidence

One difficulty comes from the fact that the environment is composed of both well-behaving and misbehaving nodes, and is typified by its unreliable nature coming from e.g., the high level of collisions. Thus, confirming or refuting the occurrence of an intrusion relying on second-hand and possibly partial evidences is error prone. It is hence critical to measure the level of confidence on the detection in such an uncertain environment. For this purpose, we use the *confidence interval* [12] that corresponds to an estimated range which is likely to include an unknown population parameter, the estimated range being calculated from a given set of sample data. Herein, based on a partial set of evidences $e_1, ..., e_n$ (namely the sample), our objective lies in estimating a range wherein the overall population of evidences is likely to fall. This range, called the confidence interval, is given by $[Detect_{\Delta t}^{A,I} - \varepsilon, Detect_{\Delta t}^{A,I} + \varepsilon]$ with ε defining the allowed margin of error. The likelihood that the overall population falls into the confidence interval corresponds to a probability, e.g., 95%, named confidence level (cl for short). This confidence level is a configuration

parameter of the trust system. Given a standard deviation σ and the standard density z of a normal law, the margin of error ε is calculated as follows:

$$\varepsilon = z\frac{\sigma}{\sqrt{n}} \tag{9}$$

With a $\sigma = \sqrt{\frac{\sum_{i=0}^{n}(\bar{m}-e_i^{A,S_i})^2}{n-1}}$ and $\bar{m}$ corresponding to the mean. It follows that the higher is the confidence level, the wider gets the confidence interval. Overall, the confidence interval depends of three factors: the required confidence level that is parameterized, the number of observations that is provided, the spread of the observations. As such it permits to guide the decision of establishing if an attack takes place whereas a limited number of recommendations is provided by potentially well-behaving and misbehaving nodes. If the confidence interval is too wide then more evidences should be provided to estimate the trustworthiness in routing (or other operation) decisions. For this purpose, we apply the following rule:

$$\begin{cases} I \text{ is well-behaving} & \text{if} \quad \gamma \leq Detect_{A,I} - Ci \leq 1 \\ I \text{ is intruder} & \text{if} \quad -1 \leq Detect_{A,I} + Ci \leq -\gamma \\ I \text{ is unrecognized} & \text{if other} \end{cases} \tag{10}$$

When the investigation falls into the unrecognized range then more evidences should be collected in order to confirm/ refuting the existence of the intrusion. However, linking the investigation to the trustworthiness of the nodes increases the accuracy of detection as it will be shown in the next section.

V. Performance Evaluation

The performances of our trust system are evaluated with regard to the percentage of attackers and liars, and the required time (expressed as a number of investigation rounds) that is necessary to detect the intrusion and establish the trustworthiness. Experiments are performed as follows. We consider 16 nodes including 1 attacker which performs a link spoofing attack and 4 (i.e., 26.3%) colluding misbehaving nodes (liars) that do not perform link spoofing but that foil the detection by providing incorrect answers. Initially, we randomly set the trust that is assigned to each node. The intruder launches a link spoofing attack that, unless specified, takes place during the overall experiment. Similarly, misbehaving nodes lie during the overall experiments. Figure 1 provides the trust values as seen by the node that is attacked. The constant maintaining of the well-behaving and misbehaving explains the (monotonous) ascending versus descending rate of the trust assigned to those nodes. One may note the defensive nature of our trust system which is characterised by the fact that the trust value assigned to a liar decreases largely regardless of its initial trust value. While the well-behaving nodes which have low initial trust values gain a little of trustworthiness during the 25 rounds. Then, if the attack ceases (Figure 2), the impact of the forgetting factor on both the liars and the well-behaving nodes is provided. One may note that the nodes with high or medium initial trust values reach the default (initial) trust

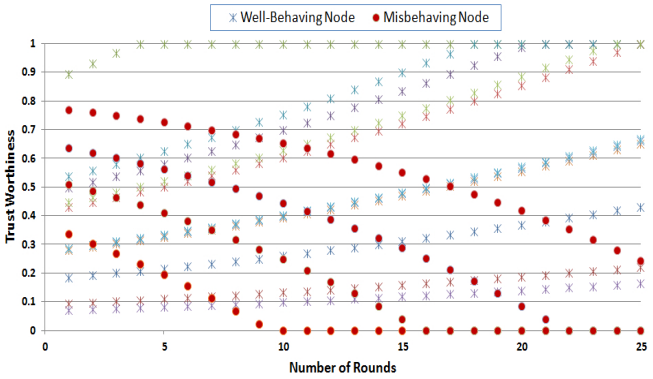

Fig. 1. Trustworthiness.

value (herein 0.4) in the last rounds. While the nodes that have initially a low initial trust value are recovered slowly and consequently may not reach this value. This represents again the defensive nature of our trust system which demands a long misconduct-less duration before trusting a former liar.

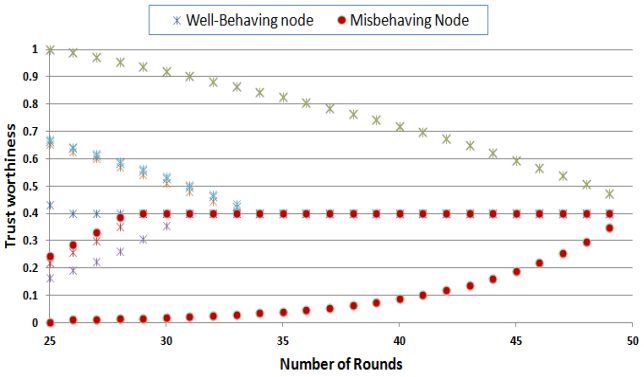

Fig. 2. Impact of the Forgetting Factor on the Trustworthiness.

The impact of liars on the investigation is shown in Figure 3. As expected, the greatest is the number of liars the slowest gets the detection. However, after 10 rounds, the result of the inves-

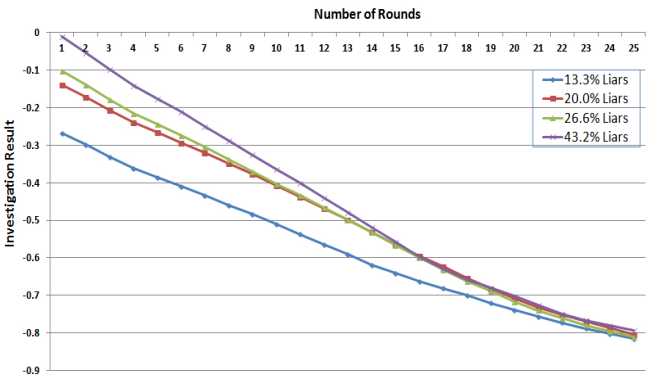

Fig. 3. Impact of liars on the detection.

tigation falls down to -0.4 even when liars represent 43.2% of the nodes. Furthermore, in the last rounds, the investigation

converges and reaches -0.8 regardless of the percentage of liars. This comes from the fact that the trust provided to liars diminishes along the investigation and consequently, liars have almost no influence on the investigation in the last rounds.

Overall, node cannot misbehave and keep its trustworthiness at a high level. Hence, the impact of liars on the investigation shrinks along the time. It is worth to be mentioned that an investigation does not necessary last until that all liars are assigned with a low trust value. It is rather terminated at any round by confirming (resp. denying) the existence of a link spoofing when the investigation result exceeds for instance -0.6 (resp. 0.6).

VI. Related Work

Securing *ad hoc* routing protocols by relying on a trust system constitutes a popular approach. In [13], [14], the DSR protocol [15] is protected against blackhole. Each node overhears the communication of its neighbors in order to monitor the number of dropped packets. If that number exceeds a given threshold, the dropper is identified as distrustful and the routes going through that dropper is either negatively rated [14] or eliminated [13]. More sophisticated trust model based on a modified Bayesian distribution is applied in [16] to protect DSR against dropping. For this purpose, nodes are organized in clusters, each governed by a cluster-head that is responsible for giving a trust certificate to a node according to the recommendations provided by its cluster's members. A subjective logic-based trust system is further used to secure the AODV protocol [17]. When a node receives a message that has gone through an untrusted node, it demands a digital signature to the source of this message. If not provided, the node ignores this message. CAP-OLSR [10] relies on an information theoretic trust system to prevent the OLSR protocol from collusion attack. A node A, which holds I as a MPR asks its 1-, 2-hops neighbor(s) whether I relays its TC messages. Based on the returned observations, A calculates the entropy-based trust of I. If the resulting trust is lower than a given threshold, then I is excluded from MPRs. In order to deal with false recommendations/accusations, only first-hand information (as it is the most trustful) are used in [14]. As a result, the system is incapable of calculating the trust for the nodes with no previous interactions whereas building the trust relationship is time-consuming. To tackle this issue, in [18], less weight is assigned to the second-hand recommendations in a Bayesian-based trust system wherein the prior distribution is a Beta function. These second-hand recommendations results from exchanging periodically self-observations between neighbors. Herein, recommendations that are not close to the actual trust value are rejected; the threshold is set using ordinary differential equation [19]. The previous system is enhanced in [20] by i) giving less weight to the old evidences to allow reputation fading and ii) maintaining for each node two indicators, its reputation (i.e., node trustworthiness) as well as recommendation trustfulness. Only trustful recommendations are considered. In [21], authors rely on an objective-based trust system to resist against attacks that aim to isolate a node. They used a modified Bayesian approach wherein a recommendation is weighted according to the trust value of its source and expires in an exponential manner.

Other works aim to restrict packets relaying within the trustful nodes [11], [22], [23], [24]. In [11], an entropy-based and probability-based models are used to merge the recommendations about a target node. While in [22] trust relationships between the nodes are modeled by a weighted direct graph wherein the vertices represent the nodes and the weighted edge represents the trust values assigned to these nodes. Node-based Trust Management (NAT) [24] aims to choose the most trustful routes in a cluster-based MANT wherein a trusted authority generates symmetric keys. These keys are supplied to the nodes in order to encrypt/decrypt the recommendations that are exchanging in the network. [23] distinguishes itself by using the threat information that are generated by local IDSs as evidences of trust. These evidences are exchanged periodically in form of reports between the nodes. The trust value assigned to a target node is calculated as the average of the reported trust values. Herein, a report is pondered according to the trust value of its source, the distance in hops toward its source and the freshness of the report.

Synthesis and Discussion:

Trust systems depend on self-observations and/or recommendations so as to build trustworthiness between the nodes, henceforth misbehaving nodes are avoided. Part of these systems ensures the robustness against falsified recommendations, which may foil trustworthiness evaluation, by giving low weight for both second-hand and past observations. Overcome, they do not take into account the case when some recommendations are missed given an uncertain environment, hence the trustworthiness evaluation is based on a limited number of recommendations. Our trust system, which aims to secure intrusion detection-related operations, tackles this issue by proposing the concept of *confidence interval*. This concept permits to measure the level of confidence on the trustworthiness evaluation according to the number and the coherence of the available evidences/recommendations.

VII. Conclusion

Signature- and specification-based detection are unpopular comparing to anomaly detection. This calls for consolidating efforts on signature- and specification-based detection while following the *habitus* that lies in coupling detection systems together. To meet this requirement, we define the signatures of the attacks targeting OLSR. These signatures are utilized by a log-based, distributed intrusion detection system. This system distinguishes itself by analyzing the logs generated by a routing protocol so as to extract intrusion evidences. These latter are compared against predefined intrusion signatures and may activate the investigation according to their degree of suspicion/gravity. In order to exemplify our system, we have developed a link spoofing attack, build the related detection rules, and evaluate the performance of the proposed system.

To prevent misbehaving nodes from providing falsified evidences/answers during the investigation, an entropy-based trust

system is utilized. This trust system evaluates periodically the trustworthiness of each node according to its behavior. Then, this trustworthiness is taken into account so as to prevent as much as possible misbehaving nodes from foiling the detection. The objective is to favor the observation provided by trustworthy nodes while being detrimental to misbehaving nodes. On difficulty is that this environment is not only composed of misbehaving nodes but is also typified by its unreliable nature coming from to e.g., the high level of collisions. Thus, confirming or refuting the occurrence of an intrusion relying on second-hand and possibly partial evidences is error prone. It is hence critical to measure the level of confidence that is put in the detection in such an uncertain environment. For this purpose, we proposed to rely on the *confidence interval*. According to this indicator, the robustness of the detection-related decisions is increased.

Overall, experiments show that the trust system efficiently distinguishes between misbehaving and well-behaving nodes during the investigation. The impact of misbehaving node is further fading along the time because its trustworthiness decreases continuously. This leads to enhance gradually the accuracy of detection. Furthermore, recovering from a negative trustworthiness requires that the node well-behave for long time.

In the near future, we envisage to use different weighting of the evidences according to their gravity/reputability. In addition, more experiences are planned in order to evaluate the impact of mobility on trustworthiness evaluation, and the resource consumption that is related to the trust system.

REFERENCES

[1] F.Cuppens, N.Cuppens-Boulahia, S.Nuon, and al., "Property based intrusion detection to secure OLSR," in *IEEE ICWMC*, 2007.

[2] W. Cohen, "Fast effective rule induction," in *ICML*, 1995.

[3] A. Adnane, R. Sousa, C. Bidan, and al., "Autonomic trust reasoning enables misbehavior detection in OLSR," in *ACM SAP*, 2008.

[4] G. Vigna, S. Gwalani, K. Srinivasan, and al., "An intrusion detection tool for AODV-based ad hoc wireless networks," in *ACSAC*, 2004.

[5] T. Clausen and P. Jacquet, "Optimized link state routing protocol (OLSR)," IETF experimental RFC 3626, October 2003.

[6] M. Alattar, J. Bourgeois, and F. Sailhan, "Log based link spoofing detection in manet," CEDRIC laboratory, CNAM-Paris, France, Tech. Rep.

[7] N. Peng and S. Kun, "How to misuse AODV: a case study of insider attacks against mobile ad-hoc routing protocols," *Ad Hoc Netw.*, vol. 3, no. 6, 2005.

[8] F. Azzedin and M. Maheswaran, "Evolving and managing trust in grid computing systems," in *Proceedings of the IEEE Canadian Conference on Electrical & Computer Engineering (CCECE 02)*, 2002.

[9] T. M. Cover and J. A. Thomas, *Elements of information theory*. John Wiley and Sons, 2006.

[10] M.N.K.Babu, A.A.Franklin, and C.S.R.Murthy, "On the prevention of collusion attack in olsr-based mobile ad hoc networks," in *Networks, 2008. ICON 2008. 16th IEEE International Conference on*, 2008.

[11] YL.Sun, S.Member, Z.Han, and al., "Information theoretic framework of trust modeling and evaluation for ad hoc networks," *IEEE Journal on Selected Area in Communications*, vol. 24, 2006.

[12] M. Smithson, *Confidence Intervales*. Sage University Papers Series on Quantitative Applications in Social Sciences, 2003.

[13] S.Buchegger and J.-Y. Boudec, "Performance analysis of the confidant protocol: Cooperation of nodes - fairness in dynamic ad-hoc networks)," in *3rd ACM international symposium on Mobile ad hoc networking & computing*, 2002.

[14] S. Marti, T. Giuli, K. Lai, and al., "Mitigating routing misbehavior in mobile ad hoc networks," in *ACM Mobicom conference*, 2000.

[15] D. Johnson and D. Maltz, "Dynamic source routing in ad hoc wireless networks," in *Mobile Computing*, 1996.

[16] L.Xu and Y.Zhang, "A new reputation-based trust management strategy for clustered ad hoc networks," in *Proceedings of the 2009 International Conference on Networks Security, Wireless Communications and Trusted Computing - Volume 01*. IEEE Computer Society, 2009.

[17] X. Li, M. Lyu, and J. Liu, "A trust model based routing protocol for secure ad hoc networks," in *IEEE Aerospace Conference*, vol. 2, 2004.

[18] S.Buchegger and J.-Y. Boudec, "The Effect of Rumor Spreading in Reputation Systems for Mobile Ad-hoc Networks," in *WiOpt '03: Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks*, 2003.

[19] J.Mundinger and J.-Y. Boudec, "Analysis of a reputation system for Mobile Ad-Hoc Networks with liars," *Performance Evaluation*, vol. 65, 2008.

[20] S.Buchegger and J.-Y. Boudec, "A robust reputation system for p2p and mobile ad-hoc networks," 2004.

[21] J.Li, R.Li, and J.Kato, "Future trust management framework for mobile ad hoc networks," *Communications Magazine, IEEE*, vol. 46, no. 4, 2008.

[22] G.Theodorakopoulos and J.S.Baras, "On trust models and trust evaluation metrics for ad hoc networks," *IEEE Journal on Selected Areas in Communications*, vol. 24, 2006.

[23] Z.Liu, A.W.Joy, and R.A.Thompson, "A dynamic trust model for mobile ad hoc networks," *Future Trends of Distributed Computing Systems, IEEE International Workshop*, vol. 0, 2004.

[24] R.Ferdous, V.Muthukkumarasamy, and A.Sattar, "A node-based trust management scheme for mobile ad-hoc networks," in *Network and System Security (NSS), 2010 4th International Conference on*, 2010.