Trust-Distortion Resistant Trust Management Frameworks on Mobile Ad Hoc Networks: A Survey

Zeinab Movahedi, Zahra Hosseini, Fahimeh Bayan, and Guy Pujolle

***Abstract*—Trust management is a promising approach to conduct nodes' transactions and establish management interactions in distributed mobile ad hoc networks (MANETs) in which nodes' collaboration is critical to achieve system goals. Lack of centralized management, severe resource constraints (e.g., computing power, energy, and bandwidth), and important network dynamics (e.g., topology changes, node mobility, node failure, and propagation channel conditions) make trust management a challenging task in such a network. Mainly, some trust management basis may be exploited to fulfill new attacks. In this work, we present a holistic view on various trust management frameworks geared for MANETs, capable to handle main existing attacks deceiving trustworthiness computation to mislead trust-based network operations, referred to as trust-distortion attacks. Besides, we propose a taxonomy of main identified trust-distortion attacks based on how the trustworthiness estimation of a node about another node is distorted. Moreover, we provide a holistic classification of main evaluation metrics, which can be used to evaluate and compare such frameworks. For each framework, a unified approach is used to describe the trust model, taking each component required for trust management as a guideline. Moreover, each framework is analyzed regarding its resistance against different trust-distortion attacks, the framework unique features, merits, demerits, and findings. Finally, we compare different trust-distortion resistant frameworks and outline the open issues and future research directions.**

***Index Terms*—Mobile ad hoc networks, trust management, trust-distortion attacks.**

I. Introduction

Mobile ad hoc network (MANET) is a self-organized wireless network formed by a set of mobile devices communicating among themselves without any established infrastructure. Because of the distributed nature of MANETs, nodes must collaborate with each other to support the functions of the network. However, due to the self-organized nature and insufficient resources, nodes in a network can behave selfishly or maliciously for individual interests, for example, refuse to forward packets. Trusting on a misbehavior node can lead to unforeseen pitfalls [45], such as slow network efficiency, high resource consumption and vulnerability to attacks. Hence, a trust management framework (TMF) becomes necessary to allow nodes infer how much they can trust on other nodes' behavior [11]. A trust management framework is composed of knowledge collection, trust level computation and trust establishment components, which interact with each other to establish trustworthy relationships in the network.

Although the trust management enables nodes to fulfill interactions only with trusted nodes, such a framework may be menaced by attackers exploiting the trust management inherent properties, mainly the reliance of networking interactions on other nodes' trustworthiness level. Such attackers endeavor to deceive nodes estimation on other nodes trustworthiness, denying the trust management functionality. Consequently, a trust management framework should particularly care of attacks which target the accuracy of nodes' view on other nodes, referred to as trust-distortion attacks in this paper.

Over recent years, a number of research [11], [14], [15], [19] has been carried out overviewing recent advancements and trends in providing trust management frameworks for MANETs. However, to the best of our knowledge, a survey of trust management frameworks dealing with attacks targeting the trust management inherent properties has not been provided so far. Such a survey allows, in one hand, to consider trust management frameworks' security in design time, identifying mechanisms that should be employed to overcome different trust-distortion attack types, and in the other hand, to lead to a convergent trust management framework which could face several types of such attacks simultaneously. Moreover, existing surveys did not use a unified approach to describe different trust management frameworks. Consequently, these works do not allow the comparison between proposed models.

To address the aforementioned issues within the current literature, this survey provides a holistic view of existing trust management frameworks able to cope with inherent trust framework vulnerabilities. For each such framework, we mainly emphasize the mechanisms and approaches taken for each of its trust management components. Moreover, each framework is analyzed regarding the technics, if any, that make it secure against different trust-distortion attacks. Such a study can lead to a convergent framework, making use of existing models' features while avoiding their shortcomings. To achieve this objective, we also propose a classification of main metrics that can be used to measure and compare the efficiency of trust management frameworks.

This paper proceeds as follows. Section II reports background and related works. Section III presents different trust-distortion attacks. Section IV surveys the existing trust management frameworks, detailing each of their trust components.

Manuscript received August 25, 2014; revised May 11, 2015 and August 23, 2015; accepted September 27, 2015. Date of publication October 29, 2015; date of current version May 20, 2016.

Z. Movahedi, Z. Hosseini, and F. Bayan are with the School of Computer Engineering, Iran University of Science and Technology (IUST), Tehran, Iran (e-mail: Zmovahedi@iust.ac.ir).

G. Pujolle is with the Laboratoire d'Informatique de Paris 6 (LIP6), University Pierre et Marie Curie (UPMC), Paris 75015, France (e-mail: Guy.Pujolle@lip6.fr).

Digital Object Identifier 10.1109/COMST.2015.2496147

Moreover, it analyzes the resistance of each presented framework to different types of trust-distortion attacks. Section V categorizes employed metrics to evaluate the efficiency of trust management frameworks. Section VI compares existing trust management frameworks, discusses their open issues and proposes some design guidelines to converge to a unified trust model resistant to simultaneous and conflicting trust-distortion attacks. In addition, it identifies a number of more complicated trust-distortion threats and proposes some countermeasures. Finally, section VII concludes the paper and outlines future works.

II. Background

In the following, we describe the concept of trust, its main components and the related work on trust management in mobile ad hoc networks.

A. Trust in Mobile Ad Hoc Networks

The concept of *trust*, as originally introduced by social sciences, is defined as the degree of subjective belief about the behaviors of a particular entity [12]. When applied in networking, trust is defined as the belief of a node about the faithful nature of another node based on the evidence obtained from previous interactions of the node under evaluation [14].

Trust is mainly important in distributed context of MANETs, in which network functions rely on the collaborative behavior of nodes. While a naive cooperation in MANETs might lead to low efficiency, high energy consumption and vulnerability to attacks, MANET's inherent characteristics may propel nodes to poorly cooperate to the network for individual interests or intentional maliciousness. A selfish node may attempt to save its battery through dropping data packets or even refusing to forward routing packets to avoid paths from itself to be selected. Malicious attackers may announce good routes via themselves to deny network services for following data packets, e.g. by dropping or modifying such packets. Consequently, a trust management framework is particulary required in MANETs, providing nodes with mechanisms to generate, manage, and exchange trust information, punish selfish/malicious behaviors and encourage the cooperation in the network.

A TMF consists of three main components: *knowledge collection*, *trust level computation* and *trust establishment*. The knowledge collection component provides the information about nodes' behavior. The behavioral data can be obtained from local or both local and remote sources [45]. The local knowledge consists in information nodes have collected by themselves on the behavior of their neighbors. The remote information, referred to as recommendations, consists in the opinion of other nodes about the trustworthiness of a given node. The computation component calculates a trust level for each entity based on the collected behavioral data or trust evidence. The result is the assignment of a trust level, representing how much a node can trust in others. The trust establishment component infers if a node can be trusted based on its trust level.

To ensure the network smooth operation, the TMF should respect MANETs' characteristics and constraints. In the following, the unique properties of trust in MANETs and some required considerations are described:

- Due to the nodes mobility or failure, trust is *dynamic*, incomplete and variable over time [17]. To capture this dynamicity and represent uncertainty, trust should be expressed as a continuous variable [2].
- Since a node may exhibit distinct behaviors over time based on its goals and constraints, trust is considered as a *subjective* property [1].
- Trust is *not transitive*, signifying that the trust on a node does not necessarily imply the trust on its opinions [44].
- Nodes may have *asymmetric* normal or abnormal behavior against each other depending on their capacities (e.g., energy or computational power) [2].
- Trust is *context-dependent* and may differ regarding different networking tasks. For example, a node may be trustworthy in packet forwarding but not with regard to its opinions on other nodes. Consequently, a trust level should be defined per type of misbehavior, reflecting the node nature in different contexts [42].

B. Related Work

A significant number of research papers [3], [26], [38] are dealt with generalized security issues and possible countermeasures in MANETs. As well, a significant body of work has been dedicated to survey existing trust-based defense approaches for mobile ad hoc networks [11], [14], [15], [19], [23], [41], [47].

In [11], authors provided an overview on various potential attacks in MANETs and enumerated metrics used in the literature to measure the performance of existing trust frameworks. In addition, the paper surveys trust management schemes developed for specific purposes, including secure routing, authentication, intrusion detection, access control, key management, and trust evidence distribution and evaluation. The surveyed models are described generally regarding their design purpose, without highlighting the mechanism used by different trust components and their inter-operation to assign trustworthiness values to network entities. Therefore, this survey does not allow to differentiate and compare among different existing solutions. Compared to this survey, we mainly propose a classification of trust-distortion attacks, which is then employed to provide an exhaustive/cohesive survey emphasizing different trust-distortion resistant trust management frameworks. In addition, a holistic classification of metrics used in the literature and some new identified metrics is provided to allow the efficiency comparison among trust proposals.

A fairly comprehensive survey on various trust computing approaches geared towards MANETs is provided in [19]. The authors analyzed the technics proposed for different trust dynamics including trust propagation, prediction and aggregation algorithms, as well as the impact of trust on security services. Although this work describes different approaches that can be used to design each trust management component separately, it does not provide a holistic view of components used by each trust management scheme. Consequently, the surveyed

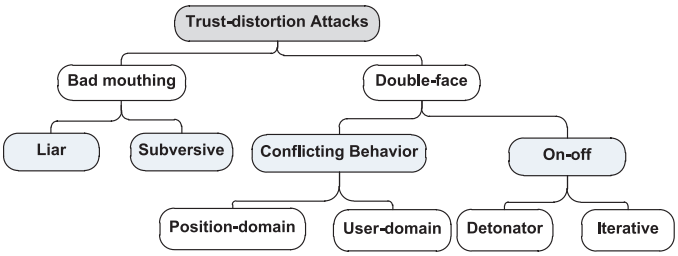

Fig. 1. Trust-distortion attacks.

frameworks could not be analyzed regarding the inter-related components deficiencies.

A recent survey on trust management frameworks able to resist packet forwarding misbehavior is presented in [18]. The authors classified existing solutions based on their resistance against different type of routing attacks, namely blackhole, wormhole and grayhole threats. Existing works are briefly described only with regard to their general scope, omitting their relevant features and key properties. Therefore, it is difficult to compare among several existing frameworks or to identify their open issues.

In [14], a survey of different trust management frameworks for MANETs is presented. The authors classified existing solutions into five groups: protocol based, system level based, cluster based, maturity based and PKI based. However, the proposed categories are based on so common characteristics that make every framework appropriate to almost all categories. Moreover, the paper does not utilize a unified approach to describe each trust management framework and to detail the mechanism used by each of its components. Finally, the paper does not emphasize different trust-distortion attacks, which, in turn, consists in the key contribution of our paper.

III. Trust-Distortion Attacks: A Taxonomy

In this section, we propose a taxonomy of trust-distortion attacks and provide a detailed description of each identified attack types. As depicted in figure 1, we categorize trust-distortion threats into bad mouthing and double-face based on how the attack distorts the trustworthiness estimation of a node about another node. In the following, a description of different trust-distortion attacks is provided.

A. Bad Mouthing Attacks

Bad mouthing attack (also known as *lying attack* [45]) consists in providing dishonest recommendations to deceive nodes trustworthiness estimation about other nodes. Indeed, a bad mouthing attacker can frame good nodes, disseminating false accusation that a disciplined node is malicious or misbehaving [11], also referred to as *blackmailing*. Moreover, it can boost the trust values of malicious peers, transmitting false praises for such nodes [43]. Bad mouthing attack is very common in recommendation based trust computing methods [19].

An attacker may execute only bad mouthing and/or perform another misbehavior(s) simultaneously. The former is referred to as the *liar attack* while the latter is called *subversive attack*. A subversive node, which behaves maliciously in network activities (e.g. dropping packets) beside disseminating dishonest recommendations, can be more easily detected since there are more indications allowing to decrease the trustworthiness level of such a node.

B. Double-Face Attacks

Double-face attack consists in performing misbehavior actions so that the attacker can remain undetected. To achieve this, such an attacker may disperse its attack over time, switch between good and bad actions alternatively, or/and target a particular user or a particular network region in order to deceive nodes perception on other nodes' behavior. We categorize double-face conducts into on-off and conflicting behavior attacks based on whether the threat is temporally or spatially distributed. In the following, each double-face attack type is described.

1) On-Off Attack: In *on-off attack*, bad actions are temporally dispersed in order to adjust the trustworthiness level of an attacker calculated by other nodes above the misbehavior threshold. This latter is performed through switching alternatively between misbehavior (on) and normal (off) conducts to remain undetected while causing damage.

We distinguish between two kinds of on-off conducts, namely detonator and iterative attacks. In a *detonator on-off attack*, an attacker starts misbehavior actions after a period of time of normal behavior and remains in new status forever [34]. *Iterative on-off attack* consists in the generalized form of detonator on-off attack in which an attacker switches iteratively to normal conduct after a certain period of misbehavior actions.

To deal with the time-domain dispersion of on-off attacks and track the resulted trustworthiness level dynamics, the past negative experiences should not be forgotten as quickly as past normal observed actions. The most commonly used technic is to introduce an adaptive forgetting factor inspired from human's behavior, which remembers bad behaviors for a longer time than they do for good behaviors. The advantage of the adaptive forgetting factor is that when an entity turns bad, the trust value can keep up more quickly with the entity's current status. As well, an entity can recover its trust value after some bad behaviors, but this recovery requires many good actions.

2) Conflicting Behavior Attack: In *conflicting behavior attack*, "an attacker node behaves selfishly in a position (or regarding a node) and normally in another position (or regarding another node) in order to remain undetected [25]". We classify conflicting behavior attacks into user-domain and position-domain categories based on whether the attack targets a particular user (or group of users) or a particular region in the network.

In *position-domain conflicting behavior attack*, malicious entities can perform bad actions in the target position and then move to another position conducting normal or abnormal actions. Since the nodes located in new position do not have any previous judgment about that attacker's malicious nature, they will interact with it during the detection time. To address this issue, a trust propagation mechanism distributing

recommendations all over the network allows nodes to timely detect such attackers wherever in the network.

In *user-domain conflicting behavior attack*, an attacker behaves well to all users except a particular user (or group of users) to cause them developing conflicting opinions about the malicious node. In such a case, if nodes' trustworthiness is based only on local information, non-victim nodes remain unaware of the attack, leaving the victim node to face the attacker lonely. However, when recommendations are utilized, the conflicting opinion received from a victim node may lead the evaluator node to distrust the victim node. Consequently, the attacker can keep its trustworthiness level at a high value regardless of its misbehaviors. Meanwhile, the nodes under attack can be excluded from the network [25]. A simple way to detect this kind of attack consists in using the trustworthiness level of recommender to judge about such conflicting behaviors.

IV. Trust-Distortion Resistant Trust Management Frameworks

In the context of this paper, we classify trust management frameworks based on their capability to face bad mouthing attacks or/and double-face conducts. The proposed classification allows to identify technics used to overcome each trust-distortion threat, which can then be used to determine strategies for handling different attack types simultaneously.

In the following, we present a detailed description of existing trust-distortion resistant trust models. Hence, some existing works on trust management frameworks are considered out of the scope of this paper [10], [13], [46] as they are not able to handle none of the trust-distortion threats. Each framework is described regarding the mechanism taken by each of its trust components and their inter-component interactions. In addition, we descriptively analyze the resistance of each scheme to different types of trust-distortion attacks. Finally, we outline the relevant characteristics and important shortcomings of each framework and propose some key technics to render it resistant to all trust-distortion attacks.

A. TMFs Resistant to Double-Face Attacks

In the following, we use the aforementioned guideline to survey different trust management frameworks resistant to double-face attacks.

1) ***Bella****:* The authors in [7] proposed a framework to face double-face attacks based on both local observation and recommendations received from remote nodes.

The *knowledge collection component* gathers local information on trustworthiness of a neighbor based on the traffic received from that neighbor. In order to construct the network-wide knowledge, a Global Reputation Table (GRT), which contains the node's view on neighbors and far nodes is periodically broadcasted to one-hop neighbors. A node uses the received recommendations to update its GRT table before rebroadcasting it to its neighbors after a predefined schedule.

The *trust level computation component* calculates the local reputation of a node j based on the ratio of information forwarded versus generated by j. Since the packets in a bidirectional communication can be constituted also by a simple ack, the number of packets and the amount of data carried by them are counted separately and considered in computation with different weights. The new value of a neighbor trustworthiness in GRT is calculated as the weighted sum of its local reputation (if a neighbor node), its current reputation value in GRT and the received remote value. In order to encourage idle or uncooperative nodes to collaborate, nodes uses an old-age mechanism which decreases the reputation of another node having an unchanged reputation value from the previous step.

Although the Bella efficiency has been proved with respect to detonator on-off attack, this proposal is not immune to iterative on-off conduct. To solve this issue, Bella should consider a mechanism for maintaining attackers misbehaving history so that the re-trust to such an attacker takes place only after a relevant period of good behavior. For any return to misbehavior mode, an observer node should stretch out the required period the attacker should behave trustworthy to be able to be re-trusted. Moreover, Bella is vulnerable to the bad mouthing attack since no mechanism is considered to differentiate between correct and false recommendations.

Furthermore, we believe that the framework gets also immune to the conflicting behavior attack since nodes' trustworthiness is evaluated based on both positive and negative recommendations received from all over the network. Indeed, remote nodes are warned through received recommendations and would relinquish interacting with attacker nodes.

To conclude, Bella consists in a promising TMF resistant to detonator on-off and conflicting behavior attacks with an acceptable amount of generated overhead compared to a poor flooding procedure. Another asset of Bella resists in its local observation method which avoids the energy-demanding promiscuous mode, prolonging the network lifetime. However, Bella is not immune to bad mouthing and iterative on-off conducts. Another issue raises from the local knowledge collection and computation strategy in which the neighbor trustworthiness level is calculated based on the traffic coming through/from that neighbor. First, the detection of a local misbehavior may become a time-consuming process, especially when the network traffic is light. Second, if the attacker drops packets of a particular next hop, the victim node can not detect this behavior. Third, Bella considers the packet generation as a negative factor in local nodes' trustworthiness level which are obviously not accurate. Finally, the old-age algorithm may erroneously punish a normal node with a continuous monotone behavior since it considers any stability in node's trustworthiness as a sign of being idle or non collaborative.

2) ***ATMS****:* Autonomic Trust Knowledge Monitoring Scheme (ATMS) [35] takes into account autonomic principles in order to propose a new self-adaptive and protocol-independent trust management scheme in MANETs. The use of autonomic management paradigm enables the permanent and overall optimization of network resources consumed by the framework in the dynamic network context.

As illustrated in figure 2, the proposed *knowledge collection component* is based on the autonomic MAPE-K model comprising of five main components, namely the monitor,

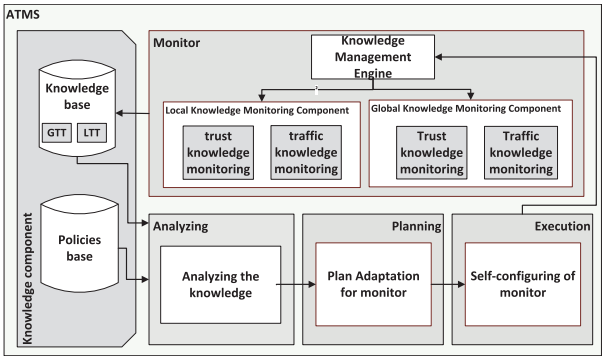

Fig. 2. Autonomic Trust Knowledge Monitoring Scheme (ATMS).

knowledge, analyzing, planning and execution components [51]. The monitor component collects local and global information required for establishing trust relationship among nodes. It is composed of a knowledge management engine, a local and a global trust monitoring modules. Similar to Bella, a node obtains the local wide trustworthiness values based on the amount of traffic it receives from that neighbor. The information acquired by local observation is stored in Local Trust Table (LTT). The global trust monitoring module is in charge of exchanging information with other nodes. Based on the table LTT, every node of the network constructs a Global Trust Table (GTT) which is gradually completed by trust values of all network nodes. The proposed Knowledge management engine uses data or signaling packets traveling in the network to exchange such information. The knowledge management process decides on when it piggybacks trust information on transiting packets based on the load status of the network.

The Knowledge Component consists in a knowledge base, in which LTT and GTT tables are stored, and a policy base, in which a set of predefined condition-event rules are stocked. The policies allow the framework to analyze its current state and adjust its monitoring rate accordingly. The Analyzing Component periodically verifies if any predefined policies threshold is exceeded considering the knowledge provided by the monitor component. If this is the case, it activates the planning component in order to react to that event.

The Planning Component executes the knowledge monitoring optimization algorithm (KMO) when the traffic rate exceeds a congestion threshold. In this case, the global knowledge monitoring process will be executed with a certain probability, denoting whether transiting packets should be piggybacked or not. The Execution Component is responsible for enforcing decisions taken by the planning component. That is, when the planning component decides a new value for monitoring generation rate, the execution component configures the knowledge management engine with this new value.

The *trust level computation component* calculates the local and global reputation of a node in the same way as Bella.

To summarize, ATMS is an autonomic protocol and framework independent trust management scheme which provides a uniform up-to-date trust knowledge throughout the network with a minimum monitoring overhead, reducing the impact of double-face attacks. The main asset of ATMS is the self-adaptation of its knowledge collection component according to the underlying network context, which optimizes the network monitoring cost. However, ATMS does not consider any mechanism to distinguish between correct and false recommendations, which makes the framework vulnerable to bad mouthing attack. Moreover, the framework requires to maintain nodes misbehaving history in order to face the iterative on-off attack. Finally, ATMS should rethink the local trust collection and computation components as already mentioned for Bella framework.

3) ***Almotiri****:* Almotiri [4] proposed a double-face resistant multi-path routing protocol for mobile ad hoc networks.

In the *knowledge collection component*, successful and unsuccessful interactions of neighbors are gathered. To collect the indirect information, the intermediate nodes piggyback their trust table in routing packets. This information is received by the original source through Route Reply (RREP) packets and used to evaluate the trustworthiness of nodes on the path.

In the *trust level computation component*, a node's trust value is increased by one when a good behavior is observed and decreased by a value proportional to the observer node's policy when a misbehavior is detected. When the node's trust value attains 0, it is considered as malicious. The source calculates another trust value, called the route trust, which consists in the average of trust value of the nodes on the path.

As *trust establishment component*, the source selects the safest path with maximum route trust value for its data transmissions. If there is more than one path with the same trust value, the shortest path will be selected.

Almotiri can somehow face the user-domain conflicting behavior attack since the trust value of nodes on the path is available at the source. However, to better cope with this attack, all intermediate nodes on the path require to update their trust table using the information stored in routing packets. Moreover, we believe that the framework would be capable to face the on-off attack, using a policy with a more significant impact for misbehavior actions than good behaviors.

To recapitulate, Almotiri consists in a TMF specialized for enabling secure routing in MANETs. The framework employs the existing routing packets in the network for recommendation dissemination, reducing the amount of extra traffic generated to the network. However, recommendations are not propagated all over the network, making the framework vulnerable to the position-domain conflicting behavior. Furthermore, Almotiri model is vulnerable to the bad mouthing attack since no mechanism is defined to detect false recommendations. Another issue relies on the inappropriate path trust computation, since averaging the nodes' trust value may cause highly dishonest nodes on a path to be compensated by very high trust value of trusty nodes. Finally, the proposed framework is protocol-dependent and relies completely on Dynamic Source Routing (DSR) [24] protocol.

4) ***Data Class****:* Data Class [39] studies the relevance of personal and general data classes as criteria for calculating the nodes' trustworthiness level in MANETs. The information returned by intermediate nodes on the behavior of forwarder nodes across the path is considered personal data to the source. The same data, when received by intermediate nodes is

considered as general. A dynamic nature-inspired model based on replicator equation is used to study the influence of each of these classes on the performance of the TMF. Based on the employed trust data, the framework may use either of these four strategies: only personal trust data, only general trust data, both data classes with the same precedence, or both data classes with higher priority for personal data class compared to the general data class.

The *knowledge collection component* evaluates a neighbor's trustworthiness based on the amount of packet forwarded by that node using the watchdog mechanism. The remote information is obtained through the communication session, i.e. when a packet is transmitted from the source to its destination, the intermediate node observing that the next hop did not forward the packet, sends a warning packet back to the source. When the warning packet attains the source, this latter can identify the trustworthiness of all nodes along the backward path and the misbehavior of the posterior node on the forward path.

The *trust level computation component* calculates the trust level based on the ratio of packets forwarded to packets discarded by a node.

The *trust establishment component* implements several features. First, any intermediate node (forwarder) calculates the source's trustworthiness to decide whether to accept or reject its packets, requiring a minimal level of trustworthiness for the acceptance of a forwarding request. Second, the trust values are used by source nodes to rate the available paths to the destination and select the most secure one.

We believe that Data Class can cope in some extent with the user-domain conflicting behavior attack since nodes overhear their neighbor's behavior, and send the collected data to their previous node in the communication session. However, if the victim node is not on the path, the user-domain conflicting behavior may remain undetected.

To summarize, Data Class introduces a new classification of collected data in the network and uses a feedback-like method for collecting these data. However, the framework can not cope with the iterative on-off attack, assuming that nodes will not change their behavior over time. Moreover, Data Class is vulnerable to bad mouthing attack as the accuracy of trustworthiness information obtained via a previous node on a next node's behavior, can not be verified in a (remote) source node. In addition, the trust information is only disseminated along the packet path backwards to the source. Consequently, the trustworthiness values are not uniformly distributed throughout the network, making the framework vulnerable to position-domain and generalized user-domain conflicting behavior attack.

5) ***OCEAN****:* Observation-based Cooperation Enforcement in Ad hoc Networks (OCEAN) [6] uses only local observations to avoid the complexity of recommendation-based schemes.

The *knowledge collection component* in a node monitors its neighborhood within a timeout after forwarding a packet. The nodes being evaluated as misbehavior are added to a faulty list. A field, called avoid list, is added to the RREQ packet of the DSR protocol in which the source initially adds its faulty list. Any intermediate node appends its faulty list to the avoid list of the RREQ packet.

The *trust level computation component* registers positive and negative events and calculates neighbors ratings accordingly. When the rating of a node falls below a certain threshold, the node is added to the faulty list. As a second chance mechanism, a timeout is used to remove misleading nodes from the faulty list after a fixed period of normal conduct.

The *trust establishment component* is highly coupled with the remote knowledge collection procedure. Indeed, the calculated neighbor ratings are used to avoid routes containing nodes registered in the faulty list. Each node that sees its name in the RREQ's avoid list, suppresses the RREQ. In addition, an intermediate node discards a RREQ or a RREP containing one of the nodes within the avoid list or a traffic received from a misleading entity registered in its faulty list. The described process allows OCEAN to handle attackers which respond positively to routing packets but fails to forward the data flows.

OCEAN uses a chipcount mechanism to face selfish nodes which refuse responding to other nodes' route requests. Each node earns chips in a neighbor's table when it forwards a packet for that neighbor. When a node receives a RREQ from a neighbor, it decides to service the forwarding request only if that neighbor's chipcount is greater than a threshold. Two schemes are considered in the trust level computation component to handle chipcount mechanism: 1) Optimistic scheme in which a node increments the chipcount for its neighbor whenever the neighbor accepts a RREQ packet from that node. 2) Pessimistic scheme in which a node increments the chipcount for its neighbor only when the neighbor is observed to forward the packet. While the pessimistic scheme may result in the deadlock problem, the optimistic counterpart could be too lenient on the misbehaving nodes. The problem with the chipcount mechanism is the famine of chipcount for good nodes receiving less RREQ or data packets. The authors propose a *Chip Accumulation Rate (CAR)* mechanism in which all chipcounts are increased per time unit.

We believe that OCEAN is capable to detect a detonator on-off attacker as soon as it begins its permanent attack phase. However, the iterative on-off attack can destabilize the network operation since positive and negative behaviors are treated in a same way. Indeed, an on-off attacker may remain infinitively undetected if it switches intelligibly to a good behavior before attaining the threshold of being considered as faulty. Moreover, OCEAN can not resist the user-domain conflicting behavior attack except in a situation where a node wants to establish a path including the misbehavior node and the victim node. In addition, as trust information are not uniformly distributed over the network, the network can be disturbed by position-domain conflicting behavior attack. Finally, OCEAN is fragile regarding bad mouthing attacks because all nodes are permitted to add their faulty list to the avoid list carried in RREQ packets.

To conclude, OCEAN is a protocol-dependent framework which uses the local trust to establish routes containing trusty nodes. The framework is immune to the detonator on-off attack and to some particular cases of the user-domain conflicting behavior threat. However, the deletion of the faulty list after a certain timeout makes the framework highly vulnerable regarding all type of threats, especially the iterative on-off attacks. To solve this issue, a solution could be to use an auxiliary faulty list

containing all attackers detected previously, considering a significantly lower threshold to return such nodes into the faulty list. Another parallel precaution can be to give limited privilege to nodes in auxiliary faulty list in early times and increase their network access only after a sufficient time of good cooperations. Finally, the proposed CAR mechanism is not capable to solve the problem of unfairness regarding normal peripheral nodes. Moreover, it may cause selfish nodes to enjoy full freedom in relying their packets since they will never run out of chips at any node.

6) ***TAODV:*** Trusted AODV (TAODV) [29] is a protocol-independent encryption based TMF proposed to handle some double-face attacks in MANETs. The paper is presented considering the AODV protocol [36] as a guideline to describe different trust-related processes.

The *knowledge collection component* is equipped with some monitoring mechanism or intrusion detection units to observe the behaviors of one-hop neighbors. Recommendations are distributed in the network using Trust Request (TREQ), Trust Reply (TREP) and Trust Warning (TWARN) messages. When a node wants to know the new trustworthiness of another node, it will issue a TREQ message to its neighbors. Nodes receiving the TREQ will reply with a TREP message. When a node believes that another node has become malicious or unreliable, it will broadcast a TWARN message. The sender of TREQ messages is Requester. The replier is Recommender, and the recommendation target is Recommendee.

The *trust level computation component* within each node maintains a belief, disbelief and uncertainty opinion about all other nodes. Belief is the probability that the node B can be trusted by the node A. Disbelief designates the probability that the node B cannot be trusted by A. Uncertainty indicates the absence of both belief and disbelief. The sum of these values is one. At the beginning of the network operation, a node's belief and disbelief towards another is 0 and the uncertainty element is equal to 1. After some successful or failed communications, A will update its opinion about B based on collected positive and negative evidences. Belief consists in the ratio of the number of positive evidences to total evidences, while disbelief is calculated as the ratio of negative evidences to total evidences. Uncertainty is computed as the complement of the sum of belief and disbelief.

Different received recommendations about a given node are combined based on whether the composing recommendations are aligned or contradictory. For aligned recommendations, the received recommendations are weighted using the recommender's belief. Regarding the contradictory recommendations, a *Relative Objective Evaluation Consensus Combination* mechanism is used which considers the uncertainty of a recommender as the weight of its opinion. The proposed consensus combination method is designed in a way that the uncertainty of other nodes' trust reduces over time.

The *trust establishment component* interacts to other nodes based on certain belief, certain disbelief or uncertainty situation. In the uncertainty situation, when the uncertainty element in A's opinion towards B is larger than or equal to 0.5, or both the belief and disbelief are lower than 0.5, A uses the cryptographic schemes and requests B the digital signature. In a certain belief situation, when the belief is larger than or equal to 0.5, A trusts B and continues routing. Finally, in the certain disbelief situation, when the disbelief is larger than or equal to 0.5, A distrusts B for an expired time and refuses to perform its routing operations. In addition, the route entry in A's table will be disabled and deleted.

Due to the network-wide TWARN messages and the employed consensus combination algorithm, the framework is resistant to conflicting behavior attacks. Moreover, TAODV can resist the detonator on-off attack in which an attacker continues to behave abnormally, and hence, will be detected after a while. TAODV considers misbehaving and normal behaviors similarly, leading to its vulnerability regarding the iterative on-off attack. In addition, the employed consensus combination algorithm, which reduces the uncertainty of the calculated trust of contrary opinions, can be at the expense of bad mouthing attackers' detection. Indeed, received contradictory opinions may be resulted from lying and, hence, the trust uncertainty of such an evaluated node should be rather augmented. A solution would be to increase the opinion's uncertainty in such a case, pushing nodes to enter the warily mode in which the cryptographic schemes are used for more precautions.

To summarize, TAODV is a trust management framework immune to conflicting behavior attacks. Although the use of consensus algorithm helps protecting from some kind of attacks, it comes in despite of bad mouthing threats. As a main issue, the reactive nature of the recommendation distribution mechanism may delay the routing and forwarding processes as well as the trustworthiness convergence.

B. TMFs Resistant to Bad Mouthing Attacks

In the following, we overview different trust management frameworks resistant to bad mouthing attacks, using the same guideline as previous subsection. Some surveyed frameworks may resist also to double-face attacks as described in the framework analysis presented hereafter.

1) ***OTMF:*** Objective Trust Management Framework (OTMF) [25] makes use of both direct and indirect information to cope with both double-face and bad mouthing attacks.

The *knowledge collection component* monitors the behavior of its neighbors using a watchdog mechanism. The second-hand information can be formed periodically between neighbors and then flooded in the network. The nodes receiving this information will check it by a deviation test and use the trustworthiness of the information provider as the weight for this information when necessary.

The *trust computation component* evaluates the trust for an object node based on a modified Bayesian approach. Due to the storage simplicity, the Beta distribution is used as prior probability which is parameterized by the number of normal behaviors (α) and misbehaviors (β). It is assumed that the subject node believes the object node behaves normally with probability θ. Initially, θ is uniformly distributed between 0 and 1. Then, if there are s observations with normal behaviors and f observations with misbehaviors, the posterior distribution is updated by $\alpha = \alpha + s$ and $\beta = \beta + f$. After training by a large number of observations, θ will be close to $\alpha/\alpha + \beta$, with

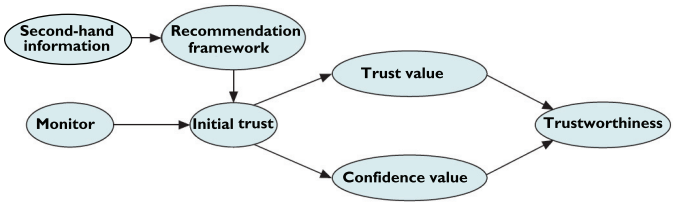

Fig. 3. Objective Trust Management Framework (OTMF).

high probability. If one node performs more normal behaviors, θ will converge to a larger constant, and this node is more trustworthy. Whatever the type of information is, less weights are given to past behavior than more recent behavior. Moreover, second-hand information has less weight than direct information. The trustworthiness of the information provider is used as the weight for its recommendations.

A node forms an elementary opinion for another node using trust value and confidence value parameters. A high trust value indicates that the subject node can trust the object node regarding a particular action, such as forwarding packets. It is computed as the expectation value of the beta distribution. The confidence value measures the accuracy of the calculated trust value. Obviously, a high confidence value represents a more reliable information for making decisions. The confidence value is computed based on the standard deviation of the beta distribution. As depicted in figure 3, a node combines these two parameters into a trustworthiness factor.

We believe that OTMF can resist the conflicting behavior attack since it provides a uniform view on nodes' trustworthiness throughout the net. Moreover, OTMF is immune to bad mouthing attack in presence of subversive nodes, using the trust level of the recommender as the weight for its opinions.

To conclude, OTMF provided a uniform view on nodes's trustworthiness to reduce the impact of double-face attacks. An important feature of OTMF is the use of a confidence value which evaluates the quality of opinions. As an evident shortcoming, the knowledge uniformity is acquired by means of flooding in expense of generating a significant amount of overhead to the resource-constrained MANETs. Moreover, OTMF requires a defense mechanism against the iterative on-off attack, for example, by means of tracking the behavior of an attacker and considering higher weights for bad actions.

*2) **LARS**:* Locally Aware Reputation System (LARS) [21] uses local information to face bad mouthing and on-off attacks.

The *knowledge collection component* monitors neighbors' behavior using the watchdog mechanism. The knowledge collection procedure proceeds as follows: During a communication session, every intermediate node whose neighbor forwards the message, can overhear the forwarding, keeping a record of the message sent and setting a timer. When a message reaches the destination, an acknowledgment is sent back to the source. If the source does not receive the ack within a certain time threshold, it initiates a special trace packet along the same path to identify the misbehaving node. Neighbors of nodes along the path which already have a record of the message will participate in the trace process. For every neighbor node, the record is sent back to the source if the trace packet is received before the timer expiration, allowing to determine the behavior of the participating nodes. The record will be discarded in neighbors which did not overhear a trace packet before the timer expiration. Each node handles a trust table in which the trust level of neighbors is stored.

The *trust level computation component* increases the trust value recorded in the table by a factor of m for any positive participation. If a node fails to forward the message, the neighbors detecting that behavior will decrease its reputation value by a factor of a ($a > m$). If a node drops both the message and the trace packet, its trust value will be decreased by b where $b > a$. If some common neighbors detects that another neighboring node did not report the dishonest behavior (collusion/cheating), they reduce the colluding neighbor's trust value by a factor of c where $c > b$. In all cases, the trust value is a weighted sum of past experiences and the actual behavior where a fading factor is used to give less weight on past experience. A node with a trust value below the untrustworthy threshold is considered as misbehaving node. When a misbehaving node is detected in the trust table, a WARNING message is sent to the k-hop neighborhood. To prevent false accusations and problems caused by inconsistent reputation values, the WARNING message should be signed by all neighbors before it can be broadcasted. Any node within one-hop distance of the misbehaved node can sign the WARNING message if the reputation value of the suspicious node has also dropped below the untrustworthy threshold. As a second chance mechanism, the misbehaved node is accepted after a time-out period but its reputation value remains unchanged so that it has to rebuild its reputation again by good cooperations.

The *trust establishment component* within the source node tries to use an alternative path to avoid the misbehaved node detected via the responses of the trace messages. Moreover, After broadcasting a WARNING message to the k-hop neighborhood, all the k-hop neighbor nodes becoming aware of the misbehaved node will deny its service.

As the WARNING message should be signed by all neighbors before it can be broadcasted (where there is likely at least one normal node in the neighborhood), the framework ensures that a well-behaved node will not become isolated by bad mouthing attack. The framework can resist iterative on-off attacks since bad behaviors are weighted more than normal behaviors, and hence, re-trusting such an attacker takes place only after a relevant period of good behavior.

To summarize, LARS consists in a protocol-independent TMF which can be implemented on top of any on-demand routing protocol. The main drawback of LARS resides in its dependence on knowing exactly the neighborhood. In low mobility, the framework acts efficiently as nodes neighborhood does not change frequently. In contrast, in high mobility scenarios, previous information about mobile node's behavior will be useless. LARS's Trustworthiness knowledge uniformity is very low, i.e. nodes in different positions of the network will have different opinions about one node. Consequently, the framework can not resist position-domain and user-domain conflicting behavior attacks. Another weakness of the framework is that it does not consider accidental message drops.

Moreover, trace and warning messages propagation burden a high amount of overhead to the framework.

*3) **AFStrust**:* AFStrust [48] is a fuzzy-logic based TMF combining the history of a node's behavior and its serving capability (remnant battery, local memory, CPU cycle, bandwidth, etc.) to evaluate the trust level of network nodes.

The *knowledge collection component* collects the local information via the watchdog or any other local monitoring mechanism executed in an interaction interval. The strategy for disseminating recommendations is not specified. The knowledge collection component also monitors other nodes' serving capability and uses it in computing nodes' trust level.

The *trust level computation component* calculates historical trust value and current trust value. Node's *historical trust* is estimated by the node's neighbors based on their historical interaction calculated at the end of each time interval. The historical trust is the weighted average of four decision factors: Direct Trust, Recommendation Trust, Incentive Function and Active Degree. Direct Trust is the weighted average of all interaction evaluations in different interaction intervals giving a higher weight to the more recent information. Moreover, intervals in which more interactions have been conducted are more weighted as they include more information. Every Recommendation Trust received from a direct neighbor (respectively through a path) is weighted by the recommender's direct trust (respectively by the path recommending credibility trust computed as the multiplication of trust values of nodes on the path). Incentive function reflects the incentive for cooperative entities and is evaluated based on the number of malicious interactions out of the total number of interactions. Active degree reflects the level of activity of an entity and is used to indicate the credibility of evaluated entity. The active degree is computed according to the cumulative number of entities interacted with the evaluated node. The Analytic Hierarchy Process (AHP) [37] is used to determine the weights of each of historical trust decision factors. Finally, a node's *current trust* value is derived from the node's historical trust value and node's serving capability level based on the fuzzy logic, predicting the node's trust in the next time interval.

The *trust establishment component* uses a black-list trust threshold below which a node is considered as attacker. The neighbors of a detected attacker will stop interacting with it for a special time period.

AFStrust can face the simple bad mouthing attack in presence of subversive nodes because it uses trust value of a node as its recommendation credibility [16], [32].

To conclude, AFStrust uses the fuzzy-logic to calculate the current trust of a node based on a set of decision factors and its serving capability level. However, the model performance depends on the employed trust knowledge collection strategy which is not specified. Though we can confirm that AFStrust can overcome conflicting behavior attacks in some extent or some simple situations (as it uses recommendations), we can not judge clearly about its resistance to more sophisticated conflicting behavior attacks. Besides, the framework is vulnerable to iterative on-off attack because it considers normal and abnormal behaviors similarly and does not track nodes' misbehaving history for a longer period. Another issue relates to use

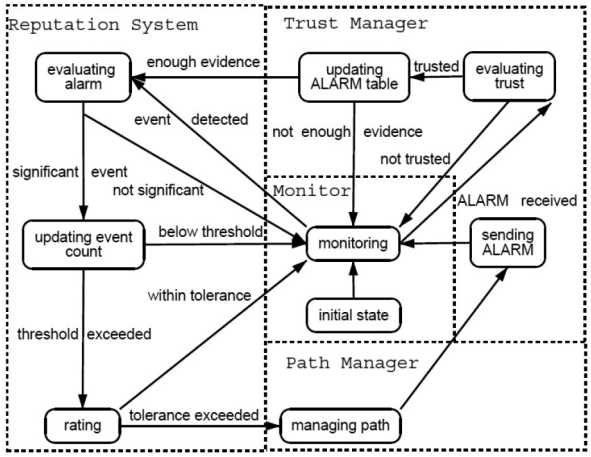

Fig. 4. CONFIDANT framework.

of enormous number of decision parameters and factors which render the framework computationally complex.

*4) **CONFIDANT**:* In CONFIDANT (Cooperation Of Nodes: Fairness In Dynamic Adhoc NeTworks) [9], trust relationships are based on experienced, observed or reported routing and forwarding behavior of nodes.

As depicted in figure 4, CONFIDANT consists of four modules: Monitor, Reputation System, Path Manager and Trust Manager. Each module is implemented within a trust management component and is present in every node. CONFIDANT is resistant to both double-face and bad mouthing attacks.

The *knowledge collection component* is composed of monitor and trust manager. For each node, the *monitor* locally looks for deviating nodes using a watchdog mechanism. By keeping a copy of a packet while listening to the transmission of the next node, any misbehavior in forwarding the packet or content change can be detected. The monitor component registers these deviations from the normal conduct. As soon as a misbehavior occurs, the reputation system is called which is in charge of trust computation. The *trust manager* component deals with incoming and outgoing ALARMs. Outgoing ALARMs are generated by the node itself after having experienced, observed or received a report of misbehavior. The recipients of these ALARM messages are "friends", which are administered in a friends list. We note that the paper does not specify how the friend list can be obtained. The trustworthiness of the source of an ALARM has to be checked before triggering a reaction, thus there is a filtering of incoming ALARM messages according to the trust level of the reporting node.

The *trust level computation component* is implemented through a reputation system which manages a table consisting of entries for nodes and their ratings. The rating is modified only when sufficient evidence on a node's malicious intent is detected to exclude coincidences. To update ratings, CONFIDANT uses a rate function which assigns different weights according to the type of behavior detection, namely "the greatest weight for own experience, a smaller weight for observations in the neighborhood, and an even smaller weight for reported experience acquired via ALARM messages" [9]. Following this process, if the rating of a node in the table

has degraded more than a tolerable range, the Path Manager is called to take an appropriate reaction.

The Path Manager plays the role of the *trust establishment component* and performs the following functions:

- Path re-ranking according to reputation of the nodes in the path.
- Deletion of paths containing misbehaving nodes.
- Ignoring a route request from a misbehaving node.
- Ignoring/alerting the source for a source-route request on a route containing a misbehaving node.

CONFIDANT can resist the conflicting behavior attack by exchanging local rating lists with friends. Furthermore, CONFIDANT can endure the generalized on-off attack because it only considers misbehavior actions in trust calculation. Indeed, the decreased trust of such an attacker during its on period will not be increased during the off interval. Regarding the bad mouthing attack, CONFIDANT avoids false accusations by checking the trustworthiness of the source of an ALARM before triggering a reaction.

As a summary, CONFIDANT is one of the few frameworks applying trust values in routing decisions. A concerning issue of this model relates to the friend list: in one hand, if the friend list is too conservative, the framework may not converge timely to the correct level of trust. In the other hand, if the friend list is defined too optimistically, the framework risks to cooperate with malicious nodes. Moreover, the model needs to exchange the friend list with other nodes which may impose extra traffic to the network. Another unspecified detail concerns the rating function which should be defined according to the quality of good/bad behaviors. Finally, CONFIDANT does not define any second chance mechanism, discouraging repentant attackers to come back to a cooperative conduct.

5) ***mTrust****:* mTrust [28] is a multi-dimensional trust management framework to handle bad mouthing and double-face on-off attacks in MANETs.

The *knowledge collection component* uses promiscuous mode to keep track of all incoming packets received by neighbors. Whenever a node observers a different behavior from a trustworthy neighbor, it will integrate this new observation to its current view and then broadcast the probability of the detected misbehavior to its neighbors. Once all nodes notice that they are not receiving any new observation from their neighbors, the process stops and all the nodes converge to a unique global view of misbehaviors. Periodically, the process is restarted in order to keep the global view up-to-date.

The *trust level computation component* integrates new observations to its current view using the mathematical *Dempster-Shafer* theory of evidence [27], [40], which allows to combine different received evidence to draw out a degree of belief. This component assesses the trustworthiness of a node regarding collaboration, behavioral and reference trust dimensions as illustrated in figure 5. The *collaboration trust* is determined by how collaborative a node would be in some network activities such as route discovery and packet forwarding. The *behavioral trust* is derived by the amount of abnormal behaviors conducted by the node under observation. The *reference trust* is computed based on the correctness of the observation that one node spreads.

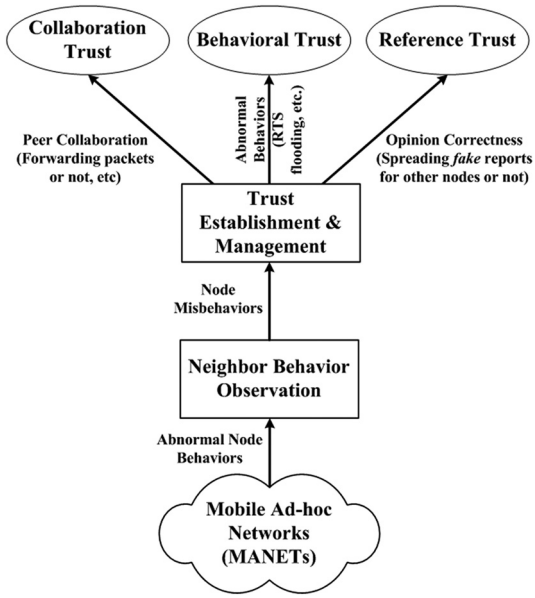

Fig. 5. mTrust framework.

An adaptive trust evolution model is deployed, by which each dimension of the trustworthiness can be adjusted according to the misbehavior's features, such as severity of the outcome, frequency of occurrence, and context in which the misbehavior occurs. Given that the collaboration trust (e.g. packet dropping) may be impacted by both malicious intent and environmental factors (e.g. overflown buffer and exhausted battery), the collaboration trust is reduced at a lower rate when compared to behavioral trust which can be only resulted from a malicious intent (e.g. packet modification). Similarly, as it is really harmful to spread fake observations, the reference trust decreases at the highest rate when compared to both collaboration and behavioral trust. Based on these facts, logarithmic model, linear model, and exponential model is utilized for collaboration trust, behavioral trust and reference trust, respectively. It is worth mentioning that the neighbor' reference trust is used as the weight of an opinion received from that neighbor when updating the node's local view using *Dempster-Shafer* theory.

As each node only exchange recommendations with its neighbors, we believe that the framework would be only capable to locally detect user-domain conflicting behavior attackers. Moreover, this property makes mTrust also vulnerable to position-domain conflicting behavior attack. However, since nodes' behavior is detected based on the ratio of their bad actions, the framework can overcome the on-off attack. Besides, when an attacker is detected, it will infinitively be memorized as such. This latter inhibits an attacker to switch iteratively its behavior, at the expense of depriving it from a second chance to return to good conduct or to adjust a false detection. Finally, mTrust pretends handling the bad mouthing attack using the reference trust, which evaluates the correctness of observations spread by a node. However, the detection of such nodes highly depends to the accuracy of correctness calculation which is not specified.

To summarize, mTrust judges the trustworthiness of a node from different perspectives, each one derived from various sets

of misbehaviors according to their nature. Furthermore, it uses an adaptive trust evolution model by which each dimension of the trustworthiness can be adjusted according to that misbehavior features. The framework minimizes the extra trust-related control overhead by occasionally and locally transmitting recommendations. However, using only local information makes mTrust vulnerable to user-domain and position-domain conflicting behavior attacks. Moreover, mTrust does not provide a uniform view of trust values all over the network which may lead to framework instability for highly mobile networks. Finally, the framework resistance to bad mouthing attacks highly depends on the accuracy of methods used to evaluate the correctness of observations spread by a node.

6) ***Trudi****:* Trudi [34] proposed a distributed trust diffusion framework in which both local information and recommendations are considered.

The *knowledge collection component* stores a list of marks encoding the results of its last interactions with other nodes. If the interaction was a success (the transmitted data was the expected data), the mark is 1, and it is 0 otherwise. The marks included in this list are used to compute the trust mark for future interactions. Each node exchanges its trust table information, including trust marks and trust mark indexes, with another node before initiating an interaction. The *trust mark index* consists in the trust level a node has on its own opinion about another node. The trust information received from another node is referred to as external trust knowledge.

The *trust level computation component* computes a trust mark about other nodes as the weighted average of the personal knowledge (own experience acquired during interactions) and the external knowledge. The personal knowledge is evaluated as the simple average of the results of past interactions with the evaluated node. The external knowledge considers only the average of bad opinions whose recommender trusts in its own opinion (their trust mark index is greater than a threshold) and whose recommendations are considered trusty by the evaluating node (the difference of the recommended trust values and the trust marks in the evaluating node's table is less than a threshold). When a node has few personal knowledge about another node with which it wants to interact, external knowledge are more important in its decision. Consequently, a higher weight is used for external knowledge. A recommender calculates its trust mark index based on the average and standard deviation of its past interactions with the evaluated node. Indeed, if the trust mark is in range of calculated average (plus/minus a predefined threshold), the trust mark index grows proportional to the standard deviation of past interactions. Otherwise, the trust mark index decreases with a greater rate.

The *trust establishment component* manages the admission of interactions with other nodes. It regularly verifies the trust mark of other nodes and decides to establish an interaction only if that node's trust mark is greater than a threshold.

As proved by simulations, Trudi can face detonator on-off attack. Regarding the iterative counterpart, the framework seems also to be resistant as it considers only the bad opinions in trust mark calculation. Besides, the framework considers the difference between a recommendation and the node's own opinion which helps protecting against the bad mouthing attack. However, we believe that the framework can not overcome the conflicting behavior attack since a recommendation different from a node's own opinion is ignored.

To summarize, Trudi allows the network to establish interactions with network elements based on their trust levels. The framework causes additional overhead to network by transmitting multiple lists during each interaction. As the framework exchanges the trust information only with an interacting node, it is not clear to which extent the knowledge uniformity is achieved. This latter is important to evaluate the framework resistance regarding the conflicting behavior attack. Another issue relies on the distribution of the trust mark of all nodes in the trust table, including untrusty recommendations. The point here is if a recommendation is going to be ignored, the framework should use a cognitive process to avoid sending such a recommendation and thus, reducing the network overhead. Moreover, the amount of overhead generated by Trudi highly depends on the used mechanism for recommendation exchange which was not specified by the authors.

7) ***Dynamic Trust Model****:* Dynamic Trust Model [31] proposes a trust model for MANETs to face double-face and bad mouthing attacks.

The *knowledge collection component* uses an Intrusion Detection System (IDS) tool such as watchdog to examine network traffic traveling to and from other nodes. In the event a node does not forward the traffic to the specified next hop in a timely manner, its trust level can be adjusted and reported. Additionally, if the data is modified, neighboring nodes can detect this modification and report it through the threat reporting procedures. Threat information from the IDSs will be communicated using standard data packets, requiring no special control transmissions. Trust reports can be transferred to one of the following network scope:

- In *one-hop report broadcast* scope, the reports are sent to direct radio range, as these nodes can directly monitor their neighbors and thus can make a trust level judgment on the reporting nodes.
- In *k-hop report broadcast* scope, nodes could use limited flooding, restricting the reports to travel a specified distance in hops. Limited flooding might also be accomplished by having each node evaluate the trust level of the reporting node and the nodes that are propagating the report. Only reports from trustworthy nodes would then be further propagated.
- In *selective report distribution*, only nodes recently using the observed node are noticed. This approach eliminates the hop-by-hop trust evaluation and minimizes overhead of trust report traffic.

False reports can be detected by IDSs on neighboring nodes which should be capable to notice large discrepancies in IDS reports and announce the false reports to interested nodes. If a false report is not detected by neighboring IDSs, it will be filtered out by the trust level computation component.

The *trust level computation component* calculates the trust level of a given node based on received reports. The ratio of reporting node's trust level to the required trust level for current message delivery is used as the weight a received recommendation. To age trust reports, a timestamp is transmitted along

with each report and is used as a progressive decreasing factor in calculating trust level. Another factor impacting the utility of a report consists in the distance between the reporter and the current node. Indeed, the distance of non-neighbor nodes retrieved from the routing cache, measured in number of hops, is considered in evaluating any reports from remote nodes. A maximum distance value would also be established as the allowable distance for threat reporting.

The *trust establishment component* decides on each message to be sent based on the data trust requirements. A source node can use the trust level of other nodes to evaluate the security of routes to destination nodes. Using these trust levels as a guide, the source node can then select a route that meets the security requirements of the message to be transmitted. Once a data message transmission is initiated, each node along the route would evaluate the route against its own trust level table. If any intermediate node determines that the trust requirements of the message cannot be met by the next hop in the selected route, an error message would be returned to the originating node. The source would then select a new route.

We believe that the framework can resist the user-domain and position-domain conflicting behavior attacks when a selective report distribution mechanism is used. The use of k-hop report broadcast can make the framework completely resistant to user-domain conflicting behavior attack while the position-domain conflicting threat can be countered only when the attacker moves at most k-hop far from its previous location. Moreover, since only reports from trustworthy nodes are propagated and reporting node's trust level is considered in trust level computation, the framework becomes immune to bad mouthing attack in presence of subversive nodes. It is also claimed that false reports can be detected by IDSs. Moreover, the framework resists to the on-off attack since only the attack reports in evaluating node's trustworthiness is considered.

To conclude, the proposed dynamic trust model makes use of both local and remote information to propose a bad mouthing, on-off and conflicting behavior resistant TMF. However, using an IDS probe installed on each node, which continuously monitors the network traffic, is not energy efficient in the resource-constrained MANETs. Moreover, the framework considers large discrepancies of a trust report as an indication of false recommendation, ignoring such a report and reporting such a reporter as a misbehavior node to other network elements. However, a different trust report may be the result of a conflicting behavior attack. Therefore, the framework is vulnerable in presence of simultaneous bad mouthing and conflicting behavior attackers.

8) ***HTP****:* Human-based Trust Protocol (HTP) [45] builds a trust relationship between nodes based on previous individual experiences and recommendations of others.

As illustrated in figure 6, the *knowledge collection component* consists in a behavior monitor and a recommendation manager. The *behavior monitor* collects knowledge on neighbors' behavior, while the *recommendation manager* is responsible for receiving, sending, and storing recommendations. The behavior monitor also indicates the presence of new neighbors to the recommendation manager. The trust level stored in the trust table combines individual experienced trust and the one

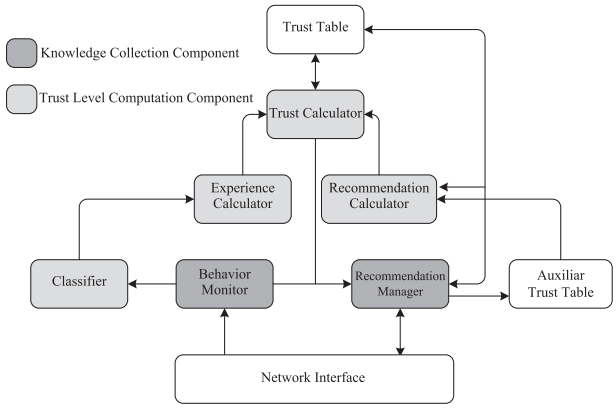

Fig. 6. Human-based Trust Protocol.

collected by recommendation manager. A Recommendation Exchange Protocol (REP) is proposed as part of the recommendation manager. According to REP, recommendations are one-hop broadcast packets which can be obtained by sending a Trust Request (TREQ) or by receiving a Trust Advertisement. The TREQ is initiated when nodes first meets. Before sending a TREQ message, a node waits for a specific period of time, trying to gather the maximum number of new neighbors. The node receiving the TREQ sends the requested trust value in the Trust Reply (TREP) message only if it has the target node as a neighbor. In order to optimize the generated traffic, the TREP messages are transmitted after a random period of time waiting for receiving other TREQs. The trust advertisement messages, which are unsolicited recommendations, are only sent when the trust level about a particular neighbor varies more than a certain threshold value. The reception of a recommendation involves two actions. First, the recommendation is stored in the *auxiliary trust table* and then, it is forwarded to the recommendation calculator module (which takes part of the trust level computation component). The Auxiliary Trust Table contains also the variance and the age (also referred to as the relationship maturity) of each trust level.

The *trust level computation component* consists in classifier, experience calculator, recommendation calculator and trust calculator modules. The *classifier* reasons about the information collected by the monitor, deciding the quality of an action according to a previously defined classification. This module then sends its verdict to the *experience calculator* which estimates the individual experience for a given node based on the information received by the classifier.

The *recommendation calculator* determines the recommended trust value of a neighbor based on the opinions of other nodes, referred to as the *neighbor recommendation*. The neighbor recommendation is a weighted average of trusty neighboring nodes' recommendation, i.e. neighbors whose trust level is above a certain threshold. The weight for a recommendation is proportional to the trust level of the recommender, the maturity of the relationship between the recommender and the node being evaluated, and the accuracy of a recommended trust level (based on standard deviation). The individual experiences (calculated by experience calculator) and the neighbors recommendation (calculated by recommendation calculator) are fed

to the *trust calculator module* which evaluates the final trust level. This latter consists in the weighted sum of the neighbor recommendation trust and the *partial trust*. The partial trust is computed using the weighted sum of the individual experiences and the last trust value stored in the Trust Table.

Malicious nodes can implement an attack exploiting the concept of relationship maturity by attributing fake trust levels. In order to minimize this effect, each node defines a maximum relationship maturity value which represents an upper bound for the relationship maturity. This value is based on the average maturity relationship value of its most trusted neighbors. The trust calculator also notifies the recommendation manager, the need of sending a trust recommendation advertisement.

In addition, the authors define three operation modes to cope with the heterogeneity that characterizes ad hoc networks: simple, intermediate, and advanced. Nodes with low power/storage capacity operate in the simple mode, in which they use just the main trust table and the REP protocol is optional. Nodes with a medium capacity operate in the intermediate mode, which also keep the recommendations of other nodes, but skip the variance of the trust level recommended by a recommender and, hence, the auxiliary trust table is not used. In the advanced mode, nodes implement the whole trust system with all features.

As each node only evaluates its neighbor's trustworthiness and exchanges recommendations just locally, the user-domain conflicting behavior attackers would only be locally detected. Indeed, if a node becomes neighbor of an attacker, the malicious node detection occurs only when the node gathers enough information on the misbehavior. In addition, the non-uniformity of trust values results in the HTP vulnerability to the position-domain conflicting behavior attack.

To conclude, HTP uses both direct and indirect information with a low energy consumption, memory space and processing for trust level calculation. The REP only considers interactions with neighbors, which makes the protocol scaling well for large networks. However, the amount of generated traffic, consisting of both TREQ messages and their corresponding replies, doubles the neighborhood overhead. Although the recommendation procedure is particularly helpful for enriching the trust monitoring, the detection of conflicting behavior nodes can be time-consuming due to the non-uniformity of trust values all over the network. In addition, the same scenario can happen in highly mobile networks where nodes may move unconsciously to another position close to an attacker. In HTP, both good and bad actions performed by a neighbor are exerted similarly in calculating trust values which results in vulnerability to on-off attack.

9) ***AOTDV:*** Ad hoc On-demand Trusted-path Distance Vector (AOTDV) [30] is a trust-based extension of AOMDV [49] (a multi-path extension of AODV) which distributes trust information through RREQ packets in order to resist bad mouthing and double-face attacks.

The *knowledge collection component* within each node is equipped with promiscuous mode to collect local information. Remote information can be obtained through RREQ packets through two supplementary fields: required trust and actual trust. Required trust consists in the trust value required by data packets set by the source. Actual trust is the progressive minimum trust value of nodes that the RREQ has passed by during the route discovery process. AOTDV's routing table adds a supplementary field to the classical AODV's routing table which stores the paths' actual trust per destination registered in the table.

In the *trust level computation component*, each node derives trust value of its neighbors from forwarding ratio of data and control packets. During trust computation, a linear aggregate method is used to combine the forwarding ratio of data and control packets into an overall value of trust.

The *trust establishment component* interferes with trust-based route discovery and route selection. Among all discovered routes, the route with minimum hop count satisfying the required trust will be selected. Moreover, each node has its own threshold for trustworthiness levels. If a node is added into the black list, i.e. is evaluated as untrustworthy by all its neighbors, it is not allowed to send packets and all its RREQ and RREP packets are discarded. However, the framework does not specify how the opinions of all neighbors of a node are collected to be able to construct the black list.

We believe that AOTDV can resist iterative on-off attack as the black list records the attackers history infinitively. Moreover, the black list mechanism is not sensible to bad mouthing attack as a node is not registered into the black list unless all of its neighbors confirm its misbehavior. However, AOTDV did not define a mechanism to handle the bad mouthing attack targeting the path trust calculation. Indeed, such an attacker may report a high/low trust value for a bad/good node through the path used by the RREQ packets in order to falsely increase/decrease the trust of a particular path.

Moreover, since AOTDV's nodes use promiscuous mode and only calculate the trustworthiness level of one-hop neighbors, the framework is resistant to user-domain conflicting behavior attack. However, the framework is vulnerable to the position-domain conflicting behavior attack as it lacks a trust propagation mechanism and, hence, does not provide the trustworthiness uniformity throughout the network.

To conclude, AOTDV is a trust-based extension of AOMDV which makes the network resistant to the iterative on-off, user-domain conflicting behavior and bad mouthing attacks. A notable positive point of AOTDV is that it considers the forwarding ratio of both control and data packets, each one computed separately. This latter allows to handle the attack targeting both routing and data forwarding processes. However, AOTDV presents some important issues. Mainly, although the use of multi-path AOMDV protocol provides the framework with load-balancing features, it renders the framework protocol-dependent and limits its widely use in MANET environment. Moreover, the framework does not consider a second chance mechanism to encourage an attacker node to participate in network operation after their involvement in the black list. Another issue consists in the neighborhood opinion distribution to compose the black list, which is not specified in the paper. The latter mechanism should enable nodes to timely possess the list of all misbehaving nodes while a minimum amount of overhead is generated to the network.

*10) **CORE**:* COllaborative REputation (CORE) [33] is a DSR-based TMF in which community members with good trust level can use the common network resources while members with bad trust level are gradually excluded from the community. The presented framework is potentially resistant to bad mouthing and double-face attacks.

The *knowledge collection component* uses both local and remote information. The framework defines subjective reputation, indirect reputation and functional reputation. The *subjective reputation* is calculated by a subject node (observer) on the behavior of a neighbor based on their direct interactions using the watchdog mechanism. The *indirect reputation* is provided by other members of the community. The *functional trust* of a node consists in the aggregation of subjective and indirect reputations with respect to different network functions. Each node is enriched with a *Functional Reputation Table (FRT)* per network function including four entries for all network entities: the unique identifier (node ID), recent subjective observations of its behavior, a list of the recent indirect reputation values and the value of the reputation evaluated for that predefined function. As remote information, the list of nodes on the selected path carried by the reply message is considered as correctly behaved. Misbehaved nodes are detected in the request phase by watchdog mechanism, while the reply phase informs the initiator and the intermediate nodes participated in the communication.

As *trust level computation component*, a global reputation table is used to combine the different functional trust values existing in different FRTs. For subjective reputation calculation, more relevance is given to past observations to avoid false detections due to link breakage. For indirect information, more relevance is given to the information collected from trusty entities. Moreover, this component decreases the reputation of an idle entity along time, encouraging nodes to cooperate.

Upon receiving a request, the *trust establishment component* of the node checks the reputation value evaluated for the requester in its global reputation table. If its reputation value is negative, the entity will not execute the request.

Theoretically, we believe that CORE is resistant to bad mouthing attack since subjective trust is calculated by the subject and not based on recommendations. In addition, CORE can exclude local user-domain conflicting behavior attackers using the watchdog mechanism. However, the remote user-domain and position-domain conflicting behavior attacks can not be handled since the trust information is propagated only in reply messages through the path and, hence, is not uniformly distributed all over the network. Besides, while CORE is resistant to detonator on-off attack, it is vulnerable to iterative counterpart, lacking a mechanism for maintaining attackers misbehaving history.

To conclude, CORE is a promising TMF in which a minimum extra overhead is used to make the network resistant to bad mouthing and local conflicting behavior attacks. However, CORE can not handle the generalized conflicting behavior and iterative on-off attacks. CORE is protocol-dependent, especially the path trust is highly related to the list of nodes carried by the RREP messages within the DSR protocol. In addition, some framework details are remained unspecified. Mainly, the framework does not determine how the subjective trust of nodes on the path of a RREP packet is rated. Finally, the framework is not implemented, making difficult to confirm its well-performance in real network conditions.

*11) **Lindsay**:* Lindsay [43] presents a structure for calculating trust, modeling trust propagation, and defending trust evaluation systems against trust-distortion attacks.

The *knowledge collection component* builds both direct and remote trust values. The source requires the trustworthiness level of all nodes on the existing paths to select the more trusted one. If some nodes' trustworthiness level does not exist in the trust table, the source sends a *Trust Recommendation Request (TRR)* to its neighbors. The TRR message should contain the IDs of nodes that the source needs their trust value and IDs of neighbors that the source knows them trustworthy. If these nodes have no sufficient information, they rebroadcast the TRR to their trustworthy neighbors. In order to reduce overhead, the TRR message also contains the maximal length of trust transit chains and the time-to-live (TTL) during which the source waits for replies. The trust values obtained by TRR messages are referred to as *recommendation trust*. Furthermore, during the data transmission from a source to its destination, a lightweight self-evaluation mechanism is used to allow the source node to collect packet forwarding statistics, forming its direct trust experience, called *forwarding trust*.

The *trust level computation component* calculates the forwarding trust based on the ratio of packets forwarded by a node to packets the source asked that node to forward. During the data transmission phase, the forwarding trust process is executed which may result in changing the recommendation trust of some previous recommenders. Indeed, if the difference of a node trust received previously by the TRR reply and that node's forwarding trust (own experience) is more than a certain threshold, the recommender's trust will be decreased. Moreover, the authors use the beta distribution of a node's (un)successful interactions to detect malicious nodes. The outcome of this process will be the expected probability of goodness of a given evaluated node, which will then be used for calculating an adaptive forgetting factor to weight its bad behaviors more than normal ones.

The *trust establishment component* within the source selects the more trusty path to transmit data. Besides, TRR messages from malicious nodes will not be replied.

We believe that Lindsay can resist detonator and iterative on-off attacks, using the adaptive forgetting factor to differently weight good and bad behaviors. Although the recommender trust update can help avoiding the bad mouthing threat, this property makes Lindsay vulnerable to conflicting behavior attacks. Indeed, even if nodes use others' recommendations, they consider the gap between the recommender's sent trust and forwarding trust as a negative point for the recommender. However, a double-face attacker may have behaved differently with that recommender and the current source. Therefore, that gap was an indication of a conflicting behavior rather than the recommender's bad mouthing.

To conclude, Lindsay uses the data transmission process to update its trust table. This framework is immune to on-off and bad mouthing attacks while it is vulnerable to conflicting behavior attacks. The proposed knowledge collection component

presents some issues. First, it generates a lot of overhead in initial network operation, when a lot of TRR messages and their corresponding TRR replies are sent to neighbors and then iteratively rebroadcasted to the new neighborhood. Second, the authors did not determine the operation of the so-called light-weight self-evaluation mechanism, which should manage the observation of far nodes' behavior. Consequently, the total amount of imposed extra overhead remains unclear.

12) ***SORI****:* Secure and Objective Reputation-based Incentive (SORI) [20] is implemented to encourage packet forwarding and discipline selfish behaviors based on local and one-hop neighbors' information. The framework can resist double-face and bad mouthing attacks in some extent.

The *knowledge collection component* collects the packet-forwarding behavior of neighbors by overhearing. In addition, each node periodically broadcasts its list of neighbors' trust to one-hop neighbors. The *trust level computation component* within each node keeps *Request-for-Forwarding (RF)* and *Has-Forwarded (HF)* parameters for each of its neighbors. Given the RF and HF, the node can create a *Local Evaluation Record (LER)* for that neighbor which consists of trust level and confidence entries. The the *trust level* is calculated as the ratio of RF to HF. The *confidence* metric, describing how confident the node is for its judgment on the reputation of its neighbor, is equivalent to the value of RF. Indeed, the more packets transmitted to the neighbor for forwarding, the better estimation can be obtained about how well the neighbor forwards. Each node periodically updates its LER for each neighbor according to the changes of RF and HF. The updated record is broadcasted to the node's neighborhood if a neighbor's trust level has been significantly changed. The new trust level of a particular neighbor is then calculated as the weighted average of the evaluator's trust level and received recommendations about that neighbor. The confidence of a recommendation and the credibility of the recommender (its trust level in the evaluator node's table) are used as the weights of a received recommendation.

The SORI's *trust establishment component* punishes a malicious neighbor having a trust level lower than a preset threshold, by probabilistically dropping the packets originated from that misbehaving neighbor. The probability of dropping i's packets is $1 - trust_i$ with a preset margin. The margin is designed to help well-behaving nodes having a marginal trust level to be treated a little bit more generously.

We believe that SORI can detect local user-domain conflicting behavior attackers using the trust propagation. However, as this propagation is local, nodes are vulnerable to position-domain and generalized user-domain conflicting behavior attacks. SORI is resistant to bad mouthing attack in presence of subversive nodes because the trust of a recommender is considered as a weight for the recommendee's trust. Furthermore, SORI can handle detonator on-off attack, reducing the trust level of a node with continuous bad actions. However, we believe that the framework is fragile regarding the iterative on-off attack as the proportion of bad behaviors can be adjusted by the attacker to remain undetected. To solve this shortage, the framework needs to maintain misbehaving nodes' list or weight misbehaviors more than normal behaviors in calculating nodes' trustworthiness level.

To conclude, SORI uses only local and one-hop information to handle some simple case of conflicting behavior and bad mouthing attacks. Due to the non-uniformity of trustworthiness values throughout the network, SORI is neither immune to more generalized conflicting behavior attacks, nor stable in high mobile networks. Another issue relates to the considered confidence parameter (defined equivalent to the Request-for-Forwarding parameter), which may inaccurately be a low value for high confident boundary nodes. Finally, SORI requires a lighter monitoring mechanism in high loaded conditions.

13) ***TEAM****:* Trust Enhanced security Architecture for MANET (TEAM) [5] is composed of a set of interactive components, namely Overlay Trust Model (SMRTI), key management mechanism, basic routing protocol, secure routing protocol and cooperation model (Fellowship). SMRTI captures the evidence of trustworthiness for other nodes from the security models, and in return assists them to make better security decisions. The fellowship model defends against both flooding and packet drop attacks.

The *knowledge collection component* collects the direct trust based on passive monitoring. Alternatively, recommendations are derived using the route contained in the packet.

The *trust level computation component* computes trustworthiness of a node, of a packet and of a route. The *node trust* is measured as the weighted sum of direct and recommended trust using a greater weight for direct trust. The direct and recommended trust are calculated based on the corresponding event rate, which is proportional to the type of event. More explicitly, direct (or recommended) trust is computed using the previous computed direct (or recommended) trust plus/minus the new observed (or received recommendation) event rate. The recommended trust is weighted by the recommender's trust. The *packet trust* is computed from the trust level of both the packet's source and destination. Finally, the *route trust* consists in a weighted sum of the trustworthiness held for all the nodes on the path, excluding the evaluating node.

The *trust establishment component* plays role during the route discovery process as well as packet forwarding. Whenever a source attempts to send a data packet to another node, it evaluates the route trust of all available route(s) by SMRTI component. The route with highest level of trustworthiness is then chosen and passed back to DSR. DSR then requests the secure routing protocol component to incorporate necessary security services through the cryptographic suite and shared secrets managed by the key management mechanism. Alternatively, if all the routes are untrustworthy, or if there is no available route to destination, then SMRTI responds back to DSR with a decision to initiate a new route discovery cycle.

Furthermore, at every intermediate node, the fellowship model component permanently checks whether the transmission rate of previous-hop indicates a flooding attack. If so, the packet is discarded and the evidence for malicious behavior is sent to SMRTI. Otherwise, it requests SMRTI to evaluate the trustworthiness for the previous-hop. The fellowship model then passes the packet to secure routing protocol component to incorporate necessary security services. If the later fails to authenticate the intermediate nodes listed in the route or is unable to verify the discovered path's integrity,

it then forwards the evidence for fabrication or modification to SMRTI. Otherwise, the secure routing protocol component requests SMRTI to evaluate trustworthiness for the packet.

In addition, SMRTI evaluates trustworthiness for the route contained in the packet, and inserts the route into its DSR's route cache only if the route is trustworthy. Once the secure routing protocol confirms that the packet is trustworthy, it applies the required security services to the packet and then passes the packet back to the fellowship model. Now, the fellowship model requests SMRTI to evaluate trustworthiness for the next-hop. For a RREQ packet, the trust evaluation for next-hop is skipped as the next hop is unknown due to the broadcast nature of route request. Nevertheless, if the next-hop drops the packet, the fellowship model investigates its transmission-rate and contention for the transmission channel before forwarding the evidence for packet drop to SMRTI. However, if the next-hop forwards the packet without performing any malicious behavior, then the fellowship model forwards the evidence for benign behavior to SMRTI.

We believe that TEAM can protect against conflicting behavior attack, using both direct and indirect information. While TEAM is capable to detect the detonator on-off attack, it should consider a higher decrement factor for malicious behavior to become immune to iterative on-off attack. Furthermore, since recommender's trust level is used as the weight of recommendations, TEAM can defend against bad mouthing attackers in presence of subversive nodes. For a general defense against bad mouthing attack, the framework should apply a trust value indicating the accuracy of any recommendation received.

To summarize, TEAM is a holistic architecture considering all required components for trust management. The generated overhead related to the knowledge collection is discounted by the use of piggybacking technic. However, the continuous piggybacking may aggravate bottleneck in a loaded scenario, requiring a cognitive approach to manage such a situation. In addition, the reactive trustworthiness evaluation accomplished per each data packet in intermediary nodes may significantly delay the packet forwarding process, especially for long paths.

14) ***DRDM****:* Dishonest Recommendation Detection Model (DRDM) [50] is a specialized solution to detect dishonest trust recommendations in mobile ad hoc networks. The proposed framework is an extension of the work proposed in [22].

The *knowledge collection component* gathers both direct trust and recommendations. However, the paper does not specify any local or recommendation monitoring mechanism.

The *trust level computation component* uses two basis in order to detect dishonest recommendations. First, recommendations shared by low trustworthy nodes are considered as dishonest. Second, a recommendation highly deviated from the mean trust value is considered as dishonest despite the trustworthy nature of its recommender. DRDM identifies dishonest recommendations in three steps. First, all *Recommendation Trust Values (RTVs)* received about a given node are stored in a RTVSet list. For each RTV in RTVSet, a dissimilarity function is calculated, dividing the square of the deviation from median of RTV, by the trust value of the recommender. The square is used to elevate the impact of deviation. The division allows to detect the trust shared by low trustworthy nodes and highly deviated trust shared by trustworthy nodes. In the second step, the RTVSet is sorted with respect to dissimilarity values, giving a list in which recommendations with highest dissimilarity value are in the top and considered as suspicious recommendations. The sorted RTVSet is then passed through a smoothing function which indicates how much the dissimilarity can be reduced by removing the suspicious recommendations from the RTVSet. In the last step, a subset of sorted RTVSet for which the smoothing factor is high with least number of recommendations is considered as the set of dishonest recommendations.

As the paper only focuses on detection of dishonest recommendations without describing the mechanism for collecting trust knowledge or computing trust values, the resistance of the proposed scheme against on-off and conflicting behavior attacks can not be judged.

To conclude, DRDM proposes a mechanism to detect dishonest trust recommendations in mobile ad hoc networks. The main asset of DRDM consists in its independence from underlying trust model and routing protocol. However, the presented dissimilarity and smoothing functions are designed without taking the multi-attacks environments into consideration. For instance, a high deviation from the median is not an appropriate criteria for considering a recommendation as suspicious since such a different recommendation may be generated by a unique victim node of a conflicting behavior attacker. In addition, the resistance of DRDM to double-face conducts depends on the employed trust model and monitoring mechanism.

In this section, we described existing trust management frameworks and analysed their performance regarding different trust-distortion attacks. This latter resulted in the identification of technics allowing a trust model to overcome several trust-distortion threats simultaneously. In next section, we first identify the main metrics to evaluate the performance of trust management frameworks. The identified metrics are then used as a basis to compare different trust management frameworks in section VI. This comparison gives some important guidelines to converge to an ideal model which takes advantage of existing frameworks while avoiding their shortcomings.

V. Evaluation Metrics for Trust Management Frameworks

A relevant step for progressing the field of trust management is to be able to evaluate and compare the performances of current solutions. This section aims to provide a holistic view of different metrics for evaluating the trust models' efficiency. First, we identify different metrics considered in the literature and propose some new metrics. Second, we propose a classification of identified metrics, regrouping them regarding their perspectives. Some metrics are merged in order to present a consistent view of the state-of-the-art in the field. This is done when several representations are used for two or more metrics referring to a same ability or functionality of the system. In addition, for a same metric employed under different nominations in the literature, we used either the more representative nomination or the one more commonly used by the research community.

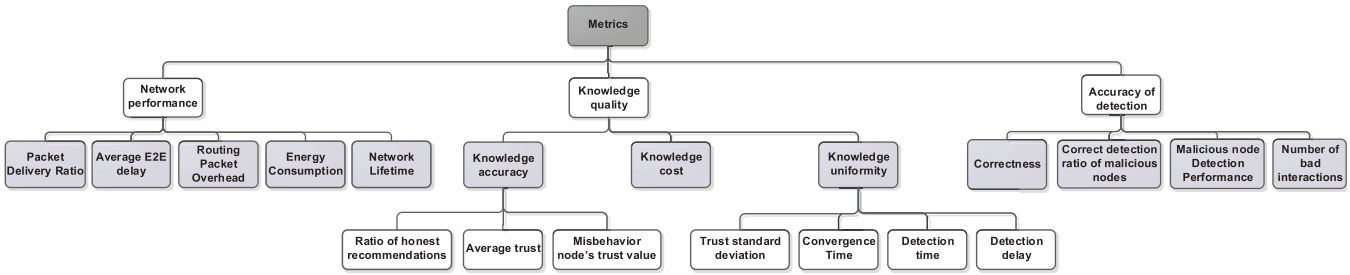

Fig. 7. Metrics classification.

It is worth mentioning that the identified metrics are only useful for comparing TMFs developed with same objectives and assumptions, and under same topology, routing protocol and network configurations. As an example, the efficiency of a TMF which develops only the knowledge collection and trust level computation components can not be compared with another framework applying the estimated trust values to enhance the routing and data forwarding processes. Obviously, the interrelated impact of the trust model on the network performance can be reflected better in the latter case.

As depicted in figure 7, the identified metrics can be classified into three perspectives, namely, the *network performance* metrics, the *knowledge quality* metrics and the *accuracy of detection* metrics.

The **network performance metrics** measure how the trust management framework preserves the network performance in presence of different attacks' kind and intensity. The main employed network performance metrics include:

- Packet Delivery Ratio (PDR) or throughput - number of data packets successfully delivered at the destination divided by the number of data packets sent from the source.
- Average E2E delay (AE2ED) - transmission delay of data packets delivered successfully. The delay consists of propagation delays, queuing delays at interfaces, retransmissions delays at the MAC layer, as well as buffering delays during the route discovery.
- Control Packet Overhead - the ratio of the number of control packets (including route request/reply/error packets) to the number of data packets transmitted in the network.
- Energy Consumption - the total amount of energy consumed by nodes. It includes the energy spent to forward data and control packets as well as the energy consumed to establish trust in the network.
- Network Lifetime - the time when the first node is discharged and hence disconnected from the network. This metric can reflect the amount of overall gain or loss in energy consumption resulted from the trust model.

The **knowledge quality metrics** evaluate the cost and quality of knowledge provided by the framework. The knowledge quality can be evaluated from three perspectives: the *knowledge accuracy*, the *knowledge uniformity* and the *knowledge cost*.

The *knowledge accuracy* metrics measure the accuracy of nodes estimation of network elements' trustworthiness. The main knowledge accuracy related metrics are described in the following:

- Average trust - average trustworthiness of a node estimated by all other nodes in the network. This metric shows the trend of the trust estimated for a normal or a misbehavior node. Average trust should converge to a high reference value (ideally 100%) for a normal node and to a low reference value (ideally 0%) for an attacker.
- Misbehavior node's trust value - ratio of the average trust value obtained for a misbehavior node and its real nature. This metric aims to measure the accuracy of nodes estimation on an attacker node trustworthiness compared to the intensity of the performed attack.
- Ratio of honest recommendations - number of accurate recommendations vs. the total number of recommendations received from other nodes.

The *knowledge uniformity* metrics evaluate the uniformity of nodes view on other nodes trustworthiness and the delay to converge to such a uniform view. This uniformity is important to enable distributed mechanisms to exclude the attacker node from the network. The main identified knowledge uniformity metrics consist in:

- Trust standard deviation - the standard deviation of average trust of nodes estimated by all other nodes in the network. This metric shows the uniformity of the trust knowledge estimation among all target nodes in the network. Ideally, the standard deviation should converge to 0, describing an uniform trust estimation on all nodes through the network.
- Convergence time - time taken to form a unique estimation on nodes trustworthiness all over the network.
- Detection time - number of interactions needed for the honest nodes to identify the dishonest ones. This metric is mainly helpful for trust management frameworks in which the knowledge collection performance depends on the traffic transferred in the network.
- Detection delay - time needed for the honest nodes to identify the dishonest ones.

The *knowledge cost* metric evaluates how much the provided knowledge costs for the network. This metric consists in the extra overhead caused by the trust management framework's extensions relative to the regular routing overhead. In some works, the knowledge cost is evaluated regarding the amount of traffic generated by the trust management framework vs. the total network traffic which can be interchangeably used instead of the previous interpretation. Along with recent green trend which aims at diminishing the environmental pollution of Information and Communication Technology (ICT) sector [8],

the energy consumption of trust management framework vs. the total energy consumption of the network can be considered as another important aspect of knowledge cost.

The **accuracy of detection** metrics measure the accuracy of trust establishment decisions taken based on the nodes' trustworthiness estimations. The following metrics of this category have been identified in the literature:

- Correctness - the number of nodes having a correct perception on other nodes trustworthiness. This metric allows comparing the accuracy of global trust knowledge. Ideally, the correctness metric should converge to 1, showing a correct view about other nodes' trustworthiness through the network.
- Correct detection ratio of malicious nodes - the ratio of correctly detected malicious nodes to the real number of them.
- Malicious node Detection Performance (MDP) - the ratio of honest nodes succeeding to identify a malicious node correctly.
- Number of bad interactions - the total number of bad interactions that the dishonest nodes succeeded to do.

It is worth mentioning that the efficiency of a TMF should be evaluated using various network configurations and attack scenarios. Indeed, a trust model may perform differently based on the mobility, traffic load and network density factors. In addition, the performance of a TMF should be evaluated under different number of attackers, intensity of attack, attack types and positioning of attackers. For instance, a framework may be immune to a particular type of attack, but not when that attacker targets a node in the boundary of the network with less number of neighbors and hence less evidence of other nodes' nature. Based on the attack type, a set of parameters should be varied to evaluate the efficiency of the framework regarding different scenarios of that attack. For instance, when the framework is evaluated with respect to iterative on-off attack, evaluations should consider various durations for on and off intervals. Also, using randomized on and off intervals within a same experiment allows to evaluate the framework in face of indeterministic nature of an attack arrival.

In addition, each trust-distortion attack type may perform different misbehavior action on packets (e.g. packet dropping, packet modification, packet delaying). Based on the misbehavior action performed, some specialized network performance evaluation metrics may become important. Mainly, when the packet dropping misbehavior is taken place, the number of dropped packets can give more evidence on the efficiency of the TMF to detect maliciousness.

VI. Discussion

In this section, we discuss the strengths and weaknesses of employed technics to handle trust-distortion attacks, their open issues and the future directions towards a convergent trust model. We identify also some future major trust-distortion attacks that were not addressed in the literature.

A. Comparison of Existing Frameworks

In order to progress to a convergent trust model, there is a need to discuss how different proposals compare, which limitations of one proposal are addressed by the others and what is the best proposal for a given case. The comparison can be done either by comparing trust management frameworks regarding the results of quantitative measurements or with regard to their qualitative properties and to the approaches taken for each of their components. Due to the inaccessibility of implementation of existing models, we use some of the identified metrics to compare qualitatively among trust proposals. Table I summarizes the characteristics of existing trust management frameworks and evaluates them qualitatively regarding some key characteristics and evaluation metrics. Based on the table, we conclude hereafter the key properties required by a trust management framework to cope with each kind of trust-distortion attacks.

Mainly, the following points can be identified from the table for **double-face conducts**:

- The *user-domain and position-domain conflicting behavior attack*, in which an attacker chooses discriminatory its victim user(s)/network region, can be generally detected by a recommendation-based framework. Indeed, recommendations allow transferring the low trustworthiness level of an attacker calculated by the victim (region) node(s) to non-victim nodes, making them aware of the malicious nature of the attacker.
- The *detonator on-off attack*, in which an attacker starts a misbehavior after a certain time of normal conduct, consists in a simplest attack to detect. Indeed, as the attack lasts, the attacker trust value will be decreased after a while. The trust management framework should specially care of the reduction of detection time, avoiding the attack potential pitfalls for the system before the attacker being excluded from the network.
- The *iterative on-off attack* tries to hide the attack by alternatively switching between good and bad conducts. The more commonly employed technic to handle this attack consists in an adaptive forgetting factor in which an already detected attacker needs to perform many good actions to increase its trust level.

For **bad mouthing attacks**, the identified results are presented hereafter for each type of attacker node:

- A *bad mouthing attack in presence of subversive nodes* is almost easy to defend as the subversive behavior of the attacker (even if we do not consider its bad mouthing actions) causes its trustworthiness level to be reduced. To defend against this type of attack, recommendations can be weighted by the trust value of the recommender in order to decrease the effect of a false opinion while the recommendations from trusty nodes are considered with high confidence [25], [33].
- A *bad mouthing attack in presence of Liar nodes* is more complex to be detected compared with subversive bad mouthing. Indeed, the normal cooperation of a liar attacker in other network operations makes its trustworthiness level dependent to the detection of its false recommendations. However, it is difficult to detect the accuracy of an opinion received from a recommender, especially for far recommendee nodes. Consequently, it is not appropriate to use nodes' trust value as the weight of

TABLE I
COMPARISON OF TRUST MANAGEMENT FRAMEWORKS

	Knowledge collection component	Trust level computation component	Trust establishment component	attacks immunity	Overhead	Knowledge uniformity	Knowledge dissemination real-timeness	Energy efficiency	Tested parameters
Bella [7]	Local and remote	forwarded traffic/generated traffic	No	Detonator, conflicting behavior	Medium	Medium	Medium	High	Trust level of a source, a transit & an ugly node over time
ATMS [35]	Local and remote	same as Bella	No	Detonator, conflicting behavior	Low	Medium	Medium	High	PDR, AE2ED, Correctness, Average trust, Trust standard deviation
OTMF [25]	Local and remote	Watchdog (local), Bayesian approach (remote)	No	Bad mouthing with subversive attackers, Detonator, conflicting behavior	High	High	High	Low	Average trust
Almotiri [4]	Local and remote	(un)Successful connection with a neighbor (local), Trust values added in routing pkts (remote)	Choosing the path with maximum average of path nodes' trust value	On-of, user-domain conflicting behavior	Medium	Low	Low	Medium	PDR, AE2ED, overhead
Data Class [39]	Local and remote	forwarded/discarded pkts (local), transmitted through communication session (remote)	A minimal trust level required for acceptance of a forwarding request. Select the most trustworthy path	Detonator, user-domain conflicting behavior	High	Low	Low	Medium	frequency of using personal & general trust data in decision making
LARS [21]	Local	Increase or decrease trust value based on its participation	No	Bad mouthing, on-off	Medium	Low	Low	Medium	PDR, Overhead, AE2ED
CONFIDANT [9]	Local and remote	Assigning different weights to different type of behavior	Path re-ranking. Deletion of paths containing malicious nodes. Ignoring a RREQ from a malicious node or a packet with route containing an attacker in the source route	Bad mouthing, on-off, conflicting behavior	Low	Low	Low	Medium	PDR, Overhead, Utility
AFStrust [48]	Local and remote	AHP, Fuzzy Logic & black-lists nodes based on a trust threshold	black-list nodes will be excluded from the local network for a special time	Bad mouthing with subversive attackers, detonator, user-domain conflicting behavior	Low	Low	Low	Low	malicious node's average trust, Satisfaction rate of network interactions, Number of good recommendations, Correct detection ratio of malicious nodes
mTrust [28]	Local and remote	Dempster-shafer theory of evidence	No	Bad mouthing, on-off, user-domain conflicting behavior(locally)	Low	Low	Low	Medium	Correct attacker detection ratio, Overhead, Convergence time

TABLE I
(*Continued.*)

Trudi [34]	Local and remote	Based on interaction results & accuracy indexes	No	Bad mouthing, detonator	High	Medium	Medium	Medium	MDP, Number of bad interactions, Detection time
Dynamic [31]	Local and remote	Weighted average of trust reports & own observations	A route that meets the security requirements of the packet is selected	Bad mouthing with subversive attackers, on-off, conflicting behavior	Medium	Medium	Medium	Low	Not implemented
TEAM [5]	Local and remote	weighted sum of evidence captured during one-to-one interactions & recommendations	Handles pkts (route/forward) based on the source, destination, next and previous hop's trustworthiness	Bad mouthing with subversive attackers, detonator, conflicting behavior	Low	Medium	Medium	Low	PDR
OCEAN [6]	Local	Rating of positive & negative events. An avoid list maintains malicious IDs	Route selection dropping RREQ & RREPs containing one of avoid list nodes	Detonator, user-domain conflicting behavior	Medium	Low	Low	Medium	Throughput
HTP [45]	Local and remote	Maturity-based	No	Bad mouthing with subversive attackers, detonator, user-domain conflicting behavior(locally)	Medium	Low	Low	Medium	Neighbor's Convergence Time, Detection delay of slanderer nodes
AOTDV [30]	Local and remote	Correctly forwarded/Requested to forward (control/data) pkts in a time interval (local). Lowest trust value in the path (path trust). A node added to a black list if evaluated as untrustworthy by all of its neighbors	A secure route will be selected. Ignoring packet sent, replied or any request initiating by the black list' nodes	Bad mouthing, on-off, user-domain conflicting behavior	Medium	Low	Low	Medium	PDR, AE2ED, overhead, Path Optimality
CORE [33]	Local and remote	Local: Watchdog, Remote: Nodes carried by RREP considered as trustworthy	Not executing the request of nodes with negative trust value	Bad mouthing, detonator, user-domain conflicting behavior	Low	Low	Low	Medium	Not implemented
Lindsay [43]	Local and remote	Forwarded/asked to forward pkts (local), TRR messages compared to behavior observation (remote)	Selecting the most trustworthy path. Malicious node detection	Bad mouthing, on-off	High	Medium	Medium	Medium	PDR, MDP
TAODV [29]	Local and remote	Based on trust elements (belief, disbelief & uncertainty) of a node obtained by combining evaluator's opinion with its neighbors opinions	Using cryptography when uncertainty high or belief low. Continuing routing when belief is high. Refusing routing for an expired time when unbelief high	Detonator, conflicting behavior	High	Medium	Low	Low	Not implemented
SORI [20]	Local and neighbor	Requested to forward/Forwarded pkts (local), Weighted average of a node's local trust & its neighbors' trust (remote)	Probabilistic dropping of malicious node's pkts	Bad mouthing with subversive attackers, detonator, user-domain conflicting behavior	High	Low	Low	Medium	Throughput
DRDM [50]	Local and remote	Based on dissimilarity and smoothing functions for detection of dishonest recommendations	No	Bad mouthing	N/A	N/A	N/A	N/A	Accuracy of detection of dishonest recommendations

its recommendations. For such situations, the trust model should maintain different types of trust information for a node: the action trust and the recommendation trust. The action trust is the trust level of a node estimated based on its last actions, while the recommendation trust consists in the confidence a node has on recommendations reported from a recommender. Only the entities which have provided good recommendations previously can earn a high recommendation trust. This latter is estimated through a deviation test on recommendations received from a given recommender on a given recommendee compared to the trust level of that recommendee retrieved from the trust table of the node. Recommendations received from a node with low recommendation trust will have minor influence on calculated overall trust [43].

B. Open Issues and Future Research Directions

In this section, we present the open issues of existing trust models and propose some new techniques to optimize their attack resistance. Moreover, we identify a number of more complicated trust-distortion attacks and express some guidelines to overcome these treats.

The efficiency of both local and recommendation-based TMFs depend before all on the cost and accuracy of information observed on the behavior of a neighbor node. The commonly employed watchdog mechanism, which rates the percentage of packets forwarded by a next hop, suffers from a number of open issues. In the following, we describe some existing deficiencies, possible solutions and future research directions in this regard:

- First, the watchdog mechanism requires a mean to differentiate between maliciously dropped packets and those not forwarded due to congestion or wireless channel collision or forwarded late due to queuing delays. This issue can lead to an inaccurate low trust level calculated for nodes located in a congested or high traffic area. Moreover, simple radio jammer attackers may be encouraged to profit from this vulnerability to interfere with current communications, reducing the trust level estimated on honest neighbors.
- The watchdog mechanism, as it is defined, can not detect the user-domain conflicting behavior performed against a remote source or forwarder node. This is due to calculation of a unique overall trust value per neighbor regardless its actions against packets arrived from individual nodes. In order to handle such a remote user-domain conflicting behavior attack, the behavior of a neighbor should be evaluated by dedicated trust levels for each packet source.
- An important research direction consists in reducing the amount of energy consumption and computation cost of the watchdog mechanism. As a primary solution, we propose to use a cognitive engine to provisionally switch off the watchdog when a satisfactory level of local/network trust is predicted, suggesting a similar trust context for a while.

In addition, the research community should investigate solutions to handle more-complicated trust-distortion attacks. Mainly, existing mechanisms against iterative on-off, bad-mouthing in presence of liar nodes or simultaneous such attacks should be reconsidered. For instance, the proposed adaptive forgetting factor proposed to handle iterative on-off threats is not capable to recognize an attacker after a long normal conduct period. In order to keep track of such attackers, we propose that each node registers the list of all nodes detected previously as untrustworthy, called misbehaving list. For each node within the misbehaving list, an attack persistence index and a current prudent trust level are stored. The attack persistence index consists in the number of times the node is detected as attacker. The prudent trust level designates the trust level of the attacker node re-initiated after being detected and is weighted inversely proportional to the attack persistence index. Nodes within the misbehaving list should earn a prudent trust level more than a predefined threshold to be considered as a normal node. However, the identifier of such a node should remain in the misbehaving list for a while to be able to react to its probable future attacks accordingly.

Another important open issue raises from the contradictory nature of existing defense mechanisms against various types of attack. In one hand, recommendations allow nodes to fortify their estimation on other nodes' behavior and get immune to double-face conducts. In the other hand, the bad mouthing attackers can communicate inaccurate recommendations, damaging nodes estimation on other nodes' trustworthiness. Assume a situation in which an entity receives a recommendation about a given node which is completely different from its opinion or from other received recommendations. The receiver confuses to consider this situation as the recommender's bad mouthing or as the evaluated node's double-face conduct. Consequently, an ideal trust management framework should use strategies to simultaneously accommodate several such contradictory situations.

We believe that, in a such contradictory environment, any bad mouthing behavior should be identified locally, i.e. exactly where the attack is performed. To achieve this, the watchdog mechanism should be enriched so that each node computes and stores the approximative trust table of each of its neighbors. We recognize that this table may be inexact since communications of some neighbors of a neighbor might not be overheard. However, it can be useful to suggest in some extent the recommendations expected to be sent by a neighbor. Recommendations significantly different from what it is expected to be overheared from a neighbor should be neglected and refused to be broadcasted to the whole network.

Another area of research relates to the identification of more complicated types of potential trust-distortion attacks and their possible countermeasures. In the following, we identify some possible attacks not discussed in the literature:

- The *modification attack* which consists in modifying the content of received recommendation packets before relaying them in order to achieve the same objective as bad mouthing recommenders. In order to identify such attackers, we propose a variant of the approach proposed for bad mouthing threat. Upon overhearing a recommendation packet, the node should keep track of the retransmission of that recommendation by its neighbor. This latter can be easily checked for TMFs rebroadcasting a

recommendation as it is received. However, for TMFs using their own trust table to update recommendations before retransmitting, the node should estimate the recommendation it expects to receive from that neighbor based on its approximated neighbor trust table. Detected suspicious recommendations will be ignored by neighbors before they are propagated to the whole network.

- The *packet-domain conflicting behavior attack* which consists in dropping all or certain recommendation packets in order to perturb nodes' view on other nodes. To handle this kind of trust-distortion attacks, the watchdog mechanism should calculate a dedicated trust level to track specially the node behavior vis-a-vis the recommendation packets.
- The *recommendation delaying attack* which consists in forwarding received recommendation packets with an imposed delay. The main objective of such an attack is to deceive nodes opinion by out-of-date recommendations. This kind of attack can be detected by considering a bad action for any recommendation packet forwarded after a delay longer that the normal delay estimated to be spent by a packet in the evaluated node.

VII. Conclusion

While trust management has attracted considerable attention from research community, the way towards an ideal framework that handles all identified attacks is long. This work provides a holistic view of research in the area of trust management, aiming at identifying the pros and cons of each existing model and their resistance to main attacks targeting the accuracy of nodes perception on other nodes' trustworthiness level, referred to as trust-distortion attacks. We proposed a taxonomy of main identified trust-distortion attacks, classifying them into double-face and bad mouthing conducts. Based on the proposed taxonomy, we categorized trust management frameworks into models resistant to double face and those immune to bad mouthing attacks. We identified four major efforts in the first category: Bella, ATMS, Almotiri, OCEAN, Data Class and TAODV. The main models resistant to bad mouthing attacks that we managed to find in the literature were OTMF, LARS, AFStrust, CONFIDANT, mTrust, Trudi, Dynamic Trust Model, HTP, AOTDV, CORE, Lindsay, SORI, TEAM and DRDM. Moreover, we identified and classified the main metrics that can be used to evaluate and compare the efficiency of trust management frameworks.

Furthermore, we compared different trust management frameworks based on their key properties and characteristics, and outlined main techniques required to cope with different trust-distortion attacks simultaneously. In addition, we described future research directions, identified a set of novel trust-distortion attacks and proposed some solutions to cope with these treats. We outlined that the use of trust values in network control and management can help protecting from security attacks. Although some surveyed trust proposals applied trust values in the routing process, there is a special need to propose protocol-independent trust-based routing schemes for mobile ad hoc networks. In addition, trust information can be largely useful to make secure inter-agent relationships between autonomic managers, ensuring an optimized management overlay in distributed environments.

References

[1] A. Abdul-Rahman and S. Hailes, "Using recommendations for managing trust in distributed systems," in *Proc. IEEE Malaysia Int. Conf. Commun. (MICC'97), Kuala Lumpur, Malaysia*, 1997.

[2] W. Adams, G. Hadjichristofi, and N. Davis, "Calculating a node's reputation in a mobile ad hoc network," in *Proc. 24th IEEE Int. Perform. Comput. Commun. Conf. (IPCCC'05)*, Apr. 2005, pp. 303–307.

[3] S. Agrawal, S. Jain, and S. Sharma, "A survey of routing attacks and security measures in mobile ad-hoc networks," *J. Comput.*, vol. 3, no. 1, pp. 41–48, Jan. 2011.

[4] S. Almotiri and I. Awan, "Trust routing in MANET for securing DSR routing protocol," *PGNet*, 2010.

[5] V. Balakrishnan, V. Varadharajan, U. Tupakula, and P. Lucs, "Team: Trust enhanced security architecture for mobile ad-hoc networks," in *Proc. 15th IEEE Int. Conf. Netw. (ICON'07)*, Nov. 2007, pp. 182–187.

[6] S. Bansal and M. Baker, "Observation-based cooperation enforcement in ad hoc networks," 2003, pp. 120–130.

[7] G. Bella, G. Costantino, and S. Riccobene, "Managing reputation over MANETS," in *Proc. 4th Int. Conf. Inf. Assur. Security* (IAS), 2008, pp. 255–260.

[8] A. P. Bianzino, C. Chaudet, D. Rossi, and J.-L. Rougier, "A survey of green networking research," *IEEE Commun. Surveys Tuts.*, vol. 14, no. 1, pp. 3–20, Jan. 2012.

[9] S. Buchegger and J.-Y. Le Boudec, "Performance analysis of the confidant protocol," in *Proc. 3rd ACM Int. Symp. Mobile Ad Hoc Netw. & Comput. (MobiHoc'02)*, 2002, pp. 226–236.

[10] I.-R. Chen, J. Guo, F. Bao, and J.-H. Cho, "Trust management in mobile ad hoc networks for bias minimization and application performance maximization," *Ad Hoc Netw.*, vol. 19, pp. 59–74, 2014.

[11] J.-H. Cho, A. Swami, and I.-R. Chen, "A survey on trust management for mobile ad hoc networks," *IEEE Commun. Surveys Tuts.*, vol. 13, no. 4, pp. 562–583, Nov. 2011.

[12] K. Cook, "Trust in society," *Russell Sage Found. Ser. Trust*, vol. 2, no. 5, pp. 3–40, Feb. 2003.

[13] W. Dai, L. E. Moser, P. M. Melliar-Smith, and M. Lomber, "The itrust local reputation system for mobile ad-hoc networks," in *Proc. Int. Conf. Wireless Netw.*, Jul. 2013, pp. 162–168.

[14] R. Dala, M. Khari, and Y. Singh, "Different ways to achieve trust in MANET," *Int. J. AdHoc Netw. Syst. (IJANS)*, vol. 2, no. 2, pp. 53–64, Apr. 2012.

[15] R. Dalal, M. Khari, and Y. Singh, "Survey of trust schemes on ad-hoc network," in *Advances in Computer Science and Information Technology. Networks and Communications*, N. Meghanathan, N. Chaki, and D. Nagamalai, Eds. New York, NY, USA: Springer, 2013, pp. 170–180.

[16] H. Durad, Y. Cao, and Z. Liehuang, "Two novel trust evaluation algorithms," in *Proc. Int. Conf. Commun. Circuits Syst.*, 2006, vol. 3, pp. 1641–1646.

[17] L. Eschenauer, V. Gligor, and J. Baras, "On trust establishment in mobile ad-hoc networks," in *Security Protocols*, B. Christianson, B. Crispo, J. Malcolm, and M. Roe, Eds. New York, NY, USA: Springer, 2004, pp. 47–66.

[18] J. Gandhi and R. Jhaveri, "Addressing packet forwarding misbehaviour using trust-based approach in ad-hoc networks: A survey," in *Proc. Int. Conf. Signal Process. Commun. Eng. Syst. (SPACES)*, Jan. 2015, pp. 391–396.

[19] K. Govindan and P. Mohapatra, "Trust computations and trust dynamics in mobile adhoc networks: A survey," *IEEE Commun. Surveys Tuts.*, vol. 14, no. 2, pp. 279–298, May 2012.

[20] Q. He, D. Wu, and P. Khosla, "SORI: A secure and objective reputation-based incentive scheme for ad-hoc networks," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC'04)*, 2004, pp. 825–830.

[21] J. Hu and M. Burmester, "Cooperation in mobile ad hoc networks," in *Proc. Guide Wireless Ad Hoc Netw.*, 2009, pp. 43–57.

[22] N. Iltaf, A. Ghafoor, and U. Zia, "A mechanism for detecting dishonest recommendation in indirect trust computation," *EURASIP J. Wireless Commun. Netw.*, vol. 2013, no. 1, pp. 101–189, 2013.

[23] A. Ingle and S. Nimbhorkar, "A review on secure communication protocol for wireless ad hoc network," in *Proc. Conf. Pervasive Comput. (ICPC)*, Jan. 2015, pp. 1–4.

[24] D. Johnson, Y. Hu, and D. Maltz, "The dynamic source routing protocol (DSR) for mobile ad hoc networks for IPv4," Internet Engineering Task Force, Feb. 2007.
[25] J. Li, R. Li, and J. Kato, "Future trust management framework for mobile ad hoc networks," *IEEE Commun. Mag.*, vol. 46, no. 4, pp. 108–114, Apr. 2008.
[26] W. Li and A. Joshi, "Security issues in mobile ad hoc networks—A survey," Dept. Comput. Sci. Elect. Eng., Univ. Maryland, Baltimore County, MD, USA, 2008, pp. 1–23.
[27] W. Li and A. Joshi, "Outlier detection in ad hoc networks using Dempster-Shafer theory," in *Proc. 10th Int. Conf. Mobile Data Manage. (MDM'09)*, May 2009, pp. 112–121.
[28] W. Li, A. Joshi, and T. Finin, "Coping with node misbehaviors in ad hoc networks: A multi-dimensional trust management approach," in *Proc. 11th Int. Conf. Mobile Data Manage. (MDM'10)*, May 2010, pp. 85–94.
[29] X. Li, M. R. Lyu, and J. Liu, "A trust model based routing protocol for secure ad hoc networks," in *Proc. IEEE Aerosp. Conf.*, Mar. 2004, vol. 2, pp. 1286–1295.
[30] X. Li, Z. Jia, L. Wang, and H. Wang, "Trust-based on-demand multipath routing in mobile ad hoc networks," *IET Inf. Security*, vol. 4, no. 4, pp. 212–232, Dec. 2010.
[31] Z. Liu, A. W. Joy, and R. A. Thompson, "A dynamic trust model for mobile ad hoc networks," in *Proc. 10th IEEE Int. Workshop Future Trends Distrib. Comput. Syst. (FTDCS'04)*, May 2004, pp. 80–85.
[32] J. Luo and M. Fan, "A subjective trust management model based on certainty-factor for MANETS," *Chin. J. Comput. Res. Develop.*, vol. 47, no. 3, pp. 515–523, 2010.
[33] P. Michiardi and R. Molva, "Core: A collaborative reputation mechanism to enforce node cooperation in mobile ad hoc networks," Deventer, The Netherlands: Kluwer, B.V., pp. 107–121, 2001.
[34] M. Morvan and S. Sene, "A distributed trust diffusion protocol for ad hoc networks," in *Proc. Int. Conf. Wireless Mobile Commun. (ICWMC)*, 2007, pp. 87–92.
[35] Z. Movahedi, M. Nogueira, and G. Pujolle, "An autonomic knowledge monitoring scheme for trust management on mobile ad hoc networks," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, Apr. 2012, pp. 1898–1903.
[36] C. E. Perkins and E. M. Royer, "Ad-hoc on-demand distance vector routing," in *Proc. 2nd IEEE Workshop Mobile Comput. Syst. Appl.*, 1997, pp. 90–100.
[37] T. L. Saaty, *The Analytic Hierarchy Process*. New York, NY, USA: McGraw-Hill, 1980.
[38] S. Sen, J. A. Clark, and J. E. Tapiador, "Security threats in mobile ad hoc networks," in *Security of Self-Organizing Networks: MANET, WSN, WMN, VANET*. New York, NY, USA: Auerbach, 2010, pp. 127–147.
[39] M. Seredynski and P. Bouvry, "Nature-inspired evaluation of data classes for trust management in MANETS," in *Proc. IEEE 26th Int. Parallel Distrib. Process. Symp. Workshops & PhD Forum*, 2011, pp. 366–373.
[40] G. Shafer, *A Mathematical Theory of Evidence*. Princeton, NJ, USA: Princeton Univ. Press, 1976.
[41] S. Sivagurunathan and C. Prathap, "Trust based security schemes in mobile ad hoc networks—A review," in *Proc. Int. Conf. Intell. Comput. Appl. (ICICA)*, Mar. 2014, pp. 291–295.
[42] S. Staab *et al.*, "The pudding of trust," *IEEE Intell. Syst.*, vol. 19, no. 5, pp. 74–88, Sep. 2004.
[43] Y. L. Sun, Z. Han, W. Yu, and K. J. Ray Liu, "A trust evaluation framework in distributed networks: Vulnerability analysis and defense against attacks," in *Proc. IEEE INFOCOM*, 2006, pp. 230–236.
[44] Y. L. Sun, W. Yu, Z. Han, and K. J. R. Liu, "Information theoretic framework of trust modeling and evaluation for ad hoc networks," *IEEE J. Sel. Areas Commun.*, vol. 24, no. 2, pp. 305–317, Feb. 2006.
[45] P. B. Velloso, R. P. Laufer, D. O. Cunha, O. C. M. B. Duarte, and G. Pujolle, "Trust management in mobile ad hoc networks using a scalable maturity-based model," *IEEE Trans. Netw. Serv. Manage.*, vol. 7, no. 3, pp. 172–185, Sep. 2010.
[46] Z. Wei, H. Tang, F. Richard Yu, M. Wang, and P. Mason, "Security enhancements for mobile ad hoc networks with trust management using uncertain reasoning," *IEEE Trans. Veh. Tech.*, vol. 63, no. 9, pp. 4647–4658, Nov. 2014.
[47] Y. Wu, Y. Zhao, M. Riguidel, G. Wang, and P. Yi, "Security and trust management in opportunistic networks: A survey," *Security Commun. Netw.*, vol. 8, no. 9, pp. 1812–1827, 2015.
[48] H. Xia, Z. Jia, L. Ju, X. Li, and Y. Zhu, "A subjective trust management model with multiple decision factors for MANET based on AHP and fuzzy logic rules," in *Proc. IEEE/ACM Int. Conf. Green Comput. Commun. (GREENCOM'11)*, 2011, pp. 124–130.
[49] Y. Yuan, H. Chen, and M. Jia, "An optimized ad-hoc on-demand multipath distance vector (AOMDV) routing protocol," in *Proc. Asia-Pac. Conf. Commun.*, 2005, pp. 569–573.
[50] Z. H Islam and A. A. Khan, "Detection of dishonest trust recommendations in mobile ad hoc networks," in *Proc. Int. Conf. Comput. Commun. Netw. Technol. (ICCCNT)*, Jul. 2014, pp. 1–7.
[51] Z. Movahedi, M. Ayari, R. Langar, and G. Pujolle, "A survey of autonomic network architectures and evaluation criteria," in *Commun. Surveys Tuts.*, vol. 14, no. 2, pp. 464–490, Second Quart., 2012.

Zeinab Movahedi received the M.Sc. and Ph.D. degrees in computer networks and telecommunications from the University of Pierre and Marie Curie (Paris 6), Laboratoire d'Informatique de Paris 6 (LIP6), Paris, France, in 2007 and 2011, respectively. She is currently an Assistant Professor at IUST. Her research interests include green communication, mobile cloud computing, software-defined networks, autonomic networking, wireless networks, network security, performance evaluation, and quality-of-service support.

Zahra Hosseini received the B.S. degree in computer engineering and the M.Sc. degree in information technology from Qom University, Qom, Iran, in 2011 and 2013, respectively. She is pursuing the Ph.D. degree at Qom University. She has worked with trust management and QoS support in mobile ad hoc networks. Her research interests include wireless ad hoc networks, security, and green computing.

Fahimeh Bayan received the B.S. degree in computer engineering and M.Sc. Degree in information technology from University of Qom, Qom, Iran, in 2011 and 2013, respectively. She cooperates currently with the network laboratory of IUST. She has worked on trust management in MANETs for some years. Her research interests include wireless networks, autonomic computing, and network security.

Guy Pujolle received the Ph.D. and "These d'Etat" degrees in computer science from the University of Paris IX and Paris XI, in 1975 and 1978, respectively. He is currently a Professor at Pierre et Marie Curie University-Paris 6, a Distinguished Invited Professor at POSTECH, Korea, a member of the Institut Universitaire de France, and a member of The Royal Physiographical Academy of Lund, Sweden. He spent the period 1994–2000 as Professor and Head of the computer science department of Versailles University. He was also Professor and Head of the MASI Laboratory at Pierre et Marie Curie University (1981–1993), Professor at ENST (1979–1981), and a member of the scientific staff of INRIA (1974–1979). He is the French representative at the Technical Committee on Networking at IFIP. He is an Editor for *ACM International Journal of Network Management, Telecommunication Systems,* and Editor in Chief of *Annals of Telecommunications*. He is a pioneer in high-speed networking having led the development of the first Gbit/s network to be tested in 1980. He has participated in several important patents like DPI or virtual networks. Guy Pujolle is Co-founder of QoSMOS (www.qosmos.fr), Ucopia Communications (www.ucopia.com), EtherTrust (www.ethertrust.com), Virtuor (www.VirtuOR.fr), and Green Communications (www.green-communications.fr).