Seo *et al. EURASIP Journal on Wireless Communications and Networking* (2017) 2017:155
DOI 10.1186/s13638-017-0938-8

EURASIP Journal on Wireless Communications and Networking

RESEARCH Open Access

Reconfiguration time and complexity minimized trust-based clustering scheme for MANETs

Sunho Seo[1], Jin-Won Kim[1,2], Jae-Dong Kim[1,2] and Jong-Moon Chung[1*]

Abstract

A trust management mechanism for mobile ad hoc networks (MANETs) is proposed to cope with security issues that MANETs face due to time constraints as well as resource constraints in bandwidth, computational power, battery life, and unique wireless characteristics. The trust-based reputation scheme *GlobalTrust* is a reliable trust management mechanism. In this paper, a clustering algorithm is applied to the *GlobalTrust* scheme (named Cluster-based GlobalTrust (*CGTrust*)) to find the optimal group size to minimize the configuration time, which consists of trust information computational time and complexity, while having to satisfy the trust reliability requirements. The optimal number of clusters is derived from the minimizing point of the computation complexity function. Simulation results show that the computational time and complexity of *CGTrust* are controllable and can be used effectively in time critical network operations that require trust analysis.

Keywords: MANET, Trust, Cluster, GlobalTrust, CGTrust

1 Introduction

Mobile ad hoc networks (MANETs) consist of distributed wireless mobile nodes. In MANETs, it is critical to make a decision of assessing the trustworthiness of participating nodes accurately by a trust authority. Trust evaluations are made on node reputation, which is the perception of a node evaluated by other nodes. A node's reputation is evaluated based on the collection of trust evaluations made on that node by other nodes [1, 2]. Cho et al. [1] analyzed the concepts and properties of trust. In addition, they surveyed MANET trust management schemes for secure routing, authentication, and intrusion detection, etc. Aberer and Despotovic [3] proposed a trust-based reputation management scheme. The authors of [4, 5] proposed distributed methods for reputation management. However, results in [3–5] are vulnerable to collusive attacks. Reputation management schemes in [6–9] subjectively evaluate reputation using direct observation that disturbs the global view. Quorum-based reputation management schemes are proposed based on *k-out-of-n* threshold signatures [10, 11]. The *k*-means-based reputation scheme is presented in [12, 13], which does not consider conflicting recommendation attacks. Chen et al. [2] proposed *GlobalTrust* for accurate decision-making of a trusted authority (TA) under various attacks, which shows an outstanding performance in trust on MANETs. The TA needs to collect trust-related data and compute trust values and set up optimal routing paths, which is a very computationally burdening task. Thus, a common issue would be the time constraints and required computing resources of the TA and MANET nodes. The computational time delay and complexity of *GlobalTrust* significantly increases as the number of nodes in the MANET increases. In addition, in a MANET, communicating nodes are mobile. Therefore, route reconfiguration may need to be conducted frequently. As a result, reconfiguration time minimization would be necessary. In this paper, a Cluster-based GlobalTrust (*CGTrust*) scheme is proposed with the objective to minimize the computational complexity and reconfiguration time delay experienced by a TA while supporting the required trust reliability requirements. *CGTrust* consists of the following three unique features.

*Correspondence: jmc@yonsei.ac.kr
[1]School of Electrical and Electronic Engineering, Yonsei University, Seoul, Republic of Korea
Full list of author information is available at the end of the article

- *CGTrust* evaluates trust at cluster heads (CHs) and at the TA in contrast to *GlobalTrust* that focuses on the TA. This approach has the benefit of requiring less time and uses less computing resources at the TA in the network.
- *CGTrust* provides a computational complexity minimized mechanism to form MANET clustering, where the complexity considers both intra and inter cluster computations. This algorithm helps to drastically reduce the complexity associated with the trust evaluation and computation.
- *CGTrust* provides a mechanism to evaluate the trust of both non-CH nodes and CHs to prevent false trustworthiness decisions of non-CH nodes by a non-trustworthy CH and minimize the setup and reconfiguration time.

2 CGTrust

The proposed scheme is for MANETs that use the *GlobalTrust* trust-based reputation scheme [2]. It is assumed that MANETs consist of multiple nodes communicating via multiple hops. The objective of *CGTrust* is to provide a trust-based clustering algorithm that derives the optimal number of clusters k (i.e., $k_{optimal}$) to minimized network setup and reconfiguration time delay based on the constraints of the required trust level that needs to be satisfied. *CGTrust* forms clusters by using a trust-based reputation algorithm, where the TA makes decisions based on trust values. The TA is a system that can determine the trust level of all nodes in the network and manage the security and performance of the network [2]. Therefore, it is assumed that the TA does not perform the role of a node in the network, rather the TA exists independently and manages the network because the TA requires a large amount of resources to perform the trustworthiness evaluation functions. The TA collects evidence periodically to assess the trust of participating nodes through CHs and makes a trustworthy decision. In the initial setup stage, if no CHs have been preselected, then the TA will select CHs and provide initial cluster information (where this initial formation may not be an optimal setting). If The number of nodes changes, the TA recomputes the optimal number of clusters, and then reassigns CHs among the nodes with the highest reputation values. Figure 1 is a diagram

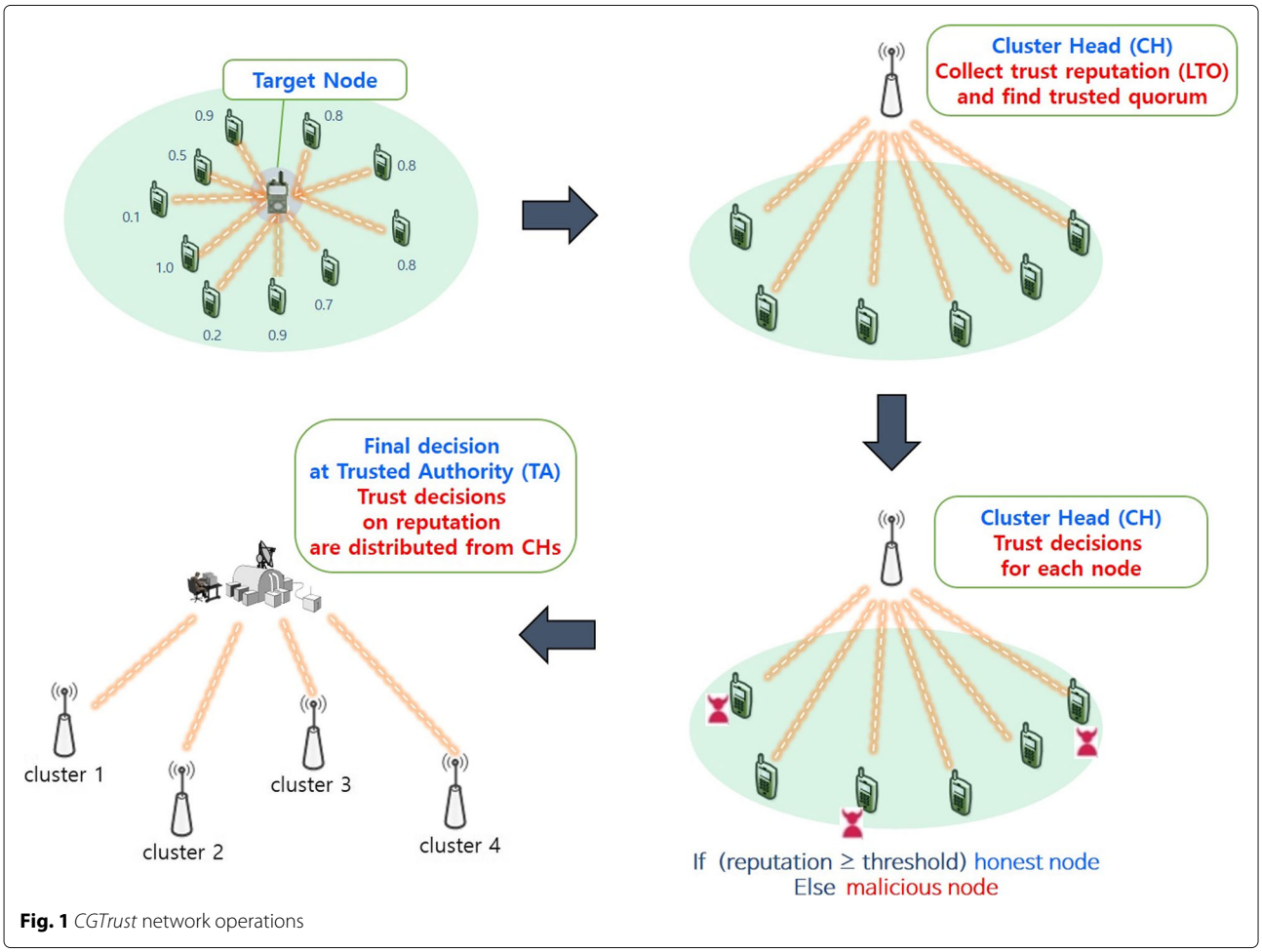

Fig. 1 *CGTrust* network operations

that illustrates the operations of *CGTrust*. The *CGTrust* process is described in the following steps (Fig. 2).

Step 1: Non-CH nodes that belong to the *i*th cluster transmit local trust opinions (LTOs) of their neighbor nodes to the local CH. $\mathrm{LTO}_{w,u}$ is node *w*'s trust opinion towards node *u* based on direct observations. $\mathrm{LTO}_{w,u}$ is calculated as below

$$\mathrm{LTO}_{w,u} = \frac{p_{w,u}}{p_{w,u} + n_{w,u}} \tag{1}$$

where $p_{w,u}$ and $n_{w,u}$ are respectively the total number of positive events and the total number of negative events. If there are no events between node *w* and node *u*, the LTO is null. The sum of positive and negative events $(p_{w,u} + n_{w,u})$ is the sum of the number of events sent by node *w* to node *u*. Positive events represented by $p_{w,u}$ are events that node *w* is determined that a packet was transmitted well to node *u*. Negative events represented by $n_{w,u}$ are events that node *w* is determined that a packet was not delivered to node *u* for some reasons (noise, wrong decision, intentional packet drop, etc.). The transmission can be determined by an acknowledgement packet, etc. As can be seen from Eq. (1), the higher the positive events, the closer to 1 the LTO and the closer to 0 the negative events become. The higher the LTO, the trust opinion of node *u* approaches the reputation of an honest node.

An honest node correctly sends a packet to a predefined route when it receives a packet from another node, except for errors that occur during the transmission process. In addition, when the LTO is reported to the TA or CH, the reporting of the LTO is performed without distortion.

A malicious node may drop a packet when it is received from another node, or intentionally send a packet to a different route. In addition, information may be distorted during the transmission and reporting process of the LTO to the TA or CH.

The *i*th CH (CH_i) aggregates trust-related evidences and computes the subjective reputation (SR) of node *u* as it is assessed by node *w* ($\mathrm{SR}_{w,u}$) using

$$\mathrm{SR}_{w,u} = \sum_{j \in S_u} \mathrm{LTO}_{j,u} \frac{\mathrm{HR}_j \mathrm{Sim}(w,j)}{\sum_{j \in S_u} \mathrm{HR}_j \mathrm{Sim}(w,j)} \tag{2}$$

where $\mathrm{Sim}(w,j)$ is the *cosine* similarity of the LTOs between node *w* and node *j*, S_u is the set of nodes that have non-null LTOs over node *u* (including *w* if *w* has one), and HR_j is the hierarchical rank of node *j*. HR_j can specify that administrators have different values, or they can all have the same value, depending on how trustworthy the node's opinion is [2]. The TA should have a hierarchy rank higher than that of any CH or node, because the TA's opinion is more strongly reflected than the CHs or nodes as it is where the reputation judgment is made. In this paper, the hierarchy rank is designated as TA = 3, CH = 2, and node = 1.

After calculating the SR, CH_i makes a SR matrix. The SR tuple in node *w*'s view is denoted as a vector. CH_i compares all SR tuples, and merges the two SR tuples with the least difference based on the agglomerative hierarchical clustering technique [2]. CHs use this technique to form its trusted quorum (D_{CH}). The process of finding the trusted quorum ends when the set having the largest number of nodes has more than half of the total number of network nodes. When the total number of nodes is *N*, the number of nodes belonging to the trusted quorum may have a value between $[N/2, N]$, depending on how it is calculated. Since all SR tuples are compared and a trusted quorum is found, the complexity does not change depending on the size of the trusted quorum.

Based on the trusted quorum, CH_i obtains the behavioral reputation (BR) of node *u* ($\mathrm{BR}_{\mathrm{CH}_{i,u}}$). $\mathrm{BR}_{\mathrm{CH}_{i,u}}$, reflecting how CH_i views node *u*'s network behavior, is computed by averaging the SR tuples using (3).

$$\mathrm{BR}_{\mathrm{CH}_{i,u}} = \frac{\sum_{w \in D} \mathrm{SR}_{w,u}}{|D_{CH}|} \tag{3}$$

Then CH_i computes the credibility reputation (CR) of node *u* ($\mathrm{CR}_{\mathrm{CH}_{i,u}}$) using (4). $\mathrm{CR}_{\mathrm{CH}_{i,u}}$ indicates how trustworthy *u*'s reported LTOs are. This is computed based on

Algorithm: CGTrust

At the CH – Compute trust of the cluster

1: **COMPUTE** local trust opinion *LTO*
2: **COMPUTE** *Sim* and *SR*
3: **FIND** CH's trusted quorum *D*
4: **COMPUTE** *BR*, *CR* and *GR*

At the TA – Compute trust of all nodes

5: **IF** cluster is using encrypted packet mode (β)
    COLLECT encrypted *LTOs* from non-CH nodes
6: **ELSE**
    COLLECT *GR* from CH
7: **COMPUTE** *Sim* and *SR*
8: **FIND** TA's trusted quorum *D*
9: **COMPUTE** *BR*, *CR*, and *GR*
10: **IF** *GR* is unknown
    SET the trust level of the node as unknown
11: **ELSEIF** *GR* ≥ decision factor θ
    SET the trust level of the node as a honest node
12: **ELSE**
    SET the trust level of the node as a malicious node
14: **IF** *N* has changed
    COMPUTE $k_{optimal}$ using (15)
    Assign nodes to the $k_{optimal}$ clusters and select CHs

Fig. 2 *CGTrust* algorithm

the difference between u's reported LTOs and BRs of the nodes that node u has reported LTOs over.

$$\mathrm{CR}_{\mathrm{CH}_{i,u}} = 1 - \sqrt{\frac{\sum_{j\in\{\mathrm{LTO}_{u,j}\neq\mathrm{null}\}}\left(\mathrm{LTO}_{u,j} - \mathrm{BR}_{\mathrm{CH}_{i,j}}\right)^2}{|j \in \{\mathrm{LTO}_{u,j} \neq \mathrm{null}\}|}} \quad (4)$$

Next CH_i computes the global reputation (GR) of node u ($\mathrm{GR}_{\mathrm{CH}_{i,u}}$) using the normalized factor γ selected from the range in [0,1].

$$\mathrm{GR}_{\mathrm{CH}_{i,u}} = \gamma \mathrm{BR}_{\mathrm{CH}_{i,u}} + (1-\gamma)\mathrm{CR}_{\mathrm{CH}_{i,u}} \quad (5)$$

Step 2: Once a CH computes all GRs of its cluster's nodes, it sends the result information to the TA. The TA computes the SR from GR based on the resultant data as in (6). The cosine similarity ($\mathrm{Sim}(\mathrm{CH}_i, \mathrm{CH}_k)$) in Eq. (6) is calculated on the basis of the LTO between the CH communication. This allows the TA to determine if a CH is infected and can make a final decision. Step 2 is performed when the number of clusters is two or more. If the network is not divided into clusters, the TA directly makes all trust decisions of all nodes in the network.

$$\mathrm{SR}_{\mathrm{CH}_{i,u}} = \sum_{\mathrm{CH}_k\in S'_u} \mathrm{GR}_{\mathrm{CH}_k,u} \frac{\mathrm{HR}_{\mathrm{CH}_k}\mathrm{Sim}(\mathrm{CH}_i, \mathrm{CH}_k)}{\sum_{\mathrm{CH}_k\in S'_u}\mathrm{HR}_{\mathrm{CH}_k}\mathrm{Sim}(\mathrm{CH}_i, \mathrm{CH}_k)} \quad (6)$$

After (6), the TA calculates the BR of node u in (7), and the CR of node u (i.e., $\mathrm{CR}_{\mathrm{TA},u}$) in (8).

$$\mathrm{BR}_{\mathrm{TA},u} = \frac{\sum_{\mathrm{CH}_i\in D}\mathrm{SR}_{\mathrm{CH}_{i,u}}}{|D_{\mathrm{TA}}|} \quad (7)$$

$$\mathrm{CR}_{\mathrm{TA},u} = 1 - \sqrt{\frac{\sum_{u\in\{\mathrm{GR}_{\mathrm{CH}_{i,u}}\neq\mathrm{null}\}}\left(\mathrm{GR}_{\mathrm{CH}_{i,u}} - \mathrm{BR}_{\mathrm{TA},u}\right)^2}{|u \in \{\mathrm{GR}_{\mathrm{CH}_{i,u}} \neq \mathrm{null}\}|}} \quad (8)$$

Then the TA computes the GR of node u (i.e., $\mathrm{GR}_{\mathrm{TA},u}$) using

$$\mathrm{GR}_{\mathrm{TA},u} = \gamma'\mathrm{BR}_{\mathrm{TA},u} + (1-\gamma')\mathrm{CR}_{\mathrm{TA},u} \quad (9)$$

Step 3: After the TA computes the GR, the TA decides the trustworthiness of each node using

$$\mathrm{Decision}(u) = \begin{cases} \text{unknown} & \text{if } \mathrm{GR}_{\mathrm{TA},u} = \text{unknown} \\ \text{honest} & \text{if } \mathrm{GR}_{\mathrm{TA},u} \geq \theta \\ \text{malicious} & \text{if } \mathrm{GR}_{\mathrm{TA},u} < \theta \end{cases} \quad (10)$$

where θ is a decision factor selected from the range in [0, 1]. The detection errors can be reduced by selecting the most appropriate θ value.

The TA evaluates each CH's trustworthiness using direct trust computation, which can be conducted using the encrypted packet mode, which encrypts packets exchanged between non-CH nodes and the TA. In this mode, non-CH nodes send (encrypted) information packets to their CH and the CH forwards these packets to the TA without trust computation. Using the encrypted packet mode, the TA computes the CHs' trustworthy level periodically considering β, which is the ratio of nodes that use encrypted packet mode in the cluster. In this computation process, β is a variable that represents the possibility that a CH is a malicious node. As the value of β increases, the number of nodes that the TA needs to directly compute increases, making it difficult to reduce the computational complexity. Considering the computational complexity of trust computation and the worst case where the majority of CHs are infected, the suitable value of β is [0, 0.5].

3 Cluster-based network analysis

3.1 Computational complexity analysis

The proposed scheme computes the GR of each node based on *GlobalTrust* that uses a trusted quorum *D*. *GlobalTrust* uses the *agglomerative hierarchical clustering technique* to find a minimum dominating cluster [2].

For an accurate complexity analysis, the method of [14] is applied to *CGTrust*, where the complexity of the pseudocode steps is computed. Each cluster has N/k nodes. Every node will collect the SRs of all other nodes in its cluster, which are ((N/k)-1) SRs. Two nodes in a cluster will pair up and compare their collected SRs, but will exclude the SR of the paired node in this comparison process. Therefore, ((N/k)-2) SRs will be compared by the node pair. In addition, since there are $\binom{N/k}{2}$ combinations of possible node pairs in each cluster, the computational complexity of one cluster is $\mathrm{O}[((N/k)\text{- }2)(N/k)((N/k)\text{-}1)/2] = \mathrm{O}((N/k)^3)$ (step 3).

In case of inter-clusters, the TA uses $((1-\beta)k+\beta N-1)$ SRs for each node since the TA computes the GR directly at the rate of β. The minimum pair complexity becomes $\mathrm{O}(((1-\beta)k+\beta N-1)^2)$. The computational complexity for all steps is $\mathrm{O}(((1-\beta)k+\beta N-1)((1-\beta)k+\beta N-2)/2))$ and based on this the computational complexity of each node is $\mathrm{O}(((1-\beta)k+\beta N-1)^2) + \mathrm{O}(((1-\beta)k+\beta N-1)((1-\beta)k+\beta N-2)/2))$. Therefore, to compute the TA's trusted quorum the required complexity is $\mathrm{O}[\,((1-\beta)k+\beta N-1)^2 + ((1-\beta)k+\beta N-1)((1-\beta)k+\beta N-2)/2)(1-\beta)k+\beta N]$ (step 9). After organizing the terms, the computational complexity can be expressed as $\mathrm{O}((1\text{-}\beta)^3k^3 + (\beta N)^3)$. In *CGTrust*, the complexity of both intra-cluster and inter-clusters are considered. Therefore, the total computational complexity (C_{total}) becomes

$$C_{\mathrm{total}} = O\left((N/k)^3 + (1-\beta)^3k^3 + (\beta N)^3\right) \quad (11)$$

3.2 Trust information computation time

In order to minimize the time required to evaluate the trust profile of a large MANET, the optimal cluster size (k_{optimal}) that minimizes the computational complexity used in evaluating the trust profile of all nodes in the network is derived. As shown in Fig. 3, the computational complexity is directly proportional to the computational time required in evaluating the trust profile of all nodes of the MANET.

3.3 Minimization of computational complexity

The main objective of the *CGTrust* scheme is to minimize the computational complexity that results in a minimized network reconfiguration time delay. The optimization statement and constraints are established as below.

$$\begin{aligned} \underset{k}{\text{minimize}} \quad & C_{\text{total}}(k) \\ \text{subject to} \quad & 0 \le \beta \le 0.5, \\ & N, k \in \mathbb{Z}_{>0}. \end{aligned}$$

The total computational complexity in the TA is a function of k in the form of C_{total}, where the computational complexity minimizing optimal number of clusters can be obtained from

$$\begin{aligned} k' &= \arg\left[\frac{\partial C_{\text{total}}(k)}{\partial k} = 0\right] \\ &= \arg\left[-\frac{3\left(N^3 + (\beta-1)^3 k^6\right)}{k^4} = 0\right] \end{aligned} \tag{12}$$

which results in the following candidate solutions.

$$k' = \left\{ \begin{array}{ccc} \frac{\sqrt{N(1-\beta)}}{1-\beta}, & \frac{-\sqrt{N(1-\beta)}}{1-\beta}, & -\sqrt{\frac{N(i\sqrt{3}+1)}{2(1-\beta)}}, \\ \sqrt{\frac{N\left(i\sqrt{3}+1\right)}{2(1-\beta)}}, & -\sqrt{\frac{N\left(i\sqrt{3}-1\right)}{2(1-\beta)}}, & \sqrt{\frac{N\left(i\sqrt{3}-1\right)}{2(1-\beta)}} \end{array} \right\} \tag{13}$$

There are 6 solutions of (13), where N is a positive number and the range of β is [0, 0.5]. Considering the fact that k needs to be a positive integer, the only feasible solution is $\frac{\sqrt{N(1-\beta)}}{1-\beta}$ which is a positive real number. In addition, the second derivative of the objective function is

$$\frac{\partial^2}{\partial k^2} C_{\text{total}}(k) = \frac{12N^3}{k^5} + 6(1-\beta)^3 k \tag{14}$$

in which $k' = \frac{\sqrt{N(1-\beta)}}{1-\beta}$ results in a positive value of the convex objective function $C_{\text{total}}(k)$. The optimal number of clusters (k_{optimal}) has to be a positive integer, and therefore, the nearest integer function (i.e., $Nint(\cdot)$) is applied to result in (15).

$$k_{\text{optimal}} = Nint\left(\frac{\sqrt{N(1-\beta)}}{1-\beta}\right) \tag{15}$$

Figure 4 presents the C_{total} profile and the k_{optimal} values for the range of interest based on N and k.

4 Performance evaluation

Malicious attack patterns investigated for the FN and FP performance evaluation include the following five attack patterns [2].

- Naive malicious attack (NMA): a malicious node provide improper services with probability α. However, it reports its LTOs honestly.
- Collusive rumor attack (CRA): In addition to providing improper services with probability α, malicious nodes collude to report false LTOs. Malicious nodes report LTOs of 1 to malicious node and LTOs of 0 to honest nodes.
- Non-collusive rumor attack (NRA): a malicious node can report a false LTO that is opposite to the observed evidence. For example, if an LTO is evaluated as p, the malicious node may report $(1-p)$ as the LTO.
- Malicious spy attack (MSA): some malicious nodes misbehave with probability α. Other malicious nodes behave honestly. These malicious nodes may collude

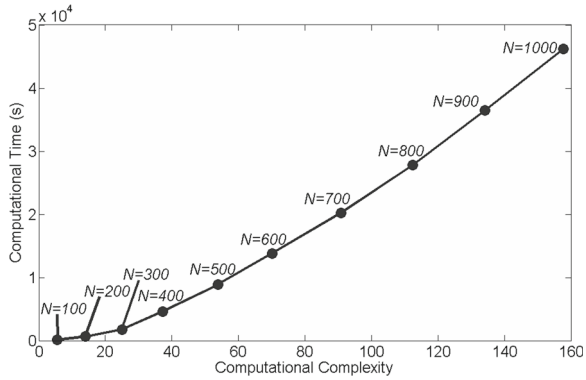

Fig. 3 Relation of computational complexity and time

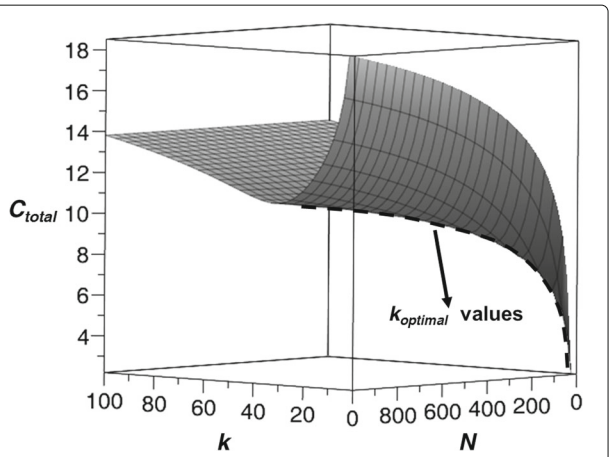

Fig. 4 Minimum points of C_{total} (k_{optimal}) according to N and k

and report LTOs of 1 to malicious node and LTOs of 0 to honest nodes to confuse the trust and reputation system.

- Conflicting behavior attack (CBA): malicious nodes can collude to confuse the trust and reputation system such as CRA and MSA. However, they misbehave only to some of honest nodes, and report LTOs of 1 to malicious node and LTOs of 0 to honest nodes to confuse the trust and reputation system. This attack causes LTO disagreement among honest nodes, which makes it difficult to find malicious nodes.

CBA is considered the most demanding type of attack because it makes it difficult to distinguish malicious nodes by confusing LTO information of honest nodes with respect to other nodes. For the above reasons, CBA was selected and evaluated.

The simulation based performance analysis of *CGTrust* and *GlobalTrust* was conducted using Matlab with N nodes randomly distributed with a uniform density in a 2×2 km^2 square area. Simulation parameters were set same to the experiments in [2], where the ratio of malicious nodes was set to 0.3 and every node randomly requests of its neighbor nodes to send a packet (which is multihop relayed) 100 times per minute, and β is in the range in [0, 0.5]. Honest nodes were made to drop packets based on a 0.05 packet error rate (PER) and the detection error probability of the monitoring system was set to 0.05. Each node was made to transmit trust data packets every 30 s and the TA computes the GRs based on the accumulated data of the past 30 min.

In addition, the probability that a malicious node drops a packet was set to 0.5, $\gamma = 0.7$, $\theta = 0.7$, and the upper bound probability of FN and FP were set to 0.1 as used in [2].

In the simulation, the TA is not a target of a malicious node. If the TA is infected, trust decisions on network nodes will not be correct. It is assumed that CH and other nodes can be malicious nodes, based on the restriction that the malicious ratio is not more than 0.5. If the malicious node ratio is greater than 0.5, the malicious nodes can take control of all the opinions in the network and the trust decision cannot be determined correctly. Although it is assumed that the overall ratio of malicious nodes is less than 0.5, the proportion of malicious nodes in a cluster is not limited. Therefore, in some clusters, malicious nodes may not properly report to the TA because they have taken control of the cluster.

Figure 5 compares C_{total} of *GlobalTrust* and *CGTrust* based on β, where it can be observed that C_{total} of *CGTrust* decreases to less than 1/1000 compared to *GlobalTrust*. The results show that *CGTrust* can significantly reduce the required computations and thereby reduce the network's trust evaluation time. Figure 6 shows the

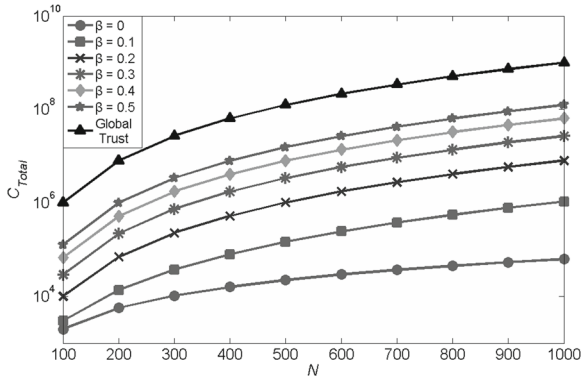

Fig. 5 C_{total} comparison of *GlobalTrust* and *CGTrust* based on N and β

detection error probability of FN and FP for *CGTrust*, where for the number of nodes of interest (i.e., $100 \leq N \leq 1000$), the probability of FN and FP are always lower than the upper bound 0.1. More significantly, the FN probability is always below 0.00012. In addition to being much faster than *GlobalTrust*, having a very low FN probability is a very advantageous feature of *CGTrust* because (compared to FP) FN is a significantly more critical security problem due to the fact that FN represents the probability that a malicious node has been not detected and still remains operational in the network. On the other hand, an erroneous FP decision on a node can be easily corrected by additional checking of the node. Figure 7 investigates influence of changes in malicious node ratios, where the results show that the FN probability of *CGTrust* is an approximate 0.1 times lower compared to *GlobalTrust*, while the FP probability of the two are similar. Abrupt changes in the performance can be observed in Figs. 4 and 5, which are due to the positive integer rounding effect applied to k_{optimal}.

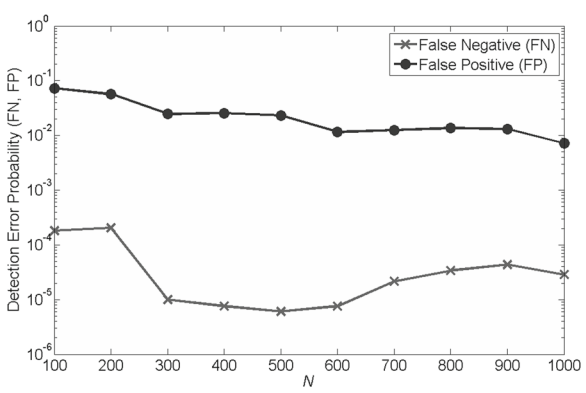

Fig. 6 Detection error probability (FN and FP) of *CGTrust*

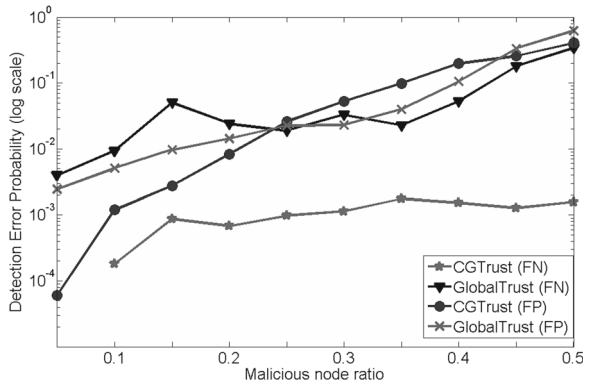

Fig. 7 Detection error probability (FN and FP) comparison of *GlobalTrust* and *CGTrust* based on various malicious node ratios

5 Conclusion

Mission supportive MANETs require fast updates on node conditions in order to properly support command and control operations. To support this objective, *CGTrust* was designed to minimize the time required to evaluate the trust profile of a MANET through optimal cluster size control applied to *GlobalTrust*. The simulation results show that for the number of nodes and malicious node ratios of practical interest (based on $\beta = 0.1$ and 0.5), *CGTrust* can be approximately 1000 and 10 times faster compared to *GlobalTrust*, respectively. In addition, the results also show that the FN probability is approximately 0.1 times lower when *CGTrust* is used instead of *GlobalTrust* for the malicious node ratio range of 0.05 to 0.5.

Acknowledgements
This work was supported by the ICT R&D program of MSIT/IITP, Republic of Korea (B0101-17-1276, Access Network Control Techniques for Various IoT Services).

Competing interests
The authors declare that they have no competing interests.

Publisher's Note

Author details
[1]School of Electrical and Electronic Engineering, Yonsei University, Seoul, Republic of Korea. [2]Republic of Korea Air Force, Gyeryong, Republic of Korea.

Received: 23 February 2017 Accepted: 25 August 2017
Published online: 18 September 2017

References

1. J-H Cho, A Swami, I-R Chen, A Survey on Trust Management for Mobile Ad Hoc Networks. IEEE Commun. Surv. Tutor. **13**(4), 562–583 (2011)
2. X Chen, J Cho, S Zhu, in *Proceedings of the IEEE International Conference on Sensing, Communication and Networking (SECON) 2014*. GlobalTrust: An Attack-Resilient Reputation System for Tactical Networks, (Singapore, 2014), pp. 275–283
3. K Aberer, Z Despotovic, in *Proceedings of the 2001 ACM International Conference on Information and Knowledge Management (CIKM)*. Managing trust in a peer-2-peer information system, (Atlanta, GA, USA, 2001), pp. 310–317
4. SD Kamvar, MT Schlosser, H Garcia-Molina, in *Proceedings of the 2003 ACM International Conference on World Wide Web (WWW)*. The eigentrust algorithm for reputation management in p2p networks, (Budapest, Hungary, 2003), pp. 640–651
5. L Xiong, L Liu, Peertrust: Supporting reputation-based trust for peer-to-peer electronic communities. IEEE Trans. Knowl. Data Eng. **16**(7), 843–857 (2004)
6. S Buchegger, JY Le Boudec, *A Robust Reputation System for Mobile ad hoc Networks*. Technical Report IC/2003/50, EPFL-DI-ICA, (Lausanne, 2003)
7. Q He, D Wu, P Khosla, in *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC) 2004*. Sori: a secure and objective reputation based incentive scheme for ad-hoc networks, (Atlanta, GA, USA, 2004), pp. 825–830
8. WL Teacy, J Patel, NR Jennings, M Luck, Travos: Trust and reputation in the context of inaccurate information sources. Auton. Agent Multi Agent Syst. **12**(2), 183–198 (2006)
9. A Jsang, R Ismail, in *Proceedings of the Electronic Commerce Conference 2002*. The beta reputation system, (Bled, Slovenia, 2002), pp. 2502–2511
10. H Chan, VD Gligor, A Perrig, G Muralidharan, On the distribution and revocation of cryptographic keys in sensor networks. IEEE Trans. Dependable Secure Comput. **2**(3), 233–247 (2005)
11. M Raya, MH Manshaei, M Felegyhazi, J-P Hubaux, in *Proceedings of ACM Conference on Computer and Communications Security (CCS) 2008*. Revocation games in ephemeral networks, (Alexandria, VA, USA, 2008), pp. 199–210
12. S Reidt, M Srivatsa, S Balfe, in *Proceedings of the ACM Conference on Computer and Communications Security (CCS) 2009*. The fable of the bees: incentivizing robust revocation decision making in ad hoc networks, (Chicago, IL, USA, 2009), pp. 291–302
13. X Chen, H Patankar, S Zhu, M Srivatsa, J Opper, in *Proceedings of the IEEE International Conference on Sensing, Communication and Networking (SECON) 2013*. Zigzag: Partial mutual revocation based trust management in tactical ad hoc networks, (New Orleans, LA, USA, 2013), pp. 131–139
14. S Kim, J-M Chung, Message Complexity Analysis of Mobile Ad Hoc Network Address Autoconfiguration Protocols. IEEE Trans. Mobile Comput. **7**(3), 358–371 (2008)