A Novel Approach to Trust Management in Unattended Wireless Sensor Networks

Yi Ren, *Member, IEEE,* Vladimir I. Zadorozhny, *Senior Member, IEEE,*
Vladimir A. Oleshchuk, *Senior Member, IEEE,* and Frank Y. Li, *Senior Member, IEEE*

Abstract—Unattended Wireless Sensor Networks (UWSNs) are characterized by long periods of disconnected operation and fixed or irregular intervals between sink visits. The absence of an online trusted third party implies that existing WSN trust management schemes are not applicable to UWSNs. In this paper, we propose a trust management scheme for UWSNs to provide efficient and robust trust data storage and trust generation. For trust data storage, we employ a geographic hash table to identify storage nodes and to significantly decrease storage cost. We use subjective logic based consensus techniques to mitigate trust fluctuations caused by environmental factors. We exploit a set of trust similarity functions to detect trust outliers and to sustain trust pollution attacks. We demonstrate, through extensive analyses and simulations, that the proposed scheme is efficient, robust and scalable.

Index Terms—Unattended wireless sensor network (UWSN), distributed trust management, subjective logic

1 Introduction

Wireless Sensor Networks (WSNs) have been used in challenging, hostile environments for various applications such as forest fire detection, battlefield surveillance, habitat monitoring, traffic management, etc. One common assumption in traditional WSNs is that a trusted third party, e.g., a sink, is always available to collect sensed data in a near-to-real-time fashion.

Although many WSNs operate in such a mode, there are WSN applications that do not fit into the real time data collection model. Consider an example of a monitoring system deployed in a natural park to detect poaching activities. The lack of regular access routes and the size of the surveillance area would require a mobile sink to collect data periodically [1]. Another example is an underwater mobile sensor network for submarine tracking and harbor monitoring. The inaccessibility of the protected area and other technical problems make it difficult to maintain continuous connections between sink and sensors [2]. Fig. 1 shows an example of Unattended WSNs (UWSNs) [1], [3]–[5] with a mobile sink visiting the network at either fixed or irregular intervals to collect data.

Trust management becomes very important for detecting malicious nodes in unattended hostile environments. It can also assist in secure routing [6], [7], secure data distribution [8], and trusted key exchange [9]. An efficient trust management system is required to handle trust related information in a secure and reliable way. It should deal with uncertainty caused by noisy communication channels and unstable sensor behavior.

We propose a trust management scheme for efficient trust generation as well as scalable and robust trust data storage in UWSNs. A central issue for trust management in UWSNs is how to store trust data without relying on a trusted third party. Initially, we consider two simple trust management schemes as a first-step attempt to address the existing trust storage problems in UWSNs. After analyzing the shortcomings of these simple schemes, we propose an advanced scheme based on a Geographic Hash Table (GHT) [10]. Our advanced scheme allows sensor nodes to *put* and *get* trust data to and from designated storage nodes based on node IDs. Sensor nodes do not need to know the IDs of storage nodes. They use a hash function to find locations of the storage nodes, which significantly reduce the storage cost. We also propose a set of similarity threshold functions to remove outliers from trust opinions. This prevents attackers from generating false trust opinions and from polluting trustworthiness. Furthermore, we provide a detailed analysis of the proposed scheme and conduct a comprehensive simulation-based study to demonstrate that our scheme is efficient, robust, and scalable.

The rest of the paper is organized as follows. Related work is reviewed in Section 2. Section 3 defines the network scenario, security model and design goals. Section 4 presents some background material on trust management in sensor networks and on subjective logic. Section 5 introduces our solutions for efficient trust data storage. Section 6 reports a simulation-based study conducted to evaluate the efficiency and the robustness of the proposed schemes. Section 7 considers advanced approaches to reliable trust generation. Section 8 offers conclusions.

- *Y. Ren is with the Department of Information and Communication Technology, University of Ager, Grimstad 4898, Norway, and also with the Department of Computer Science, National Chiao Tung University, Hsinchu, Taiwan. E-mail: yi.ren@uia.no.*
- *V. Oleshchuk and F. Li are with the Department of Information and Communication Technology, University of Ager, Grimstad 4898, Norway. E-mail: {vladimir.oleshchuk, frank.li}@uia.no.*
- *V. Zadorozhny is with the School of Information Sciences, University of Pittsburgh, Pittsburgh, PA 15260 USA. E-mail: vladimir@sis.pitt.edu.*

Manuscript received 30 Mar. 2012; revised 23 Dec. 2012; accepted 24 Jan. 2013. Date of publication 14 Feb. 2013; date of current version 2 July 2014. For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier 10.1109/TMC.2013.22

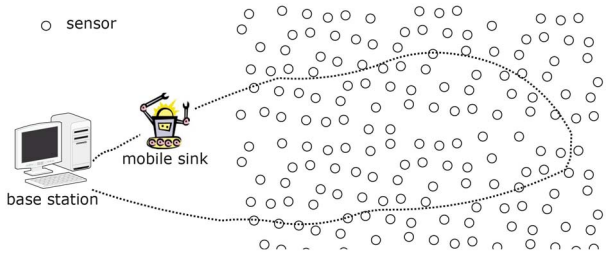


Fig. 1. Example of UWSN.

2 Related Work

In this section, we review the existing trust management schemes in WSNs, ad hoc and P2P networks.

2.1 Trust Management in WSNs

Several solutions have been recently proposed for trust management in WSNs. In [11] the authors designed a protocol to diagnose and mask arbitrary node failures in an event-driven WSN. In [12], the authors proposed a Bayesian trust management framework where each node maintains reputation metrics to assess past behavior of other nodes and to predict their future behavior. The authors in [13] proposed iTrust, an integrated trust framework for WSNs. A trust aware routing protocol for WSNs was proposed in [7]. The protocol exploits prior routing patterns and link quality to determine efficient routes. In [6], the authors proposed a trust-based routing scheme that selects a forwarding path based on the trust requirement of a packet and the trust level of neighbor nodes.

2.2 Trust Management in Ad Hoc Networks

More trust management studies were conducted in the field of ad hoc networks [14]–[17]. The authors in [14] proposed a reputation system based on Bayesian estimation of misbehavior in mobile ad hoc networks. The work in [15] introduced an information theoretic framework to measure trust and to model trust evolution. A data-centric framework for trust establishment was proposed in [16]. In [17], the authors proposed a distributed trust scheme based on distributed public key certificate management for mobile ad hoc networks.

2.3 Trust Management in P2P Networks

The authors in [18] proposed a Peer-Trust model based on public key infrastructure and trust propagation. PowerTrust [19], a robust and scalable P2P reputation scheme, was proposed to leverage the power-law feedback factors. In [20], the authors developed Credence, a decentralized object reputation and ranking system for P2P networks.

UWSNs are an emerging class of wireless networks [3]. The authors in [3] also defined a mobile adversary and proposed a set of schemes to neutralize attacks focusing on erasing data. Techniques providing forward secrecy and backward secrecy of data stored in sensors are explored in [8], [21], [22]. To the best of our knowledge, our earlier work [23] is the first study which proposed trust data storage and trustworthiness calculation to facilitate trust management in UWSNs. In this paper, we further proposed a set of schemes to mitigate trust pollution attacks based on subjective logic and various trust similarity measures.

Most of the trust management solutions developed for traditional WSNs, however, rely on the presence of an online trusted third party, e.g., to store and distribute trust data [6], [7], [11]–[13]. They cannot be applied directly to UWSNs due to the absence of the sink (or the base station). [24] is one exception that proposed a distributed scheme establishing reputation-based trust among sensor nodes. The authors anyhow did not consider significant trust attacks (as defined in Section 3.2) against the generated trust. The work in [10], [25] addressed data centric storage in WSNs, but trust management and security attacks are not considered.

To summarize, most schemes, e.g., [14]–[20], proposed for P2P and ad hoc networks are not suitable for UWSNs for the following reasons. First, UWSNs are more constrained with respect to computation, communication and power capabilities than P2P and ad hoc networks (although today's wireless sensors provide more options for storage capacity and computational capabilities, cheap hardware cost and light weight security solutions which lead to longer network lifetime are critical for UWSNs). Those schemes designed for P2P and ad hoc networks based on public key cryptography are therefore not suitable for UWSNs. Second, the number of nodes in ad hoc networks with typical applications like on-campus peer-to-peer communication among lap-tops/smart-phones is usually lower than in UWSNs which are more often targeted at environment monitoring applications. An UWSN is likely to have thousands of sensors. P2P networks may have more nodes than UWSNs, but the nodes in P2P networks do not have the same computational and energy constraints as in UWSNs. Finally, sensor nodes provide services throughout their whole lifetime, until their energy is depleted, while P2P nodes enter and exit the networks randomly.

3 Network Scenario, Security Model and Design Goals

3.1 Network Scenario

We consider an UWSN that consists of N sensor nodes, denoted as $s_j \in \mathbb{S}$, where $\mathbb{S} = \{s_j\}_{j=1}^{N}$. Each sensor s_j is located at point p_j and has a transmission range ϕ. Thus s_j at point p_j can communicate with s_m at point p_m if $\mathcal{D}(p_j, p_m) \leq \phi, (j, m \in \{1, \ldots, N\})$, where $\mathcal{D}(p_j, p_m)$ is the distance between p_j and p_m. Each sensor $s_j \in \mathbb{S}$ has n_j neighbors. We say that s_m is one of s_j's neighbors if $\mathcal{D}(p_j, p_m) \leq \phi$. The yellow (light shadow) points in the circle in Fig. 2 form a set of neighbors of s_j, $\mathcal{B}(s_j) = \{s | s \in \mathbb{S}$ and $\mathcal{D}(p_j, p) \leq \phi\}$. We assume that s_j's neighbors, $b_i \in \mathcal{B}(s_j)$, $i \in \{1, \ldots, n_j\}$, have their own trust opinions T_i^j regarding the trustworthiness Υ^{j1} of s_j, and are referred as *trust producers* of s_j. The nodes storing trust data are *trust managers* TM_j^r, $r \in \{1, \ldots, \alpha\}$ of s_j. Here α is the number of trust managers in the network. Those sensors that would like to know

1. In the context of this paper, trust opinion T is one sensor's conclusion about the trust level of another sensor; and trustworthiness Υ is a combination of trust opinions over time and across all involved sensors.

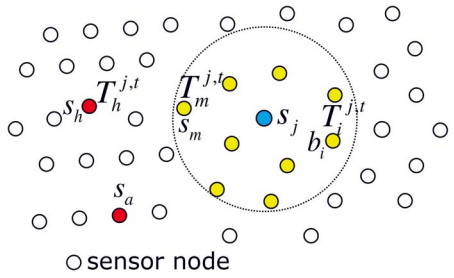


Fig. 2. Example of network topology.

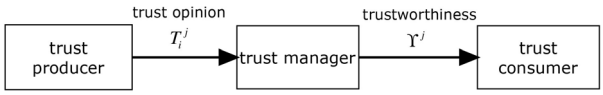


Fig. 3. Relationship between trust producer, trust manager and trust consumer.

about other sensors' trustworthiness are referred as *trust consumers*. The relationship between trust producer, trust manager and trust consumer is illustrated in Fig. 3.

We assume further that time is split into equal time intervals and that sensors maintain loosely synchronized clocks. At time interval t, s_j's neighbor b_i generates a trust opinion $T_i^{j,t}$ regarding s_j. Note that trust consumers can be anywhere in the network but trust producers are only within the transmission range of the corresponding sensor. Furthermore, there is a mobile sink visiting the network at either fixed or irregular time intervals to collect data from sensors.

3.2 Security Model

The UWSNs can be attacked in many ways. In this study, we focus on an adversary $\mathcal{ADV}$ launching attacks against trust data[2]. We divide the attacks into two categories: *trust eraser* and *trust pollution* attacks.

The effect of the trust eraser attack (denoted as $\mathcal{ADV}\_Del$) is that the trust data stored in sensors are lost and cannot be retrieved by trust consumers. For instance, $\mathcal{ADV}$ could try to compromise sensors and to erase the trust data stored in them. Moreover, when sensors are nonfunctional (e.g., due to energy depletion, natural disasters, etc.) their stored trust data are lost and are considered as non-recoverable in this study.

In case of trust pollution attack, $\mathcal{ADV}$ does not delete the trust data but rather *pollute* them. We consider the following pollution strategies:

- Environmental effect ($\mathcal{ADV}\_Noise$). Since sensors' trust opinions are generated based on sensors' previous behavior, they may generate some noise due to environmental effects.
- Homogeneous attack ($\mathcal{ADV}\_Homo$). Given a sensor s_j, $\mathcal{ADV}$ tries to increase s_j trustworthiness Υ^j monotonically or, it tries to decrease Υ^j monotonically. To do so, at each time interval $\mathcal{ADV}$ can compromise a subset of sensors to generate false trust opinions.
- Hybrid attack, denoted as $\mathcal{ADV}\_Hbd$. This attack is more severe since $\mathcal{ADV}$ aims to manipulate the trustworthiness, i.e., it is able to increase or decrease trustworthiness in any way.

For clarity, we assume that the number k of compromised sensors in each time interval is fixed. We refer to this number as the *compromising capability*. The compromised sensor can occur anywhere in the network.

3.3 Design Goals

Our trust management scheme is designed with the following goals in mind:

1) *Robustness*. The scheme remains functional even after certain amount of sensor nodes lose battery power or are physically damaged. Moreover, the trust data stored in the system should remain available to queries even if some sensor nodes fail due to the $\mathcal{ADV}\_Del$ attack.

2) *Resilience.* The generated trustworthiness Υ should be as close as possible to its real value even though $\mathcal{ADV}$ tries to inject false trust opinions in order to pollute it. In other words, the system should be resilient to $\mathcal{ADV}\_Noise$, $\mathcal{ADV}\_Homo$ and $\mathcal{ADV}\_Hbd$.

3) *Scalability*. The scheme should work for a very large unattended physical area with a high density of sensor nodes.

4) *Efficiency*. The designed trust management scheme should be efficient in terms of both communication cost and storage cost.

5) *Consistency*. Trust opinions generated by trust producers and trust queries sent from trust consumers should be routed correctly to trust managers where the trust data are stored.

3.4 Performance Metrics

The following metrics are defined to evaluate the performance of our scheme.

- $Pr[survival]_t^j$ is defined as the probability that at least one trust manager of s_j survives during time interval t.
- *Communication Cost* (denoted as $\mathcal{C}$): The communication cost consists of two parts: the cost of sending trust opinions to trust managers; and the cost of querying and retrieving trustworthiness stored in trust managers. $\mathcal{C}_j$ is defined as a communication cost of s_j. Since trust value queries and answers are short messages, we assume that sending and receiving a trust value message across each hop have the same cost, and that an approximate communication cost is $O(N)$ for message broadcast and $O(\sqrt{N})$ for point-to-point routing.
- *Storage Cost* (denoted as $\mathcal{S}$): $\mathcal{S}_j$ is the storage cost of storing the trust opinions and trustworthiness of s_j.

4 Preliminaries

4.1 Information Collection on Sensor Behavior

The information on a sensor's prior behavior is one of the most important aspects of trust management solutions [26]. This information varies from application to application. For

2. In this paper trust data means trust related data, such as opinions, trust measurement, etc., which are used by trust management scheme to make trust aware decisions.

example, it can be a watchdog mechanism that monitors the behavior of neighboring nodes. The work in [12] uses the Bayesian approach for assessing node reputation and trust evolution. Node capture attacks [27], where nodes are removed from the network for an indefinite amount of time, can be detected by their neighbors. One-shot probing is proposed in [28] to identify misbehaving nodes. The authors in [29] consider the trust inference problem as a shortest path calculation in a weighted directed graph. They utilize the theory of semirings for trust evaluation.

In this study, we assume that sensor nodes may use the analyses and scoring sensor trust approaches (e.g., [12], [27]–[29]) to generate trust opinions. That is, in a time interval t, s_j's neighbors, $\{b_i\}_{i=1}^{n_j}$, can generate trust opinions $T_i^{j,t}$ ($i \in \{1, ..., n_j\}$) regarding s_j, by monitoring s_j's prior behavior.

4.2 Subjective Logic

Monitoring sensor behavior in UWSNs based on previous communication patterns involves considerable uncertainty. Communication channels between sensors are unstable and noisy. To deal with this uncertainty, we adopt a Subjective Logic (SL) framework [30], and use SL *opinions* to assess trustworthiness. SL has two advantages: 1) it is lightweight; and 2) it takes both uncertainty and belief ownership into account. The definition of SL opinion is as follows.

Definition 1. *An opinion is a triplet, $T = \{B, D, U\}$, where $B, D, U \in [0, 1]$ and $B + D + U = 1$, and that B, D, and U correspond to belief, disbelief and uncertainty respectively.*

A trust level can be naturally defined using the SL opinions, e.g., $T_1 = \{0.0, 0.93, 0.07\}$ for low trust value and $T_2 = \{0.88, 0.0, 0.12\}$ for higher one.

Definition 2. *Let s_X, s_Y and s_Z be three sensors. Then $T_Y^X = \{B_Y^X, D_Y^X, U_Y^X\}$ and $T_Z^X = \{B_Z^X, D_Z^X, U_Z^X\}$ denote the opinions of s_Y and s_Z about the trustworthiness of s_X. Their combined consensus opinion is defined as $T_{Y,Z}^X = T_Y^X \oplus T_Z^X = \{B_{Y,Z}^X, D_{Y,Z}^X, U_{Y,Z}^X\}$ where $B_{Y,Z}^X = (B_Y^X U_Z^X + B_Z^X U_Y^X)/(U_Y^X + U_Z^X - U_Y^X U_Z^X)$, $D_{Y,Z}^X = (D_Y^X U_Z^X + D_Z^X U_Y^X)/(U_Y^X + U_Z^X - U_Y^X U_Z^X)$ and $U_{Y,Z}^X = (U_Y^X U_Z^X)/(U_Y^X + U_Z^X - U_Y^X U_Z^X)$.*

The trust value expressed as subjective opinions instead of one simple trust level provides a more flexible trust model of the real world. Therefore, according to Def. 2, the consensus of trust opinions generated by sensors $\{b_i\}_{i=1}^{n_j}$ in time interval t about sensor s_j is

$$T_1^{j,t} \oplus \cdots \oplus T_i^{j,t} \oplus \cdots \oplus T_{n_j}^{j,t} = T_{1,\cdots,i,\cdots,n_j}^{j,t} \quad . \tag{1}$$

Definition 3. *Let s_X and s_Y be two sensors. Then $\{T_Y^{X,t_1}, \ldots, T_Y^{X,t_n}\}$ denotes the opinion of s_Y about the trustworthiness of s_X for time intervals $\{t_1, \ldots t_n\}$ respectively, where $T_Y^{X,t_n} = \{B_Y^{X,t_n}, D_Y^{X,t_n}, U_Y^{X,t_n}\}$. Then s_Y's opinion about the trustworthiness of s_X on $t_1 \cup \cdots \cup t_n$ is defined as*

$$T_Y^{X,t_1 \cup \cdots \cup t_n} = \{B_Y^{X,t_1 \cup \cdots \cup t_n}, D_Y^{X,t_1 \cup \cdots \cup t_n}, U_Y^{X,t_1 \cup \cdots \cup t_n}\} \tag{2}$$

where $B_Y^{X,t_1 \cup \cdots \cup t_n} = \frac{1}{n}(B_Y^{X,t_1} + \cdots + B_Y^{X,t_n})$, $D_Y^{X,t_1 \cup \cdots \cup t_n} = \frac{1}{n}(D_Y^{X,t_1} + \cdots + D_Y^{X,t_n})$, $U_Y^{X,t_1 \cup \cdots \cup t_n} = \frac{1}{n}(U_Y^{X,t_1} + \cdots + U_Y^{X,t_n})$.

According to Def. 2 and Def. 3, we define trustworthiness Υ^j in terms of sensor consensus to combine trust opinions generated by sensors $\{s_{j,i}\}_{i=1}^{n_j}$ in time interval $\{t\}_{t=t_1}^{t_n}$ as

$$\Upsilon^j = T_{1,\cdots,i,\cdots,n_j}^{j,t_1 \cup \cdots \cup t_n} \quad . \tag{3}$$

The Υ^j can be calculated with respect to sensor consensus or time as follows: 1) with respect to *sensor consensus*: $\Upsilon^j = T_{1,\cdots,i,\cdots,n_j}^{j,t_1 \cup \cdots \cup t_n} = T_1^{j,t_1 \cup \cdots \cup t_n} \oplus \cdots \oplus T_i^{j,t_1 \cup \cdots \cup t_n} \oplus \cdots \oplus T_{n_j}^{j,t_1 \cup \cdots \cup t_n}$; and 2) with respect to *time*: $\Upsilon^j = T_{1,\cdots,i,\cdots,n_j}^{j,t_1 \cup \cdots \cup t_n} = \{B_{1,\cdots,i,\cdots,n_j}^{j,t_1 \cup \cdots \cup t_n}, D_{1,\cdots,i,\cdots,n_j}^{j,t_1 \cup \cdots \cup t_n}, U_{1,\cdots,i,\cdots,n_j}^{j,t_1 \cup \cdots \cup t_n}\}$, where $B_{1,\cdots,i,\cdots,n_j}^{j,t_1 \cup \cdots \cup t_n} = \frac{1}{n}(B_{1,\cdots,i,\cdots,n_j}^{j,t_1} + \cdots + B_{1,\cdots,i,\cdots,n_j}^{j,t_n})$, $D_{1,\cdots,i,\cdots,n_j}^{j,t_1 \cup \cdots \cup t_n} = \frac{1}{n}(D_{1,\cdots,i,\cdots,n_j}^{j,t_1} + \cdots + D_{1,\cdots,i,\cdots,n_j}^{j,t_n})$, and $U_{1,\cdots,i,\cdots,n_j}^{j,t_1 \cup \cdots \cup t_n} = \frac{1}{n}(U_{1,\cdots,i,\cdots,n_j}^{j,t_1} + \cdots + U_{1,\cdots,i,\cdots,n_j}^{j,t_n})$.

Remark. According to Def. 3, each trust opinion has the same impact over time. Meanwhile, it is more realistic to design the scheme to be time-aware such that the newer trust opinions have higher impact on the trustworthiness, while prior trust opinions should be also taken into account. A straightforward solution is to use a time factor (e.g., $f \in [0, 1]$) adding time impact into prior trust opinions, where greater f indicates newer opinion. More specifically, the time-aware trust opinion can be computed as $T_i^{j,t_1 \cup \cdots \cup t_n} = \{B_i^{j,t_1 \cup \cdots \cup t_n}, D_i^{j,t_1 \cup \cdots \cup t_n}, U_i^{j,t_1 \cup \cdots \cup t_n}\}$, where $B_i^{j,t_1 \cup \cdots \cup t_n} = \frac{1}{n}(f^{n-1} B_i^{j,t_1} + \cdots f B_i^{j,t_{n-1}} + B_i^{j,t_n})$, $D_i^{j,t_1 \cup \cdots \cup t_n} = \frac{1}{n}(f^{n-1} D_i^{j,t_1} + \cdots f D_i^{j,t_{n-1}} + D_i^{j,t_n})$, and $U_i^{j,t_1 \cup \cdots \cup t_n} = 1 - B_i^{j,t_1 \cup \cdots \cup t_n} - D_i^{j,t_1 \cup \cdots \cup t_n}$. For example, given $T_i^{j,t-1} = \{0.88, 0.3, 0.09\}$, $T_i^{j,t} = \{0.8, 0.38, 0.09\}$, and $f = 0.99$, $T_i^{j,t-1 \cup t}$ can be computed as $T_i^{j,t-1 \cup t} = \{\frac{1}{2}(0.99 \times 0.88 + 0.8), \frac{1}{2}(0.99 \times 0.03 + 0.38), 1 - \frac{1}{2}(0.99 * 0.88 + 0.8) - \frac{1}{2}(0.99 \times 0.03 + 0.38)\} = \{0.8712, 0.0297, 0.0991\}$.

However, we will consider this extension in our future work. The reasons are because that specifying a suitable value of f is not a trivial task and it needs to be further investigated. For example, given $f = 0.99$ or $f = 0.88$, it is not clear which one is more reasonable and how f varies over time. Due to page limit, we are not able to include any results on time-aware solutions in this paper. Indeed, this aspect is the focus in our more recent work on subjective logic based machine learning (Bayesian network) techniques for time-aware trustworthiness establishment. We refer interested readers to [30] for more details on subjective logic and to [31]–[33] for examples of the application of subjective logic in WSNs and social networks.

5 Efficient and Robust Storage of Trust Data

In traditional WSNs, a trusted third party, e.g., base station, is used to keep and calculate received trust opinions. The queries of sensors' trustworthiness are also sent to and answered by the base station. However, since UWSNs do not have a base station, trust opinions of sensors need to

be stored in sensors instead. Therefore, once sensor b_i generates an opinion $T_i^{j,t}$ at time interval t, it either stores $T_i^{j,t}$ locally or sends $T_i^{j,t}$ to other nodes.

Next we consider three trust data storage schemes without involving the base station. First, we introduce two basic schemes and discuss their shortcomings. Then we propose an advanced scheme to improve the basic schemes.

5.1 Basic Scheme I (SI) - Trust Data Local Storage

The main idea of the SI is to keep generated trust opinions locally, i.e., b_i generates $T_i^{j,t}$ and then stores $T_i^{j,t}$ in its own memory. In other words, b_i is not only one of s_j's trust producers but also one of s_j's trust managers.

The SI includes local storage of trust opinion, and trust opinion querying and calculation:

(1) *Local storage of trust opinion*. At every time interval, each sensor generates trust opinions about its neighbor nodes, combines it with previous trust opinions according to Eq. (2) and stores it locally. Note that the generated trust opinions are combined as a combined trust opinion resulting in very low storage cost. For instance, b_i generates T_i^{j,t_1}, T_i^{j,t_2} and T_i^{j,t_3} at t_1, t_2 and t_3, respectively, and stores the combined trust opinion in its memory as $T_i^j = T_i^{j,t_0 \cup t_1 \cup t_2 \cup t_3}$.

(2) *Trust opinion querying and calculation*. Consider the example in Fig. 2. Assume that sensor s_a wants to estimate the trustworthiness Υ^j of another sensor, s_j. It broadcasts a trust opinion request, $ASK(T_*^j)$, to ask sensors to collect opinions of other sensors about s_j. Here, we assume a suitable broadcast authentication protocol, e.g., multilevel μTESLA [34], for secure and reliable transmission of such broadcast values. If there is no direct relationship between two sensors (e.g., s_h and s_j), they have highest uncertain opinion score about each other's trustworthiness, i.e., $T_h^j = T_j^h = \{0, 0, 1\}$. Upon receiving $ASK(T_*^j)$, each sensor sends feedback messages, $ANS(T_*^j)$, to s_a if they have a direct relationship with s_j. Otherwise they just drop $ASK(T_*^j)$. Next, s_a combines received sensors' opinions using a consensus operator (Eq. (1)) to compute s_j's trustworthiness Υ^j, and stores the results.

Proposition 4. *In the Basic Scheme I, the probability that at least one trust manager node remains uncompromised within t time intervals is*

$$\begin{cases} Pr[survival]_t^j = 1, & k * t < n_j \\ Pr[survival]_t^j = 0, & k * t \geq n_j \end{cases}, \tag{4}$$

where n_j is the number of neighbor nodes and k is the compromising capability of $\mathcal{ADV}$ as defined in Section 3.2.

Proof. In SI, each sensor s_j has n_j trust managers and n_j trust producers in its transmission range. It is easy for $\mathcal{ADV}$ to find the trust managers in the transmission range of s_j. Within each time interval $\mathcal{ADV}$ compromises k sensors (trust managers) in s_j's transmission range. By the end of t-th time interval, $k * t$ sensors (trust managers) are compromised. Therefore, $k * t \geq n_j$ implies that all the trust managers are compromised, i.e., $Pr[survival]_t^j = 0$; otherwise $Pr[survival]_t^j = 1$. □

5.1.1 Communication Cost

Queries $ASK(T_*^j)$ are broadcast to all nodes at the cost of $O(N)$. Responses $ANS(T_*^j)$ are sent back to the trust consumer at the cost of $O(\sqrt{N})$ each. For t time intervals, the cost becomes $\mathcal{C}_j = O(tN) + O(tn_j\sqrt{N}) = O(t(N + n_j\sqrt{N}))$, where N is the number of sensors in the network.

5.1.2 Storage Cost

For a sensor s_j, each neighbor in its transmission range generates one trust opinion per time interval. As the generated trust opinions are combined over time, the storage cost for each neighbor is very low (i.e., $O(1)$). There are n_j neighbors that need to store the trust opinions regarding s_j at the cost of $O(n_j)$. That is, $\mathcal{S}_j = O(n_j)$.

Discussion. According to Proposition 4, $\mathcal{ADV}\_Del$ can compromise all the trust managers in a short time ($\geq \frac{n_j}{k}$). After that, both pre-compromised and post-compromised trust data can be deleted by $\mathcal{ADV}\_Del$. Therefore, we need to hide trust managers from $\mathcal{ADV}\_Del$. Next we propose Basic Scheme II that supports distributed trust data storage.

5.2 Basic Scheme II (SII) - Distributed Trust Data Storage

In order to address the shortcomings of the SI, we should ensure that: (1) a sensor s_j's trust producer and trust manager are not the same node; (2) $\mathcal{ADV}$ cannot easily find trust manager nodes; and (3) the scheme is resilient against node failures.

A straightforward solution would be to specify for each node a designated trust manager node that stores its trust data. The trust manager should not be one of the node's direct neighbors. The components of the SII scheme are defined as follows:

(1) *System initialization*. To provide trust data redundancy, at the beginning, each sensor s_j is associated with α randomly selected trust managers $\{TM_j^r\}_{r=1}^{\alpha}$. The IDs of those trust managers, $\{TM_j^r\}_{r=1}^{\alpha}$, are stored in the trust producers before deployment since the trust producers need to send the generated trust opinions to those trust managers, $\{TM_j^r\}_{r=1}^{\alpha}$. In addition, trust consumers store $\{s_j, \{TM_j^r\}_{r=1}^{\alpha}\}$ in their local memory so that trust consumers are able to retrieve s_j's trust data from $\{TM_j^r\}_{r=1}^{\alpha}$.

(2) *Trust opinion distributed storage*. After generating trust opinions about s_j, the trust producers of s_j send them to $\{TM_j^r\}_{r=1}^{\alpha}$. Note that, in every time interval, TM_j^r receives n_j trust opinions $\{T_i^{j,t}\}_{i=1}^{n_j}$ from $b_i \in \mathcal{B}(s_j)$ ($i \in [1, n_j]$). After receiving $\{T_i^{j,t}\}_{i=1}^{n_j}$, TM_j^r first removes outlier trust opinions as noise (we will further discuss this in Section 7). Then it combines the trustworthiness of previous time intervals with the received trust opinions according to Def. 3 and Def. 2 as follows: $\Upsilon_r^j = \Upsilon_r^{j,1\cup\cdots\cup t} = \Upsilon_r^{j,1\cup\cdots\cup t-1} \cup \Upsilon_r^{j,t} = \Upsilon_r^{j,1\cup\cdots\cup t-1} \cup (T_1^{j,t} \oplus \cdots \oplus T_{n_j}^{j,t})$, where $\Upsilon_r^{j,t}$ is the trustworthiness of s_j stored in TM_j^r during the time interval t.

(3) *Trustworthiness query and calculation*. Trust consumers send $ASK(T_*^j)$ to $\{TM_j^r\}_{r=1}^{\alpha}$ to retrieve trustworthiness $\{\Upsilon_r^j\}_{r=1}^{\alpha}$ from s_j's trust manager nodes. Upon receiving α trustworthiness, trust consumers remove outliers using the similarity threshold functions defined in Section 7

and compute the expected value of the rest of Υ^j as s_j's trustworthiness.

Proposition 5. *In the Basic Scheme II, the probability that at least one trust manager node is not compromised within t time intervals is*

$$Pr[survival]_t = 1 - \left(1 - \prod_{t=1}^{t}\left(1 - \frac{k}{N-(t-1)k}\right)\right)^{\alpha}. \quad (5)$$

Proof. Let $E_{t'}^r = 1$ denote the event of r-th trust manager compromised by $\mathcal{ADV}\_Del$ at time interval t', and $E_{t'}^r = 0$ denote the event of r-th trust manager survival within the time interval t'. The probability of no trust manager nodes surviving up to t is

$$\begin{aligned} &Pr[E_t^1 = 1 \cap \cdots \cap E_t^r = 1 \cap \cdots \cap E_t^\alpha = 1] \\ &\quad = Pr[E_t^1 = 1] * \cdots * Pr[E_t^r = 1] * \cdots * Pr[E_t^\alpha = 1] \\ &\quad = Pr[E_t^r = 1]^\alpha. \end{aligned}$$

Thus the probability of at least one trust manager node surviving up to t is $Pr[survival]_t = 1 - Pr[E_t^r = 1]^\alpha$.

The probability that r-th trust manager survives up to t is

$$\begin{aligned} Pr[E_t^r = 0] &= \left(1 - \frac{k}{N}\right)\left(1 - \frac{k}{N-k}\right)\left(1 - \frac{k}{N-2k}\right)\cdots \\ &\quad \cdots\left(1 - \frac{k}{N-(t-1)k}\right) \\ &= \prod_{t=1}^{t}\left(1 - \frac{k}{N-(t-1)k}\right). \end{aligned}$$

Thus, we have

$$\begin{aligned} Pr[survival]_t &= 1 - Pr[E_t^r = 1]^\alpha = 1 - (1 - Pr[E_t^r = 0])^\alpha \\ &= 1 - \left(1 - \prod_{t=1}^{t}\left(1 - \frac{k}{N-(t-1)k}\right)\right)^{\alpha}. \end{aligned}$$

□

5.2.1 Communication Cost

Once a trust opinion is generated, it is sent to and stored at the corresponding trust managers. The communication cost to store the trust opinion is $O(\sqrt{N})$. Since n_j trust producers need to send trust opinions to α trust managers at each round, the total cost is $O(tn_j\alpha\sqrt{N})$. Queries are sent to the designated node (trust manager), which also returns a response, causing a communication cost of $O(\sqrt{N})$. That is, $\mathcal{C}_j = O(tn_j\alpha\sqrt{N}) + O(2t\alpha\sqrt{N}) = O(tn_j\alpha\sqrt{N})$.

5.2.2 Storage Cost

For a sensor s_j, all its trust producers (its neighbors) have to know that a designated sensor TM_j is a corresponding trust manager node, i.e., for all sensors $b_i \in \mathcal{B}(s_j)$ $(i \in [1, n_j])$, they have to store the ID of TM_j at the cost of $O(\alpha n_j)$. In addition, any node in the network could be s_j's trust consumer, consequently every node has to know which node is s_j's trust manager. That is, every node in the network has to store the ID of s_j and the ID of its corresponding trust manager TM_j, which causes $O(N)$ of storage cost. Thus, $\mathcal{S}_j = O(\alpha(N+n_j))$.

Discussion. Once attacked by an $\mathcal{ADV}\_Del$, the trust data stored in sensors are deleted rather than polluted, so that a survived trust manager is able to report trust data. In order to compare SII with SI, we obtain numerical results of Eq. (4) and Eq. (5) using MATLAB, as illustrated in Fig. 4. We observe that SII is more robust than SI for all values of N, k, t and α, upon attacking by $\mathcal{ADV}\_Del$. Increasing the number of trust managers improves $Pr[survival]_t$ as well as trust data redundancy. However, as discussed above, the storage cost of SII is proportional to $\alpha(N + n_j)$, causing huge storage costs especially in large-scale UWSNs (see Fig. 8(a) and (b)). This limitation motivates us to design a more scalable scheme that can reduce storage cost caused by distributed storage while keeping the same $Pr[survival]_t$ as in SII.

5.3 Advanced Scheme (AS)

Our Advanced Scheme (AS) utilizes a hash-table-like interface of GHT [10] where nodes can *put* and *get* data based on their data type, i.e., $Put(DataType, DataValue)$ and $Get(DataType)$. Since a sensor ID is unique in the network, trust producers are able to *put* trust opinions to trust managers based on the ID, i.e., $Put(s_j, T_i^{j,t})$. Trust consumers are able to *get* trustworthiness from trust managers using the same sensor ID, i.e., $Get(s_j)$. In other words, trust opinions are pushed by, and stored at the same trust manager node. Meanwhile it enables trust consumers to pull trustworthiness from the trust manager nodes consistently. Neither trust producers nor trust consumers need to store the IDs of trust manager nodes, reducing storage cost significantly. Furthermore, the scheme should not be sensitive to node failures. That is, the scheme should be resilient to $\mathcal{ADV}\_Del$. Thus, trust opinions are pushed to α ($\alpha > 1$) trust manager nodes, whereas trust consumers pull trustworthiness from α trust managers. To do so, we modify the original basic operations of GHT from

$$\begin{cases} Put(DataType, DataValue) \\ Get(DataType) \end{cases} \text{ to } \begin{cases} Put(s_j, T_i^{j,t_l}, r) \\ Get(s_j, r) \end{cases} \quad \forall r \in [1, \alpha].$$

$Put(s_j, T_i^{j,t}, \alpha)$ is the function in which trust producer b_i is able to *put* its trust opinion $T_i^{j,t}$ regarding s_j to the r-th trust manager node of s_j, where α is the number of trust manager nodes specified by the mobile sink when the network is deployed. Using the $Get(s_j, r)$ function, trust consumers are able to *get* s_j's trustworthiness Υ^j from the r-th trust manager node of s_j. That is, each node has α trust manager nodes to store its trust opinions from its neighbors for data redundancy. Trust opinions regarding a sensor s_j are hashed by the sensor ID s_j to a geographical location. The node closest to the hashed geographical location is referred to as the trust manager node where data is sent to and retrieved from. Fig. 5 shows an example of a sensor ID s_j hashed to $\alpha = 3$ random geographical locations in the sensor network by using a secure hash function $\mathcal{L}_j^r = h^r(s_j) = h(h^{r-1}(s_j))$[3] $(\forall r \in \{1, \ldots, \alpha\})$; trust producers (e.g., b_i and b_m) and trust consumers (e.g., s_a) can send trust opinions and trust query requests to $\mathcal{L}_j^r$ using Greedy Perimeter Stateless Routing (GPSR) [35]. The closest node to the location $\mathcal{L}_j^r$, namely

3. Note that $\mathcal{L}_j^r$ is not the location of s_j but the location closest to the r-th trust manager node of s_j.

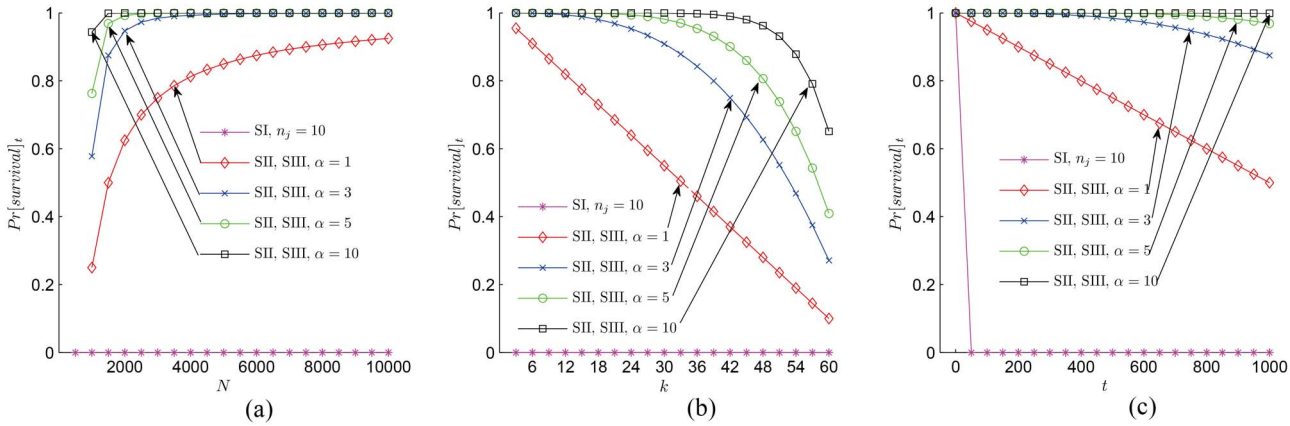


Fig. 4. Comparison of SI, SII and AS using Eq. (4) and Eq. (5) with default setting: $N = 10000$, $k = 5$ and $t = 150$. (a) $Pr[survival]_t$ vs. N. (b) $Pr[survival]_t$ vs. k. (c) $Pr[survival]_t$ vs. t.

trust manager (see $\{TM_j^r\}_{r=1}^3$ in Fig. 5), can receive the trust opinions and trust query requests. The AS includes the following phases:

(1) *System initialization*. Each node is preloaded with a secure hash function, denoted as $h(\cdot)$, and the redundancy factor α specified by the mobile sink depending on application scenarios. All nodes know their own locations, and the locations of the nodes which are one hop away.

(2) *Trust opinion storage based on GHT*. During the time interval t, and after $T_i^{j,t}$ is generated, b_i uses the function $Put(s_j, T_i^{j,t}, r)$ to *put* $T_i^{j,t}$ to α trust managers. In other words, b_i performs $h^r(s_j)$ to obtain $\mathcal{L}_j^1, \ldots, \mathcal{L}_j^\alpha$, and then sends $T_i^{j,t}$ to locations $\mathcal{L}_j^1, \ldots, \mathcal{L}_j^\alpha$ using GPSR, respectively. The closest node to location $\mathcal{L}_j^r$, denoted as TM_j^r, finally receives the trust opinion $T_i^{j,t}$ and is called the r-th trust manager node of s_j.

(3) *Trust opinion querying and calculation*. A trust consumer node, e.g., s_h, wants to know the trustworthiness of s_j. It uses the function $Get(s_j, r)$ $\forall r \in [1, \alpha]$ to *get* trustworthiness $\{\Upsilon_r^j\}_{r=1}^\alpha$ from s_j's α trust manager nodes. Similar as the *put* process, s_h performs $h^r(s_j)$ to obtain $\mathcal{L}_j^1, \cdots, \mathcal{L}_j^\alpha$, and sends $ASK(T_*^j)$ to locations $\mathcal{L}_j^1, \cdots, \mathcal{L}_j^\alpha$ using GPSR. The closest nodes to $\mathcal{L}_j^1, \cdots, \mathcal{L}_j^\alpha$, i.e., trust manager nodes $\{TM_j^r\}_{r=1}^\alpha$, finally receive $ASK(T_*^j)$ and then send $\{\Upsilon_r^j\}_{r=1}^\alpha$ to s_h.

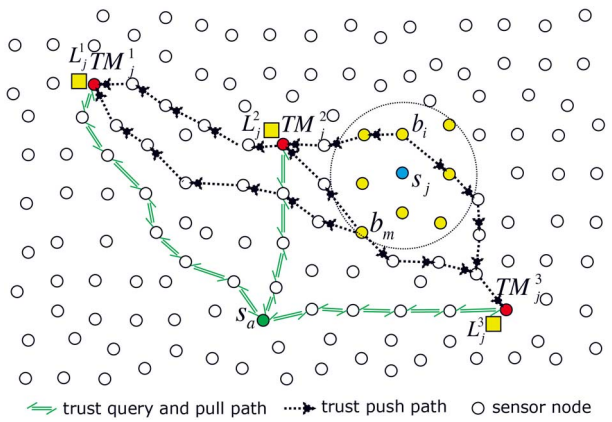


Fig. 5. Simple example of GHT techniques on UWSNs with $\alpha = 3$.

Proposition 6. *The Basic Scheme II and the Advanced Scheme have the same $Pr[survival]_t$. That is*

$$Pr[survival]_t = 1 - \left(1 - \prod_{t=1}^{t}\left(1 - \frac{k}{N-(t-1)k}\right)\right)^{\alpha}.$$

Proof. The same as for Proposition 5. The numerical results are shown in Fig. 4. □

6 Efficiency and Robustness Evaluation

In this section we conduct a set of simulations in MATLAB to show that AS has the strongest performance among these three schemes in terms of both efficiency and robustness. We consider an UWSN where 10000 nodes are randomly distributed in a 3000×3000 units area. The other parameters are set as follows. Each sensor has transmission range $\phi = 150$ units. $\mathcal{ADV}\_Del$ has compromising capability $k = 25$. The number of trust managers nodes $\alpha = 3$. The simulation results are averaged over 20 randomly deployed networks and are explained below.

Fig. 6(a)–(c) show the performance of t in terms of how many intervals the network can survive, given different α, k and ϕ. It demonstrates that SII and AS have better performance than SI does with respect to t for all values of α, k and ϕ. We observe in Fig. 6(a) that increasing α improves the performance of t. Meanwhile, increasing k decreases the performance of t. Fig. 6(c) shows that ϕ has no impact on SII and AS in terms of t but slightly increases the performance of t in SI.

Fig. 7(a)–(c) display the performance in terms of communication cost $\mathcal{C}$ for different α, k and ϕ. Distributed trust

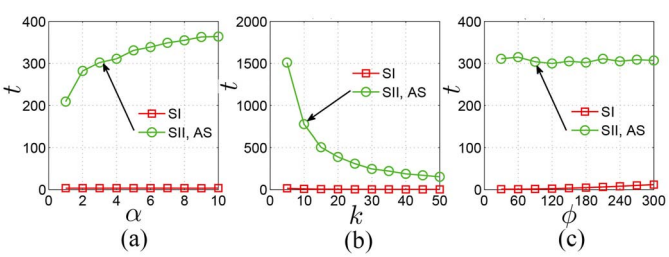


Fig. 6. Simulation results: $\alpha/k/\phi$ vs. t.

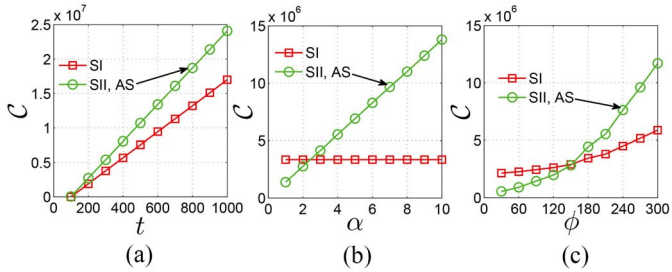


Fig. 7. Simulation results: $t/\alpha/\phi$ vs. $\mathcal{C}$.

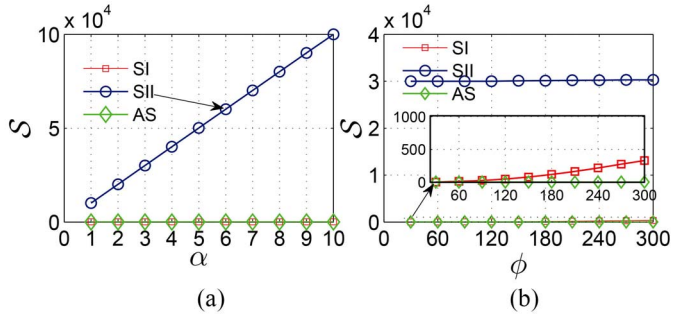


Fig. 8. Simulation results: α/ϕ vs. $\mathcal{S}$.

data storage is resilient to $\mathcal{ADV}\_Del$ and provides higher t. However, it causes higher communication costs. As shown in Fig. 7(b) and (c), the communication cost is acceptable if $\alpha \leq 3$ and $\phi \leq 120$.

As our simulation results in Fig. 8(a) and (b) indicate, SII has higher storage costs than SI and AS do in terms of α and ϕ. Meanwhile, ϕ has no impact on SII in terms of $\mathcal{S}$ but slightly increases $\mathcal{S}$ of SI and AS. α does not have any influence on SI and AS with respect to $\mathcal{S}$ but significantly increases $\mathcal{S}$ for SII.

Discussions.

(1) Robustness. Trust opinions $\{T_i^{j,t}\}_{i=1}^{n_j}$ generated for each time interval are routed to and stored in α trust managers. Therefore, the trustworthiness can be retrieved even if up to $\alpha-1$ trust managers are lost. That is, the SII and AS are resilient to $\mathcal{ADV}\_Del$, as shown in Fig. 6(a)–(c). In addition, those figures also show that SII and AS have the same probability of at least one trust manager survival, which is coincident with the analysis of Proposition 6.

(2) Efficiency. SII and AS have the same communication cost. In addition, AS has a much lower storage cost compared with SII. Unlike in SII, trust producers do not need to store the IDs of the trust managers of a sensor s_j due to the nature of *put* and *get* function of GHT. For α trust managers, the total storage cost is $O(\alpha(N+n_j))$ for SII and $O(\alpha)$ for AS. Thus, AS is more efficient than SII.

(3) Scalability. As discussed above, the storage cost of AS does not depend on the number of sensors, N. Increasing N does not increase storage costs in AS. Moreover, Section 5.2.1 demonstrates that the communication cost of AS is proportional to $\sqrt{N}$. For example, when the number of sensors N increases 10 times from 1000 to 10000, we observe only a 3 times increase in communication costs.

(4) Consistency. For a given sensor, s_j, all generated trust opinions $\{T_i^{j,t}\}_{i=1}^{n_j}$ are routed to $\{\mathcal{L}_j^r\}_{r=1}^{\alpha}$, respectively. The nodes closest to $\{\mathcal{L}_j^r\}_{r=1}^{\alpha}$ receive $\{T_i^{j,t}\}_{i=1}^{n_j}$ and store them in their local memory, where $\mathcal{L}_j^r = h^r(s_j)$. Since sensor ID is unique in the network for a given sensor s_j, the generated hash values $\{\mathcal{L}_j^r\}_{r=1}^{\alpha}$ are also unique in the network due to the one-way property of the hash function. As a consequence, all the trust producers and trust consumers are able to find the correct trust managers to store and query trust data.

7 Trustworthiness Generation

Through the simulations and discussions in the previous section, we have demonstrated that AS significantly reduces storage cost caused by distributed data storage and provides resilience to $\mathcal{ADV}\_Del$. In this section, we continue to investigate the performance of the proposed schemes against trustworthiness pollution attacks (i.e., $\mathcal{ADV}\_Noise$, $\mathcal{ADV}\_Homo$ and $\mathcal{ADV}\_Hbd$) defined in Section 3.2.

Initially, for each sensor s_j, trust opinion $T_i^{j,0}$ could be set by a mobile sink based on such information as physical protection, location, or role of the nodes. For example, a node, s_j, buried under the ground has higher $\{T_i^{j,0}\}_{i=1}^{n_j}$ than the exposed ones. After the trust opinions are generated, a trust producer b_i is able to update T_i^j using a conjunction operation (Eq. (2)), and trust consumers can calculate the trustworthiness Υ^j using consensus operation (Eq. (1)).

In the following, we conduct a simulation-based study of the trust resilience of the proposed schemes against pollution attacks. We assume that uncompromised sensor nodes generate correct trust opinions $cT^j = \{cB^j, cD^j, cU^j\}$, while compromised sensors generate false trust opinions $fT^j = \{fB^j, fD^j, fU^j\}$, where cB and fB denote *correct belief* and *false belief*. Similarly, cD, cU, fD, and fU denote correct disbelief, correct uncertainty, false disbelief, and false uncertainty, respectively. We adopt normal distribution to generate trust opinions, i.e., $B \sim \mathcal{N}(E(B), \sigma^2)$, where $E(B)$ is the expected value of B and σ is the standard deviation of B. In order to compare the impact of false trust opinions fT, in the simulations, fT values are generated after the 20th time interval so that we can observe whether there are any differences before and after the 20th time interval on the trustworthiness of a node.

We will use $d(B_i^j, E(B_m^j)) = \sqrt{(B_i^j - E(B_m^j))^2}$ to denote the Euclidean distance between B_i^j and its expected value $E(B_m^j)$ where $j \in \{1, \ldots, N\}$ and $i, m \in \{1, \ldots, n_j\}$.

7.1 Trust Consensus only Approach (TC-ONLY)

In this subsection, we consider the TC-ONLY approach, which we used in Section 5, as our baseline. This approach generates trustworthiness based on trust consensus only as in Eq. (2) and Eq. (3). Through simulation results, we first demonstrate that it is resilient to $\mathcal{ADV}\_Noise$, and then reveal its shortcomings with respect to $\mathcal{ADV}\_Homo$ and $\mathcal{ADV}\_Hbd$.

7.1.1 Trust Resilience against $\mathcal{ADV}\_Noise$

To evaluate the performance of the proposed scheme with respect to environmental effect ($\mathcal{ADV}\_Noise$), i.e., $d(a_i^j, E(a_m^j)) \approx 0$ ($a \in \{B, D, U\}$), we set the correct trust

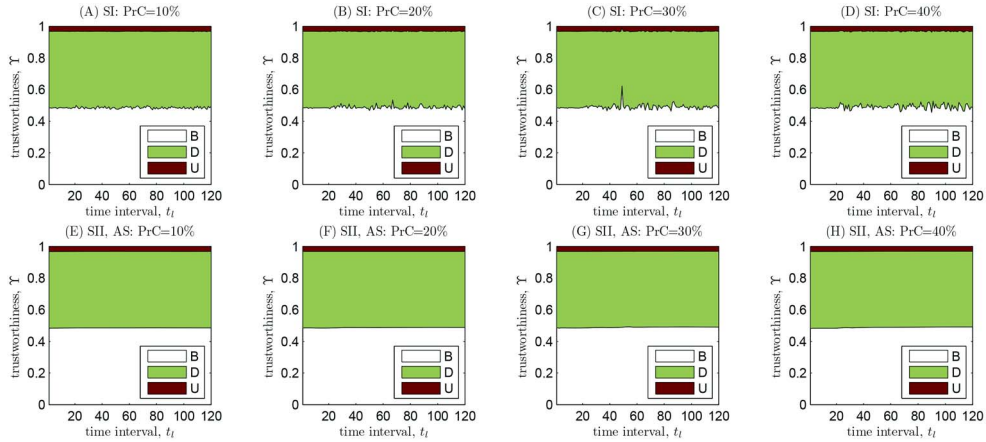


Fig. 9. Consensus is enough, $cT = \{0.3, 0.3, 0.4\}$, $\sigma_c = 0.01$, $fT = \{0.3, 0.3, 0.4\}$, $\sigma_f = 0.1$.

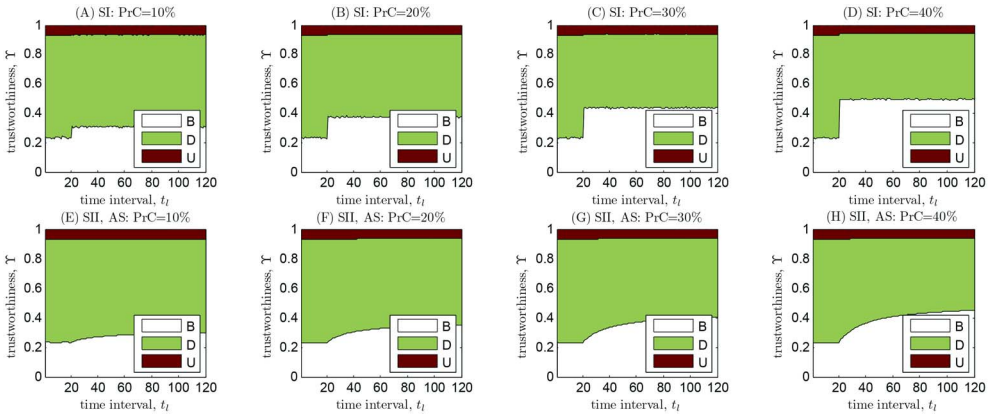


Fig. 10. Example of when $\mathcal{ADV}$ tries to increase Υ^j, $cT = \{0.1, 0.3, 0.6\}$, $fT = \{0.4, 0.1, 0.5\}$, $\sigma_c = \sigma_f = 0.01$.

opinions as $cT = \{0.3, 0.3, 0.4\}$[4] and $\sigma_c = 0.01$, where $cB, cD \sim \mathcal{N}(0.3, 0.001)$, and $cU = 1 - cB - cD$. In order to monitor environmental effect, we set a certain percentage PrC of sensors to generate trust opinions with larger $\sigma_f = 0.1$.

Fig. 9 shows the simulation results of SI, SII and AS when different values of PrC (from 10% to 40%) are specified. These figures display three elements B, D and U in white, green (light shadow), and red (black) respectively. The first row of Fig. 9 is the simulation results of SI. As one can observe that, after the 20th time interval, the obtained trustworthiness Υ starts to become unstable. In addition, increasing the percentage PrC of anomalous sensors makes Υ more unstable. The second row of Fig. 9 shows the results for the SII and AS schemes. It is interesting to emphasize that Υ is very smooth for all values of PrC. The anomalous trust opinions have almost no influence on Υ. We observe a slight increase Υ when $PrC = 40\%$ in Fig. 9(h). This is because that the consensus operation reduces trust uncertainty. Comparing the first row of Fig. 9 and the second row of Fig. 9, it is easy to see that SII and AS are more resilient against $\mathcal{ADV}\_Noise$ than SI is. Therefore, trust consensus improves resilience against $\mathcal{ADV}\_Noise$, i.e., $d(a_i^j, E(a_m^j)) \approx 0$ ($a \in \{B, D, U\}$).

4. Our simulations are conducted using random trust values and distribution paremeters. To exhibit the impact of ADV's attacks as clear as possible, we select suitable values (e.g., cT={0.3,0.3,0.4}, σ_c = 0.01, etc.) to plot simulation results (figures). The same simulation parameter configuration applies to the rest of the paper.

7.1.2 Trust Resilience against $\mathcal{ADV}\_Homo$

Since $\mathcal{ADV}\_Homo$ tries to either increase Υ or decrease Υ monotonously, we conduct two simulations. In the first simulation, $\mathcal{ADV}\_Homo$ is assumed to generate false trust opinions fT to increase Υ. In contrast, $\mathcal{ADV}\_Homo$ is set to decrease Υ in the second simulation.

Simulation one. In order to increase Υ, $\mathcal{ADV}\_Homo$ increases B and decreases D simultaneously. That is, generate fT that satisfies $E(cB) < E(fB)$ and $E(cD) > E(fD)$. We select a special case when $cT = \{0.1, 0.3, 0.6\}$, $fT = \{0.4, 0.1, 0.5\}$ and $\sigma_c = \sigma_f = 0.01$. The simulation results are shown in Fig. 10. The same as in the other figures, the results for SI are plotted in the first row, while the SII and AS results are shown in the second row. It is interesting to see that the results of SI experience *sharp* steps and jitters after the 20th time interval. Those sharp steps and jitters indicate that SI is not resilient to $\mathcal{ADV}\_Homo$ attacks. In contrast, the results of SII and AS are *smoother* compared with that of SI. The smoother result means that trust consensus does effectively mitigate the effect of $\mathcal{ADV}\_Homo$. In addition, when PrC increases, Υ starts to increase. The reason is that more sensors generate false trust opinions, increasing the impact of false trust opinion fT on trustworthiness Υ.

Simulation two. To decrease Υ as much as possible, $\mathcal{ADV}\_Homo$ decreases B and increases D simultaneously.

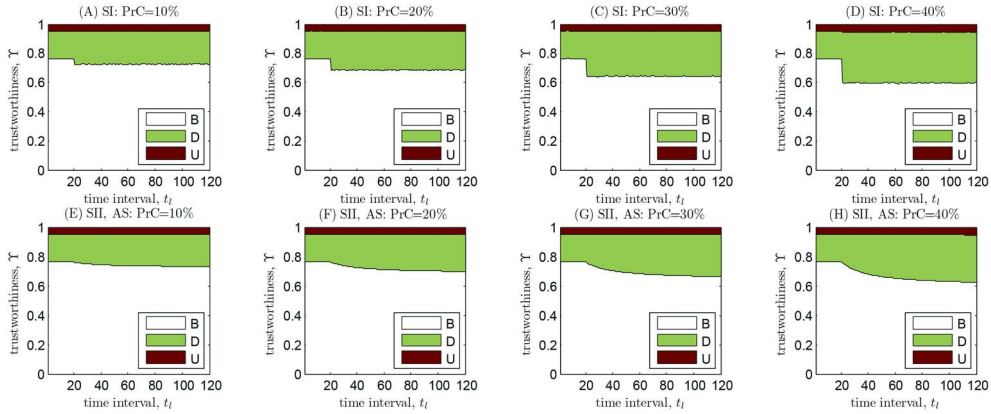


Fig. 11. Example of when $\mathcal{ADV}$ tries to decrease Υ^j, $cT = \{0.4, 0.1, 0.5\}$, $fT = \{0.1, 0.3, 0.6\}$, $\sigma_c = \sigma_f = 0.01$.

That is, $E(cB) > E(fB)$ and $E(cD) < E(fD)$. We set $cT = \{0.4, 0.1, 0.5\}$, $fT = \{0.1, 0.3, 0.6\}$, $\sigma_c = \sigma_f = 0.01$ in the simulation. We observe, in Fig. 11, that trust consensus does not mitigate *ADV_Homo*. After the 20th time interval, Υ starts to decrease sharply. Similar to simulation one, increasing *PrC* has heavier influence on Υ, and SII and AS have better performance than SI does.

Through the simulation results shown above, we conclude that TC-ONLY is not resilient against *ADV_Homo* attack.

7.1.3 Trust Resilience against $\mathcal{ADV}$_Hbd

Recall that *ADV_Hbd* aims to manipulate trustworthiness Υ, it is able to increase or decrease Υ in any way. However, as shown in Figs. 10 and 11, trust consensus does not perform well in either increasing Υ or decreasing Υ attacks.

Discussion. From the simulation results shown above, it is easy to conclude that TC-ONLY is not enough for trustworthiness calculation. It can only sustain *ADV_Noise* caused by environmental effects. The reason is that using trust consensus for trust calculation decreases uncertainty U and makes Υ stable. However, it does not performance well against *ADV_Homo* and *ADV_Hbd*. The reason is that both correct trust opinions cT and false trust opinions fT are taken into account in trustworthiness calculation as input, resulting in polluted Υ. One way to solve this problem is to reduce the effect of fT as much as possible. Thus we propose the next scheme capable of removing false trust opinions.

7.2 Trust Consensus with One Parameter Similarity Threshold Function (ONE-PARA)

As compromised trust producers may send false trust opinions to trust managers to *pollute* trustworthiness, we use ONE-PARA to remove outliers. A one parameter similarity threshold function is defined as

$$ST(T_i^j) = \frac{\sqrt{(B_i^j - E(B_m^j))^2}}{B_i^j E(B_m^j)}, \quad (6)$$

where B_i^j and B_m^j are the belief values regarding s_j generated by its neighbors s_i and s_m respectively. As a consequence, any T_i^j is considered as a outlier if $ST(T_i^j) > \epsilon$ where ϵ is a similarity threshold factor (e.g., $\epsilon = 0.1$). The similarity threshold function is expected to neutralize false trust opinions as much as possible. It is also desirable to reduce false positives (when trust opinions are considered as false trust opinions even though they are correct), as well as false negatives (when trust opinions are considered as correct trust opinions even though they are false). Therefore, we aim to increase true positives as much as possible while keeping very few false positives and false negatives. However, since the threshold function is based on how *far* B_i^j is from its expected value $E(B_m^j)$, and $E(B_m^j)$ is the average value of both *cB* and *fB*, the selection of ϵ may be problematic.

As shown in Fig. 12, decreasing ϵ increases true positives. However, it also increases false positives. Fig. 12(a) shows that a major part of *fB* and a small part of *cB* are considered to be outliers. A small part of false trust opinions are considered to be correct trust opinions (false negative), if a suitable similarity threshold factor ϵ is specified. When ϵ is too small, as shown in Fig. 12(b), all false trust opinions are considered to be outliers. However, more than half of the correct trust opinions are also considered to be outliers. In contrast, when ϵ is too large, more than half of the false

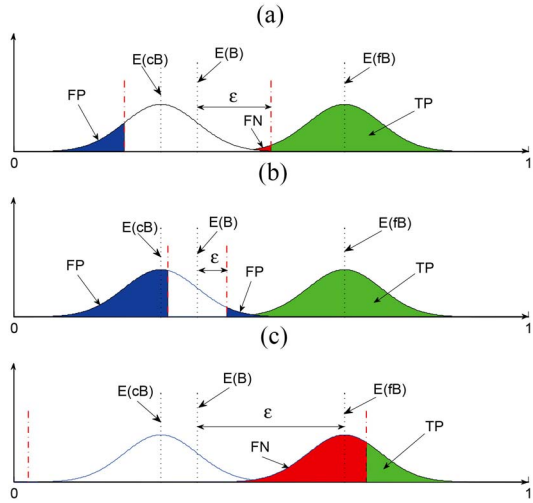


Fig. 12. Example of similarity threshold factor selection in terms of false positive (FP), true positive (TP) and false negative (FN). (a) Suitable similarity threshold factor ϵ. (b) Similarity threshold factor ϵ is too small. (c) Similarity threshold factor ϵ is too large.

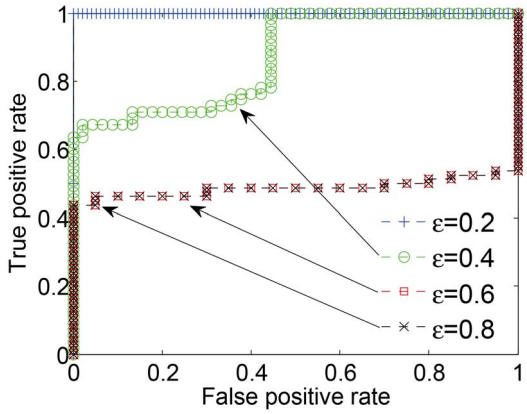


Fig. 13. ROC curve with $cT = \{0.1, 0.3, 0.6\}$, $fT = \{0.4, 0.1, 0.5\}$, $\sigma_c = \sigma_f = 0.01$ where similarity threshold factor ϵ are specified as 0.2, 0.4, 0.6 and 0.8, respectively.

trust opinions are considered to be correct trust opinions. In addition, the larger the number of false trust opinions considered to be correct trust opinions, the closer $E(B_m^j)$ is to $E(fB)$.

Fig. 13 shows the Receiver Operating Characteristic (ROC) curve with $cT = \{0.1, 0.3, 0.6\}$, $fT = \{0.4, 0.1, 0.5\}$, $\sigma_c = \sigma_f = 0.01$ where similarity threshold factor ϵ is specified as 0.2, 0.4, 0.6 and 0.8 respectively. We observe that true positives increase as false positives increase, and that the ROC curves are the same when $\epsilon = 0.6$ and 0.8. That is the reason why Fig. 14(d), (i), (e), and (j) indicate the same level of performance.

7.2.1 Trust Resilience against $\mathcal{ADV}$_Noise

Since ONE-PARA is an improved version of TC-ONLY, it works well against $\mathcal{ADV}$_*Noise* and can further filter outliers.

7.2.2 Trust Resilience against $\mathcal{ADV}$_Homo

As demonstrated above, TC-ONLY cannot sustain $\mathcal{ADV}$_*Homo*. We conduct two more sets of simulations and compare it with ONE-PARA. We use the parameters of the last two simulations of TC-ONLY with a fixed percentage of compromised sensors $PrC = 20\%$[5].

The results are as shown in Figs. 14 and 15. The first rows and the second rows are the simulation results of SI and that of SII and AS, respectively. We observe that, due to trust consensus, the generated Υ in SII and AS is more stable than Υ in SI. The first column of Figs. 14 and 15 shows the results when the threshold function is not applied (TC-ONLY). The rest of the columns of those figures are the results when the threshold function (ONE-PARA) is applied. Here similarity threshold factor ϵ varies from 0.2 to 0.8. We notice that TC-ONLY (the first column of Figs. 14 and 15) do not sustain $\mathcal{ADV}$_*Homo* and $\mathcal{ADV}$_*Hbd* as expected. Consider now the simulation results of SII and AS (columns 2,3,4,5 in Figs. 14 and 15):

- Proper ϵ. We find that ONE-PARA works well (see Fig. 14(i) and (j) as well as Fig. 15(h) and (i)) when a suitable ϵ is selected.
- ϵ is too small. It is interesting to observe that uncertainty U becomes higher while belief B and disbelief D decrease in the second column of Figs. 14 and 15. This is because that a very small similarity threshold factor treats both false and correct trust opinions as outliers. When ϵ is too small, as illustrated in Fig. 12(b), more than half of the correct trust opinions are considered as outliers, causing the bizarre behavior in the second column of Fig. 14.
- ϵ is too large. As shown in Fig. 12(c), false negatives increase with a larger ϵ. That is, more false trust opinions are considered as correct trust opinions, resulting in higher values in disbelief D and lower values in belief B, as shown in Fig. 15(e) and (j).

Discussion. As shown in Figs. 14 and 15, ONE-PARA works well in some special cases (see Table 1). However, when $d(B_i^j, E(B_m^j)) \approx 0$ and $d(D_i^j, E(D_m^j))$ are high, ONE-PARA cannot identify the difference between disbelief D, such as correct trust opinion $cT = \{0.1, 0.1, 0.8\}$ and false trust opinion $fT = \{0.1, 0.8, 0.1\}$, or $cT = \{0.1, 0.8, 0.1\}$ and $fT = \{0.1, 0.1, 0.8\}$. As shown in Fig. 16, when $cB = fB = 0.1$, $cD = 0.1$ and $fD = 0.8$, the differences in disbelief D cannot be detected by ONE-PARA, which may cause severe consequences. $\mathcal{ADV}$'s attack pollutes the trustworthiness for all schemes. We observe in Fig. 17 that $\mathcal{ADV}$'s attack is mitigated by the consensus when $d(B_i^j, E(B_m^j)) \approx 0$ and $E(cB) > E(fB)$. Furthermore, ONE-PARA can only specify one parameter among the three elements B, D and U so that it cannot sustain $\mathcal{ADV}$_*Hbd*. The reason is that the most efficient way for $\mathcal{ADV}$ to increase Υ is to increase B and to decrease D simultaneously. The threshold functions based on one parameter cannot control these two parameters, i.e., B and D. To mitigate the pollution caused by $\mathcal{ADV}$_*Hbd* for the case when $d(B_i^j, E(B_m^j)) \approx 0$ and $E(cB) < E(fB)$, we propose our next scheme.

7.3 Trust Consensus with Three Parameter Similarity Threshold Function (T-PARA)

Since ONE-PARA cannot recognize the difference between $cT = \{0.1, 0.1, 0.8\}$ and false trust opinion $fT = \{0.1, 0.8, 0.1\}$, we formulate a three-parameter similarity threshold function as follows,

$$ST(T_i^j) = \frac{\sqrt{(B_i^j - E(B_m^j))^2 + (D_i^j - E(D_m^j))^2 + (U_i^j - E(U_m^j))^2}}{B_i^j E(B_m^j) + D_i^j E(D_m^j) + U_i^j E(U_m^j)}. \quad (7)$$

7.3.1 Trust Resilience against $\mathcal{ADV}$_Noise

As an improved version of TC-ONLY, T-PARA exhibits better performance. Here, we do not show the simulation results with respect to $\mathcal{ADV}$_*Noise* due to page limit.

7.3.2 Trust Resilience against $\mathcal{ADV}$_Homo and $\mathcal{ADV}$_Hbd

We use the same simulation parameters as in ONE-PARA experiments. Figs. 18 and 19 show the simulation results.

5. Please note that the percentage of compromised sensors can be configured randomly in the range of [0,50%). Due to page limit, we only illustrate the results with 20% compromised nodes.

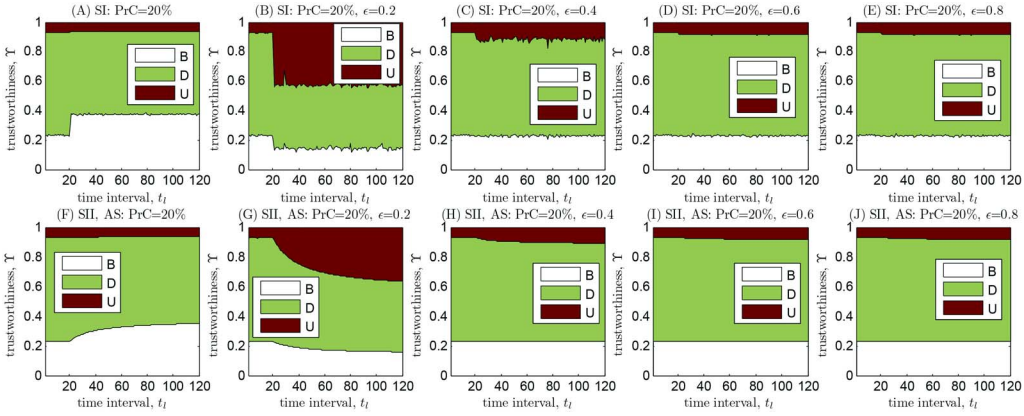


Fig. 14. Example of when $\mathcal{ADV}$ tries to increase Υ^j, $cT = \{0.1, 0.3, 0.6\}$, $fT = \{0.4, 0.1, 0.5\}$, $\sigma_c = \sigma_f = 0.01$.

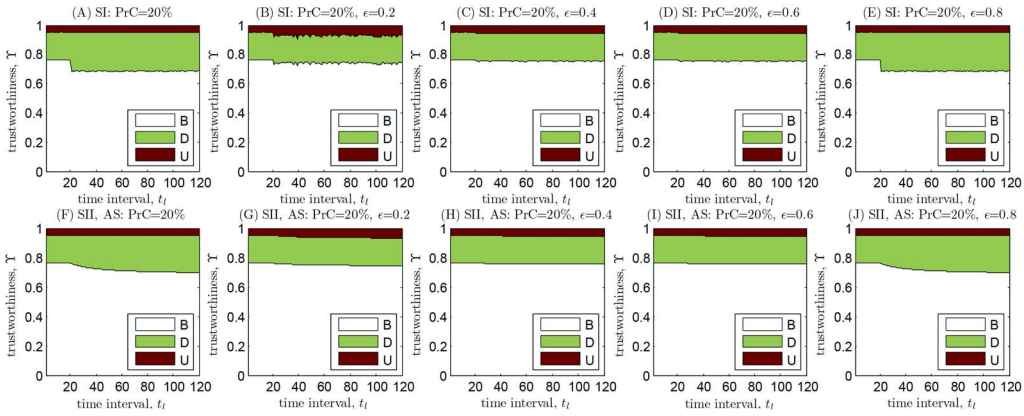


Fig. 15. Example of when $\mathcal{ADV}$ tries to decrease Υ^j, $cT = \{0.4, 0.1, 0.5\}$, $fT = \{0.1, 0.3, 0.6\}$, $\sigma_c = \sigma_f = 0.01$.

The results based on TC-ONLY are plotted in the first column.

As one can observe, the performance of T-PARA is much better in comparison with TC-ONLY when a suitable similarity threshold factor ϵ is specified (see Fig. 18(h)–(j) as well as Fig. 19(h)–(j)). In addition, we observe that T-PARA works well when $\epsilon = 0.4, 0.6$ and 0.8 (see Fig. 18(h)–(j)), while the performance of ONE-PARA is not good (Fig. 16). Moreover, Fig. 18(b) and (g) as well as Fig. 19(b) and (g) show the impact of ϵ when it is too small. Furthermore, it is worth mentioning that T-PARA performs well (see Fig. 18(h)–(j) as well as Fig. 19(h)–(j)) in terms of both $\mathcal{ADV}\_Homo$ increasing Υ attack and $\mathcal{ADV}\_Homo$ decreasing Υ attack. Therefore, it is resilient to $\mathcal{ADV}\_Hbd$.

7.4 Three Parameters with Weighted Factors (T-PARA-WF)

In order to provide a more flexible threshold function to prevent $\mathcal{ADV}$ from pollution attacks, we further develop an improved version of Eq. (7):

$$ST(T_i^j) = \frac{\sqrt{x^2(B_i^j - E(B_m^j))^2 + y^2(D_i^j - E(D_m^j))^2 + z^2(U_i^j - E(U_m^j))^2}}{xB_i^j E(B_m^j) + yD_i^j E(U_m^j) + zU_i^j E(U_m^j)}, \tag{8}$$

where $xB + yD + zU = 1$.

We introduce three weighted factors x, y and z into Eq. (7), enabling a T-PARA-WF method that can be adjusted depending on different scenarios. For example, to prevent $\mathcal{ADV}$ from increasing trustworthiness, we can define a

TABLE 1
Impacts of Consensus and Threshold Functions with Respect to $d(B_i^j, E(B_m^j))$, $d(D_i^j, E(D_m^j))$ and $d(U_i^j, E(U_m^j))$

$d(B_i^j, E(B_m^j))$		$d(D_i^j, E(D_m^j))$		$d(U_i^j, E(U_m^j))$	Threshold Function: TC-ONLY	ONE-PARA (Eq. (6))	T-PARA (Eq. (7))	T-PARA-WF (Eq. (8))
≈ 0		≈ 0		≈ 0	good	good	good	good
≈ 0		high	$E(cD) > E(fD)$	-	good	good	good	good
≈ 0		high	$E(cD) < E(fD)$	-	not enough	not enough	good	good
high	$E(cB) > E(fB)$	-		-	not enough	good	good	good
high	$E(cB) < E(fB)$	-		-	not enough	good	good	good

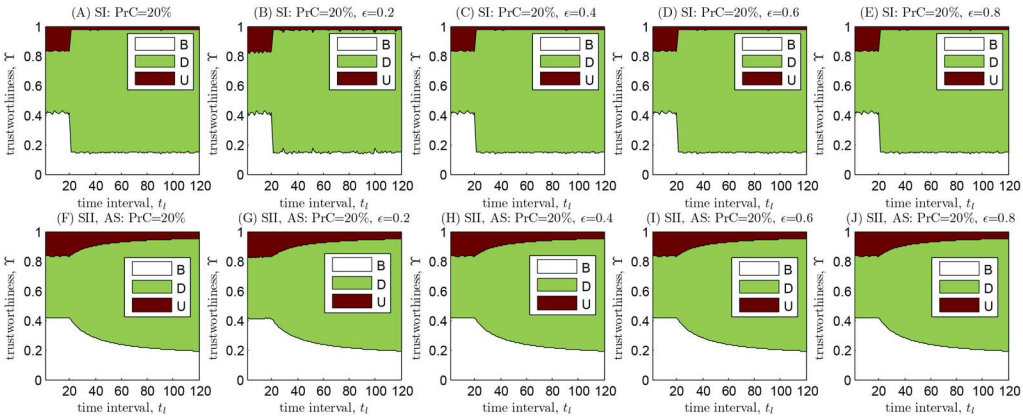


Fig. 16. Example of ONE-PARA does not work, when $cT = \{0.1, 0.1, 0.8\}$, $fT = \{0.1, 0.8, 0.1\}$, $\sigma_c = \sigma_f = 0.01$.

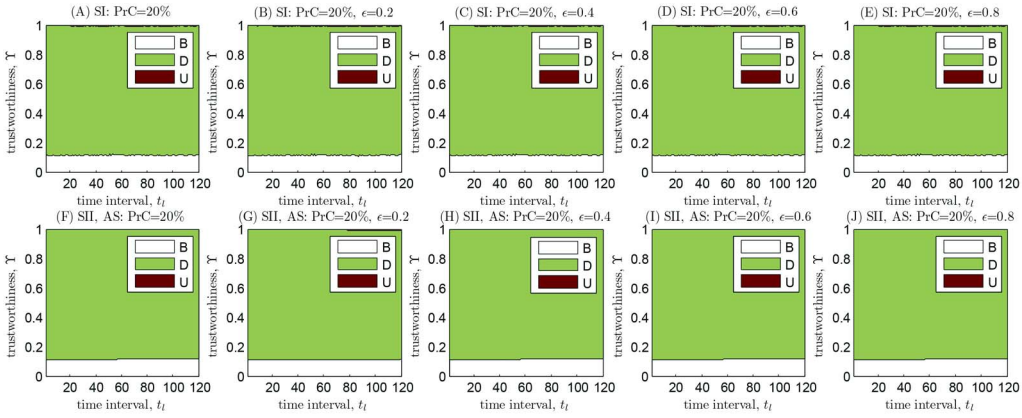


Fig. 17. Example of ONE-PARA works even it cannot identity the difference between $cT = \{0.1, 0.8, 0.1\}$ and $fT = \{0.1, 0.1, 0.8\}$, $\sigma_c = \sigma_f = 0.01$.

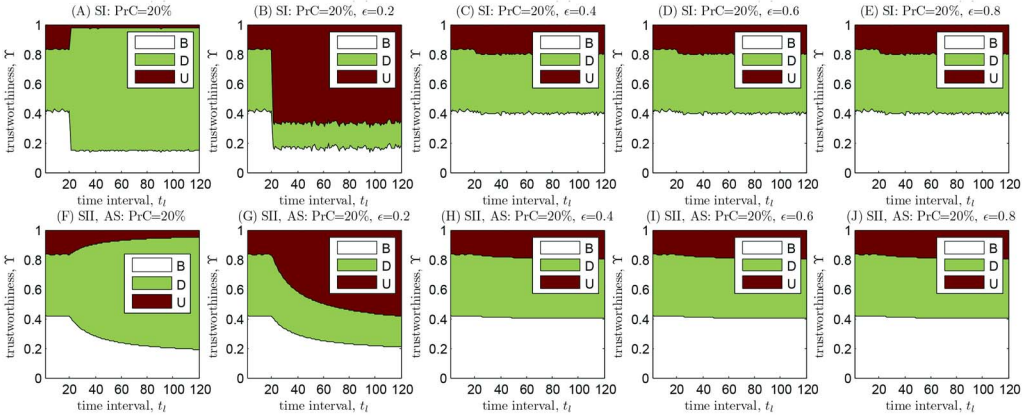


Fig. 18. Example of T-PARA works while TC-ONLY and ONE-PARA does not work well where $cT = \{0.1, 0.1, 0.8\}$, $fT = \{0.1, 0.8, 0.1\}$, $\sigma_c = \sigma_f = 0.01$.

higher value of B in Eq. (8), i.e., increase x. In contrast, we define a lower value for y to prevent $\mathcal{ADV}$ from decreasing trustworthiness. To prevent $\mathcal{ADV}\_Hbd$ (i.e., $\mathcal{ADV}$ generates false trust opinions in any way to manipulate sensor trustworthiness), a larger value of z can be defined to put more weight on uncertainty U.

Note that Eq. (6) and Eq. (7) are special cases of Eq. (8) when $x = 1$, $y = z = 0$ and $x = y = z = 1$, respectively. Finally, we have the following observations.

- If $\mathcal{ADV}\_Homo$ intends to increase Υ, ONE-PARA in terms of B is better than T-PARA since B is the only weight factor in it. That is, $x = 1$, $y = z = 0$, meaning that D and U are not taken into consideration.
- In contrast, if $\mathcal{ADV}\_Homo$ wants to decrease Υ, ONE-PARA in terms of D is better than T-PARA since D is the only weight factor in it.
- T-PARA is resilient to $\mathcal{ADV}\_Noise$, $\mathcal{ADV}\_Homo$ and $\mathcal{ADV}\_Hbd$.
- T-PARA-WF is a more flexible way to prevent $\mathcal{ADV}$ from various attacks. The selection of x, y and z is scenario dependent.

The countermeasures against $\mathcal{ADV}$'s pollution attack strategies are summarized in Table 2. Here, *Fair* and *Good* indicate

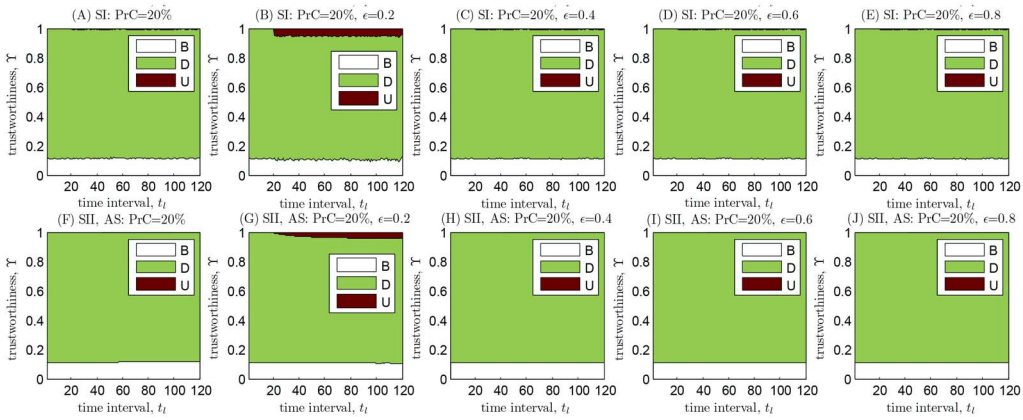


Fig. 19. Example of TC-ONLY works well while T-PARA slightly compromised the result when ϵ is selected too small. $cT = \{0.1, 0.8, 0.1\}$, $fT = \{0.1, 0.1, 0.8\}$, $\sigma_c = \sigma_f = 0.01$.

how the countermeasures are resilient to $\mathcal{ADV}$'s pollution attack strategies. *Good* means that a countermeasure (e.g., ONE-PARA) is more resilient than TC-ONLY (i.e., *Fair*), once attacked by $\mathcal{ADV}\_Noise$.

8 Conclusion

In this paper, we have proposed a family of efficient and robust trust management schemes for UWSNs based on Subjective Logic. Our advanced trust storage scheme, AS, facilitates distributed trust data storage to ensure high reliability of trust data. It takes the advantage of both GHT and GPSR routing to find storage nodes and to route trust data. We have also proposed several methods to mitigate trust pollution attacks based on various trust similarity measures. We demonstrated that our trust management schemes are resilient to major attack categories including $\mathcal{ADV}\_Del$, $\mathcal{ADV}\_Noise$, $\mathcal{ADV}\_Homo$, and $\mathcal{ADV}\_Hbd$. Moreover, our simulation results demonstrated that AS has much lower storage costs compared with the less sophisticated approaches. Combining AS with similarity threshold measures, we are able to significantly reduce trust storage costs and perform efficient node invalidation and mitigation of $\mathcal{ADV}$'s pollution attacks.

Acknowledgments

The research leading to these results has received funding from the EU FP7-PEOPLE-IRSES program under grant agreement 247083, project acronym S2EuNet. V. Zadorozhny's research was supported in part by the Research Council of Norway through the L. Eiriksson mobility program, project 209237. Y. Ren's research was supported in part by the Aiming for the Top University and Elite Research Center Development Plan by Taiwan, and this work was partially done while Y. Ren was visiting the School of Information Sciences, University of Pittsburgh. Part of this paper was presented at the IEEE MDM conference, July 2012.

TABLE 2
$\mathcal{ADV}$'s Pollution Attack Strategies and Their Countermeasures

$\mathcal{ADV}$'s strategy	Countermeasures			
	TC-ONLY	ONE-PARA	T-PARA	T-PARA-WF
$\mathcal{ADV}\_Noise$	Fair	Good	Good	Good
$\mathcal{ADV}\_Homo$: **increase** Υ	No	Good	Fair	Good
$\mathcal{ADV}\_Homo$: **decrease** Υ	No	Good	Fair	Good
$\mathcal{ADV}\_Hbd$	No	No	Good	Good

References

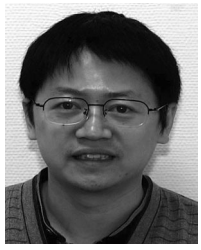
[1] D. Ma, C. Soriente, and G. Tsudik, "New adversary and new threats: Security in unattended sensor networks," *IEEE Netw.*, vol. 23, no. 2, pp. 43–48, Mar. 2009.
[2] Y. Ren, V. Oleshchuk, F. Y. Li, and S. Sulistyo, "SCARKER: A sensor capture resistance and key refreshing scheme for mobile WSNs," in *Proc. IEEE LCN*, Bonn, Germany, 2011.
[3] R. Di Pietro, L. V. Mancini, C. Soriente, A. Spognardi, and G. Tsudik, "Catch me (if you can): Data survival in unattended sensor networks," in *Proc. IEEE PERCOM*, Hong Kong, 2008.
[4] R. Di Pietro, G. Oligeri, C. Soriente, and G. Tsudik, "United we stand: Intrusion-resilience in mobile unattended WSNs," *IEEE Trans. Mobile Comput.*, vol. 12, no. 7, pp. 1456–1468, Jul. 2013.
[5] R. Di Pietro and N. Verde, "Epidemic data survivability in unattended wireless sensor networks," in *Proc. ACM WiSec*, Hamburg, Germany, 2011.
[6] K.-S. Hung, K.-S. Lui, and Y.-K. Kwok, "A trust-based geographical routing scheme in sensor networks," in *Proc. IEEE WCNC*, Kowloon, Hong Kong, 2007.
[7] A. Rezgui and M. Eltoweissy, "TARP: A trust-aware routing protocol for sensor-actuator networks," in *Proc. IEEE MASS*, Pisa, Italy, 2007.
[8] Y. Ren, V. Oleshchuk, and F. Li, "Optimized secure and reliable distributed data storage scheme and performance evaluation in unattended WSNs," *Comput. Commun.*, vol. 36, no. 9, pp. 1067–1077, May 2013.
[9] N. Lewis and N. Foukia, "Using trust in key distribution in wireless sensor networks," in *Proc. GLOBECOM Workshops*, Washington, DC, USA, 2007.
[10] S. Ratnasamy *et al.*, "GHT: A geographic hash table for data-centric storage," in *Proc. ACM WSNA*, 2002.
[11] M. Krasniewski, P. Varadharajan, B. Rabeler, S. Bagchi, and Y. Hu, "TIBFIT: Trust index based fault tolerance for arbitrary data faults in sensor networks," in *Proc. DSN*, 2005.
[12] S. Ganeriwal, L. Balzano, and M. Srivastava, "Reputation-based framework for high integrity sensor networks," *ACM Trans. Sen. Netw.*, vol. 4, no. 3, pp. 1–37, May 2008.
[13] K. Yadav and A. Srinivasan, "iTrust: An integrated trust framework for wireless sensor networks," in *Proc. ACM SAC*, New York, NY, USA, 2010.
[14] S. Buchegger and J. Le Boudec, "A robust reputation system for mobile ad-hoc networks," in *Proc. P2PEcon*, 2004.
[15] Y. L. Sun, W. Yu, Z. Han, and K. 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.

[16] M. Raya, P. Papadimitratos, V. Gligor, and J.-P. Hubaux, "On data-centric trust establishment in ephemeral ad hoc networks," in *Proc. IEEE INFOCOM*, Phoenix, AZ, USA, 2008.
[17] M. Omar, Y. Challal, and A. Bouabdallah, "Reliable and fully distributed trust model for mobile ad hoc networks," *Comput. Secur.*, vol. 28, no. 3–4, pp. 199–214, May 2009.
[18] L. Xiong and L. Liu, "PeerTrust: Supporting reputation-based trust for peer-to-peer electronic communities," *IEEE Trans. Knowl. Data Eng.*, vol. 16, no. 7, pp. 843–857, Jul. 2004.
[19] R. Zhou and K. Hwang, "PowerTrust: A robust and scalable reputation system for trusted peer-to-peer computing," *IEEE Trans. Parallel Distribut. Syst.*, vol. 18, no. 4, pp. 460–473, Apr. 2007.
[20] K. Walsh and E. G. Sirer, "Experience with an object reputation system for peer-to-peer filesharing," in *Proc. USENIX NSDI*, Berkeley, CA, USA, 2006.
[21] D. Ma and G. Tsudik, "DISH: Distributed self-healing," in *Proc. Int. Symp. SSS*, Detroit, MI, USA, 2008.
[22] Y. Ren, V. Oleshchuk, and F. Y. Li, "Secure and efficient data storage in unattended wireless sensor networks," in *Proc. IFIP Int. Conf. NTMS*, Cairo, Egypt, 2009.
[23] Y. Ren, V. I. Zadorozhny, V. Oleshchuk, and F. Y. Li, "An efficient, robust and scalable trust management scheme for unattended wireless sensor networks," in *Proc. IEEE MDM*, Bengaluru, India, 2012.
[24] M. Probst and S. Kasera, "Statistical trust establishment in wireless sensor networks," in *Proc. ICPADS*, Hsinchu, Taiwan, 2007.
[25] S. Shenker, S. Ratnasamy, B. Karp, R. Govindan, and D. Estrin, "Data-centric storage in sensornets," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 33, no. 1, pp. 137–142, 2003.
[26] M. T. Refaei, L. A. DaSilva, M. Eltoweissy, and T. Nadeem, "Adaptation of reputation management systems to dynamic network conditions in ad hoc networks," *IEEE Trans. Comput.*, vol. 59, no. 5, pp. 707–719, May 2010.
[27] A. Becher, Z. Benenson, and M. Dornseif, "Tampering with motes: Real-world physical attacks on wireless sensor networks," in *Proc. Int. Conf. SPC*, York, U.K., 2006.
[28] S. Tanachaiwiwat, P. Dave, R. Bhindwale, and A. Helmy, "Location-centric isolation of misbehavior and trust routing in energy-constrained sensor networks," in *Proc. IEEE IPCCC*, 2004.
[29] G. Theodorakopoulos and J. Baras, "On trust models and trust evaluation metrics for ad hoc networks," *IEEE J. Sel. Areas Commun.*, vol. 24, no. 2, pp. 318–328, Feb. 2006.
[30] A. Josang, "An algebra for assessing trust in certification chains," in *Proc. NDSS*, San Diego, CA, USA, 1999.
[31] K. Pelechrinis, V. Zadorozhny, and V. Oleshchuk, "Collaborative assessment of information provider's reliability and expertise using subjective logic," in *Proc. CollaborateCom*, Orlando, FL, USA, 2011.
[32] V. Oleshchuk and V. Zadorozhny, "Trust-aware query processing in data intensive sensor networks," in *Proc. Int. Conf. SensorComm*, Valencia, Spain, 2007.
[33] K. Pelechrinis, V. Zadorozhny, and V. Oleshchuk, "A cognitive-based scheme for user reliability and expertise assessment in Q&A social networks," in *Proc. IEEE Conf. IRI*, Las Vegas, NV, USA, 2011.
[34] D. Liu and P. Ning, "Multilevel μTESLA: Broadcast authentication for distributed sensor networks," *ACM Trans. Embed. Comput. Syst.*, vol. 3, no. 4, pp. 800–836, Nov. 2004.
[35] B. Karp and H. T. Kung, "GPSR: Greedy perimeter stateless routing for wireless networks," in *Proc. ACM MobiCom*, 2000.

Yi Ren is currently doing research as a postdoctoral researcher at National Chiao Tung University (NCTU), Taiwan. He received his Ph.D. degree in Information Communication and Technology from the University of Agder (UiA), Norway, in April 2012. His current research interests include security in wireless sensor networks, ad hoc, and mesh networks, LTE, smart grid, and e-health security. He received the Best Paper Award in IEEE MDM 2012. He is a member of the IEEE.

Vladimir I. Zadorozhny received the Ph.D. degree from the Institute for Problems of Informatics, Russian Academy of Sciences, Moscow, in 1993. He is an Associate Professor in the School of Information Sciences, University of Pittsburgh. Before coming to United States, he was a Principal Research Fellow in the Institute of System Programming, Russian Academy of Sciences. Since May 1998, he has been working as a Research Associate at the Institute for Advanced Computer Studies, University of Maryland, College Park. He joined the University of Pittsburgh in September 2001, where he is the head of the Network Aware Data Management Group and a member of the Advanced Data Management Technologies Laboratory. His research interests include networked information systems, complex adaptive systems, WSN data management, query optimization in resource constrained distributed environments, and scalable architectures for wide-area environments with heterogeneous information servers. He received the Best Paper Award in IEEE MDM 2012. He is a senior member of the IEEE.

Vladimir A. Oleshchuk is Professor of Computer Science at the University of Agder, Norway. He received his Ph.D. degree in Computer Science (1988) from the Taras Shevchenko Kiev State University, Kiev, Ukraine. He is a senior member of the IEEE and a senior member of the ACM. His current research interests include formal methods and information security, privacy and trust with special focus on telecommunication systems. He received the Best Paper Award in IEEE MDM 2012.



Frank Y. Li (S'99, M'03, SM'09) holds a Ph.D. degree from the Norwegian University of Science and Technology (NTNU). He worked as a Senior Researcher at UniK - University Graduate Center, University of Oslo, before joining the Department of Information and Communication Technology, University of Agder (UiA) in August 2007 where he is currently a Professor. During the past few years, he has been an active participant in several Norwegian and EU FP6/FP7 research projects. He is listed as a Lead Scientist by the European Commission DG RTD Unit A.03 - Evaluation and Monitoring of Programmes in Nov. 2007. Dr. Li's research interest includes 3G/4G and beyond mobile systems and wireless networks, mesh and ad hoc networks; wireless sensor network; Device-to-Device (D2D) communication; cooperative communications; cognitive radio networks; green wireless communications; QoS, resource management and traffic engineering in wired and wireless IP-based networks; analysis, simulation and performance evaluation of communication protocols and networks. He received the Best Paper Award in IEEE MDM 2012. He is a senior member of the IEEE.