PROVEST: Provenance-based Trust Model for Delay Tolerant Networks

Jin-Hee Cho, *Senior Member, IEEE,* Ing-Ray Chen, *Member, IEEE*

Abstract—Delay tolerant networks (DTNs) are often encountered in military network environments where end-to-end connectivity is not guaranteed due to frequent disconnection or delay. This work proposes a provenance-based trust framework, namely PROVEST (PROVEnance-baSed Trust model) that aims to achieve accurate peer-to-peer trust assessment and maximize the delivery of correct messages received by destination nodes while minimizing message delay and communication cost under resource-constrained network environments. Provenance refers to the history of ownership of a valued object or information. We leverage the interdependency between trustworthiness of information source and information itself in PROVEST. PROVEST takes a data-driven approach to reduce resource consumption in the presence of selfish or malicious nodes while estimating a node's trust dynamically in response to changes in the environmental and node conditions. This work adopts a model-based method to evaluate the performance of PROVEST (i.e., trust accuracy and routing performance) using Stochastic Petri Nets. We conduct a comparative performance analysis of PROVEST against existing trust-based and non-trust-based DTN routing protocols to analyze the benefits of PROVEST. We validate PROVEST using a real dataset of DTN mobility traces.

Index Terms—delay tolerant networks, provenance, store-and-forward, trust, trustworthiness

1 Introduction

Delay or disruption tolerant networks (DTNs) are often observed in emerging applications such as emergency response, special operations, smart environments, habitat monitoring, and vehicular ad-hoc networks where multiple nodes participate in group communications to achieve a common mission. The core characteristic of DTNs is that there is no guarantee of end-to-end connectivity, thus causing high delay or disruption due to inherent characteristics (e.g., wireless medium, resource constraints, or high mobility) or intentionally misbehaving nodes (e.g., malicious or selfish) [1]. Managing trust efficiently and effectively is critical to facilitating cooperation or collaboration and decision making tasks in DTNs while meeting system goals such as reliability, availability, quality of service (QoS), and/or scalability. Accurate trust evaluation is especially challenging in DTN environments because nodes are sparsely scattered and do not often encounter each other. Therefore, encounter-based evidence exchange among nodes may not be always possible. The lack of direct interaction experience in DTN environments hinders continuous evidence collection and can result in incorrect trust estimation, leading to poor application performance. In this work, we propose the use of provenance information for evidence propagation for sparse DTNs without solely relying on encounter-based evidence exchange. Unlike existing encounter-based trust protocols [2], [3], our protocol does not require two nodes to exchange trust evidence upon encounter to estimate trust of each other while achieving high trust accuracy by leveraging provenance information embedded in a message during message delivery.

According to the *Oxford English Dictionary*, *provenance* is defined as "the source or origin of an object; its history and pedigree; a record of the ultimate derivation and passage of an item through its various owners." Data provenance has been used to analyze scientific data in many applications. Data provenance is sometimes called "lineage" or "pedigree," describing the origins of a piece of data and how the data achieves the current state [4].

A major challenge of a provenance-based system is that it must defend against attackers who may modify or drop messages including provenance information or disseminate fake information. Leveraging the interdependency of trust in information source and information itself based on the concept of provenance, this work proposes a provenance-based trust framework, called PROVEST (PROVEnance-baSed Trust model), that aims to answer the challenge. The design goals of PROVEST are (1) minimizing trust bias; (2) minimizing communication cost caused by trust assessment; and (3) maximizing quality-of-service (QoS) by minimizing message delivery delay and maximizing correct message delivery ratio.

This paper has been substantially extended from our previous work [5] as follows.

1) We significantly refine the previous trust model in [5] by considering the following enhancements: (a) trust is scaled in $[0, 1]$ as a real number; (b) trust evidence, either direct or indirect evidence, is modeled by the Beta distribution [6] with evidence filtering, treating evidence in a Bayesian way, to make PROVEST more generic with the amount of positive and negative evidence; (c) trust dimensionality is considered by which multiple dimensions of trust

Jin-Hee Cho is with U.S. Army Research Laboratory, Adelphi, MD 20783, e-mail: jinhee.cho@us.army.mil
Ing-Ray Chen is with the Department of Computer Science, Virginia Tech, Falls Church, VA 22043, e-mail: irchen@vt.edu

can be captured independently; and (d) four variants of PROVEST are devised to deal with uncertain evidence caused by message loss or modification.
2) We consider a more comprehensive set of performance metrics to characterize QoS, including the average delay occurred to deliver a message and the ratio of correct message delivery. Furthermore, we consider trust bias per trust property to give a more comprehensive understanding of the relationships between the accuracy of trust estimation and the routing performance in PROVEST.
3) We conduct a performance analysis comparing four variants of PROVEST with existing trust-based and non-trust-based DTN routing protocols and validate the results using a real dataset of DTN mobility traces.

This work has the following unique contributions:

1) PROVEST significantly reduces communication cost, compared to existing counterparts, by using provenance information (i.e., identification and opinion towards a previous message carrier) tagged in messages. In PROVEST, a trustor does not directly request recommendations from third parties because collecting recommendations requires extra overhead, and recommendations are often not available in a sparse DTN. Rather, PROVEST allows indirect evidence (recommendations) to be collected via message delivery even for two nodes that have not encountered each other for a long time.
2) We propose to characterize a DTN node with the concept of multidimensional trust, including availability, integrity, and competence, in the context of trust evidence propagation using provenance information. Although multidimensional trust has been considered in other networks (e.g., [7]), we are the first to consider its use in a provenance-based trust model in DTN environments.
3) We consider sophisticated attack scenarios that often happen in dynamic DTN environments where various types of hostile entities exist to interrupt service availability. In particular, we consider the case in which provenance information can be dropped, modified, or forged by attackers in DTNs.
4) We develop a model-based evaluation method based on Stochastic Petri Nets (SPNs) to identify the optimal minimum trust threshold in selecting a message carrier to achieve availability, integrity, and competence.
5) We conduct a comprehensive performance analysis to demonstrate the superiority of PROVEST, over existing trust-based and non-trust-based DTN routing protocols with simulation validation, in terms of trust accuracy and routing performance.

The rest of this work is organized as follows. Section 2 discusses existing approaches using provenance techniques in the literature. Section 3 describes our network model, key management, and attack model. Section 4 provides an overview of PROVEST in terms of trust dimensionality, aggregation, and evidence computation. Section 5 develops a SPN model of describing a node's behavior. Section 6 defines performance metrics, existing schemes to be compared with PROVEST, and the experimental setup, followed by a comparative performance analysis. In Section 7, we conclude the paper.

2 Related Work

This section gives the literature background in provenance models and routing protocols in DTNs.

2.1 Provenance Model

The Open Provenance Model (OPM) was introduced to represent data provenance, process documentation, data derivation, and data annotation [8]. Since then, OPM has been widely adopted and extended by various research groups [9]. Freire *et al.* [10] surveyed diverse models of provenance management but did not discuss the use of provenance for security. McDaniel [11] addressed that accurate, timely, and detailed provenance information leads to good security decisions.

Provenance has been used to verify trust, trustworthiness, or correctness of information in many research areas. Rajbhandari *et al.* [12] examined how provenance information is associated with a workflow in a Bio-Diversity application. Dai *et al.* [13] proposed a data provenance trust model to evaluate trustworthiness of data and data providers. Yu *et al.* [14] presented an agent-based approach to managing information trustworthiness in network centric information sharing environments. Golbeck [15] used provenance information to infer trust in Semantic Web-based social networks. Zhou *et al.* [16] used data provenance computations and queries over distributed streams for effective network accountability and forensic analysis to enhance network security. However, the above studies [12]-[16] focused on evaluating trustworthiness in information without considering specific network attack behaviors that may maliciously change the original messages and disrupt system goals.

Some researchers have made efforts to secure provenance data. Hasan *et al.* [17] insisted that secure provenance is a critical aspect to increase protection of provenance information. Braun *et al.* [18] explained that "provenance" consists of relationships (i.e., a graph) and attributes (i.e., attributes of an entity). Hasan *et al.* [19] presented a provenance-aware prototype to ensure integrity and confidentiality of provenance information based on provenance-tracking of data writes at the application layer. Wang *et al.* [20] proposed a "chain-structure" provenance scheme that provides security assurance for provenance meta-data. Gadelha and Mattoso [21] proposed a security architecture framework that protects authorship and temporal information in grid-enabled provenance systems. Lu *et al.* [22] proposed a provenance scheme using the bilinear pairing techniques in order to secure provenance data of ownership and process history of data object in cloud computing. The above works [17]-[22] have studied how to secure provenance data with the existence of a centralized trusted entity.

Some researchers have proposed provenance-based trust models in sensor networks [23]-[24], but they assumed full knowledge of the network topology, and did not consider attack behaviors.

2.2 Routing Protocols in DTNs

Various routing protocols have been studied for DTNs. Due to the characteristic of the DTN with no guarantee of end-to-end connectivity, flooding or partial flooding approaches based on connectivity probability have been popularly considered such as Epidemic [25] or PRoPHET [26]. However, these approaches tend to cause network congestion or high interference, and high resource consumption to process and switch operations.

To mitigate these disadvantages, researchers have developed opportunistic routing protocols in which a relay node is selected based on certain criteria including historical mobility patterns called RelayCast [27], a fixed point opportunistic routing using inter-contact times between nodes [28], and a cluster-based routing protocol for DTNs [29] where a cluster is formed based on similar mobility patterns. Further, practical routing protocols have been proposed, such as using a metric estimating the average waiting time of a next message carrier [30], and context-aware opportunistic routing protocols (e.g., CAR [31], HiBOp [32]). These routing protocols assume that historical profiles are available to derive probabilistic predictability, which may not be available in practice.

Social behavior based opportunistic routing protocols have received significant attention in DTN routing research community. Wei *et al.* [33] described common social metrics to capture social behaviors of an entity in terms of centrality (e.g., degree, closeness, betweenness, bridging), similarity, community, and selfishness in DTNs. Zhu *et al.* [34] also categorized social behaviors based routing features as positive properties which benefit the relay selection (e.g., community, centrality, similarity and friendship) and negative properties which hurt the network performance (e.g., selfishness). Costa *et al.* [35] proposed a social-interaction based routing protocol, called SocialCast. Li *et al.* [36] studied how the performance of epidemic routing is affected by the social selfishness of nodes in DTNs. Gao and Cao [37] proposed a user-centric data dissemination based on the tradeoff between effectiveness of relay selection and cost to maintain network information. Gao *et al.* [38] developed a multidimensional routing protocol, called M-Dimensions. A hypercube-based social feature is proposed for multi-path routing in DTNs [39]. Abdelkader *et al.* [40] proposed a social group based routing scheme, named SGBR. Li and Shen [41] proposed a distributed utility-based routing protocol, called SEDUM, for DTNs.

Recently, Zhu et al. [42] proposed trust-based secure routing protocol called *iTrust* assuming that a trusted authority (TA) is periodically available to estimate reputation of DTN nodes. In contrast, PROVEST does not use a TA for centralized trust management. PROVEST is inherently distributed without a single point of failure. Ayday and Fekri [43] proposed an Iterative Trust and Reputation Mechanism (ITRM) leveraging message passing techniques for decoding low-density parity-check codes over bipartite graphs. The authors showed the outperformance of the proposed scheme over EigenTrust [44] and Bayesian Framework [45] in the presence of malicious nodes. The basic idea of ITRM is for the raters to supply their ratings toward a trustee node to a trustor node and the rater that deviates the most from the others will be flagged as a malicious rater. This process is iteratively executed until all remaining raters considered trustworthy converge to a global reputation for the trustee node. A problem with ITRM is that there is large communication overhead for a trustor node to gather sufficient rating information in DTN environments.

PROVEST differs from the works cited above in that a DTN node does not rely on recommendation information solicited or exchanged during encounters to collect indirect evidence. Rather a DTN node collects indirect evidence information through provenance information embedded in messages being routed through the node. This greatly increases trust accuracy and routing protocol performance, especially for sparse DTNs where nodes do not often encounter each other, without incurring high communication overhead.

3 System Model

We propose a distributed provenance-based trust management protocol for secure group communications in DTN environments where legitimate members communicate through a symmetric key, called *group key* (The details will be given in Section 3.2). In DTN environments, a node cannot properly monitor a neighbor node upon encounter because of short contact time. This is especially the case when a node attempts to monitor if a neighbor node selected to carry a packet actually forwards the packet, since the neighbor node selected normally will not deliver the packet immediately until it is outside of the monitoring range. We classify evidence in three categories: positive, negative and uncertain, corresponding to the three cases of being able to observe positive behavior (with a false negative probability), being able to observe the negative behavior (with a false positive probability), and being uncertain about the behavior. When a node cannot properly monitor a neighbor node upon encounter because of a short contact time, it is classified as uncertain evidence. For example, cooperativeness behavior is manifested by the behavior for executing beacon, information exchange, packet receipt acknowledgement, and trust protocol execution expected out of a node during a contact period. However, the environment condition and short contact time may not allow conclusive positive or negative evidence. In this case, uncertain evidence is the best assessment outcome. In Section 4, we investigate four different ways to combine positive, negative, and uncertain evidence in PROVEST design. In Section 6, we also demonstrate and analyze the effects on trust accuracy and routing performance.

3.1 Network Model

We assume that nodes interact with each other not only to deliver messages, but also to exchange information for other purposes. A node is able to diagnose other nodes' attack behaviors based on its past direct experience. A given mission requires that each node, as a source, must send information to a list of destination nodes. Each node, as a destination node (DN), expects to receive information from a set of source nodes (SNs). For message delivery, nodes use the "store-and-forward" technique, meaning that a node carries messages until it encounters a message carrier (MC).

TABLE 1: Design Parameters and Their Meanings

Notation	Meaning
LT	Entire session time (lifetime)
v_i	An average speed of a node's lifetime assigned for SPN model
P_{fn} / P_{fp}	Probability of false negatives or false positives
P_r	Upper bound detection error for P_{fn} / P_{fp}
P_{ur}	Probability of link unreliability
λ / μ / P_λ	Join or leave rate / join probability $P_\lambda = \lambda/(\lambda+\mu)$
P_{cp}	% of compromised nodes
P_a	Probability of attack intensity
P_f	Probability of packet forwarding
N_p	Number of packets sent by a source node
N_{SD}	Number of source and destination pairs to send and receive messages
N_c	Number of message copies sent by a source node
$K_{S,t}$	A symmetric group key
T_{min}	Minimum trust threshold
$P_{i,k}$	PI provided by node i with its direct trust opinion towards the previous MC k
$T_{i,j}^X(t)$	Overall trust value of node j evaluated by node i for trust property X at time t
$T_j^X(t)$	Ground truth trust value of node j on trust property X at time t
$r_{i,j}$ / $s_{i,j}$	Number of positive / negative evidence toward j evaluated by i
T_e	Time taken to consume energy for one token in SPN
$[e, e']$	Initial energy level assigned from $U[e, e']$
T_i^{enc}	Time interval node i encounters node j
R_t	Radio transmission range (m)
$\mathcal{B}^X$	Trust bias measuring time averaged difference between trust of node j evaluated by node i for all i's and j's
$\mathcal{C}$	Communication cost per time unit ($sec.$) for a node to deal with trust evaluation ($C_e(t)$) and message delivery ($C_d(t)$) during LT
$\mathcal{R}$	Fraction of the number of packets correctly received by DNs over the total number of messages transmitted by SNs during LT
$\mathcal{D}$	Average delay occurred for a message to be delivered to a DN ($min.$) during LT
D_m	Delay occurred for message m to be delivered to a DN
L	A set of current legitimate members in a given mission group
G / I	A set of messages sent by SNs or DNs respectively
K	A set of MCs involved in a message delivery

Without loss of generality, we assume a square-shaped operational area consisting of $m \times m$ grid areas where each grid is also a square with the width and height equal to wireless radio range R_t. To model mobility patterns of nodes, we first use a random walk model for our mathematical modeling in the SPN model (see Section 5). A node randomly moves to one of five locations (i.e., north, west, south, east, and current location) in accordance with its speed. Node i's speed, v_i, is chosen uniformly over $U[v, v']$ m/s where v and v' are minimum and the maximum speeds respectively, and v_i is then fixed during the node's lifetime. The boundary grid areas are wrapped around (i.e., a torus is assumed) to avoid end effects. To reflect more realistic mobility patterns, we also validate our model with real data of DTN mobility traces by CRAWDAD [46] in Section 6.6. Each node does not know other nodes' locations and their mobility patterns due to the nature of store-forward message delivery. Nodes are modeled with heterogeneous characteristics with different speed, energy level, monitoring capability (i.e., detection error), cooperation probabilities (i.e., packet dropping), and honesty probabilities (i.e., good/bad mouthing, fake identity, message modification) as follows:

- **Speed** (v_i): A node is assigned an average speed of its lifetime, selecting from the range $U[v, v']$ based on uniform distribution.
- **Energy level** (e_i): A node is assigned an initial energy level selected from the range of $[e, e']$ (in $hrs.$) and its energy consumption is affected by its availability to serve requests.
- **Detection error** (P_{fp}, P_{fn}): A node has monitoring capability with detection error probabilities of false positives and false negatives on integrity trust and predicting energy level for competence trust. Each node's detection error probabilities (P_{fp} and P_{fn}) are selected from the range of $(0, P_r]$ where P_r is a probability ranged in $[0, 1]$.
- **Behavior seeds**: A malicious node exhibits its packet forwarding behaviors with probability P_f (for black hole and gray hole attacks) and random attack behaviors with probability P_a (for fake/no identity, false recommendation, and message modification attacks). See Section 3.3 for a list of malicious attack behaviors. We model these by assigning the seed probabilities (i.e., P_a and P_f) in the range of $[0, 1]$.

A node may join or leave the network for tactical reasons (e.g., saving energy by being sleep mode). This is modeled by a node's network join and leave events with rates λ and μ, giving the network join probability as $P_\lambda = \lambda/(\lambda+\mu)$ considered in our SPN model.

3.2 Key Management

We assume a group communication system in a DTN environment where multiple trusted authorities (TAs) exist in the operational area so that a node is allowed to access a TA to obtain a valid symmetric key for group communication. A node encrypts the entire "packet" (consisting of the message and provenance information) using a symmetric key $K_{S,t}$ given to legitimate members. Note that TAs are only used for group key management, not for trust management or packet routing. These TAs are essential in sparse DTN environments, because contributory group key management [47]

with all group members contributing to the group key generation based on *Diffie-Hellman* key exchange to agree on a secret key will not work in sparse DTN environments. TAs rekey the symmetric key $K_{S,t}$ periodically based on their pre-deployed hash functions. The symmetric keys issued at the same time t by multiple TAs are the same so that all legitimate nodes can communicate with the same key. The symmetric key is used to prevent outside attackers, not inside attackers. A node forwards a packet to a node whose trust is no less than T_{min}.

We define the provenance information (PI) generated by node i as tuple $(i, k, O_{i,k}(t))$, where k is the identification (ID) of the previous MC, and $O_{i,k}(t)$ is i's direct trust opinion towards the attack behaviors (i.e., ID, fake recommendation, and message modification attacks), and remaining energy level of k. The first three opinions on attack behaviors are used to estimate integrity trust of k while the last two (energy status and cooperativeness behavior) are used to measure competence trust of k. k's availability trust is estimated based on whether k's authentic ID is found in the PI. $O_{i,k}(t)$ will be recorded as the number of positive evidence r and negative evidence s per behavioral category and will be considered to evaluate each MC enclosed in the PI by a DN. We will discuss the three trust dimensions in detail in Section 4. We call a message used for mission execution a *mission message* (MM) hereafter.

For simplicity, we denote $(i, k, O_{i,k}(t))$ as $P_{i,k}$ representing the PI provided by i with its direct trust opinion towards the previous MC k. For example, a DN may receive a message such as

$$[MM, (P_{0,\emptyset})_{k_n}, (P_{1,0})_{k_{n-1}}, (P_{2,1})_{k_{n-2}}, \ldots, (P_{m,m-1})_{k_{n-m}}]_{K_{S,t}} \quad (1)$$

where MM denotes a mission message and $K_{S,t}$ is a symmetric key issued at time t. The SN's ID is 0, and other intermediate MCs' IDs are $1, 2, , m$ where m is the number of intermediate MCs. The message including both MM and PIs is encrypted by a symmetric key $K_{S,t}$. Note that the SN only encloses its ID since there is no previous MC. The apparent redundancy in the carried ID information is crucial in identifying ID attacks, as discussed in Section 4. Typically, the addition of meta-data by each relay node could lead to the so-called meta-data explosion problem if the number of hops or relays, m, is too large. We avoid this problem by using an optimal trust threshold to filter untrustworthy MCs and shorten the routing path length. In case attackers add multiple fake IDs causing the message size to exceed the normal range, the message will be detected as abnormal and will be dropped by legitimate, uncompromised nodes.

To prevent modification of PIs inserted by previous MCs, we adopt an encryption key mechanism based on micro-TESLA [48]. A pair of SN and DN obtain a base PI encryption key and decryption key, (k_0, k_n), from the closest TA. We assume that TAs are able to issue the same pair of keys (i.e., (k_0, k_n)) to a pair of SN and DN. The SN encrypts its PI using k_n and generates $k_{n-1} = F(k_n)$ to dictate the next MC to use k_{n-1}. Similarly, the next MC will encrypt its PI using k_{n-1} and pass k_{n-2} to its next MC. This process continues until the message arrives at a DN. A MC does not know the previous MC's PI encryption key, so it cannot decrypt the PI of the previous MC. When the DN receives the message, it can check with (k_0, k_n) if correct keys are being used on the path, and can properly decrypt all PIs by tracing back the key chains.

Unless attackers compromise the SN or the DN to know (k_0, k_n), PIs cannot be fully altered. Attackers may collude and exchange PI encryption keys but PI modifications may occur between attackers themselves which have little impact on overall attack behaviors. If a MC does not comply with using a given PI encryption key, the DN will fail to decrypt all PIs and discard the message. This will eventually lead to identifying malicious nodes. Thus, we assume that a smart attacker might want to follow the PI encryption/decryption protocol to gain trust. However, using PI encryption/decryption keys does not guarantee that each MC provides correct PI because a compromised MC may not insert its own PI or may insert false PI for itself even.

Symmetric keys and PI encryption/decryption keys (distributed by a TA) are distributed via a public/private key pair. Each node will use a TA's public key to request proper keys and a TA is preloaded with public keys of all nodes in the network. Each node will decrypt a message carrying the symmetric or PI encryption key using its private key. Thus, non-TA nodes do not need to store public keys of all nodes.

3.3 Attack Model

We consider the following insider attacks:

- **No/fake Identity**: Our protocol requires that a MC should insert its ID in the PI tuple. However, an attacker may not add its real ID or may insert a fake ID. If this attack is successful, this attacker's misbehavior may be interpreted as another node's misbehavior, leading to inaccurate trust evaluation.
- **Fake recommendation**: A node may perform a bad mouthing attack and ballot stuffing attack (i.e., good mouthing attack) by giving a bad direct opinion towards a good node or by providing a good direct opinion towards a bad node. This hinders accurate trust evaluation. Note that self-recommendation is not allowed in the protocol.
- **Message modification**: A legitimate node with a symmetric key may modify MM. To prevent PI modification by other MCs, we use PI encryption keys using micro-TESLA [48], as discussed in Section 3.2.
- **Black hole attack**: A node may persistently drop packets to perform denial-of-service (DoS) attack. This is considered by a node's persistent packet dropping with the full strength of attack intensity.
- **Gray hole attack**: A node may randomly drop packets to perform random DoS attack. A node's random packet dropping is considered by varying the attack intensity.
- **Whitewashing**: a malicious node may leave a network and come back later with a new ID. In order to prevent this newcomer attack, the authenticity of a node is ensured through private/public key pairs which were given in network deployment. In order for a node to rejoin the network, the node should generate its private key based on its previous

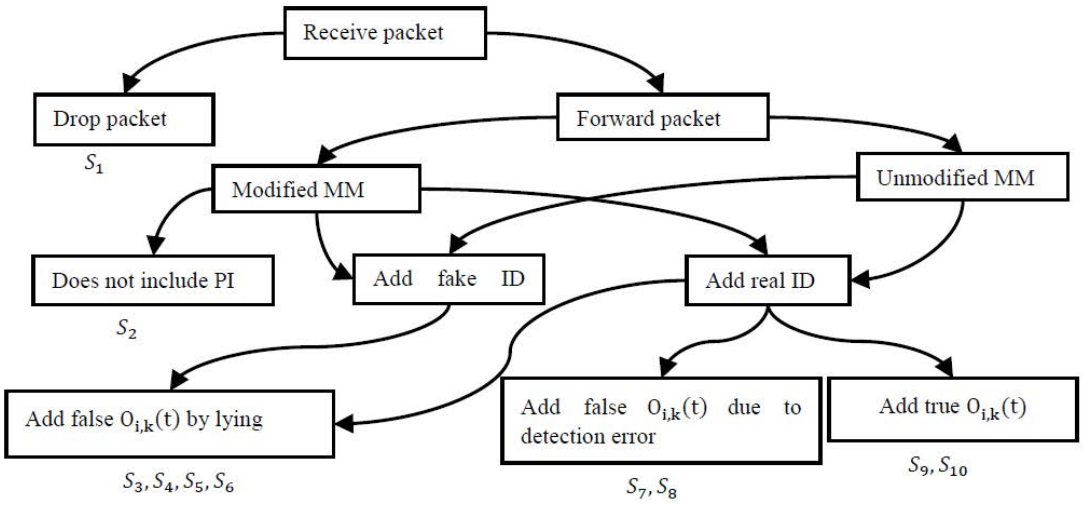


Fig. 1: Attack scenarios graph

private key, and other nodes interacting with the node will ensure the rejoined node's ID based on message exchange using a new public key generated based on the rejoined node's previous public key. If a malicious node uses a different private key in order to fake its ID, then it will be detected because it will not pass the authentication test based on the message exchange using the newly generated key pair.

- **Packet injection**: a malicious node can inject a packet to be delivered to another malicious node. This attack generates additional communication overhead which may hinder the delivery of legitimate packets.

Packet dropping can occur for reasons such as congestion, mobility, or limited resources available at the node. We differentiate between packet dropping and black/gray hole attack behaviors by two distinct trust dimensions, *availability* and *integrity*, which will be discussed in Section 4.1.

Fig. 1 illustrates the attack scenarios considered in this work. Each node's behavior path is indicated with a scenario S_i for $i = 1$ to 10. When a DN evaluates a node, if it does not see the node's ID in a received message, it will reduce a trust point for availability. If an attacker does not insert its ID or inserts a fake ID, it will be penalized by a loss of the trust level. A smart attacker may want to reveal its real ID to avoid penalty. If the attacker decides to insert a fake ID, it will provide false $O_{i,k}(t)$. Note that only S_{10} does not perform any attack, representing an honest node's path.

A compromised node may attack randomly in order to evade detection. We will analyze the effect of the random attack probability, P_a, on protocol performance in Section 6. Our trust protocol uses *integrity* to assess a node's maliciousness. A malicious node with low integrity trust will be penalized with isolation. In PROVEST, a malicious node with a trust value lower than a threshold will not be selected as a MC, practically isolating it from participating in packet routing activities.

4 PROVEST

This section gives an overview of PROVEST in terms of trust dimensions, trust aggregation, and trust evidence computation.

4.1 Trust Dimensions

The trust of each node is assessed based on three trust dimensions:

- **Availability** refers to service availability that may be affected by network or node conditions such as congestion, mobility, and limited resources available at the node.
- **Integrity** measures how well a node complies with a given protocol, without exhibiting attack behaviors.
- **Competence** reflects a node's remaining battery lifetime (a surrogate for resources available at the node) plus cooperativeness (i.e., contributing to reliable packet delivery) to serve requests received.

Note that the above trust dimensions will be assessed by PROVEST dynamically. A node running into network congestion will have low availability trust, but as soon as the congestion condition is clear, the node will regain high availability trust.

4.2 Trust Aggregation

Peer-to-peer trust estimation follows Bayesian update [6] based on the amount of positive evidence, r, and the amount of negative evidence, s, which can be derived from either direct evidence based on observations or indirect evidence based on PI embedded in a MM during message delivery. The expected trust value can be simply obtained by $\frac{r}{r+s}$. Trust value will be initiated with $r = 1$ and $s = 1$ which gives 0.5 indicating complete ignorance in the initial deployment.

The value of each trust dimension is aggregated based on accumulated evidence from the past and the new evidence. In DTNs, since two nodes cannot often encounter, new evidence may not be available. If direct evidence is not available, indirect evidence is incorporated based on Bayesian update. Direct evidence is observed upon every encounter with another node, while indirect evidence is collected when a DN receives a MM enclosing PIs. We

assume that two nodes can observe each other during their encountering period. However, monitoring may not be possible due to unreliable link or short contact time. This leads to uncertain evidence which will be handled by PROVEST. In addition, when the DN receives PI passed along with the MM, if the indirect trust evidence enclosed in the PI are detected as false, the evidence will not be used. This is also treated as uncertainty because correct evidence is not available in this update interval. In PROVEST, we capture this uncertainty using another parameter indicating the amount of uncertain evidence, denoted as u. We evaluate its impact on trust bias (i.e., the difference between ground truth trust value and measured trust value) by devising four variants of PROVEST as: *PROVEST-Pessimistic*, *PROVEST-Optimistic*, *PROVEST-Realistic*, and *PROVEST-Hybrid*. In the discussion below, we use the notation i to refer to a trustor (i.e., evaluator) and j to refer to a trustee (i.e., evaluatee).

PROVEST-Pessimistic treats uncertain evidence as negative evidence based on the nature of trusting less under no correct evidence available, and is computed by

$$T_{i,j}^{pessi-X} = \frac{r_{i,j} + r'_{i,j}}{r_{i,j} + s_{i,j} + u_{i,j} + r'_{i,j} + s'_{i,j} + u'_{i,j}} \quad (2)$$

$r_{i,j}$, $s_{i,j}$, and $u_{i,j}$ are the amount of positive, negative, and uncertain evidence accumulated from the past respectively. $r'_{i,j}$ and $s'_{i,j}$ are the amount of new positive and negative evidence, either from direct or indirect evidence. If uncertain evidence is obtained due to the lack of correct evidence, $u'_{i,j}$ will be considered as negative evidence as above. For new evidence, $r'_{i,j}$, $s'_{i,j}$, and $u'_{i,j}$, Section 4.3 explains how to compute them in terms of either direct or indirect evidence.

PROVEST-Optimistic treats uncertainty as credits based on the nature of trusting more, and estimates trust as

$$T_{i,j}^{opti-X} = \frac{r_{i,j} + u_{i,j} + r'_{i,j} + u'_{i,j}}{r_{i,j} + s_{i,j} + +u_{i,j} + r'_{i,j} + s'_{i,j} + u'_{i,j}} \quad (3)$$

While uncertain evidence is consider in the above two schemes, *PROVEST-Realistic* only relies on evidence available by ignoring the uncertain evidence as

$$T_{i,j}^{real-X} = \frac{r_{i,j} + r'_{i,j}}{r_{i,j} + s_{i,j} + r'_{i,j} + s'_{i,j}} \quad (4)$$

Under *PROVEST-Realistic*, if no new evidence is available, it does not update trust.

By leveraging the three schemes above, we propose a hybrid scheme called *PROVEST-Hybrid* which determines how to deal with uncertain evidence based on historical patterns of the amount of evidence. *PROVEST-Hybrid* computes trust as

$$T_{i,j}^{hybrid-X} = \begin{cases} T_{i,j}^{pessi-X} \text{ if } r_{i,j} < s_{i,j} \\ T_{i,j}^{opti-X} \text{ if } r_{i,j} > s_{i,j} \\ T_{i,j}^{real-X} \text{ if } r_{i,j} == s_{i,j} \end{cases} \quad (5)$$

That is, depending on the ratio of the amount of positive and negative evidence, trust is computed based on the trust estimation using *PROVEST-Pessimistic*, *PROVEST-Optimistic*, or *PROVEST-Realistic* scheme.

We omitted time t in the above equations for simplicity; the trust value above is a time-varying parameter where $r_{i,j}$, $s_{i,j}$, and $u_{i,j}$ indicate the amount of evidence until $t-\Delta$ and $r'_{i,j}$, $s'_{i,j}$, and $u'_{i,j}$ are the amount of new evidence at time t.

When nodes i and j encounter, node i will entirely rely on direct observation towards node j's behavior to collect new evidence at time t. Direct trust assessment can be conducted between any two encountering nodes who have monitoring capability characterized by detection errors (i.e., negative / positive probabilities).

Note that indirect trust assessment can be conducted only when node i is a DN. That is, node i (DN) will rely on the MM to evaluate trustworthiness of node j. In this case, time t represents the time the DN receives the MM.

4.3 Trust Computation

Now we discuss how the trust value of each trust property can be computed based on either direct evidence or indirect evidence.

4.3.1 Direct Evidence

When two nodes i and j encounter, a new trust value, $r'_{i,j}$, $s'_{i,j}$, or $u'_{i,j}$ is computed solely based on direct evidence. For each trust property, a direct trust value is computed as follows.

1) **Direct availability trust** is measured by whether a node is available to serve requests by exchanging a simple message to ensure connectivity. For trustor i to evaluate trustee j's availability, if j replies to i's message, $(r'_{i,j}, s'_{i,j}, u'_{i,j})$ is set to $(1, 0, 0)$; otherwise they are set to $(0, 1, 0)$. Note that $(0, 0, 1)$ is not applicable in this case.
2) **Direct integrity trust** is measured based on whether a node exhibits three attack behaviors: identity attack (no ID or fake ID inserted in PI), fake recommendation attack (i.e., good mouthing and ballot stuffing attacks), and message modification attack. Identity attack is detected based on challenge/response authentication message exchanges based on PKI between two nodes when they directly interact. Note that a node's public/private key pairs can be used for authentication purposes. Fake recommendation attack can be detected when a node receives a vastly different recommendation compared to a past trust for the same node. A current MC decides whether the MM forwarded by the previous MC is modified or not based on the trust of the previous MC. Each exhibiting attack behavior is counted as evidence, resulting in $r'_{i,j}+s'_{i,j}+u'_{i,j} = 3$ in this case. Integrity-related behavior of a node is triggered by a behavioral seed in integrity, P_a. If j is not available in availability trust above, direct integrity trust is uncertain and the evidence set will be recorded as $(r'_{i,j}, s'_{i,j}, u'_{i,j}) = (0, 0, 3)$.
3) **Direct competence trust** is assessed by a node's energy status and cooperativeness behavior, and thus is measured based on two pieces of evidence with $r'_{i,j}+s'_{i,j}+u'_{i,j} = 2$. Energy represents the capability or competence of node j to do the basic routing function. Node i counts the ratio of the number of acknowledgement packets received from node j (at the MAC layer) over transmitted packets to node j in the encounter interval to estimate energy status in node j. Cooperativeness behavior is manifested by

the behavior for executing beacon, information exchange, packet receipt acknowledgement, and trust evaluation protocols expected out of node j. Cooperativeness status can be computed by the number of positive experiences over the total experiences in cooperativeness behavior. Here we note that node i will not monitor if node j has forwarded a packet since it is impractical to monitor packet forwarding in DTNs. Similar to the direct integrity trust, if j is not available, the evidence is uncertain and will be recorded as $(r'_{i,j}, s'_{i,j}, u'_{i,j}) = (0, 0, 2)$.

4.3.2 Indirect Evidence

When nodes i and j are distant without direct communication, if node i is a DN, it will rely on PIs passed along with the MM delivered in order to derive indirect evidence.

1) **Indirect availability trust** of node j is evaluated as positive if (1) node j's ID is enclosed in j's PI; (2) node j's ID is authentic by ensuring that j's ID inserted by j in j's PI matches with j's ID inserted by j's next MC in the next PI; and (3) both j's previous MC and j have a trust value above the minimum trust threshold (T_{min}) based on i's evaluation. If the three conditions are met, $(r'_{i,j}, s'_{i,j}, u'_{i,j})$ is set to $(1, 0, 0)$; if j's ID is in the PI but the ID between j's PI and j's next MC's PI is not matched, $(r'_{i,j}, s'_{i,j}, u'_{i,j})$ is set to $(0, 1, 0)$. If no evidence is available for j in the enclosed PI, $(r'_{i,j}, s'_{i,j}, u'_{i,j})$ is set to $(0, 0, 1)$.
2) **Indirect integrity trust** of node j is estimated based on three pieces of evidence including fake/no identity insertion, fake recommendation, and message modification. They can be evaluated only when PI is inserted. If PI is not inserted, $(r'_{i,j}, s'_{i,j}, u'_{i,j})$ is set to $(0, 0, 3)$, treating all three pieces of evidence as uncertain. If PI is inserted, each evidence is evaluated by checking the source of opinions based on the trust value estimated in the last trust update, $t - \Delta$, using the minimum trust threshold T_{min}. In the PI, each MC inserts its opinions towards the previous MC $O_{i,k}(t)$ as discussed earlier. If j's next MC's trust value perceived by i (DN) is above T_{min}, then its opinion will be trusted and used as correct new indirect evidence to be incorporated with the past evidence. Similar to the direct integrity trust, $r'_{i,j} + s'_{i,j} + u'_{i,j} = 3$ in the indirect integrity trust.
3) **Indirect competence trust** of node j is estimated based on two pieces of evidence including remaining energy status and cooperativeness behavior. If PI is not inserted, $(r'_{i,j}, s'_{i,j}, u'_{i,j})$ is set to $(0, 0, 2)$, treating all two pieces of evidence as uncertain. If PI is inserted, each piece of evidence is evaluated by checking if the trust value of j's next MC is higher than the minimum trust threshold (T_{min}). If true, the opinion of j's next MC is accepted; otherwise, it is rejected. Similar to the direct competence trust, $r'_{i,j} + s'_{i,j} + u'_{i,j} = 2$ in the indirect competence trust.

In this work, we examine the effect of T_{min} on performance of PROVEST, as shown in Section 6.

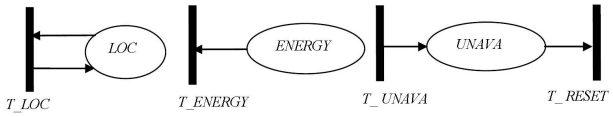


Fig. 2: SPN Model

5 Stochastic Petri Nets

We use SPN because of its efficient representations of a large number of states where the underlying model is a continuous-time Markov or semi-Markov chain. We develop a hierarchical modeling technique based on SPN to avoid state explosion problems and to improve solution efficiency for realizing and describing the behaviors of each node and obtaining objective trust values.

We develop event subnets to describe a node's behavior and its actual trust value as shown in Fig. 2. A hierarchical SPN technique is used to derive interactions or trust relationships of one node with other nodes in the system. We conduct this process by running the SPN subnet N times for the N nodes in the network. We use the information obtained from SPN for trust evaluation. In SPN, we call each oval in Fig. 2 a *place* where *mark (place name)* is the number of tokens in the place. The number of tokens in different places indicates the status (state) of a node. Each transition bar (i.e., *T_NAME*) is the rate at which the corresponding event is triggered.

In the SPN developed, we capture the location information such as the probability that a node is located in a particular area in order to estimate the encounter interval between two nodes. To validate the proposed model with real mobility patterns of nodes in DTNs, we use a real dataset of DTN mobility traces [46] to estimate the encounter interval between any two nodes and use the SPN data to describe each node's behavior. In addition, we obtain remaining energy level of each node based on its availability. Thus, we have three subnets in the SPN: Location Subnet, Energy Subnet, and Unavailability Subnet.

For other behaviors such as attack behaviors performing fake identity, fake recommendation dissemination, message modification, and packet dropping (i.e., related to competence trust), we use a probability for each behavior to model the particular behavior such as the percentage of compromised nodes (P_{cp}), the degree of attack intensity (P_a), and the degree of packet forwarding behavior (P_f). All attack behaviors considered will be triggered based on P_a which controls random attack patterns. Packet forwarding and insertion of PIs will be triggered based on P_f.

Location Subnet: This subnet computes the probability that node i is in a particular grid area k at time t. This information along with the information of other nodes' locations at time t enables us to calculate the encounter interval between two nodes. Since node movements are assumed to be independent, the probability that two nodes are in a particular location at time t is given by the product of the two individual probabilities. The transition T_LOC rate is computed as v_i/R_t where v_i is node i's average speed given and R_t is radio range.

Energy Subnet: This subnet is used to obtain each node's energy lifetime. The number of tokens in place *ENERGY*

indicates the battery life (hours) in energy. We approximately estimate energy consumption depending on a node's availability (see Unavailability Subnet below). When a node is not available, energy consumption is slowed down. The transition T_ENERGY is modeled by

$$rate(T\_ENERGY) = \begin{cases} \frac{1}{2T_e} \text{ if } mark(UNAVA) > 0 \\ \frac{1}{T_e} \text{ otherwise} \end{cases} \quad (6)$$

We assume that one token represents energy consumed for time period T_e for normal activities. When a node is in sleep mode or does not serve any request (i.e., unavailable status), it is predicted as consuming one half of normal energy consumption.

Unavailability Subnet: A token in place *UNAVA* indicates that node i is not available upon receiving a request; zero token otherwise. The rate for the transition T_UNAVA is affected by the probability of link unreliability based on network or node conditions (P_{ur}). Transition rate T_UNAVA is obtained as

$$rate(T\_UNAVA) = \frac{P_{ur}}{T_i^{enc}} \quad (7)$$

T_i^{enc} indicates the average inter-arrival time that node i encounters another node, and is computed by

$$T_i^{enc} = \sum_{j \in L} \frac{R_t}{(P_i^k P_j^{k'})(v_i + v_j)} \quad (8)$$

L is the set of legitimate members in the network where any j belongs to it. R_t is the radio range. P_i^k is the probability that node i is located at area k and $P_j^{k'}$ is the sum of probabilities that node j is located in the neighboring areas of k (i and j can communicate to each other within R_t), denoted as k'. v_i and v_j are the speeds of nodes i and j respectively. $rate(T\_RESET)$ is the rate for a token to be out, indicating availability of the node, and is computed by

$$rate(T\_RESET) = \frac{1}{T_i^{enc}} \quad (9)$$

From the SPN model above, we obtain behavioral seeds for availability, energy, and location information by using built-in reward assignment functions of SPN. The average availability of a node at time t, $P_v(t)$, is obtained by

$$P_v(t) = \sum_{i \in S} r_i^v P_i^v(t) \quad (10)$$

S is the set of all possible states that can be assumed by this particular node (for which the SPN subnet is running), r_i^v is the reward assignment to state i, and $P_i^v(t)$ is the probability that the system is in state i at time t. Thus, r_i^v is 1 when the condition $mark(UNAVA) == 0 \wedge mark(ENERGY) > 0$ is met; 0 otherwise.

Similarly, the average remaining energy at time t, $P_E(t)$, is obtained by

$$P_E(t) = \sum_{i \in S} r_i^E P_i^E(t) \quad (11)$$

r_i^E is 1 when $mark(ENERGY) > 0$; 0 otherwise.

The average probability node j is located at area k at time t is obtained by

$$P_j^k(t) = \sum_{i \in S} r_i^k P_i^{k'}(t) \quad (12)$$

r_i^k is 1 when $k == k'$ where k is $mark(LOC)$; 0 otherwise.

6 Experimental Results and Analysis

In this section, we explain our performance metrics, baseline schemes against which PROVEST is compared, experimental setup, and numerical results and analysis obtained from the experiments.

6.1 Metrics

We use the following metrics to evaluate performance of PROVEST against the baseline schemes:

- **Trust Bias** ($\mathcal{B}_X$) is the time-averaged difference between trust of node j evaluated by node i and objective trust of node j evaluated by all encountered nodes based on direct observations with no detection errors. The subjective trust assessment considers both false positives and negatives. Suppose L is a set of legitimate members in the network, the overall trust bias, $\mathcal{B}_X$, is computed by

$$\mathcal{B}_X = \frac{\sum_{t=0}^{LT} \mathcal{B}_X(t)}{LT} \quad (13)$$

$\mathcal{B}_X(t)$ is obtained by

$$\mathcal{B}_X(t) = \frac{\sum_{i,j \in L, i \neq j} \left| T_j^X(t) - T_{i,j}^X(t) \right|}{(|L| - 1)^2} \quad (14)$$

$T_{i,j}^X(t)$ is the trust value of node j on property X evaluated by node i at time t and $T_j^X(t)$ is the ground truth trust value of node j by using only direct evidence at time t, which is computed by

$$T_j^X(t) = \frac{r_j^X(t)}{r_j^X(t) + s_j^X(t)} \quad (15)$$

where $r_j^X(t)$ is the accumulated amount of positive evidence and $s_j^X(t)$ is the accumulated amount of negative evidence until time t where they can be found as features associated to the corresponding trust property, X of node j.
- **Mission message correctness** ($\mathcal{R}$) refers to the fraction of the number of packets received by DNs correctly over the total number of messages transmitted by SNs during LT. The trustworthiness of intermediate MCs significantly affects the correctness of received messages. This is computed by

$$\mathcal{R} = \frac{\sum_{m \in I} \prod_{k \in K} R_{k,m}(t)}{|I|} \quad (16)$$

$$R_{k,m}(t) = \begin{cases} 1 \text{ if MC } k \text{ did not modify } m \\ 0 \text{ otherwise} \end{cases}$$

Here I is a set of messages received by DNs and the k nodes are intermediate MCs delivering message m. K is a set of all intermediate MCs involved in delivering each message m.
- **Message delay** ($\mathcal{D}$) refers to the average delay occurred for a message to be delivered to a DN. This is obtained by

$$\mathcal{D} = \frac{\sum_{m \in I} D_m}{|I|} \quad (17)$$

D_m is the delay (*sec.*) occurred for message m to be delivered to the DN. I is a set of messages sent by SNs to DNs.

- **Communication cost** ($\mathcal{C}$) is the communication cost per time unit (*sec.*) in terms of the number of messages for a node to deal with trust evaluation ($C_e(t)$) and message delivery ($C_d(t)$) during the entire mission lifetime, LT. $\mathcal{C}$ is computed by

$$\mathcal{C} = \frac{\sum_{t=0}^{LT} C_e(t) + C_d(t)}{LT} \tag{18}$$

For ease of referencing, all notations used and their meanings are summarized in Table 1.

6.2 Schemes to be Evaluated

For the comparative performance analysis, we consider three trust-based schemes and two non-trust-based schemes. The former includes our PROVEST (and its variants), *Encounter-based* [2] and Iterative Trust Reputation Mechanism (ITRM) [43]. The latter includes epidemic (e.g., flooding) [25] and ProPHET (e.g., connectivity-based delivery prediction) [26] schemes. The details are discussed below.

PROVEST: The four variants of PROVEST, *PROVEST-Pessimistic*, *PROVEST-Optimistic*, *PROVEST-Realistic*, and *PROVEST-Hybrid*, as shown in Section 4 are tested and compared in terms of the four metrics above. Trust is used to select the next MC based on the minimum trust threshold T_{min}.

Encounter-based [2]: This scheme uses a Bayesian estimation of trust based on evidence collected upon an encountering event between two nodes. When two nodes encounter, they collect direct evidence based on observations to each other, and also exchange all other nodes' evidence observed for third party nodes as recommendations which are indirect evidence. This scheme is expected to show higher trust accuracy by paying a high communication cost. This scheme uses trust to select the next MC based on T_{min}.

Epidemic [25]: This scheme is like flooding so a node disseminates a message to whoever it encounters to ensure maximum delivery. This scheme is expected to incur high communication overhead but reaches maximum performance in message delivery and low delay.

PRoPHET [26]: This scheme selects the next MC based on the degree of connectivity estimated by historical mobility patterns. Message forwarding is performed based on opportunistic encounters where a node forwards a message to another node which has a high encountering probability based on historical mobility patterns. A node estimates another node j's delivery predictability that node j will deliver a message safely, denoted as $C_{i,j}(t)$. We set the initial value at time $t = 0$ with $C_{i,j}(0) = 0.5$. This probability is updated as i encounters other nodes over time as follows.

$$C_{i,j}(t) = \begin{cases} C_{i,j}(t-\Delta t) + (1 - C_{i,j}(t-\Delta t))P_{i,j}^{enc}(t) \text{ and} \\ C_{i,k}(t) = max\Big[C_{i,k}(t-\Delta t), C_{i,j}(t)C_{j,k}(t)\gamma\Big] \\ \text{ if } E(i,j) == 1 \\ e^{-\eta t}C_{i,j}(t-\Delta t) \text{ otherwise} \end{cases} \tag{19}$$

TABLE 2: Key Default Design Parameter Values

Parameter	Value	Parameter	Value
$\|L\|$	10, 15, 20	$\|M\|$	100, 200, 300
T_{min}	0-0.9	LT	99,000 sec.
v_i	$[1, 15]$ m/s	P_{cp}	0.1, 0.2, 0.3
P_r	0.05	P_{ur}	0.01
P_a	0.1-1	P_f	0.8
N_p	100	N_{SD}	20
N_c	2	$U[e, e']$	$U[12, 24]$ hrs.
η	0.01	γ	0.95
τ	0.1		

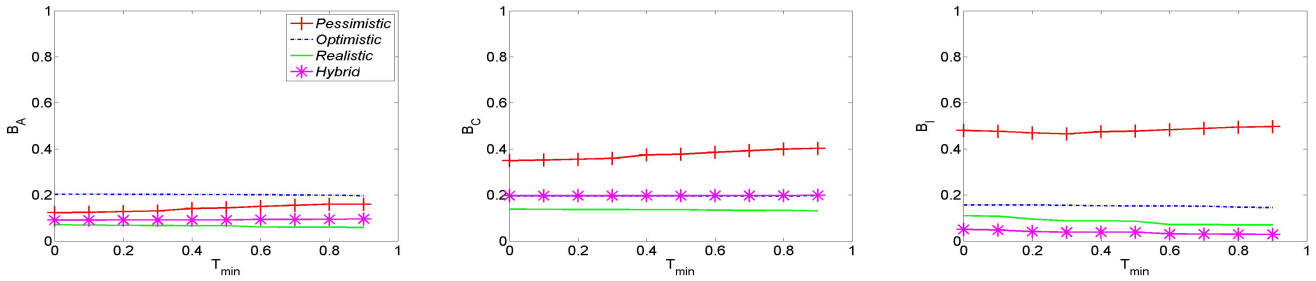
$E(i,j)$ returns 1 when i and j made a direct encounter; 0 otherwise. When $E(i,j) == 1$, i updates the degree of connectivity with j based on the encounter probability with j, denoted as $P_{i,j}^{enc}$, where Δt indicates the time elapsed since the last update of the delivery predictability. In addition, if j has a good path to k, i updates delivery predictability towards k using j's information towards k through the transitive rule. γ is a constant to apply some degree of decay introduced when using the transitive rule [26]. In PRoPHET, we use $C_{i,j}(t)$ to select the next MC based on T_{min}. $e^{-\eta t}$ is a decay factor when there is no new evidence to update the delivery predictability towards j.

Iterative Trust and Reputation Mechanism (ITRM) [43]: In this scheme, node i collects ratings from nodes (acting as raters) it encounters about node j and the rater that deviates the most from other raters is flagged as malicious. This process is applied whenever node i collects new ratings toward node j upon encountering a new rater. For fair comparison, we modify ITRM in the trust value scale (i.e., a real number in [0, 1]), consistency scale (i.e., a real number in [0, 1]), and trust bias metric (i.e., used Eq. 13, rather than mean absolute error in [43]). In addition, since two nodes may not often encounter in a sparse DTN, all nodes will play the dual role as a service provider or a rater. We use the notation τ to refer to the consistency threshold used by ITRM below which a node is considered malicious, i.e., the integrity trust is zero. It is possible that because of the experimental setting and the modification to ITRM, it may not result in a fair comparison between PROVEST and ITRM.

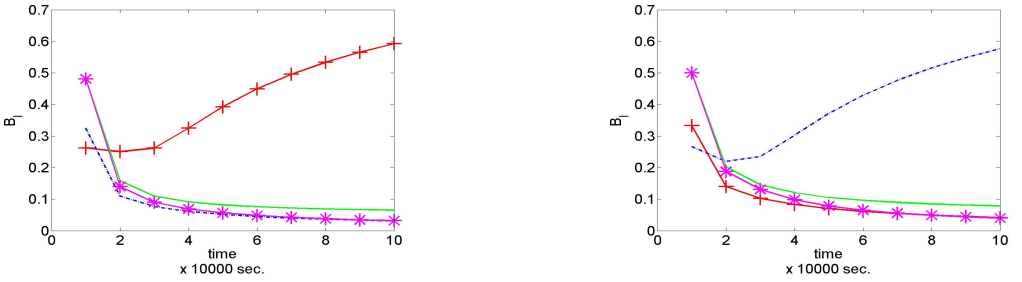
6.3 Experimental Setup

We conducted simulation experiments using Programming Language C based on data collected from our analytical model in SPNP, described in Section 5.

We summarize key design parameters and their default values used in Table 2. In the experiments, we use $n = 20$ nodes in the DTN where each node can communicate to each other within the radio range R_t which is set to 100m. The average speed of each node is randomly selected from uniform distribution with the range of $[1, 15]$. By default, packet forwarding behavior is seeded with a probability P_f meaning that a malicious node drops a packet with probability $1 - P_f$. The packet dropping behavior (i.e., black hole or gray hole attack) is considered when a node decides whether to forward a mission message (MM) with probability P_f. When a node forwards the MM, if the node



(a) Availability trust bias (B_A) (b) Competence trust bias (B_C) (c) Integrity trust bias (B_I)



(d) Integrity trust bias (B_I) for a good node (e) Integrity trust bias (B_I) for a bad node

Fig. 3: Trust Bias of PROVEST

is compromised and decides to perform an attack based on the given attack intensity, it may perform ID attack, either no-ID insertion or fake-ID insertion with equal probability. When a node is compromised and selected as a MC, it can modify the MM based on the given probability of attack intensity. When a malicious MC inserts its opinion towards the previous MC, it may provide false opinions to demote a good node's reputation or to promote a bad node's reputation. The percentage of compromised nodes (i.e., exhibiting attack behaviors described in Section 3.3) over all nodes is described by a probability parameter P_{cp}. Among the compromised nodes, the probability of their attack intensity is described by a probability parameter, P_a.

The number of source-destination pairs is $N_{SD} = 20$. The number of messages each SN transmits to a DN, N_p, is set 100. The initial energy is given randomly from uniform distribution with the range of $[12, 24]$. We summarize key design parameters and their default values used in Table 2. In the following section, we conduct a sensitivity analysis of the percentage of compromised nodes, P_{cp}, and the degree of attack intensity, P_a.

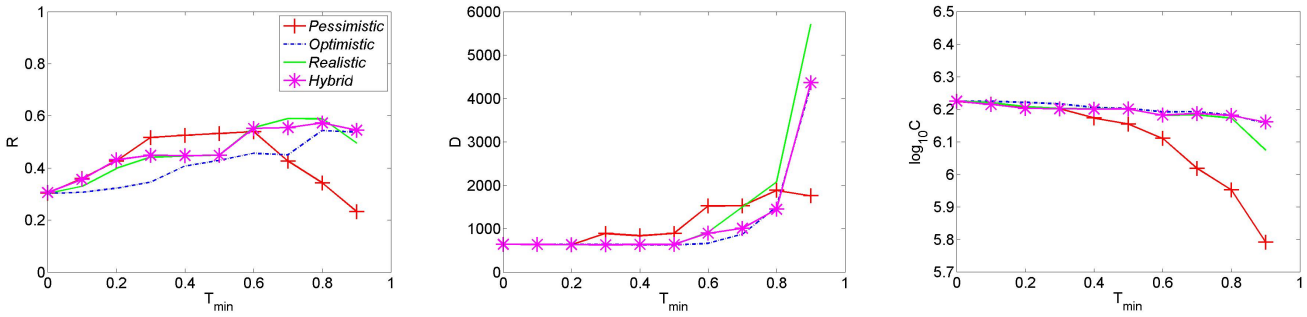
In Section 6.6, we use the CRAWDAD dataset of real human mobility traces based on daily GPS track log collected from Disney World, Florida, USA every 30 seconds [46]. The total 41 traces are collected in the dataset; we randomly selected 20 traces. The operational area is set to 600m $\times$ 600m in order for each node to encounter other nodes based on the same transmission range used in the SPN model with $R = 100$m based on 802.11n to allow for long-range communication, given the sparsity of node density in the target DTN environment. Since the mission duration (i.e., 99,000 sec.) used in this work is much longer than mobility traces dataset (i.e., 14,400 sec.), we calibrate the encounter intervals of any two nodes based on the available dataset and plug them into our simulation. We use the mean of measurements of 100 simulation runs to generate the experimental results.

In the following sections, we first perform a performance analysis of PROVEST. Then we perform a comparative performance analysis against other baseline schemes.

6.4 PROVEST Performance Analysis

In this section, we demonstrate the bias introduced in measuring availability, integrity, and competence trust based on Eq. 13. We also compare performance of the four variants of PROVEST using the three metrics in Section 6.1. We show the experimental results for the case in which the percentage of compromised nodes is 20% ($P_{cp} = 0.2$) and each attacker exhibits malicious behavior whenever possible (i.e., $P_a = 1$).

We show the average bias in measured availability, integrity, and competence trust versus the trust threshold, T_{min} in Figs. 3 (a)-(c). In addition, we show the evolution of trust bias over time where a trustee is a good node or bad node in Figs. 3 (d) and (e), respectively. Recall that low trust bias is desirable, implying less difference between ground truth trust and measured trust. In Figs. 3 (a)-(c), we observe that *PROVEST-Realistic* performs the best for availability and competence trust while *PROVEST-Hybrid* performs the best in integrity trust. In particular, the accurate estimation of integrity trust is critical to successful correct message delivery which we will report later in Fig. 4 (a). We also observe that higher T_{min} somewhat reduces integrity trust bias in all schemes except for *PROVEST-Pessimistic*. In Figs. 3 (d)-(e), we show the evolution of trust bias in integrity trust as it is critical to detecting a malicious node in the network. As expected, in estimating a good node's trust, *PROVEST-Pessimistic* performs significantly worse than other counterparts because it takes uncertain evidence as negative evidence. On the other hand, when estimating a bad node's



(a) Fraction of correct messages received ($\mathcal{R}$) (b) Message delay ($\mathcal{D}$) (c) Total communication cost ($\mathcal{C}$)

Fig. 4: Performance of PROVEST

trust, *PROVEST-Optimistic* performs badly showing significantly high trust bias. *PROVEST-Hybrid* performs fairly well in both cases because it determines how to deal with uncertain evidence based on the historical patterns observed.

In Fig. 4, we compare all trust-based schemes in terms of the fraction of correct messages received ($\mathcal{R}$), message delay ($\mathcal{D}$), and total communication cost ($\mathcal{C}$). In $\mathcal{R}$, *Realistic* performs fairly well with low T_{min} (i.e., < 0.6) while *Realistic* and *Hybrid* perform well with high T_{min}(i.e., > 0.6) because in *Pessimistic* uncertain but actually trustworthy evidence is treated as negative where only message carriers with the trust value above the high trust threshold are allowed to deliver a message. In $\mathcal{D}$, *Optimistic* and *Hybrid* perform fairly well unless T_{min} is too high (i.e., < 0.8) because trustworthy nodes which were underestimated due to uncertain evidence can be selected as a message carrier so a message has a better chance to be delivered to a destination node with more nodes involved. In $\mathcal{C}$, we observe that a higher message delivery rate can also lead to higher communication overhead. Except for *Pessimistic* which does not trust many nodes due to its pessimistic trust estimation, all schemes incur equally high communication overhead.

6.5 Comparative Performance Analysis

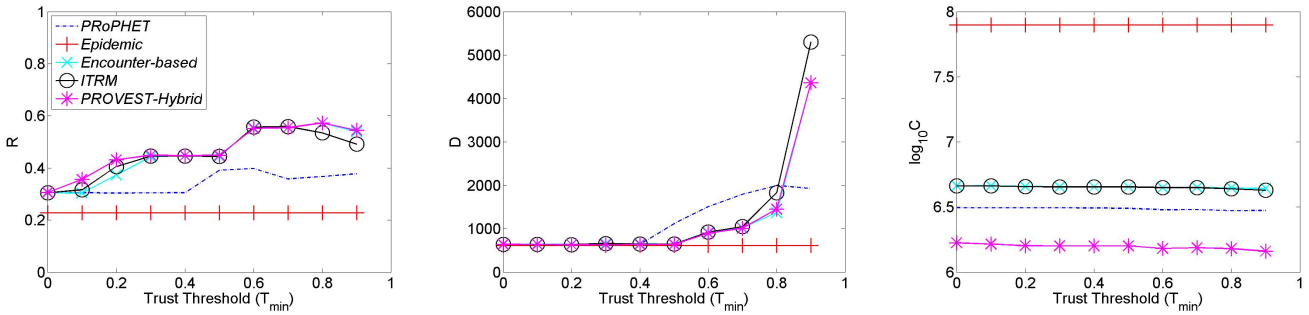
This section compares the performance of *PRoPHET*, *Epidemic*, *Encounter-based*, *ITRM*, and *PROVEST-Hybrid*. We use *PROVEST-Hybrid* to represent PROVEST due to its advantage, compared to other variants of PROVEST. Note that the average path length from a source to a destination is mainly affected by the routing protocol being used and the mobility patterns of nodes which impact the dynamics of network topology. We conduct a sensitivity analysis of the performance comparison results with respect to the trust threshold (T_{min}), percentage of compromised nodes (P_{cp}), and attack intensity (P_a). The default values used are $T_{min} = 0.6$, $P_{cp} = 0.2$, and $P_a = 1$ in the comparative performance analysis.

Fig. 5 shows the performance comparison of the five schemes in terms of $\mathcal{R}$, $\mathcal{D}$, and, $\mathcal{C}$ as trust threshold T_{min} varies. We notice that overall *PROVEST-Hybrid* performs the best in $\mathcal{R}$ while it incurs the least cost in $\mathcal{C}$. When T_{min} is in the range of $0.3 - 0.7$, *PROVEST-Hybrid*, *Encounter-based*, and *ITRM* perform comparably in $\mathcal{R}$ and $\mathcal{D}$, while *PROVEST-Hybrid* outperforms *Encounter-based* and *ITRM* in $\mathcal{C}$. Notice that *Epidemic* does not maximize its performance in $\mathcal{R}$ while incurring the highest $\mathcal{C}$ due to the nature of forwarding a packet to all neighbors including malicious nodes which may drop or modify messages. Therefore, *Epidemic* guarantees the maximum message delivery (not shown here due to space constraint) but does not ensure the maximum delivery of 'correct' messages.

In Fig. 6, we conduct a sensitivity analysis of the percentage of compromised nodes, P_{cp}. As P_{cp} increases, $\mathcal{R}$ decreases as there are more attackers in the network, which implies more damages under a more hostile environment. However, for $\mathcal{D}$ and $\mathcal{C}$, we don't observe higher damage under higher hostility. For $\mathcal{D}$, we observe a relatively shorter delay when $P_{cp} = 0.2$ than $P_{cp} = 0.1$ or 0.3. This is because under a less hostile environment (i.e., $P_{cp} = 0.1$), more messages are delivered thus there is more traffic existing in the network which increases the delay per message delivery. On the other hand, when the hostility is at a medium level, (i.e., $P_{cp} = 0.2$), attackers drop packets which reduces data traffic while generating some bogus traffic. Under a more hostile environment (i.e., $P_{cp} = 0.3$), more packets are dropped but attackers generate more bogus traffic, thereby causing a slightly increasing delay, but it does not exceed the delay under $P_{cp} = 0.1$. Similarly, for $\mathcal{C}$, we observe the tradeoff between more traffic by more delivered messages by legitimate nodes and more bogus traffic generated by attackers, leading to somewhat insensitivity of $\mathcal{C}$ over varying P_{cp} (i.e., mostly flat).

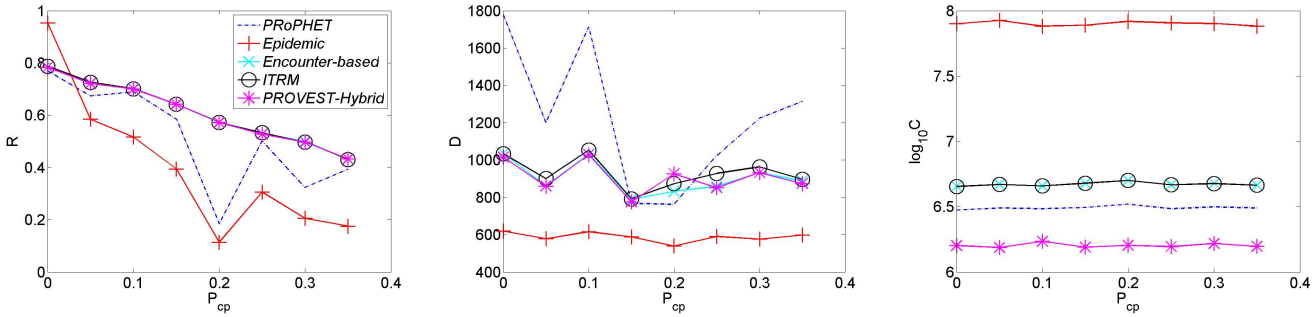
We observe that the three trust-based schemes, i.e., *PROVEST-Hybrid*, *ITRM*, and *Encounter-based*, perform comparably in $\mathcal{R}$. When P_{cp} is in the range of $[0 - 0.15, 0.25 - 0.35]$, *PROVEST-Hybrid* has a slight advantage over *ITRM* and *Encounter-based* in $\mathcal{D}$. When P_{cp} is in the range of $[0.15 - 0.25]$, *Encounter-based* has a slight advantage over *ITRM* and *PROVEST-Hybrid* in $\mathcal{D}$. In general, all three trust-based schemes perform comparably well in $\mathcal{D}$. The most striking result is that *PROVEST-Hybrid* consistently outperforms *ITRM* and *Encounter-based* in $\mathcal{C}$ over a wide range of P_{cp} values.

In Fig. 7, we investigate the effect of varying the attack intensity, P_a, on performance. Similar to Fig. 6, $\mathcal{R}$ is sensitive, while $\mathcal{D}$ and $\mathcal{C}$ are relatively insensitive to P_a. The reasons are similar to what we discussed in Fig. 6 above. *PROVEST-Hybrid*, *ITRM*, and *Encounter-based* perform com-



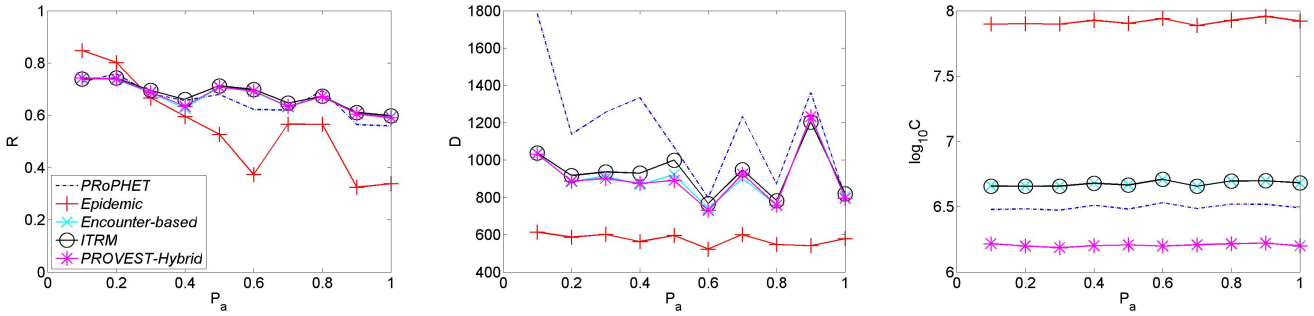
(a) Fraction of correct messages received ($\mathcal{R}$) (b) Message delay ($\mathcal{D}$) (c) Total communication cost ($\mathcal{C}$)

Fig. 5: Performance Comparison: Effect of Trust Threshold T_{min}



(a) Fraction of correct messages received ($\mathcal{R}$) (b) Message delay ($\mathcal{D}$) (c) Total communication cost ($\mathcal{C}$)

Fig. 6: Performance Comparison: Effect of Percentage of Compromised Nodes P_{cp}



(a) Fraction of correct messages received ($\mathcal{R}$) (b) Message delay ($\mathcal{D}$) (c) Total communication cost ($\mathcal{C}$)

Fig. 7: Performance Comparison: Effect of Attack Intensity P_a

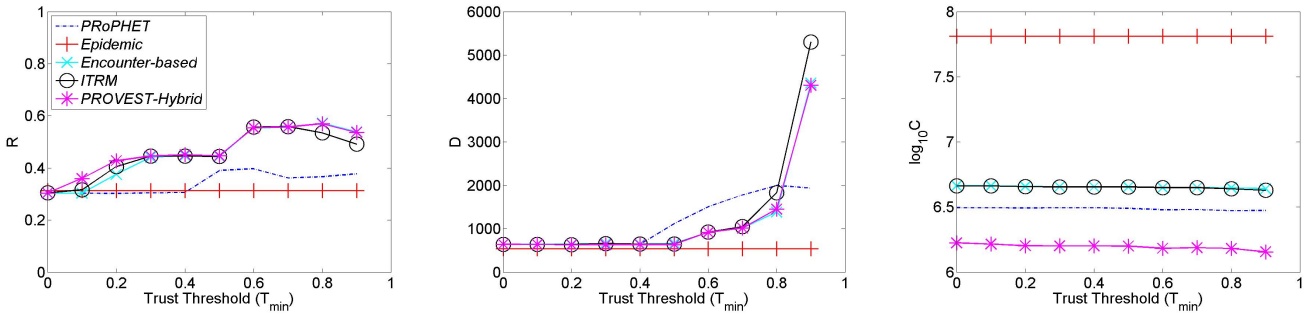
parably in $\mathcal{R}$ and while *PROVEST-Hybrid* has a slight edge over *ITRM* and *Encounter-based* in $\mathcal{D}$. *PROVEST-Hybrid* introduces significantly low communication overhead because of saving the communication overhead for exchanging the trust table when two nodes encounter.

Summarizing above, we conclude that *PROVEST-Hybrid* significantly reduces the communication cost while maintaining a high correct message delivery ratio, compared to *Epidemic*, *ITRM*, *Encounter-based*, and *PRoPHET*.

6.6 Validation using Real Mobility Traces

To validate the simulation results which are based on random mobility, we repeat the same set of experiments in Fig. 5 using real mobility traces provided by CRAWDAD [46] with 20 nodes randomly selected.

The trends observed in Fig. 8 are remarkably close to those in Fig. 5. The trend exhibited in Fig. 8 based on real mobility traces supports our conclusion that *PROVEST-Hybrid* for over a wide range of T_{min} is either better than or as good as $ITRM$ and *Encounter-based* in $\mathcal{R}$ and $\mathcal{D}$, outperforms *Epidemic* and *PRoPHET* in $\mathcal{R}$ and $\mathcal{D}$, and outperforms all in $\mathcal{C}$, generating significantly low communication overhead.



(a) Fraction of correct messages received ($\mathcal{R}$) (b) Message delay ($\mathcal{D}$) (c) Total communication cost ($\mathcal{C}$)

Fig. 8: Performance Comparison with Real Mobility Traces

7 Conclusion

In this work, we proposed a provenance-based trust model called PROVEST which evaluates trust of a node by leveraging the provenance information added by each intermediate message carrier as indirect evidence during message forwarding. In developing PROVEST, we devised four variants namely, *PROVEST-Pessimistic*, *PROVEST-Optimistic*, *PROVEST-Realistic*, and *PROVEST-Hybrid*, to deal with uncertainty caused by unavailable or uncertain evidence. In particular, *PROVEST-Hybrid* combines the benefits of *PROVEST-Pessimistic* and *PROVEST-Optimistic*, and performs adaptive control based on the historical pattern of evidence such as positive or negative evidence. This feature excels in identifying bad nodes in the network where trust evidence is uncertain.

We conducted comprehensive experiments using the hierarchical SPN model using random mobility as well as real mobility traces from CRAWDAD [46]. We compared the performance of PROVEST with *Epidemic* [25], *Encounter-based* [2], *ITRM* [43] and *PRoPHET* [26]. We found that the provenance-based approach (i.e., *PROVEST-Hybrid*) significantly reduces the communication cost while maintaining a high correct message delivery ratio, compared to *Epidemic*, *ITRM*, *Encounter-based*, and *PRoPHET*.

As future work directions, we are investigating how to refine the trust dimensionality in our protocol design to further minimize trust bias in order to achieve higher performance in message delivery. In addition, we plan to examine the applicability of provenance-based trust-enhanced security mechanisms for military settings such as access control and/or intrusion detection.

Acknowledgments

This work is supported in part by the U. S. Army Research Laboratory and the U. S. Army Research Office under contract number W911NF-12-1-0445. This research was also partially supported by the Department of Defense (DoD) through the office of the Assistant Secretary of Defense for Research and Engineering (ASD (R&E)). The views and opinions of the author(s) do not reflect those of the DoD or ASD (R&E).

References

[1] T. Spyropoulos, R. Rais, T. Turletti, K. Obraczka, and A. Vasilakos, "Routing for disruption tolerant networks: taxonomy and design," *Wireless Networks*, vol. 16, no. 8, pp. 2349–2370, 2010.

[2] I.-R. Chen, F. Bao, M. Chang, and J.-H. Cho, "Trust management for encounter-based routing in delay tolerant networks," in *IEEE Global Telecommunications Conference*, Miami, FL, 6-10 Dec. 2010, pp. 1–6.

[3] ——, "Dynamic trust management for delay tolerant networks and its application to secure routing," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 5, pp. 1200–1210, May 2014.

[4] P. Buneman, S. Khanna, and W. Tan, "Why and where: A characterization of data provenance," in *Proceedings of International Conference on Database Theory, Springer-Verlag*, 2001, pp. 316–330.

[5] J.-H. Cho, M. Chang, I.-R. Chen, and A. Swami, *Trust Management VI, IFIP Advances in Information and Communication Technology*. 6th IFIPTM, Surat, India: Springer, 2012, vol. 374, ch. A Provenance-based Trust Model of Delay Tolerant Networks, pp. 52–67.

[6] A. Jøsang and R. Ismail, "The beta reputation system," in *Bled Electronic Commerce Conference*, Bled, Slovenia, 17-19 June 2002, pp. 1–14.

[7] M. Mahmoud, X. Lin, and X. Shen, "Secure and reliable routing protocols for heterogeneous multihop wireless networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, no. 4, pp. 1140–1153, March 2015.

[8] L. Moreau, J. Freire, J. Futrelle, R. McGrath, J. Myers, and P. Paulson, "The open provenance model: an overview," in *International Provenance and Annotation Workshop, LNCS*, vol. 5272, Salt Lake City, Utah, 17-18 June 2008, pp. 323–326.

[9] Y. Liu, J. Futrelle, J. Myers, A. Rodriguez, and R. Kooper, "A provenance-aware virtual sensor system using the open provenance model," in *International Symposium on Collaborative Technologies and Systems*, Chicago, IL, 17-21 May 2010, pp. 330–339.

[10] J. Freire, D. Koop, E. Santos, and C. Silva, "Provenance for computational tasks: A survey," *IEEE Computing in Science and Engineering*, vol. 10, no. 3, pp. 11–21, 2008.

[11] P. McDaniel, "Data provenance and security," *IEEE Security and Privacy*, vol. 9, no. 2, pp. 83–85, 2011.

[12] S. Rajbhandari, I. Wootten, A. Ali, and O. Rana, "Evaluating provenance-based trust for scientific workflows," in *6th IEEE International Symposium on Cluster Computing and the Grid*, vol. 1, Singapore, 16-19 May 2006, pp. 365–372.

[13] E. B. C. Dai, Dan Lin and M. Kantarcioglu, "An approach to evaluate data trustworthiness based on data provenance," in *Proceedings of 5th VLDB Workshop on Secure Data Management, Lecture Note in Computer Science*, vol. 5159, Auckland, New Zealand, Aug. 2008, pp. 82–98.

[14] B. Yu, S. Kallurkar, and R. Flo, "A demspter-shafer approach to provenance-aware trust assessment," in *International Symposium on Collaborative Technologies and Systems*, Irvine, CA, May 2008, pp. 383–390.

[15] J. Golbeck, "Combining provenance with trust in social networks for semantic web content filtering," *Provenance and Annotation of Data, LNCS*, vol. 4145, pp. 101–108, 2006.

[16] W. Zhou, E. Cronin, and B. T. Loo, "Provenance-aware secure networks," in *IEEE 24th International Conference on Data Engineering Workshop*, 2008, pp. 188–193.
[17] R. Hasan, R. Sion, and M. Winslett, "Introducing secure provenance: problems and challenges," in *ACM Workshop on Storage Security and Survivability*, 2007, pp. 13–18.
[18] U. Braun, A. Shinnar, and M. Seltzer, "Securing provenance," in *Proceedings of the 3rd Conference on Hot Topics in Security*, 2008, pp. 1–5.
[19] R. Hasan, R. Sion, and M. Winslett, "The case of the fake picasso: preventing history forgery with secure provenance," in *Proccedings of the 7th Conference on File and Storage Technologies*, 2009, pp. 1–14.
[20] X. Wang, K. Zeng, K. Govindan, and P. Mohapatra, "Chaining for securing data provenance in distributed information networks," in *IEEE Military Communications Conference*, 2012, pp. 1–6.
[21] L. Gadelha and M. Mattoso, "Kairos: An architecture for securing authorship and temporal information of provenance data in grid-enabled workflow management systems," in *IEEE 4th International Conference on eScience*, 2009, pp. 597–602.
[22] R. Lu, X. Lin, X. Liang, and X. Shen, "Secure provenance: the essential of bread and butter of data forensics in cloud computing," in *ACM Symposium on Information, Computer and Communications Security*, 2010, pp. 282–292.
[23] S. Alam and S. Fahmy, "Energy-efficient provenance transmission in large-scale wireless sensor networks," in *IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks*, Lucca, Italy, 20-23 June 2011.
[24] H. Lim, Y.-S. Moon, and E. Bertino, "Provenance-based trustworthiness assessment in sensor networks," in *Proceedings of 7th International Workshop on Data Management for Sensor Networks*, Singapore, Singapore, 13 Sept. 2010.
[25] A. Vahdat and D. Becker, "Epidemic routing for partially connected ad hoc networks," Duke University, Tech. Rep., 2000.
[26] A. Lindgren, A. Doria, and O. Schelén, "Probabilistic routing in intermittently connected networks," in *LNCS*, ser. Service Assurance with Partial and Intermittent Resources(SAPIR) 2004, P. D. et al., Ed. Orlando, FL: Springer-Verlag Berlin Heidelberg, 19-22 Oct. 2008, pp. 218–227.
[27] U. Lee, S. Y. Oh, K.-W. Lee, and M. Gerla, "Relaycast: Scalable multicast routing in delay tolerant networks," in *IEEE International Conference on Network Protocols*, 2008, pp. 218–227.
[28] V. Conan, J. Leguay, and T. Friedman, "Fixed point opportunistic routing in delay tolerant networks," *IEEE Journal on Selected Areas in Communications*, vol. 26, no. 5, pp. 773–782, June 2008.
[29] H. Dang and H. Wu, "Clustering and cluster-based routing protocol for delay-tolerant mobile networks," *IEEE Transactions on Wireless Communications*, vol. 9, no. 6, pp. 1874–1881, June 2010.
[30] E. Jones, L. Li, J. Schmidtke, and P. Ward, "Practical routing in delay-tolerant networks," *IEEE Transactions on Mobile Computing*, vol. 6, no. 8, pp. 943–959, Aug. 2007.
[31] M. Musolesi and C. Mascolo, "Car: Context-aware adaptive routing for delay-tolerant mobile networks," *IEEE Transactions on Mobile Computing*, vol. 8, no. 2, pp. 246–260, 2009.
[32] C. Boldrini, M. Conti, J. Jacopini, and A. Passarella, "Hibop: A history based routing protocol for opportunistic networks," in *IEEE International Symposium on World of Wireless, Mobile and Multimedia Networks (WoWMoM 07)*, June 2007, pp. 1–12.
[33] K. Wei, X. Liang, and K. Xu, "A survey of social-aware routing protocols in delay tolerant networks: applications, taxonomy and design-related issues," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 1, pp. 556–578, 2014.
[34] Y. Zhu, B. Xu, X. Shi, and Y. Wang, "A survey of social-based routing in delay tolerant networks: Positive and negative social effects," *IEEE Communications Surveys & Tutorials*, vol. 15, no. 1, pp. 387–401, 2013.
[35] P. Costa, C. Mascolo, M. Musolesi, and G. Picco, "Socially-aware routing for publish-subscribe in delay-tolerant mobile ad hoc networks," *IEEE Journal on Selected Areas in Communications*, vol. 26, no. 5, pp. 748–760, 2008.
[36] Y. Li, P. Hui, D. Jin, L. Su, and L. Zeng, "Evaluating the impact of social selfishness on the epidemic routing in delay tolerant networks," *IEEE Communications Letters*, vol. 24, no. 12, pp. 2472–2481, 2010.
[37] W. Gao and G. Cao, "User-centric data dissemination in disruption tolerant networks," in *2011 Proceedings IEEE INFOCOM*, Shanghai, China, 10-15 April 2011, pp. 3119–3127.
[38] L. Gao, M. Li, A. Bonti, W. Zhou, and S. Yu, "Multidimensional routing protocol in human-associated delay-tolerant networks," *IEEE Transactions on Mobile Computing*, vol. 12, no. 11, pp. 2132–2144, 2013.
[39] Y. Wang, W.-S. Yang, and J. Wu, "Analysis of a hypercube-based social feature multipath routing in delay tolerant networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 9, pp. 1706–1716, 2013.
[40] T. Abdelkader, K. Naik, A. Nayak, N. Goel, and V. Srivastava, "SGBR: A routing protocol for delay tolerant networks using social grouping," *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 12, pp. 2472–2481, 2013.
[41] Z. Li and H. Shen, "SEDUM: Exploiting social networks in utility-based distributed routing for dtns," *IEEE Transactions on Computers*, vol. 62, no. 1, pp. 83–97, 2013.
[42] H. Zhu, S. Du, Z. Gao, M. Dong, and Z. Cao, "A probabilistic misbehavior detection scheme toward efficient trust establishment in delay-tolerant networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 1, pp. 22–32, 2014.
[43] E. Ayday and F. Fekri, "An iterative algorithm for trust management and adversary detection for delay-tolerant networks," *IEEE Transactions on Mobile Computing*, vol. 11, no. 9, pp. 1514–1531, 2012.
[44] S. Kamvar, M. Schlosser, and H. Garcia-Molina, "The EigenTrust algorithm for reputation management in P2P networks," in *Proceedings of 12th International Conference World Wide Web (WWW '03)*, 2003, pp. 640–651.
[45] S. Buchegger and J. Boudec, "A robust reputation system for P2P and mobile ad-hoc networks," in *Proceedings of Second Workshop the Economics of Peer-to-Peer Systems*, 2004.
[46] Dartmouth University, "Mobility traces data from CRAWDAD (a community resource for archiving wireless data at dartmouth)," *http://crawdad.org/thlab/sigcomm2009/*.
[47] M. Steiner, G. Tsudik, and M. Waidner, "Key agreement in dynamic peer groups," *IEEE Transactions on Parallel and Distributed Systems*, vol. 11, no. 8, pp. 769–780, Aug. 2000.
[48] A. Perrig and J. Tygar, *Secure Broadcast Communication in Wired and Wireless Networks*. Kluwer Academic Publishers, 2002.

Jin-Hee Cho received the MS and PhD degrees in computer science from the Virginia Tech. She is currently a computer scientist at the U.S. Army Research Laboratory (USARL), Adelphi, Maryland. Her research interests include network security, trust and risk management, cognitive modeling, and network science. She received the best paper awards in IEEE TrustCom09 and BRIMS13. She is a recipient of the 2015 IEEE Communications Society William R. Bennett Prize in the Field of Communications Networking. She is a senior member of the IEEE and a member of ACM.

Ing-Ray Chen received the BS degree from the National Taiwan University, Taipei, Taiwan, and the MS and PhD degrees in computer science from the University of Houston. He is a professor in the Department of Computer Science at Virginia Tech. His research interests include mobile computing, wireless systems, security, trust management, data management, real-time intelligent systems, and reliability and performance analysis. Dr. Chen currently serves as an editor for IEEE Communications Letters, IEEE Transactions on Network and Service Management, Wireless Personal Communications, Wireless Communications and Mobile Computing, The Computer Journal, Security and Network Communications, and International Journal on Artificial Intelligence Tools. He is a member of the IEEE and ACM.