A Taxonomic Study of the Recent Security Concerns in Opportunistic Networks

Smritikona Barai[1] Dr. Parama Bhaumik[2]
[1,2]Department of Information Technology
[1]Heritage Institute of Technology, Kolkata [2]Jadavpur University, Kolkata

Abstract— Opportunistic networks contain mobile nodes with wireless networking capability which deliver messages in a hop-by-hop fashion completely based on local information and using opportunistic connectivity created due to device proximity. There exists no predefined path between the source and destination. The source only forwards the data to the next node based on its capability to carry the data packet one step closer to the destination. Due to the absence of any concrete path connected by trustful nodes, these types of networks suffer from a number of security issues. The intermediate nodes may be malicious and lead to loss of packets, breach of user privacy, compromising data confidentiality by unwanted intrusion etc. General cryptography algorithms cannot be applied here directly due to the absence of any end-to-end secure path. The nodes have to solely depend on the next hop node for data transfer. So there is a need of such security mechanisms which would ensure that the intermediate nodes do not behave maliciously. This survey is an attempt to study and compare some security techniques being applied to Opportunistic Networks so far.

Key words: Pattern Classification, Web Mining, C 5.0 Algorithm

I. INTRODUCTION

Over the last decade, the advances in wireless technologies have led to the evolution of the generic Mobile Ad Hoc Network (MANET) into a network referred to as Opportunistic Networking. Opportunistic networks initially contain the source node of data packet (seed node). It then invites other nodes to join its network who are ready to carry forward its data towards the destination (helper nodes). The helper nodes can store and carry a packet until a next hop node is encountered, unlike MANET, where the packet is dropped by the node. Thus Oppnets are disconnected as a rule rather than an exception and mobility plays a vital role in determining its performance. Such systems are likely to experience prolonged delays in the absence of mobile devices in proximity and hence the applications should be resistant to large delays.

Oppnets can be very efficient and useful in disaster prone areas, battlefields, wildlife monitoring networks etc where connectivity is frequently intermittent and delays are tolerable. Moreover, establishing an opportunistic network merely needs some mobile nodes such as mobile phones, PDAs etc. which are now carried by almost every person. Thus in an era when Green Computing1 is a burning issue, setup of opportunistic networks can be considered as environment-friendly. There is no need for any additional infrastructural support like external antennas, wireless access points etc as the existing resources can be put to use, thus reducing the radiations from these large devices and hence the environmental impact.

But in spite of being so useful, their inherent structure makes them very prone to different safety issues. Oppnets are highly heterogeneous with nodes having different processing capabilities, power sources, modes of transmission etc. Therefore it is difficult to guarantee that malicious nodes would never join an oppnet and disrupt its desirable behavior. Delivering secret keys securely to the non-malicious devices only is another challenge in such an ad hoc environment. Hence, relying alone on cryptography-based authentication mechanisms (e.g., Kerberos) will not help in all situations [22]. So the existing security protocols applicable to wireless and ad hoc networks cannot be applied directly to the oppnets.

According to [22], ways to deal with the security issues of oppnets can be summarized as (i) increasing trust and secure routing, (ii) ensuring helper and oppnet privacy, (iii) protecting data privacy, (iv) ensuring data integrity, (v) identifying dangerous attacks like id spoofing, packet dropping, DoS attacks etc. and (vi) intrusion detection.

But so far in our knowledge there has been no concise literature dealing with all these security issues in a formal manner. So, in this survey we have attempted to capture the state of the art of the current oppnet security issues in a formal taxonomic approach, along with their possible solutions that have been proposed so far.

II. SECURITY PROTOCOLS

The security mechanisms in opportunistic networks are a relatively unexplored and challenging area of research. The risk of having a node's privacy violated may prevent potential helper nodes from joining an oppnet or helping in data forwarding. In this paper, we have surveyed some approaches to minimize the privacy leakage during opportunistic networking. Some of these protocols use cryptographic tools. Cryptographic primitives help in data forwarding, keeping information hidden through the main flow but it comes at the cost of a more complicated communication protocol. Comparatively, protocols without cryptographic tools proved to be more efficient and appropriate for mobile devices in terms of computational overhead.

Similarly, ensuring trust among nodes in an Oppnet is another important security issue. Malicious nodes may join the network to disrupt its work by dropping the packets instead of routing them to the next hop. So, the trustworthiness of each node becomes a significant metric in determining whether a particular node should be chosen as the next hop or not.

In this survey, we have introduced a taxonomy based on the security issues in oppnets and their proposed solutions. They have been categorized according to the above discussed criteria, such as use of cryptographic tools, the basic underlying concepts, the protocols that they implement and specific application scenarios.

As shown in Figure 1, the Security mechanisms in oppnets have been broadly grouped as Trust based and Privacy based protocols. Depending on the basic concept used to solve the issue, the trust based protocols have been further categorized as Friend vector based, Familiarity based, Reputation based and Hybrid trust models, each model being used in a number of frameworks. Similarly, the Privacy based models have been further categorized as Cryptography based and Cryptography free protocols. A number of protocols using or not using cryptography have been discussed under them.

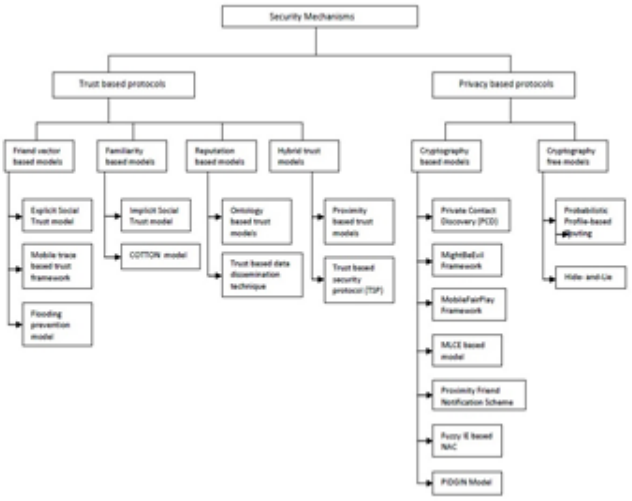

Fig. 1:Taxonomy to categorize the security issues in Oppnets

The next subsections are dedicated to the detailed description of the proposed taxonomy. Section III describes the protocols proposed to ensure privacy in Oppnets. Section IV presents the Trust-based protocols to ensure security in oppnets. Section V concludes the paper by summarizing the different forms of security mechanism, their pros, cons and the further research scopes in this field.

III. PROTOCOLS TO ENSURE DATA PRIVACY

As already discussed, privacy is an important security issue in Oppnets. In a highly mobile and heterogeneous network like oppnet, ensuring the privacy of the user's identity, location and data confidentiality is quite challenging. Data confidentiality can be provided by encryption using secret shared key known only to the seed nodes (source/destination). But as described in [29], a seed node is always required in the vicinity of each intermediate node in order to encrypt the intermediate node's id into the authentication message. But owing to the nodes' highly dynamic nature, this situation may not be always guaranteed.

Based on the underlying technique that is used to preserve the privacy of data in oppnets, some proposed solutions can be roughly categorized into Cryptography-Based and Cryptography-Free protocols [35].

A. *Cryptography-Based Protocols*

Cryptography-based approaches challenge the limited computational and/or storage resources available on existing mobile devices; yet, they provide stronger privacy preservation guarantees than cryptography-free approaches. Some cryptography-based protocols designed and implemented on mobile platforms are discussed below:

1) Private Contact Discovery (Pcd)

A Private Contact Discovery primitive was proposed by [9] that enable two users, on input of their contact lists, to know their common contacts (if any) but keeping the non-common contacts hidden from each other. This protocol also prevents users from claiming unwarranted friendships by introducing contact certification. For e.g. X needs a certificate from Y authenticating the friendship if she wants to include Y in her contact list.

PCD is based on the Index-Based Message Encoding (IBME) cryptographic tool which combines a set of indexed input messages into a single data structure. IBME fulfils a security property called Index-Hiding Message Encoding (IHME) which ensures that no adversary, by observing the IBME structure, will be able to learn any useful information about the deployed indices, even if the adversary knows some of the indices and/or messages.

A Contact Discovery Scheme (CDS), at the heart of PCD, is defined as a tuple of four algorithm and protocols: (1) Init(1k) is the parameter initialization for a generic user U; (2) AddContact(U↔V) is run by users U and V, when user V wishes to become a contact in user U's list; (3) RevokeContact(U,V) is a function, in which the identity of V is revoked from U's contact list; (4) Discover (V↔V') algorithm is executed by V and V' to discover whether they have common friends. At the end of the algorithm, both users know whether they have common friends, while not disclosing the private contact list to the other party.

The Discover algorithm uses Okamoto's technique [26] for RSA-based identity-based key agreement. All transferred messages are IMHE-encoded into a single structure before transmission. On receiving the IMHE messages, each of them is decoded and the probabilistic padding applied before it is removed. Then a new round of message exchanges encoded with the IHME is performed. Finally, each common user found is added to a list, and the algorithm ends with either "accept" or "reject" depending on whether the list size is larger than zero.

2) Mightbeevil Framework

The MightBeEvil Framework was proposed in [7] to run a secure-two party computation (STC) in mobile environment. STC involves two parties, each holding some private data, say x and y. The goal of STC is to jointly compute the outcome of a function f(x,y) , without disclosing the input of one party to the other party. At the end of the execution, both parties know the outcome of f(x,y) but they do not know each other's input. The general idea of MightBeEvil Framework is to create a sort of "encrypted circuit" for the function f and then to obliviously compute the output of the circuit without learning any intermediate value. Compared to PCD primitive, the MightBeEvil framework is supposedly a more flexible system using similar friend discovery mechanism.

This model was designed to counter the semi-honest threat model. A semi-honest attacker is one who follows the protocol but may attempt to learn additional information about the other party's input. The authors claim that this framework achieves security against malicious attackers by adopting an oblivious-transfer protocol but at

the expense of increased protocol complexity. Another aspect is that a secure-two party protocol does not prevent a party from being dishonest during the execution. Thus, these protocols provide privacy to the participants' inputs but cannot guarantee that a malicious user could design a circuit that produces an incorrect result without decision, or that he uses a fake input value.

3) Mobilefairplay Framework

The MobileFairPlay framework by [8] presents a mobile implementation of the FairPlay framework for secure two-party computation. More specifically, the protocol first uses a Bluetooth scan operation to find people in the user's neighborhood and then connect to another user to discover whether the two users have similar interest profiles, without disclosing sensitive information. The message is then shared between the user's devices only if their profiles are similar (interest-cast match). To achieve this, MobileFairPlay has been used to implement interest-casting, a novel forwarding strategy used in OppNets [25].

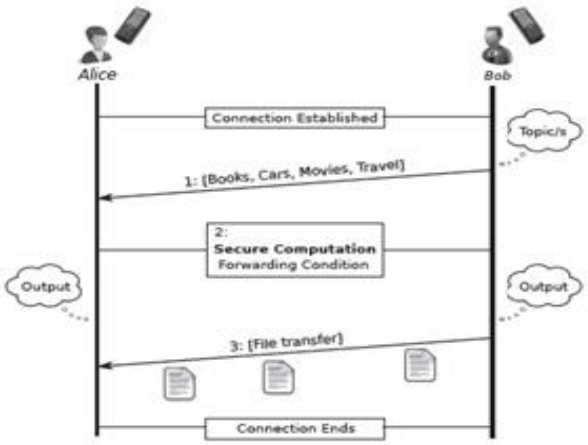

Fig. 2: Privacy-preserving hand-shaking

Each user in an opportunistic network is characterized by an interest profile, which is used to drive the information dissemination process within the network. Interests are expressed as integers in the range [1, max], with 1 representing no interest and max (an arbitrary integer > 1) representing maximum interest. In the interest-cast implementation presented in [12], the authors use a similarity metric called the vector-component-wise (vcw) metric to measure the similarity between the interest profiles of the message source and the nodes to which it should be delivered. Figure 2 above represents the main phases of the application.

MobileFairPlay inherits the threat model from the FairPlay framework, since the ``secure'' part of the framework uses SFDL (Secure Function Definition Language). It was established in [24] that FairPlay (and, hence, MobileFairPlay) has strong security properties in the context of two-party computation and is secure against a malicious attacker. A malicious attacker cannot learn more information about the other party's input than it can learn from a TTP that computes the function; neither can it change the output of the computed function. But it is to be noted that there is no way to prevent Alice from terminating the protocol prematurely, and not sending the outcome of the computation to Bob. This situation can be detected by Bob, but cannot be reversed.

4) Mlce Based Model

An original design was proposed by Abdullatif et.al [3] to solve the problem of privacy enforcement in content-based opportunistic networking. As advertisements and published contents are forwarded through opportunistic intermediate nodes that are not necessarily trusted, the contents have to be encrypted to enforce privacy. The intermediate nodes build forwarding tables based on these encrypted information. Thus, forwarding decisions are taken based on the content of the packet. But content publishers or receivers may not wish to reveal this content to some intermediate nodes whose only task is forwarding. In order to ensure networking together with security, intermediate nodes require two main secure forwarding primitives: Secure setup of forwarding tables (by constructing the forwarding tables based on encrypted receivers' advertisements) and Secure look-up (ability to take correct forwarding decisions based on its forwarding table).

The basic idea behind the proposed solution is to use a Multiple Layer Commutative Encryption (MLCE) that allows intermediate nodes to perform secure transformations without having access to the processed data. This feature of MLCE lends itself very well to solving the problem of routing encrypted data.

In multiple layer encryptions, data is encrypted several times with different keys. The idea is for a receiver to encrypt its receiver advertisement with r>=2 layers corresponding to the r next hops using r different keys, and for the publishers to do the same with their published content. An intermediate node Bk en-route can remove only one encryption layer so that the data is always protected by at least r-1 layers of encryption. Thus Bk performs the setup of forwarding tables and takes forwarding decisions on data encrypted r-1 times without access to data in clear text. Then Bk adds a new encryption layer corresponding to the rth next hop without destroying the other layers and transmits the message.

MLCE allows secure look-up as well as setup of efficient and secure forwarding tables based on encrypted data. Furthermore, the source of a packet does not need to pre-establish an end-to-end secure communication with the destinations or to know ultimately who the destinations are. The nodes only need to share keys for the addition or removal of r layers of encryption, hence requiring only a local view of the network corresponding to the r-hops neighborhood. Thus no end-to-end key management is needed which fits the absence of end-to-end connectivity in oppnets.

5) Proximity Friend Notification Scheme (Pfs)

Though smartphones have revolutionized mobile and pervasive computing around the world, their applications have inevitably introduced higher risks of security and privacy, if not properly taken care of. Chris Carver et.al [6] utilized opportunistic networking to propose an efficient privacy-preserving proximity friend notification scheme (PFS) to simultaneously find the proximity friends and protect smartphone users' identity privacy.

For the discovery of friends within proximity, let us consider a scenario where a smartphone user A may expect to find a friend in a mall to hang around with. Without smartphone, the traditional way to blindly call all her friends to check their availability and willingness to accompany her

is not efficient. However by combining the Bluetooth and 3G techniques of a smartphone, A can opportunistically find nearby friends with low costs. As shown in Figure 3, A can first forward her friend notification packet to the passingby smartphones using Bluetooth. Then, once these passingby smartphone users come into contact with A's friends, the friends can be notified, and they can phone back A immediately if they want to use the 3G network and to disclose their location. Although this kind of smartphone-based proximity friend discovery is efficient and cost-effective, if A's privacy information is disclosed during the opportunistic networking, she may be reluctant to accept it. Therefore, how to efficiently find friends within a proximity while maintaining the smartphone user's privacy is the key to its success but challenging.

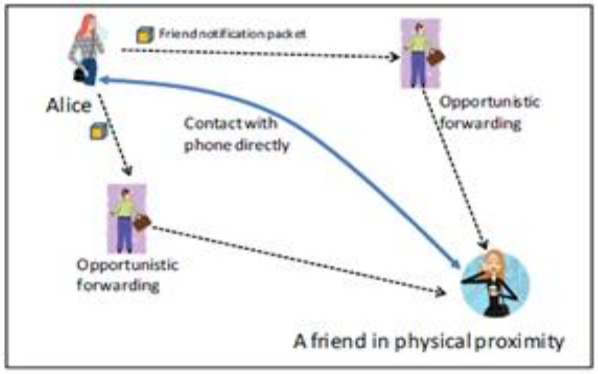

Fig. 3: Proximity friend notification with smartphone

Closely related to this PFN scheme, there are several recent research works on smartphone-enabled applications like [34], [23], [11] etc. [34] proposes the concept of near-me area network (NAN), which focuses on communication among mobile device in close proximity, creating a new kind of promising applications. However, NAN requires a central server and pays less attention to the privacy currently. [23] develop a secure handshake protocol for a patient to find other patients who have the same symptoms in physical proximity. Similarly, [11] also present a secure friend discovery scheme in mobile social network, which enables smartphone users to identify potential friends by computing social proximity in a privacy-preserving manner. However, both [23] and [11] do not utilize the opportunistic networking to extend the finding capability.

6) Fuzzy IBE Based NAC

The Fuzzy Identity-Based Encryption (Fuzzy-IBE) specifies access privileges in terms of attributes directly, providing group access control and is suitable for mobile devices as it does not require intensive calculation. It provides a whitelist mechanism to indicate which users in a group can access data. But as the overhead in terms of data size increases linearly with respect to the whitelist size, this mechanism is not sufficiently scalable. Tzu-Hsin Ho et.al [17] proposed a Fuzzy-IBE based Negative Access Control scheme (NAC) that allows users to selectively exclude specific members from accessing data dynamically. The NAC scheme is supposed to provide the capability of blacklist for secure group communications, enabling senders to dynamically and flexibly exclude specific group members from decrypting data.

The security system using the proposed negative access control scheme (NAC) is composed of two entities: Trusted Server (TS) and Users. Initially TS decides a group wise threshold parameter d and a dummy attribute set and generates secrets and public parameters. For each group member, TS generates a private key PrK and delivers it via a secure channel. Senders use the public parameters to encrypt messages, and receivers use their private keys to decrypt messages.

Depending on whether the message is encrypted by a group ID attribute or a blacklist, the receiver performs the decryption. If the receiver's user ID is in the received attribute set ω', it indicates the receiver is in the blacklist and the information is not sufficient for the receiver to decrypt the message. Otherwise it is implied that the receiver is not in the blacklist and the receiver may decrypt the message. As a revocation is needed, TS generates rekey information RK and delivers the information to non-revoked members to update their private keys by using NAC to protect rekey information from unauthorized users and revoked members.

It has been observed that the proposed security scheme is comparable (similar or better) with some existing schemes, including Os07 [16] and Ro09 [36] in terms of encryption and decryption complexity and communication overhead [17].

A similar technique based on group access was proposed by Maggi Goyal and Manoj Chaudhary [15]. Here the whole network is divided into a number of clusters. Each cluster has a fixed node that maintains a table of all the node ids and a corresponding virtual id. The nodes in the clusters communicate among themselves via these fixed nodes whose responsibility is to generate secure session keys to encrypt the messages and assign virtual ids to nodes. All the intermediate nodes know the virtual ids of the source and destination, not their actual ids, thus ensuring privacy. Though this technique is helping in keeping the user identities private, it increases the work load of the sender node.

7) Pidgin Scheme

In the context of opportunistic networks, content sharing has attracted significant attention. To support content sharing, oppnets often implement a publish-subscribe system in which users may publish their own content and indicate interest in other contents through subscriptions. Using a smartphone, any user can act as a broker by opportunistically forwarding both published content and interests within the network. Unfortunately, untrusted brokers can not only compromise the privacy of subscribers by learning their interests but also can gain unauthorized access to the disseminated content. PIDGIN, an interest and content sharing solution was presented by [1] that preserves privacy in oppnets. Nodes can gain access to content only if they satisfy fine-grained policies specified by the publishers. The system provides scalable key management in which loosely-coupled nodes communicate with each other without any prior contact.

The main idea behind PIDGIN is regulation of access to content using CP-ABE and extension of clear text CP-ABE (Cipher text-Policy Attribute-Based Encryption) policies with the Public-key Encryption with Keyword Search (PEKS) scheme to protect attributes, interests, tags, and leaf nodes in the policy tree. For each attribute, a key pair is calculated that consists of both public and private keys. Using PIDGIN, a publisher encrypts content with CP-

ABE policies and protects the policies. While encrypting CP-ABE policies using PEKS, PIDGIN also incorporates tags that are associated with the published content. Once a subscriber receives a piece of encrypted content along with its encrypted policy, they must first recover the original CP-ABE policy that can be used by the CP-ABE decryption function to get the symmetric key needed to decrypt the content.

Thus, PIDGIN addresses the case where brokers collude but must still not gain access to content, policies, or subscriptions. In the case where two subscribers collude to receive content that each of them alone cannot get otherwise, PIDGIN remains safe due to the randomness embedded in each subscriber's (CP-ABE) decryption key.

In this paper PIDGIN is implemented and its performance evaluated by measuring the overhead incurred for cryptographic operations on a smartphone. As evident from the performance evaluation, the real bottleneck is the overhead incurred by pairing operations at the brokers. An efficient pairing implementation would drastically improve the performance of the system.

There are some other cryptography-based opportunistic networking protocols also that have been proposed in the literature. One of them is FindU [21]. [3] and [38] have also proposed some protocols. Unlike some of the protocol discussed above, these have not been implemented on a mobile platform.

B. Cryptography-Free Protocols

As described in the previous section, the use of cryptographic primitives when executing communication protocols poses significant computational challenges to today's mobile devices. While the computational power of mobile devices is expected to increase constantly in years to come it is still true that cryptography-free communication protocols are much more efficient and apt for mobile devices than cryptography-based protocols. In this section, two approaches are described that have been recently proposed to provide some form of privacy preservation without using cryptographic tools.

1) Probabilistic Profile-Based Routing

Probabilistic Profile-Based Routing (PPBR) has been proposed in [2] as a protocol for privacy-preserving geo-routing in opportunistic networks. The goal of a geo-routing protocol is to deliver a message to all users located in a specific place, such as a building, a shopping center etc. while disclosing minimal information about users and their mobility patterns. PPBR considers user mobility profiles to identify individuals who are good candidates to deliver a message towards the targeted location. To build the mobility profile, the area of interest is divided into square cells, and a user's GPS coordinates, i.e. longitude and latitude, are periodically saved.

PPBR is based on the principle that when a device A carrying a message encounters another device B, it delivers the message to B without a preliminary ``friend discovery'' phase typical of many opportunistic networking protocols, thus improving privacy preservation. It is device B that performs a local computation based on its own mobility profile and the message information, to estimate its ``suitability'' of acting as message carrier. In case the estimated ``suitability'' is below a pre-defined threshold, the message is removed from the buffer and dropped. Otherwise, it remains stored in device B's buffer. To improve privacy preservation, a node becoming carrier is allowed to forward k copies of the message to candidate carriers only when it has moved at distance at least d away from the current position, where d is a protocol parameter. This heuristic prevents an eavesdropper from observing a successful interaction between a message carrier and a new forwarder.

In the simulations, PPBR has been compared with two probabilistic protocols namely probabilistic random walk and probabilistic flooding. In the former protocol, a user accepts a carrier's message with a probability of 1/k, i.e., 10%, independent of her mobility profile. In the latter protocol, whenever a carrier meets a new node, the message is duplicated with probability 0.1. Regarding the delivery rate and latency, PPBR outperforms random walk. Results also show that probabilistic flooding has the highest network load (the average number of message duplicates in the system across all simulations), random walking probabilistic the lowest network load, and PPBR has an intermediate load.

PPBR has been designed to counter both passive and active adversaries. However, adversaries who are able to physically follow a carrier and track her are currently not considered in PPBR, since this attack is considered too costly in an opportunistic network scenario. PPBR also guarantees the de-anonymization of the packet sender, i.e. messages do not have trace of the sender node. However, this situation does not cover the case in which an attacker is able to observe the first step of the message delivery and can easily trace the originator.

2) Hide-And-Lie

L. D´ora and T. Holczer [12] proposed a solution for increasing the users' privacy in an interest-casting application. Each user has an interest profile IP that defines the degree of interest for different topics with a probability e, with $0 < e \leq 1$. A message belonging to a category (topic) k is called primary for a node if IP[k] = 1, i.e., if a node has an interest in the message topic; otherwise, the message is called secondary. The attacker is assumed to estimate user profile UP(u) of a node u as follows:

$$UP_u(t) = (EIP_u(t), CHM_u(t), IDL_u(t)) \quad (1)$$

where the estimated interest profile (EIP) is a binary vector of k positions, with the ith position equal to 1, meaning that u has an interest in the i-th topic. For each category, the category histogram messages (CHM) shows how many messages in the ID list belong to that category. Finally, (IDL) is the ID list of offered messages.

It has been observed that the more UPu(t0) is different from UPu(t1), the less likely it is that the attacker can link both profiles to the same user. Thus, a possible strategy to defend against an attacker is to purposely change the user's profile (obfuscation), using, e.g., randomization techniques. The authors propose the obfuscation strategy called Hide-and-Lie. The term Hide refers to the fact that users can hide interesting categories, showing them as uninteresting. The term Lie refers to the fact that users can falsely claim uninteresting categories as interesting ones. To run the Hide-and-Lie strategy, each node generates its obfuscated EIP from its real interest profile by inverting each category with a given probability λ ($0 < \lambda < 1$). When λ

= 0.5, it brings a totally randomized interest profile. Generally, a good strategy of the Hide-and-Lie mechanism is to apply a high value of λ (close to 0.5). Following this approach, no attacker is able to better distinguish a node, independently from the t value, than a naive attacker does by randomly choosing a node.

Similar obfuscation strategy has been used by [10} and [37} to achieve privacy of link and location respectively. In [10}, Bernhard Distl et al. has proposed to form contact graphs of oppnet nodes based on their past interactions and then randomizing the social links in such a sequence that maintains the routing utility ranking in the graph, thus controlling the amount of link privacy. Similarly, [Zakhary et al., 13] presented a Hybrid and Social-aware Location-Privacy in Opportunistic mobile social networks (HSLPO) that offers location-privacy k-anonymity. HSLPO discovers the users' own social network and use it to obfuscate requests and hide the original sender's location from the Location based service (LBS).

From the above discussions, it can be observed that some cryptography-based solutions have recently proved to perform efficiently on mobile devices, although the experienced running time depends heavily both on the data to be concealed (interest profile, contact list, etc.), as well as on specific protocol parameters. On the other hand, cryptography-free solutions, while lightweight and easily executable in mobile environments, provide weaker privacy guarantees as compared to their cryptography-based counterparts.

IV. PROTOCOLS TO ENSURE TRUST

As Oppnets have no predefined infrastructure and every node is expected to forward the data towards the destination, it raises the issue of selfish behavior and trust. So a node X has to decide if it can trust the next node Y to forward its data or if Y can simply drop it, showing selfish behavior. Over the past decade, several trust based protocols and frameworks have been proposed to ensure data forwarding in oppnets using trustworthy next hop nodes. Depending on the basic principle that is used to select the next-hop trustworthy node, we have categorized the protocols into the following groups:

A. Friendship Vector Based Models

In these types of protocols, trust is based on consciously established friend ties by building a robust tree-like graph of paired users. Each time a node is encountered the friends lists are exchanged and saved in a friendship graph. Trust is calculated as a function of hop distance and interconnection resulting in decreasing trust with increasing hop distance and higher trust in users that are well connected in the resulting graph [33}. Following are some friendship vector based approaches that has been proposed.

1) Explicit Social Trust Based Framework

The basic elements of explicit social trust are consciously selected friend ties. Due to the mobility of the devices, users can establish secure and reliable friend ties whenever they meet by secure pairing. Such a social trust establishment algorithm was proposed by [33]. Each time a node is encountered the friends lists are exchanged and saved in a friendship graph GF. The friendship graph is organized in Ld levels, comprising nodes at the same distance d from the local node n0. Edges only exist between nodes in sequenced levels (say between Ld and Ld+1). For every node in the friendship graph GF, a trust value tei is calculated using Algorithm 1.

Algorithm 1 Explicit Social Trust

1: n_i: A node (local node: n_0)
2: $e_{i,j}$: Edge from n_i to n_j
3: FR_i: Set containing all friends of n_i
4: G_F: Friendship Graph of n_0
5: te_i: Explicit social trust value of n_i
6: L_d: Set of nodes with distance d from n_0 in G_F
7: $te_i = 1 \; \forall \; n_i \in L_1$
8: **for all** nodes n_i in proximity **do**
9: acquire FR_i from n_i and update G_F
10: build G_F and get $L_d \; \forall \; d$
11: **for all** $d \geq 1$ **do**
12: **for all** n_j in L_{d+1} **do**
13: $te_j = \sum_{n_k \in L_d : \exists e_{k,j}} \frac{te_k}{max(\sum_{n_l \in L_{d+1} : \exists e_{k,l}} 1,\; c) \cdot d}$
14: **end for**
15: **end for**
16: **end for**

Fig. 4: Algorithm 1: Explicit Social Trust

The algorithm gives all direct friends an initial trust value of 1. A portion of each node's trust propagates to the next level, depending on the number of child nodes. This results in more trust for well-connected nodes (i.e. with many parent nodes). Since the number of nodes increases with each level, a node's trust decreases with the hop distance d, depending on the connections to the previous level. The propagated trust is independent of the network size and structure, depending only on the average degree of a node, thus making this approach highly scalable.

Since explicit trust is based on pairing, its resilience depends on the user's understanding of only selecting trustworthy peers to pair with. But a device may be compromised with malware, an orthogonal problem which was not in the scope of this work. In related works like [4], trust is transitive, independent of the chain length or the number of disjoint paths. A sybil user would thus only need to establish one trusted relation to gain full trust with all the others. Other approaches such as [5} do not allow for transitivity and only paired friends are trusted. Therefore sybils have to establish trust with all victims one by one, increasing the time and complexity resulting in very sparse trust relations. This explicit social trust based model is a tradeoff between [4] and [5] that allows friendship transitivity depending on the hop distance and connectivity in the social graph.

2) Mobile Trace Based Trust Framework

A trust framework was proposed in [29] for data forwarding in oppnets using mobile traces. Here the seed nodes (source or destination) are initiated with maximum trust value, i.e. 1. The trust value of the helper nodes are calculated based on the friendship vectors associated with these nodes. The friendship vectors consist of the known other nodes to this helper node. Say there are n nodes whose friendship vectors contain the helper node Ni. The trust value of these nodes is taken and the trust value of Ni is calculated from it using the following formula (2):

$$T_i(N_i) = \left\{ \sum_{j=0}^{n} T_j(N_j) \right\} \Big/ n \quad (2)$$

When the source node is ready to send the data, it is assumed that all seed nodes have knowledge of the probable destination position in the form of the trace file

information of other seed nodes. So the data is not broadcasted but sent to selective ones who can forward the data towards the destination, thus reducing network traffic. The data forwarding packet consists of some additional fields of the probable destination position, next hop address and an authentication message. The authentication message is encrypted by the source using a shared secret key known only to the source and destination. So the intermediate nodes cannot access the data, but will only forward it to a trusted next hop node, thus ensuring privacy. To ensure that a malicious node is not selected, the next hop forwarder is chosen such that its trust value is more than or equal to the trust threshold value. The initial trust value calculation for the nodes can be summarized in Algorithm 2.

Algorithm : Initial Trust Value Setup
Input : nodes, friendship vector
Output: nodes with trust value
N_i : a node in the network
S_i : seed node in the network
T_i : trust value of node N_i
FR_i : friendship vector of a node N_i
Step 1: If node N_i is a seed node then trust value of N_i= 1
(ie) $T_i(N_i)$= 1
Step2: For a helper node N_i, search for N_i in other node's friendship vector FR_j.
Step3: Count the number of friendship vectors FR_j where N_i appears.
Step4: Take the trust values of the nodes whose friendship vectors are selected in step2.
Step5: Calculate the trust value for a helper node N_i using the following formula,
$T_i(N_i)= \{\Sigma^n_{j=0} T_j(N_j)\}/ n$;

Fig. 5: Algorithm 2 - Initial Trust Value Setup

The authentication message AM contains the node ids of the helper nodes which help in forwarding the data towards destination. On receiving the packet, the destination node decrypts both the message and the AM using the secret shared key and verifies the correctness of the received data. If the message is correctly received, then the destination increases the trust value of all the helper nodes appearing in the AM. Otherwise it will reduce the trust value of the helper nodes. All these trust value updates are reflected in the trust tables of the respective helper nodes and shared with other neighboring nodes in the network.

Simulation results show that the proposed trust based data forwarding performs better than the routing algorithm without trust framework. Selecting the trust threshold in such a way that the delivery probability is high and overhead ratio is low can be tricky. Also it has been assumed that all seed nodes have updated trace file information of other seed nodes but implementing this in a highly dynamic network may be quite difficult. Furthermore the assumption that a seed node is always present near each helper node to encrypt its id into the AM may not be ensured always.

3) Flooding Prevention Framework

Malicious nodes may modify messages, or flood the network with messages in an attempt to drain other nodes' resources (e.g., battery, storage or bandwidth). Iain Parris and Tristan Henderson focused on this type of attack called flooding in their paper [27]. The goal is to mitigate such flooding attacks, while maintaining the utility of the oppnet. Each message is to be signed by the original sender and then only retransmitted by the trusted social contacts (“friends") of its original sender. A friend will retransmit only after checking the message signature to verify that the message's origin is their trusted friend. Such a defense is lightweight, relying only on local knowledge at each node. The proposed scheme relies on leveraging trusted social contacts.

Each node requires a public/private key pair. Each message is signed by its original sender, enabling any node knowing the sender's public key to verify the message origin. Since messages in the network are only relayed by the original sender's friends, each relay node can thus verify that the message sender is truly their trusted friend by checking the signature (Algorithm 3): if the message is not signed by their friend, then it has been spoofed and is discarded. This mitigates the flooding attack.

Algorithm 3 Message check: only accept a message for relaying if the original message sender is a trusted friend.

```
1. if friends_with(message's original sender) and
   has_valid_original_sender_signature(message) then
2.    accept message for relaying
3. else
4.    discard message
```

It is however possible that a node with genuine friendship links to other nodes may flood messages into the network; these messages will be authenticated and relayed by the attacker's friends. But the attacker must create genuine “friendship" relations with the nodes being attacked which is more expensive than spoofing a message. Additionally if it is observed that a particular sender has generated excessive network traffic then this node can be blocked, thus restricting flooding.

The paper presents simulated flooding attack using real-world data-sets capable of disrupting an opportunistic network, both at the node level by taking nodes offline, and at the global network level by lowering delivery ratio. The proposed strategy is lightweight and effective- for one dataset, the median proportion of time spent offline by nodes was reduced from 42.7% to 6.3%. The results also indicate that the flooding attack is more effective in dense than in sparse datasets. But it assumes reasonably the existence of some mechanism for out-of-band key distribution amongst socially-connected nodes. If a node is “friends" with another node, then they may have sufficient opportunity to exchange keys prior to encountering each other in an opportunistic network scenario. An avenue for future work might be to explore whether it is indeed possible to enable epidemic routing while maintaining public key cryptography, for instance by delegating trust to “friends of friends".

B. Familiarity Based Models

In everyday life, there are certain individuals we regularly share the same space or activity with, i.e. the familiars. These familiars can be easily identified by analyzing contact duration and/or contact frequency of the surrounding peers and sharing this information with those. The advantage of this approach is the automatic operation without the need for conscious user interactions (e.g. pairing). Compared to the friendship graph, the mobility dynamics are captured, resulting in more opportunities to establish trusted relations

in the vicinity. Following are some approaches using the concept of familiarity:

1) Implicit Social Trust Based Framework

Implicit social trust relies on the familiarity and the similarity of the nodes. Familiarity denotes the accumulated contact time and similarity describes to which degree two nodes familiars coincide. Implicit social trust leverages mobility properties using complex network tools, since one might not pair with every encountered user (e.g. some friends or familiar strangers).

Algorithm 4 Implicit Social Trust

1: n_i: A node (local node: n_0)
2: $f_{i,j}$: Familiarity value n_i has for n_j
3: F_i: Set containing $f_{i,j}$ of all n_j
4: ti_i: Implicit social trust value of n_i
5: $fs_i = \sum_j f_{i,j}$
6: **for all** nodes n_i in proximity **do**
7: update $f_{0,i}$
8: acquire F_i from n_i
9: **for all** n_j **do**
10: $ti_j = \underbrace{\frac{f_{0,j}}{fs_0}}_{\text{familiarity}} + \underbrace{\sum_k \frac{f_{0,k}}{fs_0} \cdot \frac{f_{k,j}}{fs_k - f_{k,0}}}_{\text{similarity}}$
11: **end for**
12: **end for**

Fig. 6: Algorithm 4: Implicit Social Trust

A social trust calculating algorithm was proposed by [33]. Both values of familiarity and similarity are normalized, so the sum of all familiarities and all similarities is 1 each. Algorithm 4 below assesses trust of a node by the node's familiarity and similarity. It keeps the familiarity values f0,i up-to-date by keeping track of the connection times with the surrounding nodes. The implicit social trust tij in another node j is calculated by adding its familiarity and similarity. This results in a trust value in the range [0,2), whereas values greater than 1 are negligibly rare.

The main goal of implicit social trust is to make sure the node is not a fast switching identity. The process of assigning trust should be resilient to attacks. To become trusted with a target, an attacker would have to increase its familiarity and decrease the familiarity of all the target's familiars. For the former, the attacker needs to be physically near the target and for the latter, to jam all beacons in the targets surrounding which requires a big effort and is easily detectable. Likewise, the similarity can be forged by increasing the familiarity with all nodes in the targets familiar set, also requiring physical presence. Nevertheless, mobility anomaly detection as well as other Sybil counter measures [28], [32] can be used to further increase the effort needed for an attack.

The three distributed algorithms: Simple, k-Clique and Modularity proposed by Hui et al. [18] can all be manipulated in several ways by an attacker in order for him/her to be included in a community by exchanging manipulated familiarity and community sets for example. With this proposed approach, trust assigned to a node only depends on direct observables (i.e. contact time), without relying on information received by that node (i.e. the nodes familiars set). However, this approach cannot guarantee that a certain entity is behind the proclaimed identity and thus is not as secure as explicit social trust [33]. Nevertheless, a certain amount of trust in a familiar can be justified since the identity cannot be a fast living, which is useful against sybil attacks.

2) COTTON Model Framework

Oppnets consist of highly heterogeneous helper nodes with diverse software and hardware capabilities. For a seamless interaction, a common basis for communication must first be established to help a seed oppnet node to discover potential helpers that possess the desired resources and to invite the chosen candidates to join the oppnet. [31] advocated in their paper that by learning from the context of trust and trust management in the Semantic Web using ontologies, it is possible to introduce trust management into the oppnet context. They proposed a solution named the COTTON model in which the software agents are expected to use trust information from the Semantic Web framework to make security decisions.

This COTTON model advocates the use of the semantic service discovery process using ontologies for handling the capability discovery task. In oppnets, this discovery process is applied to the seed oppnet nodes, by introducing two concepts borrowed from the Semantic Web world - (1) Helper Registry – which stores a list of services that have some degree of functional and non-functional characteristics that can be matched with request requirements, and (2) Helper Advertisement – which describes the capabilities and services provided by an entity (foreign node, helper or seed node).

Typically, an oppnet structure [22] includes: (1) a seed oppnet – a self-configured ad hoc network; (2) distributed Control Center nodes (CC nodes) – a subset of the seed nodes looking into the overall operations of oppnet; (3) helpers –able to capture, communicate, and transmit information; (4) lites – helpers with limited capabilities. It is assumed that only agents can discover the needed helper nodes and identify their capabilities through helper or service advertisements.

The COTTON model classifies the helper nodes according to their access, and capability specifications as private unknown helpers (whose access to their capabilities is private); public unknown helpers (whose access policies and capability descriptions are public; trusted known helpers (whose access policies are known to the oppnet) and oppnet reservists (which are highly trusted by the oppnet).

On identifying the type of a helper node, the capability discovery process works as follows: If the discovered node invited to join the oppnet is an unknown one, a matchmaking agent is uploaded to this potential helper node. Feedback from the targeted node with respect to the quality of service (QoS) is captured and transferred to the Control Center, to be retrofitted into the ontology. Once the matching information is loaded into the node, it follows the helper discovery process, but this time is identified as an existing node. If the matching process finds that there is a match between the oppnet requirements and the node's capabilities, then the node is formally invited to join the oppnet as a helper.

C. Reputation Based Models

In Computer Science, the reputation a user maintains is a quantifier for his behavior in the network he participates in. Regarding to collaborative environments, such evaluation can reflect how he acts as a potential source of resource sharing. Figure 7 below represents a typical scenario using user reputation for future communication.

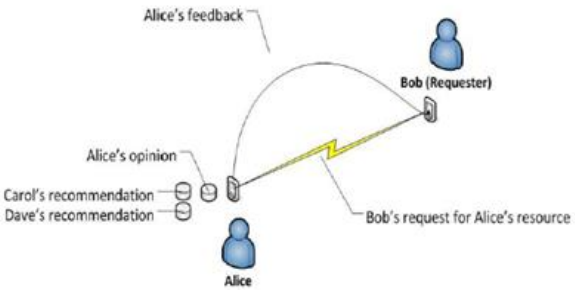

Fig. 7: Typical scenario concerning reputation concepts

As in the real world, someone's reputation is not a fixed value that he maintains by himself, but a combination of everyone else's evaluation about his behavior. In opportunistic networks, a reputation based trust computation is assumed to be very similar to the human notion of trust: Before a node agrees to interact with another, it gathers information about the other node, and determines a trust value depending on previous interactions, on reputation values provided by third parties and other application dependent data. This trust value is applied to access-control problems, to ensure confidentiality and similarly to other security problems. Some models based on this concept are described below:

1) Trust-Based Data Dissemination Technique

Natalia Krystyna Kulesza [20] proposed a Trust based data dissemination technique in which each node has a computational trust value. Computational trust is a security concept that enables nodes in a network to predict the future behavior of another node based on the previous experience with that node, and thus make an informed decision of whether or not to interact with that node again in future. Every time a node helps in forwarding a data packet towards its destination, its trust value increases. If it drops the packet, its trust value decreases. So a node, before choosing another node as its next hop, checks the trust value of that node. If the node's trust value is more than a pre-defined threshold value, then this node is regarded trustworthy enough to be used as the next hop.

An event organiser creates an advertisement with the desired content, includes a trust threshold in it for reference for the trust algorithm and then dispatches the message to the network. Each node stores its own individual trust data consisting of a log of all events attended previously. Each log entry contains an indication of the user's behaviour at that event (either positive or negative). Every time a message is to be passed to a user's device, the potential recipient's trust value is computed, to make sure that he has the right to receive it. The data about this individual, that allow determining his trust, are stored on the recipient's device. The current holder of the message can verify that they have not been altered by using a verification key that is disseminated as part of the message. If the recorded trust data is high enough, the recipient is given the message. After an event, each user device is updated with a log of the current event directly by the organiser. This log consists of the new trust data that reflect how the user behaved at this event.

This thesis contributed towards the conception and creation of an intuitive and realistic mobility trace. It closely follows the temporal mobility model proposed in [19], but surpasses it in the spatial model. This work [20] uses Google Maps to determine paths in a real map, which guarantees that these paths are possible and likely. It also concludes that a trust scheme can be a powerful tool for trust based dissemination; however it should ensure that the fraction of bad disseminators in the network does not exceed a certain limit. It has been experimentally observed that with more than 13.5% of the nodes being bad disseminators, the positive effects of the trust scheme vanish, irrespective of the trust threshold. However, some nodes behaving in a trustful manner for a very long time may turn malicious once it is trusted enough. Also some existing trust frameworks like e-Bay create a new account for the same network when their trust value decreases and can continue their business with a clean reputation. These issues need to be taken care of in future studies.

2) Ontology Based Trust Model

Observing that during the opportunistic encounters the users may need to analyze how reliable the other users are, [14] proposed a trust management system based on ontology. The ontology relies on a reputation system to support the decisions of whether to trust another node or not. A simple prototype is presented to show the system's capabilities in figure 8 below.

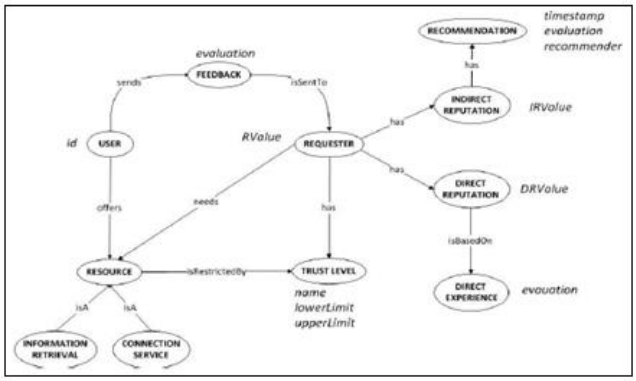

Fig. 8: Ontology proposed for mapping the reputation concepts

To classify the value range used for the reputation system, the Trust level has been divided into five different classes: Very Untrustworthy, Untrustworthy, No Opinion, Trustworthy and Very Trustworthy. Resources assigned with trust level Very Trustworthy indicate that only requesters with highest reputations will be able to access them. Similarly, a Very Untrustworthy resource indicates that anyone can access it. Both direct reputation (direct previous interactions established with the requester) and indirect reputation (other people's opinions about who needs the resource) are considered in behavior evaluation of any node. In this work, the direct reputation of a peer (user) Pj from peer Pi's point of view is given by eqn.(3):

$$DR(P_i, P_j) = \begin{cases} \sum_{k=1}^{h} e_{ij}^{h}, & h \neq 0 \\ 1, & h = 0 \end{cases} \quad (3)$$

Equation (3) represents a mean of the last h experiences that peer Pi had with Pj, with being the quantifier for the kth element. If Pi has no entries about Pj, the result will be one instead of zero. The indirect reputation of a peer Pj from Pi's point of view is given by eqn.(4):

$$IR(P_i,P_j) = \begin{cases} \frac{1}{L}\sum_{k=1}^{L} R(P_i,P_k)R(P_k,P_j)\alpha(t_k), L \neq 0 \\ 1, \quad L = 0 \end{cases} \quad (4)$$

Equation (4) represents a weighted mean of L recommendations, where R(Pk,Pj) (i.e. the reputation that the kth recommender Pk maintains about the requester Pj) is weighted by two terms: R(Pi,Pk) and α(tk). The first is Pk's reputation from Pi's point of view. α(tk) measures the relevance of the kth recommendation in function of how old it is, i.e. more the period of time Pk's opinion about Pj has not been updated the less its impact in the final result. Such factor is calculated by.

$$\alpha = \frac{1}{\left| t - t_k / \mu \right|} \quad (3)$$

Here, t is the current time in Pi's device, tk is the time at which Pk updated the reputation about Pj and α is a constant representing the fade factor for the recommendations received by Pi. After calculating the direct and indirect components, the reputation Pi maintains about Pj is given by eqn. (6):

$$R(P_i,P_j) = \theta DR(P_i,P_j) + (1-\theta) IR(P_i,P_j) \quad (4)$$

Here, the constant θ ϵ [0, 1] indicates the contribution of each component in the final result. After Pi finds out the value for Pj's reputation, the requester has to be mapped on one of the five trust levels mentioned above. Then the requester has to be noticed about the request (acceptance or refusal). Such feedback makes him able to register a new experience about Pi.

As can be concluded, if the request succeeds, an entry with the maximum value is added to Pj's experience history with Pi. Otherwise, request refusal results in values that decrease as R(Pi,Pj) increases. The prototype has demonstrated that the ontology is consistent and can provide good results with relation to trust decisions.

D. Hybrid Trust Models

Apart from the above discussed models, some other approaches have also been proposed. These models either follow a combination of the above discussed methods or some new concepts. We have categorized them as the Hybrid Trust models. Some of them are discussed below.

1) Proximity Based Trust Model

Social studies have shown that people tend to have similarities with others in close proximity. Social interactions are built around trust. People tend to communicate, socialize and potentially trust each other in such clustered communities of interest. [13] studied this trend and proposed a novel proximity based trust model taking into consideration different social aspects like cooperation, honesty, similarity and activity.

Cooperation is measured by making every node, along the path to the destination, report to the source about the previous forwarder. For example: If there is a path between source node A and destination node D like: A→F→D→E

D reports to A that F forwarded the message. Similarly, E reports to A that D forwarded the message.

Honesty is measured by taking into account the distance between the average secondary ratings and the rating provided by a certain node.

$$Honesty_{ij} = 1 - \frac{1}{M}\sum_{k=1}^{M} \frac{|Rating_{j,k} - Avg.rating_{i,k}|}{Avg.rating_{i,k}} \quad (7)$$

where M is the number of nodes in the network.

For Similarity, a vector is incorporated for each node that represents the amount of time that each node is available around a fixed point e.g. home, work, shopping mall etc.

In this literature, Activity was taken into consideration only for routing as a characteristic of reachability but not when establishing trust. The trust is computed as:

$Trust_{ij} = w_1 * f(Cooperation) + w_2 * f(Honesty) + w_3 * f(Similarity) + w_4 * f(Activity)$ (8)

where Trustij is the trust inferred by node i about node j and w1+w2+w3+w4=1. The activity is calculated as the probability of node j meeting the destination. If (Trustij) ≥ Trust_threshold, then the data is forwarded.

Experimental results showed that trust including activity results in a higher delivery rate but at the expense of an increase in the delay. Thus incorporating activity into trust computation increases reachability but can introduce weaknesses. Further improvements are needed in order to decrease the delay, while keeping an acceptable level of trust. Moreover, while using the cooperation parameter, each node is supposed to report to the source node about the previous forwarder. This process involves additional feedback message-passing in addition to actual data forwarding, thus increasing the network traffic and overhead.

2) Trust Based Security Protocol (TSP)

The Trust-based Security Protocol (TSP) was proposed by [16} to secure oppnets against blackhole attacks. When a message is routed by a node to its next hop, a malicious node may advertise itself as an honest node, and provide forged information to attract and intercept the packets, thus preventing the packets from reaching their destinations. This is known as blackhole attack. Sahil Gupta et al. studied this attack and proposed TSP in which the trust value is not only based on the number of successfully transferred messages, but also on three fundamental pillars: SGV, Creadits and Hop count. Nodes in the network are divided into a number of groups. Each group is assigned a priority number called the social group value (SGV). Trust is considered as a function of social group value because it describes and incorporates the importance of the participation of a node in the message passing procedure. For each hop in the message vector, the destination calculates the trust value. Trust is distributed among other peers who have participated in the delivery process. The destination node of the message uses a backward path to achieve this distribution. Through this security feature, a malicious node can be quickly identified since its trust value will never increase as this node does not participate in the routing operation.

In general, if there are m intermediate nodes Ni (i =1... m), that are involved between the source and destination, the trust value of Ni for source is obtained as:

$$Trust(N_i) = \frac{(R1) \times \gamma \times Credits}{m - i + 1} \quad (9)$$

where R1 is the social group value, is a degradation factor and Credits is the aggregate cost of transferring the message from on node to another. The total trust of any Node Nj who has participated in the routing of the messages for all source destination pairs is obtained as:

$$\sum_{j \in Allsources\ contribution} Trust\ (N_j) = \sum_{Allsources\ contribution} \frac{(R_j) \times \gamma \times Creadits}{m - i + 1} \quad (10)$$

where Rj is the social group value. The trust value of a node is 0 if it does not help in forwarding the message to another node. The simulation results compared TSP with the popular PRoPHET routing protocol for DTNs and concluded that TSP significantly improves the message drop ratio compared the PRoPHET protocol with no security under black hole attack (so-called PBH scheme). TSP improves the message drop ratio by about 49.42%. The overhead ratio ({number of relayed packets - number of delivered packets} / number of delivered packets) generated by TSP is significantly lower compared to that generated by PBH. Also, the malicious count, i.e. the number of messages captured by the malicious nodes, is lower when using TSP. Using TSP; the nodes' delivery probabilities are also comparatively lower.

These results show that the PBH scheme leads to higher wastage of network resources compared to TSP. However, TSP involves dividing the network into a number of groups in order to calculate the SGV. Dividing such a highly dynamic network like an oppnet where nodes are entering and leaving at will, may pose to be an additional overhead.

Recently, Chen Xi et al. [36] has proposed a three-layered security model for trust management based on behaviour feedback. This secure scheme includes efficient social context-based key management algorithm, trust model and secure forwarding scheme which can directly be applied in the social context routing protocols. This novel trust model establishes the trust relationships between mobile nodes not only by the existing certificate paths, but also by the feedback information propagated by others.

V. Conclusion

The very nature of opportunistic networks of not having a secure and predefined path from source to destination requires the introduction of an efficient security mechanism for reliable message delivery. Many such protocols have been proposed. Through this survey paper we have attempted to summarize the recent works, categorizing them based upon the basic working concept.

The cryptography based approaches provide strong privacy preservation in opportunistic networks. However, the use of cryptography often requires execution of very complex and computationally intensive operations to obtain the desired level of protection. Mobile devices are built to be portable and energy efficient, hence they are equipped with much less powerful hardware than those equipping traditional PCs. For this reason, most cryptographic protocols seem unsuitable to be run on opportunistic networks containing smartphones, tablet PCs, and similar portable devices. In addition, most of cryptography-based schemes assume that there are certification authorities, which generate keys and certificates for mobile nodes. However, in oppnets, as the mobile nodes interact with each other in a totally distributed and self-organized way, there exists no trusted third party to provide the security services. On the other hand, non-cryptography based schemes often utilize the reputation system to protect the data forwarding in opportunistic networks. But most of them still assume that pre-existing key management already have been built so that each nodes can get their private/ public key pairs easily, which is impractical in opportunistic networks.

The Trust based protocols, on the other hand, are based on the calculation of a trust value or computational trust which signifies the trustworthiness of a node in the network. The efficiency of these protocols, thus, depends on what basic framework is followed to calculate the trust value. But the determination of the initial trust value of each node and the trust threshold value involves some assumptions, thus keeping some vagueness. Also in the absence of a central authority in an oppnet, the generation of many identities by a single user is possible, which raises the need of solid trust metrics. Trust is mainly assessed for that part of the network which is relevant to a node, which makes this approach scalable.

It is our intention that this survey enables readers to have a better insight into the current state of art of the security mechanisms in oppnets and also provide future scopes in identifying a trust based or privacy based secure path for delivering messages which is practically implementable.

References

[1] Muhammad Rizwan Asghar, A.Gehani, B.Crispo,G.Russello, PIDGIN: Privacy-Preserving Interest and Content Sharing in Opportunistic Networks, Proceedings of the 9th ACM symposium on Information, computer and communications security (ASIA CCS 2014), ISBN: 978-1-4503-2800-5, pp.135-146.

[2] A. J. Aviv, M. Sherr, M. Blaze, and J. M. Smith: Privacy- aware message exchanges for geographically routed human movement networks. In Sara Foresti, Moti Yung, and Fabio Martinelli, editors, Computer Security—ESORICS 2012, volume 7459 of Lecture Notes in Computer Science, pages 181–198. Springer Berlin Heidelberg, 2012.

[3] E. Baglioni, L. Becchetti, L. Bergamini, U.M. Colesanti, L. Filipponi, A. Vitaletti, and G. Persiano: A lightweight privacy preserving SMS-based recommendation system for mobile users. In ACM RecSys, 2010.

[4] S. Capkun, L. Buttyan, and J.-P. Hubaux: "Self-organized public-key management for mobile ad hoc network," IEEE TMC, vol. 2, 2003.

[5] S. Capkun, J.-P. Hubaux, and L. Buttyan: "Mobility helps peer-to-peer security," IEEE TMC, vol. 5, 2006.

[6] Chris Carver, Xiaodong Lin: A Privacy-preserving Proximity Friend Notification Scheme with Opportunistic Networking, IEEE ICC 2012 - Wireless Networks Symposium, ISSN:1550-3607, E-ISBN:978-1-4577-2051-2, Page(s): 5387 – 5392.

[7] P. Chapman ,Y. Huang and D. Evans: Privacy-preserving applications on smartphones, In Proceedings of the 6th USENIX Conference on Hot

Topics in Security, HotSec'11, Berkeley, CA, USA, 2011,USENIX Association.
[8] G. Costantino, F. Martinelli, P. Santi, and D. Amoruso: An implementation of secure two-party computation for smartphones with application to privacy preserving interest-cast. In Proceedings of the 18th Annual International Conference on Mobile Computing and Networking, Mobicom '12, pages 447–450, New York, NY, USA, 2012, ACM.
[9] E. De Cristofaro, M. Manulis, and B. Poettering: Private discovery of common social contacts. In Proceedings of the 9th International Conference on Applied Cryptography and Network Security, ACNS'11, pages 147–165, Berlin, Heidelberg, 2011. Springer -Verlag.
[10] Bernhard Distl, Theus Hossmann: Privacy in opportunistic network contact graphs, 15th International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2014, IEEE, INSPEC Accession Number: 14651597, Page(s): 1-3.
[11] W. Dong, V. Dave, L. Qiu, and Y. Zhang: "Secure friend discovery in mobile social networks," in Proc. The 30th IEEE International Conference on Computer Communications (INFOCOM), Shanghai, China, 2011, pp. 1647–1655.
[12] L. D´ora and T. Holczer: Hide-and-lie: enhancing application-level privacy in opportunistic networks. In Proceedings of the Second International Workshop on Mobile Opportunistic Networking, MobiOpp '10, pages 135–142, New York, NY, USA, 2010, ACM.
[13] Mai H. EL-Sherief, Marianne A. Azer: A Novel Proximity Based Trust Model for Opportunistic Networks, International Conference on Availability, Reliability and Security, 2013, Page(s): 281 – 284, INSPEC Accession Number: 13894387, IEEE.
[14] Müller Roberto Pereira Gonçalves, Luciana Andréia Fondazzi Martimiano, Trust Management in Opportunistic Networks, Ninth International Conference on Networks 2010, IEEE, Print ISBN: 978-1-4244-6083-0, Page(s): 209 – 214.
[15] Er. Maggi Goyal, Er. Manoj Chaudhary: Ensuring Privacy in opportunistic Network IOSR Journal of Computer Engineering (IOSR-JCE) e-ISSN: 2278-0661, p- ISSN: 2278-8727 Volume 13, Issue 2 (Jul. - Aug. 2013), Pages 74-82.
[16] Sahil Gupta, Isaac Woungang, Sanjay Kumar Dhurandher, Arun Kumar: Trust-Based Security Protocol Against Blackhole Attacks in Opportunistic Networks, 9th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob), IEEE, 2013, Page(s): 724 – 729, ISSN: 2160-4886.
[17] Tzu-Hsin Ho, Chih-Wei Yi and Chien-Chao Tseng, Group Access Control with Blacklist for Data Dissemination in Mobile Opportunistic Networks, Wireless Communications and Networking Conference (WCNC), 2013 IEEE, ISSN :1525-3511, E-ISBN :978-1-4673-5937-5,Pages: 4410 – 4415.
[18] P. Hui, E. Yoneki, S. Y. Chan, and J. Crowcroft: "Distributed community detection in delay tolerant networks," in MobiArch, 2007.
[19] Vinay Sridhara Jonghyun Kim and Stephan Bohacek: Realistic mobility simulation of urban mesh networks, Ad Hoc Network, pages 411-430, 2009.
[20] Natalia Krystyna Kulesza, User Behaviour and Security in Opportunistic Networks, BSc Thesis, Imperial College of London Department of Computing, 2009.
[21] M. Li, N. Cao, S. Yu, and W. Lou. Findu: Privacy-preserving personal profile matching in mobile social networks. In IEEE Infocom, 2011.
[22] L. Lilien, Z.H. Kamal, V. Bhuse, and A. Gupta: "Opportunistic networks:the concept and research challenges in privacy andsecurity," Intl. Workshop on Research Challenges in Security and Privacy for Mobile and Wireless Networks (WSPWN 2006), Miami,Florida, March, pp. 134-147, 2006.
[23] R. Lu, X. Lin, X. Liang, and X. Shen, "Secure handshake with symptoms-matching: The essential to the success of mhealthcare social network," in Proc. 5th International ICST Conference on Body Area Networks (BodyNets), Corfu Island, Greece, September 2010.
[24] D. Malkhi, N. Nisan, B. Pinkas, and Y. Sella. Fairplay: A secure two-party computation system, In Proceedings of the 13th conference on USENIX Security Symposium—Volume 13, SSYM'04, Berkeley, CA, USA, 2004.USENIX Association.
[25] A. Mei, G. Morabito, P. Santi, and J. Stefa: "Social-aware stateless forwarding in pocket switched networks," in IEEE Infocom, 2011.
[26] E. Okamoto and K. Tanaka: Key distribution system based on identification information. IEEE J.Sel. A. Commun., 7(4):481–485, May 1989.
[27] Iain Parris and Tristan Henderson: "Friend or Flood? Social prevention of flooding attacks in mobile opportunistic networks", IEEE 34th International Conference on Distributed Computing Systems Workshops 2014, ISSN: 1545-0678, Page(s): 16 – 21.
[28] C. Piro, C. Shields, and B. N. Levine: "Detecting the sybil attack in mobile ad hoc networks," in Securecomm and Workshops, 2006.
[29] B.Poonguzharselvi and V.Vetriselvi: Trust framework for data forwarding in opportunistic networks using mobile Traces, International Journal of Wireless \& Mobile Networks, 2012.
[30] A. Shikfa, M. Onen, R. Molva: Privacy in Content-Based Opportunistic Networks, In International Conference on Advanced Information Networking and Applications Workshops 2009, IEEE.
[31] Elvira Bonilla Tamez, Isaac Woungang, Leszek Lilien, Mieso K. Denko, Trust Management in Opportunistic Networks: A Semantic Web Approach, Aug 2009, E-ISBN: 978-0-7695-3805-1, Print ISBN: 978-1-4244-5344-3, Page(s): 235 – 238.
[32] A. Tangpong, G. Kesidis, H. yuan Hsu, and A. Hurson: "Robust Sybil Detection for MANETs," in ICCCN, 2009.
[33] Sacha Trifunovic, Franck Legendre, Carlos Anastasiades: Social Trust in Opportunistic Networks, INFOCOM IEEE Conference on Computer Communications Workshops, 2010, E-SBN: 978-1-4244-6739-6, Print ISBN: 978-1-4244-6739-6, IEEE.

[34] A. K. Y. Wong: "The near-me area network," IEEE Internet Computing, vol. 14, no. 2, pp. 74–77, 2010.
[35] Jie Wu and Yunsheng Wang: Opportunistic Mobile Social Networks, edited by Jie Wu and Yunsheng Wang, CRC Press 2015, Chapter 11.
[36] Chen Xi, Sun Liang, Ma JianFeng, MA Zhuo: A Trust Management Scheme Based on Behavior Feedback for Opportunistic Networks, China Communications, April 2015, IEEE, ISSN: 1673-5447, Page(s): 117 – 129.
[37] Sameh Zakhary, Milena Radenkovic, Abderrahim Benslimane: The Quest for Location-Privacy in Opportunistic Mobile Social Networks, The 9th International Conference on Wireless Communications and Mobile Computing (IWCMC), 2013, IEEE, Print ISBN: 978-1-4673-2479-3, Page(s): 667-673.
[38] R. Zhang, Y. Zhang, J. Sun, and G. Yan: Fine-grained private matching for proximity-based mobile social networking. In IEEE Infocom, 2012.

