

OPPORTUNISTIC NETWORKS: THE CONCEPT AND RESEARCH CHALLENGES IN PRIVACY AND SECURITY

Leszek Lilien,^{1,2} Zille Huma Kamal,¹ Vijay Bhuse,¹ and Ajay Gupta¹

1. WiSe (Wireless Sensornet) Lab
Department of Computer Science, Western Michigan University
Kalamazoo, MI 49008, USA
2. Affiliated with the Center for Education and Research in Information Assurance and Security (CERIAS), Purdue University
West Lafayette, IN 47907, USA

1 Introduction

Critical privacy and security challenges confront all researchers and developers working on ever more pervasive computing systems. We belong to this group. We proposed a new paradigm and a new technology of *opportunistic networks* or *oppnets* to enable integration of the diverse communication, computation, sensing, storage and other devices and resources that surround us more and more. We not only find ourselves in their midst but depend on them increasingly as necessities rather than luxuries. As communications and computing systems are becoming more and more pervasive, the related privacy and security challenges become tougher and tougher.

With oppnets, we charted a new direction within the area of computer networks. One of us invented opportunistic *sensor* networks [3]. The idea was

later generalized by two of us to general opportunistic networks¹ [31]. To the best of our knowledge we are now the first to scrutinize privacy and security challenges inherent in oppnets.

1.1 Goal for opportunistic networks

The goal for oppnets is to leverage the wealth of pervasive resources and capabilities that are within their reach. This is often a treasure that remains useless due to “linguistic” barriers. Different devices and systems are either unable speak to each other, or do not even try to communicate. They remain on different wavelengths—sometimes literally, always at least metaphorically.

This occurs despite devices and systems gaining ground in autonomous behavior, self-organization abilities, adaptability to changing environments, or even self-healing when faced with component failures or malicious attacks. It might look somewhat ironic to a person unaware of interoperability challenges that such ever more powerful and intelligent entities are not making equally great strides in talking to each other.

The oppnet goals can be realized by alleviating first of all the communication problems—including bottlenecks and gaps—that are often the root causes of resource shortages (similarly as transportation inadequacies—not the lack of food in the world—are the root causes of famines).

1.2 Seed oppnets, helpers, and expanded oppnets

Oppnets and their salient features can be described succinctly as follows. Typically, the nodes of a single network are all deployed together, with the size of the network and locations of its nodes pre-designed (either in a fully “deterministic” fashion, or with a certain degree of randomness, as is the case with ad hoc or mobile networks). In contrast, the size of an oppnet and locations of all but the initial set of its nodes—known as the *seed nodes*—can not be even approximately predicted. This is the category of networks where

¹ The name “opportunistic” is used for networks other than our oppnets [41]. In cases known to us, their “opportunism” is quite restricted, e.g., limited to opportunistic communication, realized when devices are within each other’s range. In contrast, our oppnets realize opportunistic growth and opportunistic use of resources acquired by this opportunistic growth.

diverse devices, *not* employed originally as network nodes, are invited to join the seed nodes to become oppnet *helpers*. Helpers perform certain tasks they have been invited (or ordered) to participate in. By integrating helpers into its fold, a *seed oppnet* grows into an *expanded oppnet*.

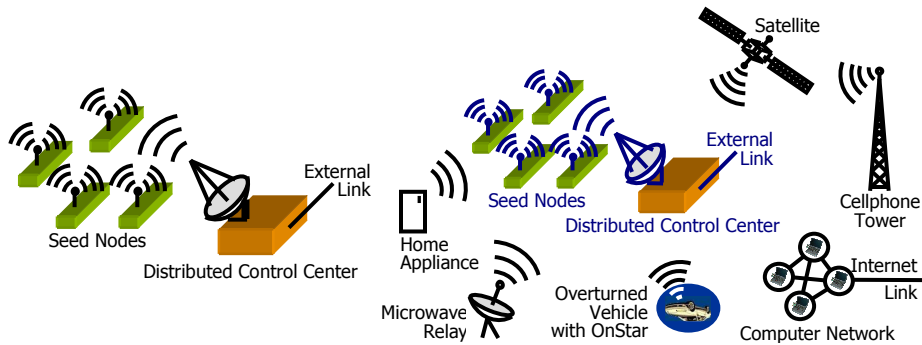


Fig. 1. Seed oppnet.

Fig. 2. Expanded oppnet.

For example, the seed oppnet shown in Fig. 1 grew into the expanded network shown in Fig. 2 by having admitted the following helpers: (a) a computer network from a nearby college campus, (b) a cellphone infrastructure (represented by the cellphone tower), (c) a satellite, (d) a smart appliance (e.g., a smart refrigerator) providing access to a home network, (e) a microwave relay providing access to a microwave network, (f) a vehicular computer network, connected with wearable computer networks on, and possibly within, the bodies of the occupants of a car.

In general, the set of *potential helpers* for oppnets is very broad, incl. communication, computing and sensor systems; wired and wireless; free-standing and embedded. As computing devices continue to become more and more pervasive, the pool of candidates will continue increasing dramatically around us: in infrastructures, buildings, vehicles, appliances, our pockets, etc.

More densely populated areas will have, in general, a denser coverage by potential helpers. As a result, it will be easier to leverage capabilities of an oppnet in more densely populated areas. This is a very desirable natural property, since more resources become available in areas with a possibility of more human victims and more property damage.

With many potential helpers available in an oppnet environment, we need “only” to integrate them in a clever way. We believe that our oppnet paradigm provides a very useful framework—including a conceptual frame of thought—for such integration.

The following scenario illustrates a possible use of an oppnet. A seed oppnet is deployed in a metropolitan area after an earthquake. It finds many potential helpers, and integrates some of them into an expanded oppnet. One of the nodes of the expanded oppnet, a surveillance system, “looks” at a public area scene with many objects. The image is passed to an oppnet node that analyzes it, and recognizes one of the objects as an overturned car (cf. Fig. 2). Another node decides that the license plate of the car should be read. As the oppnet currently includes no image analysis specialist, a helper with such capabilities is found and integrated into the oppnet. It reads the license plate number. The license plate number is used by another newly integrated helper to check in a vehicle database whether the car is equipped with the OnStar™ communication system. If it is, the appropriate OnStar center facility is contacted, becomes a helper, and obtains a connection with the OnStar device in the car. The OnStar device in the car becomes a helper and is asked to contact BANs (body area networks) on and within bodies of car occupants. Each BAN available in the car becomes a helper and reports on the vital signs of its owner. The reports from BANs are analyzed by scheduling nodes that schedule the responder teams to ensure that people in the most serious condition are rescued sooner than the ones that can wait for help longer. (Please note that with the exception of the BAN link that is just a *bit* futuristic—its widespread availability could be measured in years not in decades—all other node and helper capabilities used in the scenario are already quite common.)

1.3 Impacts of oppnets

If the researchers, developers, and manufacturers succeed in building oppnets, the payoff will be swift and substantial. Armies of helpers, mobilized by oppnets, will be capable of contributing towards oppnets’ objectives at a very low or no cost, the latter especially in emergency situations.

The potential of oppnets in all kinds of emergencies—including man-made and natural disasters—is especially noteworthy. In the past few years we have seen great disasters, such as the 9/11 terrorist attack, the tsunami in the

Southeast Asia, and Hurricane Katrina. Casualties and damages are too often compounded by problems faced by the first responders and relief workers. There is a common thread to all these problems: a lack of adequate communication facilities in the disaster areas and beyond. Therefore, providing means of dependable communication in emergencies via oppnets should produce swift and substantial payoffs.

The impact of oppnets on research and development can be significant, especially in the broad and expanding field of pervasive computing. We believe that oppnets are an epitome of pervasive computing. The most critical problems inherent to pervasive computing were very aptly expressed as follows [46]: *Pervasive computing has pervasive problems, not the least of which are interoperability, security and privacy*. Oppnets confront all three enumerated problems head on (though in this chapter we concentrate on the discussion of privacy and security issues). Therefore, work on privacy and security problems in oppnets will be a good test case for attacking the privacy and security problems in pervasive computing.

1.4 Chapter contents

The next section describes the basics of oppnet operation. Section 3 describes example oppnet applications and use scenarios. Section 4 presents related work in privacy and security. Section 5 emphasizes the critical significance of privacy challenges in oppnets. Section 6 presents the privacy and security challenges in oppnets, and sketches proposed research directions for solutions to some of these challenges. Finally, Section 7 concludes the paper.

2 Opportunistic networks: Basics of operation

2.1 Seed oppnets and their growth into expanded oppnets

Each opportunistic network grows from a *seed* that is a set of nodes employed together at the time of the initial oppnet deployment. The seed is pre-designed (and can therefore be viewed as a network in its own right). In the extreme case, it can consist of a single node.

The seed grows into a larger network by extending invitations to join the oppnet to foreign devices, node clusters, networks, or other systems which it is able to contact. Any new node that becomes a full-fledged oppnet member, that is a *helper*, may be allowed to invite external nodes. By inviting “free” collaborative nodes, the opportunistic networks can be very competitive economically. The issues that have to be addressed are proper incentives or enforcements so that invited nodes are willing or required to join, and potentially lower credibility of invited collaborators that, in general, can’t be fully trusted (at least till they prove themselves). Helpers of an oppnet collaborate on realizing the oppnet’s goal. They can be deployed to execute all kinds of tasks even though, in general, they were not designed to become elements of the oppnet that invited them.

2.2 Oppnet helpers and oppnet reserve

Potential oppnets helpers

The set of potential helpers includes even entities not usually thought of as powerful network nodes, both wired and wireless, free-standing and embedded. Even nodes without significant processing, communication, or sensing capabilities, can collectively contribute to processing or communication capabilities of an oppnet in a significant way. After all, any networked PC or embedded processor has *some* useful sensing, processing, or communication capabilities. As examples of minimal useful capabilities, we can consider information about user’s presence or absence, her work habits and Internet access patterns collected by her desktop and her PDA; information about user’s location collected by his cellphone (even one without GPS can be triangulated); and data about food consumed by user’s household collected by a processor embedded in a refrigerator and RFID-equipped food packages and containers.

Before a seed oppnet can grow, it must discover its own set of *potential helpers* available to it. As an example of a discovery, a PC can be discovered by an oppnet once the oppnet identifies a subset of Internet addresses (*IP addresses*) located in its geographical area. Another example of discovery could involve an oppnet node scanning the spectrum for radio signals or beacons, and collecting enough information to be able to contact their senders.

Helper functionalities

It should be noted that, in general, working in the “disaster mode” does not require any new functionalities from the helpers. For example, in case of fire monitoring tasks, the weather sensornet that became a helper can be simply told to stop collecting precipitation data, and use the released resources to increase the sampling rates for temperature and wind direction.

It is possible that more powerful helpers could be reprogrammed on the fly. Also, oppnet nodes might be built with excess general-purpose communication, computation, storage, sensing, and other capabilities useful in case of unforeseen emergencies. For example, excess sensing capabilities could be facilitated by multisensor devices that are becoming cheaper and cheaper as new kinds of sensors are being developed all the time (for example, novel biosensors for detection of anthrax [21]).

Use of helper functionalities can be innovative in at least two ways. First, oppnets are able to exploit *dormant capabilities* of their helpers. For instance, even entities with no obvious sensing capabilities can be used for sensing: (a) a desktop can “sense” its user’s presence at the keyboard; (b) a smart refrigerator monitoring opening of its door can “sense” presence of potential victims at home in a disaster area. As another example, the water infrastructure *sensornet* (sensor network) with multisensor capabilities, which is positioned near roads, can be directed to sense vehicular movement, or the lack thereof.

Second, helpers might be used in novel combinations, as illustrated by the scenario from Section 1.2. In the scenario, a complex interaction of many oppnet nodes and helpers starts when a surveillance system, serving as an oppnet node, receives an image of an overturned car.

Asking or ordering helpers and oppnet reserve

Helpers are either invited or ordered to join [33, 32]. In the former case, contacted potential helpers can either volunteer or refuse the invitation. In the latter case, they must accept being conscripted in the spirit of citizens called to arms (or suffer the consequences of going AWOL).

The issue of ordering candidate helpers may seem controversial, and requires addressing. First, it is obvious that any candidate can be asked to join in any situation. Second, any candidate can be ordered to join in life-or-death situations. It is an analogy to citizens being required by law to assist with their property (e.g., vehicles) and their labor in saving lives or critical resources.

Third, some candidates can always be ordered to become helpers in emergencies. Such helpers include many kinds of computing and communication systems serving police, firemen, National Guard, and military. Also the federal and local governments can make some of their systems available for any oppnet deployed in an emergency.

The category of systems always available on an order coming from an oppnet includes systems that volunteer—actually, “are volunteered” by their owners. In an obvious analogy to the Army, Air Force, and other Reserves, they all can be named collectively as the *oppnet reserve*. Individually they are *oppnet reservists*. As in the case of the human reserves, volunteers sign up for oppnet reserve for some incentives, be they financial, moral, etc. Once they sign up, they are “trained” for an active duty: facilities assisting oppnets in their discovery and contacting them are installed on them. For example, a standard Oppnet Virtual Machine (OVM) software, matched to their capabilities—either heavy-, medium- or lightweight—is installed on them. (OVM is discussed in [32].) The “training” makes candidates highly prepared for their oppnet duties.

By employing helpers working for free (as volunteers or conscripts), opportunistic networks can be extremely competitive economically in their operation. Full realization of this crucial property requires determining the most appropriate incentives for volunteers and enforcements for conscripts.

Preventing unintended consequences of integrating helpers

Examples of unintended consequences when integrating helpers are disruptions of operations of life-support and life-saving systems, traffic lights, utilities, PTSN and cell phones, the Internet, etc. [32].

To protect critical operations of oppnets and of helpers joining an oppnet, oppnets must obey the following principles:

- Oppnets must not disrupt critical operations of potential helpers. In particular, they must not take over any resources of life-support and life-saving systems.
- For potential helpers running non-critical services, risk evaluation must be performed by an oppnet before they are asked or ordered to join the oppnet. This task may be simplified by potential helpers identifying their own risk levels, according to a standard risk level classification.

- Privacy and security of oppnets and helpers must be assured, especially in the oppnet growth process.

2.3 Critical mass for an oppnet and growth limitations

Critical mass

Oppnets can be really effective if they are able to expand their reach enough to reach a certain “critical mass” in terms of size, node locations, and node capabilities. Once this threshold is passed, they are ready to communicate, compute, and sense their physical environment. They can gather data for damage assessment when used in emergencies or disaster recovery. Some sensornets that become helpers—such as sensor nodes embedded in roads, buildings, and bridges—are designed primarily for damage assessment. Other helpers, whether members of sensornets or not, can gather data—legitimately or not—on general public, employees, or other monitored individuals.

Growth limitations

The network stops inviting more nodes when it obtains enough helpers providing sufficient sensing, processing, and communication capabilities (cost/benefit analysis of inviting more nodes might be performed). It should avoid recruiting superfluous nodes that wouldn’t help and might reduce performance by using resources just to “gawk.” This does not mean that network configuration becomes frozen. As the area affected by the monitored activity (e.g., an earthquake) changes and the required monitoring level in different locations shifts (due, say, to the severity of damage), the oppnet reconfigures dynamically, adapting its scope and its capabilities to its needs (e.g., to the current disaster recovery requirements).

3 Example oppnet applications and use scenarios

3.1 Characteristics of oppnet-based applications

Use of oppnets is most beneficial for applications or application classes characterized by the following properties:

- It can start with a seed
- It requires high interoperability
- It uses highly heterogeneous software and hardware components
- It can benefit significantly from leveraging diverse resources of helpers
- It is able to maintain persistent connectivity with helpers once it is established

We are working on a Standard Implementation Framework for oppnets [32] which will facilitate creating oppnet-based applications by providing a standard set of primitives. The primitives for use by application components will, for example, facilitate discovering potential helpers, integrating them, and releasing them when they are not needed any more.

3.2 Example oppnet application classes

We can envision numerous applications and application classes that can be facilitated by oppnets. Some of them are described next.

Emergency applications

We see important applications for opportunistic networks in all kinds of emergency situations, for example in hurricane disaster recovery and homeland security emergencies. We believe that they have the potential to significantly improve efficiency and effectiveness of relief and recovery operations. For predictable disasters (like hurricanes or firestorms, whose path can be predicted with some accuracy), seed oppnets can be put into action and their build-up started (or even completed) *before* the disaster, when it is still much easier to locate and invite other nodes and clusters into the oppnet. The first helpers invited by the seed could be the sensornets deployed for structural damage monitoring and assessment, such as the ones embedded in buildings, roads, and bridges.

Home/office oppnet applications

Oppnets can benefit home/office applications by utilizing resources within the domestic/office environment to facilitate mundane tasks. Consider contrast between the two scenarios for viewing a visual message on a PDA in a living

room. Without an oppnet-based software, PDA has to present the message using the miniscule PDA screen and its substandard speakers. With an oppnet-based software, PDA (now being a single-node seed oppnet) can quickly find helpers: a TV monitor and an audio controller for HiFi speakers available in the living room. PDA can ask these helpers to join, and integrate them into an expanded oppnet. The expanded oppnet, now including 3 nodes (the PDA, the TV monitor, and the audio controller), can present the visual message on high-quality devices.

A similar scenario can be realized in MANETs [38] but with much more programmer's efforts since MANETs do not provide high-level application-oriented primitives to simplify implementation. Only oppnets do [32].

Benevolent and malevolent oppnet applications

As most technologies, opportunistic networks can be used to either benefit or harm humans, their artifacts, and technical infrastructure they rely upon. Invited nodes might be “kept in the dark” about the real goals of their host oppnets. Specifically, “good guys” could be cheated by a malevolent oppnet and believe that they will be used to benefit users. Similarly, “bad guys” might be fooled by a benevolent oppnet into believing that they collaborate on objectives to harm users, while in fact they would be closely controlled and participate in realizing positive goals.

On the negative side, home-based opportunistic networks could be the worst violators of individual's privacy, if they are able to exploit PCs, cellphones, computer-connected security cameras, embedded home appliance processors, etc.

Predator oppnets

To counteract malevolent oppnets threats, *predator* networks that feed on all kinds of malevolent networks—including malevolent oppnets—can be created. Using advanced oppnet capabilities and primitives, they can detect malevolent networks, plant spies (oppnet helpers) in them, and use the spies to discover true goals of suspicious networks. Their analysis must be careful, as some of the suspicious networks might actually be benevolent ones, victims of false positives. Conversely, intelligent adversaries can deploy malevolent predator networks that feed on all kinds of benevolent networks, including benevolent oppnets.

3.3 Example oppnet application scenarios

We now discuss two example oppnet application scenarios: a benevolent one and a malevolent one. Both rely on some reconfiguration capabilities of non-opportunistic (regular) sensornets.

Benevolent oppnet scenario —“Citizens called to arms”

A seed oppnet is deployed in the area where an earthquake occurred. It is an ad hoc wireless network with nodes much more powerful than in a “typical” ad hoc network (more energy, computing and communication resources, etc.). Once activated, the seed tries to detect any nodes that can help in damage assessment and disaster recovery. It uses any available method for detection of other networks, including radio-based detection (including use of Software Defined Radio and cellphone-based methods), searching for nodes using the IP address range for the affected geographic area, and even AI-based visual detection of some appliances and PCs (after visual detection, the seed still needs to find a network contact for a node to be invited).

The oppnet “calls to arms” the optimal subset of detected and contacted “citizens,” inviting all devices, clusters, and entire networks, which are able to help in communicating, computing, sensing, etc. In emergency situations, entities with any sensing capabilities (whether members of sensornets or not), such as cellphones with GPS or desktops equipped with surveillance cameras, can be especially valuable for the oppnet.

Let us suppose that the oppnet is able to contact three independent sensornets in the disaster area, deployed for weather monitoring, water infrastructure control, and public space surveillance. They become helper candidates and are ordered (this is a life-or-death emergency!) to immediately abandon their normal daily functions and start assisting in performing disaster recovery actions. For example, the weather monitoring sensornet can be called upon to sense fires and flooding, the water infrastructure sensornet with multisensor capabilities (and positioned under road surfaces) —to sense vehicular movement and traffic jams, and the public space surveillance sensornet —to automatically search public spaces for images of human victims.

Malevolent oppnet scenario — “Bad guys gang up”

Suppose that foreign information warriors use agents or people unaware of their goals to create an apparently harmless weather monitoring sensornet. Only they know that the original sensornet becomes a seed of a malevolent oppnet when activated. The sensornet starts recruiting helpers.

The seed does reveal its true goals to any of its helpers. Instead, it uses a cover of a beneficial application, proclaiming to pursue weather monitoring for research. Actually, this opportunistic sensornet monitors weather but for malicious reasons: it analyzes wind patterns that can contribute to a faster spread of poisonous chemicals. Once the “critical mass” in terms of geographical spread and sensing capabilities is reached, the collected data can be used to make a decision on starting a chemical attack.

4 Related work in privacy and security

In this section we discuss briefly some privacy and security solutions proposed in: (a) pervasive computing, (b) ambient networks, (c) grid computing. We also discuss privacy and security solutions based on: (a) trust and reputation in open systems, (b) intrusion detection in ad hoc, mobile, or wireless systems, and (c) honeypots and honeyfarms.

Privacy and security solutions in pervasive computing

Pervasive computing environments require security architecture based on trust rather than just user authentication and access control [25]. Campbell *et al.* [7] looked at the development of several middleware solutions that can support different aspects of security, including authentication, access control, anonymity, and policy management. They also looked at the instantiations of these aspects with diverse mechanisms.

Chen *et al.* [10] described a risk assessment model and proposed an estimator of risk probability that can form the core part of a risk assessment in a ubiquitous computing environment. This estimator is based on a general definition inspired by traditional probability density function approximation, and an implementation by a clustering procedure. To take a distribution of points into account, the authors adopted the Mahalanobis distance for calculating similarities of interactions. They proposed to develop the SECURE

framework into which this risk probability estimator is embedded. This risk estimator is feasible and the authors have demonstrated that it fits well within the framework.

Transportation has traditionally been the realm of the machine [13]. Today, as vehicles become increasingly computerized, the authors propose to see this technology moving from under the hood to pervasively connect with passengers, other vehicles and the world. Security and privacy consequences are significant.

Wagealla *et al.* [52] propose a model for trust-based collaboration in ubiquitous computing. The model ensures secure collaboration and interaction between smart devices, by addressing the concerns of security and trust.

Undercoffer *et al.* [47] designed a communications and security infrastructure that goes far in advancing the goal of anywhere-anytime computing. Their work securely enables clients to access and utilize services in heterogeneous networks. It provides a service registration and discovery mechanism implemented through a hierarchy of service management. The system was built upon a simplified PKI that provides for authentication, non-repudiation, anti-playback, and access control. Smartcards were used as secure containers for digital certificates. The system is dependent solely on a base set of access rights for providing distributed trust model. The authors presented the implementation of the system and described the modifications to the design that are required to further enhance distributed trust. They claim that the implementation is applicable to any distributed service infrastructure, whether the infrastructure is wired, mobile, or ad hoc.

Kagal *et al.* [27] used an agent-oriented paradigm to model interactions between computationally enabled entities in pervasive environments. They presented an infrastructure that combined existing authentication features, like SPKI, with notions of policy-driven interaction and distributed trust in order to provide a highly flexible approach for enforcing security policies in pervasive computing environments. They implement the system on a variety of handheld and laptop devices using Bluetooth and 802.11.

Privacy and security solutions in ambient networks

The key problem privacy and security issues in *ambient networks* [43] can be categorized and summarized as follows:

1. Trust establishment and secure agreements

This includes: (a) a foundation for trust modeling, and (b) security for establishment and execution of general agreements between parties in a dynamic and scalable way.

2. Access security

This includes: (a) security services at a network edge, e.g., means for a mobile terminal connecting to an access network assuring that it receives the configuration parameters in a secure way, (b) required security services below the IP layer and interfaces to higher-layer control components, and (c) security aspects of ad hoc and multi-hop networks that extend a fixed public network in two cases: (i) where the extension is controlled by the network operator, and (ii) where individual nodes owned by different parties co-operate to provide a better coverage.

3. Security for mobility and multi-homing

This includes: (a) security for mobility mechanisms, focusing especially on approaches that do not assume shared authentication infrastructure between all parties, (b) security challenges in mobility mechanisms that optimize movement for groups of nodes simultaneously, (c) security aspects of session mobility, i.e., moving an ongoing session from one device to another, and (d) secure traversal and management of middleboxes, such as firewalls and Network Address Translators (NATs).

4. Special topics

This includes: (a) group security, e.g., the creation of dynamic, efficient and scalable key management infrastructures for distribution of keys in large groups, and (b) attack resistance dealing with intrusion detection and other methods for protection against threats to availability.

Privacy and security solutions in grid computing

Humphrey and Thompson [20] and Welch *et al.* [53] discuss security-related research in *grid computing*. The Authorization Accounting Architecture Research Group proposes the following high-level requirements [50, 14]:

1. Authorization decisions must be made on the basis of information about the user, the service requested and the operating environment. Information about a user must include extensible attributes as well as the identity. Unknown users must be supported.

2. Identity and attribute information must be passed with integrity, confidentiality, and non-repudiation.
3. Authorization information must be timely (and revocable).
4. Supporting application proxying for users.
5. Supporting ways of expressing trust models between domains.
6. Protocol must support context-sensitive decisions and transactions.
7. Both centralized and distributed administration of authorization information.
8. Separate or combined messages for authentication and authorization.
9. Authorization information should be usable by applications, including accounting and auditing applications.
10. Support negotiation of security parameters between a requestor and a service.

Johnston *et al.* [23] have also written about the special security considerations for grids based on the experience of the NASA Production IPG grid as well as the experience with several DOE collaborators. They considered the threat model and risk reduction in some detail and came up with a security model based on using available grid security services.

Privacy and security solutions based on trust and reputation in open systems

Burnside *et al.* [6] described a resource discovery and communication system designed for security and privacy. All objects in the system, e.g., appliances, wearable gadgets, software agents, and users have associated trusted software proxies that either run on the appliance hardware or on a trusted computer. They described how security and privacy are enforced using two separate protocols: a protocol for secure device-to-proxy communication, and a protocol for secure proxy-to-proxy communication. Using two separate protocols allows running a computationally inexpensive protocol on *thin* devices, and a sophisticated protocol for resource authentication and communication on more powerful devices. The authors designed a device to proxy protocol for lightweight wireless devices, and the proxy-to-proxy protocol which is based on SPKI/SDSI (Simple Public Key Infrastructure / Simple Distributed Security Infrastructure).

The CONFIDANT protocol [5] detects misbehaving nodes by means of observation or reports about several types of attacks. It allows to route around

misbehaving nodes and to isolate them from the network. Nodes have a *monitor* for observations, *reputation records* for first-hand and trusted second-hand observations, *trust records* to control trust given to received warnings, and a *path manager* for nodes to adapt their behavior according to reputation.

A collaborative reputation mechanism proposed by Michiardi and Molva [35], has a *watchdog* component. It is complemented by a reputation mechanism that differentiates between *subjective reputation* (observations), *indirect reputation* (positive reports by others), and *functional reputation* (task specific behavior). They are all weighted to derive a combined reputation value that is used to make decisions about cooperation with or a gradual isolation of a node.

Bansal and Baker [2] propose a mechanism that relies exclusively on first-hand observations for *ratings*. If a rating is below the pre-defined *faulty threshold*, the node is added to the *faulty list*. The faulty list is appended to the route request by each node broadcasting the request, and is used as an *avoid list*. A route is rated good or bad depending on whether the next hop is on the faulty list. In addition to the ratings, nodes keep track of the *forwarding balance* with their neighbors, by maintaining a count for each node.

Li *et al.* [29] proposed a new model to quantify trust level of nodes in MANETs. The scheme is distributed and effective without reliance on any central authority. In this scheme, both pre-existing knowledge and direct interaction among nodes in the network can be taken into account as a direct experience for their trust evaluation. To quantify the trust value for direct experiences, the authors defined a new computation function, in which the effect of different direct experience instances can be adjusted individually. To combine own trust value and the recommendation trust value from others, they defined a new trust relationship equation. This scheme deals with the fundamental trust establishment problem and can serve as a building block for higher-level security solutions, such as key management schemes or secure routing protocols.

Venkatraman *et al.* presented [49] an end-to-end data authentication scheme that relies on mutual trust between nodes. The basic strategy is to take advantage of the hierarchical architecture that is implemented for routing purposes. They proposed an authentication scheme that uses TCP at the transport layer and a hierarchical architecture at the IP layer. In this way, the number of encryptions needed is minimized, thereby reducing the

computational overheads and resulting in substantial savings, as each node has to maintain keys for fewer nodes.

Privacy and security solutions based on intrusion detection

Mishra *et al.* [36] reviewed many intrusion detection approaches for wireless ad hoc networks.

Nordqvist, Westerdahl and Hansson [39] consider an intrusion detection system for MANETs. Another intrusion detection approach relevant for oppnets comes from the AAFID project [56], in which autonomous agents perform intrusion detection using embedded detectors. An *embedded detector* is an internal software sensor that has added logic for detecting conditions that indicate a specific type of attack or intrusion. Embedded detectors are more resistant to tampering or disabling, because they are a part of the program they monitor. Since they are not executing continuously, they impose a very low CPU overhead. They perform direct monitoring because they have access to the internal data of the programs they monitor. Such data does not have to travel through an external path (a log file, for example) between its generation and its use. This reduces the chances that data will be modified before an intrusion detection component receives them.

Balfanz *et al.* [1] proposed a solution to the problem of secure communication and authentication in ad-hoc wireless networks. The solution provides secure authentication using almost any established public-key-based key exchange protocol, as well as inexpensive hash-based alternatives. In this approach, devices exchange a limited amount of public information over a privileged side channel, which then allows them to complete an authenticated key exchange protocol over the wireless link. This solution does not require a public key infrastructure, is secure against passive attacks on the privileged side channel and all attacks on the wireless link, and directly captures users' intuitions whether they want to talk to a particular, previously unknown device in their physical proximity.

Cross-feature analysis is proposed by Huang, Fan, Lee, and Yu [19] to detect routing anomalies in mobile ad-hoc networks. They explore correlations between features and transform the anomaly detection problem into a set of classification sub-problems. The classifiers are then combined to provide an *anomaly detector*. A sensor facility is required on each node to provide statistics information.

Wireless networks are vulnerable to many identity-based attacks in which a malicious device can use forged MAC addresses to masquerade as a specific client or to create multiple illegitimate identities [12]. A transmitting device can be robustly identified by its *signalprint*, a tuple of signal strength values reported by access points acting as sensors. Apart from MAC addresses or other packet contents, attackers do not have much control regarding the signalprints they produce. By tagging suspicious packets with their corresponding signalprints, the network is able to robustly identify each transmitter independently of packet contents, allowing detection of a large class of identity-based attacks with high probability.

Čapkun *et al.* [9] introduced *integrity regions*, a security primitive that enables integrity protection of messages exchanged between entities that do not hold any mutual authentication material (e.g., public keys or shared secret keys). Integrity regions make use of lightweight ranging techniques and of visual verification within a small physical space. The main application of integrity regions is key establishment. The proposed scheme effectively enables authentication through presence, and therefore protects key establishment from the man-in-the-middle (MITM) attacks. Integrity regions can be efficiently implemented using off-the-shelf components such as ultrasonic ranging hardware.

Privacy and security solutions based on honeypots and honeyfarms

Honeypots and honeyfarms can be considered special types of mechanisms for intrusion detection. A *honeypot* is a decoy whose value lies in being probed, attacked or compromised. It is designed to trap or delay attackers, and gather information about them. Honeypot have resources dedicated to these goals that no other productive value. A honeypot should not see any traffic because it has no legitimate activity. Any interaction with a honeypot is most likely an unauthorized or a malicious activity, and any connection attempts to a honeypot are most likely probes, attacks, or compromises [34]. Honeypot logs can be used to analyze attackers' behaviors and design new defenses.

Honeypots can be categorized with respect to their implementations [22]. A *physical honeypot* is a real machine on the network with its own operating system and address, while a *virtual honeypot* is a Virtual Machine hosted in a physical machine. Virtual honeypots require far less computational and network resources than physical honeypots, and they provide far greater

flexibility in emulating various operating systems.

Single honeypots or multiple but independently operated honeypots suffer from a number of limitations, like a limited local view of network attacks, a lack of coordination among honeypots on different networks, inherent security risks involved in honeypot deployment (requiring non-trivial efforts in monitoring and data analysis), and lack of centralized management features. Having a decentralized honeypot presence while providing uniform management in honeypot deployment and operation is a challenging task [22].

A possible solution overcoming the limitations of individual honeypots comes from *honeypot farming*. Instead of deploying large numbers of honeypots in various locations, all honeypots are deployed in a single, consolidated location [44]. This single network of honeypots becomes a *honeyfarm*. Attackers are then redirected to the honeyfarm, regardless of what network they are on or are probing, using *redirectors*. A redirector acts as a proxy transporting an attacker's probes to a honeypot within the honeyfarm, without the attacker ever knowing it. An attacker thinks she is interacting with a victim on a local network, when in reality her attack is transported to the honeyfarm.

5 The critical significance of privacy challenges in oppnets

The proposed opportunistic network technology is one of possible approaches for moving towards the ultimate goal of pervasive computing. Since huge privacy risks are associated with all pervasive computing approaches, oppnets—being such an approach—must face significant privacy perils.

Pervasiveness must breed privacy threats, as we explain in our 2004 paper [3]:

Pervasive devices with inherent communication capabilities might [...] self-organize into huge, opportunistic sensor networks, able to spy anywhere, anytime, on everybody and everything within their midst. [...] Without proper means of detection and neutralization, no one will be able to tell which and how many snoops are active, what data they collect, and who they work for (an advertiser? a nosy neighbor? Big Brother?). Questions such as “Can I trust my refrigerator?” will not be jokes—the refrigerator will be able to snitch on its owner’s dietary misbehavior to the owner’s doctor.

We very clearly recognize the crucial issue of privacy in oppnets (as well as in all other pervasive computing approaches). Privacy guarantees are indispensable for realization of the promise of pervasive computing. We strongly believe that without proper privacy protection built into any technology attempting to become pervasive, the public will justifiably revolt against it. Any oppnet solution (or other pervasive computing solution) compromising on privacy protection is doomed to a total failure. Simply, *privacy protection is the “make it or break it” issue for oppnets and pervasive computing in general.*

There is no inherent reason why an oppnet would need to enslave the device asked to help it, exploiting its sensitive resources. There is no inherent reason why the helper device would need to disclose all such resources to the oppnet. In the simplest solution, the candidate helper will keep its private data in a secure vault (e.g., enciphered in its storage) before agreeing to join an oppnet that asked for help. In case of an involuntary conscription (in an emergency situation), the oppnet will allow the candidate helper to save private data in helper’s own vault before mustering it.

Other solution we consider will rely on a strict separation of private and public areas within the helper device or network. This will ensure that a benevolent oppnet will never (even when it malfunctions) attempt to capture helper’s private data. It will also provide protection against malevolent oppnets that might attack privacy of other devices or networks pretending they need them as their helpers for legitimate needs.

Still other approaches include protecting privacy of helpers and other entities that are under oppnet management or surveillance by, for example, assuring their anonymity or pseudonymity; providing algorithms for detecting malevolent oppnets, which masquerade as benevolent oppnets in order to attack prospective helpers (detection will deny them opportunity to compromise privacy of helpers); and developing methods to protect oppnets against all kinds of privacy attacks, including malicious uses of oppnets for privacy attacks by malicious helpers. The next section describes more privacy solutions.

Some relaxation of the strictest privacy protection standards might be permissible in emergency situation, especially in life-and-death situations. For example, a victim searching for help will probably not object to an oppnet taking over her Body Area Network (BAN), controlling devices on and within her body. We will consider exploring this possibility with a full concern for legal and ethical issues involved. If we do, we will follow two basic

assumptions: (1) an entity should give up only as much privacy as is indispensable for becoming a helper for the requesting oppnet; and (2) an entity's privacy disclosure should be proportional to the benefits expected for the entity or to a broader common good. The latter is especially important in emergencies, when the goals like saving a life of one person takes precedence over the comfort of another.

Our earlier work on privacy includes a solution for privacy-preserving data dissemination [30], which we might adapt to improve the oppnet-helper privacy relationships.

Finally, we need to note that privacy (and security) in pervasive computing is a very active investigation area. We can use many other privacy solutions conceived by other researchers working on networks and in the area of pervasive computing.

6 Privacy and security challenges in oppnets

One of the main sources of security and privacy threats in oppnets is the fact that even a perfect helper authentication, performed before helpers join oppnets, will not guarantee excluding malicious devices from oppnets. The reason is that even a perfect helper authentication will not preclude abuses of authorizations by insiders. In general, oppnets have to use two lines of defense: (a) *preventive defense*, by blocking malicious helpers from joining them (e.g., by best authentication possible), and (b) *reactive defense*, by detection of malicious devices only after they join them, and their notorious behavior is detected (e.g., by intrusion detection systems).

The most important security and privacy challenges for opportunistic networks, discussed in turn in the following subsections, are:

1. Increasing trust and providing secure routing
2. Helper privacy and oppnet privacy
3. Protecting data privacy
4. Ensuring data integrity
5. Authentication of oppnet nodes and helpers
6. Dealing with specific most dangerous attacks
7. Intrusion detection
8. Honey pots and honey farms

Fig. 3 displays a general security scheme for oppnets. In the absence of

a highly trustworthy authentication mechanism all five steps marked by outgoing arrows from the adder circle are mandatory.

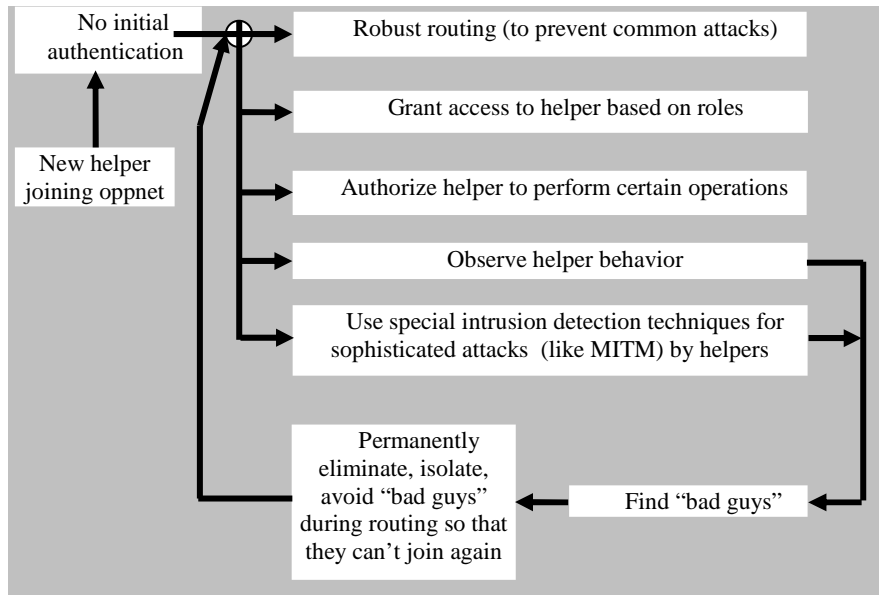


Fig. 3. The general security scheme for oppnets.

6.1 Increasing trust and providing secure routing

A list of “more trusted” devices, based on direct experience and second-hand reputation, can be maintained by an oppnet. For example, an oppnet can trust more oppnet reservists, or devices owned by certain institutions, such as devices at police stations, government offices, hospitals, public libraries, universities or reputable companies. Once a list of trusted devices is made, these devices can be used for more critical tasks that should not be entrusted to unknown devices or, even worse, distrusted devices. (A ‘black list’ of distrusted entities can be maintained as well.)

Secure routing can use both lists. Selecting a route that passes through only trusted devices (or as many trusted devices as possible) is challenging. Numerous papers have been written on individual ad hoc routing protocols. Hu

and Perrig wrote a survey of secure wireless ad hoc routing [17].

The secure wireless ad hoc routing protocol that seems most relevant to oppnets is Ariadne [18]. It is an on-demand protocol that works in the presence of compromised nodes. Ariadne uses symmetric cryptography. It authenticates routing messages using one of the three schemes:

- Shared secrets between each pair of nodes
- Shared secrets between communicating nodes combined with broadcast authentication
- Digital signatures

General solutions proposed for securing routing protocols in wireless or ad hoc networks or the Internet cannot be used directly in oppnets because of their special characteristics. Among others, oppnets are highly heterogeneous, with different processing abilities, power sources, modes of transmission, etc.

Trusted devices powered by batteries should be used sparingly to increase their lifetime, and in this way optimizing oppnet connectivity and thus routing. Having adequate battery power might be easier in oppnets than in other systems since oppnets can rely on harvesting needed resources via their growth.

6.2 Helper privacy and oppnet privacy

Some approaches for assuring privacy were mentioned in Section 5. More details for some of these solutions and other solutions are presented here.

Helper privacy

To be accepted, oppnets must assure privacy of helpers. A fear of having its privacy violated can prevent candidate helpers invited by an oppnet from joining it, or can cause reluctance (a passive or an active resistance) of candidate helpers ordered by an oppnet to join.

The first line of privacy defense for a helper are its access controls (authentication and authorization) and its intrusion prevention (using security primitives, relying on trust, using secure routing etc.). Intrusion detection should be the second line of privacy defense for helpers, helping when prevention fails or cannot be used due to its inefficiency. Elimination or isolation of bad entities from oppnets via intrusion detection is very important for benevolent nodes. The problem of enforcing access control and performing

real-time intrusion detection for oppnets are more difficult than for the Internet, wireless networks, or ad hoc networks in general because of the highly heterogeneous nature of oppnet components and the spontaneous manner in which oppnets are formed.

We investigate three helper privacy approaches: (1) extending initiator anonymity protocols ([15, 42]); (2) providing responder anonymity and anonymous data transfer via proxy techniques [42]; and (3) use of *active certificates* [4] to safeguard sensitive information or resources on helper nodes from software agents sent by an oppnet.

Oppnet privacy

Guarding an oppnet against privacy violations by a helper or by another oppnet node is equally important. Malicious helpers can join an oppnet with the purpose of violating its privacy. Since it is very difficult to uncover the motives of any helper invited or ordered by an oppnet to join, the only way to find bad helpers may be by intrusion detection.

We investigate three helper privacy approaches: (1) a solution based on automatic trust negotiation [55]; (2) using Semantic Web technologies to manage trust [8]; and (3) automatic enforcement of privacy policies, described by the Semantic Web Rule Language (SWRL) [45]. We also started investigation of a Semantic Web framework with an OWL-based ontology [11], and Rei [40, 24] and KAoS [26] policy languages to move towards context-aware policies for oppnets.

6.3 Protecting data privacy

Privacy of messages in oppnets is our next concern, considered separately from the privacy issues considered above.

Multicast from the controller

Many controller messages are intended for only a few selected nodes in the oppnet and require privacy. The lack of a shared secret or a key between the controller and the intended recipients makes the problem of providing data privacy difficult. If there is a shared secret key (for the symmetric key cryptography encryption) between the controller and intended recipients,

a capture of even a single device leads to the failure of the whole scheme. The capture might be more probable in crisis situations when providing physical protection is even more difficult.

Messages from nodes to the controller

Many messages from oppnet nodes to its controller also require privacy. Encryption is a standard way of providing such data privacy. Asymmetric key cryptography (or a public key cryptography using PKI) can be used to protect privacy of these messages.

A malicious device can pose as an oppnet controller and distribute its own public key. To prevent distribution of such a forged public key, the legitimate controller needs a secure mechanism to broadcast a public key to oppnet nodes, including candidate helpers and integrated helpers.

Messages in oppnets can be sent from one device to another device (peer to peer) in a wide area or locally. The latter case includes an intra-cluster communication among devices in a neighborhood. A local cluster head (a trusted device) can again use public key cryptography in communicating with its neighbors. A malicious device posing as a cluster head must be prevented from distributing its own forged public key.

6.4 Ensuring data integrity

Data integrity is a part of any secure communication. Digital signatures can be used to guard integrity of messages. They are often too expensive computationally for *thin* devices (like cellphones, PDAs, etc.), typically running on a limited battery power. Lightweight alternatives should be devised to guarantee integrity of data packets.

Message size may vary when it travels through an oppnet. Suppose that a message is sent from a cellphone to a base station through a PC connected to the Internet. The size of the packets traveling from the cellphone to the PC will be different from the size of the packets when they travel from the PC to the base station. If packet fragmentation and aggregation cannot be performed securely, the end-to-end security mechanisms assuring data integrity could fail.

6.5 Authentication of oppnet nodes and helpers

Delivering secret keys securely to all non-malicious devices (and only to non-malicious devices) is very difficult in ad hoc oppnet environments. Hence, relying alone on cryptography-based authentication mechanisms (such as Kerberos) is not sufficient. We need to deal with a host of sophisticated attacks, such as MITM, packet dropping, ID spoofing (masquerading), and distributed DoS attacks—all significant and potentially disabling threats for oppnets.

We investigate two helper privacy approaches: (1) a solution integrating existing techniques of authentication, authorization and accounting (AAA) ([37, 1]) to provide authentication of nodes in oppnets; (2) use of Identity Based Encryption (IBE) [16] for creating and storing pre-shared secrets, public keys and revocation lists.

6.6 Proposed solutions for dealing with specific attacks

The most dangerous attacks on oppnets and their effects can be described briefly as follows:

1. *MITM attacks*: Suppose that a malicious device is on the path connecting a victim and the rescue team. When the victim sends a help request message to the rescuers, the malicious device might capture it and maliciously inform the victim that help is on the way. It could also tamper with messages sent by the rescuers.
2. *Packet dropping*: The malicious device in the above scenario might drop some or all packets sent between the victim and the rescue team. It might capture packets at random, or forward packets containing insignificant information and drop packets containing critical information.
3. *DoS attacks by malicious devices*: False requests for help can be generated by malicious devices. They will keep the rescue team busy and unavailable for real emergencies.
4. *DoS attacks on weak links*: DoS attacks may target a “weak” device, such as a cellphone, that is critical to oppnet operation (e.g., if it is the device that connects two clusters of users). The battery of such a cellphone is a very precious resource and should be used sparingly till an alternative inter-cluster connection is found. Attacks to exhaust

battery power can occur. Some DoS attacks may target only critical weak devices.

5. *ID spoofing*: Mapping some node properties (like location of a node) into node ID by a controller can be dangerous. A masquerading malicious device can generate requests with multiple IDs, resulting in many false alarms for the rescue team. Services that need authentication can be misused if IDs can be spoofed. A device capable of spoofing ID of a trusted node or a node with critical functions can pose many kinds of attacks.
6. *Helpers masquerading as oppnet members*: Helper nodes that masquerade as oppnet members can attack the oppnet not only individually, but can form a gang for attacking the oppnet.

Research directions or initial solutions, explored by us to prevent the above attacks, can be sketched as follows:

1. *Solution directions for MITM attacks*: A person in need can send redundant messages to the controller through multiple neighbors. This will increase the chances that least one of the multiple message copies will reach the controller, even if there are attackers on some paths. So, redundancy of routes can be exploited to avoid the MITM attackers. Use of integrity regions [9] is another solution to be investigated for preventing MITM attacks.
2. *Solution directions for packet dropping*: The above idea of sending redundant messages via multiple neighbors may work if no packet-dropping adversary is situated on at least one path. Again, redundancy of routes can be exploited to avoid attackers.
3. *Solution directions for DoS attacks by malicious devices*: Upper limit can be placed on the number of requests any device can generate. Thus, it will limit the number of times any device can send a false help request. In addition, the rescue team can attempt contacting the requester to confirm an emergency request.

Other solutions under investigation include: (1) integrating a trust evaluation technique [29] and locality driven key management architecture [54]; and (2) a solution based on tagging packets with signalprints [12] and using appropriate matching rules to detect DoS attacks based on MAC address spoofing.

4. *Solution directions for DoS attacks on weak links*: Identification of weak devices, their strengthening (e.g., providing backups for them),

or minimizing their workload can counteract such attacks and maintain connectivity in oppnets.

- 5 *Solution directions for ID spoofing:* Although it is difficult to guarantee that malicious nodes will not join an oppnet, oppnet nodes can watch their neighbors for possible attempts of ID spoofing. The SAVE protocol [28] can provide routers with information needed for source address validation. This protocol needs to be modified to suit the heterogeneous nature of oppnets.
- 6 *Solution directions for helpers masquerading as oppnet members:* Helpers should be required to at least authenticate/authorize themselves before they can start inviting/ordering other nodes to join the network. We investigate a solution based on signalprints [12], which are highly correlated to physical node locations, and can detect malicious nodes lying about their MAC addresses.

6.7 Intrusion detection

Malicious devices or malicious networks can join an oppnet if an initial authentication mechanism is not adequate. There is a need to detect and isolate malicious nodes, clusters, or networks. Securely *distributing* information about malicious entities in the presence of malicious entities is a challenge. If shared securely, this reputation information can be used by all oppnet nodes to protect themselves from attackers. Even if this information can be distributed securely, avoiding the suspected entities while maintaining connectivity is a challenge.

We are investigating requirements for efficient algorithms and protocols for intrusion detection in oppnets, based on existing solutions for MANETs [39]. The characteristics of oppnets make real-time intrusion detection and response in them even more challenging than in other types of networks.

6.8 Honeypots and honeyfarms

Design of low-cost honeypots for oppnets is challenging because physical security of honeypots cannot be guaranteed for the entire lifetime of an oppnet. Observations from honeypots cannot be trusted unless secure channels of communication are established. Attackers masquerading as honeypots or

posing DoS attacks on honeypots are examples of problems that need to be solved.

We are investigating a hybrid honeyfarm architecture for oppnets that integrates the high-interaction technologies of Collapsar honeyfarm [22] and Potemkin honeyfarm [51], providing both *centralized* management and *decentralized* honeypot presence. The resulting system can be made scalable and efficient, using late binding of resources, flash cloning, and redirectors.

7 Conclusions

This chapter describes the concept of *opportunistic networks* (*oppnets*), and presents the related research challenges in privacy and security.

Oppnets constitute a newly identified category of computer networks. When deployed, oppnets attempt to detect candidate helper systems existing in their relative vicinity—ranging from sensing and monitoring, to computing and communication systems—and integrate them under their own control. When such a candidate is detected by an oppnet, the oppnet evaluates the benefits that it could realize if the candidate joins it. If the evaluation is positive the oppnet invites the candidate to become its helper. In this manner, an oppnet can grow from a small seed into a large network with vast sensing, communication, and computation capabilities.

Oppnets will facilitate many applications. As an example, they can help building an integrated network called for in various critical or emergency situations [48]. Oppnets can be used to enable connectivity in an area where any existing communication or information infrastructure has been fractured or partially destroyed. Oppnets will integrate various systems that were not designed to work together. The integration will enhance the flow of information that, for example, can assist in rescue and recovery efforts for devastated areas, or can provide more data on phenomena that are just developing, such as wildfires or flash torrents.

Answering to the identified privacy and security challenges in oppnets will contribute to advancing knowledge and understanding not only for the opportunistic networks, but will simultaneously advance the state of the art of computer privacy and security in ad hoc and in general-purpose computer networks.

We continue working on a number of the identified challenges, continuing our investigation of privacy and security in oppnets. The planned prototype opportunistic network will provide a proof of concept for our solutions, as well as stimulation and feedback necessary for fine-tuning the proposed solutions.

Acknowledgements

This work was supported in part by the National Science Foundation under Grant IIS-0242840, and in part by the U.S. Department of Commerce under Grant BS123456.

The authors would also like to acknowledge Western Michigan University for its support and its contributions to the WiSe (Wireless SensorNet) Laboratory, Computational Science Center and Information Technology and Image Analysis (ITIA) Center.

L. Lilien, a co-PI on the NSF grant providing a partial support for this research, would like to thank Professor Bharat Bhargava from Purdue University, the PI for this grant.

L. Lilien would like to thank the participants of the *International Workshop on Research Challenges in Security and Privacy for Mobile and Wireless Networks (WSPWN 2006)* for their helpful comments and feedback on oppnets. In particular, he would like to thank Mr. Hien Nguyen, a Ph.D. student at the Florida International University, for a fruitful discussion that resulted in crystallizing the idea of the oppnet reserve.

L. Lilien also expresses his thanks to the following students of his advanced computer security course for their contributions to the following research projects: (a) contributors to helper privacy and oppnet privacy: N. Bhargava, T. Goodman, V. Kalvala, H.R. Ravi, R. Rekala, A. Rudra, V. Talati, and Y. Yoder ; (b) contributors to authentication of oppnet nodes and helpers: V.V. Krishna, P.E. Miller, and A.K. Yedugani; (c) contributors to dealing with specific attacks: S. Chittineni, N. Jawanda, D. Koka, S. Pulimamidi, and H. Singh; and (d) contributors to intrusion detection, honeypots and honeyfarms: R. Dondati and S. Mapakshi.

Any opinions, finding, conclusions or recommendation expressed in the paper are those of the authors and do not necessarily reflect the views of the funding agencies or institutions.

References

1. D. Balfanz, D. K. Smetters, P. Stewart and H. C. Wong, "Talking To Strangers: Authentication in Ad-Hoc Wireless Networks," *Symposium on Network and Distributed Systems Security (NDSS '02)*, San Diego, CA, Feb. 2002.
2. S. Bansal and M. Baker, "Observation based cooperation enforcement in ad hoc networks," CoRR, July 2003. Available at <http://www.informatik.uni-trier.de/~ley/db/journals/corr/corr0307.html#cs-NI-0307012>.
3. B. Bhargava, L. Lilien, A. Rosenthal, and M. Winslett, "Pervasive Trust," *IEEE Intelligent Systems*, vol. 19(5), Sep./Oct.2004, pp. 74-77.
4. N. Borisov, "Active Certificates: A Framework for Delegation," M.S. Dissertation, University of California, Berkeley, 2002.
5. S. Buchegger and J. Le Boudec, "Performance Analysis of the CONFIDANT Protocol: Cooperation Of Nodes — Fairness in Dynamic Ad-hoc Networks," *13 IEEE/ACM Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc 2002)*, Lausanne, Switzerland, June 2002.
6. M. Burnside, D. Clarke, Mills, A. Maywah, S. Devadas, R. Rivest, "Proxy-Based Security Protocols in Networked Mobile Devices", *17th ACM Symp. on Applied Computing (SAC'02)*, Madrid, Spain, March 2002, pp. 265 – 272.
7. R. Campbell, J. Al-Muhtadi, P. Naldurg, G. Sampemane and M.D. Mickunas, "Towards Security and Privacy for Pervasive Computing," *IEEE Computer*, vol. 34 (12), Dec. 2001, pp. 154-157.
8. O. Can, and M. Unalir, "Distributed Policy Management in Semantic Web," Dept. of Computer Engineering, Ege University Bornova, Izmir, Turkey, 2006.
9. S. Čapkun and M. Cagalj, "Integrity Regions: authentication through presence in wireless networks," *5th ACM Workshop on Wireless Security (WiSe'06)*, Los Angeles, CA, Sep. 2006, pp. 1 – 10.
10. Y. Chen, C. Jensen, E. Gray, V. Cahill, J. Seigneur, "A General Risk Assessment of Security in Pervasive Computing," Technical Report TCD-CS-2003-45, Dept. of Computer Science, Trinity College, Dublin, Ireland, Nov. 2003.
11. A. Dersingh, R. Liscano, and A. Jost, "Using Semantic Policies for Ad Hoc Coalition Access Control," *International Workshop on Ubiquitous Access Control (IWUAC'06)*, San Jose, CA, 2006.
12. D.B. Faria and D.R. Cheriton, "Detecting Identity Based Attacks in Wireless Networks Using Signalprints," *5th ACM Workshop on Wireless Security (WiSe'06)*, Los Angeles, CA, Sep. 2006.
13. K. Farkas, J. Heidemann, and L. Iftode, "Intelligent Transportation and Pervasive Computing," *IEEE Pervasive Computing*, vol. 5 (4), Oct. 2006,

- pp. 18 - 19.
14. S. Farrell, J. Vollbrecht, P. Calhoun, L. Gommans, G. Gross, B. DB Bruijn, C. DB Laat, M. Holdrege, and D. Spence, "AAA Authorization Requirements," RFC 2906, The Internet Society, Aug. 2000. Available at: <http://www.faqs.org/rfcs/rfc2906.html>.
 15. I. Goldberg and D. Wagner, "Taz Servers and the Rewebber Network: Enabling Anonymous Publishing on the World Wide Web," *First Monday*, 1998.
 16. K. Hoeper and G. Gong, "Bootstrapping Security in Mobile Ad Hoc Networks Using Identity-Based Schemes with Key Revocation," Technical Report CACR 2006-04, Centre for Applied Cryptographic Research, Waterloo, Canada, 2006.
 17. Y.-C. Hu and A. Perrig, "A Survey of Secure Wireless Ad Hoc Routing," *IEEE Security & Privacy, Special Issue on Making Wireless Work*, Vol. 2(3), May/June 2004, pp.28-39.
 18. Y.-C. Hu, A. Perrig, and D.B. Johnson, "Ariadne: A Secure On-Demand Routing Protocol for Ad Hoc Networks," *8th Ann. Intl. Conf. Mobile Computing and Networking (MobiCom 2002)*, Atlanta, Georgia, Sep. 2002, pp. 12–23.
 19. Y. Huang, W. Fan, W. Lee, and P. S. Yu, "Cross-feature analysis for detecting ad-hoc routing anomalies," *23rd International Conference on Distributed Computing Systems (ICDCS 2003)*, Providence, RI, May 2003, pp. 478–487.
 20. M. Humphrey and M. Thompson, "Security Implications of Typical Grid Computing Usage Scenarios," *10th IEEE International Symposium on High Performance Distributed Computing*, San Francisco, CA, Aug. 2001, pp. 95-103.
 21. H. Inerowicz, S. Howell, F. Regnier, and R. Reifenberger, "Protein Microarray Fabrication for Immunosensing," *224th American Chemical Society (ACS) National Meeting*, Aug. 2002.
 22. X. Jiang and D. Xu, "Collapsar: a VM-based Architecture for Network Attack Detection Center," *13th Usenix Security Symposium*, San Diego, CA, Aug. 2004. Available at: www.ise.gmu.edu/~xjiang/pubs/JPDC06.pdf
 23. W. E. Johnston, K. Jackson, and S. Talwar, "Security Considerations for Computational and Data Grids," *10th IEEE Symposium on High Performance Distributed Computing*, San Francisco, CA, Aug. 2001.
 24. L. Kagal and T. Berners-Lee, "Rein: Where Policies Meet Rules in the Semantic Web," Technical Report, MIT, 2005.
 25. L. Kagal, T. Finin, and A. Joshi, "Trust-Based Security in Pervasive Computing Environments," *IEEE Computer*, vol. 34 (12), Dec. 2001, pp. 154-157.
 26. L. Kagal, M. Paolucci, N. Srinivasan, G. Denker, T. Finin, and K. Sycara,

- “Authorization and Privacy for Semantic Web Services,” *First International Semantic Web Services Symposium, AAAI 2004 Spring Symposium*, March 2004.
27. L. Kagal, J. Undercoffer, F. Perich, A. Joshi, T. Finin, and Y. Yesha, “Vigil: Providing Trust for Enhanced Security in Pervasive Systems,” Dept. of CSEE, University of Maryland Baltimore County, August 2002. Available at: <http://ebiquity.umbc.edu/paper/html/id/54/Vigil-Providing-Trust-for-Enhanced-Security-in-Pervasive-Systems>
 28. J. Li, J. Mirkovic, M. Wang, P. Reiher, and L. Zhang. "SAVE: Source Address Validity Enforcement Protocol," UCLA Technical Report 01-0004, Los Angeles, CA, 2001.
 29. X. Li, J. Slay, and S. Yu, “Evaluating Trust in Mobile Ad hoc Networks,” *The Workshop of International Conference on Computational Intelligence and Security*, Dec. 2005, Xi’an, China. Available at: http://esm.cis.unisa.edu.au/new_esml/resources/publications/evaluating%20trust%20in%20mobile%20ad-hoc%20networks.pdf
 30. L. Lilien and B. Bhargava, “A Scheme for Privacy-preserving Data Dissemination,” *IEEE Transactions on Systems, Man and Cybernetics Cybernetics, Part A: Systems and Humans*, Vol. 36(3), May 2006, pp. 503-506.
 31. L. Lilien and A. Gupta, Personal Communication, Department of Computer Science, Western Michigan University, Kalamazoo, MI, Dec. 2005.
 32. L. Lilien, A. Gupta, and Z. Yang, "Opportunistic Networks and Their Emergency Applications and Standard Implementation Framework," submitted for publication.
 33. L. Lilien, Z. H. Kamal, and A. Gupta, "Opportunistic Networks: Research Challenges in Specializing the P2P Paradigm," *3rd International Workshop on P2P Data Management, Security and Trust (PDMST'06)*, Kraków, Poland, Sep. 2006.
 34. M. Locasto, J. Parekh, A. Keromytis, S. Stolfo, “Towards Collaborative Security and P2P Intrusion Detection,” *2005 IEEE Workshop on Information Assurance and Security*, June 2005. Available at: <http://www1.cs.columbia.edu/ids/publications/locasto2005iaw.pdf>
 35. P. Michiardi and R. Molva, “CORE: A collaborative reputation mechanism to enforce node cooperation in mobile ad hoc networks,” *Sixth IFIP Conference on Security Communications, and Multimedia (CMS 2002)*, Portorož, Slovenia, Sep. 2002.
 36. A. Mishra, K. Nadkarni, A. Patcha, "Intrusion Detection in Wireless Ad Hoc Networks", *IEEE Wireless Communications*, Vol. 11(1), Feb. 2004, pp. 48-60.
 37. H. Moustafa, G. Burdon, and Y. Gourhant, “Authentication, Authorization and Accounting (AAA) in Hybrid Ad hoc Hotspot's Environments,” *4th*

- International Workshop on Wireless Mobile Applications and Services on WLAN Hotspots (WMASH 2006)*, Los Angeles, CA, Sep. 2006.
38. M. Mutka, Personal Communication, Department of Computer Science and Engineering, Michigan State University, East Lansing, MI, Dec. 2006.
 39. D. Nordqvist, L. Westerdahl and A. Hansson, "Intrusion Detection System and Response for Mobile Ad hoc Networks," FOI-R 1683, Command and Control Systems, User Report, July 2005.
 40. D. Olmedilla, "Security and Privacy on the Semantic Web," in: M. Petkovic and W. Jonker (editors), *Security, Privacy and Trust in Modern Data Management*, Springer, 2006.
 41. L. Pelusi, A. Passarella, and M. Conti, "Opportunistic Networking: Data Forwarding in Disconnected Mobile Ad Hoc Networks," *IEEE Communications*, Vol. 44(11), Nov. 2006, pp. 134-141.
 42. A. Pfitzmann and M. Waidner, "Networks Without User Observability — Design Options," *Eurocrypt '85, Workshop on the Theory and Application of Cryptographic Techniques*, Linz, Austria, April 1985, pp. 245–253.
 43. G. Selander *et al.*, "Ambient Network Intermediate Security Architecture," Deliverable 7.1, v. 3.2, Ambient Networks Project, Sixth Framework Programme, European Union, Feb. 2005. Available at: www.ambient-networks.org/phase1web/publications/D7-1_PU.pdf.
 44. L. Spitzner, "Definitions and Value of Honeypots", GovernmentSecurity.org, May 2002. Available at: <http://www.trackinghackers.com/papers/honeypots.html>
 45. "SWRL: A Semantic Web Rule Language Combining OWL and RuleML," The World Wide Web Consortium (W3C), May 2004. Available at: <http://www.w3.org/Submissions/SWRL/>
 46. P. Thibodeau, "Pervasive computing has pervasive problems," *ComputerWorld*, Vol. 36(41), Oct. 7, 2002.
 47. J. Undercoffer, F. Perich, A. Cedillnik, L. Kagal, A. Joshi, "A Secure Infrastructure for Service Discovery and Access in Pervasive Computing," Technical Report, TR-CS-01-12, Dept. of CSEE, University of Maryland Baltimore County, 2001. Available at: <http://citeseer.ist.psu.edu/cedilnik01secure.html>.
 48. U.S. Government Printing Office via GPO Access, "Combating Terrorism: Assessing the Threat of a Biological Weapons Attack." Last accessed on December 15, 2005. Available at: http://www.armscontrolcenter.org/cbw/resources/hearings/snsvair_20011012_combating_terrorism_assessing_biological_weapons_attack.htm
 49. L. Venkatraman and D. Agrawal, "A novel authentication scheme for ad hoc networks", *Wireless Communications and Networking Conference (WCNC 2000)*, Vol. 3, Chicago, IL, Sep. 2000, pp. 1268-1273.

50. J. Vollbrecht, P. Calhoun, S. Farrell, L. Gommans, G. Gross, B. de Bruijn, C. de Laat, M. Holdrege, D. Spence, "RFC 2905 - AAA Authorization Application Examples", Network Working Group, The Internet Society, Aug. 2000. Available at: www.faqs.org/rfcs/rfc2905.html.
51. M. Vrable, J. Ma, J. Chen, D. Moore, E. Vandekieft, A. C. Snoeren, G. M. Voelker, and S. Savage, "Scalability, Fidelity and Containment in Potemkin Virtual Honeyfarm," *ACM Symposium on Operating System Principles (SOSP'05)*, Brighton, UK, Oct. 2005.
52. W. Wagealla, C. English, S. Terzis, and P. Nixon, "A Trust-based Collaboration Model for Ubiquitous Computing," *UbiComp2002 Security Workshop*, Goteborg, Sweden, Sept./Oct. 2002.
53. V. Welch, F. Siebenlist, I. Foster, J. Bresnahan, K. Czajkowski, J. Gawor, C. Kesselman, S. Meder, L. Pearlman, and S. Tuecke, "Security for Grid Services," *Intl. Symp. on High Performance Distributed Computing*, Seattle, WA, June 2003, pp. 48-57. Available at: citeseer.ist.psu.edu/welch03security.html.
54. G. Xu and L. Iftode, "Locality Driven Key Management Architecture for Mobile Ad hoc Networks," *IEEE International Conference on Mobile Ad-hoc and Sensor Systems*, Fort Lauderdale, FL, Oct. 2004.
55. T. Yu, M. Winslett, and K. E. Seamons, "Supporting Structured Credentials and Sensitive Policies through Interoperable Strategies for Automated Trust Negotiation," *ACM Transactions on Information and System Security (TISSEC)*, 6(1), Feb. 2003.
56. D. Zamboni, "Using Internal Sensors for Computer Intrusion Detection," CERIAS Technical Report 2001-42, CERIAS, Purdue University, West Lafayette, IN, Aug. 2001.