Christian Gottron, André König, Ralf Steinmetz :
A Survey on Security in Mobile Peer-to-Peer Architectures—Overlay-Based vs. Underlay-Based Approaches. In: Future Internet 2, no. 4, p. 505-532, October 2010. ISSN 1999-5903.
www.mdpi.com/1999-5903/2/4/505/.

future internet
ISSN 1999-5903
www.mdpi.com/journal/futureinternet

Article

A Survey on Security in Mobile Peer-to-Peer Architectures—Overlay-Based *vs.* Underlay-Based Approaches

Christian Gottron *, André König and Ralf Steinmetz

Multimedia Communications Lab, TU Darmstadt, Rundeturmstrasse 10, 64283 Darmstadt, Germany;
E-Mails: andre.koenig@kom.tu-darmstadt.de (A.K.); Ralf.Steinmetz@kom.tu-darmstadt.de (R.S.)

* Author to whom correspondence should be addressed; E-Mail: Christian.Gottron@kom.tu-darmstadt.de;
Tel.: +49 6151 164577; Fax: +49 6151 166152.

Received: 10 September 2010; in revised form: 1 October 2010 / Accepted: 8 October 2010 / Published: 13 October 2010

Abstract: Mobile Ad hoc networks (MANET) and Peer-to-Peer (P2P) networks share central characteristics such as their distributed and decentralized nature. Combining both networking paradigms results in a Mobile Peer-to-Peer (MP2P) system that operates independently from a preexisting infrastructure. Securing MP2P networks in terms of availability and robustness as basic demands in envisioned application scenarios like first responder operations is a challenging task. In this article, we present a survey of selected threats and of state of the art countermeasures for MANETs and P2P networks. Further, we discuss the efficiency of MANET and P2P security mechanisms when applied in MP2P networks.

Keywords: mobile ad hoc network; peer-to-peer; network security; mobile peer-to-peer

---

1. Introduction

Mobile Ad hoc Networks (MANETs) and Peer-to-Peer (P2P) networks share concepts of self-organization and decentralization. Both systems operate without a central, coordinating entity and do not require a preexisting communication infrastructure for operation. Considering these similarities, combining MANETs and P2P networks is reasonable to obtain a fully distributed, decentralized and infrastructure-less communication substrate. Several architectures for such mobile P2P (MP2P) networks already exist. The proposed approaches differ in terms of the layer(s) on which they are implemented and in terms of the deployed mechanisms for the lookup process that is responsible for

locating services available in the system as well as in terms of the self-organized network maintenance. Combining both architectures at a single layer or enabling a cross-layer information exchange may increase network efficiency since the overhead for network operation and maintenance can be reduced.

Mostly, an MP2P network benefits from its inherent self-x features as, e.g., a single node failure will (most probably) not affect the whole network. Yet, on the other hand, securing decentralized networks in common and MP2P networks in particular is a challenging task. The routing algorithms for both P2P networks and MANETs have to rely on the benign behavior of the nodes forming the network to forward messages to the destination correctly. In both architectures, nodes join and leave the network in a flexible way. Due to this, both are susceptible to routing or Sybil attacks. On the other hand, various MANET and P2P related security measures were proposed to increase the robustness of the routing algorithms and to detect Sybil attacks. However, the question on which layer the countermeasures has to be implemented to optimize the network security and efficiency in MP2P networks is still open.

In this paper, we survey different approaches proposed for establishing Mobile P2P (MP2P) networks and briefly summarize the basics of the underlying MANET and P2P architectures. Further, we survey selected attacks on both basic architectures independently from each other and derive the effects of those attacks on MP2P networks. As the attacks we selected are well known, several countermeasures exist. We compare these countermeasures and discuss their efficiency concerning the requirements of MP2P networks. Further, we propose a first outline of a cross-layer countermeasure for MP2P networks that may increases the robustness to multiple routing attacks.

In Section 2 we introduce P2P networks; we describe basic architectures including lookup algorithms and point out major challenges. In the next section, we focus on MANET architectures and challenges. In Section 4, we present the three different categories of MP2P architectures that have been proposed by now and, in Section 5, we survey attacks and countermeasures for the three presented architectures. In the last section, we summarize the discussed attacks on mobile P2P networks and provide an outlook on our future work in this area.

2. Foundations of Peer-to-Peer

This chapter provides background information on P2P networks. After a short introduction, we present P2P architectures and lookup algorithms. We further identify major challenges regarding P2P security.

2.1. Introduction

P2P networks are used for file sharing, voice chat and many more services nowadays. While we focus on Distributed Hash Tables (DHT) in this paper, several other architectures were also proposed. We basically have to distinguish between *structured* and *unstructured* networks [1]. *Unstructured P2P* networks can be further subdivided into *centralized*, *pure*, and *hybrid P2P* networks.

Centralized P2P networks require a central entity for the lookup of the provided services (e.g., locating stored objects in a file sharing scenario). Requests for content and notifications of new content have to be sent to this entity which stores the IP addresses of nodes offering services, thus providing lookup information. This way, every request can be served after a single hop in the P2P overlay but at

the cost of a very high load at the central entity and a single point of failure. Napster [1] is a well known example of a *centralized P2P* network.

While *centralized P2P* systems still contain elements of traditional Client/Sever architectures, all nodes in *pure P2P* networks are treated equally. Flooding-based lookup algorithms are required as the source node is unaware of the logical location of services available in the P2P overlay (*i.e.*, the requesting peer has no knowledge on which peer offers the requested service). No single point of failure exists and no complex lookup algorithm is required. Yet, a considerable lookup overhead is introduced since the lookup requests are flooded through the network. Freenet [2] and Gnutella 0.4 [3] are examples for *pure P2P* networks.

The third kind of unstructured architectures are *hybrid P2P networks*. In hybrid networks, we distinguish between leafnodes and superpeers. Every superpeer is responsible for a set of leafnodes and maintains information about services provided by these nodes. Lookup requests are usually not flooded in the whole network but sent to the superpeers. This results in a reduced overhead at the cost of a more complex lookup algorithm compared to *pure P2P* networks. Gnutella 0.6 [3] and KaZaA [4] are examples for *hybrid P2P* networks.

Distributed Hashtables (DHT) are *structured P2P* networks in which each node is assigned an ID (e.g., by hashing its IP address). By this ID, nodes are identified and can be looked-up within the P2P network. During lookup, cooperation of other nodes in the network is required, as in the routing tables that are used to route lookup requests only the addresses of a fraction of the available nodes in the network is stored. While the particular lookup algorithm differs depending on the specific architecture, most DHTs scale logarithmically to the network size regarding the routing table size and the number of hops required to forward a request from source to destination.

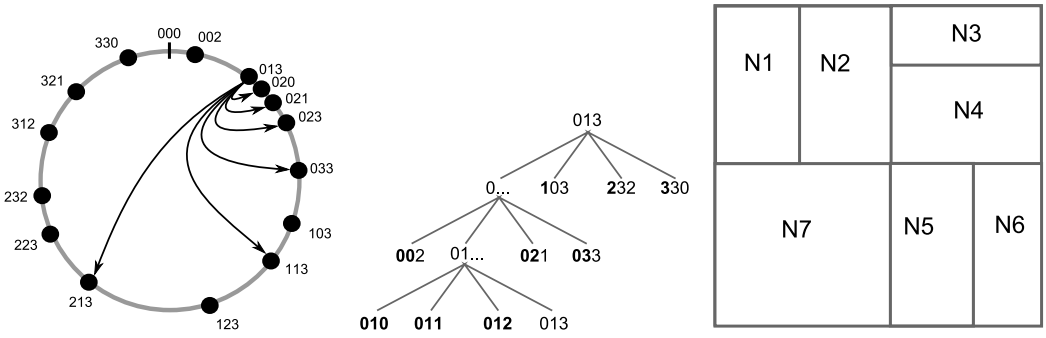
2.2. Architectures

As described in the previous section, structured P2P networks provide a good trade-off between complexity and scalability. Many DHT algorithms were proposed which can be distinguished by the network topology. We will introduce the most common topologies in the following.

2.2.1. Ring

Ring based architectures as, e.g., Chord distribute the node-IDs on a circular ID-space as shown in Figure 1(a). In Chord, every object with an object-ID o is stored at the node with the next higher node-ID n ($n > o$). The lookup is based on the node-ID and is done by forwarding the request clockwise on the circular ID-space. To offer shortcuts to nodes located far away in the ring topology, Chord uses a finger table that stores the required mapping of IP addresses to distant node IDs (the fingers are represented by arrows in Figure 1(a)). The finger table of node n is structured in i rows where each row provides the address of the node with the ID $nodeID_i = nodeID_n + 2^{i-1}$. This results in lookup algorithm that is scalable to the network size regarding the routing table size, as only a small part of the networks nodes are stored at the finger table, and to the hop length of the request due to the structure of the finger table.

Figure 1. DHT architectures.



(a) Ring (Chord) (b) Tree (Tapestry) (c) Hypercube (CAN)

2.2.2. Tree

Tapestry is a well-known example for a DHT that is based on a tree architecture. Each node maintains a prefix-based lookup table. Each row i of the routing table links to nodes with i matching prefixes as shown in Figure 1(b). E.g., Node 013 provides links to Nodes **00**2, **02**1 and **03**3 at the second row ($i = 1$). Whenever a node initiates or receives a lookup request, the object-ID is compared with the node-ID. The length of the matching prefix determines the row of the routing table where the address of the next hop node is stored. This way, the size of the routing table as well as the efficiency of the lookup algorithm in terms of overlay hops required to deliver a lookup request scale logarithmically with respect to the network size in number of peers.

2.2.3. Hypercube

The Content Addressable Network (CAN) is a hypercube architecture that is based on a d-dimensional ID-space. Nodes are not identified by a singular point in this d-dimensional space but are represented by an area within the ID-space (as shown in Figure 1(c)). Objects, on the other hand, are assigned a singular point in the ID-space for which the node that is assigned to this area is responsible for. The lookup algorithm of CAN is kept simple as every node forwards the request to the direct neighbor in the direction of the object's d-dimensional coordinates.

2.2.4. Hybrid

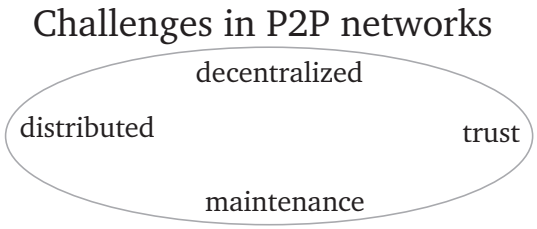
A hybrid architecture combines the algorithms of other architectures. Pastry for example uses a tree-based primary routing table and a ring-based leafset. While the primary routing table determines how to forward requests to nodes that are far away in terms of the difference of the node IDs, the leafset is used when requests have to be forwarded in the direct neighborhood.

2.3. Challenges

DHTs operate completely *distributed* and, due to this, no *trusted* entity for key management is available. Thus, offering security services such as access control in a *decentralized* P2P network is challenging. As outlined in the previous section, DHT lookup algorithms have to rely on the cooperation

of other nodes in the network. However, this can not be assumed in real-world applications and malicious nodes may misuse this weakness to launch attacks on the P2P network. As nodes may join and leave the network at any time, routing tables have to be *maintained* on a regular basis. Objects maintained by nodes which leave the network have to be managed by another node with an appropriate node-ID. This also may be misused by malicious nodes, since the information exchanged during the maintenance phase may be manipulated such that invalid entries are added to the routing table. All mentioned challenges are shown in Figure 2.

Figure 2. Challenges in Peer-to-Peer networks.



3. Foundations of MANETs

In this section, we provide basic information on MANETs. Starting with a short introduction, we will present routing algorithms for MANETs and introduce major security challenges.

3.1. Introduction

MANETs are self-configuring networks established by mobile wireless nodes. In order to transmit messages to nodes which are not in direct transmission range, nodes between source and destination have to operate as router and forward the messages. No preexisting infrastructure is required. Like P2P networks, MANETs work in a distributed and decentralized way, thus avoiding single points of failure as the routing tasks of each node may be performed by nearby nodes. On the other hand, MANETs require special routing algorithms and several challenges regarding security occur due to the characteristics of the network. These challenges have to be addressed to achieve a reliable availability of the network which is required in envisioned application scenarios where no infrastructure is available, like first responder scenarios (e.g., *DUMBO* [5], *HiMoNN* [6]), development projects (e.g. *One Laptop per Child* [7]) or car-to-car communication (e.g., *CAR 2 CAR Communication Consortium* [8]).

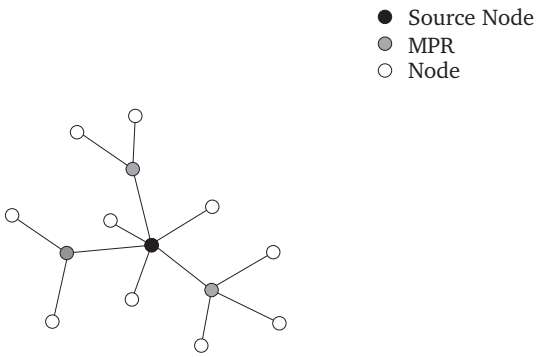
3.2. Architectures

Several routing algorithms for MANETs were proposed in the last years. These may be distinguished by their routing characteristics. For this paper, we differentiate between *proactive*, *reactive*, *hybrid*, and *geographical* algorithms.

In *proactive* approaches, routing tables are updated periodically and, thus, fresh routes to all nodes in the network are readily available (if the network is not partitioned). Due to this, no routing delay occurs before (and during) data transmission. On the other hand, overhead is generated as routing tables must be maintained regularly due to node mobility. The *proactive* Optimized Link State Routing Protocol (OLSR) [9], as an often referenced example, utilizes a link state algorithm for routing but also reduces

overhead by selecting Multipoint Relays (MPR). Every node has to determine its MPRs by identifying the smallest set of direct neighbors required to cover every second hop neighbor as shown in Figure 3. In order to reduce the routing overhead, link state messages are sent to the MPRs only.

Figure 3. MPRs selected by an OLSR node.



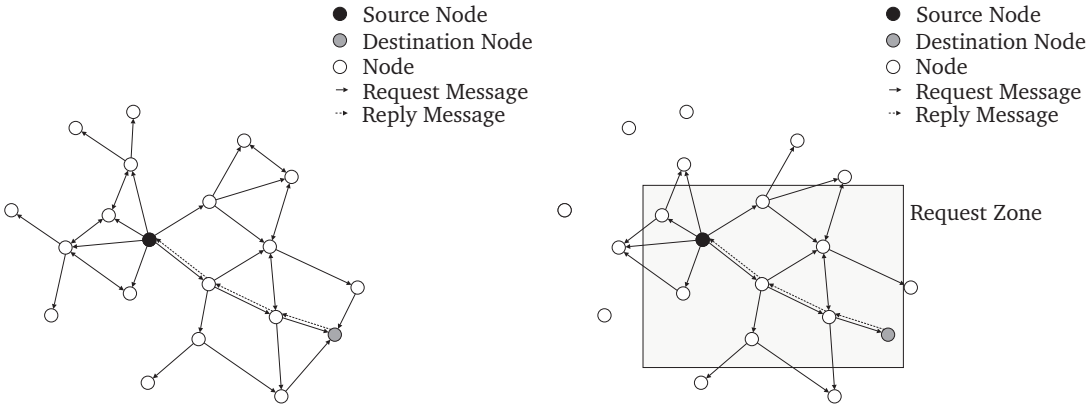
Reactive algorithms, in contrast to *proactive* approaches, start routing for a specific destination only on demand, not unless a route is required for data transmission. As the topology in MANETs may change constantly due to node mobility, routing overhead may be significantly reduced when a route discovery is initiated only when a route is required. On the downside, when data has to be transmitted to a node for which no entry in the routing table exists, a route has to be established before the first packet can be sent. The Ad hoc On-Demand Distance Vector (AODV) [10] routing protocol is a well-known and often referenced *reactive* protocol for MANETs. Here, when a route is required, the source node has to broadcast a Route Request message (RREQ). Each node in transmission range has to check whether it is the destination of this request or whether a valid route to the destination is stored in the routing table. If so, the node sends a Route Reply message (RREP) to the source. Otherwise the RREQ is re-broadcasted (as shown in Figure 4(a)). A node drops duplicates of RREQs to prevent loops. A route is established when the source node receives the RREP.

Hybrid protocols combine proactive and reactive approaches. While routing to nodes close to the source is done in a *proactive* way, routing to nodes that are further away is performed *reactively*. This way, routing overhead is reduced and a communication with nodes close to the source is not affected by routing delay. The Zone Routing Protocol (ZRP) [11] is a well known *hybrid* protocol. Routes to nodes within a specific local zone are maintained *proactively* while routes to nodes further away are established on demand. When the destination is not within the local zone, a request is sent to the nodes at the border of this zone. Those nodes check their own local zone for the requested node and forward the request to their own border nodes again if the node could not be found within their zone. This proceeds until the destination node is found.

When the network nodes are aware of their *geographical* position, routing can be optimized by harnessing this information. The geographical position of the destination node can be used to limit the dissemination of routing messages to a specific area. This results in a reduced routing overhead as routing messages are not broadcasted in the whole network. For example, the Location-Aided Routing (LAR) determines an expected zone around the last known position of a destination node. Based on the expected zone, a request zone is defined around the position of the source node and the expected zone. Route requests are broadcasted in this request zone only and, due to this, routing overhead is reduced (as

shown in Figure 4(b)). An alternative approach, LAR proposes varying the request zone by intermediate nodes as they may have more recent information about the position of the destination node.

Figure 4. MANET routing algorithms.

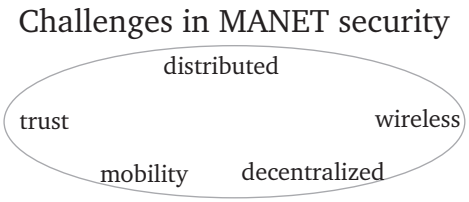


(a) Reactive (b) Geographical

3.3. Challenges

MANETs are *distributed* and *decentralized* networks. No central entity is available to provide security services. Thus, security mechanisms for key distribution, access control, *etc.*, have to be implemented in a decentralized and, therefore, more complex way. Examples are approaches utilizing threshold cryptography to replace a central instance by a majority vote of the network nodes. Further, the available bandwidth is strongly limited and has to be shared with all nodes in transmission range. Due to this, countermeasures have to be limited in bandwidth requirements. In order to transmit messages to nodes not adjacent to the source, multi-hop transmissions are required. Therefore, every node forwarding those messages has to be *trustworthy*. Otherwise, intermediate nodes showing a malicious behavior are able to redirect or drop messages instead of forwarding them correctly. As the nodes are *mobile*, the topology of a MANET changes permanently. Due to this, routes have to be adapted frequently and neighbors must be detected periodically. As mobile nodes are mostly battery-powered, energy is strongly limited. As adopted CPUs with low energy consumption are mostly deployed in mobile nodes, processing power is also limited. Due to this, complex and highly resource consuming cryptography might not be applicable for all scenarios. Further, as MANETs are *wireless* networks, they are susceptible to passive attacks as eavesdropping and, as the devices are mobile, they may be stolen and compromised. Those challenges are also shown in Figure 5.

Figure 5. Challenges in mobile Ad hoc networks.



4. Mobile Peer-to-Peer Architectures

In this chapter, we introduce MP2P networks as combination of a MANET underlay and a P2P overlay. After a short introduction, we provide information on existing approaches and identify major challenges for network security in MP2P networks.

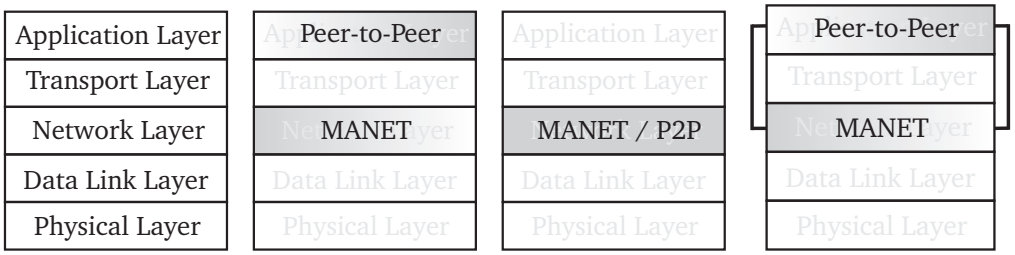
4.1. Introduction

As MANETs and P2P networks share central characteristics, combing both leads to a fully distributed architecture. Those MP2P networks consist of a MANET underlay combined with a P2P overlay. The P2P algorithm provides a lookup service on the MANET underlay. The major challenges in developing MP2P architectures are the overhead caused by the routing table maintenance combined with the strongly limited resources of the MANET underlay.

4.2. Architecture

While underlays based on (mostly) wired and infrastructure-based systems such as the Internet provide sufficient bandwidth for DHTs even when routing tables have to be maintained constantly due to the changing topology, resources of MANETs as underlay are more limited. Further, as the topology changes more often due to the mobility of nodes in MP2P networks, routing tables have to be maintained more frequently. Due to this, more control traffic is generated in an MP2P network compared to a traditional infrastructure-based P2P network but less bandwidth is available. Therefore, MP2P architectures have to reduce the overhead generated by the routing algorithm and the routing table maintenance. Recently, several approaches were proposed to combine MANETs and P2P networks. The resulting MP2P architectures can be distinguished by the implementation of the P2P algorithm into *layered*, *integrated* and *cross-layered* approaches as shown in Figure 6.

Figure 6. MP2P architectures.



(a) Basic layer model (b) Layered MP2P model (c) Integrated MP2P model (d) Cross-layered MP2P model

4.2.1. Layered

Layered architectures are a straightforward approach for adapting traditional P2P concepts to MP2P. Here, the MANET underlay and the P2P overlay are strictly separated; no cross-layer information is harnessed to optimize the system. As the routing and maintenance overhead of traditional P2P

architectures would exceed the strongly limited resources of the MANET underlay, several techniques to reduce the maintenance overhead were proposed for *layered* approaches.

In order to reduce overhead on mobile nodes with limited resources, the Mobile Stealth DHT architecture proposed in [12] distinguishes between mobile and static nodes. Static nodes are preferred for data storage and routing, while mobile nodes only have to manage the traffic generated by their own requests. This way, most overhead is shifted to the static nodes which are assumed to have stronger resources than mobile nodes. Further, the overhead generated by object relocation as a result of the mobility of nodes (mobility related churn) is avoided. This results in a reduced number of sent messages and an increased availability of objects.

To influence the routing maintenance phase directly, Castro *et al.* propose the Bamboo DHT as P2P overlay of an MP2P network [13]. The Bamboo routing algorithm is similar to Pastry but differs in the routing maintenance. While Pastry updates the routing tables whenever the topology changes, Bamboo maintains the routing tables at specific time intervals. By increasing these intervals, routing related traffic is reduced.

Several approaches for MP2P networks harness location awareness of nodes to minimize routing overhead. Whenever node-IDs are generated based on the geographical position of the nodes, the routing algorithm may benefit from this information. The PeerNet [14] architecture splits the ID-space in several equal zones and assigns each virtual zone to a geographical area. Virtual residences are introduced as a geographical area where a node may be found with the highest probability. A proxy is required for each virtual residence to provide information about the nodes that reside in this virtual area. Each mobile node stores a mobility profile at the proxy of the virtual residence when leaving this zone. The mobility profile provides information based on which the current geographical position of the node can be estimated. As PeerNet is combined with the geographical MANET routing algorithm PILOT [14], overhead may be reduced as no flooding based routing algorithm is required. Another example for geographical routing is the location aware Bamboo DHT presented by Millar *et al.* [15]. This approach uses a landmarking system to assign node-IDs to nodes. Landmarking architectures split the ID-space in equal sized groups and assign a landmark-ID to a specific node within this group. This node broadcasts the landmarking signal. Each node within a landmarking zone forwards the landmarking signal. A node that receives a landmarking message of a foreign landmark group compares the distance to the foreign landmark node with the distance to its own landmark node. If the new landmark node is closer, the node changes the node-ID and joins the closer landmark zone. As each landmark zone assigns node-IDs with a specific prefix, nodes with matching prefixes are physically close to each other. Due to this, a location aware routing is combined with a reduction of the overhead of the routing table maintenance phase by the Bamboo DHT. A major advantage of a location aware routing is the reduced overhead as the number of underlay hops is reduced.

4.2.2. Integrated

Integrated approaches reduce routing overhead by combining the MANET and P2P protocols at the network layer as shown in Figure 6(c). This way, the number of required routing tables is reduced as no separate MANET and P2P routing tables are required. Further, the routing algorithm may benefit

from synergy effects such as updating overlay routing tables based on overheard underlay messages as described below.

By reducing the complexity and the number of required entries in the routing table, routing and maintenance overhead may be reduced. Further, knowledge of the physical neighborhood may improve the routing efficiency of the routing protocol. Scalable Source Routing (SSR) proposed by Fuhrmann [16] introduces an MP2P architecture based on Chord. Similar to Chord, SSR is based on a ring architecture. However, SSR simplifies routing maintenance as no finger tables are used but only direct virtual neighbors. Further, physical neighbors are listed in the routing table. During routing, a node decides whether a virtual or physical neighbor is closer to the destination. Due to the simplified routing table, maintenance overhead is reduced. Further, by combining MANET and P2P routing tables, the route length can be reduced without increasing the complexity or the routing maintenance required.

Additionally, awareness of physical neighbors at intermediate nodes between source and destination in the underlay can improve the efficiency of the routing protocol. The Virtual Ring Routing (VRR) algorithm presented in [17] combines the set of virtual and physical neighbors in a single routing table at the network layer. As VRR is based on a Ring architecture, the set of virtual neighbors consists of r nodes with a node-ID next to the node-ID of the routing table owner. Those r nodes consist of the $r/2$ virtual neighbors clockwise and the $r/2$ virtual neighbors counter-clockwise. VRR provides bidirectional links to the neighbors and maintains them proactively. The physical node set consists of the nodes in direct transmission range. The routing table provides the next underlay hop with an endpoint closest to the requested ID when an object is requested. As each intermediate node checks its own routing table before forwarding the request, requests may be rerouted when intermediate nodes are aware of shorter routes. This way, routing is simplified and overhead is reduced.

As a further optimization, every overheard or forwarded packet may be harnessed to update the routing tables in order to reduce maintenance overhead. This is the basic concept of EKTA [18] and DPSR [19]. Both integrated approaches implement a Pastry DHT on the network layer. Similar to Pastry, a primary and a leafset routing table are provided. Complete multi-hop routes are stored at these routing tables to avoid underlay flooding during overlay routing. Shortest routes are preferred to reduce overhead. Whenever a message is overheard, the routing table is updated by routing information obtained from the message. Due to this, overhead generated by routing table maintenance is strongly reduced.

4.2.3. Cross Layered

Besides layered and integrated approaches, cross-layered architectures were proposed (as shown in Figure 6(d)). By providing network layer information to the P2P overlay, routing maintenance overhead can be reduced and the performance of the MP2P network can be improved.

CROSSRoad [20] combines a proactive MANET routing like the OLSR algorithm with a DHT overlay. As proactive routing algorithms have routes to all nodes of the network readily available, a cross layer approach could provide a complete knowledge of the network topology for the DHT. This way, each lookup may be accomplished after a single overlay hop. Yet, on the other hand, using a proactive routing protocol in a highly mobile, large scale network would result in a very high overhead.

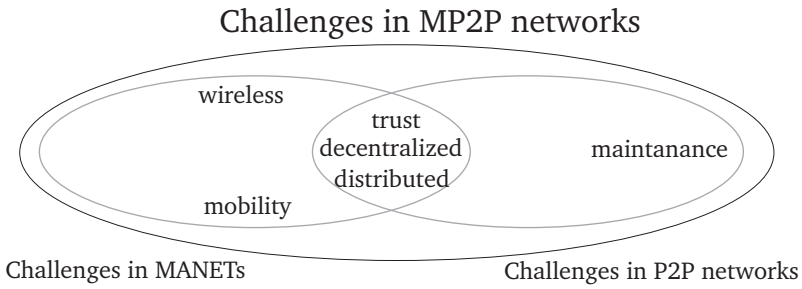
Similar to the layered approach presented in [15], MADPastry [21] builds upon a landmark-based geographical node-ID distribution. In order to reduce the maintenance overhead, MADPastry truncates

Pastry's routing table. MADPastry uses a routing table with a single row and, thus, provides only the first hop in the same way as Pastry in order to route to the correct landmark cluster. The rest of the routing is based on the leafset routing table. Besides the direct next and previous hop neighbor which are maintained proactively, routing tables are maintained by cross-layer information obtained from overheard messages. This way, overhead generated by maintaining the routing tables can be reduced by harnessing cross-layer information. Further, the assignment of node-IDs based on geographical information reduces the number of required underlay hops per request similar to the approach presented by Millar *et al*. [15].

4.3. Challenges

As MP2P systems combine MANETs and P2P networks, challenges of both architectures are inherited as shown in Figure 7. For instance, as routing table maintenance and object relocation of P2P networks requires a high amount of bandwidth and MANETs provide only *limited resources*, security algorithms have to be strongly limited in bandwidth. Further, as both architectures are *decentralized*, the availability of a Certification Authority (CA) can not be assumed. Due to this, *trust* is a major problem as the routing algorithm has to rely on the cooperation of the nodes. As both architectures have a constantly *changing topology*, in the case of MANETs due to node mobility and in the case of P2P systems due to nodes going offline, an increased amount of churn is generated compared to traditional P2P networks. This results in an increased overhead for object relocation.

Figure 7. Challenges in mobile Peer-to-Peer networks.



5. Attacks and Countermeasures

In this section, we present selected attacks on MANETs and P2P networks and discuss possible countermeasures. As MP2P networks are susceptible to both MANET and P2P attacks, we discuss the effects of these attacks on the combination of a MANET underlay with a P2P overlay. We further discuss the efficiency of the presented countermeasures for MANETs and P2P systems in the context of MP2P networks.

5.1. Attacking P2P

DHTs provide an efficient and decentralized lookup service. Yet, due to decentralization and multi-hop transmission, DHTs are vulnerable to attacks that exploit the lookup algorithm or the lack

of admission control to deny services. We present selected attacks on DHTs in this chapter and discuss countermeasures that were proposed in the recent years.

5.1.1. Attacks on the P2P Routing Algorithm

The DHT lookup algorithm has to rely on the benign, cooperative behavior of the nodes that form the P2P network. Due to this, nodes that manipulate the lookup process have a serious influence on the network.

During a *Storage and Retrieval Attack* [22], a malicious nodes denies the access to objects maintained by itself. As it is hard to distinguish between unavailable and denied objects during a lookup, most P2P systems are not inherently able to detect this attack. The efficiency of this attack depends directly on the fraction of malicious nodes, since objects are equally distributed to nodes.

Sit and Morris [22] introduced the *Incorrect Lookup Routing* attack. Malicious nodes drop or redirect incoming route requests instead of forwarding them to the correct destination. This is a simple and straight forward attack that affects the routing of lookup requests in DHTs. During the lookup process, a request is sent from the source to a node that is closer to the destination regarding the node-ID. Whenever a node receives a request, it checks whether it is the destination or whether it has to forward the request to the next overlay hop closer to the destination. For this recursive routing, Castro *et al.* introduced an analytical model that describes the effects of malicious nodes that drop packets (as shown in Equation 1). The packet delivery ratio (σ) is based on the average number of hops (h) per request and the fraction of malicious nodes (f). While dropping lookup requests affects the routing process strongly, redirecting requests is a more subtle attack and may affect even DHTs protected by countermeasures.

$$\sigma = (1 - f)^h \tag{1}$$

5.1.2. Attacks on the P2P Routing Tables

P2P routing tables have to be updated regularly to provide reliable routes when lookup requests have to be forwarded. Most DHTs update and optimize their routing tables by using entries provided by the routing tables of other nodes in the DHT. Malicious nodes may forge routing tables and provide routing table entries with links to inappropriate or non-existing nodes [22]. Further, corrupted routing updates can increase the fraction of routing table entries that refer to malicious nodes [23]. As most DHTs use metrics like proximity or age to optimize their routing table entries, malicious nodes may exploit these metrics to maximize the impact of the attack. Manipulation of the routing updates in order to increase the number of malicious nodes in a routing table of a benign node also results in a cascading effect. As each malicious node that is added to a routing table will provide routing updates, the fraction of malicious nodes increases as shown in Equation 2 [23]. The fraction of malicious routing entries f_{table} increases whenever a routing update is initiated as each benign node provides entries with a fraction of $f_{network}$ malicious nodes, but each malicious node provides malicious entries only.

$$f_{table} = (1 - f_{table}) * f_{network} + f_{table} * 1 \tag{2}$$

5.1.3. Sybil Attacks on P2P Networks

A malicious node may join a distributed network multiple times to generate multiple identities. A malicious node having multiple virtual identities can increase the impact of other attacks like the *Incorrect Lookup Routing*. Further, as P2P networks often use reputation mechanisms to avoid *free riders*, single nodes with multiple virtual identities may improve their own reputation.

5.2. P2P Countermeasures

The presented attacks may severely affect the availability of services in the P2P network. Therefore, multiple countermeasures were proposed. As a basis for all countermeasures presented in the following, node-IDs in P2P networks have to be distributed by a central entity or randomly (e.g., by hashing the IP-address). Otherwise, malicious nodes could benefit from freely selectable node-IDs (e.g., by selecting an ID next to the object which should be denied).

5.2.1. Countermeasures for P2P Routing Attacks

As the *Incorrect Lookup Routing* affects the routing algorithms of DHTs strongly, several countermeasures were proposed. Most of those countermeasures increase the robustness of the routing algorithm by adding redundancy to the network.

The recursive routing algorithms are susceptible to routing attacks as the source of the request can not detect malicious nodes due to the recursive behavior of the routing. To overcome this drawback, the *Iterative Routing* algorithm was proposed by Sit and Morris [22] as one of the first and most promising approaches. When an *Iterative Routing* is initiated, intermediate nodes have to send routing information about the next hop back to the source node. This way, the source node is able to interact with the intermediate nodes directly. Whenever an intermediate node proposes an incorrect next hop address or does not respond to a request at all, the source node is aware of this malicious behavior due to the missing reply or as the proposed next hop node does not match the expected one. After detecting a malicious node, the source node can reroute the request and bypass this node. On the downside, the lookup delay and the data overhead is increased as the source node has to resend lookup messages whenever a intermediate node does not respond and as intermediate nodes have to reply to the source node which adds one additional message per hop in the overlay.

Another prominent approach to increase the routing robustness of DHTs was introduced by Castro *et al*. [23]. They proposed a redundant recursive routing algorithm. When multiple redundant requests are sent, the probability that all of them are dropped or redirected by malicious nodes is decreased. Due to this, the *Redundant Routing* algorithm improves the delivery rate of lookup requests. On the downside, redundant routing increases lookup overhead strongly.

Srivatsa and Liu analyzed the importance of independent paths during redundant routing [24]. As long as the paths are not disjoint, a single malicious node may affect all sent requests.

Sánchez-Artigas *et al*. [25] modified the *Redundant Routing* algorithm and combined it with a reputation system. The resulting *Resistant Routing* algorithm forwards lookup requests not only to the next hop, but also to n neighbors of the next hop. Those $n + 1$ nodes forward the request to the next set of $n + 1$ nodes around (and including) the next hop node. This way, the destination node and

n of its neighbors receive the request after $log(N)$ hops. As Sánchez-Artigas *et al.* propose to store the replicas around the root of an object, the source node receives $n + 1$ keys at the end of the lookup procedure as each neighbor of the destination stores a valid replica. In order to reduce the resulting routing overhead, an additional reputation system is introduced. During the routing, each node has to rate the behavior of all of its neighbors. When a node is rated malicious (*i.e.*, when it does not forward lookup requets / answers lookup requests incorrectly), routing requests are not forwarded to this node anymore. Further, as neighbors could forward the requests to different next hop nodes, the lookup process has to be coordinated (e.g., by iterative routing) as otherwise the number of forwarded requests may increase strongly. Due to the parallel lookup, probability of a successful lookup increases. According to Sánchez-Artigas *et al.*, the reputation system works efficiently as long as the fraction of malicious nodes is below 50 %.

Combining *Redundant* and *Iterative* approaches, Hildrum and Kubiatowicz [26] proposed a parallel and iterative routing algorithm. Whenever a request is initiated, the source node has to send n lookup request messages in parallel. Each intermediate node sends a reply message proposing a set of next hop nodes to the source of the request. Based on this information, the source node selects n next hop nodes and forwards n request to those nodes. These steps are repeated until the first request reaches the destination. Due to the combination of *Redundant* and *Iterative* routing, the packet delivery rate is increased strongly. Yet, this is paid for by a high overhead and delay as multiple request messages have to be sent iteratively.

Besides increasing the redundancy of routing algorithms directly, distributing replicas can also increase the reliability of the lookup process. As every countermeasure discussed so far has to rely on the benign behavior of at least the destination node, the success probability of the lookup process can not be improved beyond the fraction of nodes behaving benign on average. Using replicas can overcome this drawback. While Castro *et al.* [23] proposed to spread the replicas in the neighborhood of the node that is responsible for the original object according to the DHT specification, Harvesf and Blough [27] proposed a replica set evenly distributed in the P2P overlay. As a result of equally distributed replicas, DHTs as Pastry can provide guaranteed disjunct multi-path routing. Due to the tree based structure of the primary routing table, storing a replica in each prefix domain of the highest level of the tree will ensure disjunct paths as different parts of the lookup table are used during the lookup for each replica. Replicas can further reduce the impact of *Storage and Retrieval Attacks* on the network, to deny access to a specific object, all nodes that are responsible for this service (*i.e.*, all nodes holding replicas) would have to behave maliciously.

As an intermediate node could pretend to be the destination of a request and respond with an incorrect result, the source has to validate each responding node that delivers an unexpected result. For this, Castro *et al.* proposed a *Routing Failure Test*. The source node has to check the virtual distance to the neighbor nodes and estimates an average node density. As most DHTs distribute the node-IDs equally in the ID space, the node density at the destination is equal to the node density at the source on average. Whenever object-ID and node-ID differ strongly regarding the node density estimated in the neighborhood of the source node, the responding node might not be the legitimate node that is responsible for the object.

5.2.2. Countermeasures for Routing Table Manipulation

The original specifications of most DHT-based P2P systems do not prevent corrupting routing tables of benign nodes by sending forged routing table updates. To address this issue, several countermeasures for a malicious routing table manipulation were proposed recently. Sit and Morris [22] suggested to send a probing message to links provided by other nodes to check the correctness of the routing information. This way, incorrect entries such as not existing nodes can be detected. Castro [23] proposed to generate a second, more secure routing table. The first routing table is the routing table as defined by the DHT, that is using the standard metrics to maintain the routing table. As most proposed metrics for routing table maintenance were developed to reduce traffic and, therefore, favor links to nodes that are in physical proximity, malicious nodes may misuse this metric and feign proximity to increase the probability of being added to the routing table. To prevent this, the second routing table uses a metric that ignores the physical proximity but adds nodes to the routing table that are virtually close to a specific node-ID. For example, whereas for the first routing table in Pastry the entries are chosen based on matching prefixes and a proximity metric, the second routing table as proposed by Castro *et al.* prefers entries that are next to the required prefix for the routing table entry combined with a predefined ending of the node-ID. As node-IDs are assigned randomly, malicious nodes can not misuse the metric of the second routing table. During routing, the first routing table is used until a lookup process fails. Whenever a routing failure occurs, rerouting is initiated based on the second routing table.

5.2.3. Countermeasures for Sybil Attacks

Douceur [28] proposed three types of authentication schemes to prevent *Sybil* attacks. Using a trusted entity (CA) to authenticate each node that wants to join the network is a centralized approach. Besides authenticating nodes centrally, each node may authenticate new nodes on its own. However, this approach does not scale to the network size. The last authentication scheme proposed by Douceur is based on trust relationships between authenticated nodes. This way, each node has to be authenticated only once by a single node when it joins the network. Other nodes accept this new node without further authentication. This approach is decentralized and scales to the network size. Yet, malicious nodes that do not show misbehavior until they are authenticated may misuse this scheme and allow other malicious nodes to join the network. Further, an open question is how to distinguish between benign nodes with a single identity and malicious nodes with multiple identities. To prove that one node does not have multiple identities, Douceur proposed that computationally expensive tasks have to be preformed by each node. By doing so, a Sybil node with n identities has to provide n times as much resources as a node with one identity, and the number of identities per node can be limited. Yet, as the resources available at a node can only be estimated, and due to the heterogeneity of devices in a P2P network, the tasks are designed for the weakest device assumed such that each node has to perform the same tasks to join the network. Due to this, malicious nodes may generate multiple virtual identities as long as they are able to perform the required tasks. Still, at least the number of identities is limited by this countermeasure for which Douceur proposes three tasks that demand (1) communication, (2) storage, or (3) computation resources from all identities simultaneously. An example for limiting the number of identities by computationally expensive tasks was proposed by Borisov [29]. Each node in the network

has to solve *computational puzzles* at regular intervals. This way, even nodes with high resources are limited in generating multiple identities. The puzzles require a large amount of computation to be solved but the solutions are simple to verify. Due to this, the neighbors of the challenged node can verify the solution without a high computational cost.

Castro *et al*. [23] proposed to bind the virtual node to a real world identity. This way, users have to authenticate themselves instead of their node. Also paying real world resources (e.g. money) to enter the network would limit the number of identities. Yet, these approaches require a CA for reliable user authentication.

Analyzing the relationship of nodes can also be used to avoid or at least to reduce the impact of *Sybil* attacks. *SybilGuard* [30] is a prominent approach that analyzes graphs generated by using social networks in order to separate malicious nodes from honest ones. Besides social relationships, relationships between nodes during booting can also provide insights that can be harnessed to reduce the impact of *Sybil* attacks. As an example, Danezis *et al*. [31] proposed a modified routing metric to reduce the impact of malicious nodes in a DHT. Danezis *et al*. assume that *Sybil* nodes join the network at a single point of the bootstrapping graph. Due to this, malicious nodes are mostly connected to other malicious nodes only and, therefore, can not affect the rest of the network. Thus, the routing algorithm can benefit when related nodes are preferred during routing.

Further, several approaches were proposed to improve the robustness of trust-based metrics to Sybil attacks [32,33]. While those trust-metrics were developed to limit the impact of free riders, the proposed solutions also prevent that *Sybil* nodes improve their own reputation.

5.3. Attacking MANETs

MANETs are vulnerable to several attacks due to their decentralized and wireless characteristics. In the following, we present selected attacks on MANETs that affect both the routing algorithm and the data transmission. We further discuss *Sybil* attacks in the context of MANETs.

5.3.1. Malicious Behavior during Route Discovery and Maintenance

In order to transmit packets to nodes which are not in direct transmission range of the source node, intermediate nodes have to operate as routers to forward the packets to their destination. For this, the MANET routing algorithm has to rely on the cooperation of the intermediate nodes. Yet, the required cooperation can not be assumed without restrictions in real-world environments. Misbehaving nodes may tamper with the routing algorithm by, e.g., forging, redirecting, or dropping routing messages.

A wide range of routing attacks was introduced in the recent years. The *Blackhole* attack is one of the most aggressive routing attacks on MANETs. *Blackholes* manipulate the routing algorithm to redirect network traffic. For this, the routing metric is misused to increase the probability that the traffic is routed via the *Blackhole*. As a result, the malicious node is capable of analyzing the transmitted data or of denying the service by dropping the received packets.

Further, the routing algorithm may be misused for *Resource Consumption* attacks. We consider for example a reactive routing protocol and a malicious node that initiates a high number of route requests. As the bandwidth of MANETs is strongly limited and request messages are forwarded by each neighbor

of the malicious node, those attacks can affect large parts of the network. Further, by forging the source field of the request messages, benign nodes can be accused for misbehavior and countermeasures can be evaded.

Besides an intended misbehavior, a selfish behavior can also affect the functionality of a MANET. Selfish nodes refuse to cooperate in acting as routers for other nodes in order to save resources. Since the functionality of MANETs is based on the cooperation of nodes, selfish nodes affect the availability of the network.

5.3.2. Malicious Behavior during Data Transmission

After routes are established successfully, data transmission may still fail as a result of malicious behavior. Malicious nodes may e.g. cooperate during route discovery but deny to forward packets whenever data has to be transmitted. Those *Selective Forwarding* attacks may be hard to detect, especially when a node behaves benignly temporarily and only drops packets sent by specific nodes.

Further, by sending forged messages for route maintenance, transmission loops can be created. Those loops result in an increased traffic and a denial of service as the packets are dropped when their *time-to-live* exceeds. The ADOV routing protocol for example has been shown to be vulnerable to *Loop Forming* attacks [34].

5.3.3. Sybil Attacks on MANETs

The susceptibility of MANETs to Sybil attacks is comparable to the susceptibility of P2P systems. MANETs are distributed networks and, therefore, in most scenarios no centralized admission control exists. Due to this, a single node can generate multiple virtual identities in a network. Like in P2P systems, malicious nodes may generate multiple virtual identities to increase the impact of other attacks, to evade trust-based countermeasures, or to claim an increased amount of resources.

5.4. MANET Countermeasures

In order to increase robustness and reliability of MANET routing algorithms, several countermeasures were proposed in the recent years. We present selected countermeasures for the MANET attacks introduced previously.

5.4.1. Countermeasures for Attacks during Routing and Data Transmission Phase

Reliable routes and correct data transmission are crucial for a MANET. Most countermeasures introduced in this chapter are capable of preventing malicious behavior during both routing and data transmission phases. The countermeasures can be categorized into intrusion prevention, intrusion detection, and intrusion response mechanisms.

Preventive mechanisms are mostly based on cryptographically secured routing protocols. The *Authenticated Routing for Ad hoc Networks* (ARAN) [35] is one of the first proposed crypto-based intrusion prevention algorithms for MANETs. *ARAN* uses a Public Key Infrastructure (PKI) to authenticate routing messages. Both route request and route reply messages are signed by the originating

node. Each intermediate node validates received messages before these messages are again signed and forwarded. This way, ARAN can prevent attacks that are based on altering routing messages. Yet, on the other hand, a trusted entity that provides cryptographic certificates is required. Further, private key cryptography is computationally expensive. Therefore, further approaches as, e.g., the *Secure Efficient Ad hoc Distance vector (SEAD)* [36] routing protocol, the *Secure Routing Protocol (SRP)* [37], and the *Secure AODV (SAODV)* [38] routing protocol were proposed to overcome or reduce the impact of those drawbacks. By using one-way hash chains instead of asymmetric cryptography, *SEAD* reduces computational resources required for authentication. *SRP* is based on a similar approach by using message authentication codes (*i.e.*, symmetric cryptography and a shared key) instead of asymmetric cryptography for authentication of routing messages. On the other hand, *SAODV* combines asymmetric cryptography with hash chains. *SAODV* distinguishes between mutable fields (as, e.g., the hop count) and static fields (as, e.g., source and destination addresses) of routing messages. As mutable fields have to be modified on each hop from source to destination, computationally cheap hash chains are used for authentication. As static fields have to be signed only at the source, asymmetric cryptography is used.

Besides cryptographic approaches, Yi *et al.* [39] introduced a trust-based and less costly approach to prevent malicious behavior. By mapping predefined real-world relations to the network, trusted nodes can be identified. Those trusted nodes are assumed to behave benignly. As long as sufficient trusted nodes are available, no malicious node is involved in routing or data forwarding. Yi *et al.* motivated their work with a military scenario where the military rank is defined as a trust level that is mapped directly to nodes. Nodes with high trust level are preferred for routing to ensure that messages are delivered correctly. Although the trust-based approach is less costly compared to previously presented cryptographic intrusion prevention mechanisms, it can only be used in scenarios where trust-relationships between nodes exist.

As intrusion prevention can fail, a network has to be capable of detecting ongoing malicious behavior. Therefore, intrusion detection is required. One of the first Intrusion Detection Systems (IDS) for MANETs is the *Watchdog* IDS introduced by Marti *et al.* [40]. Assuming bidirectional wireless links, each node in a wireless network receives all messages sent by nodes in transmission range. The *Watchdog* algorithm is based on analyzing these overheard messages to verify the behavior of nodes in the neighborhood. Whenever a malicious node drops a received message, the predecessor on the route can detect this misbehavior as the malicious node has received but not forwarded a message. This information can be used to exclude the malicious node from routes. Furthermore, as long as the data is not encrypted, neighbor nodes are able to detect forged or modified messages. In order to operate correctly, *Watchdog* requires bidirectional links and wireless interfaces operating in promiscuous mode. Since *Watchdog*, several other IDSs were proposed. For instance, Kurosawa *et al.* [41] introduced a more specialized IDS for the *Blackhole* attack in AODV-based MANETs. In AODV, each routing message sent by a node is marked by a unique sequence number. These sequence numbers are used to determine the freshness of the routing information. *Blackholes* tamper with the sequence numbers to feign fresh routes and, therefore, increase the probability that routes are established based on the forged messages. To detect forged messages, Kurosawa *et al.* proposed a metric based on the difference between the received sequence number and stored sequence numbers from previously established routes with the same destination node. When the sequence numbers differ strongly, a *Blackhole* may be involved.

After detecting malicious nodes, Intrusion Response Systems (IRS) exclude malicious nodes from routes. The *Pathrater* was proposed by Marti *et al*. [40] in combination with the *Watchdog* IDS. Each node in the neighborhood has a trust level defined by the *Pahtrater*. The data transmission is monitored by the *Watchdog* and whenever a node does not forward a packet correctly, the trust level is decreased. During routing, the average trust level per route is calculated and the route with the highest trust level is preferred. Further, the path length is considered by the *Pathrater* to ensure that short routes (regarding the number of hops) are preferred over longer ones with an equal trust level.

A prominent combination of a trust-based IRS with an IDS was proposed by Buchegger *et al*. [42]. The *Cooperation of Nodes-Fairness in Dynamic Ad-Hoc Networks* (CONFIDANT) system consists of the *Monitor*, the *Trust Manager*, the *Reputation System* and the *Path Manager*. The *Monitor* detects misbehavior of neighbor nodes while the *Reputation System* records detected misbehavior of monitored nodes. When a threshold is exceeded, the *Trust Manager* sends an alarm message to neighbors and the *Path Manager* removes malicious nodes from routing tables. This way, the routing algorithm avoids malicious nodes when they are detected. Both approaches, *Watchdog* with *Pathrater* and *CONFIDANT* are based on detecting malicious nodes by overhearing packets sent in the neighborhood combined with a trust-based metric. However, while a *Watchdog* node has to detect each malicious node on its own, *CONFIDANT* nodes cooperate and share information about malicious nodes. Furthermore, besides bypassing malicious nodes, *CONFIDANT* also denies routing and forwarding services to detected malicious nodes to discourage malicious behavior.

A more recent IDS and IRS combination was presented in [43]. Here, a distinction is made between *unknown*, *known* and *companion* trust levels for neighbor nodes. The trust level depends on the behavior of the node and can be improved from the initial *unknown* level up to the high *companion* level. The metric according to which the behavior of neighbor nodes is rated, is defined as the ratio of correctly forwarded foreign packets and originated own packets. This results in more reliable routes, as trustworthy nodes are preferred for routing.

5.4.2. Countermeasures for Sybil Attacks

As MANETs are decentralized and, therefore, a central trusted entity is (most probably) not available, detecting *Sybil* attacks is very challenging. The countermeasures for Sybil attacks proposed by Douceur [28] can also be used in the context of MANETs. Yet, as resources in a MANET are strongly limited and as network nodes differ strongly in the availability of resources (from mobile phones to high-performance computers), those countermeasure may not be efficient.

Newsome *et al*. analyzed *Sybil* attacks in sensor networks. They proposed multiple countermeasures [44] to prevent or, at least, limit *Sybil attacks*. Newsome *et al*. assumed that each node in a network has a single network card and, therefore, is only capable of transmitting and receiving data at a single channel simultaneously. The *Radio Resource Testing* was proposed as a countermeasure that is based on this assumption. Each node detects *Sybil* nodes in transmission range by assigning each neighbor to a specific channel. Each neighbor has to transmit a message at a specific time on this channel. Whenever a neighbor does not transmit the message on the assigned channel at the expected time, a potential *Sybil* node is detected. This test has to be repeated multiple times to detect as many *Sybil* nodes as possible in the neighborhood. The approach is simple to implement but requires bandwidth for

the active probing for malicious nodes. Further, *Radio Resource Testing* has to be repeated by each node in the network as no collaboration is performed. Newsome *et al.* additionally introduced the *Code Attestation* and the *Registration* countermeasures. As the code deployed on malicious and benign nodes differs, verifying the memory of a node provides information on whether this node behaves benignly. Also, registering and authenticating each node in the network at a central entity and using a list of known benign nodes may prevent *Sybil* nodes.

Another promising approach uses the mobility of nodes in MANETs to detect *Sybil* attacks. For this, Piro *et al*. introduced the *Passive Ad hoc Sybil Identity Detection (PASID)* [45]. Each node of the network has to keep track of all nodes in transmission range by overhearing sent messages. This way, the sender of a transmitted message can be identified and, thus, neighbors can be detected. As *Sybil* nodes share the same physical location, nodes which always appear together are suspicious to be a single malicious node. Nodes in transmission range have to be monitored for a sufficiently long time until a node can be accused to be malicious. The specific amount of required time differs due to the mobility of nodes and the scenario size. In order to optimize the efficiency of the detection system, Piro *et al*. proposed a collaboration of nodes during monitoring. By exchanging collected data, malicious nodes can be detected more efficiently and faster. Yet, Piro *et al*. lack to provide a mechanism to identify trustworthy nodes that can be used for a collaborative monitoring. As *PASID* may falsely suspect benign nodes that travel in a group, Piro *et al*. introduced an extension to their algorithm. The *PASID with Group Detection (PASID-GD)* algorithm is capable of distinguishing between malicious and benign nodes that travel together in close proximity by monitoring collisions on the Medium Access (MAC) Layer. Whenever multiple benign nodes are in transmission range, collisions occur since nodes eventually start transmitting at the same time. In contrast, as *Sybil* nodes share the same network card, data has to be sent in a serial way and, thus, no collisions occur. This results in a strongly reduced probability of false positives during *Sybil* detection.

Tangpong *et al*. [46] followed a similar approach to identify neighbors by overheard messages. In contrast to Piro *et al*, Tangpong *et al*. suggested an approach on how to prevent a misuse of the collaboration scheme for the detection of *Sybil* attacks. They proposed a PKI-based approach to enable a reliable collaboration and to validate exchanged observations between nodes to increase the effectiveness of the countermeasures. The observations are signed before they are transmitted in order to avoid benign nodes getting accused falsely by modified or forged observation messages. Due to the combined effort of multiple nodes, efficiency can be increased. Yet, this happens at the cost of a required PKI and the resource requirements of asymmetric encryption.

5.5. Attacking MP2P

MP2P networks are affected by both MANET and P2P attacks. Due to the characteristics of MP2P networks, (some of the) attacks may show more severe effects on MP2P networks than on MANETs or P2P networks individually.

As both systems are susceptible to insider attacks that deny cooperation during data transmission, MP2P networks are affected strongly by this kind of attacks. Especially in layered MP2P networks, denying correct forwarding on network layer would affect the packet delivery ratio and, thus, the lookup process strongly as, due to the multi hop data transmission on application and network layer, a high

amount of network layer requests is required to perform a lookup for an object. Due to the high number of required hops, few malicious nodes in the network are sufficient to affect the routing process with a high probability.

Regarding Sybil attacks, MP2P networks may be attacked on both layers. Due to this, a malicious node will most probably attack the layer with the weakest countermeasure. Therefore, the probability of success for generating multiple identities in an MP2P network is increased compared to a MANET with a more robust overlay or a P2P system with a more robust underlay, respectively.

Further, MANET attacks can be modified to attack the P2P services. E.g., modified *Blackholes* may redirect only routes to specific nodes to deny objects managed by them. On the other hand, malicious nodes may misuse P2P mechanisms like the routing table maintenance to increase the traffic and initiate a resource consumption attack on the MANET. Therefore, MANET robustness is required to provide reliable services by the P2P network but a benign behavior on the P2P network is also required as, otherwise, the underlying MANET can be affected.

5.6. Countermeasures in MP2P

For both P2P systems and MANETs, countermeasures for the attacks presented in the previous sections were proposed in the recent years. (Some of) the approaches may be used to provide security and robustness in MP2P networks. Yet, questions regarding possible synergy effects as well as possible limitations when the countermeasures are deployed in MP2P networks are still open. We discuss the efficiency of countermeasures for P2P systems and MANETs when applied in MP2P networks in this section.

MANETs and P2P systems are vulnerable to malicious behavior during routing or data transmission. Whereas countermeasures proposed for MANETs are mostly based on trust, encryption, and monitoring the behavior of other nodes, most P2P countermeasures provide robustness by increasing the redundancy of the lookup requests and the stored objects. These redundancy-based countermeasures such as *Iterative*, *Redundant*, or *Resistant Routing* require an increased amount of bandwidth and, thus, may be not efficient regarding the bandwidth-constraint MANET underlay of an MP2P network. Further, also *Replicas* require an increased amount of resources, since objects have to be stored and maintained at multiple nodes. Yet, as the availability of nodes can not be guaranteed in a MANET, using *Replicas* might be unavoidable in MP2P networks. The efficiency of the *Routing Failure Test* depends strongly on the implementation of the MP2P system. For example, this test can not be used in architectures as *MADPastry* since the distribution of node-IDs also depends on the number of nodes in the particular physical area. On the other hand, countermeasures developed for MANET routing algorithms are still applicable in MP2P systems as no further limitations occur due to the combination with a P2P network. Therefore, MANET countermeasures are only limited by the same well known factors as when used in a pure MANET architecture. For instance, computational power required for asymmetric encryption or predefined trust relationships can still be problematic. On the other hand, some of the security mechanisms developed for MANETs such as *Watchdog* can be improved by harnessing cross-layer information to increase the overall security of the MP2P network. We present an example for such a cross-layer approach in Section 6.1.

Similar to the countermeasures for the routing attack, countermeasures for the *Sybil* attack in P2P systems are also limited by the MANET underlay. Therefore, the countermeasures proposed by Douceur that are based on resource testing (e.g., computational puzzles) are limited due to the underlay resource restrictions. Further, approaches that are based on trust or social relations such as *SybilGuard* can be compromised as the mobile devices can get stolen easier than fixed devices locked up at home or office in, e.g., an Internet-based P2P system. On the other hand, as CAs are most likely not available in an MP2P network, binding nodes to real-world identities or charging money for each node that joins the network is hardly realizable. Yet, providing robustness of the routing algorithm by using the bootstrapping graph to establish routes as proposed by Danezis *et al*. is applicable in an MP2P network. Other promising countermeasures for *Sybil* attacks in MP2P networks were originally proposed for MANETs. The *PASID-GD* system works similar to *Watchdog* and is efficient in MP2P networks but can be improved by harnessing cross-layer information. Altogether, a single countermeasure may be sufficient to prevent *Sybil* attacks in MP2P networks if an integrated or a cross-layered MP2P architecture is used.

Also to prevent manipulations of routing tables in P2P networks, multiple countermeasures were proposed but have to be analyzed regarding their applicability in MP2P networks. Sending test messages to each entry added to the routing table to check whether a node is available can be reasonable when the MP2P routing table stores complete underlay routes. Otherwise, a routing approach would be required that is based on flooding the network when the requested node does not exist. On the other hand, generating and maintaining a second routing table in parallel to the main routing table as proposed by Castro *et al.* could consume large amounts of the available bandwidth. Still, this approach could be reasonable in MP2P implementations that maintain routing tables by harnessing overheard messages.

To sum it up, the resource restriction of MP2P networks affects most security measures that were proposed for P2P systems operating on top of a less resource-constraint underlay such as the Internet. Most security mechanisms proposed for MANETs are still applicable in the context of MP2P but can be improved by harnessing cross-layer information.

To the best of our knowledge, only sparse related work on security in MP2P networks exists. Čapkun *et al*. proposed a face-to-face authentication system for MP2P networks [47]. As nodes are mobile, nodes can be carried during bootstrapping to the bootstrapping node for a direct authentication. The authentication process is based on social interaction as the owner of the bootstrapping node has to decide whether a node may join the network. Kutzner *et al.* propose a cryptographic approach to secure the integrated Scalable Source Routing (SSR) protocol [48]. They authenticate routing messages and certify links to prevent malicious nodes from forging messages based on a PKI. In the PEPERS project [49], a basic framework for MP2P security mechanisms was developed. By now, only the framework itself is introduced and no further MP2P security mechanisms were proposed.

6. Future Work and Conclusion

In the following, we introduce the basic concept of our cross-layer countermeasure for routing attacks in MP2P systems and summarize the core findings on security mechanisms for MP2P networks.

6.1. Future Work: A Cross Layer Approach to Increase the Robustness of a MP2P System

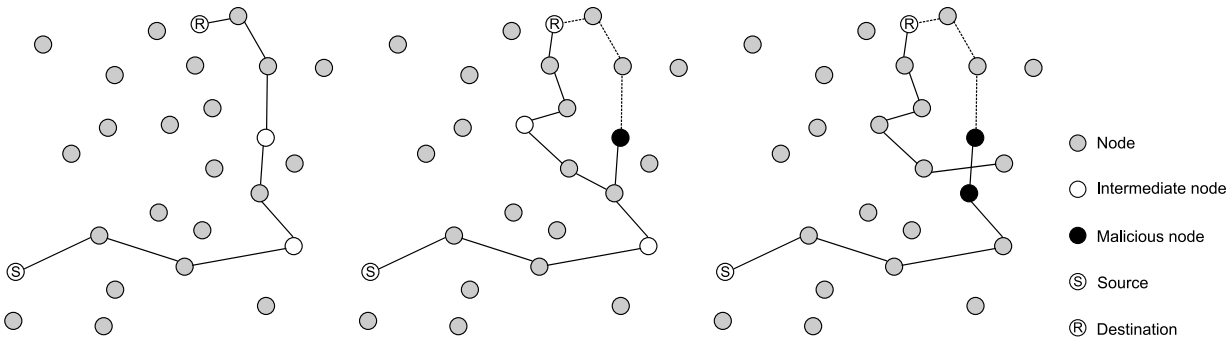
While most traditional security mechanisms for P2P systems are efficient when an underlay with sufficient bandwidth is available, they fail when resources are limited. Therefore, we discuss a cross-layered approach in this section. Cross-layering can improve the robustness of MP2P networks without strongly increasing the required bandwidth.

In promiscuous mode, the neighbors of malicious nodes are able to detect dropped messages, as proposed for the *Watchdog*. Further, most DHT algorithms as Pastry or Chord forward requests closer to the destination with each hop in the P2P overlay. Therefore, malicious nodes that redirect messages can be detected whenever a packet is forwarded to a node that is further away regarding the logical distance to the object-ID. In the case of dropped or redirected lookup requests, the monitoring neighbor node initiates a new lookup and forwards the request to the determined next intermediate overlay node. As there may be more than a single node in transmission range, benign neighbor nodes have to coordinate themselves before flooding the network with several redundant lookup requests. In the best case, the node that has forwarded the request before a misbehaving node was reached restarts the lookup process.

Several countermeasure were proposed to provide secure and robust MANET routing. In the following, we assume a benign behavior of each node regarding the underlay routing and message forwarding and only consider malicious behavior in the overlay. We assume that both source and destination nodes behave benignly. As the source node benefits from a correct routing, a malicious behavior regarding a routing attack is very unlikely. Whenever a destination node behaves maliciously, replicas can be used to increase the availability of objects stored at malicious nodes. Therefore, we consider monitoring intermediate overlay nodes only.

While in a P2P system using a recursive lookup algorithm on top of a wired underlay, malicious nodes can most probably drop messages unnoticed, in wireless networks neighbors can detect dropped or redirected messages. For this, each node in transmission range of sender r_s and malicious receiver r_r (please note that we refer to a receiver regarding a one-hop physical wireless transmission, not a destination node of a multi hop route) monitors the packets sent by both nodes. If a malicious or selfish node receives a packet but neither forwards it nor responds with a reply message to the source node (as shown in Figure 8(b)), all neighbors of both nodes ($r_s \cap r_r$) including the sender can detect this attack.

Figure 8. Examples for an attack on an MP2P network and the reaction of the cross-layer countermeasure.



(a) Benign nodes (b) Single malicious node (c) Colluding malicious nodes

Selfish intermediate overlay nodes affect the delivery rate of lookup requests as they drop received requests instead of forwarding them in order to reduce the energy consumption. This behavior can be monitored by neighbors which, in turn, can forward the packet on their own as each node in the network maintains a lookup table for the P2P overlay. In order to prevent loops, unique request IDs and a time to live field have to be used. Also, when a malicious node redirects lookup request, neighbor nodes could forward the request to the correct next hop node or initiate a new lookup process. Due to this, colluding nodes may not generate a routing loop.

Regarding the cross-layer approach, several factors that affect the efficiency of security measures have to be analyzed, including node mobility and physical node density in the underlay. As an example, an intermediate node could leave reception range of a monitoring neighbor before a packet is forwarded correctly. This way, false positives could be generated, *i.e.*, the monitoring neighbor detects a packet loss where none occurred. Therefore, only neighbors that are in close proximity (regarding transmission/reception range) to the monitored node should be considered for performing monitoring tasks.

The node density defines the probability that nodes may be available, when both sender and receiver behave maliciously. Thus, the efficiency of the cross-layer countermeasure has to be evaluated regarding the node density of the network.

The following three aspects were originally presented by Marti *et al.* in the context of *Watchdog* [40]. Yet, these aspects also affect our proposed approach. (1) An ambiguous collision may occur when packets sent by intermediate nodes collide with other packets. In this case, a monitoring node can not be sure whether a packet was transmitted correctly by the monitored node. (2) Colluding malicious nodes could generate a collision on purpose. Neighbor nodes may not detect this collision and assume the packet was transmitted correctly. (3) The sender may limit transmission range. When all monitoring neighbor nodes are within reach of the limited transmission range but the next hop node is not, the monitoring may also be misled.

6.1.2. Open Problems

To sum it up, we are able to prevent lookup requests from being dropped or redirected as long as the underlay routing algorithm operates correctly. For this, each node has to monitor all nodes in transmission range and a sufficient density of monitoring nodes is required. Yet, we are still facing open problems regarding the monitoring of neighbor nodes as mentioned in [40].

6.2. Conclusion

In this paper, we discussed different approaches of combining MANETs and P2P systems into MP2P systems. We analyzed major challenges in securing MP2P networks by analyzing attack vectors of MANETs and P2P systems. We considered attacks such as Sybil or routing attacks that affect both MANETs and P2P systems and, thus, will also affect MP2P networks. We discussed the efficiency of security mechanisms that were proposed for MANETs and P2P systems independently in the context of MP2P systems. Most P2P networks and most security measures for P2P systems were developed under the assumption of a large amount of available resources. Due to this, limitations introduced by

the MANET underlay affect the P2P overlay and the security measures for P2P systems strongly. On the other hand, security mechanisms developed for MANETs are still applicable in MP2P systems. Yet, the security measures can be improved when the security mechanisms designed for MANETs and P2P systems are combined in a cross-layered approach. We introduced an adapted security mechanism for MP2P networks that harnesses a cross-layer information exchange. This security mechanism provides robustness against multiple routing attacks while keeping the generated overhead on a reasonable level, thus considering the resource constraints of MP2P systems.

References

1. Steinmetz, R.; Wehrle, K.; Götz, S.; Schollmeier, R.; Rieche, S.; Eberspächer, J.; Heckmann, O.; Darlagiannis, V.; Mauthe, A.; Koppen, C. *Peer-to-Peer Systems and Applications*; Springer: Berlin, Germany, 2005.
2. Clarke, I.; Sandberg, O.; Wiley, B.; Hong, T. Freenet: A distributed anonymous information storage and retrieval system. In Proceedings of the ICSI Workshop on Design Issues in Anonymity and Unobservability, Berkeley, CA, USA, 25–26 July 2000.
3. Kirk, P. Gnutella—A Protocol for a Revolution, 2010. Available online: http://rfc-gnutella.sourceforge.net/ (accessed on 29 September 2010).
4. Kazaa. Download KaZaA, 2010. Available online: www.kazaa.com/ (accessed on 29 September 2010).
5. Kanchanasut, K.; Tunpan, A.; Awal, M.A.; Das, D.K.; Wongsaardsakul, T.; Tsuchimoto, Y. *A Multimedia Communication System for Collaborative Emergency Response Operation in Disaster-affected Areas*; TR_2007-1; Technical Report, Internet Education and Research Laboratory (intERLab), Asian Institute of Technology (AIT): Bangkok, Thailand, 2007.
6. Industrieanlagen-Betriebsgesellschaft mbH. IABG–InfoCom–HiMoNN—An efficient and highly mobile Ad-hoc Network Node, 2010. Available online: www.iabg.de/infokom/fachthemen/himonn_en.php (accessed on 29 September 2010).
7. One Laptop per Child (OLPC), A Low-cost, Connected Laptop for the World's Children's Education, 2010. Available online: http://laptop.org/en/ (accessed on 29 September 2010).
8. CAR 2 CAR Communication Consortium. Mission & Objective, 2010. Available online: www.car-to-car.org/ (accessed on 29 September 2010).
9. Clausen, T.; Jacquet, P. Optimized Link State Routing Protocol (OLSR). *RFC* **2003**, RFC 3626.
10. Perkins, C.; Belding-Royer, E.; Das, S. Ad hoc On-Demand Distance Vector (AODV) Routing. *RFC* **2003**, RFC 3561.
11. Haas, Z. A new routing protocol for the reconfigurable wireless networks. In Proceedings 6th International Conference on Universal Personal Communications, San Diego, CA , USA, 12–16 October 1997.
12. MacQuire, A.; Brampton, A.; Rai, I.A.; Mathy, L. Performance Analysis of Stealth DHT with Mobile Nodes. In Proceedings of the 4th Annual IEEE International Conference on Pervasive Computing and Communications Workshops, Pisa, Italy, 13–17 March 2006.

13. Castro, M.; Villanueva, E.; Ruiz, I.; Sargento, S.; Kassler, A.J. Performance Evaluation of Structured P2P over Wireless Multi-hop Networks. In Proceedings of the 2nd International Conference on Sensor Technologies and Applications, Cap Esterel, France, 25–31 August 2008.
14. Gopalan, A.; Znati, T. PeerNet: A peer-to-peer framework for service and application deployment in MANETs. In Proceedings of the 1st International Symposium on Wireless Pervasive Computing, Phuket, Thailand, 16–18 January 2006.
15. Millar, G.P.; Ramrekha, T.A.; Politis, C. A Peer-to-Peer Overlay Approach for Emergency Mobile Ad Hoc Network Based Multimedia Communications. In Proceedings of the 5th International ICST Mobile Multimedia Communications Conference, London, UK, 7–9 September 2009.
16. Fuhrmann, T. Performance of scalable source routing in hybrid MANETs. In Proceedings of the 4th Annual Conference on Wireless on Demand Network Systems and Services, Obergurgl, Austria, 24–26 January 2007.
17. Caesar, M.; Castro, M.; Nightingale, E.B.; O'Shea, G.; Rowstron, A. Virtual ring routing: Network routing inspired by DHTs. *ACM SIGCOMM Comput. Commun. Rev.* **2006**, *36*, 351–362.
18. Pucha, H.; Das, S.M.; Hu, Y. Ekta: An Efficient DHT Substrate for Distributed Applications in Mobile Ad Hoc Networks. In Proceedings of the 6th IEEE Workshop on Mobile Computing Systems and Applications, English Lake District, UK, 2–3 December 2004.
19. Hu, Y.C.; Das, S.M.; Pucha, H. Exploiting the Synergy between Peer-to-Peer and Mobile Ad Hoc Networks. In Proceedings of the 9th conference on Hot Topics in Operating Systems, Lihue, HI, USA, 18–21 May 2003.
20. Delmastro, F. From Pastry to CrossROAD: CROSS-Layer Ring Overlay for Ad Hoc Networks. In Proceedings of the 3rd IEEE International Conference on Pervasive Computing and Communications Workshops, Koloa, HI, USA, 8–12 March 2005.
21. Zahn, T.; Schiller, J. MADPastry: A DHT Substrate for Practicably Sized MANETs. In Proceedings of the 5th IEEE Workshop on Applications and Services in Wireless Networks, Paris, France, 29 June–1 July 2005.
22. Sit, E.; Morris, R. Security Considerations for Peer-to-Peer Distributed Hash Tables. In Proceedings of the 1st International Workshop on Peer-to-Peer Systems, Cambridge, MA, USA, 7–8 March 2002.
23. Castro, M.; Druschel, P.; Ganesh, A.; Rowstron, A.; Wallach, D. Secure routing for structured peer-to-peer overlay networks. In Proceedings of the 5th Symposium on Operating Systems Design and Implementation, Boston, MA, USA, 9–11 December 2002.
24. Srivatsa, M.; Liu, L. Vulnerabilities and Security Threats in Structured Overlay Networks: A Quantitative Analysis. In Proceedings of the 20th Annual Computer Security Applications Conference, Tucson, AZ, USA, 6–10 December 2004.
25. Artigas, M.S.; López, P.G.; Gómez-Skarmeta, A.F. Bypass: Providing secure DHT routing through bypassing malicious peers. In Proceedings of the IEEE Symposium on Computers and Communications, Marrakech, Morocco 6–9 July 2008.
26. Hildrum, K.; Kubiatowicz, J. Asymptotically Efficient Approaches to Fault-Tolerance in Peer-to-Peer Networks. In Proceedings of the 17th international Symposium on Distributed Computing, Sorrento, Italy, 1–3 October 2003.

27. Harvesf, C.; Blough, D.M. The Effect of Replica Placement on Routing Robustness in Distributed Hash Tables. In Proceedings of the 6th IEEE International Conference on Peer-to-Peer Computing, Cambridge, UK, 6–8 September 2006.
28. Douceur, J. The Sybil Attack. In Proceedings of the 1st International Workshop on Peer-to-Peer Systems, Cambridge, MA, USA, 7–8 March 2002.
29. Borisov, N. Computational Puzzles as Sybil Defenses. In Proceedings of the 6th IEEE International Conference on Peer-to-Peer Computing, Cambridge, UK, 6–8 September 2006.
30. Yu, H.; Kaminsky, M.; Gibbons, P.B.; Flaxman, A. SybilGuard: Defending Against Sybil Attacks via Social Networks. In Proceedings of the conference on Applications, technologies, architectures, and protocols for computer communications, Pisa, Italy, 11–15 September 2006.
31. Danezis, G.; Lesniewski-Laas, C.; Kaashoek, M.F.; Anderson, R.J. Sybil-Resistant DHT Routing. In Proceedings on the 10th European Symposium on Research in Computer Security, Milan, Italy, 12–14 September 2005.
32. Cheng, A.; Friedman, E. Sybilproof Reputation Mechanisms. In Proceedings of the ACM SIGCOMM workshop on Economics of peer-to-peer systems, Philadelphia, PA, USA, 22 August 2005.
33. Ji, W.; Yang, S.; Chen, B. A Group-based Trust Metric for P2P Networks: Protection against Sybil Attack and Collusion. In Proceedings of the International Conference on Computer Science and Software Engineering, Wuhan, China, 12–14 December 2008.
34. Ning, P.; Sun, K. How to misuse AODV: A case study of insider attacks against mobile ad-hoc routing protocols. *Ad Hoc Netw.* **2005**, *3*, 795–819.
35. Sanzgiri, K.; Levine, B.N.; Shields, C.; Dahill, B.; Belding-Royer, E.M. A Secure Routing Protocol for Ad Hoc Networks. In Proceedings of the 10th IEEE International Conference on Network Protocols, Paris, France, 12–15 November 2002.
36. Hu, Y.C.; Johnson, D.B.; Perrig, A. SEAD: Secure efficient distance vector routing for mobile wireless ad hoc networks. In Proceedings of the 4th IEEE Workshop on Mobile Computing Systems and Applications, Callicoon, NY, USA, 20–21 June 2002.
37. Papadimitratos, P.; Haas, Z.J. Secure Routing for Mobile Ad hoc Networks. In Proceedings of the SCS Communication Networks and Distributed Systems Modeling and Simulation Conference, San Antonio, TX, USA, 27–31 January 2002.
38. Zapata, M.G.; Asokan, N. Securing Ad Hoc Routing Protocols. In Proceedings of the 1st ACM workshop on Wireless Security, Atlanta, GA, USA, 28 September 2002.
39. Yi, S.; Naldurg, P.; Kravets, R. Security-Aware Ad hoc Routing for Wireless Networks. In Proceedings of the 2nd ACM International Symposium on Mobile Ad Hoc Networking & Computing, Long Beach, CA, USA, 4–5 October 2001.
40. Marti, S.; Giuli, T.J.; Lai, K.; Baker, M. Mitigating Routing Misbehavior in Mobile Ad Hoc Networks. In Proceedings of the 6th Annual International Conference on Mobile Computing and Networking, Boston, MA, USA, 6–11 August 2000.
41. Kurosawa, S.; Nakayama, H.; Kato, N.; Jamalipour, A.; Nemoto, Y. Detecting Blackhole Attack on AODV-based Mobile Ad Hoc Networks by Dynamic Learning Method. *Int. J. Netw. Secur.* **2007**, *5*, 338–346.

42. Buchegger, S.; Boudec, J.Y.L. Performance Analysis of the CONFIDANT Protocol (Cooperation Of Nodes: Fairness In Dynamic Adhoc NeTworks). In Proceedings of the 3rd ACM International Symposium on Mobile Ad Hoc Networking & Computing, Lausanne, Switzerland, 9–11 June 2002.
43. Bhalaji, N.; Shanmugam, A. Reliable Routing against Selective Packet Drop Attack in DSR based MANET. *J. Softw.* **2009**, *4*, 536–543.
44. Newsome, J.; Shi, E.; Song, D.; Perrig, A. The Sybil Attack in Sensor Networks: Analysis & Defenses. In Proceedings of the 3rd International Symposium on Information Processing in Sensor Networks, Berkeley, CA, USA, 26–27 April 2004.
45. Piro, C.; Shields, C.; Levine, B.N. Detecting the Sybil Attack in Mobile Ad hoc Networks. In Proceedings of the 2nd International Conference on Security and Privacy in Communication Networks, Baltimore, MD, USA, 28 August–1 September 2006.
46. Tangpong, A.; Kesidis, G.; Hsu, H.Y.; Hurson, A. Robust Sybil Detection for MANETs. In Proceedings of the Proceedings of 18th International Conference on Computer Communications and Networks, San Francisco, CA, USA, 2–6 August 2009.
47. Čapkun, S.; Hubaux, J.P.; Buttyan, L. Mobility Helps Peer-to-Peer Security. *IEEE Trans. Mob. Comput.* **2006**, *5*, 43–51.
48. Kutzner, K.; Wallenta, C.; Fuhrmann, T. Securing the Scalable Source Routing Protocol. In Proceedings of the World Telecommunications Congress, Budapest, Hungary, 1–3 May 2006.
49. Walkerdine, J.; Lock, S. Towards Secure Mobile P2P Systems. In Proceedings of the 2nd International Conference on Internet and Web Applications and Services, Le Mourne, Mauritius, 13–19 May 2007.