Distributed Trust Establishment in MANET's: Swarm Intelligence

Laurent Eschenauer, John S. Baras, Virgil Gligor
Center for Satellite and Hybrid Communication Networks
Department of Electrical and Computer Engineering & Institute for Systems Engineering
University of Maryland, College Park, MD 20742, USA

Abstract

We present some properties of trust establishment in mobile, ad-hoc networks and illustrate how they differ from those of trust establishment in the Internet. We present a framework for trust establishment in mobile ad-hoc networks and argue that peer-to-peer networks are especially suitable to solve the problems of generation, distribution, and discovery of trust evidence in mobile ad-hoc networks. We develop a new scheme based on swarm intelligence and demonstrate its advantages over the peer to peer scheme. We evaluate our approach through simulation with NS-2.

1 Introduction

We view the notion of "trust" among entities (e.g., domains, principals, components) engaged in various protocols as a set of relations established on the basis of a body of supporting assurance (trust) evidence and required by specified policies (e.g., by administrative procedures, business practice, law).

In traditional networks, most trust evidence is generated via potentially lengthy assurance processes, distributed off-line, and assumed to be valid on long terms and certain at the time when trust relations derived from it are exercised. Authentication and access-control trust relations established as a consequence of supporting trust evidence are often cached as certificates and as trust links (e.g., hierarchical or peer links) among the principals included in these relations or among their "home domains." Both certificates and trust relations are later used in authorizing client access to servers.

In contrast, few of these characteristics of trust relations and trust evidence are prevalent in *mobile ad-hoc networks (MANETs)*. Lack of a fixed networking infrastructure, high mobility of the nodes, limited-range and unreliability of wireless links are some of the characteristics of MANET environments that constrain the design of a trust establishment scheme. In particular, trust relations may have to be established using only on-line-available evidence, may be short-term and largely peer-to-peer, where the peers may not necessarily have a relevant "home domain" that can be placed into a recognizable trust hierarchy, and may be uncertain.

In this work we argue that for trust establishment in MANETs a substantial body of trust evidence needs to be (1) generated, stored, and protected across network nodes, (2) routed dynamically where most needed, and (3) evaluated "on the fly" to substantiate dynamically formed trust relations. In particular, the management of trust evidence should allow alternate paths of trust relations to be formed and discovered using limited backtracking though the ad-hoc network, and should balance between the reinforcement of evidence that leads to "high-certainty" trust paths and the ability to discover alternate paths.

Although we focus on authentication and access-control trust in this work, similar notions can be defined for "correctness" trust relations required by system design goals. System correctness is established by using layer decomposition and abstraction such that correctness of a lower layer can be used as evidence for the correctness-trust of a higher layer (i.e. Layer A "uses" layer B $\Leftrightarrow$ (Correctness of A $\Rightarrow$ Correctness of B)). In the rest of this introduction, we present the Mobile *Ad-Hoc* Network environment and some exemples of (1) the generation of evidence for correctness-trust establishment of a secure routing protocol, and (2) the generation of on-line evidence for trust establishment in sensor networks.

The absence of a routing infrastructure that would assure connectivity of both fixed and mobile nodes precludes supporting a stable, long-term, trust infrastructure, such as a hierarchy of trust relations among subsets of network nodes. It also constrains the trust establishment process to short, fast, on-line-only protocols using only subsets of the established trust relations, since not all nodes that established trust relations may be reachable.

In general, the Internet relies on a fixed trust infrastructure of certification-authority and directory servers for both fixed and mobile nodes (i.e., Mobile IPv6 nodes). These

servers must be available on-line and reachable by principals when needed; e.g., certification authority servers, when certificates are created and signed, and directory servers permanently.

In contrast, a fixed infrastructure of certification-authority and directory servers may not always be reachable in a MANET (viz. Section 2.3, scenarios 2 and 3). This is because MANETs cannot assure the connectivity required to these servers; e.g., both a mobile node and the foreign-domain nodes with which it communicates can be disconnected from the directory server storing the certificates defined in that node's home domain. Note that this is not the case for mobility in the Internet: Mobile IPv6 takes care of roaming by providing a "care of" address bound to the actual mobile address. This solution is not possible for MANETs since the home of a node and its "care of" address may be physically unreachable. Therefore, MANETs cannot rely exclusively on trust relations that are represented as certificates stored in directory hierarchies, since connectivity to the required servers may not be available when needed. MANETs must support *peer-to-peer relations* defined as the outcomes of any principal's evaluation of trust evidence from *any* principals in the network, and must store these trust relations in the nodes of the *ad-hoc* network.

In the Internet, trust relations are established for the long term and are stable. This is possible if security policies and assurances do not change very often and therefore do not need to be re-evaluated frequently.

n contrast, there is little long-term stability of evidence in MANETs. The security of a mobile node may depend of its location and cannot be a priori determined. For example, node capture by an adversary becomes possible and probable in some environments such as military battlefields. Trust relations involving a captured node need to be invalidated, and new trust evidence need to be collected and evaluated to maintain node connectivity in the *ad-hoc* network. Therefore, trust relations can be short-lived and the collection and evaluation of trust evidence becomes a recurrent and relatively frequent process. This process has to be fast to avoid crippling delays in the communication system; e.g., two mobile nodes may have a short time frame to communicate because of wireless range limitations, and trust establishment should not prevent these nodes from communicating securely by imposing a slow, lengthy process. To be fast, the trust establishment process may have to be executed entirely on-line since off-line collection and evaluation of evidence is impractical; e.g., visually verifying an identity document is not possible.

In the Internet, it is highly improbable that some trust relation remains unavailable for extended periods of time (e.g., a certificate verification on a trust path cannot performed for a day) due to connectivity failures. Network connectivity is guaranteed through redundancy of communication links, and routes and servers are replicated to guarantee availability. In general, it is fair to assume that the entire body of evidence necessary for trust establishment is available in the Internet when needed. In contrast, node connectivity is not guaranteed in MANETs and all established evidence cannot be assumed to be available for all nodes all the time. Trust establishment has to be performed with incomplete and hence uncertain trust evidence.

2 A Framework for Trust Establishment in MANETs

In this section, we present our framework for trust establishment in the MANET. We first give an overview of the scheme and its three components: generation, distribution, and evaluation of trust evidence. We then detail our evidence distribution scheme, based on peer-to-peer file-sharing systems. We also propose a swarm based scheme for evidence distribution that has the same properties as a p2p system without some of its drawbacks.

2.1 Generation of trust evidence

In our approach, any node can generate trust evidence about any other node. Evidence may be an identity, a public key, a location, an independent security assessment, or any other information required by the policy and the evaluation metric used to establish trust. Evidence is usually obtained off-line (e.g. visual identification, audio exchange [2], physical contact [32][33], etc.), but can also be obtained on-line. When a principal generates a piece of evidence, he signs it with its own private key, specify its lifetime and makes it available to other through the network. PGP is an instance of this framework, where evidence is only a public key.

A principal may revoke a piece of evidence it produced by generating a revocation certificate for that piece of evidence and making it available to others, at any time before the evidence expires. Moreover, a principal can revoke evidence generated by others by creating contradictory evidence and distributing it. Evidence that invalidates other extant evidence can be accumulated from multiple, independent, and diverses sources and will cause trust metrics to produce low confidence parameters.

It may seem dangerous to allow anyone to publish evidence within the *ad-hoc* network without control of any kind. For example, a malicious node may introduce and sign false evidence thereby casting doubt about the current trust relations of nodes and forcing them to try to verify the veracity of the (false) evidence. To protect against malicious nodes, whenever the possibility of invalidation of extant trust evidence (e.g., evidence revocation) arises, the

policy must require redundant, independent pieces of (revocation) evidence from diverse sources before starting the evaluation process. Alternatively, the evaluation metric of the policy may rate the evidence provided by certain nodes as being low-confidence information. In any case, the policy and its evaluation metric can also be designed to protect against false evidence.

2.2 Distribution of trust evidence

Every principal is required to sign the pieces of evidence it produces. A principal can distribute trust evidence within the network and can even get disconnected afterwards. A producer of trust evidence does not have to be reachable at the time its evidence is being evaluated. Evidence can be replicated across various nodes to guarantee availability. This problem of evidence availability is similar to those that appear in distributed data storage systems, where information is distributed across multiple nodes in a network, and a request for a piece of stored information is dynamically routed to the closest source.

However, trust evidence distribution is more complex than a simple "request routing" problem. A principal may need more than one answer per request, and hence *all* valid answers to a request should ideally be collected. For example, `REQUEST(Alice/location)` should return all pieces of evidence about the location of Alice. Typical distributed data storage systems do not return all valid requests; e.g. `REQUEST(my_song.mp3)` would return one file even if there are multiple versions of my_song each having different bit rates and length. Moreover a principal may simply not know what evidence to request, and hence wildcard requests have to be supported; e.g. `REQUEST(Alice/*)` should return all pieces of evidence about Alice available in the network.

2.3 Application of an evaluation metric to a body of evidence

In specifying a trust management policy, we distinguish between a *policy decision* and a *trust metric* for practical rather than fundamental reasons. A metric is used to assign a confidence value to pieces of evidence of the same nature. For instance, if we have three sources of evidence providing three different locations for Alice, how do we determine Alice's actual location and how confident are we of that determination? Different metrics may be used for different type of evidence (e.g. one may use a discrete level metric to characterize confidence in location, but a continuous metric to characterize confidence in a public key).

In contrast, a policy decision is a local procedure which, based on a set of evidence parameters and their required confidence value, outputs the outcome of the decision. In practice, policy decisions are locally enforced but may be based on trust metrics shared by other local policies. Similarly, the same policy decision may use different trust metrics (as in the case of UK3's metrics in Scenario 3 above) for different parameters. Different types of policy decisions have been proposed that apply a policy to a set of credentials and output a decision [4], [5].

Trust metrics to evaluate uncertain and incomplete sets of evidence has been an active field of research. Different "trust metrics" have been developed [37], [30], [22] and properties of these metrics have been studied [19]. However, the only practical trust metric developed and implemented has been the one of PGP [38]. Based on a very limited notion of uncertainty, this metric handles only the evaluation of trust in a chain of keys, with limited "levels of trust" (i.e. untrusted, marginal, full). There is a need to develop new trust metrics that apply to different types of evidence, not just chains of keys, are fine-grained in the sense that output wide set of uncertainty levels, and are flexible, in the sense that they can apply to incomplete sets of evidence.

2.4 Peer-to-peer file sharing for evidence distribution.

The problem of evidence distribution shares many characteristics of distributed data storage systems, and yet is different. It is interesting to examine current peer-to-peer, file-sharing systems to understand their characteristics and limitations regarding trust evidence distribution. Peer-to-peer networking has received a lot of attention recently, particularly from the services industry [24],[13], the open-source [8] and research communities [1], [34]. They evolved from very simple protocols, such as Napster (which uses a centralized index) and Gnutella (which uses request flooding) to more elaborate ones, such as Freenet (which guarantees request anonymity and uses hash-based request routing) [8] and Oceanstore (which routes requests using Plaxton trees)[20].

2.5 Overview of Freenet

Freenet [8] is a distributed storage system that supports the distribution of information while protecting the anonymity of both the generator and the requestor of a piece of information. It is a strictly peer-to-peer network, no centralised index is used, in place an efficient request routing protocol is used to find information in the network. All nodes contribute to Freenet by providing storage space, helping to route request in the network; however it is not possible for a node (or an outsider) to know what is stored in its local cache; therefore a node can't be held liable for its content and it is not possible to know which node to bring down to remove a document from the Freenet.

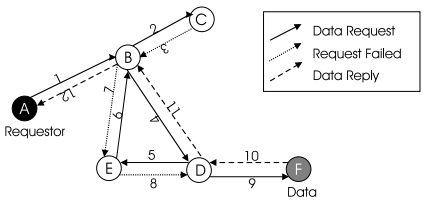

Figure 1. An example of a request routing in Freenet

The request routing in freenet is based on *hashed keyword*. To search for a document, a node hashes the requested document's name and use the hash as the search key. A request is routed towards the destination that is the more likely to have a document corresponding to that key in cache. To determine the next hop for a request, a node maintain a table mapping hash of succesfull requests with nodes; when a new request arrives, the node search the routing table for the entry which hash is the closest to the request hash and forward the message to the corresponding node. If the request is successful, it is answered using the reverse path and every node update its routing table by adding the request hash and the corresponding node in its table. Figure 1 shows an example of request routing in freenet. Note than when B receives the `data reply` for `hash1` it can either add an entry for the corresponding hash with D or F as the next hop, depending on implementation.

To complement the routing, a caching mechanism is implemented in freenet to increase availability of highly requested documents through the network. When a request is answered, the node on the reply path have the possibility to cache the document locally. This has the effect to bring documents towards the places where they are the most requested and therefore optimize futher requests. Different caching policies have been proposed for freenet, trying to determine which node should cache what and when. A new approach based on a *small world* analysis of freenet has been proposed by Zhang *et al.*[39].

2.6 Freenet for evidence distribution

We analyzed Freenet as a tool for evidence distribution because of the characteristics of its request routing architecture. In particular, in Freenet requests are routed in the network instead of flooding. Files are replicated by caching at every node and frequently requested files are highly replicated across the network while file that are rarely requested are slowly evicted from caches. Request routing in Freenet is adaptive and improves with time; combined with the caching policy it shows an interesting locality property: information converges where needed and is forgotten where not requested. This suits particularly well the locality property of trust establishment in the MANET (a node tends to establish trust with nearby neighbors). This optimized routing allows faster distribution and revocation of pieces of evidence.

However, the Freenet approach does not support wildcard requests and provides only one answer per request (due to the nature of its routing mechanism). Moreover, access to various sources of information evolves only by path reinforcement. As a consequence, some sources of information providing non-usable data are reinforced, and other sources are not discovered. The reinforcement strategy of Freenet does not preserve the diversity of information sources in the network. A new system has to be designed that shares the advantages of Freenet without exhibiting its drawbacks.

2.7 Swarm intelligence for trust evidence distribution.

Swarm intelligence [6] is a framework developed from the observation of ants' colonies. While a single ant is a very simple insect, groups of ants can cooperate and solve complex problems such as finding the shortest path to a food source or building complex structures. Ants do not communicate directly with each other; instead they induce cooperation by interacting with their environment (e.g., leaving a pheromone trail). When trying to find an optimum solution (e.g., shortest path to food source), cooperation leads to reinforcement of good solutions (positive feedback); more over, the natural decay of a pheromone trail enables regulation (negative feedback) that helps the discovery of new paths.

Numerous algorithms have been developed from these observations and applied to problems such as the traveling salesman, graph coloring, routing in networks [35][10]. Swarm intelligence is particularly suited for solving optimization problems in dynamically changing environments such as those of MANETs because of the balance between positive feedback that helps reinforce a good solution and the regulation process that enables discovery of new solutions appearing because of changes in the environment.

The problem of discovering proper sources of trust evidence in a MANET (and the problem of resource discovery in a network in general) is similar to the discovery of food supplies for an ant colony. It requires exploration of the environment with reinforcement of good solutions but also regulation that allows new sources to be discovered.

We now describe the conceptual ideas behind our ant-based scheme. The goal of this design is to achieve the same performances as the Freenet routing/caching while preserving diversity of evidence by discovering all sources in the network. This design is built following the experience of Subramanian *et al.* [35], and Di Cargo and Dorigo [10] in

their various routing protocol for dynamic networks.

We build our ant protocol directly above the link layer. Ant packets and requests are routed by the ant algorithm and don't depend on another routing protocol. We believe that if an ant-based routing protocol is used also for route discovery, it could be easily integrated with this protocol for resource (evidence) discovery.

Routing is still based on the hash of the request, so that the space of possible requests is known in advance. It also allows us to have similar anonimity properties to those of the Freenet system.

Ants exploring the network: Periodically, each host sends a "fake" request for a chosen hashed keyword. This hash may be randomly chosen in the hash space (simplest design) or chosen based on the previous requests by that host. If a host generates a lot of requests for evidence about Alice but none about Bob (two different hashed keywords) then the host will generate more ants towards the first hash than the second. The request is of the form ($hash_r$, source,TTL), where $hash_r$ is the requested hash, source the initiator of the request, and TTL is an upper limit on the number of hops that the request can traverse. This small message is the *ant* of our protocol.

The ant is routed in the network towards a host in possession of a document with a corresponding hash. At each hop the packet is routed via a probabilistic routing and the TTL is decremented. When the ant finds a document with corresponding hash a backward ant is generated and routed back to the source. If the TTL goes to zero before a document is found, the ant is destroyed. The backward ant is the one responsible for updating the routing tables.

Probabilistic ant routing: Unlike Freenet, which routes requests always to the host with the closest hash, our ant routing is probabilistic. Each host h maintains a routing table with entries of the form ($hash_k$, (y_1, p_1), ..., (y_n, p_n)) where $\forall i$, y_i is a one-hop wireless neighbor of h. When h receives a request for $hash_k$ it will forward the request to y_1 with probability p_1.

Update of routing tables by backward ants: A backward ant is generated when an ant finds a document matching the requested hash. The backward ant is the message ($hash_r$, source). This ant is routed back to the source on the reverse path and updates all routing tables on its way back.

When a host receives a backward ant from neighbor y_i, it updates all entries in its routing table. For all hash entries in the table, the probabilities (h_k, (y_1, p_1), ..., (y_n, p_n)) are updated as follows:

$$p_i = \frac{p_i + \Delta p}{1 + \Delta p}, p_j = \frac{p_j}{1 + \Delta p}, 1 \leq j \leq n, i \neq j$$

where $\Delta p = \frac{k}{f(d)}, k > 0$, d the distance between $hash_k$ and $hash_r$, and $f(d)$ is a non-decreasing function of d.

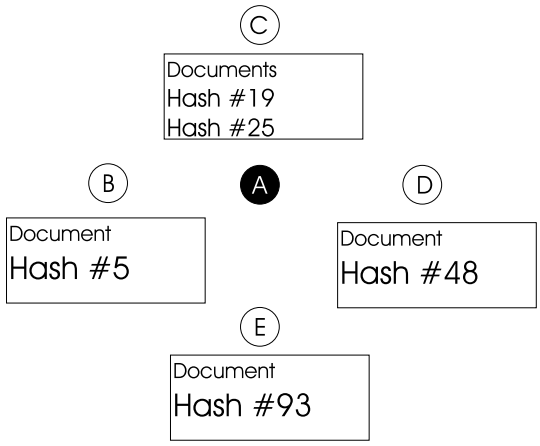

Figure 2. The topology used for example 3.3.3 Node A is in wireless range of B, C, D, E. The document stored and their respective hash is also showed

In the next section we present a simple example and show how this scheme converges in similar routing decisions than freenet while preserving knowledge about all sources of evidence.

2.8 An example

We describe a very simple example showing intuitively how the ant search works and why it produces results similar to Freenet, while preserving all sources of evidence. For this example, we choose *k=0.1* and $f(d) = e^{\frac{1}{2}d}$ and we assume a hash space of one hundred entries (while it should be on the order of 2^{32} in real operations as in Freenet).

Figure 2 shows the neighborhood in wireless range of node A. To forward a request, A must decide which of its neighbor is the most likely to answer it or properly forward it to find an answer. We assume that each node stores at least one document and show the corresponding hash on the figure.

Scenario 1. Node A initialise its routing table by assigning an equal probability for every output node, for every hash. A then starts the process of generating ants and eventually generates an ant for hash #5, this ant has one chance over four to be forwarded towards B. If this is the case, there is a match at B, and the backward ant updates A's routing table as shown on table 3.1. After enough ants are generated, all knowledge is found (hash #19 at C, hash #48 at D, and hash #93 at E) and the probabilistic routing table is shown in figure **??**. Note than there is no need of special bootstrapping of the system as this is the case for Freenet, but that such a bootstrapping (all neighbors broadcasting the

hash	B	C	D	E
0	0.25	0.25	0.25	0.25
...				
4	0.37	0.21	0.21	0.21
5	0.4	0.20	0.20	0.20
6	0.37	0.21	0.21	0.21
...				
99	0.25	0.25	0.25	0.25

Table 1. The probabilistic routing table of node A after receiving an ant from B in scenario 1.

hash of their first document) may accelerate this process.

To send a request (or insert a document), A selects the next hop with the highest probability for the hash of the request. This part of the routing is deterministic, only the routing of ants and wildcard requets are probalistic.

Scenario 2. We now show how our algorithm "rewards" nodes storing more documents than other nodes in the network. We assume that node C also has documents corresponding to hash #25 in its repository and it is found by an ant from A (after generating an ant for hash #25 and routing to C, with probability .31), A updates its routing table. In Freenet, this new entry would not affect at all the cluster of B (i.e. node B would still receive requests for hash #0 to #12 from A), but it can be easily seen that the cluster for B is now only covering #0 to #9.

Scenario 3. When node A needs to send a wildcard request or need more than one answer for a request it selectively floods the network based on the probabilistic table. For example, we assume that A needs all possible documents of hash #17 but no more than 50 (not to overload the network). It generates 50 requests and forward them using the probabilistic routing table. On the average A will send 13 requests to B, 18 o C, 10 to D and 9 to E (these requests can be grouped in a same packet with format ($hash_r$, source, nbr_requests, TTL)). The next hop proceeds the same way, splitting the remaining requests using its probabilistic routing table.

3 Conclusions and future work

The notion of trust establishment in mobile *ad-hoc* networks (MANETs) can differ from that in the (mobile) Internet in fundamental ways. Specifically, it has the trust establishment process has to be (1) peer-to-peer, (2) short, fast, and on-line-only, and (3) flexible enough to allow uncertain and incomplete trust evidence.

We presentend a framework for trust establishment that supports the requirements for MANETs and relies on peer-to-peer file-sharing for evidence distribution through the network. The problem of evidence distribution for trust establishment is somewhat different than the usual file sharing problem in peer-to-peer networks. For this reason, we proposed to use a "swarm intelligence" approach for the to design of trust evidence distribution instead of simply relying on an ordinary peer-to-peer, file-sharing system. In future work, we plan to evaluate the performance of "swarm"-based algorithms for trust evidence distribution and revocation in a MANET environment.

Finally, we also argued that the design of metrics for the evaluation of trust evidence is a crucial aspect of trust establishment in MANETs. In future work, we plan to develop a trust management scheme integrating the confidence valuation of trust evidence with real-time, policy-compliance checking.

References

[1] O. Babaoglu, H. Meling, and A. Montresor, "Anthill: A Framework for the Development of Agent-Based Peer-to-Peer System," Technical Report UBLCS-2001-09, University of Bologna, Italy.

[2] D. Balfanz, D.K. Smetters, P. Stewart, and H. Chi Wong, "Talking To Strangers: Authentication in Ad-Hoc Wireless Networks," in Proc. of the ISOC 2002 Network and Distributed Systems Security Symposium, February 2002.

[3] T. Beth, M. Borcherding, and B. Klein, "Valuation of trust in open networks," in Proc. of ESORICS 94. Brighton, UK, November 1994.

[4] M. Blaze, J. Feigenbaum, and J. Lacy, "Decentralized trust management", in Proc. of the 1996 IEEE Symposium on Security and Privacy, pages 164–173, May 1996.

[5] Matt Blaze, Joan Feigenbaum, and Angelos D. Keromytis, "KeyNote: Trust management for publickey infrastructures", in Proc. Cambridge 1998 Security Protocols International Workshop, pages 59–63, 1998.

[6] E. Bonabeau, M. Dorigo and G. Theraulaz, *Swarm Intelligence: From Natural to Artificial Systems*, Santa Fe Institute on the Sciences of Complexity, Oxford University Press, July 1999.

[7] D. W. Carman, P. S. Kruus and B. J. Matt Constraints and Approaches for Distributed Sensor Network Security, dated September 1, 2000. NAI Labs Technical Report #00-010

[8] I. Clarke, O. Sandberg, B. Wiley, and T. W. Hong, "Freenet: A Distributed Anonymous Information Storage and Retrieval System," in Proc. of the International Computer Science Institute (ICSI) Workshop on Design Issues in Anonymity and Unobservability, Berkeley, CA, 2000.

[9] *Common Criteria for Information Technology Security Evaluation – Part 3: Security Assurance Requirements*, version 2.0, CCIB-98-028, National Institute of Standards and Technology, May 1998. *http://niap.nist.gov*

[10] G. Di Caro and M. Dorigo, "AntNet: Distributed Stigmergetic Control for Communications Networks," Journal of Artificial Intelligence Research, 9:317–365, 1998.

[11] L. Eschenauer and V. Gligor, "A Key Management Scheme for Distributed Sensor Networks", to appear in Proc. of the 9th ACM Conference on Computer and Communications Security, November 17-21, 2002, Washington, DC, U.S.A

[12] V. D. Gligor, S.-W. Luan, and J. N. Pato, "On inter-realm authentication in large distributed systems," in Proc. of the 1992 IEEE Symposium on Research in Security and Privacy, May 1992.

[13] GNUTELLA, http://www.gnutellanews.com/

[14] A. Herzberg, Y. Mass, J. Mihaeli, D. Naor, and Y. Ravid, "Access Control Meets Public Key Infrastructure, Or: Assigning Roles to Strangers," in Proc. of the 2000 IEEE Symposium on Security and Privacy, 14-17 May 2000, Berkeley, California, USA, pages 2-14

[15] Y.-C. Hu, A. Perrig, and D. B. Johnson, "Ariadne: A Secure On-Demand Routing Protocol for Ad Hoc Networks", Technical Report TR01-383, Department of Computer Science, Rice University, December 2001.

[16] Y-C. Hu, D. B. Johnson, and A. Perrig, "SEAD: Secure Efficient Distance Vector Routing for Mobile Wireless Ad Hoc Networks", Proceedings of the 4th IEEE Workshop on Mobile Computing Systems & Applications (WMCSA 2002), IEEE, Calicoon, NY, June 2002 (to appear).

[17] J.-P. Hubaux, L. Buttyan and S. Capkun, "The Quest for Security in Mobile Ad Hoc Networks," in Proc. of the ACM Symposium on Mobile Ad Hoc Networking and Computing (MobiHOC 2001).

[18] D. B. Johnson, D. A. Maltz, and J. Broch, "DSR: The Dynamic Source Routing Protocol for Multi-Hop Wireless Ad Hoc Networks" in Ad Hoc Networking, edited by Charles E. Perkins, Chapter 5, pp. 139-172, Addison-Wesley, 2001.

[19] R. Kohlas and U. Maurer, "Confidence Valuation in a Public-key Infrastructure Based on Uncertain Evidence," in Proc. of Public Key Cryptography 2000, Lecture Notes in Computer Science, vol. 1751, pp. 93-112, Jan 2000.

[20] J. Kubiatowicz, D. Bindel, Y. Chen, S. Czerwinski, P. Eaton, D. Geels, R. Gummadi, S. Rhea, H. Weatherspoon, W. Weimer, C. Wells, and B. Zhao, "OceanStore: An Architecture for Global-Scale Persistent Storage," in Proc. of the Ninth international Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS 2000), November 2000.

[21] B. W. Lampson, M. Abadi, M. Burrows, and Edward Wobber, "Authentication in distributed systems: Theory and practice,", ACM Transactions on Computer Systems, 10(4):265–310, November 1992.

[22] U. Maurer, "Modelling a Public-Key Infrastructure." in Proc. ESORICS '96 (4th European Symposium on Research in Computer Security), Rome, LNCS 1146, Springer-Verlag, Berlin 1996, 325–350.

[23] G. Montenegro and C. Castelluccia, "Statistically Unique and Cryptographically Verifiable (SUCV) Identifiers and Addresses", Proceedings of the 2002 Network and Distributed System Security conference (NDSS02), San Diego, February 2002.

[24] NAPSTER, http://www.napster.com

[25] NS-2, http://www.isi.edu/nsnam/ns

[26] G. O'Shea and M. Roe, "Child-proof Authentication for MIPv6 (CAM)," ACM Computer Communication Review, April 2001.

[27] P. Papadimitratos and Z. Haas, "Secure Routing for Mobile Ad-Hoc Networks", Proceedings of the Communication Networks and Distributed Systems Modeling and Simulation Conference (CNDS2002), San Diego, CA, January 2002.

[28] C. Perkins and P. Bhagwat, "Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers", Proceedings of the ACM SIGCOMM, October 1994.

[29] C. E. Perkins and E. M. Royer, "Ad hoc On-Demand Distance Vector Routing", Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications, New Orleans, LA, February 1999, pp. 90-100.

[30] M. K. Reiter and S. G. Stubblebine, "Toward acceptable metrics of authentication," in Proc. of the IEEE

Conference on Security and Privacy, Oakland, CA, 1997.

[31] M. K. Reiter and S. G. Stubblebine, “Path independence for authentication in large-scale systems,” in Proc. of the 4th ACM Conference on Computer and Communications Security, April 1997.

[32] F. Stajano and R. Anderson, “The Resurrecting Duckling: Security Issues for Ad-hoc Wireless Networks,” in Proc. of the 8th International Workshop on Security Protocols, Lecture Notes in Computer Science, Springler-Verlag, Berlin, Germany, 1999.

[33] F. Stajano, “The resurrecting duckling – What next?,” in Proc. of the 8th International Workshop on Security Protocols, Lecture Notes in Computer Science, Springler-Verlag, Berlin, Germany, April 2000.

[34] I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, and H. Balakrishnan, “Chord: A scalable peer-to-peer lookup service for Internet applications,” in Proc. of the 2001 ACM SIGCOMM Conference, San Diego, CA, 2001, pages 149–160.

[35] D. Subramanian, P. Druschel, and J. Chen, “Ants and reinforcement learning: A case study in routing dynamic networks,” in Proc. of the 15th International Joint Conference on Artificial Intelligence (IJCAI), 1997.

[36] E. Wobber, M. Abadi, M. Burrows, and B. Lampson, “Authentication in the Taos operating system,” ACM Transactions on Computer Systems, 12(1):3–32, Feb. 1994.

[37] R. Yahalom, B. Klein, and T. Beth. “Trust relationships in secure systems—A distributed authentication perspective,” in Proc. of the 1993 IEEE Symposium on Research in Security and Privacy, pages 150–164, May 1993.

[38] P. R. Zimmermann, *The Official PGP User’s Guide*, MIT Press, 1995. (http://www-mitpress.mit.edu/mitp/recent-books/comp/pgp-user.html)

[39] H. Zhang, A. Goel, and R. Govindan, “ Using the Small-World Model to Improve Freenet Performance,” in Proc. of the 2002 IEEE INFOCOM, New-York, NY, 2002.

[40] L. Zhou and Z. Haas, “Securing ad hoc networks,” IEEE Network, 13(6):24–30, November/December 1999.