DISTRIBUTED TRUST ESTABLISHMENT IN MANETS: SWARM INTELLIGENCE(*)

Laurent Eschenauer, John S. Baras, Virgil D. Gligor
Department of Electrical and Computer Engineering
and the Institute for System Research
University of Maryland College Park
College Park, MD 20742
Email: {laurent, baras, gligor}@eng.umd.edu

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.

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." Certificates and trust relations are 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 have a relevant "home domain" that can be placed into a recognizable trust hierarchy, and may be uncertain. 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 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.

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. 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. Therefore, trust relations can be short-lived and the collection and evaluation of trust evidence becomes a recurrent and relatively frequent

(*) Prepared through collaborative participation in the Collaborative Technology Alliance for Communications & Networks sponsored by the U.S. Army Research Laboratory under Cooperative Agreement DAAD19-01-2-0011. Also partially supported by the U.S. Army Research Office under grant No DAAD19-01-1-0494.

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.

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.

TRUST ESTABLISHMENT IN MANETS

In this section, we present our framework for trust establishment in MANETs.

A. 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 [9] [10], etc.), but can also be obtained on-line. When a principal generates a piece of evidence, he signs it with his own private key, specifies its lifetime and makes it available to others 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 diverse 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. The policy and its evaluation metric can also be designed to protect against false evidence.

B. 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 problems 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.

C. 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 [8], [6], the open-source [4] and research communities [1], [11]. They evolved from very simple protocols, such as Napster (uses a centralized index) and Gnutella (uses request flooding) to more elaborate ones, such as Freenet (guarantees request anonymity and uses hash-based request routing) [4] and Oceanstore (routes requests using Plaxton trees) [7].

D. Overview of Freenet

Freenet [4] is a distributed storage system that supports the distribution of information while protecting the anonymity of both the generator and the requester of a piece of information. It is a strictly peer-to-peer network, no centralized index is used. Instead 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.

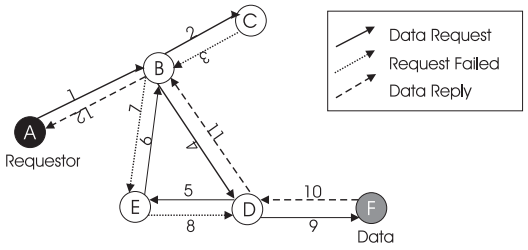

Figure 1. An example of a request routing in Freenet

The request routing in Freenet is based on *hashed keyword*. To search for a document, a node hashes the requested document's name and uses 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 maintains a table mapping hash of successfull requests with nodes. When a new request arrives, the node searches the routing table for the entry whose hash is the closest to the request hash and forwards the message to the corresponding node. If the request is successful, it is answered using the reverse path and every node updates 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 nodes on the reply path have the possibility to cache the document locally. This has the effect of bringing documents towards the places where they are the most requested and therefore optimize further requests.

E. 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 files 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 MANETs (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 its drawbacks.

SWARM INTELLIGENCE FOR TRUST EVIDENCE DISTRIBUTION

Swarm intelligence [3] is a framework developed from observations of ant 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 [12] [5]. 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 an existing good solution and the regulation process that enables discovery of new good solutions, needed due to 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 sources 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 Freenet routing/caching while preserving

diversity of evidence by discovering all sources in the network. We build our ant protocol directly above the link layer. Ant packets and requests are routed by the ant algorithm and do not 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 anonymity 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. Backward ants are responsible for updating 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.

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

AN EXAMPLE

We describe a very simple example showing intuitively how 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).

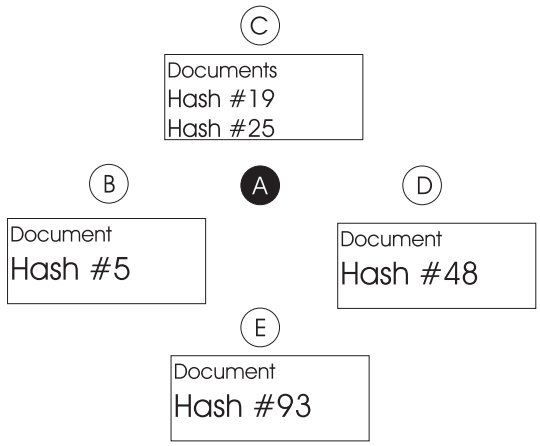

Figure 2. The topology used in the example. Node A is in wireless range of B, C, D, E. Documents stored and their hashes are also shown

Figure 2 shows the neighborhood in wireless range of node A. To forward a request, A must decide which of his neighbors 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 initializes 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 in 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 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 Table 1. Note that there is no need of special bootstrapping of the system as it was needed for Freenet. Such bootstrapping (all neighbors broadcasting the 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 requests are probabilistic.

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.

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). He generates 50 requests and forwards 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 the 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.

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, 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 presented 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 a "swarm intelligence" approach for the design of trust evidence distribution schemes, instead of simply relying on an ordinary peer-to-peer, file-sharing system. In future work, we plan to evaluate the performance of "swarm intelligence"-based algorithms for trust evidence distribution and revocation in a MANET environment.

Finally, 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] 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.

[4] 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.

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

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

[7] 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.

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

[9] 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.

[10] 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.

[11] 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.

[12] 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.

The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory or the U.S. Government.