MASTER'S THESIS

On Trust Establishment in Mobile Ad-Hoc Networks

by Laurent Eschenauer
Advisor: Virgil D. Gligor

CSHCN MS 2002-4
(ISR MS 2002-10)

ABSTRACT

Title of Thesis:	ON TRUST ESTABLISHMENT IN MOBILE *AD-HOC* NETWORKS
Degree candidate:	Laurent Eschenauer
Degree and year:	Master of Science, 2002
Thesis directed by:	Professor Virgil D. Gligor Department of Electrical and Computer Engineering

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 motivate these differences by providing an example of ad-hoc network use in battlefield scenarios, yet equally practical examples can be found in non-military environments. 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 evaluate our approach through simulation with NS-2.

ON TRUST ESTABLISHMENT IN MOBILE *AD-HOC* NETWORKS

by

Laurent Eschenauer

Thesis submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Master of Science
2002

Advisory Committee:

Professor Virgil D. Gligor, Chairman/Advisor
Professor William Arbaugh
Professor John Baras

DEDICATION

To my parents Renée and Erny,
my sister Cécile,
and to Bénédicte
for their invaluable love and support

ACKNOWLEDGEMENTS

I am grateful to my advisor Dr. Virgil Gligor for his advice, support and ecouragement. I would also like to thank Dr. William Arbaugh and Dr. John S. Baras for agreeing to serve in my committee and for providing feedback.

Also I would like to thank my office-mates Bob, Himanshu, Radostina, Rakesh and Emilian for creating a joyful and refreshing work environment.

The research reported in this thesis was 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.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 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

Figure 1.1: A soldier polls a sensor using its PDA through the mobile ad-hoc network

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.

1.1 Mobile *Ad-Hoc* networks

Ad-hoc networking refers to the spontaneous formation of a network of nodes without the help of any infrastructure, usually through wireless communication channels. Figure 1.1 is an example of MANET: various type of units (infantry, artillery, satellites, sensors) with different computation and communication capabilities. In *ad-hoc* networks, a basic routing infrastructure emerges through the collaboration of every node with its neighbors to forward packets towards chosen destinations. This basic infrastructure is highly dynamic not just because of node mobility but also because of lack of guaranteed node connectivity. In *ad-hoc* networks, lack of guaranteed connectivity is caused by the limited-range, potentially unreliable, wireless communication. The absence of a routing infrastructure that would assure connectivity of both fixed and mobile nodes precludes using the traditional internet protocols for routing, name resolution, trust establishment, etc.

1.1.1 Secure routing in the MANETs

Early protocols that performed routing in MANETs [19][29][30] assumed that all nodes were trusted; i.e., none of the nodes deliberately disrupted the routing protocol. More recently, several protocols were proposed to secure the routing layer from nodes that act maliciously. These protocols integrate security features within traditional routing protocols, such as DSR, AODV, DSDV, and aim to protect against message modification, fabrication or address spoofing through cryptographic means [16][17][28]. However, all these protocols assume that secure associations between the nodes of the network exist or can be established on-line. This assumption is used as evidence to support the correctness-trust establishment of the routing layer (e.g. proof of correctness of SRP by Papadimitratos and Haas [28]).

Typically, these associations consist of either symmetric keys shared between any two nodes distributed with the help of a trusted key distribution center (KDC), or public-key certificates associated with individual nodes and signed by a trusted certification authority (CA). Security associations and trust relations among nodes forms the basis for building the security features of the routing layer; e.g., message authentication, replay detection.

The assumption of pre-established secure associations may be practical in environments where such associations can be established off-line [33]. However, this assumption is less suitable for secure routing in large MANETs where secure associations have to be setup on-demand and on-line. Traditional Internet protocols relying on centralised servers (KDC, CA) cannot be used here not only because of the lack of guaranteed connectivity but also because there is cyclic dependency airsing between security services (e.g., certificate distribution, shared key generation, distributed trust establishment) and routing services since security services require routing layer security themselves. Because of this cyclic dependency the correctness of the components establishing the secure association depends on the correctness of the routing layer. It is therefore impossible to generate the evidence necessary to establish a trusted routing layer.

In other work with Bobba, Gligor, and Arbaugh [6] we proposed a solution for bootstrapping the security associations for secure routing without assuming any trusted authorities or distributed trust-establishment services. We proposed to rely on the use of statistically unique and cryptographically verifiable (SUCV) identifiers [24], and public-secret key pairs generated by the nodes themselves, in much the same way SUCVs are used in MobileIPv6 (MIPv6) to solve the address "ownership" problem [24][27] and to counter the "bidding down" attack [24] in return routability. The correctness of SUCV does not depend on any other component in the system and can be used as evidence to bootstrap the trust establishment of the routing layer.

1.1.2 Distributed sensor networks

Distributed Sensor Networks (DSNs) are a particular kind of MANETs characterised by a large size (e.g., ten thousand as opposed to tens or hundreds of nodes) and higly limited computation and communication capabilities. They present the same challenges that any other MANET (abscence of infrastructure, mobility, lack of guaranteed connectivity) but the computation constraint makes the design of solutions even harder.

As for any other MANET, there is a need for secure communication between nodes of a sensor networks and therefore a need to establish trust between nodes. However the extreme power, computational, and communication limitations of sensor nodes and the network scale preclude the of use the traditional cryptographic tools to generate trust evidence and establish trust. For example, public key cryptosystems and random-number generators cannot be used since they are computationally intensive and consume a significant amount of power [8]. Use of low-power, symmetric-key ciphers and modes of encryption becomes the only viable means of protecting communication against monitoring by hostile adversaries.

Traditional Internet style key exchange and key distribution protocols based on infrastructures using trusted third parties are also ruled out by sensor-node processing limitations, unknown network topology, intermittent sensor-node operation, network scale and dynamics. To date, the only options for the distribution of keys to sensor nodes of DSN whose physical topology is unknown prior to deployment would have to rely exclusively on key pre-distribution. Keys would have to be installed in sensor nodes to accommodate full secure connectivity between nodes. However, traditional key pre-distribution offers two inadequate solutions: either a single mission key or a set of separate n-1 keys, each being pair-wise privately shared between every two nodes, must be installed in every sensor node.

In other work with V.D. Gligor [12] we propose a key pre-distribution scheme that requires memory storage for only few tens of keys, and yet has similar security and superior operational properties to those of the pair-wise private, key-sharing scheme. It relies on probabilistic key-sharing among the nodes of a random graph and uses a simple secure shared-key discovery protocol for key distribution, revocation and node re-keying. We distribute a ring of keys to each sensor node, each key ring consisting of randomly chosen k keys from a very large pool of P keys, which is generated off-line, prior to DSN deployment. This secure distribution of key-rings form the basis of the evidence used in the trust establishment during operations. Because of the random choice of keys on key rings, a shared key may not exist between some pairs of nodes precluding them to establish trust. Although two nodes may or may not share a key, if a trust path of nodes sharing pair-wise private keys exists between the two nodes at network initialization, the two nodes can use that trusted path to exchange a key that will establish a direct trust link. In this case the nodes use already established trust relations with other nodes as evidence to establish a new trust relation.

We use random graph analysis and simulation to show that what really matters in key pre-distribution is the shared-key connectivity of the resulting secure network. Therefore, the full shared-key connectivity offered by pair-wise, private key sharing between every two nodes becomes unnecessary. For example, we show that to establish shared-key connectivity in a 10,000-node network, a key ring of only 250 keys have to be pre-distributed to every sensor node where the keys were drawn out of a pool of 100,000 keys. We also show that the security characteristics of probabilistic key distribution and revocation based on random graphs are suitable for solving the key management problem of DSNs.

1.2 Organization

This work is organized in five chapters. The first chapter is this introduction, defining the new environment of the MANET and presenting a set of specific problems related to authentication, access control, and correctness trust establishment in the MANET. The rest of this thesis focuses on the problem of authentication-trust establishment and evidence distribution.

The second chapter introduces trust establishment. Basic notions are explained, prior and related work is presented, and trust establishment in the MANET is discussed and compared to the traditional networks. In the third chapter our approach to (trust) evidence distribution is explained. We present different schemes based on peer-to-peer file-sharing and swarm intelligence. The fourth chapter covers the evaluation of our scheme through an implementation in *NS-2* and simulations. The final chapter concludes this work and present possible future work.

Chapter 2

Trust Establishment

In this chapter, we review some of the basic notions of trust establishment and explore how these notions differ in the MANET environment from those in the Internet environmnet. We also derive a set of requirements for trust establishment in MANETs. Much of the theory underlying the presentation of basic notions can be found in Maurer [23], Kohlas and Maurer [20], Lampson and Abadi [22], and Gligor[13]. We focus exclusively on some empirical properties of evidence for trust establishment that help differentiate the traditional Internet notions from those of MANETs.

2.1 Basic Notions of Trust Establishment

We view the process of trust establishment as the application of an evaluation metric to a body of trust evidence. The outcome of the trust establishment process is a trust relation. The evidence may be obtained on- or off-line and may include already established trust relations. An established trust relation constitutes evidence that can be used in other trust establishment processes, and can be composed with other relations to form more abstract or more general trust relations. The composition of trust relations usually requires the composition of evidence and of evidence evaluations.

2.1.1 An Example of Authentication-Trust Establishment

Consider the trust relation “A accepts B’s authentication of X”, which is established between principals A, B, and X. This relation is established as the composition of two basic relations resulting from two separate trust-establishment processes; i.e., “certification authority B accepts X’s authentication evidence,” and “certification authority A accepts B’s authentication of any principal registered by B”. The first relation may be established by principal B’s off-line evaluation of a body of trust evidence presented by principal X. For example, B may require several pieces of evidence attesting to X’s identity. Specifically, B may require two pieces of authentication evidence from the following set: driver license, passport, employment identity card, documentation indicating current property ownership or credit-line activity. Once the trust relation is established, it is cached as (1) a certificate signed by B associating X's public key with X, and (2) a relation stored in B's "trust database" registering principal X with B. The domain of certification authority B becomes X's "home domain."

The second relation, namely "certification authority A accepts B's authentication of any principal registered by B," may be established by principal A's *off-line* evaluation of a body of trust evidence presented by principal B indicating that:

- certification authority B's authentication of the principals registered with it (e.g., X) is done using "acceptable" mechanisms and policies; and

- certification authority B's registration database, which includes principal X's registration, is protected using "acceptable" mechanisms and policies;

- certification authority B's server is managed using "acceptable" administrative, physical, and personnel policies;

- certification authority B does not have skills and interests that diverge from those of A.

Evidence regarding the "acceptability" of various mechanisms and policies is collected off-line, using potentially lengthy assurance procedures, such as those prescribed by the Common Criteria's assurance evaluation levels [10]. Certification authority A uses an evaluation metric to determine whether B's authentication mechanisms and policies are (at least) as good as his own, and the evidence used by the metric is *stable* and *long-term.* Evidence is stable if the authentication mechanisms and policies used by B do not change, either intentionally or accidentally, unbeknownst to A. Evidence is long-term, if it lasts at least as long as the process of gathering and evaluating assurance evidence, which can be of the order of weeks or months. After the trust relation "certification authority A accepts B's authentication of any principal registered by B" is established by A, it is cached (1) as a certificate associating B's public key with B that is signed by A, and (2) as a relation stored in A's "trust database" registering principal B with A. The domain of certification authority A becomes B's "home domain."

Although we focus on authentication in this example, similar notions can be defined for trust establishment in the access control arera.

2.1.2 Transitivity of Trust Establishment

Trust relation "certification authority A accepts B's authentication of any principal registered by B" is clearly *reflexive* since A accepts its own authentication of principals it registers. However, should it be *transitive*? That is, should the trust establishment process be transitive? For example, if "A accepts B's authentication of any principal registered by B" and "B accepts Y's authentication of principal Z registered by Y," does it mean that "A accepts Y's authentication of principal Z registered by Y"? And if so, does this hold for any principals Y and Z?

Before accepting that transitivity should hold, A uses his "evaluation metric" to determine two properties of evidence. First, A determines that B's evaluation of Y's body of evidence is the same as (or stronger than) A's evaluation of B's body of evidence (viz., example 2.1.1). Second, A determines that B's trust relation with Y is (at least) as stable and long-term as his A's own with B. If these two properties of evidence hold for all Y's and Z's, then the more general trust relation "A accepts Y's authentication of any principal" should also hold. In practice, this general trust relation would hold for all Y's whose home domains are sub-domains of B's home domain. This is the case because B would control the adequacy, stability, and duration of Y's authentication mechanisms and policies, and hence could provide the evidence that would satisfy A's evaluation metric. However, evidence regarding Y's authentication mechanisms and policies may not pass A's evaluation metric, and A would not accept Y's authentication of any principal. For example, the evidence used in establishing B's trust relation with Y may be short-lived or unstable. In this case, Y could change its authentication policies, thereby invalidating evaluated evidence, unbeknownst to A and B. A would want to be protected from such events by denying transitivity regardless of whether B accepts Y's authentication of Z.

The principal characteristics of evidence used to establish transitive trust in the example given above are "uniformity" and "availability." Uniformity means that all evidence used to establish transitive trust satisfied the same, global, "metrics" of adequacy, stability, and long-term endurance. Availability means that all evidence could be evaluated either on-line or off-line at any time by a principal wishing to establish a trust relation.

2.1.3 Uncertainty in Trust Establishment

Transitive trust formed the basis for the definition of simple trust hierarchies, possibly interconnected by "peer" links. All early system designs supporting such hierarchies assumed either implicitly [22] or explicitly [13] that evidence for recommending trust from principal to principal was "uniform" and "available." In contrast, starting with Yahalom *et al.* [38], it was realized that, in general, trust evidence need not be uniform and hence could be uncertain. Pretty Good Privacy (PGP) [39] provides the first practical example where some "uncertainty" is allowed in authentication, although PGP does not support transitive trust. Later work by Kohlas and Maurer [20] formalizes the notion of evidence uncertainty and provides precise and fairly general principles for evaluating trust evidence.

2.1.4 Guaranteed Connectivity to Trust-Infrastructure Servers

To be scalable, Public Key Infrastructures (PKIs) establish trust among certification authorities rather than among individual principals. Transitive trust relations among certification authorities allows us to establish authentication trust among principals registered by different certification authorities, since it allows the traversal of certification authorities separating pairs of principals; i.e., the traversal of trust paths. Traversal of trust paths does not require that certification authorities be on-line permanently. Certification authorities store certificates in directories associated with "home domains" whenever trust relations are established, and hence directory hierarchies mirror trust hierarchies. Therefore, directory servers must be available and on-line permanently to enable trust path traversals by any principal at any time, whereas certification authority servers need be on-line only when trust relations are established and certificates are signed and stored in directories. Nevertheless, principals establishing trust relations or traversing directory hierarchies to establish, or verify the validity of, trust paths need guaranteed communication connectivity to certification authority and directory servers.

2.2 Why is the Mobile *Ad-Hoc* Network different?

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.

2.2.1 Trust Establishment without a Trust Infrastructure

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.

2.2.2 Short-lived, Fast, and On-line-only Trust Establishment

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.

In 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.

2.2.3 Trust Establishment with Incomplete Evidence

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.

2.2.4 Summary of the requirements

In summary, trust establishment in MANETs requires protocols that are:

- peer-to-peer, independent of a pre-established trust infrastructure (i.e., certification authority and directory servers);
- short, fast, and on-line; and
- flexible and support uncertain and incomplete trust evidence.

2.3 An Example with Three Scenarios

We present an example to intuitively show the differences between the internet and the MANET environment in respect to trust establishment. The three related scenarios take place in a battelfield environment, but we could have come with similar examples in the civil world.

2.3.1 Scenario 1

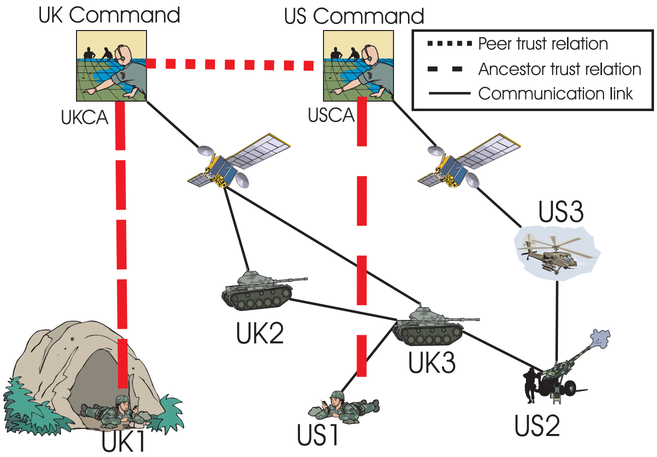

Figure 2.1: A battlefield scenario. UK1 is lost and can only communicate with US1

In Figure 2.1 we illustrate a battlefield environment in which units of coalition of United States (US) and United Kingdom (UK) forces perform separate operations. To support these operations, various communication systems are involved, ranging from short-range wireless (e.g., for infantry), to long-range directional wireless links (e.g., used between artillery pieces), and to satellite communication (e.g., connecting the battlefield with the US and UK operation commands).

In this scenario, assume that a British unit (UK1) is lost and takes refuge in a nearby cave. UK1 needs to call for backup, but the only unit in communication range is an American unit (US1) taking part in a different operation than that of UK1. The British unit, UK1, has to authenticate itself to US1 to get access to the *ad-hoc* US network and call the UK operations command for help. UK1 requests access to the *ad-hoc* US network and presents an identity certificate signed by UKCA, the British certification authority. The US network access policy requires that any accessor presents a valid identity certificate from a US-recognized and trusted authority. Node US1 needs to decide whether the node claiming to be UK1 should be allowed access to the *ad-hoc* US network. To decide wether UK1's certificate is valid, US1 contacts the directory server at US operations command and obtains a UKCA certificate signed by USCA, the US certification authority. US1 verifies and accepts USCA's signature on the UKCA's certificate, then accepts UKCA's signature on UK1's certificate, thereby exercising the transitive trust relations established between the US and UK operations commands and their respective units. Node US1 grants access to the *ad-hoc* US network to UK1. Note that the established trust infrastructure of the Internet helps solve UK1's problem, since all necessary trust relations (i.e., evaluated evidence) are available on-line.

2.3.2 Scenario 2

Assume that, due to inclement weather conditions, satellite links are unavailable. When US1 receives UK1's request and certificate signed by UKCA, it can't contact its operations command center to retrieve UKCA's certificate from a directory server, and therefore it cannot verify the signature on UK1's certificate. However, suppose that a couple hours ago while in a different operation, a US helicopter unit, US3, visually identified the lost British unit, UK1. US3 could have *proactively* generated a certificate for UK1 and made it available in the *ad-hoc* US network. Alternately, US3 could generate and sign a certificate for UK1 now. This piece of evidence is the only one that can be helpful in this scenario; however there is currently no scheme to specify how and when it should be generated, how it can be distributed to others in the network, how it is evaluated by US1 to make its final decision and finally how it can be revoked by US3 if needed. In chapter 3 we present our approach on how to solve these issues.

2.3.3 Scenario 3

Figure 2.3 illustrates a United Nations humanitarian convoy (UN1) that is approaching and preparing to cross a bridge separating two battlefield "zones". Be-

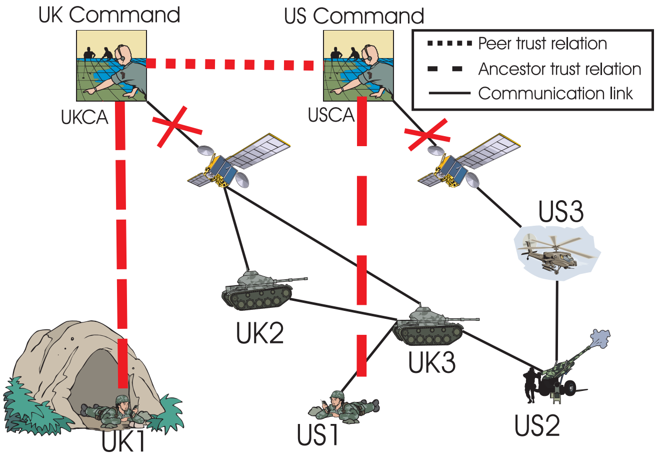

Figure 2.2: A battlefield scenario. UK1 is lost and can only communicate with US1. The satellite links are down due to inclement weather

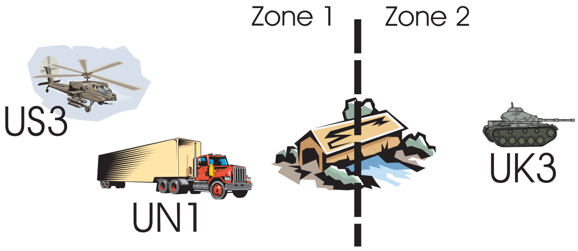

Figure 2.3: A battlefield scenario

fore crossing the bridge to enter the new zone, UN1 must request a "zone report" from nearby military units to verify that the zone is safe. UN1 sends a request for a zone report and attaches its credentials (Table 2.1.b) as authentication evidence to the request. A British unit, UK3, receives the request and is in a position to issue a zone report. However, to issue the zone report, UK3 needs to apply its evaluation metric (Table 2.1.d and 2.1.e) to the presented evidence (and the evidence already in its possession by other means) and to verify that it satisfies the policy it must enforce for providing zone reports (Table 2.1.a). However, UK3 has a limited set of already established trust relations (Table 2.1.c) and it is not hard to see that some evidence provided by UN1 (1) is useful but cannot be verified (i.e., certificates signed by USCA and US3 cannot be verified by UK3 since it does not have a direct trust relation to USCA and US3 and the satellite links are unavailable); or (2) can be verified but is not useful (i.e., GPS1 is trusted to provide location information but the UK3 evaluation metric rates any GPS source to provide only low-confidence information whereas high-confidence information is required by the UK3 policy). Therefore, UK3 needs to collect and evaluate evidence regarding

USCA and US3 using the *ad-hoc* network only, since the central directory at its operation command remains unavailable.

a. UK3's policy for providing "zone reports": (*Role* = UK/US military $\vee$ UN convoy) with confidence = medium $\wedge$(*Location*=neighbors) with confidence = high
b. UN1's request presents credentials: Cert(*Role*=UNConvoy)$_{USCA}$ Cert(*Location*/GPS=zone2)$_{GPS1}$ Cert(*Location*/Visual=zone2)$_{US3}$
c. UK3's trust relations: UKCA for *Role*; GPS1, UAV1, and UK1 for *Location*
d. UK3's metric for confidence evaluation of location evidence Type(source) = GPS and source trusted → confidence = low Type(source) = UAV and source trusted → confidence = low Type(src1) = UAV $\wedge$ Type(src2) = GPS and src1 and src2 trusted → confidence = medium Type(source) = Visual and source trusted → confidence = high Other → confidence = null
e. UK3's metric for confidence evaluation of role evidence: Type(source) = CA and source trusted → confidence = high Other → confidence = null

Table 2.1: An Example of a Policy Statement, Evaluation Metric, and Credentials and Trust Relations

2.4 Related Work

2.4.1 Pretty Good Privacy

In PGP [39], any user can sign another user's key. These signatures form a network of peer trust relations, often described as the *web of trust* [39]. The confidence in a trust path between two nodes of the web of trust is evaluated via a simple metric consisting of 4 "levels of trust" and a set of rules (e.g.: a key is marginally trusted if signed by two independent, marginally trusted, keys).

Although the PGP web of trust is fully peer-to-peer in its concepts, it is not in implementation. Public keys are published in *key servers* [32] maintaining a database of keys and discovering trust paths amongst them. This solution is efficient for the Internet but not possible for the MANET since there is no guaranteed connectivity with a key server. Hubaux *et al.* [18] propose a distributed implementation of PGP where each user stores a subset of the trust graph and proceeds to fusion of his set with other users' sets to discover trust path.

The trust metric implemented in PGP is simple and can lead to counter intuitive decision being made, as discussed by Mauer[20].

2.4.2 IBM's Trust Establishment system

IBM Research Laboratory developed a trust establishment framework [15] allowing the "bottom-up" emergence of a public-key infrastructure through exchange of certificates, containing various pieces of evidence about principals, and evaluation of these by a *Trust Policy Language*. When certificates about a principal are missing, they are automatically collected from peer servers. The policy language supports negative certificates, which allows complex non-monotonous policies. However, the trust policy language does not support *uncertain evidence* explicitly; as this is considered part of the policy specification.

This work is targeted to the Internet, where connectivity is guaranteed between servers. Missing certificates are collected from peer servers (either known *a priori* or referenced in other certificates). The collection mechanism is not suitable for the MANET environment were connectivity is not guaranteed. Our peer-to-peer evidence distribution mechanism would be a suitable solution to replace the certificate repositories and support the IBM's *trust engine* to provide a full peer-to-peer implementation.

2.4.3 The resurrecting duckling

Stajano and Anderson's resurrecting duckling [33] and its descendants [34] [2] represent a peer-to-peer trust establishment framework in which principals authenticate their communication channel by first exchanging keying material via an out-of-band physical contact. The goal of this approach is different from ours; i.e., it is not intended to provide peer-to-peer entity authentication, nor is it intended to handle uncertain evidence. The established trust is binary: the communication channel is either secure or is not.

Chapter 3

A Framework for Trust Establishment in the MANET

In this chapter, 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.

3.1 Overview

3.1.1 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 [33][34], 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.

3.1.2 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.

3.1.3 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 [38], [31], [23] and properties of these metrics have been studied [20]. However, the only practical trust metric developed and implemented has been the one of PGP [39]. 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.

3.2 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 [25],[14], the open-source [9] and research communities [1], [35]. 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) [9] and Oceanstore (which routes requests using Plaxton trees)[21].

3.2.1 Overview of Freenet

Freenet [9] 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

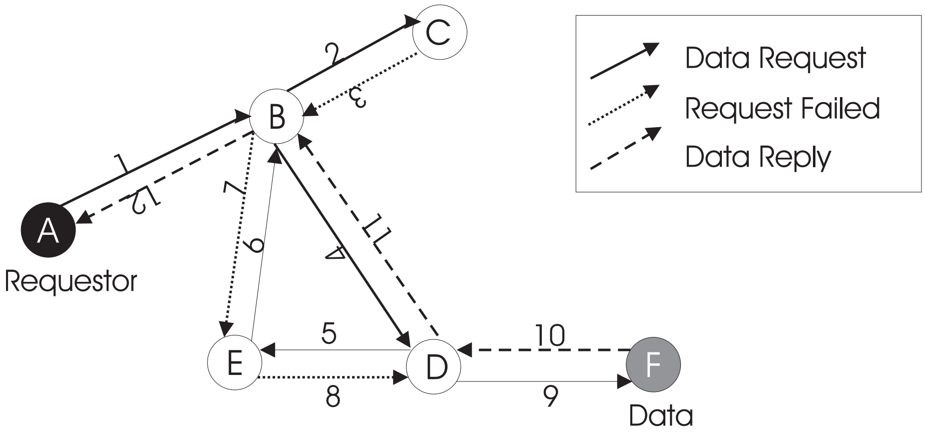

Figure 3.1: An example of a request routing in Freenet

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.

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 3.1 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.*[40].

3.2.2 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.

3.3 Swarm intelligence for trust evidence distribution.

3.3.1 Basic notions

Swarm intelligence [7] 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 [36][11]. 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.

3.3.2 A swarm-intelligence based scheme for evidence discovery

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.* [36], and Di Cargo and Dorigo [11] 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

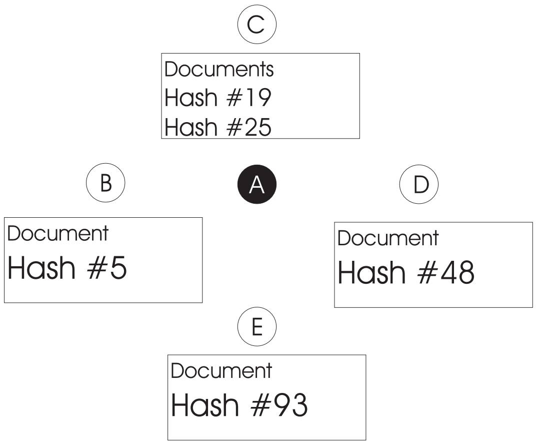

Figure 3.2: The topology used for example 3.3.3 Node A is in wireless range of B, C, D, E. The document stored and their respective hash is also showed

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 in similar routing decisions than freenet while preserving knowledge about all sources of evidence.

3.3.3 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).

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

Figure 3.2 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 3.3. 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 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. It can be seen on figure 3.3 that the routing decision for A will be exactly the same if Freenet was used instead of our swarm algorithm. Up to now the "clustering" of the hash space is identical with Freenet or with our swarm algorithm (e.g. node B will receive requests/inserts from A for hash #0 to #12).

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 as shown on figure 3.4. 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 on the routing table that the cluster for B is now only covering #0 to #9.

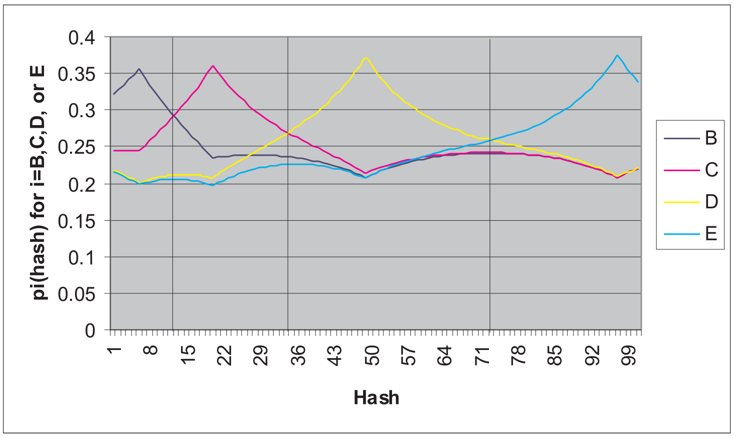

Figure 3.3: The probabilistic routing table of A, after scenario 1

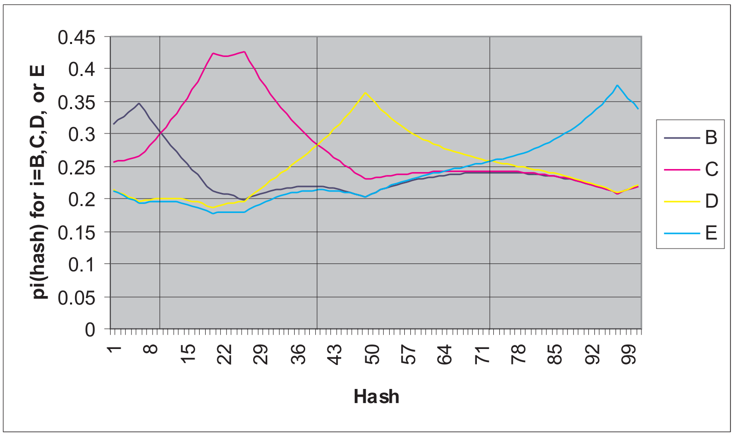

Figure 3.4: The probabilistic routing table of A, after scenario 2

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.

Chapter 4

Evaluation

In this chapter, we present the result of our simulations of freenet in a MANET to distribute trust evidence. We were interested in understanding the effect of mobility and routing on the performaces of freenet; we also wanted to measure the impact of the request routing on the diversity of evidence stored in the network.

4.1 Simulations framework

We implemented freenet in *NS-2* [26] above the *CMU Mobility extensions.* The agent is implemented as an application above the network layer such that the freenet packets (from one freenet node to another) are routed using standard protocol such as DSR, AODV, DSDV. However it is possible to disable the routing layer and to use the agent directly above the link layer; in this case the next hop of a request has to be a neighbor.

The mobility model is the random waypoint model. Nodes are characterised by a speed, randomly chosen between 0 and max_speed, and a pause time p during which a node stop moving before changing, randomly, of direction. Decreasing the pause time corresponds to increasing the mobility in the network, therefore to study the effects of mobility we run an experiment for various values of the pause time.

An experiment consists of multiple rounds. Each round has two phases: during the first phase 300 random documents are inserted and retrieved from the network from randomly chosen nodes; during the second phase exactly 100 requests, for documents known to have been inserted in the network, are performed and measurement is collected. As the round proceeds the routing is naturally improving and documents are being replicated through the network.

The network consist of 50 nodes (wireless range of 250 meters) randomly dispersed in a 1km x 1km zone. It means that the network is highly connected and that the average number of hops between two nodes is of 2.

Figure 4.1 shows a typicall experiment visualised under `NAM`. The nodes bouded by a square box shows where a specific document is stored (replicated via caching)

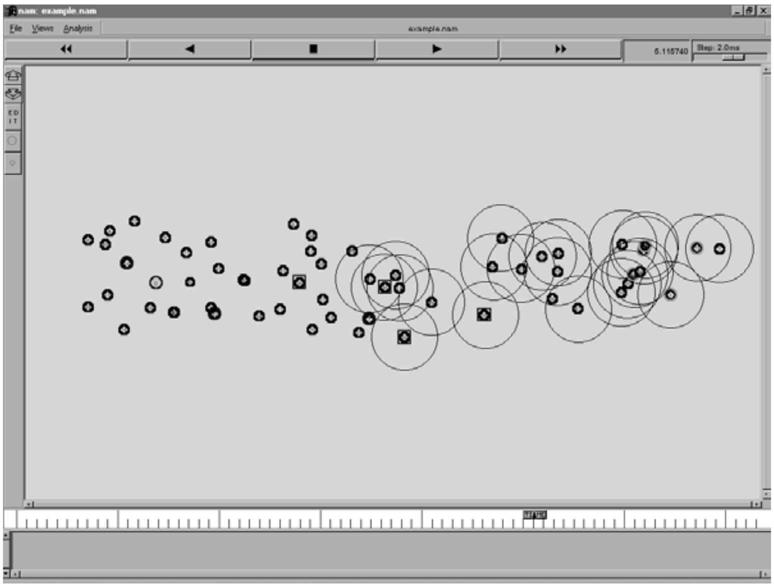

Figure 4.1: Visualizing an experiment with `NAM`

while the nodes bouded in a circle shows the visited nodes during a search.

4.2 Evaluation

4.2.1 Comparing Freenet to Gossiping

To understand the advantages of the hash keyword routing over a random selection of the next hop we compared the success rate and the average path lenght of freenet and a gossiping protocol.

Figure 4.2 shows that the Freenet routing converges quickly and outperforms the gossiping protocol. The average path lenght is computed only on request that return a positive result, this explain the high variance of the gossiping curve for early rounds. The gossiping without caching is stateless and provides results after a search on an average of 10 hops. The gossiping with caching improves with time since documents get replicated (more likely to be found). The difference between gossiping-with-caching and freenet is only the hashed keyword routing, which provides the expected improvement on the average path lenght and the success rate (figure 4.3).

4.2.2 Effect of underlying routing protocol

The goal of routing layer in the MANET is to maintain connectivity amongst node in presence of mobility and link failiures. Since the Freenet application depends on

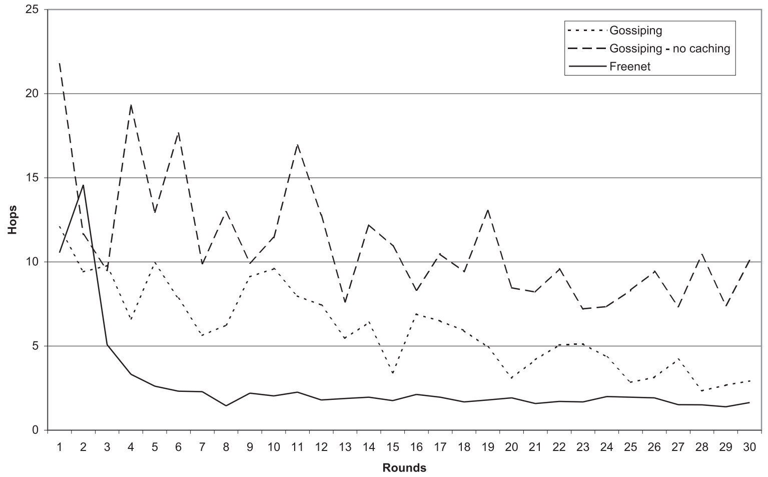

Figure 4.2: Average path lenght

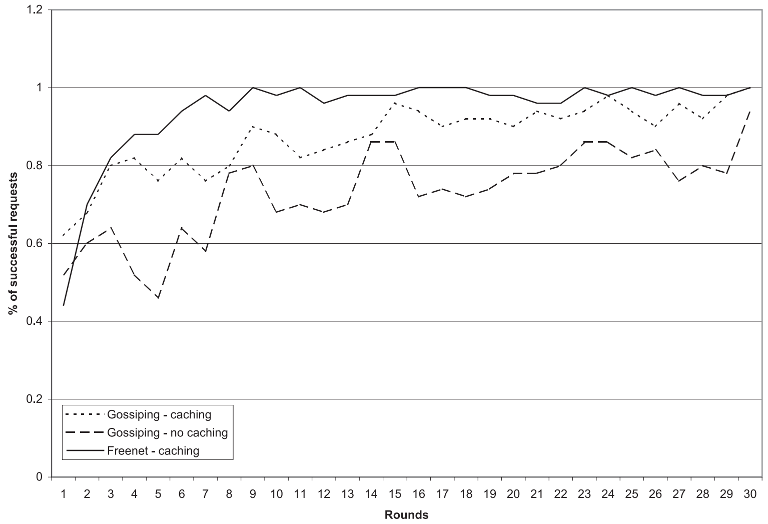

Figure 4.3: Success rate

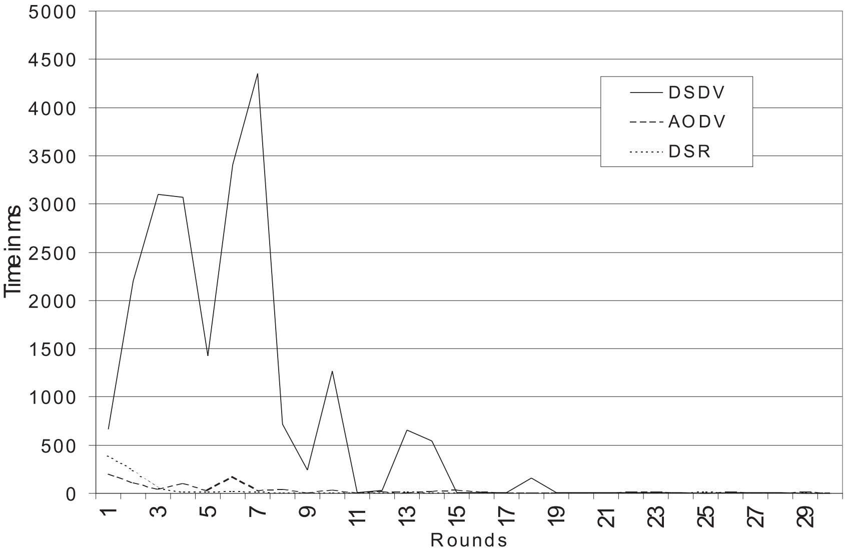

Figure 4.4: Avg. time to receive an answer (only successful requests are counted) while running freenet above different routing protocols

the routing layer to forward its data packets, we looked at the effect of the routing layer on the performances of freenet. It is usual to evaluate routing protocols by using Constant Bit Rate flows between mobile nodes and measure parameters such as the throughput, goodput, packets lost, etc. However the Freenet layer depends more on a short delay than a high reliability or low overhead of the routing layer.

As figure 4.4 shows, the two on-demand routing protocols (DSR and AODV) provide comparable results but DSDV needs a certain time (almost 20 rounds) before it can provide satisfying delays. The average time to answer a freenet request over DSDV is so large that it cannot be explained by just looking at DSDV; does this suggest the presence of bugs in the NS-2 implementation of DSDV?

4.2.3 Diversity of evidence

A characteristic of Freenet and many other p2p system is that for one request, one document is provided as an answer. With a gossiping protocol a user can reiterate its request multiple time to discover more than one document (since the exploration is random). However this is not possible with freenet. Repeating the same request will lead to the same result since the routing is reinforcing good path and there is no regulation (negative feedback). Moreover the caching of documents is helping to replicate highly requested documents but is also destroying the diversity of documents in the system.

Figure 4.5 shows the difference between Freenet and a gossiping protocol at

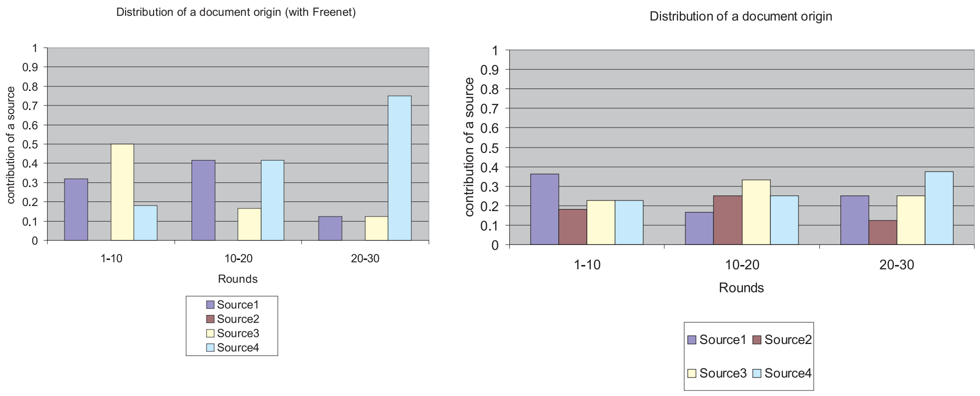

Figure 4.5: Comparing Freenet and Gossiping on the diversity of evidence preserved

preserving diversity. The same documents have been inserted and are requested in both experiment. A specific document is provided by four different sources (i.e. a piece of evidence is provided by 4 different principals); however in Freenet one of the source (source 2) is never discovered and, worst, a source dominates the other (source 4) after enough rounds. This is not the case with the gossiping protocol, the exploration being completely random (and stateless) all sources are exploited at the same level through the simulation.

Chapter 5

Conclusion

5.1 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, and 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.

BIBLIOGRAPHY

[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] R.B. Bobba, L. Eschenauer, V. Gligor, and W. Arbaugh, "Bootstrapping Security Associations for Routing in Mobile Ad-Hoc Networks," submitted for publication.

[7] 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.

[8] 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

[9] 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.

[10] *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*

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

[12] 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

[13] 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.

[14] Gnutella, http://www.gnutellanews.com/

[15] 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

[16] 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.

[17] 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).

[18] 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).

[19] 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.

[20] 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.

[21] 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.

[22] 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.

[23] 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.

[24] 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.

[25] Napster, http://www.napster.com

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

[27] G. O’Shea and M. Roe, “Child-proof Authentication for MIPv6 (CAM),” ACM Computer Communication Review, April 2001.

[28] 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.

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

[30] 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.

[31] 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.

[32] 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.

[33] 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.

[34] 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.

[35] 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.

[36] 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.

[37] 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.

[38] 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.

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

[40] 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.

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