Survivable Multi-Level Ad-Hoc Group Operations

Dan Zhou and Jie Wu
Department of Computer Science and Engineering
Florida Atlantic University
Boca Raton, FL 33431

Abstract

In this paper we investigate the survivability of multi-level ad-hoc groups for critical operations through a case study. We aim to provide a formal trust framework for establishing security policies. We identify various issues in group formation and evolution. We define components and rules of direct trust and trust recommendation. We then apply them in constructing functioning groups. We also propose some policy guidelines for addressing these issues.

1 Introduction

Numerous task groups are formed when there are new missions to be actualized, such as searching for a new chancellor of a university. Task groups have critical information, such as transcripts of committee meetings, that may come under attack from both within and without. We say a project survives and succeeds when it can withstand setbacks due to limited resource or fallibility of human beings and technology on which they rely. Survivability is defined as the ability of a system to function as desired in the event of passive component failure and active attacks [3].

In this paper we investigate the trust components of systems to make them more survivable. Our objective is to build a formal framework for trust establishment by studying the structure of complex projects. The formal trust model provides a formal foundation for operational policies.

One type of complex projects encompasses multiple tasks running across multiple domains, for instance, an international anti-terrorist alliance and a multi-track conference (Figure 1). The tasks associated require specific skills and have certain need for secrecy. They may be either domain independent or dependent. Some tasks are common to all domains. Others are restricted to certain domains. A domain may have information specific to itself. Assets in a project include members, information (such as mission and group membership), and other resources (such as fund). We use *node* to refer to both human beings and their machines. A project failure could originate from compromised nodes that lead to information disclosure, injection of wrong information, or lost resources. An outsider can compromise or impersonate a node. It can also conduct traffic pattern analysis to gain access to information that is otherwise unavailable.

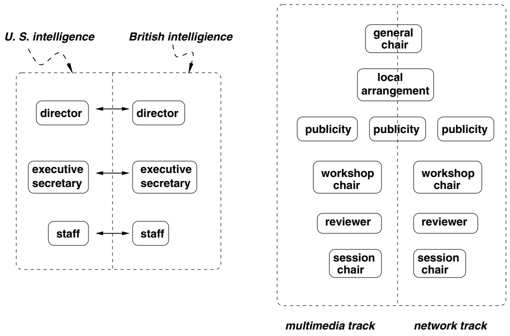

Figure 1. International anti-terrorist alliance and a multi-track conference.

An ad-hoc group charged with a project is born on a need basis with limited infrastructure support. The success of a critical project relies on the trust among group members and reliable group communication. The steering committee as the first sub-group formed for a project defines the goal for the project and the policy for group access control, therefore controls the nature of group dynamics. In particular it defines the credentials for accessing the group and the rules for trust evolution.

We organize the group into an access hierarchy and knowledge domains, and categorize information as access-level sensitive, domain sensitive, or both. A high-level group member has access to all the access-level sensitive information that a low-level group member has. Domain sensitive information is known only to members of a specific domain.

We structure the rest of the paper as follows. We give some preliminary of projects in Section 2. In Section 3 we propose a formal trust model. A case study is shown in Section 4. In Section 5 we discuss issues in critical projects and

Proceedings of the 23 rd International Conference on Distributed Computing Systems Workshops (ICDCSW'03)
0-7695-1921-0/03 $17.00 © 2003 IEEE

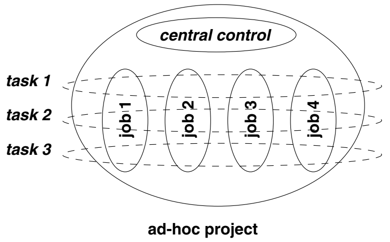

Figure 2. A complex, critical project.

policies for a survivable project. We conclude in Section 7.

2 Elements of a Project

Realizing a project is a complex process that needs careful planning. For a critical, confidential project involving multiple groups implementing multiple tasks in different domains, special attention is called for the planning and execution of the project. A project has three phases—*project initiation, planning and execution*—and two auxiliary elements—*group communication and trust building*.

Initiation. A project begins when someone or some group explores the possibility of carrying out some idea into reality. The group identifies the *credentials* needed for participation based on the objectives. The credentials come from the existing, trusted sources.

Planning. As the project moves to planning phase, a steering committee is formed out of the initial core group. The steering committee divides the project into tasks (see Figure 2) and assigns a sensitivity label to each task. Sensitivity label indicates the level of the need for confidentiality [7]. Sensitivity labels of tasks form a lattice. The steering committee also establishes the scope of the project through domains. We can represent a project $\mathcal{P}$ as a set of tasks ($\mathcal{T}_j$) spanning over a set of domains ($\mathcal{D}_i$): $\mathcal{P} = (\{T_i\}, \{D_j\})$.

Execution. In this phase nodes are assigned roles in the project. A role group is an assignment of a set of entities to a task or a domain. A node is assigned a role that carries out a task. Roles have clearance levels that permit them to access sensitive information pertinent to tasks. The clearance level of a node grants it access to confidential information at a certain sensitivity level. Clearance level of roles also form a lattice.

We use *project group* to refer to all the nodes that are a part of the project. A *task group* carries out one specific task of the project. A *job* group implements one instance of the project in a specific domain. Task and job groups are orthogonal partitions of project groups. Members of task groups are trained appropriately. Each job group has one leader which has the highest level of clearance of the group.

Without loss of clarity, we use $\mathcal{P}, \mathcal{T}_i, \mathcal{J}_j$ to represent the project group, task group i and job group j, respectively. The union of the job group is a subset of the project group because of domain independent tasks. The union of the task group is the project group: $\cup_j \mathcal{J}_j \subseteq \mathcal{P}$ and $\mathcal{P} = \cup_i \mathcal{T}_i$.

Group Communication. Communication is an indispensable component of any project. Intra-project communication falls into one of the four categories: inter- and intra-group communication for both task- and job-groups (that is, within $\mathcal{T}_i$, from $\mathcal{T}_i$ to $\mathcal{T}_j$, within $\mathcal{J}_i$, and from $\mathcal{J}_i$ to $\mathcal{J}_j$).

Trust. A project can not succeed without trust within. Nodes are *trustworthy* if they have *integrity* and proper *capabilities*. The levels of trust we place on nodes are based on credentials and built over time. There are two types of credentials: one for integrity and one for specific capabilities, say x. Credentials are issued by various sources to entities on various subjects.

3 Trust

We propose a trust model based on boolean values. This simplifies the model and provides clarity to complete trust relationship.

Trust between Entities. Let CA be a certification authority and n be a node. CA could be a regular node. We use predicates $trust$, int, and cap to represent trust, credential for integrity and credential for the ability to perform a task, respectively. Predicates $eval$ and $trrec$ represent a credential for ability to judge the ability of other nodes and our confidence in a node's recommendation, respectively. We use $intrec_x$ to denote the special recommendation.

CA int_x n: CA has confidence in n's integrity on subject x

CA cap_x n: CA has confidence in n's capability on subject x

CA $eval_x$ n: CA has confidence in n's ability to evaluate capabilities of other nodes on subject x

CA $trust_x$ n: CA trusts node n on subject x

CA $trrec_x$ n: CA has confidence in n's judgment of other nodes on subject x

CA $intrec_x$ n: CA has confidence in n's judgment on the integrity of other nodes on subject x

We define trust as the conjunction of integrity and capability. Confidence in a recommendation is the conjunction of integrity and certification ability.

$$\text{CA } trust_x \text{ n} =_{def} \text{CA } int_x \text{ n} \wedge \text{CA } cap_x \text{ n}$$

$$\text{CA } trrec_x \text{ n} =_{def} \text{CA } int_x \text{ n} \wedge \text{CA } eval_x \text{ n}$$

$$\text{CA } intrec_x \text{ n} =_{def} \text{CA } int_x \text{ n} \wedge \text{CA } eval_{int_x} \text{ n}$$

Integrity, capability, and trust can be recommended (see Figure 3). For instance, if David trusts Alice as a referral and Alice trusts Bob (or his integrity, or capability), then David trusts Bob (or his integrity, or capability). The relationship among these functions are as follows: David $intrec_x$ Alice $\wedge$ Alice int_x Bob $\Rightarrow$ David int_x Bob, David

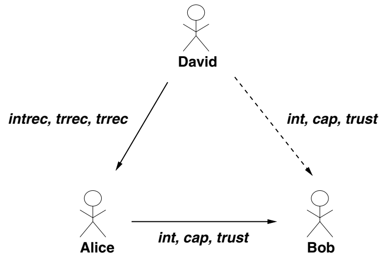

Figure 3. $Trust$ and int for trust and trust in integrity are transitive relationships.

$trrec_x$ Alice $\wedge$ Alice cap_x Bob $\Rightarrow$ David cap_x Bob, and David $intrec_x$ Alice $\wedge$ David $trrec_x$ Alice $\wedge$ Alice $trust_x$ Bob $\Rightarrow$ David $trust_x$ Bob.

It is assumed that certifications from some specific, well-known, standard certification agencies are accepted credentials for capabilities: $\forall m.c$ is a certification agency $\Rightarrow$ (c cap_x n $\Rightarrow$ m cap_x n). If nt is a task that does not require special capabilities, then $trust_{nt} = int_{nt} = trrec_{nt}$.

Group Trust. A node n trusts a group $\mathcal{G}$ if n trusts every member of $\mathcal{G}$ (n $trust_x$ $\mathcal{G}$ $=_{def}$ $\forall$j $\in$ $\mathcal{G}$. n $trust_x$ j). A group $\mathcal{G}$ trusts a node n if every member of $\mathcal{G}$ trusts n ($\mathcal{G}$ $trust_x$ n $=_{def}$ $\forall$j $\in$ $\mathcal{G}$.j $trust_x$ n). A group $\mathcal{G}$ trusts node n's referral if every member of $\mathcal{G}$ trusts n's referral ($\mathcal{G}$ $trrec_x$ n $=_{def}$ $\forall$j $\in$ $\mathcal{G}$.j $trrec_x$ n and $\mathcal{G}$ $intrec_x$ n $=_{def}$ $\forall$j $\in$ $\mathcal{G}$.j $intrec_x$ n). Group $\mathcal{G}_1$ trusts group $\mathcal{G}_2$ if every node in $\mathcal{G}_1$ trusts every node in $\mathcal{G}_2$ ($\mathcal{G}_1$ $trust_x$ $\mathcal{G}_2$ $=_{def}$ $\forall m \in \mathcal{G}_1$.$\forall$ n $\in$ $\mathcal{G}_2$. m $trust_x$ n).

Based on above definitions, we conclude the following recommended trust ($\mathcal{G}$ $intrec_x$ $m \wedge \mathcal{G}$ $trrec_x$ $m \wedge m$ $trust_x$ n $\Rightarrow$ $\mathcal{G}$ $trust_x$ n) and (m $intrec_x$ n $\wedge m$ $trrec_x$ n $\wedge$ n $trust_x$ $\mathcal{G}$ $\Rightarrow$ m $trust_x$ $\mathcal{G}$). A group $\mathcal{G}$ for task x is functional if there is a mutual trust within the group: $\mathcal{F}_x$ $\mathcal{G}$ $=$ $\wedge_{\mathrm{j} \in \mathcal{G}}$j $trust_x$ $\mathcal{G}$, where $\mathcal{F}_x(g)$ checks if group g is functional.

When two groups trust each other, the joint group is functional: $((\mathcal{G}_1\ trust_x\ \mathcal{G}_2) \wedge (\mathcal{G}_2\ trust_x\ \mathcal{G}_1)) \Rightarrow \mathcal{F}_x\ (\mathcal{G}_1 \cup \mathcal{G}_2)$. Node n joining a group is a reduced form of merging two groups: $((\mathcal{G}\ trust_x n) \wedge (n\ trust_x\ \mathcal{G})) \Rightarrow \mathcal{F}_x\ (\mathcal{G} \cup \{n\})$.

Nodes are trained for specific skills needed to perform particular tasks. Training accords nodes capability credentials. Integrity of nodes increases with the lapse of time. The time-dependence of integrity is not modelled in this paper due to its complexity. We can use $\mathcal{C}_{p_x}$ to represent a specific credentials where $\mathcal{C}_{p_x}(CA, n) = CA\ p_x\ n,\ p \in \{int, cap, eval, trust, trrec, intrec\}$.

4 Components of a Project

To illustrate the issues inherent in critical, ad-hoc projects we investigate one successful implementation of a Synchronized Deployment of an assignment in Multiple Locations (SDML) [5]. In this section we describe the project group organization and its operational policy.

Group Organization. SDML involved four tasks and four locations. Initially two small groups of people came together to form a steering committee and two task groups. More people were recruited and trained as the project progressed. Eventually the group was divided into location-specific domains. The final phase of the project was executed by these domain specific job-groups. Figure 4 shows the evolution of the project group.

Preparation. Three nodes, A, B, and C, formed a group $\mathcal{G}_1$: $\{A, B, C\}$ over the time. Another group, $\mathcal{G}_2$: $\{D, E, F\}$, was formed similarly. The trusts among these two groups were built through attending schools, growing up together, among other things. These two groups were functional based on a general, common goal (gg): $\mathcal{F}_{gg}$ $\mathcal{G}_1 \wedge \mathcal{F}_{gg}$ $\mathcal{G}_2$.

Initialization. Node D was inducted to group $\mathcal{G}_1$ through a trusted third node, M. Group $\mathcal{G}_1$ organized the project and divided it into four tasks: central control (cc), technique support (te), financial support (fi) and auxiliary support (au). These tasks fell into different levels of sensitivity: cc at the highest level of sensitivity, te and fi at the middle level, and au at the lowest level. Let $\mathcal{S}_t$ denotes the sensitivity level of a task t, where sensitivity represents the need for confidentiality. We have $\mathcal{S}_{cc} > \mathcal{S}_{fi} > \mathcal{S}_{au}$, $\mathcal{S}_{cc} > \mathcal{S}_{te} > \mathcal{S}_{au}$. Group $\mathcal{G}_1$ undertook both the tasks cc and te. Nodes E and F filled in for task fi.

Expansion. Task group $\mathcal{T}_{te}$ was later expanded with two nodes, H and I, through some credentials $\mathcal{C}_{trust(te)}$. Task au was filled with 12 nodes, $n_i, 1 \leq i \leq 12$, with credentials $\mathcal{C}_{trust(au)}$. The required credentials for au were less stringent than those for te and fi, which were less stringent than those for cc. The final task groups were as follows: $\mathcal{T}_{cc}$: $\{A, B, C, D\}$, $\mathcal{T}_{te}$: $\{A, B, C, D, H, I\}$, $\mathcal{T}_{fi}$: $\{E, F\}$, $\mathcal{T}_{au}$: $\{n_i, 1 \leq i \leq 12\}$.

Division. For the final deployment of the project, the project group was broken down into job groups. Each job group included at least one member with technical credential. The job groups were: $\mathcal{J}_A$: $\{A, n_1, n_2, n_3, n_4\}$, $\mathcal{J}_B$: $\{B, n_5, n_6, n_7, n_8\}$, $\mathcal{J}_C$: $\{C, n_9, n_{10}, n_{11}, H\}$, $\mathcal{J}_D$: $\{D, E, F, n_{12}, I\}$.

Operational Policy. SDML implemented an operational policy to maximize the probability of success while minimize the effect of damage. It set guidelines with respect to information sharing, data communication, and redundancy.

Information Sharing. The central control $\mathcal{T}_{cc}$ was aware of all the information essential to the completion of the project. Only minimum information was shared to other members whose tasks necessitated their knowledge. Knowledge of a lower-level task group was a subset of that of a higher-level task group. Let *info* denote the information known to a task group. The relationship among the information set was as follows: *info* $_{\mathcal{T}_{au}} \subset$ *info* $_{\mathcal{T}_{fi}} \subset$ *info* $_{\mathcal{T}_{cc}}$,

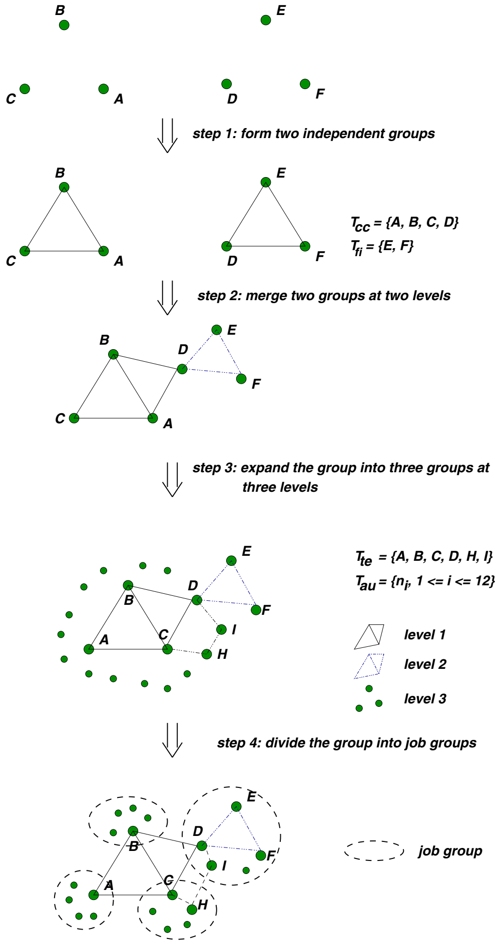
B
E
C
A
D
F
step 1: form two independent groups
B
E
C
A
D
F
Tcc = {A, B, C, D}
Tfi = {E, F}
step 2: merge two groups at two levels
B
D
E
C
A
F
step 3: expand the group into three groups at three levels
E
Tte = {A, B, C, D, H, I}
Tau = {ni, 1 <= i <= 12}
B
D
F
A
C
I
H
level 1
level 2
level 3
step 4: divide the group into job groups
E
D
B
F
A
C
I
H
job group

Figure 4. Group evolution: groups forms, merges, expands, and divides in four steps.

$\textit{info}_{\mathcal{T}_{au}} \subset \textit{info}_{\mathcal{T}_{te}} \subset \textit{info}_{\mathcal{T}_{cc}}$. Information pertinent to a group was shared by all members of the group. That is, $n \in \mathcal{G} \Rightarrow \textit{info}_{\mathcal{G}} \subset \textit{info}_{n}$.

The objective of the project, the structure of the project group, and group memberships were known to $\mathcal{T}_{cc}$, which consisted of nodes with highest sensitivity level. Others were not made aware of the objective nor the group structure. Job groups were formed in the last stage of the project deployment when domain-specific work was called for. In early stages domain specific information was known only to the central control.

Members were aware of the membership of their own group. A job-group member was also aware of the leader and other attributes, such as location, of its own job-group. For example, H was unaware of who the leader of job group $\mathcal{J}_D$ was. The membership information of a higher-level group was not disclosed to a lower-level group. H, I, E and F, members of tasks group $\mathcal{T}_{te}$ and $\mathcal{T}_{fi}$, had no knowledge of the membership of $\mathcal{T}_{cc}$. Moreover, H and I had no knowledge of that of $\mathcal{T}_{fi}$, nor did E and F of $\mathcal{T}_{te}$.

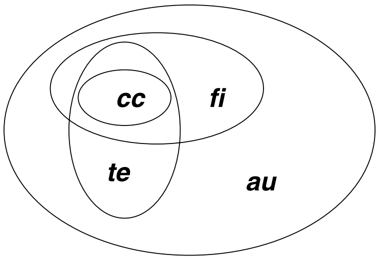

Figure 5. Multi-level group trust structure.

Communication. There were inter- and intra-group communications in SDML. Communications were both confidential and authentic within the project group. An intra-group message was known only to the members of the particular group and the sender was identified.

There were three types of inter-group communication: (1) A node in one group communicating with a node in a different group, for example, a private communication between H and E; (2) An individual in one group sending a message to a different group, for example, a multicast message from F to $\mathcal{T}_{cc}$ (used for emergency); and (3) One group sending a message to another group, for example, an announcement made by $\mathcal{T}_{cc}$ to $\mathcal{T}_{fi}$.

Central control $\mathcal{T}_{cc}$ spoke through one voice which was recognizable to all members of the project group. Each member of $\mathcal{T}_{cc}$ could speak for the task group. A message coming from $\mathcal{T}_{cc}$ was identified as from the group, not from any individual member. A message coming from a lower-level group to a higher-level group was identified as from the individual sender. Anonymity was not maintained for the low-level group members.

Redundancy. Adequate redundancy was built into the project in case of node failure. Each task group had a multiplicity greater than one so that a task could still be accomplished if one node was lost.

Each job group required a technical expert. There were six nodes in task-group $\mathcal{T}_{te}$ and four job groups for the project. Lose of up to two (arbitrary) nodes before deployment would not affect the course of the project. Suppose A had been lost, H would have become the technical expert for job-group $\mathcal{J}_A$ and group $\mathcal{J}_C$ would have one less member: $\mathcal{J}'_A : \{H, n_1, n_2, n_3, n_4\}$, $\mathcal{J}'_C : \{C, n_9, n_{10}, n_{11}\}$. H would have been made aware of the information relevant to job group $\mathcal{J}_A$, among other information, but not the detail attributes of other job groups.

5 Operation Survivability

Neither members of a project group nor technology the project depending upon is infallible. In addition, environments are often unfriendly. Therefore, a project needs measures to withstand external and internal attacks.

Survivability Issues. There are three categories of assets in a project: nodes, information and infrastructure. Nodes are physical entities that need various levels of protections depending on their roles. A skilled member is more valuable than a non-skilled one. A leader is more valuable than a staff. There is a variety of information within a project: its objectives, organizational structure, membership, messages including their sources and destinations, message flow patterns, and resource utilization information. Infrastructure of a project is the collection of nodes and the communication path among them.

An attacker could target any one of the assets to find out about the existence of a project so as to destroy it or to defend again it. The attacker could find out the group leader and make it a target to maximize the damage of an attack.

Table 1 lists measures against various types of interception of assets. Table 2 lists measures against interruption, modification and fabrication attacks. For example, to protect information infrastructure against interruption, there should be at least two communication paths out of a group. Traffic should be uniformly distributed among paths to avoid hot spots in communication. However, not all paths are equal. Some routes should be avoided for certain type of information.

Mechanisms need to be in place to detect compromises of nodes. A project needs constant monitoring to evaluate trusts placed on nodes. One evaluation criterion is to monitor their behavior and detect deviation from normality.

Trust Establishment and Evaluation. Groups are established through trusts. Trusts in nodes enable us to authorize their actions. Trust is initialized through credentials and evolve over time. Credential for capability $\mathcal{C}_{cap}$ has discrete values and is relatively easy to verify. It can be obtained through appropriate training. Credential for integrity $\mathcal{C}_{int}$ is continuous and its evaluation is intrinsically ambiguous. Its value changes over time and can be calibrated through continuous monitoring.

Credentials for integrity can be self-issued or other-issued (i.e., issued by another node). An other-issued credential has a higher credibility than a self-issued one. Self-issued credentials evolve over time and will be superseded by other-issued credentials. A credential has no more credibility than that of its issuing party.

Creditability of information depends on two factors: the source and the path that it has travelled. The more we trust the source and the nodes along the path, the more credible the information is. The shorter the route, the more the creditability. Some critical information can only be trusted if it goes through specific routes. In the extreme case, some information is trusted only when the exchange is direct.

In a multi-level system trust relationship can be organized hierarchically. In terms of trust, members of high-level group are also members of lower-level group. For the example project, the trust relationship among groups are shown in Figure 5. $\mathcal{T}_{cc}$ is the least trusting. $\mathcal{T}_{au}$ is the most trusting. $\mathcal{T}_{te}$ and $\mathcal{T}_{fi}$ fall in between.

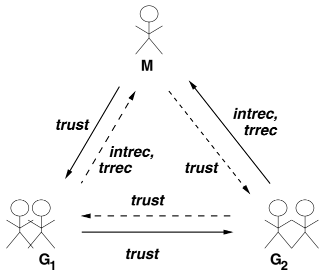

Figure 6. Recommended trust for B and D through M.

Trust Initiation. The trust relationship between two entities builds either from scratch or on some other trust relationship. For low criticality-level operations we can afford to take risks on trusting an unknown entity. Participation in critical operations require a well-established trust relations as a foundation.

Groups $\mathcal{G}_1$ and $\mathcal{G}_2$ were formed before the start of the project. In addition, they had established trust relationship with other entities. In particular, both $\mathcal{G}_1$ and $\mathcal{G}_2$ trust recommendations from a third party M, while M trusts both $\mathcal{G}_1$ and $\mathcal{G}_2$ on the goal gg (see Figure 6). That is, $(\mathcal{G}_1\ intrec_{gg}\ M) \wedge (\mathcal{G}_2\ intrec_{gg}\ M) \wedge (M\ int_{gg}\ \mathcal{G}_1) \wedge (M\ int_{gg}\ \mathcal{G}_2)$.

The goal gg does not require any technical credential. Based on the above discussion, we can derive $trust_{gg} = int_{gg}$ and $(\mathcal{G}_1\ trust_{gg}\ \mathcal{G}_2) \wedge (\mathcal{G}_2\ int_{gg}\ \mathcal{G}_1)$. Also, we have $\mathcal{G}_1\ trust_{gg}\ \mathcal{G}_2$ and $\mathcal{F}_{gg}(\mathcal{G}_1 \cup \mathcal{G}_2)$. Among the four tasks, task te is the only one requires technical credential. Hence we have $trust_u = trrec_u = int_u, u\ \in \{au, fi, cc\}$.

Trust Evolution. Group evolves similar to the trust initiation. We can similarly prove that $\mathcal{T}_{au}, \mathcal{T}_{te}, \mathcal{T}_{fi}$ and $\mathcal{T}_{cc}$ are all functioning. The functionality of a multi-level group is defined as the trust among groups based on their perspective view of other groups. This view hides the internal structure and information of a high-level group from low-level groups. That will be the topic of a subsequent paper.

6 Related Work

Among related work, Eschenaure, Gligor and Baras presented their observations of trust establishment in ad-hoc networks and argued for a "swarm-intelligence" approach for the trust distribution and establishment between peers [4]. PGP presents a mesh trust-network where trust is tied to the acceptance of a key as legitimate or a node's integrity in issuing certificates [8]. PKI provides a domain-based certificate hierarchy where trust is present in the decision of a wholesale acceptance of certificates issued by a certificate authority [1]. In these instances trusts between

Table 1. System Survivability Measures Against Interception Attacks

asset	*measure*	*objective*
message	encryption	confidentiality
	physically getting together	
message flow and resource utilization patterns	multiple ports for each group	avoid hot spot
	multiple paths among groups	avoid hot route
	evenly distribute traffic	hide patterns
	localize traffic	reduce travel time
membership and organization structure	hierarchical access control	restrict access to sensitive information
	drop-box for high-level nodes	hide high-level nodes
	mix with outside nodes	mask nodes in the project group
mission	least privilege	information is known only to high-level nodes
	least privilege	a node knows as little and as late as possible

Table 2. Other System Survivability Measures

attack	*measure*	*objective*
interruption	multiple paths between any two groups	protect communication infrastructure
	multiple nodes in each group	maintain group availability
	shared control of resources and command	sustain node failure
modification	error checking code	integrity
fabrication	shared control	sustain insider compromise
	message authentication code (MAC)	authenticity, accountability

entities are implicit in the certificates that they issue and accept. There has been some work in separating trust and access control policies, for example, Mahoui, Bhargava and Zhong proposed a system for trust management [6].

7 Conclusion

In this paper we investigated the process and structure of complex, critical projects through a case study. We identified issues paramount to survivability of these projects and proposed operational policies and guideline for a sustainable project. We addressed the often overlooked area of initialization of ad-hoc groups (and networks).

We proposed a simple trust model for nodes and groups. We assigned boolean values to trusts and recommendations. We distinguished integrity (behave as expected) from capability for both direct trust and recommendations. We investigated trust separately from security. We identify a hierarchical access structure for groups and studied the relationship between trust and functioning group projects.

Our work is applicable to the security of ad-hoc networks be it closed, semi-open, or open [2]. Any network must be initialized, which is critical to the security of the network during its lifetime.

References

[1] A. Nash, W. Duane, C. Joseph, and D. Brink. *PKI: Implementing and Managing E-Security*. Osborne/McGraw-Hill, 2001.

[2] B. Dahill, B. N. Levine, E. Royer, and C. Shields. A secure routing protocol for ad hoc networks. Technical Report Technical Report UM-CS-2001-037, University of Massachusetts, Amherst, August 2001. ftp: ds.internic.net.

[3] R. J. Ellison, D. A. Fisher, R. C. Linger, H. F. Lipson, T. A. Longstaff, and N. R. Mead. Survivability: Protecting your critical systems. *IEEE Internet Computing*, November/December 1999. condensed version of technical report Survivable Network Systems, CMU/SEI-97-TR-013.

[4] L. Eschenauer, V. D. Gligor, and J. Baras. On trust establishment in mobile ad-hoc networks. In *10th International Workshop on Security Protocols, April 2002, Cambridge, UK*, 2002.

[5] V. E. Krebs. Mapping networks of terrorist cells. *Connections, the official journal of International Network for Social Network Analysis (INSNA)*, 24(3):43–52, 2001.

[6] M. Mahoui, B. Bhargava, and Y. Zhong. Separating between trust and access control policies: A necessity for web applications. In *IEEE workshop on Reliable and Secure Application in Environment, New Orleans, Louisiana*, October 2001. Also appeared as CERIAS TR 2002-07.

[7] R. C. Summers. *Secure Computing : Threats and Safeguards*. McGraw-Hill, 1996.

[8] P.R. Zimmermann. *The Official PGP User's Guide*. MIT Press, Cambridge, Massachusetts, 1995.