

**Dynamic Cryptographic Context Management
(DCCM)¹**

Report #3

**Cryptographic Context Negotiation
Protocol**

February 24, 1999

Mr. David M. Balenson, PI
Dr. Dennis K. Branstad²
Mr. Peter T. Dinsmore
Mr. Michael Heyman
Ms. Caroline Scace

Cryptographic Technologies Group
TIS Labs at Network Associates, Inc.
3060 Washington Road (Rt. 97)
Glenwood, MD 21738

¹ Supported by the Defense Advanced Research projects Agency (DARPA) under Rome Laboratory Contract No. F30602-97-C-0277.

² Previously an employee of Trusted Information Systems and a Principal Investigator; presently a part-time, temporary employee of TIS Labs at Network Associates, Inc.

Abstract

The Cryptographic Context Negotiation Protocol (CCNP) report is the third publication resulting from the Dynamic Cryptographic Context Management (DCCM) project. This project is sponsored by the Defense Advanced Research projects Agency (DARPA) and is being performed by Network Associates TIS Labs (previously Trusted Information Systems, Inc.) This report specifies communication protocols for use in negotiating a cryptographic context among a very large number of participants from various organizations working together on a large, sensitive computer project. A cryptographic context negotiation template (CCNT) to be used during this negotiation was specified in report number 2 [Bal98a] of this project. This template specification has been revised and augmented, making it more extensible and easier to automate [Bal98b]. The protocol is used to establish a cryptographic association among a large group of participants and to enforce the cryptographic provisions of the security policy imposed on a sensitive computer application or project.

The DCCM project assumes that a number of organizations, perhaps with differing security policies, have agreed to work cooperatively on a specific project under the control of a designated project manager. The CCNT is designed to accommodate negotiating a single cryptographic-based security policy based on the desires of the project manager and the constraints of the organizations and individuals participating in the project.

The significant results of the DCCM project contained in this report include a detailed description, in text and flow diagrams, of the context negotiation for the DCCM project.

Keywords

Access control; authentication; authorization; confidentiality; cryptographic context; cryptography; group keys; integrity; group-key management; large-group management; multicast; multi-party security; peer-peer security protocols; policy negotiation; security policy.

Contents

1. INTRODUCTION	1
1.1 Project Overview	1
1.1.1 Project Goals	2
1.1.2 Project Scope	2
1.2 Report Overview	3
2. SYSTEM POLICY	4
3. CRYPTOGRAPHIC CONTEXT NEGOTIATION TEMPLATE.....	6
4. CRYPTOGRAPHIC CONTEXT NEGOTIATION PROTOCOL	7
4.1 Background, Scope of the Cryptographic Context	7
4.1.1 The Security Representatives, Hierarchical and Flat Models.....	7
4.1.2 Security During Negotiation.....	8
4.1.3 The Role of Project Initiator	8
4.1.4 The DCCM Mediator	8
4.2 The Cryptographic Context Negotiation Template (CCNT)	9
4.3 Sequential Narrowing of Cryptographic Contexts	11
4.4 Cryptographic Context Negotiation Protocol (CCNP) in 3 Phases	12
4.4.1 Phase 1: The Proposal	13
4.4.2 Phase 2: The Responses.....	14
4.4.3 Phase 3: Resolution and Dissemination of Final Context	14
4.5 Proposal Strategies and Negotiator Orientations.....	16
5. DCCM MESSAGE FORMATS	18
5.1 Negotiation Messages.....	18
5.2 System Messages	22
5.2.1 Concept of Operation	23
5.2.2 Scalability and Keeping Capabilities Secret.....	24
5.2.3 Starting a Session from Scratch.....	25
5.2.4 Evicting a Participant from a Project/Session	40
5.2.5 Resynchronize	42
5.2.6 Joining a Session	44
5.2.7 Key Derivation	45
6. NEXT STEPS	46
ACKNOWLEDGMENTS	47
REFERENCES	48

List of Figures

Figure 1: Relationship between policy, context and templates.....6
Figure 2: Two Negotiators using DCCM system mediator and templates10
Figure 3: Sequential constraints on the cryptographic context solution space12
Figure 4: Negotiation high-level flow.....13
Figure 5: DCCM communication flow26

List of Tables

Table 1: Security representative preference matrix15

1. Introduction

1.1 Project Overview

The Dynamic Cryptographic Context Management (DCCM) project is addressing the problem of providing security for sensitive information technology projects involving large, dynamically-changing groups of participants. For example, command and control of tactical military forces from different armed forces units and perhaps from different countries and working together under one commanding officer for a period of time or for a specific military exercise will require a variety of security services. By “large” we mean groups with a number of members ranging from approximately 1,000 to 100,000. By “dynamic,” we mean that new members may be added to the group at any time and existing members may be evicted from the group (e.g., a position may be overrun by enemy forces), thereby requiring immediate changes to some of the security provisions. Members need not be humans; they can include a variety of communicating entities such as sensors, mobile client workstations, server workstations, or network nodes. Participants in a project (i.e., members of the group authorized to participate actively in the project) can be organized in several ways, ranging from one large uniform group under a single management to a complex organizational structure with several layers of management.

In the DCCM Architecture and System Design report, we laid a framework for:

- Categorizing large-group projects and their security requirements;
- Defining large-group management models and authorization needs;
- Identifying candidate security context management solutions; and
- Evaluating candidate solutions with regard to the requirements of particular projects.

In the DCCM Cryptographic Context Negotiation Template (CCNT) reports we presented:

- An overview of the policies that can be encoded in a CCNT; and
- a specification of a CCNT in the Backus-Naur Form (BNF) syntax specification language.

In this report, we present:

- A description of the cryptographic context negotiation protocol of the DCCM project which uses the revised CCNT [Bal98b].

1.1.1 Project Goals

The goals of the DCCM project are to:

- Design a management system which identifies, authenticates, authorizes, and manages members of sensitive large-group computer projects;
- Create a security policy language for a project initiator to easily define the security policy to be invoked for the project;
- Create a translator for translating the selected security policy into a preferred security context using a cryptographic context template;
- Develop a large-group security context negotiation protocol that derives a group cryptographic context to be used for a project;
- Develop a testbed to demonstrate all aspects of dynamic cryptographic context management;
- Implement software in the testbed to demonstrate a manager workstation and a number of client workstations managing and participating in a large, dynamic group project.
- Transfer the technology and the software to other researchers and developers working on large-group sensitive projects.

This report addresses the fourth goal above.

1.1.2 Project Scope

The scope of the DCCM project includes all aspects of managing the security of a large-group information technology project. Specifically, it includes: security policy definition, security context specification, cryptography-based security context negotiation, group authorization management, group authentication management, large-group keying algorithms and protocols, and efficient re-keying following dynamic changes of the group.

The scope of this report includes using the security context specification language in the negotiation of a single cryptographic context for any sensitive large, multi-party project. The negotiated context must satisfy the policies of the project's manager while satisfying the constraints of the organizations participating in the project to the greatest extent possible.

1.2 Report Overview

While focusing on the Cryptographic Context Negotiation Protocol (CCNP), this report begins by presenting background information on the DCCM concepts of security policy, policy negotiation template, and cryptographic context. Each of these concepts is presented and then the relationships between the three are explored. Note that this report does not cover the internals of the negotiation template. This material is covered in DCCM Report #2, Version 2.

Next, the CCNP itself is presented. The final section of the report specifies the detailed message format for the application messages necessary to implement the protocol and for the application messages supporting the DCCM system.

2. System Policy

Within the DCCM architecture, the primary information to be exchanged to establish secure group communications involves policy regarding cryptographic security and group operating procedures. Group managers and members seek to establish secure associations by agreeing on acceptable cryptographic mechanisms, algorithms and parameters specifying how the ensuing communications are to be protected. The DCCM architecture also supports establishing rules governing dynamic group membership and the ways a security association should accommodate those rules. It presents a framework for establishing policy governing the management of, and communications within, multi-organizational groups. The framework encompasses all categories of project policy and organizational policy necessary to establish an acceptable cryptographic context for any arbitrary group.

Policy and context negotiation use inputs from different sources, including the architects and implementers of DCCM. The inputs include goals and constraints of both people and organizations. Negotiation also depends greatly on the availability of cryptographic mechanisms and support services for a project. In order to facilitate this negotiation, the template supports a wide variety of security mechanisms described at various levels of specificity. A static template was used as a working model during architecture development, was replaced by a dynamic formal model in the first CCNT report, and has been revised to an extensible dynamic formal model in the present template design. The cryptographic context negotiation template and protocol will continue evolving during system development towards a target of completeness and flexibility.

Within the DCCM model, very large dynamic groups are created, maintained and managed. In the initial DCCM model, each project group must agree upon an overriding policy that applies to all group communications for the duration of the project. This includes all sessions established under that project. Hence, the DCCM system policy must cover both traditional systems security issues as well as group dynamic actions.

Policy exists at different layers of an organization. Policy is normally broad when established at the top of an organization and is more detailed at the lower layers. Cryptographic policy is a part of security policy that is a part of information technology policy that is a part of corporation or government policy. Higher levels of policy from all participating organizations influence the security policy of a project. A cryptographic context must satisfy the specific policy negotiated among the participating organizations of each sensitive project.

By **security policy** we mean the layered set of objectives, rules, regulations, principles and practices which specify secure system operation. Security policy includes, but is not limited to, a **cryptographic policy** that defines the levels of protection needed in the traditional cryptographic security services of confidentiality, authentication and data

integrity. A security policy also covers diverse areas such as key management, access control, availability assurance, and related factors. **Mechanisms** are objects or functions that are used to enforce rules or deliver services. They are policy enablers and enforcers. Cryptographic algorithms, which encrypt, hash and digitally sign data, are the mechanisms that facilitate cryptographic security services. Passwords, hardware tokens and biometric procedures are examples of mechanisms associated with personal identification.

A **context** is a set of protocols and mechanisms, along with their associated parameters, that have been selected to carry out a specified security policy. A **cryptographic context** is a set of cryptographic protocols and algorithms, along with their modes of operation, intended to address the appropriate categories of the cryptographic policy. Relative to each of these categories, the policy will specify the level of protection to be attained. The algorithms and their modes selected for the cryptographic context must be consistent with these prescribed protection levels.

Hence, policy issues reside at a higher level than that addressed in a cryptographic context. The context is a more detailed version of policy. It is an instantiation or translation of a policy using description specifications of finer granularity. Cryptographic mechanisms enforce a protection rule (or deliver a security service) at some prescribed level that has usually been determined by a group of experts. This translation from policy to mechanism is not static since the confidence in the robustness of a particular mechanism may change over time. Nonetheless, this **mapping** from specified policies to sets of mechanisms is essentially automatic and hence can be performed by an automated, (e.g., expert) system. DCCM uses an **expert mapping** system to transform policy specifications into a detailed context.

3. Cryptographic Context Negotiation Template

Policy issues can be expressed with varying degrees of granularity. As the granularity increases, rules and procedures become more definitive. A context is a natural extension of policy when the policy is expressed in specific, finely tuned terms. This extension is performed by the DCCM policy-to-context translation system. The **DCCM Cryptographic Context Negotiation Template (CCNT)** is used by negotiators to negotiate policy specifics and understand each other's mechanism availability.

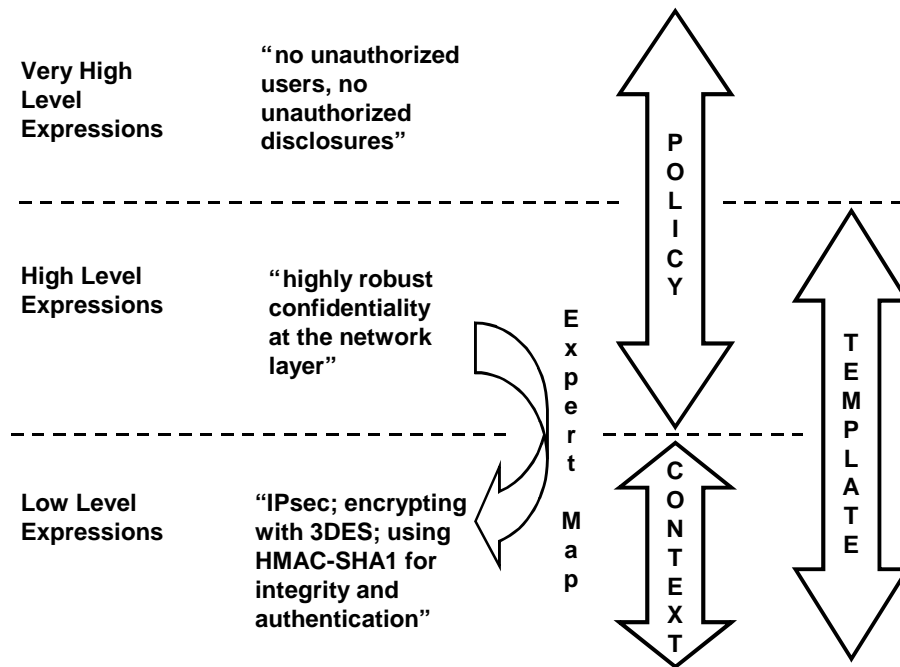


Figure 1: Relationship between policy, context and templates

Figure 1 illustrates the relationship between policy, the context, and the template. Context negotiators will enforce their organization's cryptographic policy and constraints established by their managers while focusing on specific policy issues at the next level of granularity in the negotiation process. In particular, they will be constrained by availability of cryptographic mechanisms in the equipment of the proposed project participants and the efficiency and security provided by the mechanisms.

4. Cryptographic Context Negotiation Protocol

4.1 Background, Scope of the Cryptographic Context

The role of the DCCM system is to support secure communications among a large number of users. Once a set of DCCM users has identified a mutual need for secure communications, they use the CCNP to negotiate security services, cryptographic mechanisms, and corresponding parameters to be used to secure the group communications. The system architecture considers such a group of users to be a project group and supports up to 100,000 participants in a single project.

The cryptographic context, along with agreed-upon values for all parameters required by the algorithms, completely determines how project communications are to be secured. The scope of the context is the duration of a project, i.e. once the project participants negotiate a project context the project context will not change.

The sessions that arise within the project are conducted in accordance with the project context. Secure sessions within the umbrella of the project are easily initiated without further security negotiation requirements. The DCCM system supports highly dynamic session membership changes. Up to 100,000 project participants may join a single session. Thus the negotiation of the longer-term project cryptographic context among a fixed membership enables the desired highly dynamic sessions.

The CCNP is the set of negotiation steps through which an extensible set of cryptographic mechanisms is gradually narrowed through the use of a structured template to a fully specified cryptographic context. The goal is to efficiently reach agreement among the many negotiators, while giving precedence to a project initiator on a final cryptographic context agreement.

4.1.1 The Security Representatives, Hierarchical and Flat Models

Security representatives initiate DCCM projects and negotiate cryptographic contexts for the projects. Depending on the presence or lack of a hierarchical organization, a participant may be represented by another security representative or may take on that role for themselves.

The DCCM system supports negotiations among various organizations and representatives having various structures and relationships. For example, within hierarchical organizations such as private companies, government agencies, or military units only, individuals designated as managers or officers may take on a security representative role in the DCCM System. Those representatives are appropriately authorized to determine for their organization security levels and mechanisms in accordance with the organizational security policy. On the other hand, consider a model

such as an Internet USENET newsgroup which has no pre-established authorization hierarchy. The DCCM system, specifically the negotiation protocol, also accommodates this unstructured negotiation model. Each user or project participant then takes on the role of security representative for himself or herself during the context negotiation.

4.1.2 Security During Negotiation

DCCM negotiations are conducted in accordance with an initial, or default, cryptographic context. The default communications protocol during negotiation will be reliable, or connection-oriented, unicast.

An extension to include multicast communication during negotiation may be practical. The practicality of a default cryptographic context for securing multicast communication depends on the number and diversity of negotiators. Among a few negotiators, unicast communications will suffice. Thus it is desirable, within an existing hierarchical organization, for a security representative to negotiate on behalf of many participants.

4.1.3 The Role of Project Initiator

To start a project, one of the security representatives is given (or takes) the role of project initiator. The project initiator provides the project's name and the identities of other organizations participating in the project, or security representatives required for the negotiation, to the DCCM System. The project initiator formulates the proposal for the cryptographic context negotiation.

4.1.4 The DCCM Mediator

The CCNP can be thought of as step by step narrowing of the broad set of all cryptographic mechanisms available to all participants for all possible projects to that subset acceptable to the project initiator and the other security representatives. As the options are narrowed, the mediator adjusts the context template to guide negotiators' responses to a more restricted solution space. The mediator constrains negotiators during the response and resolution phases according to the other negotiators' submissions received during the previous phases, e. g. responses are constrained by the initial proposal. Resolution is aimed toward finding acceptable and optimum intersections among the negotiator's responses.

The mediator uses local constraints to guide the negotiator in selecting acceptable security service levels and to insure that the negotiator selects appropriate cryptographic mechanisms to support those policies. Negotiators may negotiate different cryptographic contexts for different projects. It is likely that negotiators will select service levels, cryptographic mechanisms, parameters etc. subject to long-term constraints within their organization's security policies, subject to available cryptographic mechanisms, and/or subject to personal preferences. In order to optimize the negotiation process, negotiators

may provide the mediator with configuration files designed to limit their own options in accordance with local policies. The mediator accepts configuration files including organization policies and available mechanisms and adjusts the template to present to the negotiators only those options consistent with the local configuration.

The mediator also uses the template to assure that selections for security service levels and mechanisms correspond. A negotiator wishing to designate strong confidentiality must designate at least one cryptographic mechanism that supports strong confidentiality.

4.2 The Cryptographic Context Negotiation Template (CCNT)

Initially the DCCM team envisioned a DCCM system-wide "expert mapping" which could translate policy statements into cryptographic mechanisms. The system-wide "expert mapping" was envisioned to support negotiation of high level policies followed by the DCCM System mapping of those policies into a set of cryptographic mechanisms or a cryptographic context. Both of the context negotiation protocols presented in Report #2 rely on the system-wide expert mapping to insure negotiated policies could be implemented via the negotiated cryptographic mechanisms. On reflection the DCCM team determined DCCM system's reliance on a single, system-wide, expert mapping inflicted unreasonable constraints on the system and it's users. Some of the concerns addressed include:

- Not all users will agree with an expert mapping.
- In the current climate of cryptographic advances, the mappings will require constant upgrades.
- Constant upgrades may, in turn, inspire disagreement among users.

While giving up nothing in terms of ability to negotiate cryptographic contexts, the DCCM team has decided to support configurable, local mappings in addition to a DCCM system-wide "expert mapping." This broader approach will accommodate easy upgrades and avoid inflicting security service level to cryptographic mechanism mappings on those who prefer to map them themselves.

Negotiators may choose to work with a DCCM system expert mapping or to use the mediator to generate a local template configuration file of mappings between security service levels and cryptographic mechanisms. Whether one prefers an expert mapping or a customized local mapping, there is always an implied mapping between higher-level policy statements and the mechanisms which support them.

Thus in terms of cryptographic mechanisms, each negotiator is allowed to speak their own high-level policy language. The mechanisms I refer to when I request strong authentication and integrity services may differ from the mechanisms another negotiator maps to the same level of service. Local mappings are used exclusively at the local level. The negotiators may provide the Mediator with local mappings in a Template

configuration file. The mediator may then use the local mapping to support a negotiator who wishes to generate proposals or responses based on high-level policies. The system will disallow proposal or response sets of cryptographic mechanisms inconsistent with the local mapping.

In the broader domain of many negotiators, to resolve multiple responses into a cryptographic context, the mediator relies exclusively on the common language of cryptographic mechanisms. The mediator disregards high-level policy statements and evaluates only the cryptographic mechanisms and the corresponding operational parameters in the proposals and responses. Thus it is both efficient and logical, once any higher-level policy statements are mapped into mechanisms, to disregard the higher-policy statements at the local level. The simple solution for the DCCM negotiation protocol is to communicate only cryptographic mechanisms among negotiators.

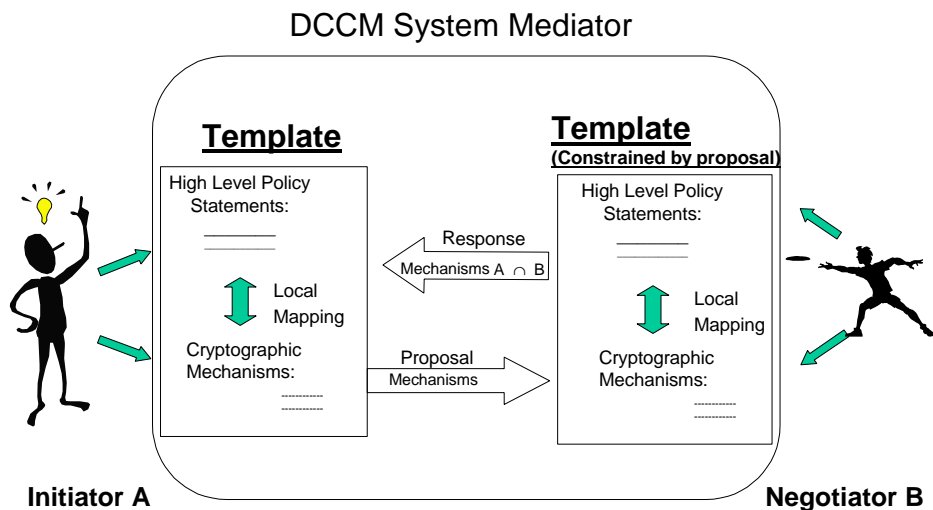


Figure 2: Two Negotiators using DCCM system mediator and templates

Figure 2 depicts the following DCCM negotiation protocol principles:

- Negotiators use templates based on local mappings of high-level policies to cryptographic mechanisms to generate proposals or responses.
- High-level policy statements are used only in the local domains.
- High-level policy statements are not passed back and forth among negotiators.
- Only sets of cryptographic mechanisms form the proposals and responses that the mediator passes among negotiators.

If one negotiator designates 3-DES for confidentiality and considers 3-DES to deliver "strong" confidentiality, and another negotiator designates 3-DES and considers it to deliver "very strong" confidentiality, the resulting context will

include 3-DES for confidentiality. The failure of the negotiators' mappings to agree with each other is irrelevant to the negotiators and to the system. The high-level statements need not be transferred beyond a local boundary.

Example 1

Example 1 contrasts the local scope of high-level policy statements with the system-wide scope of the mechanism language.

4.3 Sequential Narrowing of Cryptographic Contexts

As previously stated, the CCNP can be thought of as step by step narrowing of the “solution space” for the problem of finding a context that is acceptable to a large set of proposed participants in a project. At each step in the protocol, negotiators have the option to constrain their own responses to comply with their organization's policies or personal preferences. In addition, as the negotiation proceeds, options are narrowed. The mediator adjusts the template to limit negotiators' responses to the most recently narrowed solution space.

The mediator uses local constraints pointed to by the initiator to guide the initiator to select security service levels within local policy constraints and to insure the initiator selects sufficient cryptographic mechanisms to support those policies. In Figure 3 the local constraints on the proposal are represented by Region 1 while local constraints on responses are represented by Region 3.

The mediator constrains negotiators according to other negotiators' submissions during negotiation. Negotiators generating responses are also constrained by the original proposal. The mediator tailors the template for the negotiators such that they may only designate cryptographic mechanisms from the proposal set. The response to the proposal must be selected from the intersection of Regions 2 and 3, highlighted in Figure 3 with stars. Region 4 represents such a response. Because the final cryptographic context is determined by the intersection of all responses, it will necessarily be constrained to the starred region.

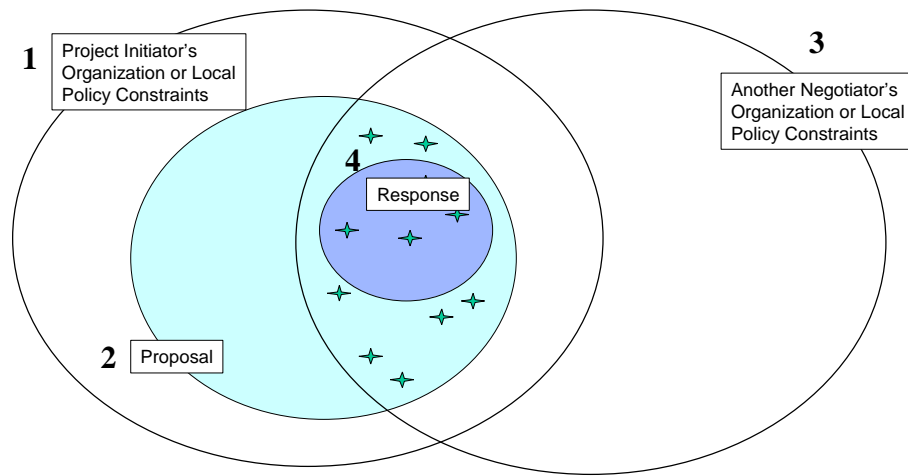


Figure 3: Sequential constraints on the cryptographic context solution space

4.4 Cryptographic Context Negotiation Protocol (CCNP) in 3 Phases

In terms of communication passes, the cryptographic context negotiation protocol is a 3-way protocol. It is based on 3 communications of reduced cryptographic solution spaces, as described in Section 4.3. A number of security representatives may negotiate a final cryptographic context in as few as 3 communication passes. Proposal strategies which increase the likelihood of efficiently reaching resolution in 3 passes are discussed in Section 4.5. Section 4.4.3.1 includes a discussion of implementations that expand communication requirements to include 2 additional passes in the Resolution phase.

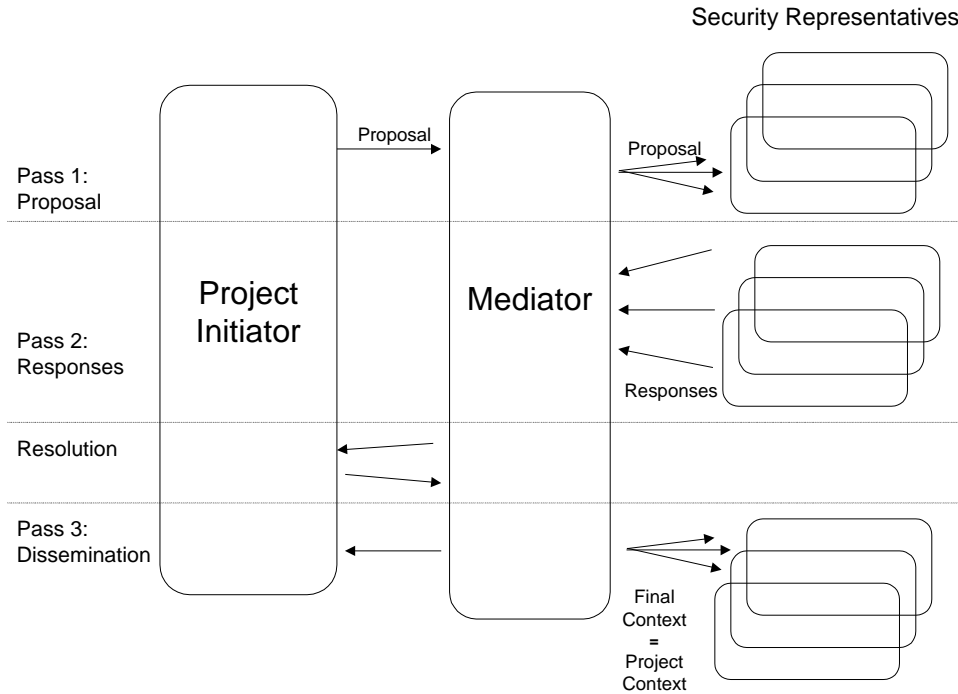


Figure 4: Negotiation high-level flow

Figure 4 is a sequential representation of the messages that the security representatives and the mediator communicate through the course of the CCNP. To expand upon the protocol and to include the interactions required in creating each message prior to dispatch, it is appropriate to structure the protocol in 3 phases,

1. Proposal/Solicitation of Responses
2. Response
3. Resolution and Dissemination

where each phase involves possible reduction of the cryptographic context solution space followed by communicating the reduction to the negotiator(s) for the subsequent phase. The phases are described in detail in the following paragraphs.

4.4.1 Phase 1: The Proposal

4.4.1.1 Formulation of the Proposal

The project initiator uses the DCCM mediator and the CCNT to generate a proposed CC for the project. The initiator has the option to point the mediator to stored context configuration files that are used to customize the negotiation template. Configuration files include local constraints due to organization policies or personal policies, locally available cryptographic mechanisms, operating parameters, and multicast tools, and the local mapping of high-level policies to cryptographic mechanisms. The mediator uses local constraints pointed to by the initiator to guide the initiator to select security service

levels within local policy constraints and to insure the initiator selects sufficient cryptographic mechanisms to support those policies.

4.4.1.2 Communication of Proposal, Solicitation of Responses

The project initiator transmits a proposed policy context to the mediator that transmits the proposal to all the other negotiators for the project. Recall the DCCM system mediator is distributed software that each negotiator executes and interacts with other parts of the distributed mediator. Each negotiator for the project executes a local copy of the DCCM system mediator. The distributed mediator receives and interprets the proposal, soliciting responses from all negotiators subject to the initiator's constraints.

4.4.2 Phase 2: The Responses

The security representatives respond with the *subset* of acceptable service levels which satisfy their security requirements (and their constituent's requirements) and the corresponding set of mechanisms they are prepared to use. For example, if elements of the proposed set include insufficient security levels or require security mechanisms the organization or user can not implement, the negotiator returns a correspondingly *reduced* set of service levels and mechanisms.

4.4.2.1 Response Formulation

Like the project initiator, each negotiator has the option to point the mediator to configuration files, which the mediator can use to customize the template. Configuration files include local constraints due to organization policies or personal policies, locally available cryptographic mechanisms, operating parameters, and multicast tools, and the local mapping of high-level policies to cryptographic mechanisms. The mediator uses local constraints pointed to by the negotiator to guide the negotiator to select security service levels within local policy constraints and to insure the negotiator selects sufficient cryptographic mechanisms to support those policies.

The mediator tailors the template such that responders may only designate cryptographic mechanisms from the proposal set. To support negotiation using higher-level policy statements such as "Confidentiality - Strong", the mediator tailors the template so that high-level policy statements utilize mechanisms in accordance with the *local* mapping.

4.4.2.2 Communication of Responses

All of the negotiators send their responses to the DCCM mediator running on behalf of the project initiator.

4.4.3 Phase 3: Resolution and Dissemination of Final Context

4.4.3.1 *Deriving a Cryptographic Context from the Responses*

The mediator attempts to resolve the responses into a final cryptographic context to be the project cryptographic context. One possible implementation would allow the mediator to consult the project initiator as required for intelligent resolution of conflicts.

The mediator receives and evaluates all the responses from all the project security representatives (except the project initiator). Specifically, the mediator seeks an intersection or solution space among the responses. There are 3 possible outcomes the mediator must then manage: The intersection space may represent more than one possible cryptographic context, less than one possible cryptographic context or precisely one cryptographic context.

Should the intersection include multiple options for at least one cryptographic mechanism or parameter, the mediator presents the options to the project initiator. The mediator is poised to assist the initiator in several capacities. For each option set from which the initiator must choose, the mediator can supply guidance. The mediator may note certain algorithms or parameters are better suited to real-time applications than another. It may indicate which algorithms are encumbered. What the initiator knows when handed multiple options for a policy or mechanism is that all the negotiators and their constituents can work with any option presented. The initiator will be in a better position to please all the negotiators if it knows the negotiators' order of preference. Should the Template support weighted inputs, the mediator is poised to present the initiator with negotiator preferences in the form of a matrix or bar graph. If the mediator tabulates preferences and presents an initiator with Table 1, the initiator can optimize selections to satisfy the majority of Security Representatives. In this example the initiator will please more Security Representatives by selecting 3-DES over IDEA. Alternatively the mediator can tabulate which choice will satisfy the most constituents.

	1 st Choice	2 nd Choice
3-DES	75% Security Representatives	25%
IDEA	25%	75%

Table 1: Security representative preference matrix

The mediator selects a cryptographic mechanism from each set of possibilities to remove all ambiguities.

Consider the case for which no intersection exists for at least one cryptographic mechanism or parameter. Several alternative courses of action may produce an intersection. The mediator transmits possible lists of organizations to exclude and corresponding cryptographic contexts to the project initiator. To create an intersection, the initiator must request a negotiator to add a cryptographic mechanism, or be excluded from the project. As for the multiple intersection case discussed above, the mediator can

supply supporting statistics such as the negotiator preference matrix in Table 1. The initiator selects a course of action and transmits it to the mediator.

All the negotiators except for one are willing to work with SHA-1 for integrity. The mediator notes that if the "outlier" will agree to work with SHA-1, an intersection exists.

Example 2

All negotiators except for 1 are willing to work with SHA-1 for integrity. All negotiators except for 2 are willing to work with MD-5 for integrity. The mediator allows the initiator to consider who may be excluded and select the corresponding mechanism.

Example 3

What if the responses resolve directly into a completely specified cryptographic context? What if for each security service there is precisely one mechanism both acceptable and available to all negotiators? This scenario seems least likely. A top-down negotiation, for which the initiator proposes a fully specified cryptographic context, is likely to yield a certain number of negotiators to be excluded. A bottom-up negotiation is likely to yield a few ambiguities. But should a single intersection exist, the mediator then has the simple task of communicating that intersection as the negotiated cryptographic context to the negotiators.

4.4.3.2 Dissemination of the Negotiated Cryptographic Context

The mediator transmits the final cryptographic context to all the security representatives. The security representative then distributes the project information including project name, project ID, multicast address and ports, DCCM system public key, and project context to their employees or constituents. All the project participants now have all the information they need to communicate for the duration of the project.

4.5 Proposal Strategies and Negotiator Orientations

The project initiator forms a *set* of acceptable security service levels and corresponding mechanisms to propose to the other negotiators. In forming a proposal set, the initiator will consider that the protocol requires each response to the proposed set of cryptographic mechanisms and parameters be a *subset* of the proposed set. Thus the relative size (i.e. solution space) of the proposal set will determine the orientation of the negotiation, the range of information the negotiator responses may include, and likelihood for an intersection within the three-way exchange. The set-size extremes for proposal set sizes and the corresponding negotiation orientations are described below. Of course many practical proposal sets fall between the extremes.

To initiate a top-down negotiation the initiator proposes a completely specified context. On receipt of a fully specified context, negotiators can infer the initiator is not flexible i.e. an incompatible response will result in exclusion from the group rather than increased flexibility. Negotiators, whose required security service levels or mechanisms are incompatible, may indicate the incompatibility by returning a reduced set. This scenario is appropriate for a content provider distribution model. The provider determines which cryptographic mechanisms to support. Subscribers are empowered only to the extent that they can select from the approved mechanism(s).

For a flatter model or a more flexible negotiation within existing hierarchies, the initiator proposes the complete set of possible contexts. Thus the initiator conducts a *comprehensive* assessment of the other organization or user policies and available mechanisms (i.e. The initiator and in turn the mediator does not limit the response set size.) Conducting the comprehensive assessment allows the initiator to view and select from among all intersections among the negotiators before determining the final context.

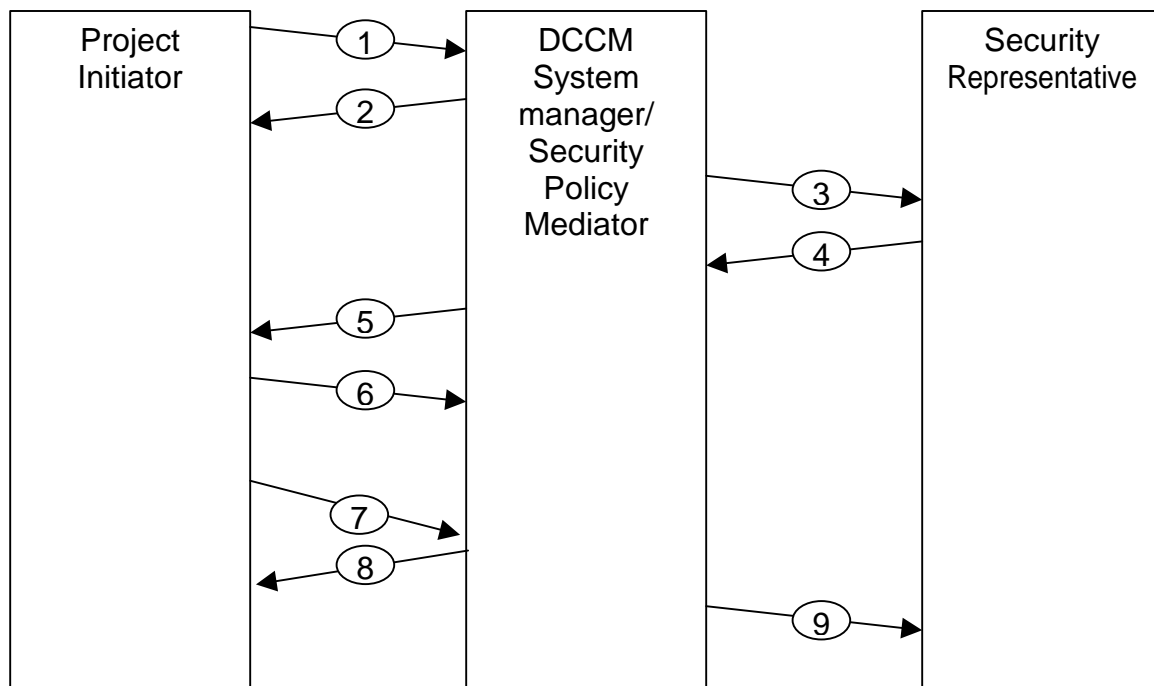
5. DCCM Message Formats

The CCNP is an application level protocol designed to run over TCP/IP services. Each message in the protocol has a common header consisting of a magic number, message type, and message length followed by a message body. Each of the header fields is 32 bits in length. The magic number is used to identify these messages as belonging to the DCCM protocol, and to verify correct decryption of DCCM messages. The message type specifies the operation to be performed by the recipient as well as the format of the body of the message. The message length specifies the length of the message, including the header and body, in octets.

There are a large number of messages required to implement the DCCM system. This section will focus first on the message specific to the context negotiation, and will then specify the remainder of the messages in the DCCM system.

5.1 Negotiation Messages

This section describes the message flow and formats for the messages used for the cryptographic context negotiation.



1. The project initiator (a special security representative) opens up a channel with the DCCM System manager. The project initiator requests a list of security representatives if it needs one. The project initiator requests the start of a project by

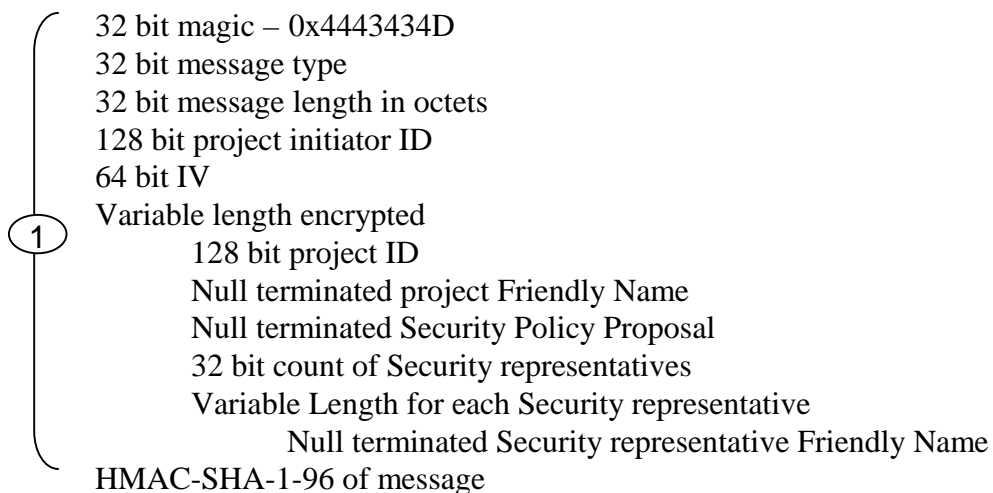
sending the system manager a context template and a list of security representatives for the project.

2. The start of a new project is acknowledged.
3. The security policy mediator opens up a channel to the security representative. A policy proposal is sent. After receiving a security policy from the security policy mediator, the security representative must agree to all or part of the policy. There are parts of the SPL that are ignored when they come from security representatives (such as the axis label and category information), so the SPL for this message 4 does not include these parts.
4. A response is returned along with a list of participants for the project.
5. The security policy mediator opens up a channel to the project initiator. A reduction of the policy proposal is sent with a list of security representatives in the project.
6. Acknowledge.
7. The project initiator then decides on a project security context and transmits that information to the DCCM system manager. Included in this transmission is a list of participants for the project that is under the project initiator.
8. Acknowledge.
9. The DCCM system manager transmits the project security context to all the security representatives (and possibly all the participants).

Rationale: Relatively high cost exchange of participant list only done once per project (if at all – we may not separate the Security Policy mediator from the DCCM System manager).

Keys derived from the project initiator's pair-wise secret with the DCCM System manager secure communications with the project initiator. If the project initiator is not found, then project will not get started.

Message Contents:



- 2
- 32 bit magic – 0x4443434D
 - 32 bit message type
 - 32 bit message length in octets
 - Encrypted 32 bit return code
 - HMAC-SHA-1-96 of message

Rationale: A key derived from the project initiator’s pair-wise secret with the DCCM System manager secures all communications. The secret comes from the registration of project initiator as a security representative. The project initiator must have permission to be a project initiator.

The system manager assures that all Security representative friendly names are unique.

- 3
- 32 bit magic – 0x4443434D
 - 32 bit message type
 - 32 bit message length in octets
 - 64 bit IV
 - Variable Length Encrypted
 - 128 bit project ID
 - Null terminated friendly project name
 - Null terminated project Session Description Protocol (SDP) Session Description per RFC 2327. Tells the socket, duration, etc for the project.
 - Null terminated Security Policy Proposal
 - HMAC-SHA-1-96 of message

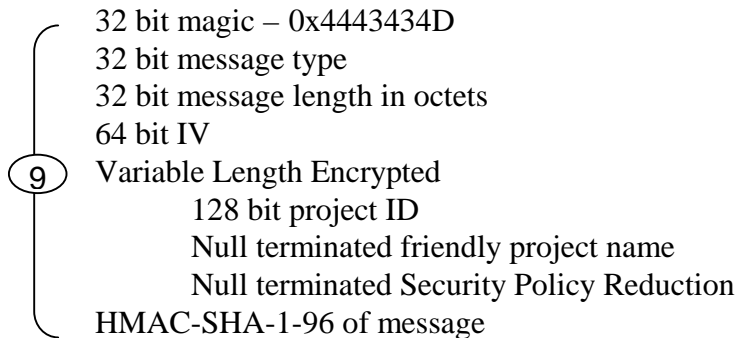
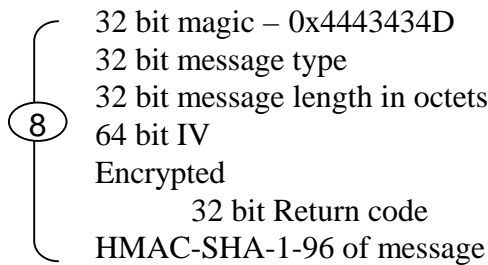
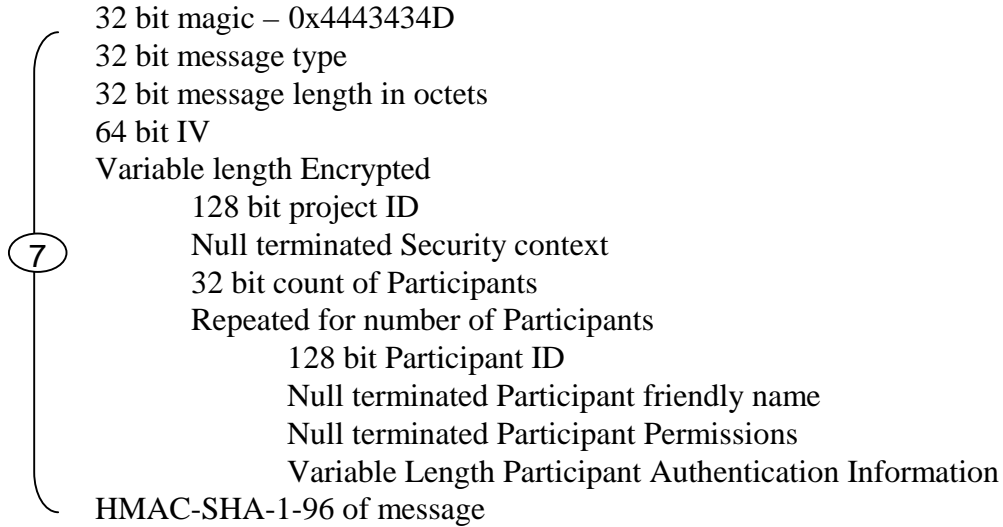
- 4
- 32 bit magic – 0x4443434D
 - 32 bit message type
 - 32 bit message length in octets
 - 64 bit IV
 - Variable Length Encrypted in negotiation key
 - Null terminated Security Policy Response
 - 32 bit count of Participants
 - Variable length encrypted in Participant dump key (mediator does not know this key)
 - Repeated for number of Participants
 - 128 bit Participant ID
 - Null terminated Participant friendly name
 - Null terminated Participant Permissions
 - Variable Length Participant Authentication Information
 - HMAC-SHA-1-96 of message

Rationale: A key derived from the security representative's pair-wise secret with the DCCM system manager secures communications (3) and (4). If the security representative is not found, then the participants represented by that security representative will not get in the project.

The "Session Description Protocol (SDP) Session Description" is a standard (rfc 2327) for multicast announcements. The format is in "x=<text>" style.

5 { 32 bit magic – 0x4443434D
 32 bit message type
 32 bit message length in octets
 64 bit IV
 Variable Length Encrypted
 128 bit project ID
 Null terminated friendly project name
 Null terminated Security Policy Reduction
 32 bit count of Security representatives
 Repeat "count" times
 Security representative Friendly name
 HMAC-SHA-1-96 of message

6 { 32 bit magic – 0x4443434D
 32 bit message type
 32 bit message length in octets
 64 bit IV
 Encrypted
 32 bit Return code
 HMAC-SHA-1-96 of message



5.2 System Messages

The previous section dealt with just the negotiation. This negotiation must exist in the larger context of the DCCM System. This section deals with the full DCCM system communications, highlighting how the negotiation messages fit in.

5.2.1 Concept of Operation

A DCCM system is designed to manage up to 100,000 participants working concurrently in one or more secure, real-time, interactive projects. Projects are divided into sessions and are protected in accordance with a Cryptographic Context (CC) that is negotiated for each project. A DCCM system during operation will consist of authorized users accessing project communications and using system features that are not accessible to unauthorized network users.

For efficient negotiation of a security policy for a new project and an initial CC, a security representative represents every potential participant of the project during the negotiation. The security representatives utilize the DCCM Cryptographic Context Negotiation Protocol (CCNP) while representing their organization's interests for the proposed multi-organization project. The negotiation interactions and the communications required to create new projects and negotiate CCs are conducted in accordance with a DCCM system policy that has been created by the DCCM system manager. Thus a hierarchy of policy is created within, and enforced by, the DCCM system to protect sensitive projects.

A new project can be initiated by an individual who has been given authority to be a project initiator by the DCCM system manager. The project initiator is authorized to work with a set of security representatives to initiate the project. The initiator and the security representatives negotiate a project CC that meets their policies. All communications for the project's sessions are then protected in accordance with the CC. In addition to policy levels and cryptographic mechanisms used to provide the needed security services, the CC also designates operating parameters for the DCCM system. Security representatives also identify and authorize individuals within their organizations to participate in the project, providing a list of these participants to the DCCM system.

The project initiator submits the project name to the DCCM system that returns a multicast address and 2 ports: one for secure session announcements and one for project session group key management

The DCCM CCNP is used within the DCCM Negotiation Mediator (NM). The NM facilitates negotiation among the security representatives of a CC. In addition to providing their security policies and available cryptographic mechanisms, the security representatives are responsible for maintaining the list of authorized participants from their organization for the project.

Each project participant shares a secret with DCCM system. The system can use the shared secrets as base keys for other cryptographic operations such as One-way Function Tree (OFT) group keying. Once the project's CC is determined, the mediator distributes it to the security representatives. They then distribute the Project Information (PI) including project name, project ID, multicast address and ports, DCCM system public key, and Project Cryptographic Context (PCC), to the project participants. All project communication is protected in accordance with this PCC.

The DCCM system manages group keying subject to the PCC. The system generates the project group key and/or group key generation material and multicasts to the project participants over the project key address/port. Secrets, shared between the participants and the system (salted with project specific information), are used to encrypt the group keys or group key generation data. The project participants monitor the project multicast key address/port and pick-up their group key or group key generation data. Participants decrypt the group key or group key generation data with their project secrets. In the case of an OFT, participants generate their group key locally using encrypted, blinded ancestor keys. Once they hold a project group key, the participants can decrypt the secure session announcements that are multicast over the project announcement address/ port.

Participants monitor the project multicast announcement address/ port for secure session announcements encrypted with the project group key. They may use the attached certificate to authenticate the announcement origin. All project participants are authorized to join all project sessions. The system logs session membership and distributes it to project participants on request.

Session and project memberships are monitored and adjusted by the DCCM System. Participants may voluntarily leave sessions as easily as they joined them. On receipt of a session leave request, the system re-keys the session and logs the session membership change.

Security violations are noted to the session initiator who notifies the project initiator and appropriate organizational supervisor. The supervisor of the participant committing the violation can request the DCCM system to revoke the participant's privileges. The system re-keys all projects and active sessions such that the offending participant is excluded.

5.2.2 Scalability and Keeping Capabilities Secret

In the real world a security policy mediator will need to communicate with a possibly large group of security representatives. Also, the negotiation protocol has the highest likelihood of arriving at a context in one pass when the security representatives are willing to admit to all of their capabilities (which they may not want widely known). To this end, it is prudent to split the security policy mediator functions from the DCCM system manager functions. Multiple security policy mediators can operate in parallel doing the mediation. Plus, the mediator has no need to know whom it is dealing with (as long as the security representative is authorized) while the DCCM system manager would know whom it is dealing with if it conducted the negotiations.

5.2.3 Starting a Session from Scratch

This picture shows the initial series of messages that occur to get a session started in the DCCM system. The communications are numbered in the order that they occur:

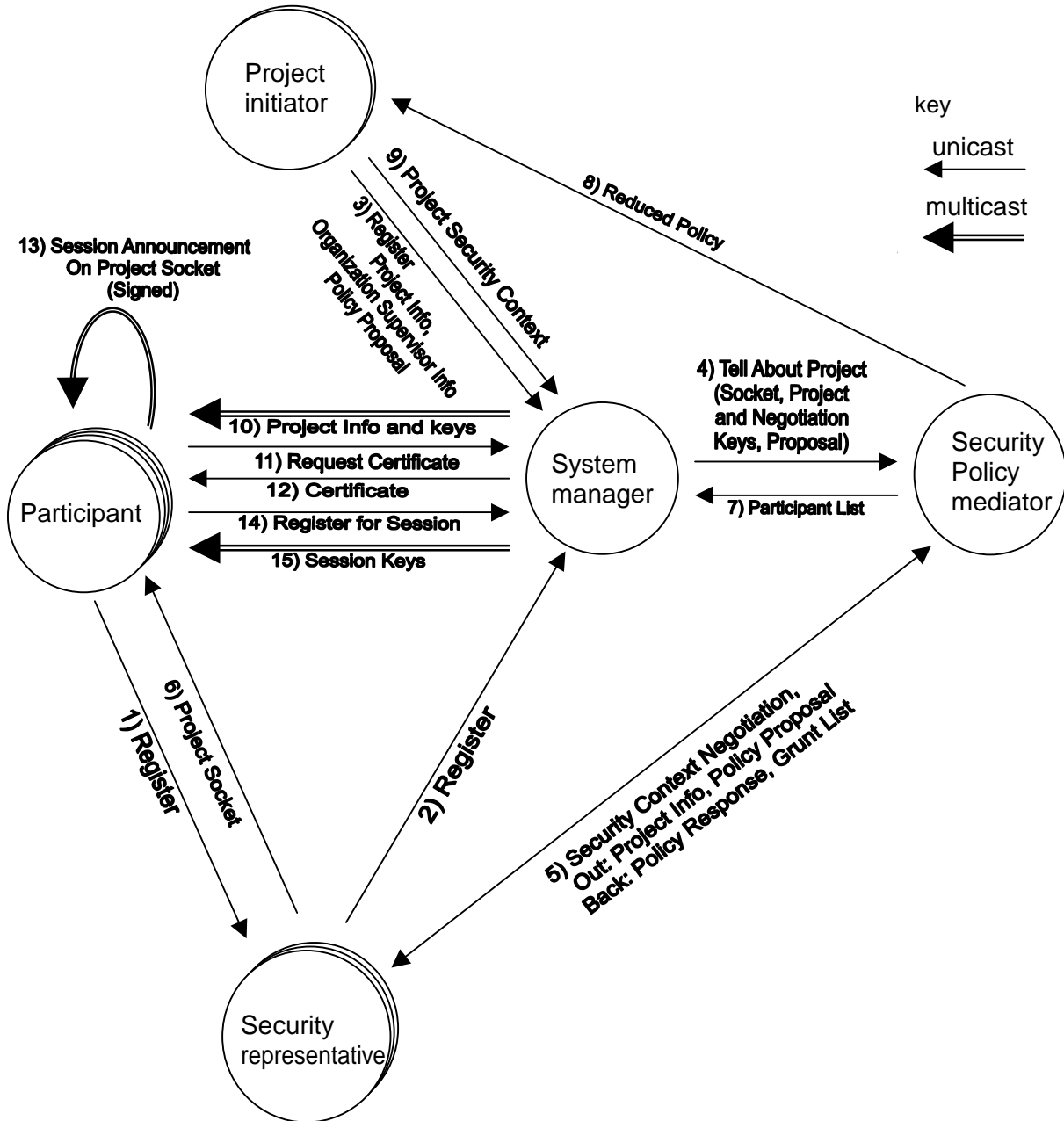
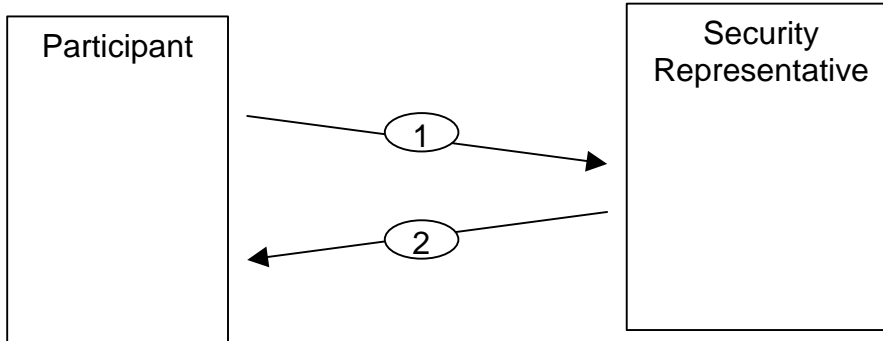


Figure 5: DCCM communication flow

5.2.3.1 *Participant Registration*
Starting a session from Scratch



- Participant opens up secure channel through IKE with security representative.
- Participant transfers participant secret to security representative.
- This repeats for as many participants as are in the organization.

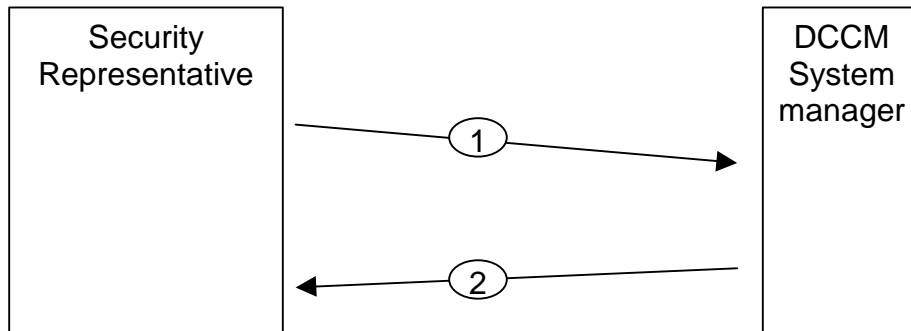
Rationale: Relatively high cost IKE channel only gets used once per participant. Future communications can use the pair-wise secret set up in this first exchange.

Message Contents:

1 { 32 bit magic – 0x4443434D
 32 bit message type
 32 bit message length in octets
 128 bit secret
 Null terminated friendly name
 Variable size authentication information

2 { 32 bit magic – 0x4443434D
 32 bit message type
 32 bit message length in octets
 32 bit return code

5.2.3.2 *Security representative Registration*
Starting a session from Scratch



- Security representative opens up secure channel through IKE with DCCM System manager.
- Security representative transfers secret and related information to DCCM System manager.
- This repeats for all security representatives.

Rationale: Relatively high cost IKE channel only gets used once per security representative.

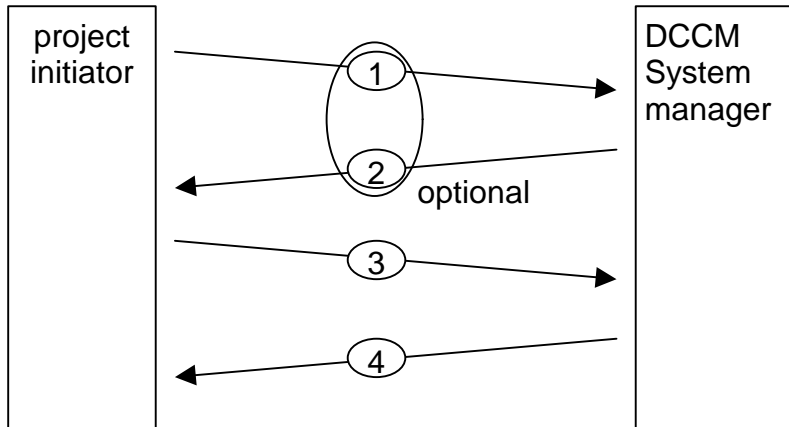
IKE is used to secure communications. The 128-bit secret is used in future communications for confidentiality and authentication. The security representative must remain active at the contact address. If the security representative is not there when the security policy mediator calls, then no participants that the security representative represents will get into the project.

Message Contents:

- 1 {
 - 32 bit magic – 0x4443434D
 - 32 bit message type
 - 32 bit message length in octets
 - 128 bit secret
 - Null terminated security mediation contact socket
 - Null terminated friendly name
 - Variable size authentication information

- 2 {
 - 32 bit magic – 0x4443434D
 - 32 bit message type
 - 32 bit message length in octets
 - 32 bit return code

5.2.3.3 *Start of project* *Starting a session from Scratch*



- The project initiator (a special security representative) opens up channel with DCCM System manager. The project initiator requests list of security representatives if it needs one. The project initiator requests the start of a project.
- This repeats every time a new project is registered.

Rationale: A key derived from the project initiator’s pair-wise secret with the DCCM System manager secures all communications. The secret comes from when the project initiator registered as a security representative. The project initiator must have permission to be a project initiator.

The system manager assures that all security representative friendly names are unique.

The “weight” value in (2) will be used if there is no common resolution between all participants. The “weight” times the number of participants represented will be used to pick the best result.

A failure in (1) will result in a return code message like (4).

Message Contents:

- ①

 - 32 bit magic – 0x4443434D
 - 32 bit message type
 - 32 bit message length in octets
 - 128 bit project initiator ID.
 - HMAC-SHA-1-96 of message

- ②

 - 32 bit magic – 0x4443434D
 - 32 bit message type
 - 32 bit message length in octets
 - 64 bit IV
 - Variable length Encrypted
 - 32 bit count of Security representatives
 - Variable Length for each Security representative
 - Null terminated Security representative Friendly Name
 - 32 bit weight
 - 32 bit authentication information length
 - Variable length authentication information
 - HMAC-SHA-1-96 of message

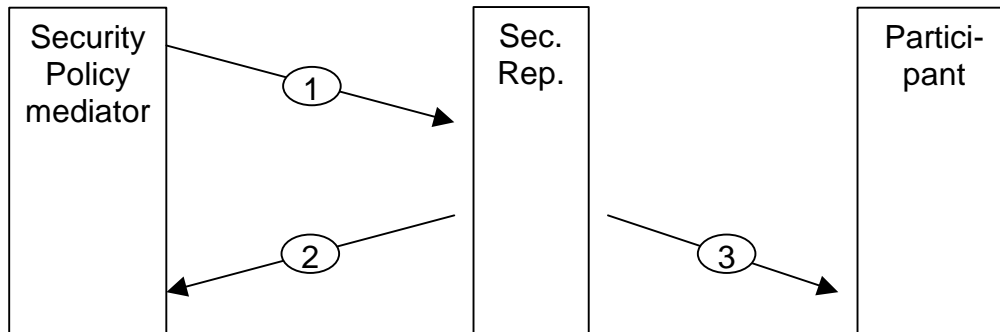
- ③

 - 32 bit magic – 0x4443434D
 - 32 bit message type
 - 32 bit message length in octets
 - 128 bit project initiator ID.
 - 64 bit IV
 - Variable length encrypted
 - 128 bit project ID
 - Null terminated project Friendly Name
 - Null terminated Security Policy Proposal
 - 32 bit count of Security representatives
 - Variable Length for each Security representative
 - Null terminated Security representative Friendly Name
 - HMAC-SHA-1-96 of message

- ④

 - 32 bit magic – 0x4443434D
 - 32 bit message type
 - 32 bit message length in octets
 - Encrypted 32 bit return code
 - HMAC-SHA-1-96 of message

5.2.3.4 Security Context Negotiation Starting a session from Scratch



- Security Policy mediator opens up a channel to the security representative. A policy proposal is sent and a response is returned with a list of participants for the project.
- After receiving a Security Policy from the Security Policy mediator, the security representative must agree to all or part of the policy. There are parts of the SPL that are ignored when they come from security representatives (such as the axis label and category information), so don't spend time figuring those out when creating the SPL for this message.

Rationale: A key derived from the security representative's pair-wise secret with the DCCM system manager secures communications (1) and (2). If the security representative is not found, then the participants represented by that security representative will not get in the project.

The "Session Description Protocol (SDP) Session Description" is a standard "x=<text>" format style definition for multicast announcements.

When the participant list is returned to the Security Policy mediator, they are encrypted in a key that the Security Policy mediator does not know. This is to keep the list of participants in the project from being known by more parties than is required.

Message Contents:

1

- 32 bit magic – 0x4443434D
- 32 bit message type
- 32 bit message length in octets
- 64 bit IV
- Variable Length Encrypted
 - 128 bit project ID
 - Null terminated friendly project name
 - Null terminated project Session Description Protocol (SPD) Session Description
 - Tells the socket, duration, etc for the project.
 - Null terminated Security Policy Proposal
- HMAC-SHA-1-96 of message

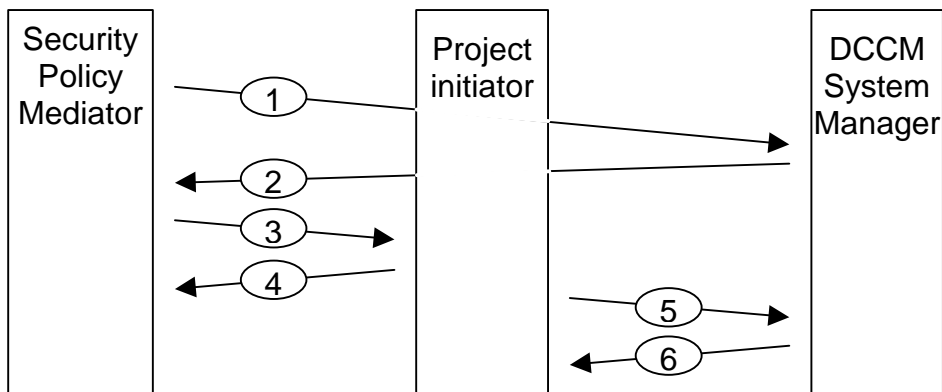
2

- 32 bit magic – 0x4443434D
- 32 bit message type
- 32 bit message length in octets
- 64 bit IV
- Variable Length Encrypted in negotiation key
 - Null terminated Security Policy Response
 - 32 bit count of Participants
- Variable length encrypted in Participant dump key (mediator does not know this key)
 - Repeated for number of Participants
 - 128 bit Participant ID
 - Null terminated Participant friendly name
 - Null terminated Participant Permissions
 - Variable Length Participant Authentication Information
- HMAC-SHA-1-96 of message

3

- 32 bit magic – 0x4443434D
- 32 bit message type
- 32 bit message length in octets
- Project information - Yet to be defined format but includes:
 - Null terminated project Session Description Protocol (SPD) Session Description - tells the socket, duration, etc for the project.

5.2.3.5 *Security Context Resolution*
Starting a session from Scratch



- Security Policy mediator opens a secure channel to the DCCM System manager and sends the project participants.
- Security Policy mediator opens up a channel to the project initiator. A reduction of the policy proposal is sent with a list of security representatives in the project.
- The project initiator then decides on a project security context and transmits that information to the DCCM System manager. Included in this transmission is a list of participants for the project that are under the project initiator.

Rationale: Relatively high cost exchange of participant list only done once per project (if at all – we may not separate the Security Policy mediator from the DCCM System manager).

Keys derived from the project initiator's pair-wise secret with the DCCM System manager secure communications with the project initiator. If the project initiator is not found, then the project will not get started.

Message Contents:

- 1 {
 - 32 bit magic – 0x4443434D
 - 32 bit message type
 - 32 bit message length in octets
 - 128 bit project ID
 - 32 bit count of Participant Lists
 - Repeat “count” times
 - 128 bit Security representative ID
 - Size of encrypted Participant List
 - 64 bit IV
 - Encrypted Participant List

- 2 {
 - 32 bit magic – 0x4443434D
 - 32 bit message type
 - 32 bit message length in octets
 - 32 bit return code

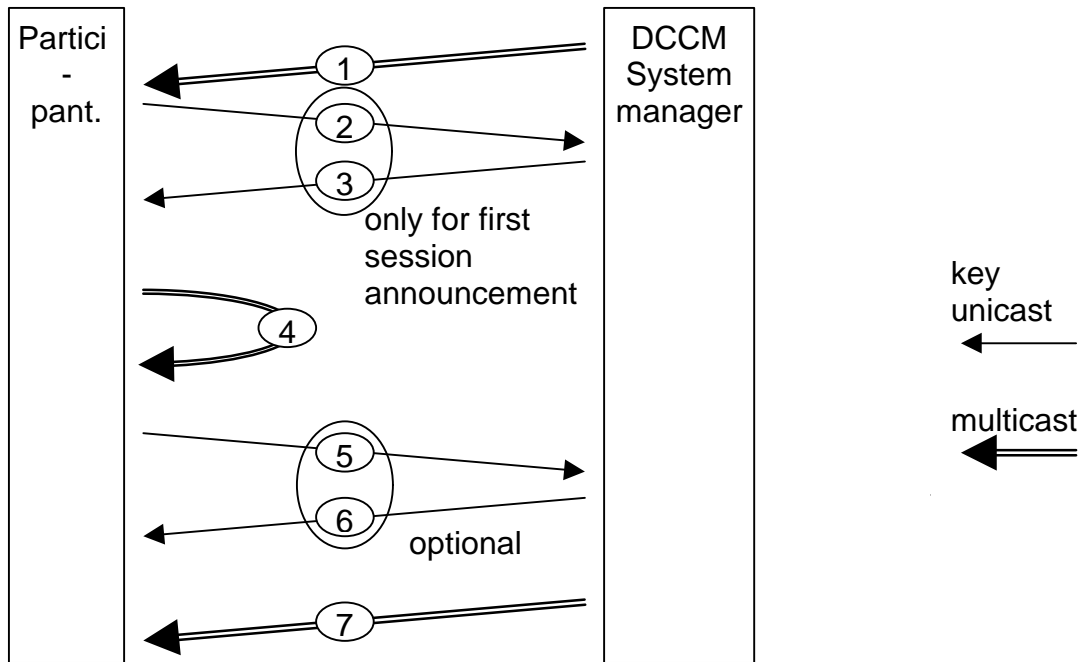
- 3 {
 - 32 bit magic – 0x4443434D
 - 32 bit message type
 - 32 bit message length in octets
 - 64 bit IV
 - Variable Length Encrypted
 - 128 bit project ID
 - Null terminated friendly project name
 - Null terminated Security Policy Reduction
 - 32 bit count of Security representatives
 - Repeat “count” times
 - Security representative Friendly name
 - HMAC-SHA-1-96 of message

- 4 {
 - 32 bit magic – 0x4443434D
 - 32 bit message type
 - 32 bit message length in octets
 - 64 bit IV
 - Encrypted
 - 32 bit Return code
 - HMAC-SHA-1-96 of message

- 5 { 32 bit magic – 0x4443434D
- 32 bit message type
- 32 bit message length in octets
- 64 bit IV
- Variable length Encrypted
 - 128 bit project ID
 - Null terminated Security context
 - 32 bit count of Participants
 - Repeated for number of Participants
 - 128 bit Participant ID
 - Null terminated Participant friendly name
 - Null terminated Participant Permissions
 - Variable Length Participant Authentication Information
- HMAC-SHA-1-96 of message

- 6 { 32 bit magic – 0x4443434D
- 32 bit message type
- 32 bit message length in octets
- 64 bit IV
- Encrypted
 - 32 bit Return code
- HMAC-SHA-1-96 of message

5.2.3.6 *Initial Keying*
Starting a session from scratch



- The DCCM System manager multicasts project keying information on the project socket.
- If needed, the participant session initiator requests a session announcement certificate from DCCM System manager. Using the session announcement certificate, the participant session initiator multicasts the session announcement.
- If participants must register (5) to be included in a session, they will open up a channel with the DCCM System manager and register for inclusion in the session.
- At some later time, the initial session keying information is multicast on the session socket.

Rationale: Multicast is lower cost in bytes transferred and communication setup then opening up individual channels.

Participant node number information is sent in the clear, but, since in general participants do not know other participants' IDs, it will be difficult to find arbitrarily which participant has which node number. This is not a high concern; this bit of indirection should be enough.

The DCCM System manager public key will arrive with the project information. This allows future verification of the session announcement.

To start a session, a participant needs to get a Session Starting Authorization Certificate from the DCCM System manager. The DCCM System manager will not give out a certificate unless the participant is allowed to start a session. Using the certificate, the

participant can send out a session announcement. This allows a participant to find out if it is allowed to start a session before setting up a session announcement. Once a participant has a certificate, they will be able to start multiple sessions without further contact with the DCCM system administrator.

The “64 bit Key Salt” is used to mix with the pair-wise secret to form the leaf key for the OFT. This allows the rapid re-keying of a whole group of participants.

The packets sent out over multicast must be limited to ensure they will fit in a datagram. IPv4 maximum datagram size is 64K, there are a few bytes of header that must fit in the datagram but the rest is available for application data. Any nodes in the Participant Node Number Information must have a corresponding entry in the “Key Data” section, otherwise it is invalid because there would be no MAC for that entry. Also, Participant’s Node Number Information will be transmitted before that participant’s node information so participants do not have to cache information they may not need.

The MACs in the keying messages will be done over the entire message up to and including the “32 bit key count” plus the 160 bits immediately prior to the MAC itself. The MAC will use a key derived from the same key the encryption key is derived from.

Message Contents:

32 bit magic – 0x4443434D
 32 bit message type
 32 bit message length in octets
 32 bit project ID
 32 bit Key ID
 Variable Length Participant Information
 32 bit count
 64 bit Key Salt/IV (if count > 0) (salts project Key and Participant’s pair-wise secret)
 (3232 bit for DSA?) Encrypted in key derived from project Key (if count > 0)
 DCCM System manager Public Key
 (160*count) bit node number info (128 bit ID | 32 bit node number)
 Variable Length Key Data
 32 bit key count
 (256*(key count)) bit key info
 32 bit node number | 128 bit encrypted key | HMAC-SHA-1-96

1

2 { 32 bit magic – 0x4443434D
 32 bit message type
 32 bit message length in octets
 128 bit Participant ID
 64 bit IV/Salt
 Encrypted in key derived from Participant Pair-wise Secret
 Public Key
 HMAC-SHA-1-96

3 { 32 bit magic – 0x4443434D
 32 bit message type
 32 bit message length in octets
 Certificate
 Signed
 32 bit magic – 0x4443434D
 32 bit certificate type
 32 bit length in octets
 32 bit project ID
 Public key
 Signature (by DCCM System Administrator)
 HMAC-SHA-1-96

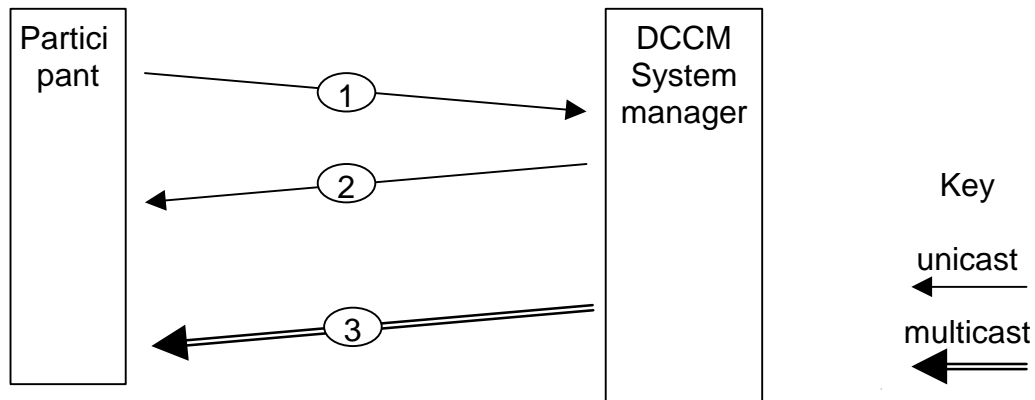
4 { 32 bit magic – 0x4443434D
 32 bit message type
 32 bit message length in octets
 Certificate
 Signed
 32 bit magic – 0x4443434D
 32 bit certificate type
 32 bit length in octets
 32 bit project ID
 Public key
 Signature (by DCCM System Administrator)
 64 bit IV
 Variable Length Encrypted in project Key
 Null Terminated DCCM Session Socket
 Null Terminated Session Description Protocol (SDP) Session Description
 Signature (by Session Initiator Participant)

5 { 32 bit magic – 0x4443434D
 32 bit message type
 32 bit message length in octets
 128 bit Participant ID
 64 bit IV/Salt
 Encrypted in key derived from Participant Key
 32 bit Session ID
 HMAC-SHA-1-96

6 { 32 bit magic – 0x4443434D
 32 bit message type
 32 bit message length in octets
 64 bit IV/Salt
 Encrypted in key derived from Participant Key
 Return code
 HMAC-SHA-1-96

7 { 32 bit magic – 0x4443434D
 32 bit message type
 32 bit message length in octets
 32 bit project ID
 32 bit Key ID
 Variable Length Participant Information
 32 bit count
 64 bit Key Salt/IV (if count > 0) (salts project Key and Participant’s pair-
 wise secret)
 (160*count) bit node number info (128 bit ID | 32 bit node number)
 Variable Length Key Data
 32 bit key count
 (256*(key count)) bit key info
 32 bit node number | 128 bit encrypted key | HMAC-SHA-1-96

5.2.4 Evicting a Participant from a Project/Session



In order to evict a participant, an authorized party opens up a socket to the DCCM System manager and sends an eviction request. DCCM System manager multicasts any key update material. This multicast may take many packets.

Rationale: The unicast transmissions are encrypted and authenticated with keys derived from the pair-wise secret.

The DCCM System manager can determine if the participant requesting the eviction has the authority to evict the other participant. If the 128 bit participant to evict ID is zero, the DCCM System manager will look to the friendly names for the participant to evict. The DCCM System manager assures no two security representatives have the same friendly name and the security representatives assure that no two participants under them have the same friendly name.

For the key update:

Multicast is lower cost in bytes transferred and communication setup than opening up individual channels to each participant. Participants do not know what other participant's IDs are so they cannot tell through this communication who else is in the session. The Root Key of the OFT set up here will be the key used to secure the session.

The "64 bit Key Salt" is used to mix with the participant key to form the participant leaf key for the OFT. The participant data is encrypted and MAC'ed with keys derived from the participant key to provide confidentiality and authentication.

The packets sent out over multicast must be limited to ensure they will fit in a datagram. IPv4 maximum datagram size is 64K, there are a few bytes of header that must fit in the datagram but the rest is available for application data. Any nodes in the

“Participant data” must have a corresponding entry in the “Key Data” section, otherwise it is invalid because there would be no MAC for that entry.

The MACs will be done over the entire message up to and including the “32 bit key count” plus the 160 bits immediately prior to the MAC itself. The MAC will use a key derived from the same key the encryption key is derived from.

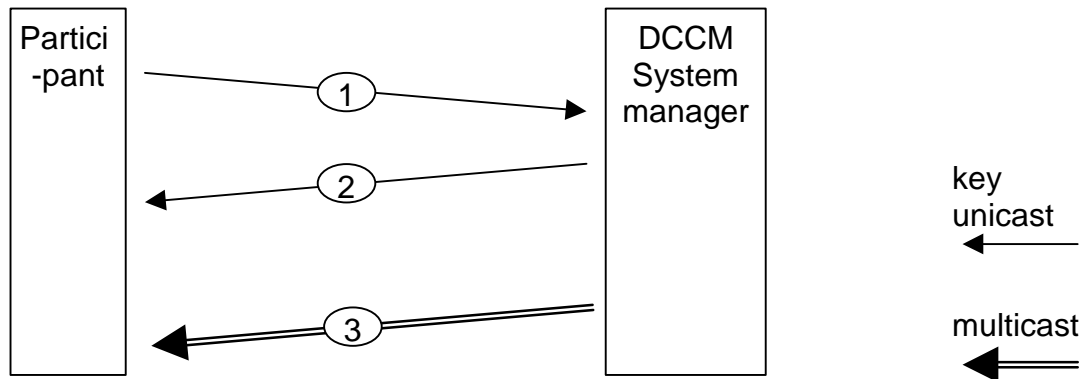
Message Contents:

- 32 bit magic – 0x4443434D
- 32 bit message type
- 32 bit message length in octets
- 32 bit project or Session ID
- 128 bit Participant ID
- 64 bit IV
- ① Variable Length Encrypted
 - 128 bit Participant to evict ID
 - Null Terminated Participant to evict Friendly Name
 - Null Terminated Security representative of Participant to evict Friendly Name
- HMAC-SHA-1-96

- 32 bit magic – 0x4443434D
- 32 bit message type
- 32 bit message length in octets
- 32 bit project or Session ID
- 64 bit IV
- Encrypted
 - 32 bit return code
- HMAC-SHA-1-96

- 32 bit magic – 0x4443434D
- 32 bit message type
- 32 bit message length in octets
- 32 bit project or Session ID
- 32 bit Key ID
- Variable Length Participant data
 - 32 bit count
 - 64 bit Key Salt (if count > 0)
 - (160*count) bit node number info (128 bit ID | 32 bit node number)
- Variable Length Key Data
 - 32 bit key count
 - (256*(key count)) bit key info
 - 32 bit node number | 128 bit encrypted key | HMAC-SHA-96

5.2.5 *Resynchronize*



A participant realizes or just thinks it does not have the proper key material. It opens a socket to the DCCM System manager and requests a re-sync. The DCCM System manager multicasts a key update which will have all the information that the participant needs to make the proper group key.

Rationale: The unicast transmissions are encrypted and authenticated with keys derived from the pair-wise secret. The re-sync key material is multicast because of the possible overlap of data from multiple requests within a short time frame.

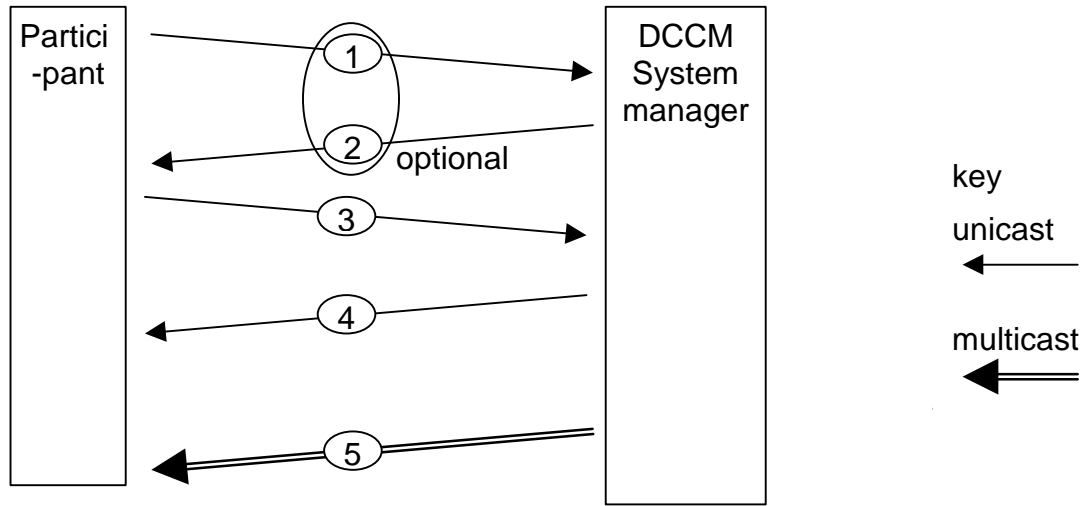
Message Contents:

- 1 {
 - 32 bit magic – 0x4443434D
 - 32 bit message type
 - 32 bit message length in octets
 - 32 bit project or Session ID
 - 128 bit Participant ID
 - HMAC-SHA-1-96

- 2 {
 - 32 bit magic – 0x4443434D
 - 32 bit message type
 - 32 bit message length in octets
 - 32 bit project or Session ID
 - 64 bit IV
 - Encrypted
 - 32 bit return code
 - HMAC-SHA-1-96

③ See Key Update section (3) in eviction.

5.2.6 *Joining a Session*



A Participant decides to enter a session already in progress. It opens a socket to the DCCM System manager. If it needs session information, it will request it. If and when it knows what session to join, it requests to join. The DCCM System manager multicasts a key update shortly thereafter which will have all the information that the participant needs to make the proper group key.

Rationale: The unicast transmissions are encrypted and authenticated with keys derived from the pair-wise secret. The re-sync key material is multicast because of the possible overlap of data from multiple requests within a short time frame.

Message Contents:

- 32 bit magic – 0x4443434D
- 32 bit message type
- 32 bit message length in octets
- 128 bit Participant ID
- 64 bit IV/Salt
- Encrypted in key derived from Participant Key
 - 32 bit project ID
- HMAC-SHA-1-96

- 2
- 32 bit magic – 0x4443434D
 - 32 bit message type
 - 32 bit message length in octets
 - 128 bit Participant ID
 - 64 bit IV/Salt
 - Variable Length Encrypted in key derived from Participant Key
 - 32 bit count of sessions
 - Variable length repeated “count” times
 - Null Terminated Session Description Protocol (SDP) Session Description
 - HMAC-SHA-1-96
- 3
- 32 bit magic – 0x4443434D
 - 32 bit message type
 - 32 bit message length in octets
 - 128 bit Participant ID
 - 64 bit IV/Salt
 - Encrypted in key derived from Participant Key
 - 32 bit Session ID
 - HMAC-SHA-1-96
- 4
- 32 bit magic – 0x4443434D
 - 32 bit message type
 - 32 bit message length in octets
 - 32 bit project or Session ID
 - 64 bit IV
 - Encrypted
 - 32 bit return code
 - HMAC-SHA-1-96
- 5
- See Key Update section (3) in eviction.

5.2.7 Key Derivation

Keys are derived from the shared secret. The secret is concatenated to a mixing value and then sent through SHA-1. The appropriate number of bits is then copied from the leading bits of the resulting digest. If the resulting digest is too short for the desired key then this digest is expanded to two digests. The digest is XORed into two buffers. The first buffer is a 64 byte long buffer where each byte has the value of 0x36. The second buffer is also 64 bytes long and each byte has a value of 0x5C. These two resulting buffers are then digested and the resulting digests are concatenated. From this double digest, the appropriate number of leading bits is taken for the key.

6. Next Steps

This document has the protocols for policy and context negotiation. This section outlines several of the steps that may be taken in enhancing the negotiation protocol.

The CCNP could be enhanced in several dimensions. Additional specifications for alternative communications, security environments, and organizations can be added to the specifications. For example, existing secure connectivity among subgroups of participants within an organization (e.g., protected private domains within physically secure and electronically isolated buildings) may affect the cryptographic protection negotiated in the policy and context. While all combinations of such factors are not specified in the existing syntax, how such extensions can be made may be addressed in the future. Precedence of policies or priorities of policy negotiators might be added. Levels of security at various granularities may be added. Semantics for the entire supported syntax must be created and encoded within an expert mapping program that translates cryptographic policies into their equivalent cryptographic contexts.

Acknowledgments

Dr. Dennis K. Branstad managed the Cryptographic Technologies Group until Trusted Information Systems, Inc. was acquired by Network Associates, Inc. Subsequently, he has joined the group as a part-time employee to assist in preparing reports and technical research proposals. We thank Dr. Branstad for his initiating the DCCM project and subsequent contributions to both the overall direction and technical foundations of the project. He has played a major role in defining the architecture and components of the DCCM system.

Mr. Peter Dinsmore joined the Cryptographic Technologies Group in TIS Labs and the DCCM project in February, 1999. We thank Mr. Dinsmore for helping to edit and prepare Report Three.

References

- [Bal98] Balenson, David M., Dennis K. Branstad, David A. McGrew, Alan T. Sherman, *Dynamic Cryptographic Context Management (DCCM): Report 1: Architecture and System Design*, TIS Labs at Network Associates, Inc., TISR #0709, June 2, 1998.
- [Bal98a] Balenson, David M., Dennis K. Branstad, David A. McGrew, Jay W. Turner, *Dynamic Cryptographic Context Management (DCCM): Report 2: Cryptographic Context Negotiation Template*, TIS Labs at Network Associates, Inc., TISR #0745, September 18, 1998.
- [Bal98b] Balenson, David M., Dennis K. Branstad, David A. McGrew, Jay W. Turner, Michael Heyman, *Dynamic Cryptographic Context Management (DCCM): Report 2, Version 2: Cryptographic Context Negotiation Template*, TIS Labs at Network Associates, Inc., TISR #0745-2, February 24, 1999.