

PROJECT PROFILE

Dynamic Cryptographic Context Management

Introduction

The Dynamic Cryptographic Context Management (DCCM) project efficiently provides security for very large, dynamically changing groups of participants. For example, command and control of tactical military forces requires several types of protection among a very large group of participants, perhaps from different countries or from different Armed Forces units, grouped together under one command for some time period or for a specific exercise. By “large,” we mean groups with number of members ranging from 10,000 to 100,000 or more. By “dynamic,” we mean new members may be added to the group at any time and existing members may be evicted from the group, thereby requiring immediate changes to some of the security provisions. Members need not be humans; they can be a variety of communicating entities, including sensors, mobile client workstations, server workstations, or network nodes.

The DCCM system has two novel distinguishing characteristics. First, policy plays a key role in DCCM. Groups at all levels have policies. These policies are represented; they are negotiated; they are managed; and a *cryptographic context*—an unambiguous set of mechanisms and configuration—is created to make particular interactions possible subject to these policies. Second, DCCM has a scalable key management system that can handle group sizes up to 100,000 members and can dynamically handle members entering and leaving groups.

Policy

A security *policy* is a set of rules specifying how to protect information. Policies can be described by *whom* they cover and by *what* they cover. In DCCM, every organizational entity has a policy. When a participant joins the DCCM system, they bring with them their security policy for the protection of their information. It may be a policy specific to them, such as in a flat group model, or the policy may be an organization policy covering all of the members of some hierarchy.

A policy that contains a range, or set of allowable actions, cannot be enforced by multiple participants with any expectation of interoperability. Interoperability can only be achieved when it can be guaranteed that all of the participants will enforce the policy exactly the same. DCCM accomplishes interoperability by distributing a policy that is completely unambiguous; there are no ranges or options. A policy that specifies a singular instantiation in DCCM is referred to as a *context*, specifically a *cryptographic context*.

A project context is created through a negotiation protocol during project creation in the DCCM system. The Cryptographic Context Negotiation Protocol (CCNP) creates a fully specified cryptographic context for the project that fulfills the individual policies of the group participants.

Research Focus

Two Distinguishing Characteristics

The DCCM system has two novel distinguishing characteristics:

- First, policy plays a key role in DCCM. Groups at all levels have policies. These policies are represented; they are negotiated; they are managed; and a *cryptographic context*—an unambiguous set of mechanisms and configuration—is created to make particular interactions possible subject to these policies.
- Second, DCCM has a scalable key management system that can handle group sizes up to 100,000 members and can dynamically handle members entering and leaving groups.

- **Pete Dinsmore, Manager,
Cryptographic Technologies
Group**

Policy (continued)

Viewed graphically in Figure 1, cryptographic context negotiation finds the intersection between the allowable policies of the project participants.

Key Management

The DCCM project developed a new hierarchical method for large dynamic group keying based on the novel application of One-way Function Trees (OFTs). The OFT method is scalable and efficiently changes the key when group members are added or evicted. The OFT method represents group members as the leaves and the group key as the root of a logical tree. Rather than “pushing” the group key down the tree, the OFT method “pulls” the group key up the tree, using one-way functions. As shown in Figure 2, an OFT is a binary tree, each node x of which is associated with two cryptographic keys: a node key k_x and a blinded node key $k'_x = g(k_x)$. The blinded node key is computed from the node key using a one-way function g ; it is blinded in the sense that a computationally limited adversary cannot find k_x from k'_x .

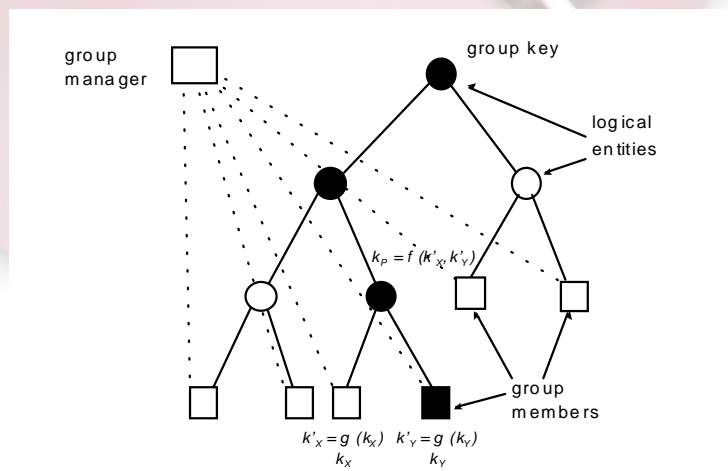


Figure 2: A one-way function tree

where x and y denote the left and right child of the node p , respectively. The function g is one-way, and can be based on a cryptographic hash function such as MD5 or SHA-1. The function f does not need to be one-way; it needs to mix its inputs—the bitwise exclusive-or function is a fast, simple, and effective choice. The node key associated with the root of the tree is the group key, which the group can use to communicate with privacy among group members and/or authentication of group membership.

Results

The DCCM architecture can handle up to 100,000 members. A demonstration implementation written in Java successfully negotiated a cryptographic context, created a group key, and then managed the group. The system removed a single member from a group of 100,000 members in 1 second utilizing a multicast message of 1105 bytes.

Additional Information

For additional information on the DCCM project, contact Peter Dinsmore (Peter_Dinsmore@nai.com) at 443-259-2346 or visit our Web page at: <http://www.pgp.com/research/nailabs/cryptographic.asp>.

1/5/01

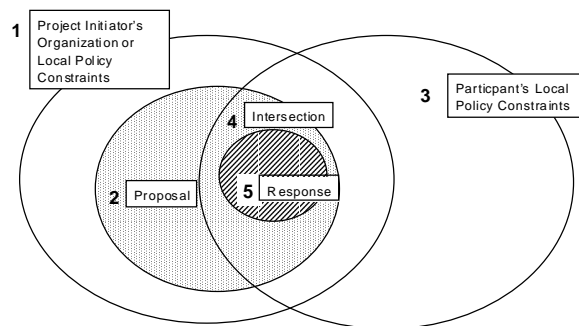


Figure 1: Cryptographic context management

A group manager maintains a one-way function tree. Each leaf is associated with a member of the group. The manager uses a symmetric encryption function E to communicate securely with subsets of group members, using unblinded keys as encryption keys as explained in Figure 2.

A randomly-chosen key is assigned to each member. This key is shared with the manager (via an external secure channel), and the key is assigned as the node key of the member's leaf. A variety of choices are possible governing who chooses the keys. In particular, the key could be chosen by the manager, member, or a combination thereof. OFT can be implemented in a manner where every member contributes towards the group key.

Each internal node p of the tree has exactly two children. The interior node keys are defined by the rule:

$$k_p = f(g(k_x), g(k_y)), \quad (1)$$



Who's watching your network

For more information on NAI Labs, contact your authorized NAI Sales Representative, or visit <http://www.nailabs.com>.

CORPORATE Headquarters

3965 Freedom Circle
Santa Clara, CA 95054-1203
Tel (800) 764-3337*
Fax (888) 203-9258

*Call for additional Worldwide Sales Offices