Secure and Reliable Group Communication

In response to Next Generation Internet -
Research in Basic Technologies - Program Notice LAB 99-08

Lawrence Berkeley National Laboratory

Principal Investigator
Mary R. Thompson
Computer Engineer III
Lawrence Berkeley National Laboratory
1 Cyclotron Rd., MS-50B-2239, Berkeley, CA 94720
(510) 486-7408   Fax: (510) 486-6636
MRThompson@lbl.gov

____

Official Signing for the Lab
Stewart Loken
Director, Information and Computing Sciences Division
(510) 486-7474 Fax: (510) 486-4300
SCLoken@lbl.gov

____

Submitted in conjunction with the "Secure and Reliable Group Communication" proposal submitted by USC/ISI

Funding Requested:
Year 1: $248,840
Year 2: $252,938
Year 3: $262.773

Use of human subjects - No

Table of Contents

Abstract

Many next-generation Internet applications will depend on communication between groups of users. Some prominent examples are: shared visualization and virtual reality as well as collaborative remote control of instruments. If group members are not mutually authenticated, and if the communication among them is not secure, many researchers will be unwilling to use these applications. Current secure protocols, such as SSL and IPSEC, work only between pairs of users.

The main goal of this proposal is to provide an integrated solution for **secure** collaborative applications, i.e., a comprehensive middleware component that includes reliable and secure group communication complete with access control and authorization (policy) services. We propose to achieve this goal by integrating, enhancing and extending, three existing technologies: Akenti distributed access control server; CLIQUES group security services toolkit; and Totem, a reliable, fault-tolerant group communication protocol.

In addition to our main goal, there are several fundamental research issues in group security that this proposal aims to address. Dynamic groups present interesting additional policy considerations that traditional static resources do not. New policy constraints might include: altering the group membership; policy contingent on the current membership; time based constraints. Group certification is also an open research issue, including issues such as: how to issue, manage and revoke certificates for constantly changing groups. Finally there is the need to extend group security to two-tiered groups composed of a small number of senders and a comparatively large number of receivers. We will consider scalable (efficient) ways to extend the security to the larger group.

The research and integration activities outlines below are proposed as a **collaborative** effort between the University of Southern California (Information Sciences Institute) and the Computing Sciences Department of the Lawrence Berkeley National Laboratory (LBNL). To this end, a complementary proposal is being submitted by ISI.

1 Background and Significance

Many next-generation Internet applications depend on communication between groups of users. If members of the group are not authenticated to each other, and if the communication links cannot be made secure, most users will be unwilling to use these applications. Current secure communication protocols, such as SSL and IPSEC, work on a peer-to-peer basis, i.e., between two endpoints. The main goal of this proposal is to provide an integrated solution for **secure** collaborative applications, i.e., a comprehensive middleware component that includes reliable and secure group communication complete with access control and authorization (policy) services.

To achieve the above we propose to integrate, and -- in the process -- extend and enhance, three major components:

- Akenti distributed access control services to perform policy enforcement with respect to group membership (e.g., when users attempt to join a group)
- CLIQUES group security toolkit that includes contributory group key agreement protocols, member authentication and other peer group-oriented security services.
- Totem group communication toolkit to provide fault-tolerant and reliable group communication platform

A coarse overview of the proposed architecture is illustrated in Figure 1. Although the integration and enhancement of existing components is a major theme of this proposal, the other, no less important, direction is the investigation of more fundamental, long-term research issues outlined in Section 3 below.

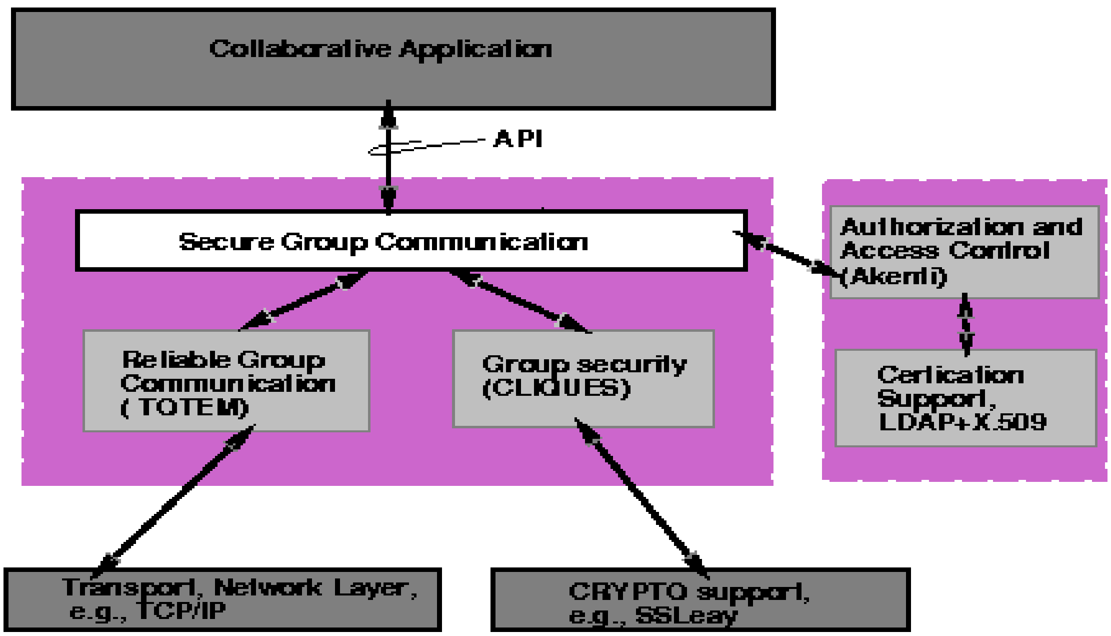

Figure 1. Overview of Proposed Architecture

The rest of this proposal is organized as follows. The next section presents brief overviews of the technologies we propose to integrate. Section 3 discusses the fundamental research challenges we plan to explore. Then, Section 4 sketches out the approach (both short- and long-term) to achieving the proposal goals and, Section 5 concludes the narrative part of the proposal with the schedule of anticipated milestones and division of tasks among the collaborators.

2 Technology Background (Preliminary Studies)

This section presents the summaries of the three technology components that will serve as building blocks for the proposed research.

2.1 CLIQUES Overview

In many collaborative applications, underlying peer groups can collectively provide critical services and safeguard important secrets. In this setting, the ability to join a group, leave a group or in any way alter group membership must be tightly controlled. Moreover, group members must have the means to communicate securely, in a private and authentic manner.

CLIQUES[1] is a research project at USC/ISI Computer Networks Division sponsored by DARPA ITO. It is also a name of the protocol suite for decentralized contributory key agreement and associated cryptographic services in a dynamic peer group environment. CLIQUES[16] addresses the issues of initial key agreement among a peer group of entities (e.g. hosts, users, processes); the subsequent adding or removal of members from the group; the subsequent partitions and merges of sub-groups of members; and the issue of key refreshment.

CLIQUES emphasizes the following properties:

- Provable Security -- protocols are formally demonstrated to be secure with respect to their goals (see [12],[3],[13]).
- Fairness -- all protocol participants are treated equally; no one party can avail itself of greater security than any other party, i.e., security is attained cooperatively.
- Efficiency -- protocol costs are minimized (with respect to the desired security guarantees).
- Adaptability and Fault tolerance -- protocols are able to adapt to dynamic factors, e.g., mobility as well as failure of protocol participants.
- Self-Containment -- protocols do not require any extraneous encryption, authentication or other support services.
- Generality -- protocols are applicable to any communication scenario; no specific network topology or communication paradigm is assumed or favored.
- Simplicity -- simplicity of design results in smaller, cleaner implementation makes security much easier to demonstrate.

Status During the initial stage of the CLIQUES project, the basic protocols were specified, analyzed and tested. Subsequently, the protocols were implemented in JAVA and, later, in C. The initial implementation was strictly a "proof-of-concept" used for testing and demonstration purposes. More recently, the implementation was revamped in the guise of CLQ_API, a modular interface to CLIQUES key management services. Testing and initial integration of CLQ_API are currently underway.

CLQ_API [2] implements the basic group key agreement and auxiliary services: key adjustments and updates due to membership changes. It also includes member authentication and non-repudiation service whereby a member "proves" knowledge of its contribution to the group key.[2] CLQ_API also includes a generalized authentication (MAC) service as well as a sample

1. Project web page at: *http://www.isi.edu/div7/cliques*
2. This is realized using a well-known Schnorr digital signature method.

encryption function. (Due to export controls and the diverse privacy requirements of anticipated applications, the choice of encryption is left to the application.)

2.2 Akenti Overview

Policies governing resource access traditionally have been expressed via access control lists (ACLs). The ACL model assumes centralized control: only one “owner” administers and enforces the access control requirements. This approach evolved in the context of central data repositories such as file systems or databases whose contents are used by members of a single administrative domain. However, in distributed computing environments such as envisioned by many of the proposed NGI applications, shared resources may physically span several administrative domains, may be subject to control by several stakeholders and may be accessed by clients from various domains.

To change the access control requirements in a centralized system, the access control enforcer must verify that the change request originated from an authorized party, check that the request was not altered in transit, and only then make the appropriate change. Such centralized and essentially manual updating does not scale well, particularly if the parties are geographically or organizationally dispersed. The user IDs in ACL's are typically expressed in terms that are meaningful only to the access control enforcer, thus requiring out-of-domain users to get a separate ID to be used for referencing this set of resources.

Akenti[14][3] is an access control system designed by the Imaging and Distributed Collaboration Group at LBNL to support the kind of distributed environments found in the DOE2000 Collaboratories and NGI applications [7],[15]. Its goal is to support the same sort of distributed authority over resources that is used in non-computer group endeavors. Akenti addresses the issues that stem from access to resources controlled by multiple owners (stakeholders). Each responsible entity, i.e., stakeholder, should be able to make its assertions without reference to a mediator, and especially without reference to a centralized system administrator who must act on its behalf.

As shown in Figure 2, entity authentication is done via X.509 identity certificates issued by trusted certification authorities. Authorization is granted (or denied) by comparing the authenticated identity to a distributed collection of signed use-condition and attribute certificates that define the access policy of a resource and the corresponding user attributes. Since Akenti is designed to support administratively and geographically distributed stakeholders and resources, the access policy information can be stored and maintained separately from the server that controls access to a resource. Akenti aims to achieve the same level of expressiveness of access control that would be available if a human controller were in the decision loop, and to reflect the existing policy (authority, delegation, and responsibility) present in these environments. More specifically, Akenti's goals are:

- allow each stakeholder to enforce its access control requirements independently of the other stakeholders,

3. Project web page at: *http://www-itg.lbl.gov/Akenti*

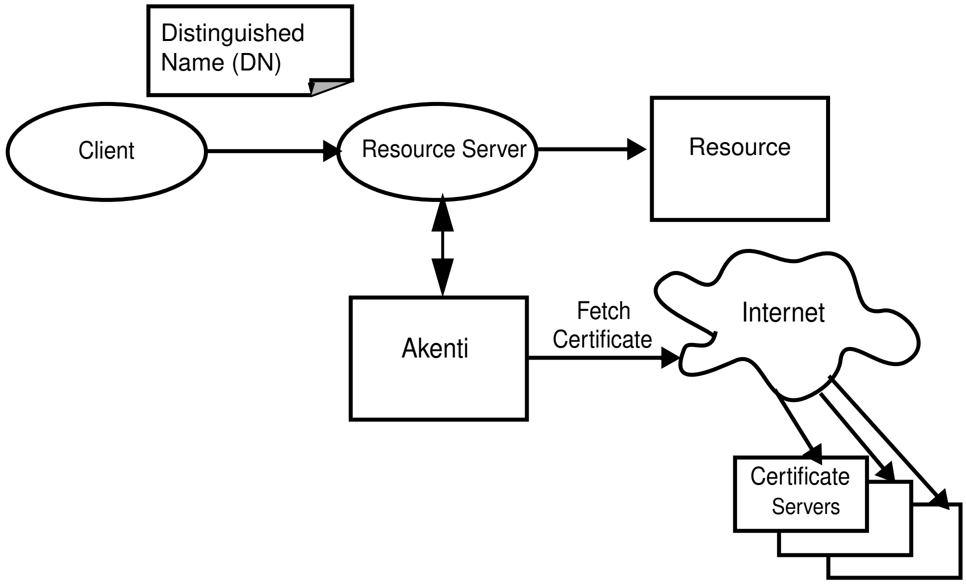

Figure 2. Akenti Architecture

- allow each stakeholder to change its requirements at any time and to be confident that the new requirements would take effect immediately,
- provide high assurance of integrity and non-repudiation in the expression of the access control requirements.

The resource that Akenti controls may be information, processing or communication capabilities, or a physical system such as a scientific instrument. Access can be the ability to obtain information from the resource (e.g., "read" access;), to modify the resource (e.g., "write" access), or to cause that resource to perform certain functions (e.g., changing instrument control set points).

Status The Akenti policy engine has been implemented as C++ library module and a Java class (using Java Native Interface). The C++ implementation has been in use for about a year now in several different applications. The group at LBNL has had considerable experience in setting up the Public Key infrastructure necessary to support Akenti policy control.

2.3 Totem Overview

The Totem system [1],[9] is a set of communication protocols that aid the construction of fault-tolerant distributed systems. It has been developed at the University of California at Santa Barbara. The message ordering mechanisms provided by Totem allow an application to maintain the consistency of distributed and replicated information in the presence of faults. The salient features of the Totem system are:

- Ordered multicasts to process groups

- High throughput and low predictable latency
- Highly portable code through the use of standard Unix features
- Rapid reconfiguration to remove failed processors, add repaired and new processors, and remerge a partitioned system
- Continued operation of all parts of a partitioned system to the greatest possible extent that is consistent with correctness.

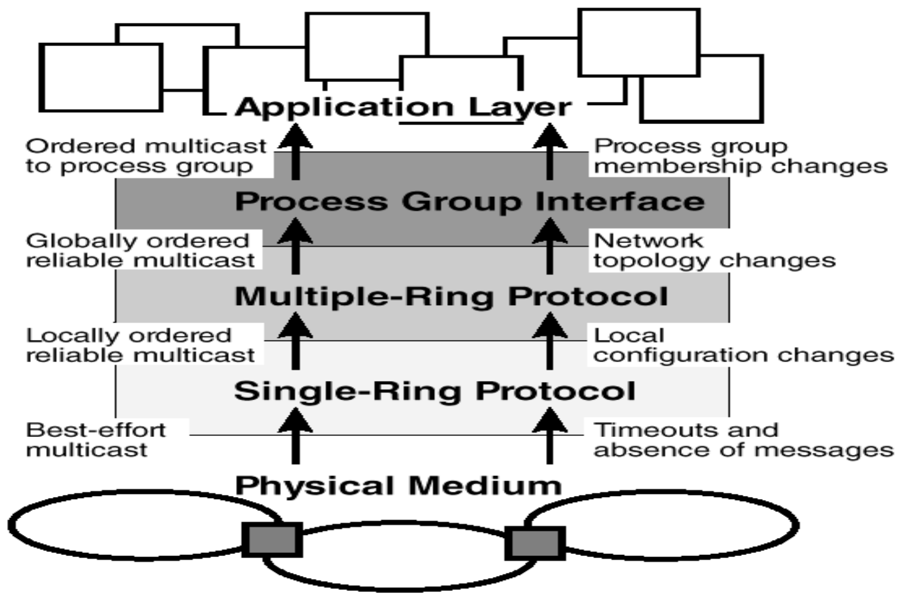

Figure 3. Totem Architecture

The high-level architecture of Totem is illustrated in Figure 3 [9]. Totem provides reliable totally ordered delivery of messages to processes within process groups on a single local-area network, or over multiple local-area networks interconnected by WAN gateways. Superimposed on each local-area network is a logical token-passing ring. The fields of the circulating token provide reliable delivery and total ordering of messages, confirmation that messages have been received by all processors, flow control, and fault detection. Message ordering is consistent across the entire network, despite processor and communication faults, without requiring all processes to deliver all messages.

The Totem system handles processor failure and recovery, and also network partitioning and remerging, and provides membership and topology maintenance services. Application processes receive, in agreed order, all messages timestamped within the current process group configuration and broadcast to the group by a member of the group. An optional higher level of message delivery, safe order, allows a process to know that every other process in its group has also received the message.

Status Totem is a mature system having been in experimental and limited production deployment for a number of years. The code is freely available and ported to most popular hardware/OS platforms. In addition, the DOE2000 Collaboratory Interoperability Framework (CIF) currently supports Totem for reliable multicast. So it is likely that a secure group protocol based on Totem would be incorporated into CIF.

3 Research Challenges

As mentioned above, our main goal in this proposal is to develop a comprehensive and integrated security and communication platform for the next generation dynamic collaborative (i.e., peer group-oriented) applications. In this section we present several fundamental research issues that this proposal aims to address.

3.1 Integrating Security Mechanisms with Reliable Group Communication

Many different reliable group communication platforms have been proposed and built in the past decade. LOCUS, ISIS, TRANSIS, Totem, HORUS, and SPREAD are just a few examples. Most emphasize fault tolerance and reliability while security is often done as an afterthought. The state-of-the-art in reliable group communication security is exemplified by the ENSEMBLE system recently developed at Cornell [6]. Although ENSEMBLE provides strong security guarantees [10], it utilizes ad hoc extensions of conventional (non group-oriented) security tools, namely PGP [17], to distribute and refresh group keys. The main drawbacks of this approach are:

- Generation of group keys is centralized (done by group leader) which goes contrary to peer group model. In particular, the entire group relies on one member's ability to generate strong, secure keys.
- Key distribution is performed with the help of encryption (via PGP) which is expensive since the key must be encrypted individually for each member. Moreover, this method is insecure since compromise of just one member's long-term secret exposes all previous group communication. (A more secure approach would avoid this problem by providing *Perfect Forward Secrecy*, i.e., it would preclude explicit distribution of keys encrypted under long-term secrets.)

In contrast, we intend to approach security in a more systematic, group-oriented manner. In choosing CLIQUES as the stepping stone for the proposed security architecture, we start out with key agreement protocols expressly developed for the peer group model. In CLIQUES, group keys are generated jointly by all group members with each member contributing an equal share to the common key. This affords us a number of useful security properties and services:

- CLIQUES key agreement is self-contained; it requires no additional encryption or authentication services. The security is based on one fundamental principle: the security of the well-known Diffie-Hellman problem. (This is because CLIQUES protocols extend the 2-party Diffie-Hellman protocol to groups and the security of CLIQUES protocols has been formally shown to be equivalent to that of Diffie-Hellman [12],[13].)
- CLIQUES provides both *Perfect Forward Secrecy* (PFS) and resistance to
- *Known-Key Attacks* (attacks by very powerful adversaries).

- CLIQUES allows for efficient authentication (and signature capability) of an individual group member based on its private share of the common group key.
- Unlike all current mechanisms, CLIQUES provides membership authentication, a property that results in members agreeing on the same group key **only if** their view of the membership is the same.

However, we note that CLIQUES provides only a security layer. There is a mutual reliance between CLIQUES and the underlying group communication platform. The latter needs CLIQUES to satisfy its security requirements while CLIQUES, by design, assumes a reliable group communication (multicast) channel.

3.2 Policy in a Group Context

Current access control mechanisms treat groups only as a convenient way of aggregating subjects (users) so that access control lists can be kept short. We observe that -- in the context of dynamic collaborative applications -- groups must be also treated as **objects** of access control. In other words, all operations changing the composition (membership) of a group must be handled similar to different operations on other, more traditional access control objects such as files. At the same time, reliable group communication protocols tend to treat membership in an ad hoc manner and most (e.g., ENSEMBLE) leave authorization entirely up to the application. There are no general mechanisms to control admission to, and, eviction from, a group.

In its present form, CLIQUES does not have any way of controlling access to group membership, i.e. there is no mechanism to decide who should be allowed to join a group and, hence, participate in key agreement and other security operations. There is also no unified strategy for handling and processing credentials such as public key certificates of group members. CLIQUES assumes a certification infrastructure but it must be actually integrated with one to provide a truly useful service.

On the other hand, an authorization and access control system such as Akenti has no means of supporting collaborative applications. It -- like others, e.g., Kerberos -- has been originally designed to provide a service in a client/server or peer-to-peer environment. In the case of securing group communication, the resource to be controlled is the membership in a uniquely named group. While access control policies for traditional resources (files, printers, workstations, instruments) are well-understood, it is still unclear what policies can apply to group membership and how to express and enforce them.

In the long run, we envisage that one or more core group members would be chosen as stakeholders (resource owners) for the group and allowed to issue admission certificates authorizing other entities to join the group. The actual entity that makes the decision to permit a request to join, can be yet another party, as long as it checks with a trusted Akenti server before allowing any prospective members to join.

In summary, one of the main research challenges is understanding (and subsequently expressing) access control semantics as well as authorization rules and policies that apply to dynamic collaborative groups.

3.3 Group Certification

The problems of issuance and maintenance of public key certificates have been well-researched in the last two decades. As a result, mechanisms, such as X.509 and PGP, are enjoying wide use. However, if we view a tight, collaborative group as an aggregated entity (a composite user of sorts), it is only natural to ask how this entity can be certified. Before answering we first need to consider the reasons for wanting to certify a group:

- A group may be a relatively long-lived entity. For example, a group of experts (in a certain field) might form a long-term group. Anyone wishing to get an expert opinion can ask the group-at-large by sending a query encrypted with the group's public key. This public key must be obtainable from some type of a certificate.
- Outsiders (non-members) may need to communicate with the group or some of its members. In doing so, they need to be assured that communication really emanates from a *bona fide* group member. In other words, group member(s) may need to digitally sign certain communication and this also requires the availability of a group public key certificate.

Group certification is a new research topic; it is not investigated in the contemporary research literature. In trying to address it, we anticipate having to contend with some difficult issues, e.g.:

- Implications of dynamic membership -- Recall that individual, traditional certificates are issued by an off-line Certification Authority (CA). If members frequently join and leave a group, and, consequently, group keys change, how can group certificates be issued efficiently? Who should issue them?
- Individual vs. opaque certification and authentication -- Should membership certificates be issued to every group member individually thus enabling finely-granular group credentials and individual member authentication? Or, should certificates be issued *opaquely* to the entire group thus allowing for the anonymity of individual members?

3.4 Two-Tiered Groups

Many group applications and settings are two-tiered in nature: they involve a small number of senders and a comparatively large number of receivers. While the senders can be viewed as a collaborative peer group, the entire group is not collaborative. The challenge in this environment is twofold:

- How to provide *scalable* (efficient) and reliable group communication with stronger guarantees for senders and weaker for receivers?
- How to preserve security in a way that treats senders and receivers as different types of security principals? (For example, only senders can participate in key generation while receivers only obtain a fraction of the group key.)

To address the former, new lighter-weight group communication protocols are needed. One approach to scalable and efficient group protocols is being proposed by D. Agarwal (LBNL) in the context of collaborative visualization applications. Another relevant new development is the InterGroup protocol proposed by K. Berket from UCSB [4]. InterGroup targets precisely those multicast groups where receivers greatly outnumber senders while reliable delivery is nonetheless required.

The second challenge is more difficult: one possible solution is to extend CLIQUES protocols to allow key agreement only among senders while offering key distribution for the larger group of receivers. The main difficulty lies not in the actual mechanisms but rather in the associated policy issues. For example, different actions (key changes) might be appropriate if the population of senders changes as opposed to that of receivers.

4 Research Design and Methods

We plan to achieve our main goal of developing a comprehensive secure group communication solution in incremental steps. First, we will integrate CLIQUES group key management toolkit with Totem reliable group protocol. Second, we will establish a trusted authority server to name, store policy and make access decisions. Third, we will integrate calls to this trusted server into the CLIQUES/Totem group protocol to provide authorization of membership requests.

4.1 CLIQUES and Totem Integration

The first step towards our goal is to integrate CLIQUES and Totem. When that integration is complete, we will obtain a **secure and reliable** group communication platform.

The Totem protocol is implemented as three layers: A Single-Ring Protocol that ensures locally ordered and reliable multicast on a LAN; the Multi-Ring Protocol that extends the order and reliability across a WAN to other Totem rings and keeps track of the Totem network topology; and the Process Group Interface that processes group membership changes. Since CLIQUES is only interested in changes in the group membership, the integration only needs to take place at the Process Group level of Totem. The layers of Totem that implement fault tolerance, reliable and ordered delivery would be unaffected.

We propose to add a thin layer called the Secure Group Communication on top of the Totem Process Group Interface to co-ordinate CLIQUES calls with the Totem join, leave, partition and merge messages. This layer will also be responsible for keeping track of CLIQUES members ordering and the shared group key.

To join a secure Totem process group, the application would do the following:

- Call the Secure Group Communication layer which in turn calls the Process Group Interface join function. When that call completes, the process has a list of all the group members including itself. The new process becomes a provisional member of the Totem process group, since it is unable to understand any messages until a new group key is generated.
- When the process that is the current CLIQUES controller, receives a message from the Process Group Interface that a new member has joined the group, it sends a pass_context message to the new member which starts a new key update phase. At the end of this step a new CLIQUES shared key is broadcast to all the group members, and the role of controller is passed on to the newest member.

- When the new process receives the pass_context message from the current CLIQUES controller it uses the information to call clq_join to get the new shared key. (The pass_context message includes the CLIQUES-ordered list of all group members and their partial group keys.) At the end of the clq_join call this process becomes the new group controller.
- Once the new member has the key, it can become an official member of the Totem process group and start receiving secure group messages.

The final change to the Totem protocol is to allow the group members to take advantage of the shared group key to encrypt or sign data messages.

4.2 CLIQUES and Akenti Integration

There are two parts to integrating Akenti and CLIQUES. One is to implement a trusted authority server that would be used to store the names and policy files for communication groups, and use Akenti access control to grant signed capability certificates that authorize a process to join a group. (Akenti currently generates and uses signed capabilities for a security context caching strategy primarily as a performance enhancement mechanism.) As part of this implementation, we need to investigate the range of policies that stakeholder might want to enforce on the communication group, in addition to simply authorizing membership. Possibilities include: a "listen-only" membership mode, allowing a member to sponsor additional entities to join, allowing a member to eject other members from the group, or limiting the time period during which a member can be part of the group.

The second part of the integration is the addition of an authorization step before a prospective member is allowed to join a process group. We expect the authorized group join to proceed as follows:

- Before attempting to join a group, the new process must get a capability certificate granting it membership permission. To obtain it, the process contacts, and authenticates itself to, the Akenti server. The latter issues (signs) the certificate setting the validity interval to a pre-set, short time interval.
- Next, the process calls its local secure communication layer handing it the communication group name and the capability certificate. The communication layer performs the group join command which triggers the current CLIQUES group controller to verify the capability certificate and, once verified, provisionally pass the group context to the new member. Authentication of the new member is implicit: only the authorized process would be able to process the group context and become the new group controller. (A process is not counted as a member until all other members receive an authentic key update from it.)

4.3 Longer-Term Activities

As discussed in Section 3, policy as applied to group membership is an unexplored issue (with the sole exception of some initial exploratory work at SPARTA [5]). We will endeavor to identify and specify common group membership policies. Some are likely to prove trivial to enforce while others -- very difficult. For example, policies contingent on long-term entity attributes (such as rank, security clearance, explicit userid) are the most basic and therefore easy to specify and enforce. Whereas, policies based on highly dynamic attributes (such as admittance based on membership or non-membership of other entities) are hard. Once we develop a taxonomy (and associated difficulty ranking) of membership policies, we can proceed with their incremental implementation.

The understanding of basic membership policies will be helpful in tackling the issue of group certification. As discussed in Section 3.3, many group settings would need (or benefit from) some sort of a group-wide public key certificate. Such a certificate would be explicit or opaque (depending on whether its lists members) and would contain the current group public key and other usual fields. This would allow outsiders (non-members) to communicate securely with the group as a whole. An orthogonal and more granular approach is to issue member-specific group certificates; this would make it possible for group members (and outsiders) to communicate with strong authentication and, perhaps even non-repudiation, guarantees.

Many emerging group applications demand both scalability and reliable delivery. The former is difficult to achieve with a system like Totem as it favors LAN-based communication and involves a lot of communication overhead. For this reason, in the second stage of the project, we plan to experiment with newer, more efficient and scalable reliable multicast protocols. More concretely, we will consider replacing Totem with lighter-weight protocols such as the InterGroup developed at UCSB [4].

Reliable scalability, as a research direction, is not only related to the efficient group communication platform. As discussed in Section 3.4, large groups often have a two-tiered structure with few senders and many receivers. In this setting, the two types of members might demand different security services and guarantees. For example, senders would need to jointly agree on a common key while receivers would simply have the key handed to them. Also, members may need to authenticate themselves as senders or receivers. (Note that simply having a certificate naming one as sender is not sufficient.) Other, more exotic cryptographic services can be considered, e.g.:

- Anonymous group signatures whereby a member digitally signs a message in a way that protects its identity and precludes linking a signed message to others signed by the same member.
- Anonymous voting or petitions: members may need to vote anonymously (e.g., in order to eject or admit a member) and/or sign petition-style documents.

Another item of longer-term study is faster cryptographic primitives for CLIQUES key agreement operations. The basic mathematical building block in CLIQUES protocols is modular exponentiation, a relatively expensive procedure. Recent developments in cryptography have resulted in faster analogues of public key operations that avoid modular exponentiation or greatly reduce its use. Specifically, elliptic curve cryptography [8],[11] is a promising technique for implementing analogues of Diffie-Hellman key agreement. Its main attraction is the significantly reduced (approximately 1-to-5) key length as well as superior performance while retaining equivalent security.

5 Schedule and Milestones

Following is the tentative schedule of project milestones:

Year 1

1. Extend the CLIQUES API to include support for encryption, non-repudiation and authentication of message payloads. (ISI)
2. Integrate CLIQUES and Totem: create Secure Group Communication layer and library. (Design responsibility to be shared between ISI and LBNL, coding to be done at LBNL)
3. Develop a group admission/membership policy; extend Akenti to support novel group policies. (Lead at ISI)
4. Specify and develop a trusted authority server that provides an interface to create communication groups and to store policy files for them. This server is also responsible for issuing the short term capability certificates that authorize group membership. (LBNL)
5. Demonstrate CLIQUES/Totem integration with a simple group application (e.g., group chat, MUDs or teleconferencing) (ISI)

Year 2

1. Integrate CLIQUES/Totem and Akenti to achieve authentication of the group members to produce a secure, reliable group communication. (Any needed extensions to the CLIQUES protocol will be done at ISI, the design work will be done jointly by ISI and LBNL, the code integration will be done by LBNL)
2. Integrate dynamic policy updates for long-lived groups (ISI)
3. Investigate types of group certificates and develop a framework for handling long- and short-lived group certificates, including issuance, processing, revocation, etc. (Design lead by ISI, framework lead by LBNL)
4. Introduce support for 2-tiered groups (few senders, many receivers). This may be used in support of the InterGroup protocol.(ISI)
5. Demonstrate results with sample application possibly from an NGI application(e.g., collaborative visualization, instrument control).(LBNL)

Year 3

1. Scalability study (ISI)
2. Alternative cryptographic methods: recasting CLIQUES with elliptic curve cryptography (LBNL)
3. Integration with other applications and deployment trials (LBNL and ISI)
4. Intensive technology transfer (LBNL and ISI)

Literature Cited

[1] D. Agarwal. The totem project web page. In *http://alpha.ece.ucsb.edu/totem.html*.

[2] G. Ateniese, D. Hasse, and G. Tsudik. *The CLIQUES Application Programming Interface*.

[3] G. Ateniese, M. Steiner, and G. Tsudik. “Authenticated group key agreement and friends”. In *ACM Symposium on Computer and Communication Security*, November 1998.

[4] K. Berket, L. Moser, and P. Melliar-Smith. “The intergroup protocols: Scalable group communication for the internet”. In Proceedings of the 3rd Global Internet Mini-Conference}, November 1998.

[5] H. Harney, C. Muckenhirn, and T. Rivers. “Group key management protocol (gkmp) architecture”. In INTERNET-DRAFT, *work in progress* Version 1.0 1996

[6] M. Hayden. *The ensemble system*. Technical Report TR98-1662, Cornell University, January 1998. Available from *http://simon.cs.cornell.edu/Info/Projects/HORUS/Papers.html*

[7] W. Johnston, S. Mudumbai, and M. Thompson. “Authorization and attribute certificates for widely distributed access control”. In *Proceedings of the IEEE 7th International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises - WETICE’98*, 1998.

[8] A. Menezes. *Elliptic Curve Public Key Cryptosystems*. Kluwer Academic Publishers, 1993.

[9] L. Moser, P. Melliar-Smith, D. Agarwal, R. Budhia, and C. Lingley-Papadopoulos. “Totem: A fault-tolerant multicast group communication system”. *Communications of the ACM*, April 1996.

[10] O. Rodeh, K. Birman, M. Hayden, Z. Xiao, and D. Dolev. *Ensemble security*. Technical Report TR98-1703, Cornell University, January 1998. Available from *http://simon.cs.cornell.edu/Info/Projects/HORUS/Papers.html*

[11] M. Rosing. *Implementing Elliptic Curve Cryptography*. Manning Publishers, October 1998.

[12] M. Steiner, G. Tsudik, and M. Waidner. “Diffie-hellman key distribution extended to groups”. In *ACM Symposium on Computer and Communication Security*, March 1996.

[13] M. Steiner, G. Tsudik, and M. Waidner. “Cliques: A new approach to group key agreement”. In *IEEE Conference on Distributed Computing Systems*, May 1998.

[14] M. Thompson. The akenti project web page. In *http://www-itg.lbl.gov/security/Akenti*.

[15] M. Thompson, W. Johnston, S. Mudumbai, G. Hoo, and K. Jackson. “Certificate-based access control for widely distributed resources”. In *Usenix Security Symposium’99 (submitted)*, 1999.

[16] G. Tsudik. The cliques project web page. In *http://www.isi.edu/div7/cliques*.

[17] P. Zimmerman. *The Official PGP User’s Guide*, 1994.

PI Biographical Sketch

Mary R. Thompson

Education

- MS Computer Science, University of California at Santa Barbara (1976)
- BS Physics, Stanford University (1963)

Current Position

Computer Engineer III, Lawrence Berkeley National Laboratory

Research Interests

- Distributed systems: distributed access control; secure communications, fault tolerant systems PKI based applications,
- Operating systems: distributed file systems, multi-processor systems

Professional Experience

Mary Thompson has worked in the field of Computer Science for thirty years. She joined the Imaging and Distributed Computing Group at Lawrence Berkeley National Laboratory five years ago, where she developed a Web-based image library system in support of microscopists. This system has subsequently been used to provide on-line access to part of the Lab's historic photo-archive and used in conjunction with a high-speed network distributed storage system to manage medical video clips. For the last two years her work has been in the area of certificate based access control in distributed systems, helping implement and maintain the Akenti distributed access control system.

Prior to joining LBNL, Ms. Thompson was a staff member of the Computer Science Department of Carnegie Mellon University for seventeen years. While there she contributed to the development of the Mach operating system and was eventually responsible for the quality control and external distribution of this system. During her time at CMU, she also worked on the Hydra capability based operating system and a central file system.

Ms. Thompson also worked for five years at MIT on the Multics operating system and for two years as a scientific applications programmer at Boeing Aerospace.

Related Publications

W. Johnston, S. Mudumbai, and M. Thompson. "Authorization and attribute certificates for widely distributed access control". In *Proceedings of the IEEE 7th International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises - WETICE'98*, 1998

M. Thompson, W. Johnston, S. Mudumbai, G. Hoo, and K. Jackson. "Certificate-based access control for widely distributed resources". In *Usenix Security Symposium'99 (submitted)*, 1999.

Consortium Arrangements

This work is going to be done in conjunction with Gene Tsudik at the Information Sciences Institute of the University of Southern California. CLIQUES is a research project led by Dr. Tsudik and he will be responsible for enhancements to the protocol necessitated by the integration with Totem and Akenti. He will also be the lead on the investigation of access policy for dynamic and two-tiered groups.

Deb Agarwal, who is a member of the Imaging and Distributed Collaboration Group at LBNL and one of the originators of the Totem protocol, has agreed to provide expert advice on the protocol as needed.

Description of Facilities and Resources

The Imaging and Distributed Collaboration Group at Lawrence Berkeley Laboratory has a network of approximately 30 Sun workstations, 10 Linux/BSD workstations and 3 Sun Enterprise servers connected by 100Mb ethernet LAN. This is approximately 2 workstations for each full time researcher. The group runs a Netscape Certificate Authority and LDAP server for investigating PKI based access systems.

Other Support of Investigators

75% DOE2000 Diesel Combustion Collaboratory Security

25% Distributed Security Architecture

Budget Explanation

The principal investigator will supervise the work of two graduate students and coordinate the design effort between LBNL and ISI. The students will be responsible for detailed design work and implementation. They will coordinate with the implementation work at ISI. The travel money its intended to cover three trips to ISI per year for one or two of the researchers. One of the graduate students will be a new hire, so will require the purchase of a workstation.