2005 International Command and Control Research and Technology Symposium

The Future of Command and Control

Enabling Coalition Operations with a New Standard for Group Security and Key Management

Topic: Coalition Interoperability

15 March 2005

Hugh Harney
Principle Investigator
hh@sparta.com
SPARTA, Inc.
7075 Samuel Morse Dr.
Columbia, MD, 21042
Phone: 410 872-1515
Fax: 410 872-8079

Rod Fleischer
Senior Engineer
rodf@sparta.com
SPARTA, Inc.
13400 Sabre Springs Parkway, #220
San Diego, CA 92128
Phone: 858 668-3570
Fax: 858 668-3575

Enabling Coalition Operations with a New Standard for Group Security and Key Management

Abstract

Today's military operates almost exclusively through coalition operations. The reality of operating in coalitions poses increasing operational, managerial and security issues. This mandates effective, efficient and assured information sharing among coalition partners, while preserving security for sensitive information. New group security protocols have been developed which provide full end-to-end (publisher-to-consumer) information security.

Current point-to-point protocols only provide connection-level security, resulting in a loss of end-to-end security services when used with multiparty servers. These point-to-point systems only secure data between users and servers, with no rigorous method to define or enforce synchronized group security policy, or to provide secure end-to-end associations directly between users.

In this paper, we will discuss a new generation of group security protocols and standards that have caught up with the requirements for 'Assured Sharing', while simultaneously enabling flexible group key management. The Group Secure Association Key Management Protocol (GSAKMP) provides a standard for distributing cryptographic keys as well as a trustable architecture for defining and enforcing group security policy. With this functionality, a new class of secure group applications for content-based approaches, such as Secure Group Objects (SGO) can provide end-to-end security services to coalitions, resulting in increased network infrastructure performance while simultaneously enhancing overall security.

1. Coalition Requirements

A coalition is defined as a temporary alliance among people, organizations and nations to achieve a shared common goal. Coalition memberships can change rapidly, making dynamic configuration an important requirement. In some cases, coalitions must support multiple Communities of Interest (COI) or 'enclaves', each with their own group security protocols. Keeping communications and resources secure in the face of such diverse requirements becomes a particularly challenging problem.

This paper will examine solutions for securing coalition communications and applications. There are several reasons why this is such a difficult dilemma: information must be shared between coalition members, but not necessarily all members; coalition communications are often only among subsets of members; and coalitions are transient – with constant membership changes in an *ad hoc* manner. The point-to-point security paradigm is at best awkward when applied to coalitions, and identifies that a more elegant approach is needed.

1.1. Communication of Information among Members

The nature of coalition communications varies as greatly as the missions that it supports. However, in all cases, coalitions require that shared information be made available to members in order to support a coordinated effort. For the purposes of this paper, the process of making this shared information available to group members is referred to as *Group Communications*.

Secured Group Communications fall into two distinct categories: "tunneled" and "content-oriented". "Tunneled" is defined as connection-oriented, or point-to-point communication. An active communication channel is created between interested coalition members, and remains active for the duration that communication is desired. "Content-oriented" refers to the store-and-forward model, also known as publish-and-subscribe, a prime tenet of Net-Centric operations. In this model, information objects are

created by a coalition member for distribution among other coalition members, but there is no active connection.

1.1.1. Tunneled Coalition Communication

Tunneled protocols are used to create a communication association among multiple hosts, which is then used to transmit data between those hosts. Each host must cooperate to establish and maintain the association, including the security of the association. The secure communication association acts as a pipe through which information securely flows. Secure Socket Layer (SSL) and Internet Protocol Security (IPSec) are two examples of secure tunneled communication. These protocols create connections which transparently provide security for all data sent through them. However, upon "arriving" at a tunnel-endpoint, the data is no longer protected.

Establishment of secure tunnels between exactly two hosts is vastly different from setting up tunnels among three or more hosts, as illustrated in Figure 1. In pair-wise connection-oriented communications, one important security feature is a complete understanding of the endpoints (e.g., users) that are sending and receiving the packets. The protocol (and presumably the data sender) "knows" the identity of all the recipients prior to sending the data. This feature is important if one wants to review the privileges of the endpoint users prior to sending them the data, as well as verifying that data came from a known and trusted identity. Another important feature is the capability to directly negotiate the security mechanisms and keys that are needed immediately before the communications.

Complete understanding of the endpoint users and the capability to directly negotiate the security mechanisms are aspects of tunneled pair-wise communication that are vastly more difficult, if not impossible, to provide in tunneled communications for groups. Because of the potential for large numbers of group members, security mechanisms and keys must be provided to each member as he/she joins the group (usually by a trusted infrastructure component). This resulting case is that members who join the group early do not necessarily know the identities of the users who join the group later.

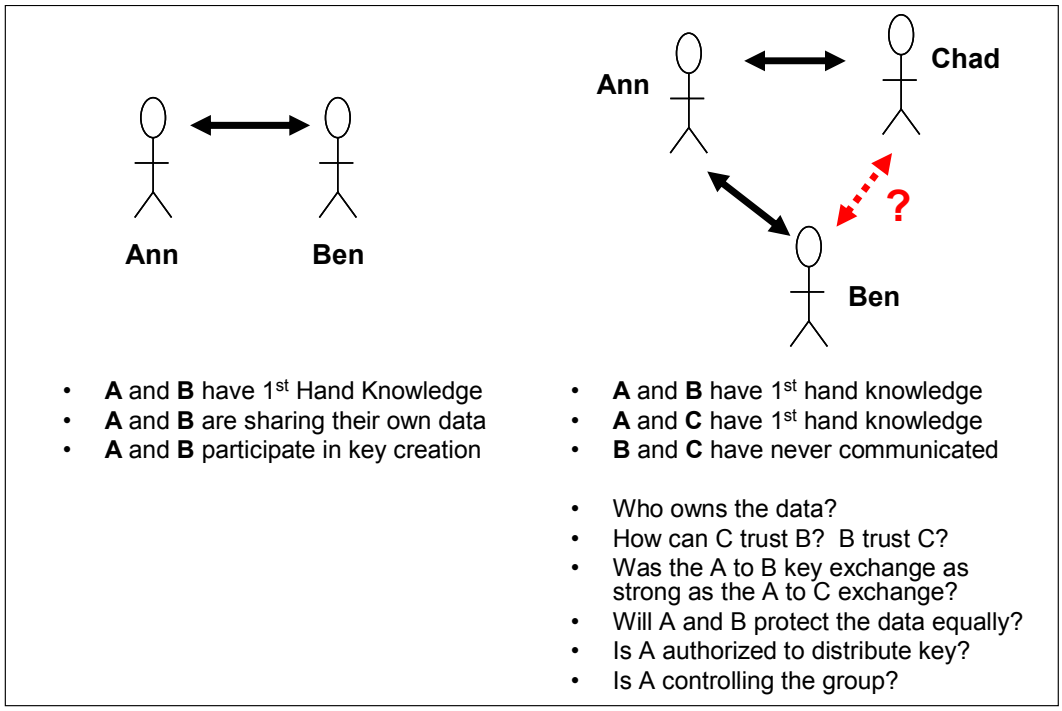

Figure 1: Pair-Wise Communication

1.1.2. Content-Oriented Coalition Communication

Content-oriented communication systems, such as those which implement the Net-Centric "Task, Post, Process, Use" (TPPU) model, do not provide a direct and active connection between the data sources and data receivers. In systems of this type, coalition members post the data to intermediary servers, which store the data until it is later retrieved by other members. The members from which the data originates often have no knowledge of the eventual recipients of the data. As such, there is no cooperative protocol between the data source and the data recipient which is responsible for maintaining the security of the information. Content-oriented communications are object based, with data objects being defined as files of any type. These objects (or information) are the important aspect of this communication model, and not the transport system.

The content-oriented nature of this type of system leads to an interesting phenomenon: data objects can be moved intact between many different forms of transport medium. For example, the data source may create an object and e-mail it to a friend, who may in turn post the object to a web server. An unknown third party may retrieve the object from the web server and place it on a peer-to-peer file-sharing network. The adaptability and power of a content-oriented system is displayed by this example, where the data object is the same in all cases, but the transport medium by which it is manipulated varies from situation to situation.

One final observation about content-oriented communication is illustrated by the discussion of the multiple-protocol delivery paradigm: the data source may have no knowledge of the recipients. This is problematic from a security perspective because secure systems need to be guaranteed that the data contained in objects will only be made available to persons authorized to receive the data. It must be possible to grant access to objects as the need to share information is determined.

1.2. Coalition Security Principles

Coalitions are group based and, as such, they inherit all issues related to creating trustable groups. In general, the five principles to group security (all of which must be met for a group to be considered secure) are as follows:

- Principle 1: Group security policy enforcement must be consistent across a group.
- Principle 2: Only authorized entities can affect the group's security posture.
- Principle 3: Group content must be protected.
- Principle 4: Groups must be capable of recovering from security-relevant failures to a secure state.
- Principle 5: Groups have dynamic membership: access to information at time X does not guarantee access to information at any other time.

2. Current State of Practice and Problems with Secure Coalition Applications

Current coalition approaches were built with the best protocols on hand several years ago. At that time, the security protocols of choice, by and large, were peer-to-peer protocols such as Internet Protocol Security (IPSec), Internet Key Exchange (IKE) and Secure Socket Layer (SSL). These protocols worked well and ensured secure connections between two entities; however, coalitions frequently needed to secure groups of entities, and IPSec/IKE and SSL were not designed to provide group-level security. This section examines some common coalition approaches, and their security-relevant shortcomings. This discussion is not intended to be critical of coalition network architects, as the required security protocols for end-to-end solutions were not available to achieve group-level security at that time.

One of the primary challenges that coalition architects face is multiparty information delivery – data has to be delivered with reliability, and the core IP networks have not met this requirement as a general rule. The architecture developed by many designers is a centralized application server that everyone connects to using Transmission Control Protocol (TCP), so that the server can send the same data to many people. This architecture is illustrated in Figure 2. In cases of *non-real-time* connections (e.g., store-and-forward applications), servers are also used. These servers provide a data store and registration point for coalition applications and have been the best solution available over the last decade.

On the surface, it appears reasonable to have coalition members create secure associations between the end user's computer and the coalition application server. The secure association enables a peer-to-peer, mutually suspicious review of each side's credentials and creates a cryptographically protected tunnel for data transmission. The theory is that if all application users undergo a mutually suspicious credential review and the server deems each member to be acceptable (thereby granting a secure association), then coalition applications are secure.

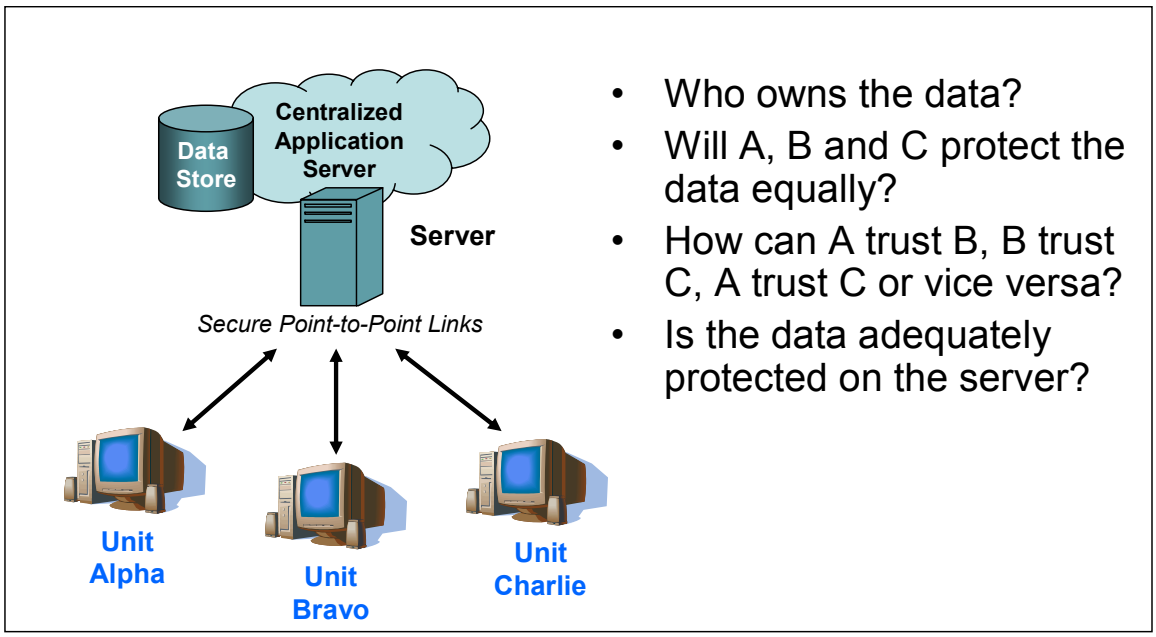

Figure 2: Centralized Application Server

2.1. Security Issues with the Current Practice

Unfortunately, there are several key security concerns with the old style security servers with respect to coalition architectures. The biggest issue is the lack of a group security association, resulting in each data producer (e.g., coalition member in a community of interest) lacking assurances of an end-to-end security association between the server endpoints. Put another way, the data sources have no control or participation in the security policy, and have no guarantees that the data which they send to the group will only be received by authorized members.

2.1.1. No End-to-End Security Association among Coalition Members

There are multiple inherent problems with the current practice of using point-to-point security protocols for group applications. First and foremost, point-to-point protocols secure only the connections *between users and the servers*, and not among the users themselves. Use of these protocols to provide multiparty communication is an insufficient solution, since it does not provide end-to-end security for the users. These

secure connections successfully authenticate users to the servers, but provide no guarantee to other users. A member receiving data is unable to know where it originated from, since the point-to-point protocol only reliably guarantees that the data was sent by the *server* that brokered the connection. The connections encrypt the data *during* transit, but only while on the network *between users and the servers*. Users are then left without any knowledge of what happens to the data while it is on the server, as illustrated in Figure 3. For coalition members or end-users, it means that they must trust the intermediary server (which generally requires joint/coalition development on the server security policy and application design).

Figure 3: Secure Point-to-Point Designs Do Not Guarantee the Security of the Data While on the Server

The lack of an end-to-end group security solution means that the group-level security policies are not reviewed by members, thereby requiring members of these networks to release classified data to the group connected to these servers *without having any idea to whom they are releasing the data.* No policy review occurs prior to data release and the best that can be inferred is that members trust the centralized server absolutely since the server is making all of the security decisions for coalition members.

2.1.2. Server Insecurities

The servers in group-based coalition networks are critical to the security of all of the data that each member submits to it. Since the cryptographic protection of the data ends with the point-to-point security association, the server has full access to the data. Further, since all security associations are created from users to servers, the servers have access to all data sent by all coalition members. The concern is that the server could violate access controls on the data by sending it to an unapproved user. It could also modify the data

without detection by group members, or masquerade as any member that it chooses – leading to Information Operations (IO) questions. In addition, all server administrators have access to the data to the same extent as the server itself, resulting in a high-risk of compromise from a single person.

The list of server issues is too long to cover in this paper, but points to the fact that the common answer is to protect the server. This is easier said than done, as few operating systems can partition members in a high-assurance manner, which means that most servers cannot guarantee that data intended for one group on a server will not be accidentally exposed to any other group on the server. The common solution of isolating groups, at least to those with the same protection requirements, is to use separate servers for each set of users. This requires that a coalition network has a server for each set of coalition attributes (or communities of interest and each level of security), leading to expensive operations and headaches over registration and administration issues.

3. Approaches for Secure Coalition Applications

Coalition networks will require both “tunneled connection” and "content-oriented" communication paradigms to accommodate both real-time (e.g., voice, chat, collaboration) and non-real time (store-and-forward) situations. In both instances, coalitions can now use proven security principles to create systems that are trusted and meet modern coalition requirements. In this section, we will introduce new approaches to achieving secure group applications.

3.1. Group Security Associations (GSA)

A Group Security Association (GSA) is a set of methods and protocols defined by the Internet Engineering Task Force (IETF) Multicast Security (msec) working group to address group key policy and management. The central idea is that a GSA creates a group of entities that share cryptographic keys in order to protect group data. The GSA concerns itself with more than just key management, because keys and cryptography are

only mechanisms to enforce a group *policy*. The GSA enables the keys and policy in a group to be distributed to group members in a way that facilitates trust in the security of that group by all group members. This approach is critical for negotiating the group security policy and key management among the coalition members.

3.2. End-to-End Protection

The primary benefit of a GSA in which all group members share cryptographic keys is that security can be performed at the endpoints of communication. End-to-end security approaches describe a secure cryptographic design that protects data from source to destination, so that there are no intermediary storage points (such as a server) where protective cryptography is stripped off, leaving the data open for disclosure or modification. This simple concept is shown in Figure 4, contrasted to the earlier point-to-point centralized server application model used frequently today.

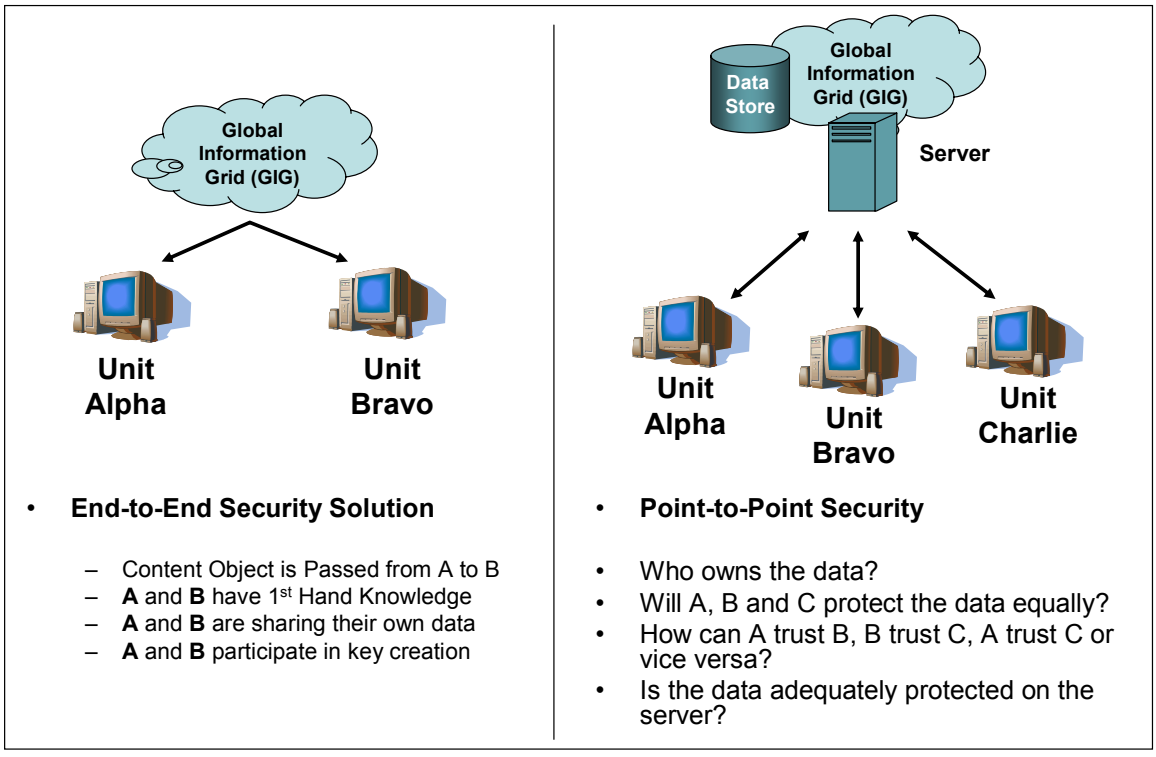

Figure 4: Enabling End-to-End Security

A critically important aspect of the end-to-end model is that the security is performed independent of the communication medium. There are several immediate benefits from this, first and foremost among them being that the communication method can be changed without impact to the security of the data. Another major benefit is that the servers are unburdened from having to provide security services, freeing them to merely route network traffic, and often yielding noticeable network performance improvements. Additionally, since the servers no longer perform any security services, they cease to be a point of vulnerability for the data.

End-to-end secure systems also offer mutually suspicious protocols where all endpoint users must prove their identity and authorities within the security system. This important feature is critical for proving that the security system can successfully enforce the security policy throughout the system life cycle.

3.3. Synchronized Security Mechanisms

By nature, coalitions are a combination of groups that frequently have different, but equivalent, mechanisms for securing coalition data. In a coalition network, the first priority for secure operations is the definition of a policy that is flexible enough to allow for disparate security labels, attributes, identity formats and infrastructures. In a coalition, policy parameters are likely to be dynamic as the coalition membership changes; therefore, the security mechanisms must be synchronized across all coalition members to allow a graceful evolution of policy.

One of the principals of GSA is that these heterogeneous mechanisms can be *synchronized* and managed within the coalition, providing a way to utilize the full capabilities of coalition members and provide a common level of secure operation. Key synchronization is the final secure coalition network requirement. If cryptography is to be used effectively in a coalition, the state of keys must be tightly synchronized and must accommodate new member additions, member deletions, member expulsions, system start-ups, and key rollovers.

4. Technologies for Supporting Secure Coalition Applications

Coalition application designers have been hampered by a lack of group-based security protocols, technologies and applications -- they had to make do with point-to-point technologies. However, this situation is changing in that the IETF is now publishing standards and protocols applicable to group security. These new protocols allow coalition designers to create end-to-end protected applications that truly secure the coalition.

4.1. Group Secure Association Key Management Protocol (GSAKMP)

The Group Secure Association Key Management Protocol (GSAKMP) is a new group security protocol for managing group key data and security policy. GSAKMP operates by defining a strong security policy for the distribution of group keys, and rigorously enforcing that policy. It also provides a mechanism for recovering the group to a secure state from any compromise of group key data. GSAKMP has applicability to IPSec, SSL and Secure Group Objects (SGO) as described next in this section.

GSAKMP provides a number of attractive features, which collectively address the five principles of group security as defined in section 1.2. These features include:

- Strong cryptographic key management.
- Security policy definition with synchronized enforcement.
- Balanced cryptographic and access control mechanisms.
- Ability to recover to a secure state following a security-relevant failure.
- Assigned roles with delegation, allowing group scalability to Internet sizes.

4.2. IPSec for Network Layer Broadcast with GSAKMP

IP Security (IPSec), an extension to the International Organization for Standardization (ISO) 7-layer network model, provides security associations at the network layer (host to

host). Located between the network and transport layer, it adds encryption, authentication and replay protection to all in- and out-bound network communications. IPSec, however, protects data only while it is being transmitted between points or hosts.

IPSec has recently made it possible to send and receive IPSec packets from multicast addresses, provided replay protection is not enforced. However, several issues complicate our ability to completely rely on IPSec for support of coalition applications. IPSec, in itself, is not reliable (i.e., self correcting for loss of packets); it relies on protocols above itself to provide reliability. In addition, the turning off of replay protection in order to accommodate sending and receiving on multicast addresses mandates that replay protection be provided by another higher protocol.

As a network layer protocol, IPSec does not support group level policy negotiations; these could be provided if an appropriate key and policy management system such as GSAKMP were used to supply a shared group key to IPSec. The combination of IPSec and GSAKMP provides an appropriate network layer security protocol suite appropriate for coalition applications.

4.3. Security for Session Layer Reliable Multicast (SLRM) with GSAKMP

Multicast, with its one-to-many communication architecture, is conceptually very well suited to group applications. However, reliable multicast communication at the IP layer is extremely difficult. Since multicast communication is transmitted over the network via User Datagram Protocol (UDP), there are no reliability guarantees made by the underlying protocol. Additionally, due to the tremendous impact which multicast transmission can have on network bandwidth, most Internet routers do not allow the forwarding of these multicast packets. However, the potential benefit of one-to-many communication to a group collaborative application is too great to ignore. So, we must explore alternative means of achieving this communication model while circumventing the unreliability and network utilization problems.

4.3.1. Reliable Multicast via Overlay Networks

One technique which seems to work well for delivering multicast messages to group applications in a reliable manner is the use of Overlay Multicast networks. Overlay networks provide multicast delivery above the IP layer. They generally utilize the reliable transmission characteristics of the TCP protocol suite to achieve the reliability needed for multiparty real-time communication applications such as VoIP, IM, and teleconferencing. These applications are particularly sensitive to data loss during transmission, and benefit greatly from the reliable multicast architecture which can provide real-time delivery of coalition data to multiple group members simultaneously. This type of reliable multicast architecture is referred to as Session Layer Reliable Multicast (SLRM).

An overlay network creates an SLRM by utilizing TCP between group members and servers that distribute multicast traffic. Overlay networks set up a system of servers that act as multicast message repeaters and registration points. Spread (www.spread.org) is an example of an existing overlay network using the described architecture. This architecture can be conceptualized as a set of hubs with spokes, as diagrammed in Figure 5. In this architecture, each group member connects to a local hub to send multicast traffic. The hubs pass multicast traffic between themselves via a reliable protocol tunnel. Each hub then delivers the multicast data to each of the members connected to one of that hub's spokes.

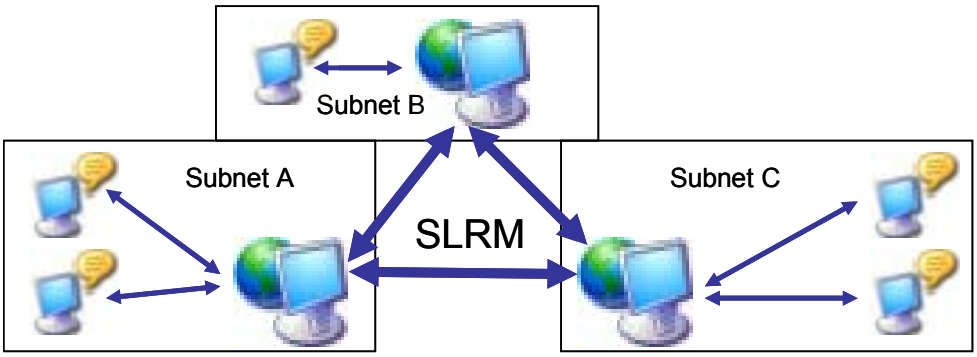

Figure 5: Session Layer Reliable Multicast (SLRM) Provides Reliable Multicast Tunneling Between Different Subnets.

While this is somewhat analogous to using point-to-point connections for group communication, the total number of point-to-point connections between the overlay hubs is much smaller than the requirement to directly connect every group member. In addition, new group members can be introduced on a subnet without requiring additional overlay connections. There is also no security burden placed on the overlay servers, so they only need to act in a message-forwarding capacity.

4.3.2. Securing SLRM with Group Secure Associations

Once the basic communication issues are resolved, we can create a secure service for SLRM. The first observation is that the security endpoints are the group members, not the communication components. The group members own the group data and need to protect it. An end-to-end security approach provides a GSA between all the group members and specifies a set of cryptographic mechanisms to protect the data in that group.

This is an application well suited for GSAKMP, which can create a GSA for the Secure-SLRM services in order to provide security in a separate fashion from the underlying communications architecture. Therefore, the military can use existing SLRM systems in conjunction with GSAKMP, as illustrated in Figure 6, without having to retrofit special security options into the existing SLRM architecture.

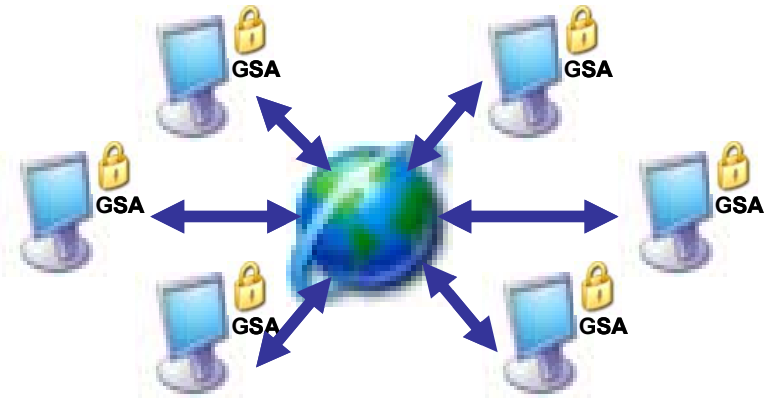

Figure 6: Coalition Members Sharing a GSA Creates an End-to-End Secure Session Layer

A Secure-SLRM application provides encryption, authentication and non-repudiation services to messages, which are then enveloped into a SLRM datagram. The SLRM communications service delivers the traffic to the group as designed. Once the Secure-SLRM datagram is retrieved by the group members, the Secure-SLRM application ensures that the group member has the correct and authorized keys for coalition operations.

4.4. Secure Group Objects (SGO)

Information on the Internet can be distributed in a group paradigm, as an encrypted data object. An information source can create an "information object," and place that object in an application (e.g., a web page). These applications offer a large number of potential recipients (a group) access to the information object. Security for Internet distribution can then follow a secure group paradigm. Information can be protected at the source before being offered to a group of potential receivers. Only those receivers with permission to read or modify the information can gain access to the underlying information. The secure group paradigm for data objects on the Internet is called Secure Group Objects (SGO), an application of GSAKMP for content-oriented network communication.

Conceptually, a Secure Group Object is defined as a group resource (typically a file of any type) which has been encrypted using a GSAKMP group key. The encrypted data is wrapped within an envelope that contains metadata about the GSAKMP group to which the data belongs. Since the data content of the SGO is encrypted, it can be published or transmitted in any desired fashion. Only recipients who have the ability to participate in the GSAKMP group which owns the SGO will be able to access the contents.

4.4.1. What SGO Provides

An application using SGO provides a generic information protection infrastructure for many Internet applications. A single SGO application can protect information of a similar sensitivity on the following:

- Web pages
- Peer-to-peer applications
- Wireless network
- Large integrated applications

The SGO application is able to support such a wide range of Internet transport mechanisms because it separates security from information delivery. All secure systems must provide certain common security functions; SGO provides these without disturbing delivery applications. If modern Internet applications are the race horses of technology, the SGO system frees these horses from pulling the security plow.

The separation of security from information delivery allows operational systems that are handling sensitive information to use modern Internet applications to publish, search and distribute information. Military organizations, then, are able to use commercial applications (i.e., Commercial-off-the-Shelf (COTS) products) to deliver sensitive data.

Coalition networks employing SGO applications require fewer high-assurance computers, resulting in lower operational costs. SGO applications provide content-based 'source-to-destination' security, so that sensitive data is protected throughout distribution of the information, significantly reducing the vulnerability of intervening distribution systems. The distribution systems only see encrypted data; they never have access to raw sensitive information, once again lowering the military's operating costs.

An application using SGO also allows dynamic changes in group membership using cryptographic methods. Therefore, a coalition can now control the distribution of critical information without having to resort to "hands on" system administration at every receiving site. The separation between communication and security also allows objects to be cryptographically protected with incomplete knowledge of recipients. The access control rules for the object are managed by the security system, and need not be fully known at the time of object creation. This feature maps exactly to the Internet model of

information sharing, and distinguishes SGO from other techniques to encrypt objects. The best way to describe an application using SGO is to step through the process of creating an SGO group, distributing the secure object, and retrieving the key to decrypt the secure object, as discussed in the next sections.

4.4.1.1. Creating an SGO

The following process maps the creation of the initial SGO, which is the most complicated case – but one that best illustrates the system's capabilities. The first step is to determine the parameters that the SGO application will follow for a particular class of objects. Information such as level of classification, access control rules (people or organizations), and security mechanisms must be identified in this step to create the policy token. Once the group parameters are set, the security infrastructure automatically creates the cryptographic group and waits for people to join the group and obtain the key. The key associated with the cryptographic group is passed to the SGO application and used to encrypt sensitive information.

4.4.1.2. Distributing the Secure Group Object

The act of distributing an SGO is elegantly simple – use any Internet transport mechanism. This capability appears so simple that it is easy to overlook the breakthrough involved with being able to use any Internet application as illustrated in Figure 7. With the use of SGO, the application doesn't have to be modified to be secure or be "protected". The military need not invest development dollars into distribution applications, as every SGO is encrypted. The encrypted objects' sensitivity level is much lower than the level of the information contained in the object; in fact, every SGO can be considered to be at a common classification (unclassified), which means that a single Internet application or web page may distribute SGO resources containing information at many different levels of sensitivity. The need for redundant distribution networks and servers to segregate information disappears. The cost of building military data sharing infrastructures drops, the complexity drops, and the time to deploy drops.

Figure 7: An SGO Encrypted by a GSA Stays Secure Without Any Special Servers or Connections.

An SGO is secure because only those authorized (as dictated by group access control parameters in the security policy) has the ability to obtain the key in order to decrypt the SGO and read the information.

4.4.1.3. Retrieving the Key to Decrypt the Secure Object

When an authorized person goes on the web to retrieve an SGO, if they are already a member of this group, they already possess the key; if they are not a member of this group, they must join. One key feature of GSAKMP, but beyond the scope of this paper, is automatic key re-distribution or recovery from compromise.

Under normal operations, when a person downloads an SGO, they open it with the SGO application software. The workings of the SGO application are complicated, but invisible to the user. When the SGO application receives a new SGO, it reads the unencrypted information that makes up the SGO *envelope*. It then passes the relevant information to the GSAKMP protocol associated with that user. The envelope gives GSAKMP enough information to find the SGO infrastructure for that object. These discovery protocols happens in several ways, but for brevity, assume that an *anycast* protocol is used. *Anycast* takes requests for service from users and connects those users with service access points. The details of joining the group and obtaining the key are handled transparently by GSAKMP.

5. Conclusions

Coalitions have group communication requirements that will be best served by end-to-end secure services. Unfortunately, group end-to-end cryptographic security services have not been available until recently. Coalition application designers have compensated with complicated point-to-point security architectures because those were all that were available. These technologies allowed coalition applications to be created, but the applications suffered from security vulnerabilities caused by the point-to-point nature of the existing technology.

Security standards have advanced to provide a GSA, which offers a more flexible group security policy mechanism for many coalition applications. A GSA also fixes a number of security, architectural, and communication issues with coalition applications. The GSAKMP policy and key management protocol is a recent addition to the standards community, and meets the security requirements for many coalition applications. GSAKMP provides end-to-end support for cryptographic services used in coalition applications.

Two new security approaches using GSAKMP support more secure coalition applications: secure session layer cryptography and Secure Group Objects. Secure session layer encryption allows real-time secure associations to be created among a group of coalition members. These coalition members in turn use the GSA to perform real-time coalition applications like Chat, Instant Messaging (IM), Voice over Internet Protocol (VoIP) and video. Secure Group Objects (SGO) provide a GSA for Internet data objects, and allow coalition networks to use commercial data transmission methods such as web pages or e-mail. The security for an SGO is applied at the data source and data destination endpoints to create content-based security, using an approach that is invisible to the intervening Internet resources.

The state of security has advanced to address security and architectural issues faced by coalition application developers. More importantly, the new security technologies provide greatly improved security services for the coalition war fighters.

6. References

1. IETF RFC 2093(Group Key Management Protocol Specification)

2. IETF RFC 2094 (Group Key Management Protocol Architecture)

3. IETF Internet Draft*: http URL www.ietf.org/internet-drafts/draft-ietf-msec-gsakmp-sec-07.txt*

4. IETF Internet Draft*: http URL www.ietf.org/internet-drafts/draft-ietf-msec-policy-token-sec-02.txt*

5. H. Harney, A. Colegrove, and P. McDaniel, *Principles of Policy in Secure Groups*, In Proceedings of Network and Distributed Systems Security 2001. Internet Society, February 2001. San Diego, CA.

6. P. McDaniel, H. Harney, A. Colegrove, A. Prakash , and P. Dinsmore, Multicast Security Policy Requirements and Building Blocks (*Draft)*. Internet Research Task Force, Secure Multicast Research Group (SMuG), November 2000. (`draft-irtf-smug-polreq-00.txt`). [Draft Text: txt Abstract]

7. P. McDaniel, H. Harney, P. Dinsmore, and A. Prakash, Multicast Security Policy (*Draft)*. Internet Research Task Force, Secure Multicast Research Group (SMuG), June 2000. (`draft-irtf-smug-mcast-policy-00.txt`). [Draft Text: txt Abstract]

8. T. Hardjono, H. Harney, P. McDaniel, A. Colegrove , and P. Dinsmore, Group Security Policy Token (*Draft)*. Internet Research Task Force, Secure Multicast Research Group (SMuG), November 2001. (`draft-ietf-msec-gspt-01.txt`). [Draft Text: txt Abstract]