

## Overview of Quality of Security Service

Cynthia Irvine and Tim Levin  
Center for INFOSEC Studies and Research  
Naval Postgraduate School  
[irvine | levin]@cs.nps.navy.mil

We present an overview of our approach to managing security as a dimension of Quality of Service. Relative to traditional QoS attributes (e.g., jitter, deadline, latency) security has been handled rather statically and indirectly. We have developed a theory of Quality of Security Service and a related security-costing framework that supports extension of QoS functionality to embrace existing and emerging security technologies. Our goals have been to leverage existing security mechanisms to improve availability, predictability, and efficiency, while maintaining, if not increasing security of the distributed system.

Variability in *user and application* security requirements allows the underlying control system to be more adaptable in responding to requests for resources, and variability in *system and resource* security requirements allows the distributed system, e.g., through quality of service (QoS) middleware, to offer security choices to users or applications. The availability of user security choices along with support for management of security resources in response to user requests enables quality of security service (QoS). We have found that many existing mechanisms and policies allow for security variance. For example MAC and DAC allow for whole sets of solutions via their “dominance” and set inclusion relationships, and many so-called *fixed* requirements can be seen to actually define only minimums, allowing for a range of solutions. Some examples of security service attributes that provide ranges are the choice of cryptographic algorithm, number of rounds or key length, assurance level or strength of boundary control in a remote environment, or even the capability level of the environment’s security administrators.

QoS can be seen as the modulation of resources to deliver requested services to users, which depends on the control and variability of resources. Similarly, QoS involves the modulation of *security* resources, and depends on the control and variability of those security resources. In a typical distributed system, the security restrictions and requirements confronted by a user emanate from many layers, components and services. How can QoS or resource management middleware make sense out of this apparent chaos in attempting to manage the system efficiently? Our approach involves several abstractions: the first is to view all security restrictions as service attributes. The second is to view all security restrictions as a range that defines a set of partially ordered possibilities, where some values are “more secure” than others. Of course, in some cases the range is degenerate, meaning the related service can be used in only one way.

To understand how these ranges can be used in a layered distributed architecture, consider how a request for execution of a task is passed between different layers, and security services are provided in response to these requests. As this *task sequence* is processed, there are both choices and limits regarding each security restriction or requirement. A *choice* is the security range request passed to the next layer. A *limit* defines which requests from previous layers are acceptable. In the end, if the task is realized by the system, meaning that the various choices and limits have been successfully processed, the user’s expectations for quality of security service will have been met. Additionally, the QoS middleware will have had additional latitude, by way of variant security requirements, in fulfilling user and system-wide goals, thereby potentially increasing the availability, predictability, and efficiency of the system.

We have developed a prototype demonstration for the QoS management of a low-level security service, IPsec. In this demonstration, a high level interface captures changes to the security posture of the network and passes that information to a customized IPsec policy module. IPsec dynamically adjusts its low-level security settings to the new security posture. A second demonstration shows how complex sets of IPsec security settings can be formulated through a visual XML security policy editor.