URI Security Architecture: Trial Implementation

Working Document

Paul Wernick
Computer Laboratory
University of Cambridge

17th July 1995

1 Management Summary

This document describes the rationale, objectives and implementation plan for a first exploratory prototype system towards the British Telecom URI security architecture.

The prototype is intended to enable us to appraise our work to date, *viz.* the underlying assumptions and the decision to use GSS-API as an interface to authentication and authorisation software, in a working environment. The implementation process will allow a practical test to be made of the validity of the current set of assumptions, and an opportunity to make explicit any additional implicit assumptions found. It is also intended to advance our understanding of and test the idea of using GSS-API for URI under close to real conditions.

The objectives of this implementation have been deliberately limited to ensure that information gathering remains its main objective. It is intended as an exploration in code, using implementation as a way of exploring the environment within which the URI security architecture is to be specified..

2 Background

The work performed so far at Cambridge for the URI security architecture has consisted in the main of a high-level examination of the issues involved in such an architecture. This has resulted in a draft of a list of assumptions to guide implementers of the architecture, and in the selection of GSS-API as an interface to security mechanisms for a prototype implementation of the architecture. In addition, some network infrastructure support preparation has been performed.

The broad objective of the work described here is to test this existing work in an environment which is intended to be a model of the real world in which production implementations will have to be developed and operate.

3 Rationale: Objectives of this Work

The detailed objectives of the prototype implementation are as follows:

- to gain practical experience in the requirements of a secure management system of a multi-service network, to inform the selection of issues and information for inclusion in the security architecture,

- to test the assumptions made so far for reasonableness, usability and completeness in a live implementation and operation environment, and allow us to modify these existing assumptions as required,
- to test the usefulness and limitations of GSS-API in distributed network security applications,[1] possibly resulting in proposals for modifications to the API and at least partial implementations of those modifications, and
- to test the features of a distributed programming support environment (ANSA) in the context of a security architecture – does it support or hinder the implementation of such an architecture?

It is intended that the conclusions and results of the prototyping work be as generally useful and applicable as possible.

In addition to these general objectives and results, if it is found that the use of GSS-API without modification gives sufficient functionality for the URI security architecture, the prototype may itself become a tool for evaluating GSS-API-compliant security protocols and products for usability and protection in a network management environment.

4 Setting Security-Related Priorities

There are a number of differing requirements of a secure network service – integrity, authentication/authorisation, protection against denial of service, and confidentiality. Due to our desire to limit the objectives of an initial implementation, these requirements have been prioritised for the prototype work.

Integrity in network control message passing, ensuring that the messages passed between a human or automatically-operated network controller and an ATM switch to control it are transmitted completely and correctly, is a key security requirement. Additionally, these messages need to be validly *authorised* and originated by *authenticated* individuals. Secure *logging* of audit trails is vital to determine who was responsible for any particular action, especially in a cross-security-domain network management environment. These security aspects will be given the highest priority.

Functionality to support other aspects of network control security may be introduced into the prototype at a later stage. These include protection against *denial of service* attacks,[2] and the preservation of *confidentiality.* The prototype's design will however take account of the potential need for later inclusion of these additional requirements.

5 The Prototype Described

5.1 Functionality of the Prototype

The network functionality selected for the prototype is the implementation of security mechanisms over the remote control of permanent virtual circuit setup and teardown on an ATM switch. The detailed functional requirements for the prototype will arise from the need to control the ATM switch in a secure manner, and the non-functional requirements from the existing version of the assumptions document. Relevance to the TINA architecture will also be a major consideration in the design of the software.

[1] Does it, for instance, provide sufficient functionality to support the security required of a network management system, and if not, what are the additional requirements?

[2] In an extreme case, denial of network management service in a public ATM network could result in a loss of public access to the emergency services.

The implementation work will be based on a client-server model as supported by the ANSA Real-Time architecture and software tools. The prototype will be implemented initially within the Computer Laboratory at Cambridge, and will be tested later across a multi-site ATM network.

5.2 Focus of the Prototype Implementation

The prototype is not intended to be a real world implementation of the URI security mechanisms. It is seen as an exploration in code of the environment and required functionality of these mechanisms. The specific areas in which the requirements of a full implementation will be constrained in the interests of learning at minimum cost include:

- the provision of only a subset of the range of functions needed for remote ATM switch control;
- the generation of clear and working rather than efficiency-optimised, code. Since the conclusions to be drawn from the prototype are intended to be generally applicable, its development will be performed in such a way as to avoid implementation- or tool-specific tricks or abuses. This approach is likely to result in less efficient code, but more useful results in the context of the objectives of the implementation work; and
- the modelling rather than full implementation of some aspects of the security environment, such as
  - the use in the first instance of dummy rather than fully operational security mechanisms responding to GSS-API requests, removing the need to set and maintain a security-related database,
  - the assumption for the prototype that an ATM switch, which might in fact have many workstations attached each of which can control it, is actually controllable only from one of these workstations; this one workstation would be seen as a model of either a unique gateway to the switch's management facilities or of a general-purpose processor forming a part of the switch itself, and
  - the omission of code to handle correctly some aspects of fault and error handling – it may be sufficient for the code to capture these conditions and report failure.

As a part of the post-implementation review, it will be necessary to consider the significance of each case in which a feature is emulated rather than implemented, to ensure that insights relevant to the project are not lost.

Whilst the prototype's functionality is related to network management, the requirements of customer network use will also be borne in mind when the results of the prototype are considered.

6 Summary

The next work to be performed towards Work Package 5 (Security) of the BT URI Management of Multiservice Networks project will be a prototype implementation, based on the current architectural assumptions and designed to elicit knowledge concerning how required security mechanisms will interact with the URI environment. Work will concentrate on accurately representing the major features of a real world implementation rather than achieving complete functionality.