Using Encryption for Authentication in Large Networks of Computers

Roger M. Needham and
Michael D. Schroeder
Xerox Palo Alto Research Center

Use of encryption to achieve authenticated communication in computer networks is discussed. Example protocols are presented for the establishment of authenticated connections, for the management of authenticated mail, and for signature verification and document integrity guarantee. Both conventional and public-key encryption algorithms are considered as the basis for protocols.

Key Words and Phrases: encryption, security, authentication, networks, protocols, public-key cryptosystems, data encryption standard
CR Categories: 3.81, 4.31, 4.35

Introduction

In the context of secure computer communications, authentication means verifying the identity of the communicating principals to one another. A network in which a large number of computers communicate may have no central machine or system that contains authoritative descriptions of the connected computers, of the purposes for which they are used, or of the individuals who use them. We present protocols for decentralized authentication in such a network that are integrated with the allied subject of naming. There is minimal reliance on network-wide services; in particular there is no reliance on a single network clock or a single network name management authority.

Three functions are discussed:

(1) Establishment of authenticated interactive communication between two principals on different machines. By interactive communication we mean a series of messages in either direction, typically each in response to a previous one.

(2) Authenticated one-way communication, such as is found in mail systems, where it is impossible to require protocol exchanges between the sender and the recipient while sending an item, since there can be no guarantee that sender and recipient are simultaneously available.

(3) Signed communication, in which the origin of a communication and the integrity of the content can be authenticated to a third party.

Secure communication in physically vulnerable networks depends upon encryption of material passed between machines. We assume that it is feasible for each computer in the network to encrypt and decrypt material efficiently with arbitrary keys, and that these keys are not readily discoverable by exhaustive search or cryptanalysis. We consider both conventional encryption algorithms and public-key encryption algorithms as a basis for the protocols presented.

We assume that an intruder can interpose a computer in all communication paths, and thus can alter or copy parts of messages, replay messages, or emit false material. While this may seem an extreme view, it is the only safe one when designing authentication protocols.

We also assume that each principal has a secure environment in which to compute, such as is provided by a personal computer or would be by a secure shared operating system. Our viewpoint throughout is to provide authentication services to principals that choose to communicate securely. We have not considered the extra problems encountered when trying to force all communication to be performed in a secure fashion or when trying to prevent communication between particular principals in order to enforce restrictions on information flow.

Our protocols should be regarded as examples that expose the authentication issues in large networks rather than as fully engineered solutions to the overall security problems of a particular application. While providing an adequate solution to the authentication problems specified and meeting most common security objectives, our protocols would need elaboration to meet other security goals such as preventing traffic analysis, withholding all matching cleartext-ciphertext pairs from an eavesdrop-

Authors' present addresses: R.M. Needham, University of Cambridge Computer Laboratory, Corn Exchange Street, Cambridge, England; M.D. Schroeder, Xerox Palo Alto Research Center, 3333 Coyote Hill Road, Palo Alto, California 94304.

per, and ensuring instantaneous detection of tampering, and also to maximize efficiency in particular networks. It is possible to devise other protocols similar to those presented that also meet the stated objectives.

There is a modest amount of literature on our subject, and methods have been proposed for several of the individual functions we describe [1, 3, 5, 6], although no work is reported that integrates these techniques and applies them in a decentralized environment, or that provides functionally equivalent protocols based on both conventional and public-key encryption.

1. Encryption Algorithms

The important difference between conventional and public-key encryption algorithms is the way keys are used. With a conventional encryption algorithm, such as the NBS Data Encryption Standard [7], the same key is used for both encryption and decryption. Authentication depends upon the two participants in a conversation being the only two principals (apart possibly from trusted servers) who know the key that is being used to encrypt the transmitted material. With a public-key encryption algorithm, a concept originated by Diffie and Hellman [3], two keys are necessary: one that is used in the conversion of cleartext to ciphertext, and another that is used in the conversion of ciphertext to cleartext. Furthermore, knowledge of one key gives no help in finding the other, and the two keys will act as inverses for each other. Elegant systems may be devised in which each principal has one public key and one secret key. Anyone may encrypt a communication for A using his public key, but only A can decrypt the result using his secret key. Likewise, only A can encrypt messages that will decrypt sensibly with A's public key. The first example of a public-key encryption algorithm was devised by Rivest et al. [9], and others are sure to follow.

2. Authentication Servers

With both kinds of encryption the basis of authenticated communication is a secret key belonging to each principal using the network, and there is need for an authoritative source of information about these keys. We use the term *authentication server* for a server that can deliver identifying information computed from a requested principal's secret key.

Since the main database of an authentication server is indexed by name, the management of authentication servers is related to the management of names. In an extended network it is inexpedient to have a single central name registration authority, so we suppose that there are multiple naming authorities, each of which assigns and cancels names as it wishes. With this organization, principals have names of the form "NamingAuthority.SimpleName." Associated with each naming authority are one or more name lookup servers and one or more authentication servers.[1]

A name lookup server is prepared to provide various network addresses associated with a given SimpleName, for example, the address of that principal's mail system buffer. One or more instances of a master name lookup server will provide the network addresses of appropriate name lookup and authentication servers when given a naming authority's name. Authentication servers perform strikingly similar functions for the two classes of encryption algorithms; the differences will be brought out as they arise.

3. Means of Encryption

One significant issue in this area of study is where the encryption and decryption are done. Branstad [2] suggests that these actions take place in the network interface of a computer. It is a requirement of some of our protocols that the encryption be done elsewhere, because it is necessary to prepare an encrypted message without actually sending it yet or to receive an encrypted message without knowing at the network interface what the key is. Accordingly we have assumed that any hardware encryption aid is located so one can say

$X :=$ encrypt(Y, Key)

and still have X in hand, or say

if ($X :=$ decrypt(Y, $Key1$)) = nonsense
then $X :=$ decrypt(Y, $Key2$) **fi**

4. Protocols for Establishing Interactive Connections

Protocol 1. With Conventional Algorithms

If a conventional algorithm is used then each principal has a secret key that is known only to itself and to its authentication server, the contents of which are accordingly secret. The essential step in setting up secure communication between A and B is for the initiator, say A, to generate a message with two properties:

(a) It must be comprehensible only to B, i.e. allow only B to use its contents to identify himself to A.

(b) It must be evident to B that it originated with A.

The use of encryption to achieve these properties was first described by Feistel [4] and applied to a network context by Branstad [1].

[1] Naming authorities are independent of network topology; they need have nothing to do with subnetworks or with particular computers on the network. Multiple identical name lookup servers and authentication servers for a single naming authority may be used to make sure that these services are topologically "close" to those needing to use them, and to enhance reliability. Our multiple authentication servers must be carefully distinguished from those proposed by Diffie and Hellman [3], which perform the quite different function of checking one another. In that case every user is registered with every authenticator, the aim being to defend against corruption of particular authenticators.