The implementation of a cryptography-based secure office system

by CHRISTIAN MUELLER-SCHLOER
Siemens Corporation
Cherry Hill, New Jersey
and
NEAL R. WAGNER
Drexel University
Philadelphia, Pennsylvania

ABSTRACT

A cryptography-based secure office system is discussed, including design criteria and a specific implementation. The system is intended to be practical, simple, and inexpensive, but also highly secure. The implementation uses a hybrid scheme of conventional (DES) and public-key (RSA) cryptography. Randomly generated DES keys encrypt messages and files, and the DES keys themselves and a one-way hash of the messages are encrypted and signed by RSA keys. The system provides secure electronic mail (including electronic registered mail and an electronic notary public), secure two-way channels, and secure user files. Timestamps and a special signed file of public keys help decrease the need for an online central authority involved in all transactions.

INTRODUCTION

This article discusses the design and experimental implementation of a secure data processing system for an office or similar organization. The basic design should be usable by any number of users (from one to thousands or more), and the hardware may be distributed. We have attempted to design a practical, simple, inexpensive system whose underlying method of implementation is transparent to users. However, the principal design goal has been security. We also wanted to limit the need for an online (i.e., "always-present") central authority.

We visualize users each having access to a *personal work station* (PWS), i.e., a station with local computing power. While in use the hardware of each PWS will be assumed secure, but no security is assumed for the connections between PWSs or for an individual PWS when not in use. Users are not limited to a single PWS, but may sign on at any convenient one.

The system described here is intended as a supplement to the functions normally provided by operating systems and network managers. Our system provides the following basic functions:

1. Secure electronic mail, including electronic registered mail and an electronic notary public (one or more users who can authenticate a signature, provide a timestamp, and save a copy of the message)
2. Secure two-way communications channels (to give simultaneous interactive dialogue)
3. A secure user file system

Such a secure distributed system requires some use of cryptography.[1, 2] In order to achieve simplicity and low cost along with high security and ease of use, we used a hybrid system of conventional and public-key cryptography.

FUNCTIONAL DESCRIPTION

First we list the system's user-level security-related commands, suppressing other commands needed for an electronic mail system, such as "SearchMailbox," etc. The commands are independent of the particular form of implementation (whether conventional, public-key or hybrid like ours), and the user need not know anything about cryptography.

1. SignOn. (Input: Username, Password. Result: User is authenticated by the Password, and the PWS is initialized and thereby dedicated to the user.)
2. SignOff. (Input: None. Result: The PWS is deinitialized by overwriting sensitive areas and keys, etc.)
3. NewUser. (Input: Username, Password. Result: A new user is enrolled in the system with the password as the means of subsequent authentication.)
4. UpdatePublicKey. (Input: Username, Password. Result: A new public and secret key pair is substituted for the old.)
5. UpdatePassword. (Input: Username, New Password, Old Password. Result: A new password is substituted for the old.)
6. SendSecure. (Input: Destination-name, File [i.e., a "message"]. Optional input: Intermediate-destination-name, Request to register or notarize. Result: The file is timestamped, signed, and encrypted for the user whose name is Destination-name. In case of optional input, the file will first be routed to Intermediate-destination-name for registration or notarization.)
7. ReceiveSecure. (Input: Sender-name. Result: File from user implied by Sender-name is decrypted and authenticated.)
8. Register. (Input: Destination-name, Encrypted file. Result: The file is timestamped, signed, and forwarded.)
9. Notarize. (Input: Destination-name, Encrypted file. Result: The file is timestamped and signed, and a copy is retained before forwarding.)
10. AcknowledgeSecure. (Input: Sender-name, Encrypted file. Result: The file is timestamped, signed, and sent back to sender. This is for use with registered mail.)
11. OpenSecure. (Input: Destination-name. Result: A secure channel is created for immediate interactive use.)
12. CloseSecure. (Input: Destination-name. Result: The secure channel is closed.)
13. SaveSecure. (Input: Filename. Result: The file is saved in the user's mass storage in a secure way.)
14. RestoreSecure. (Input: Filename. Result: The saved encrypted file is made available as unencrypted cleartext.)

Initially, users must give the "NewUser" command with a password that they can remember. "NewUser" requires the physical presence of users at the central authority if authentication that a username corresponds to a particular physical individual is desired. "SignOn" must be given with a password matching the username. Messages or files of any sort can be sent to other users with the "SendSecure" command and can be received with the "ReceiveSecure" command. Various encryptions, decryptions, signatures, timestamps, and authentication steps are built into these commands at a lower level, as described below.