AN OVERVIEW OF THE ADVANCED SMARTCARD ACCESS CONTROL SYSTEM (ASACS)

Jim Dray <dray@st1.ncsl.nist.gov>
Computer Security Division / Computer Systems Laboratory
National Institute of Standards and Technology
Gaithersburg, MD 20899

David Balenson <balenson@tis.com>
Trusted Information Systems, Inc.
Glenwood, MD 21738

ABSTRACT

The Advanced Smartcard Access Control System (ASACS) was developed by the National Institute of Standards and Technology in conjunction with Datakey and Trusted Information Systems. The system includes a smartcard with public key capabilities and a portable reader/writer with computational capabilities, including a microprocessor, programmable memory, a keypad, and an LCD display. Through the use of a layered interface, ASACS was integrated into several demonstration programs and into the TIS Privacy Enhanced Mail (TIS/PEM) system. This paper provides a brief overview of the ASACS.

INTRODUCTION

Computer access control systems which rely solely on password-based authentication have proven to be inadequate in many environments, particularly where network systems are involved. The security of access control systems can be significantly strengthened if the authentication process is based on something the user possesses, such as a smartcard, in addition to a memorized password or Personal Identification Number (PIN). Modern smartcards have the ability to process as well as store information, and this capability has significant advantages over passive memory card technology for security applications. Smartcards can implement secure cryptographic authentication and automated key distribution protocols, provide secure data storage, and perform a variety of other functions which increase the security of an access control system. This increase in security can be realized while maintaining or even enhancing the level of convenience for the system user.

The Advanced Smartcard Access Control System (ASACS) has been developed by the National Institute of Standards and Technology in conjunction with Datakey and Trusted Information Systems. The primary goal of the project was to develop an advanced smartcard system which exploits recent advances in semiconductor and cryptographic technology for secure login authentication. ASACS also provides secure data storage, automated key management, and digital signature capabilities. The services supported by the ASACS implementation are designed for use within networking environments, including both local area networks and wide area networks such as the Internet.

The ASACS smartcard provides cryptographic capabilities based on standard cryptographic algorithms and techniques, in combination with software running on a host computer. Many of the underlying concepts applied to the design of ASACS have been successfully demonstrated in the NIST/Datakey Token Based Access Control System (TBACS) [1] as well as the Secure Access Control System (SACS) [2] projects. Each of these systems provides token-based secure access to a host computer through a cryptographic handshake protocol based on the Data Encryption Standard (DES) algorithm. However, the ASACS project involves the development of a smartcard with greater capabilities through the addition of public key cryptographic functions. A new smartcard reader/writer with significantly greater capabilities has also been developed for ASACS. The ASACS reader/writer has

This work partially sponsored by the U.S. Government Defense Advanced Research Projects Agency (DARPA) under contract number 8139 MOD 01 to the National Bureau of Standards and under contract number F30602-89-C-0125 to Trusted Information Systems.

computational capabilities, and includes a microprocessor, programmable memory, a keypad, and an LCD display. These features support the needs of mobile users who require a portable reader/writer for authentication from remote sites. To demonstrate the capabilities of ASACS, several applications have been developed, most notably a system maintenance program and several other useful demonstration programs. In addition, ASACS has been integrated with the TIS Privacy Enhanced Mail (TIS/PEM) system.

SYSTEM OVERVIEW

Figure 1 depicts the ASACS system components. A user possessing a smartcard inserts the card into the reader/writer which is attached to a local workstation. The workstation is connected to a local area network (LAN), which in turn may be connected to other networks. The smartcard may be used to control the user's access to both the local workstation as well as to other workstations and host computers on the attached networks.

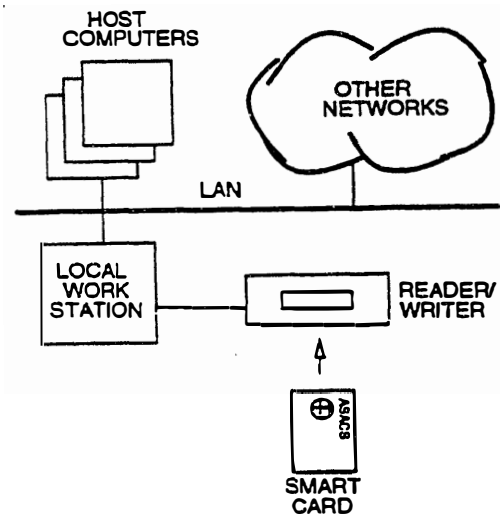

Figure 1: ASACS system components.

From an architectural standpoint, ASACS is divided into several different functional layers, comprising both the hardware and software components of the system (see Figure 2). The lowest layer consists of the ASACS hardware, including the public key smartcard and either the SACS reader/writer or the ASACS portable reader/writer. The next layer of ASACS is comprised of host system software, which is functionally divided into four layers. This software is used to provide to provide a convenient and standard method for integrating the ASACS public key smartcard into a wide variety of host system application software. The top layer is a Smartcard Application Program Interface (SCAPI) which is directly accessed by applications software to interface with the ASACS system. The other layers provide command set interfaces for the smartcard commands and the reader/writer commands, a smartcard communications protocol, and hardware-level I/O support.

Finally, the top layer of ASACS represents the various applications with which the ASACS system can be integrated. ASACS can be integrated into these applications using either the SCAPI or the command set interfaces. A security officer maintenance program and several demonstration programs, including a signature utility program and a login manager were developed as a part of the ASACS project. In addition, using the SCAPI, the ASACS system has been integrated into the TIS Privacy Enhanced Mail (TIS/PEM) system.

PUBLIC KEY SMARTCARD

The ASACS smartcard is based on the Smartcard-based Access Control System (SACS) developed by NIST under a previous DARPA sponsored contract. The SACS and ASACS smart cards contain a Hitachi H8/310 integrated circuit, designed specifically for smart card applications [3]. The H8 is configured with 256 bytes of RAM, 10K bytes of ROM, and 8K bytes of EEPROM. In order to meet ISO requirements for contact spacing and arrangement, the H8 die has pads for power (+5V), ground, clock (10MHz), reset, and serial I/O [4]. An ISO-standard micromodule is bonded to the H8 die, and this assembly is then mounted in a plastic card with the same dimensions as a standard credit card.

Smartcard Firmware

The ASACS public key smartcard firmware implements a set of commands which support card maintenance, key management, user authentication,

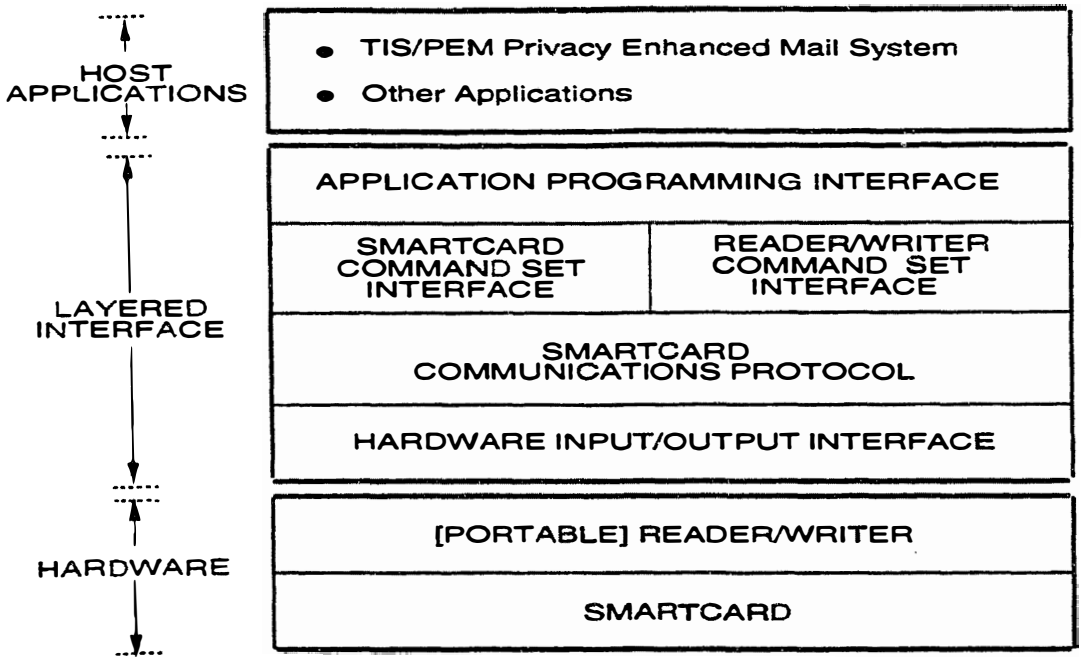

Figure 2: ASACS functional layers.

data storage, and data encryption and authentication. Access control software running on a host computer issues commands to the smartcard through the reader/writer interface. The firmware of the card then executes the requested function and returns the appropriate response to the host computer. It is the responsibility of the host access control software to mediate the authentications between the user, the user's smartcard, and the host computer.

The ASACS command set is the successor to the smartcard command set developed for the Smartcard-based Access Control System (SACS). The cost and time constraints of the ASACS project did not allow for the production of a new ROM mask. Therefore, the ROM mask developed for the SACS project was also used for the ASACS smartcard. ASACS retains the symmetric key capabilities of the original SACS system, since the authentication protocol is based on the Data Encryption Standard (DES) algorithm. This challenge-response authentication protocol provides a rapid and secure method for two parties to perform mutual identity verification based upon the possession of a shared secret key and the use of that key to encrypt randomly generated cryptographic challenges. This protocol is described in detail in NIST Special Publication 500-157 [5]. The ASACS smartcard is capable of accepting or generating the initial cryptographic challenge, and therefore complies with the requirements of ANSI X9.26 [6] for secure sign-on.

The principal difference between the ASACS and SACS command sets is the addition of public key cryptographic capabilities. There are certain arithmetic operations, such as modular exponentiation and modular multiplication, which are common to a variety of public key algorithms. These operations have been implemented in the ASACS firmware as distinct routines which can be used to support most of the currently available public key algorithms. The development and optimization of firmware which performs these modular operations is the most difficult aspect of implementing public key cryptography on a smartcard. A variety of public key algorithms can be realized in the ASACS smartcard firmware by calling the low-level arithmetic routines in the required sequence. Both the Digital Signature Algorithm (DSA), which has been proposed by NIST as a Digital Signature Standard (DSS) [7], and the

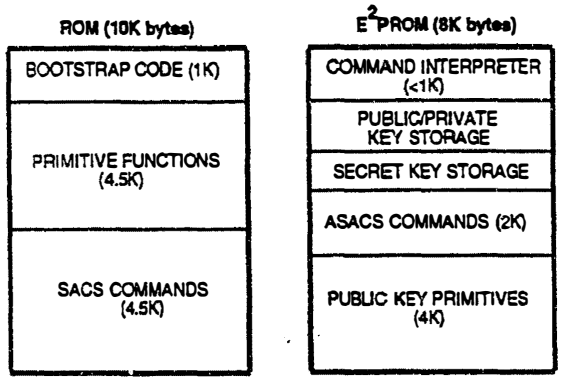

Figure 3: ASACS smartcard memory layout.

Rivest-Shamir-Adleman (RSA) [8] cryptographic algorithm have been implemented in the ASACS smartcard firmware.

Figure 3 depicts the layout of the ASACS smartcard memory from a high level perspective. The majority of the firmware is stored in ROM, including a bootstrap routine and code for the commands from the SACS smartcard. The Data Encryption Standard (DES) [9] algorithm is also located in ROM. The EEPROM contains the firmware for the public key algorithms, a command interpreter, and a jump table which points to the firmware routines associated with each command. Since the addresses in the jump table can be modified, new firmware routines can be loaded into EEPROM to replace existing routines and to add new functions. Specific locations in EEPROM are reserved for the storage of symmetric and asymmetric key components. In addition, a number of general purpose data storage zones are available in EEPROM.

See [10] for a more detailed description of the ASACS public key smartcard.

SMARTCARD READER/WRITER

The ASACS public key smartcard can be interfaced to a workstation using either the SACS reader/writer or the new ASACS portable reader/writer. Both the SACS and the ASACS reader/writers provide an RS-232 serial communications connection between the smartcard and the host computer. RS-232 was chosen because a serial port is standard equipment on the majority of computers. Therefore, the reader/writer can be connected to most computers without the need for a custom interface or hardware modifications.

SACS Reader/Writer

The SACS reader/writer is a relatively unsophisticated device which simply serves as a direct I/O interface between the smartcard and a host. It cannot perform any processing itself since it does not contain a microprocessor. Its main purpose is to provide power, ground, clock and I/O signals to a SACS or an ASACS smartcard. To interface the smartcard to the host, the reader/writer performs level conversion between the 12V RS-232 I/O signals used by the host and the 5V I/O signals used by the card. See [11] for a more detailed description of the SACS reader/writer.

The SACS reader/writer features an ISO standard smartcard receptacle, external power and data indicator lights, and an RS-232 port for connecting to a host. In addition, the SACS reader/writer's card receptacle features a locking mechanism which holds the card internally after insertion into the reader/writer, and an automatic ejection mechanism to remove the card from the reader/writer.

An RS-232 cable is required to attach the SACS reader/writer to a host, whereupon it functions as data communications equipment (DCE). Signals are sent by the reader/writer to the host which indicate that the reader/writer is powered-up and that a card is inserted. The SACS reader/writer is a rectangular box approximately 2 1/2 inches high, 5 inches deep, and 5 inches wide. An ISO smartcard receptacle and indicator lights are located on the front of the reader/writer, and the power cord and RS-232 jacks in the rear. The power supply for the SACS reader/writer is internal.

The SACS reader/writer is designed to accept a smartcard whose physical characteristics, dimensions and contact locations adhere to ISO International Standard 7816, Parts 1 and 2 [4,12]. The electrical signals that the SACS reader/writer supplies to the smartcard also meet most of the requirements specified in ISO International Standard 7816, Part 3 [13], with the exception of the initial clock (CLK)

frequency, which is 10MHz as opposed to 3.5795.

ASACS Portable Reader/Writer

The ASACS portable reader writer was built to provide functionality not offered by the earlier SACS reader/writer. As a portable device, it allows users the option to authenticate themselves using hosts not equipped with a smartcard reader/writer. Several significant improvements have been made to the design of the reader/writer. The overall size has been reduced to less than half that of the SACS reader/writer, so that the device can easily be carried for use at remote sites. The new reader/writer is powered by rechargeable batteries, and includes a transformer for use with 110V line power. The front panel has a keypad and liquid crystal display which allow the user to interact directly with the smartcard. This feature is useful in situations where the reader/writer cannot be connected to the user's workstation. A protocol has been developed which allows the user to perform authentications manually via the keypad and display. A remote host computer can then require manual ASACS authentication even if the user's workstation is a dumb terminal. In this case, all interactions with the card are through the keypad and display. After the user personal identification number (PIN) has been submitted to the card, the remote host will generate a random challenge and send this to the user's workstation. The user reads this challenge from the screen and types it on the reader/writer keypad. The smartcard encrypts the challenge and displays the encrypted result, so that the user can submit it to the remote host. When a serial connection to the workstation is available, the user still has the option of entering the PIN through the keypad on the reader/ writer. Since the user's PIN does not travel through the workstation, system security is enhanced.

The ASACS reader/writer has an 8-bit microprocessor with 256 bytes of internal RAM. In addition, the reader/writer has 256 bytes of EEPROM used for data and setup parameter storage, 32K bytes of RAM used for scratch pad and data buffering, and an industry standard 32K byte EPROM chip which holds firmware implementing the internal logic and external commands. The EPROM chip can be easily removed for custom firmware development. See [14] for detailed specifications for the ASACS portable reader/writer and firmware.

The reader/writer supports a set of commands that are executed directly on the reader/writer, as opposed to on the smartcard. These commands use the same protocol that is used for smartcard commands. Several of the reader/writer commands allow the host to load the default parameters into the reader/writer's non-volatile memory to control such things as baud rate, and the date/time. These same default values can also be specified manually from the keypad by pressing the F1 key to access the reader/writer's set-up menu. Another command can be used by the host to determine if a smartcard is inserted into the reader/writer. Two commands can be used to temporarily put the reader/writer in manual keypad entry mode. The first of these two commands, as discussed above, is used by the host to allow the user to enter their PIN to the smartcard via the reader/writer's keypad. The latter command can be called to allow the user to perform a manual challenge/response with a remote host. The remaining reader/writer commands can be used by the host to utilize the ASACS reader/writer's communications buffer for more efficient DES encryption, DES decryption or MAC calculation with the smartcard.

SMARTCARD LAYERED INTERFACE

The ASACS host system software is comprised of a set of four interface layers. Each layer corresponds to a specific set of functions needed to integrate the ASACS system into a software application on a host system (see Figure 2).

Smartcard Applications Program Interface

The Smartcard Application Program Interface (SCAPI) [15] was developed to provide a consistent, but robust interface designed to ease the integration of smartcard technology into applications. The SCAPI is intended to insulate applications from the differences among the various smartcards, as well as differences likely to appear as smartcard technology evolves. The SCAPI is not tied to specific smartcards or to specific capabilities (e.g., memory capacity) of smartcards. In fact, the SCAPI can be, and has been, completely implemented in software, thus providing a simple, but useful tool for integrating smartcard technology into applications. The functional capabilities of a particular smartcard

determines how much of the SCAPI functionality is implemented in software on the host computer and how much is performed on the smartcard. Thus, as technology advances, more of the SCAPI functionality may be directly implemented on the card or on the reader/writer while leaving applications unaffected.

The SCAPI currently defines four types of functions:

- Initialization Functions,
- Account Functions,
- Cryptographic Functions, and
- File and Directory Functions.

The SCAPI is intended to be consistent with the ANSI C standard. The file functions are designed to map directly upon those defined by Kernighan and Ritchie [16]. Since C is known for its portability, it makes sense to extend this platform independence to smartcard systems. Further, this flexibility and consistent feel for C programmers is likely to promote the use of the SCAPI. The directory functions reflect widely used operating system calls. Unfortunately, ANSI C does not address the cryptographic functionality to which smartcard technology is so well-suited. Therefore, the SCAPI defines a set of cryptographic functions which provide an algorithm-independent interface for cryptographic operations which may be implemented on a smartcard.

Smartcard and Reader/Writer Command Set Interfaces

The Command Set Interface Layer consists of C language object module libraries. The libraries each provide a set of C function calls, each directly corresponding to a command from the firmware command sets for the public key smartcard [17] and the portable reader/writer [18]. The function which represents a particular command is called with the appropriate input data for that command as arguments. The function returns the output data from the command and a status code. Status codes are mapped onto a set of error messages defined in a header file. This layer is called indirectly through the SCAPI, thus making the choice of reader/writer invisible to the application.

Communications Protocol and Hardware I/O Interface

The Smartcard Communications Protocol Layer transmits the data assembled by the Command Set Interface Layer to the ASACS portable reader/writer and the public key smartcard. The data is transmitted according to the communications protocol used by both the reader/writer and the smartcard. The Communications Protocol Layer interacts with the Hardware I/O Interface in order to send and receive each byte of the
data.

The Hardware I/O layer consists of a software driver which provides low-level input/output routines for communicating with the smartcards. Currently, the Hardware I/O Layer consists of a serial interface, since both the SACS and ASACS reader/writers employ serial interfaces. This layer can support other types of hardware interfaces for reader/writers that do not employ an RS-232 interface.

The Serial I/O Interface is written to be as portable as possible across a broad range of hardware/software platforms, such as SUNOS (Sun's UNIX Operating System) and MSDOS. However, some systems may require that this layer be customized. The interface to this layer is clearly defined, and can be modified with minimal effort.

APPLICATIONS SOFTWARE

Security Officer Maintenance Program

The Security Officer Maintenance (SOMAINT) Program [19] provides functions which are used by a security officer or system manager. These functions include the initialization of cards for new users, synchronization and maintenance of key databases stored on the cards and host computers, deactivation of cards, and reactivation of cards which have been inadvertently deactivated or corrupted. The programs which support the system management functions are restricted to use by authorized security managers through the standard UNIX operating system file protections.

Signature Utility Program

The DSS Signature Utility Program [20] was developed to demonstrate the generation and verification of digital signatures using the ASACS public key smartcard. The program utilizes the algorithm proposed by the Standard Hash Standard (SHS) [21] to calculate a hash value on a file of arbitrary size. The hash value is transmitted by the host computer to the smartcard, which applies the Digital Signature Algorithm (DSA) to this value to generate a digital signature with the cardholder's private key. The signature can then be verified by the host computer or the smartcard using the cardholder's public key.

Login Manager

The ASACS Login Manager [22] is a collection of programs which control login access to host computers. These programs manage the series of authentications between the user, the smartcard, and a host computer. When a user requests access to the host, the login manager establishes communications with the user's card through the reader/writer. The login manager prompts the user for the user PIN, and transmits it to the card in order to authenticate the user to the card. The card and host will then authenticate to each other using a random challenge-response protocol based on the Data Encryption Standard (DES). This protocol provides a means for rapid authentication of two parties with protection from wiretapping and playback attacks. If the authentications are successful, the user is granted a session on the host.

The login demonstration software also supports login authentication to remote host computers. When a system user wishes to access a remote computer, the user executes a program which communicates with the user's card to obtain a list of host computers with which the card shares authentication keys. This list of host computer names is displayed in a menu, so that the user can select the particular host to access. The software establishes a connection with the ASACS authentication server process running on the remote host selected by the user. The remote host then performs the challenge-response authentication with the user's card in order to verify the identity of the user.

Privacy Enhanced Mail

The Internet Privacy Enhanced Mail (PEM) protocols are an extension to the existing Internet electronic mail protocol (RFC 822) which provide simple end-to-end security services including optional message confidentiality, message integrity, and source authentication with non-repudiation. The protocols are specified in a 4 part series of specifications [23,24,25,26] which are currently published as Internet Drafts, and are targeted to be published as Internet Request For Comments (RFCs) with Proposed Standard status.

The PEM security services are provided through the use of standard cryptographic techniques, including message encryption using the DES in the Cipher Block Chaining (CBC) mode of operation to protect message text and the RSA algorithm to provide for distribution of DES keys, digital signatures using RSA algorithm in conjunction with either Message Authentication Code (MAC), Message Digest Algorithm MD2 [27], or the Message Digest Algorithm MD5 [28]. RSA public keys are managed as public key certificates using a distributed certification hierarchy based on CCITT X.509 [29].

The TIS Privacy Enhanced Mail (TIS/PEM) System is a UNIX-based implementation of PEM. At the core of the TIS/PEM system is the Local Key Manager (LKM), which, as its name implies, is responsible for all the local key management activities on a multi-user host system. This includes (1) maintaining a database for local users' private keys, (2) controlling the use of private keys to compute digital signatures and decrypt message tokens (encrypted message encryption keys), (3) maintaining a database for local and remote users' public key certificates, and (4) providing access to validated public key certificates. In addition, the LKM shares the responsibility for the registration of a local user, that is, the generation of a public/private key pair and the construction and digital signing of a certificate embodying the public key.

The ASACS system was integrated with the TIS/PEM system by integrating it with the LKM. In particular, a user's private key is generated by the LKM and then stored on the smartcard, where it remains in the protected confines of the smartcard. When called upon to perform the cryptographic operations involving the user's private key, the LKM, instead of

performing those operations directly, now invokes the functions of the smartcard via the SCAPI. The smartcard then performs the necessary computation of a digital signature or decryption of a message token, using the private key stored on the smartcard.

The storage of a user's private key provides added protection that cannot be achieved in a shared database. The inherent security features of the smart card allow for limiting access to the private key to the user, who must be authenticated to the card before the private key can be used.

ACKNOWLEDGEMENTS

Lots of people at NIST, Datakey, and TIS have contributed to the design and development of ASACS. Some of the developers deserving special thanks include Tom Cain, Paul Clark, Steve Crocker, Mike Indovina, Gary Ostrem, Miles Smid, and Robert Warnar.

REFERENCES

1. Dray, James F., Miles E. Smid and Robert B. J. Warnar, Implementing an Access Control System with Smart Token Technology, National Institute of Standards and Technology, U.S. Department of Commerce, Washington, D.C., April 12, 1989.

2. NIST SACS Smartcard Specification, Datakey, Inc., Report #065-0097-000, July 11, 1991.

3. Hitachi H8/310 Single-Chip Microcomputer, Hitachi, Ltd., Tokyo, Japan, 1989.

4. International Standard 7816-2, Identification Cards - Integrated Circuit(s) Cards with Contacts -- Part 2: Dimensions and Location of the Contacts, International Organization for Standardization, 1988.

5. Haykin, Martha E., and Robert B. J. Warnar, Smart Card Technology: New Methods for Computer Access Control, NIST Special Publication 500-157, National Institute of Standards and Technology, U.S. Department of Commerce, Washington, D.C., September 1988.

6. American National Standard X9.26-1990, Financial Institution Sign-on Authentication for Wholesale Financial Systems, American Bankers Association, Washington, D.C., 1990.

7. Proposed Digital Signature Standard (DSS), National Institute of Standards and Technology, U.S. Department of Commerce, Washington, D.C., August 30, 1991.

8. Ronald L. Rivest and Adi Shamir and Leonard M. Adleman, A Method for Obtaining Digital Signatures and Public Key Cryptosystems, Communications of the ACM, Volume 21, Number 2, February 1978, pp. 120-126.

9. Federal Information Processing Standard Publication (FIPS PUB) 46-1, Data Encryption Standard, National Institute of Standards and Technology, U.S. Department of Commerce, Washington, D.C., Reaffirmed January 22, 1988 (Supersedes FIPS PUB 46, January 15, 1977).

10. ASACS Smartcard Specification, Datakey, Inc., Report #065-0130-000, April 24, 1992.

11. NIST SACS Reader/Writer Specification, Datakey, Inc., Report #065-0098-000, July 11, 1991.

12. International Standard 7816-1, Identification Cards - Integrated Circuit(s) Cards with Contacts -- Part 1: Physical Characteristics, International Organization for Standardization, 1987.

13. International Standard 7816-3, Identification Cards - Integrated Circuit(s) Cards with Contacts -- Part 3: Electronic Signals and Transmission Protocols, International Organization for Standardization, 1989.

14. ASACS Portable Reader/Writer Specification, Datakey, Inc., Report #065-0131-000, April 24, 1992.

15. Smartcard Application Program Interface for the Advanced Smartcard Access Control System (ASACS), Trusted Information Systems, Inc., Glenwood, MD, October 1992.

16. Kernigan, B. and D. Ritchie, The C Programming Language, 2nd Edition, Prentice Hall, 1988.

17. Advanced Smartcard Access Control System (ASACS): Smartcard Command Set Interface, National Institute of Standards and Technology, U.S. Department of Commerce, Washington, D.C., 1992.

18. Advanced Smartcard Access Control System (ASACS): Reader/Writer Command Set Interface, National Institute of Standards and Technology, U.S. Department of Commerce, Washington, D.C., 1992.

19. Security Officer Maintenance (SOMAINT) Program User's Manual, National Institute of Standards and Technology, U.S. Department of Commerce, Washington, D.C., 1992.

20. Advanced Smartcard Access Control System (ASACS): The DSS Signature Utility Program Manual, National Institute of Standards and Technology, U.S. Department of Commerce, Washington, D.C., 1992.

21. Proposed Secure Hash Standard (SHS), National Institute of Standards and Technology, U.S. Department of Commerce, Washington, D.C., January 22, 1992.

22. Advanced Smartcard Access Control System (ASACS): UNIX Access Control Software Manual, National Institute of Standards and Technology, U.S. Department of Commerce, Washington, D.C., 1992.

23. John Linn, Privacy Enhancement for Internet Electronic Mail: Part I -- Message Encipherment and Authentication Procedures, Internet Draft (draft-ietf-pem-msgproc-02.txt), Digital Equipment Corporation, July 23, 1992, (RFC in progress; will obsolete RFC 1113).

24. Stephen Kent, Privacy Enhancement for Internet Electronic Mail: Part II -- Certificate-Based Key Management, Internet Draft (draft-ietf-pem- keymgmt-01.txt), BBN Communications, August 6, 1992, (RFC in progress; will obsolete RFC 1114).

25. David Balenson, Privacy Enhancement for Internet Electronic Mail: Part III -- Algorithms, Modes, and Identifiers, Internet Draft (draft-ietf-pem- algorithms-01.txt), Trusted Information Systems, September 3, 1992, (RFC in progress; will obsolete RFC 1115).

26. Burt Kaliski, Privacy Enhancement for Internet Electronic Mail: Part IV -- Key certification and Related Services, Internet Draft (draft-ietf-pem- forms-01.txt), RSA Laboratories, September 1, 1992.

27. Kaliski, B., The MD2 Message-Digest Algorithm, Internet Request for Comments (RFC) 1319, April 1992.

28. Rivest, R., The MD5 Message-Digest Algorithm, Internet Request for Comments (RFC) 1321, April 1992.

29. CCITT Recommendation X.509, The Directory - Authentication Framework, The International Telegraph and Telephone Consultative Committee, November 1988.