THE APPLICATION OF SMART CARDS FOR RSA DIGITAL SIGNATURES IN A NETWORK COMPRISING BOTH INTERACTIVE AND STORE-AND-FORWARD FACILITIES

J.R. Sherwood and V.A. Gallo
Computer Security Limited
Brighton, UK

Abstract. *Smart card technology is relatively new but offers an economic and convenient solution to the problems of user-authentication. This paper discusses the requirements for user authentication and digital signature in complex networks and examines the problems of integrating a smart-card sub-system. It proposes some design approaches for providing a useful lifetime for a smart card and for handling the computations required for 512-bit RSA digital signatures.*

Environment

Many data communications networks are known to be based on interactive working between a user on a workstation and a remote central application on a mainframe host. Many other networks are based on store-and-forward facilities for message or file transfer using the mail-box principle. Increasingly, large corporate networks are offering both of these facilities in an integrated data communications network. At the same time, network services providers and users are becoming more conscious of the need to implement security in these environments.

Introduction

This paper addresses the problems of security sub-system design in a network environment of some complexity. It outlines a design approach which is suited to both interactive and store-and-forward working, bringing together a wide range of cryptographic techniques and system components. The paper explores some of the design considerations that are relevant to the development of an integrated solution for system security and provides a framework within which various cryptographic techniques can be interwoven. In particular, it discusses the applicability of each technique and examines the contribution that each can make to the overall design.

S. Goldwasser (Ed.): Advances in Cryptology - CRYPTO '88, LNCS 403, pp. 484-496, 1990.

Smart cards are relatively new and as yet there are few systems which exploit their potential as a user token providing an automated logon protocol. This is one area that the paper explores in some detail; we develop some existing ideas on the use of a one-way function to encrypt a random challenge for constructing a logon protocol and in particular some techniques are described for ensuring attack-resistance over the expected lifetime of the smart-card token.

Security Requirements

The type of network under consideration here comprises both interactive and store-and-forward mailbox facilities, based on personal computer workstations connected to mainframe computers.

Broadly speaking, the security requirements can be summarised as follows:-

1. *Access control over local workstations and their applications.*

2. *Access control over remote hosts and their applications.*

3. *Privacy of communications over data networks.*

4. *Integrity checks on the contents of communications (message authentication).*

5. *Proof of message origin.*

To achieve these requirements a security sub-system is required which integrates into all possible configurations of the network, and which is applicable to both store-and-forward and interactive environments. It must also be capable of integrating successfully with existing security sub-systems, such as that provided in IBM SNA networks [1] [2] involving the use of the SNA encrypt/decrypt facility, and such as the access control facilities of RACF [3] and ACF2 [4]. Where SNA encrypt/decrypt is employed, the host mainframes are either equipped with an IBM 3848 Cryptographic Unit [5] or with the IBM Programmed Cryptographic Facility Program Product [6] with ACF/VTAM [7].

System Requirements

The system component requirements for constructing a suitable security sub-system are as follows:

1. *User tokens*

2. *A suitable collection of cryptographic algorithms for implementing personal authentication, data encryption, message authentication and digital signature.*

3. *A cryptographic key management architecture which provides both security and manageability.*

4. *Cryptographic units (possibly with tamper resistance* [8] [9]*) or cryptographic programs.*

Additionally, the security sub-system should not unduly affect response times for network users nor should it present an unfriendly and complex user interface.

Security Sub-System Overview

The security sub-system architecture proposed here uses a smart card as the user token. This provides the basic mechanism for access control, and also stores user-specific cryptographic keys. Local access control is effected by PIN-protection of the smart card. Hence the user needs to possess both the card and knowledge of the PIN. Remote access control is achieved by means of a challenge-response protocol designed to be thoroughly resistant to cryptanalytic attack for the entire lifetime of the card. For the encryption of data and the generation of message authentication codes, symmetrical encryption algorithms such as DES [10] [11] are used. Digital signatures are generated using the RSA asymmetric public key algorithm [12], and each authorised signatory carries a personal RSA key pair as part of the data on the smart card user token [13].

Top-level key management protocols are also implemented using RSA, which has the advantage of enabling a fully automated and therefore very manageable key distribution scheme. [14]. To maintain acceptable response times the cryptographic units and smart card readers are all equipped with a digital signal processor providing "fast RSA" processing facilities. [15]

To overcome the problems of mutual trust between a smart card and an intelligent cryptographic smart card reader, one of two approaches is possible. The card and the reader can mutually authenticate one another using a zero-knowledge proof protocol such as Fiat-Shamir, [16] [17], or the card can delegate some of the heavier computations to the card reader without disclosing its secret information. [18]

System Detail

Local Access Control

Every workstation is provided with an integral smart card reader, into which an authorised user inserts a smart card. The smart card itself is only activated for further functions if the correct user PIN is supplied. The user is prompted by the local application to supply this PIN, which is then submitted to the smart card for validation. If the PIN validation is successful, the card may then enter into a mutual authentication process with the smart card reader using the Fiat-Shamir protocol. The card reader is equipped with a digital signal processor which performs all cryptographic processing in that unit. At this stage, the system has achieved the following authentication:

1. *User to smart card*

2. *Smart card to reader*

3. *Reader to smart card*

By implication, the user has also been authenticated to the reader, and hence to the application which is driving it. The smart card now provides the data required for remote access control and for digital signature using RSA.

Also stored on the smart card is a user privileges profile which is sent to the application and which controls the range of application facilities to which the user is to be granted access. If the privileges profile data is too great to store on the card it can be stored encrypted on the workstation database under a secret DES storage key which is generated and held on the card.

Remote Access Control

One-time passwords are used for logons which involve plaintext transmission across the network. This prevents an eavesdropper from capturing a useful password. For interactive working a challenge-response protocol [19] is used to authenticate the user to a remote host and its applications. If RACF or ACF2 are in use, these packages provide "exits" via which one-time password sub-systems can be interfaced. The one-time password is checked at the host and if valid, the user is granted access. In challenge-response mode this is achieved by the host security module generating a 128-bit random number, which is sent across the network to the workstation and from there to the smart card. A one-way function is now applied to the challenge to obtain the response. The smart card encrypts the random number under a secret user (DES) key which we shall call KU. The 128-bit ciphertext output from DES ECB mode [11] is then subjected to another algorithm which selects individual bits and combines them to form a 96-bit output. The mask for bit selection constitutes the user key for this selection algorithm. We shall call this key KS. There are 96 bits to be selected from 128. Hence the keyspace for KS is ${}^{128}C_{96}$ or 2^{218}.

The 96-bit output from the selection algorithm is transmitted back across the network to the host, where it is processed by the host security module to verify it against the issued challenge.

Considering the possible attacks on this challenge-response system there are two possible threats - firstly that an opponent will collect transmitted challenge-response pairs to build a dictionary and secondly that the opponent will construct a DES engine to perform a brute-force attack on KU by using known plaintext/ciphertext pairs.

The dictionary attack must be judged against the expected number of logons over the required lifetime of the smart card. Assuming an average of one logon per day for a period of three years this will give an opponent 365 x 3 = 1095 matching challenge-response pairs. This is reasonable since it includes all weekends and holidays and will hence allow for multiple logons on some days. For convenience we approximate this value to $1024 = 2^{10}$. We now examine the probability of the opponent having the necessary dictionary entry for a given challenge at the end of this three year period. Meyer and Matyas [20] have shown that the probability (p) of finding a correct look-up table entry is :-

$p = 1 - e^{-mn/N}$; $n/2N < 1$

where N = total number of possible response values

n = number of challenge-response pairs available to an opponent

m = number of exhaustive trials to obtain equivalent values of the keys that will generate the same output.

If we examine the use of 64-bit DES alone and for the present time neglect the effect of the selection algorithm, this gives the following values:-

$N = 2^{64}$
$n = 2^{10}$
$m = 2^{56}$
$n/2N = 2^{-55}$

Hence $mn/N = 2^2 = 4$

$p = 1 - e^{-4} = 0.98$

This is a totally unacceptable probability, but if we now include the selection component the picture changes dramatically. Any one of the possible 2^{56} DES keys could have been used to generate the final 96-bit output. This means that on any trial it will always be possible to find a pair of equivalent values of KU and KS which generate the observed output, hence:

$m = 1$
$N = 2^{96}$
$n/2N = 2^{-87}$

Hence $mn/N = 2^{-86}$

and p is negligible

If we were to extend the required lifetime of the card to (say) 2^{20} logons it makes no substantial impact on the value of p, and hence we have a scheme that is for practical purposes completely resistant to a dictionary-style attack.

Now let us examine the brute-force attack on KU and KS. Since a given output is possible with all values of KU we have completely decoupled the DES process from any direct attack on sets of matched plaintext/ciphertext. The opponent must search every value of KS for each and every value of KU, making a complete search of 2^{274}.

We estimate that it is feasible to build a DES engine that would work at 10^{12} tests per second and would perform an average DES key search in one day. Assuming that we employ the same engine to search for KS and that the selection algorithm operations are 100 times faster than DES operations, a complete search of KS and KU would take 2^{212} days, which of course is infeasible.

The incorporation of the selection algorithm extends the resistance to attack well beyond the resources of an opponent and adds virtually nothing to the cost of implementation. It could be used equally effectively with a non-DES preprocessor, thus extending the scope of its applications to organisations which are prohibited the use of DES.

Host-End Management for Remote Access Control

Every issued smart card has a matching user record on the host database. When an authentication request is received at the host, the appropriate record is retrieved and sent into the host security module. The secret part of the record is stored on the host database encrypted under a storage DES key, and storage keys are changed regularly to reduce their exposure to attack. The record is processed and then written back to the database encrypted under the latest storage key. A dummy request is also provided to enable the host application to refresh infrequently used records which would otherwise fall out-of-date with the storage key window.

A new "session" starts with a request for a random number challenge. The host security module provides a session sequence number and sets up a temporary store of session variables, including the values of KS, KU and the random challenge. On receipt of the encrypted challenge the host security module uses the session sequence number to index the appropriate block of session variables and hence verify the response.

If the remote logon is not interactive but forms part of a batch submission, the same user key is used to generate the next one-time password in a psuedo random sequence. [21] This sequence is tracked at the host-end and the password is validated when the batch job is processed. Password windows are used to improve system resilience, and database management is as before.

Additional resilience is incorporated by providing dual host security modules. The units each have their own unique RSA key pair and the public keys are used to organise encrypted, certified replication of storage DES keys between the two units via the host application.

Issue Authority

All smart cards for use in the network are issued at one central point. They are loaded with keys and PINs and then mailed to users. PINs are secretly printed inside special envelopes and mailed to users under separate cover. The card is loaded with a newly generated RSA key pair (512-bit keylength is used), the public key of which is certified by means of a digital signature made with an issue authority secret RSA key. [22] The certified public key is entered into the system directory and is available for reference by all other users. The card-issue function and directory function are performed on a PC acting as a key distribution centre (KDC). The KDC is available to all network users as a central reference library for certified public keys. Each PC or host application can obtain these via the network, and can store local directories of frequently used keys. Every system node also has a copy of the issue authority public key with which the certified public keys can be authenticated at any time by validating the issue authority RSA digital signature. Hence only authorised users' public keys can be used.

Privacy of Communications

In addition to the user RSA keys held on the smart cards, each cryptographic unit in the system has a unique RSA key pair, issued to the unit in certified form just as for the smart cards, and also stored on the public system directory. It is therefore possible to have the following relationships:-

1. *User to user*

2. *User to application (and its crypto unit)*

3. *Application to application (and their crypto units)*

When messages are to be encrypted for transmission to protect against eavesdropping, a data key (DES) is generated at the originating user or application inside the cryptographic unit. For duplex communications different data keys are used for each direction. A DES key is encrypted under the RSA public key of the destination unit and signed with the RSA secret key of the source. The encrypted, signed data key is sent with the message and at the destination it is recovered by using the public key of the source to validate the signature, and the secret key of the destination to decrypt the data key. All RSA processing is performed on fast RSA processors (such as a digital signal processor) to maintain acceptable response times.

The Texas Instruments TMS 32010 DSP with suitable software can provide a 512-bit RSA secret key operation in approximately 2.8 seconds. This can be substantially improved with the TMS 32020 to approximately 2 seconds and with the TMS 320C25 to approximately 800 milliseconds. We estimate that using two Motorola 56000 DSPs an execution time of less than 50 milliseconds can be achieved.

Key Management Protocol

The transmission of keys requires a suitable protocol at the application level. This protocol must exchange messages, the contents of which are encrypted keys, unit identities, key counters and other control information. One such protocol suitable for this purpose is the group of cryptographic service messages (CSMs) described in ANSI X9.17. [23] The CSMs defined in the standard accommodate only single and double length DES keys, with no provision being made for RSA keys or RSA-encrypted DES keys. However, it is not difficult to extend the CSM set to include new field-tags and new field definitions that can handle these RSA blocks; proprietary implementations of ANSI X9.17 CSM protocol do make these facilities available, but of course their precise definition does not conform to a standard since non exists.

The key management technique described above is applicable to both interactive and store-and-forward networks. However, in the case of store-and-forward mailbox systems, it may be a function of the mail server to broadcast messages to all system users or to closed user groups. In this case the mail server is equipped with its own cryptographic unit which performs only key translation services. Broadcast messages have their data key encrypted by the source under the RSA public key of the mail server, and the cryptographic unit on the server then translates these key blocks under the RSA public key of all authorised recipients, placing the translated key block into the mailbox of each. The data key remains unchanged and so the key translation unit does not need to process the message itself.

Message Authentication

Message contents are authenticated by generating at source and validating at destination a message authentication code (MAC). This can be of the type defined in ANSI X9.9 [24] using DES. The data authentication key is carried in exactly the same way as described above for data encryption keys, using RSA for authenticating both source and destination, and using a key translation server for broadcast messages.

Digital Signatures

When a message is sent encrypted and/or authenticated as described above the data keys are already signed using RSA. However, for additional security the MAC itself is signed by the source RSA secret key, thus providing a digital signature that can be validated using the source RSA public key from the system directory.

The RSA secret key which is used to generate the digital signature belongs either to an application or to a user. In the case of it being the property of an application it is stored securely in a tamper-resistant crypto-unit which is attached to the host machine. However, where the RSA key belongs to a user it is stored in the personal smart card and is carried around by that user.

Smart card technology does not at present support 512-bit RSA processing to meet the response time requirements, and so this must be performed on the digital signal processor in the smart card reader. To achieve this the smart card must give up its secret RSA key to the reader, which is why the mutual authentication process between these two components is so important. Additionally, the reader can be made to be tamper resistant [8] [9] to protect secret keys during their residence in the unit.

Alternatively, if the speeding-up techniques discussed by Matsumoto, Kato and Imai [18] can be successfully implemented to achieve acceptable response times there is no need for the Fiat-Shamir mutual authentication protocol. In this case the smart card will not surrender its secret information to the reader and this latter device need not be either trusted or tamper resistant

Integration with IBM SNA Environments

The IBM SNA encrypt/decrypt facilities do not include RSA key management. However, on each IBM host an additional cryptographic unit with fast RSA processing capability is provided so that master DES keys can be moved around the network between hosts. PCs are equipped with plug-in cryptographic boards that emulate IBM 3848 capability [5] and also provide the additional RSA key management layer. The applications on the IBM hosts are responsible for organising the automated management of master keys under the RSA layer, using a protocol similar to that described above using ANSI X9.17 CSMs. [23]

Integration with non-IBM Interactive Environments

IBM SNA is somewhat unique in its provision of an encrypt/decrypt feature within the network architecture. Other proprietary network architectures leave cryptography largely to the implementers of the applications. The approach described in this paper is ideally suited to this latter environment, since the application drives both the key management protocols and the service requests to the presentation layer for encryption/decryption facilities. Integration into these environments therefore poses no substantial problems, since a standard interface can be defined which requires the application to incorporate only the necessary message handler for requesting and receiving cryptographic services.

Summary

The system solution described here provides a multi-layer security architecture using unified key management and multi-purpose system components to support a wide range of environments and can be integrated with both IBM-style and other proprietary security sub-systems. User participation is limited to carrying a secure token and supplying a PIN, after which the layered authentication processes are automated. System response time is maintained at acceptable levels for the user, and system management is eased by the use of automated techniques. Above all an elegant and highly secure end-to-end solution is created.

Bibliography

[1] Anura GURUGE, *"SNA - Theory and Practice"*, Pergamon Infotech, 1984

[2] *"IBM Cryptographic Subsystem Concepts and Facilities"*, (GC22-9063), 1985

[3] *"IBM RACF"*, Datapro Reports on Information Security, Report No. IS52-504, McGraw Hill, 1987

[4] *"CA-ACF2"*, Datapro Reports on Information Security, Report No. IS52-187, McGraw Hill, 1988

[5] *"IBM 3848 Cryptographic Unit Product Description and Operating Procedures"*, (GA22-7073), 1982

[6] *"IBM OS/VSI and OS/VS2 MVS Programmed Cryptographic Facility, General Information"*, (GC28-0942), 1980

[7] *"ACF/VTAM General Information"*, (GC38-0254), 1980

[8] Andrew CLARK, *"Physical Protection of Cryptographic Devices"*, Eurocrypt '87, Amsterdam, 1987

[9] Andrew CLARK, *"Physical Protection of Cryptographic Devices (Revised)"*, Proc. of Corporate Computer Security '88 Conference, Brighton, UK 1988

[10] *"Information Processing - Data Encryption Algorithm"*, ANSI X3.92, 1981

[11] *"Data Encryption Algorithm - Modes of Operation"*, ANSI X3.106, 1983

[12] Ronald RIVEST, Adi SHAMIR and Leonard ADLEMAN, *"A Method of Obtaining Digital Signatures and Public Key Cryptosystems"*, Comm. of ACM, Vol.21, No.2 Feb 1978

[13] John SHERWOOD, *"Digital Signature Schemes Using Smart Cards"*, Proc. of Smart Card '88 Conference, London, 1988

[14] John SHERWOOD, *"Automatic Key Management for Transparent Security on Corporate Data Networks"*, Proc. of International Systems Security Conference, London, 1986

[15] Paul BARRETT, *"Implementing the RSA Public Key Encryption Scheme on a Digital Signal Processor"*, Proc. of Crypto '86, Springer-Verlag, 1986

[16] Amos FIAT and Adi SHAMIR, *"How to prove yourself: Practical Solutions to Identification and Signature Problems"*, Proc. of Crypto '86, Springer-Verlag, 1986

[17] Amos FIAT and Adi SHAMIR, *"Unforgeable Proofs of Identity"*, 5th SECURICOM, Paris, 1987

[18] Tsutomu MATSUMOTO, Koki KATO and Hideki IMAI, *"Speeding-Up Secret Computations with Insecure Auxiliary Devices"*, CRYPTO '88 - to appear

[19] Raymond WONG, Thomas BERSON and Richard FEIERTAG, *"Polonius: An Identity Authentication System"*, Proc. of IEEE Symposium on Security and Privacy, 1985

[20] Carl MEYER and Stephen MATYAS, *"Cryptography: A New Dimension in Computer Data Security"*, Wiley, 1982

[21] Raymond EISELE, *"Host Access Security"*, Presented at Interact '86, Orlando, Florida, 1986

[22] Vince GALLO and Andrew CLARK, *"Issue Authority"*, 2nd Nordic Conference on Information Security, Stockholm, 1988

[23] *"Financial Institution Key Management (Wholesale)"*, ANSI X9.17, 1985

[24] *"Financial Institution Message Authentication (Wholesale)"*, ANSI X9.9, 1984