South African
Computer
Journal
Number 17
September 1996

Suid-Afrikaanse
Rekenaar-
tydskrif
Nommer 17
September 1996

# Computer Science and Information Systems

Special Edition: Computer Security

# Rekenaarwetenskap en Inligtingstelsels

**The South African Computer Journal**
*An official publication of the Computer Society of South Africa and the South African Institute of Computer Scientists*

**Die Suid-Afrikaanse Rekenaartydskrif**
*'n Amptelike publikasie van die Rekenaarverenging van Suid-Afrika en die Suid-Afrikaanse Instituut vir Rekenaarwetenskaplikes*

**Editor**
Professor Derrick G Kourie
Department of Computer Science
University of Pretoria
Hatfield 0083
Email: dkourie@dos-lan.cs.up.ac.za

**Subeditor: Information Systems**
Prof Lucas Introna
Department of Informatics
University of Pretoria
Hatfield 0083
Email: lintrona@econ.up.ac.za

**Production Editor**
Dr Riël Smit
Mosaic Software (Pty) Ltd
P.O.Box 23906
Claremont 7735
Email: gds@mosaic.co.za

World-Wide Web: http://www.mosaic.co.za/sacj/



**Subscriptions**

| | Annual | Single copy |
|---|---|---|
| Southern Africa: | R50,00 | R25,00 |
| Elsewhere: | $30,00 | $15,00 |

An additional $15 per year is charged for airmail outside Southern Africa

to be sent to:
*Computer Society of South Africa*
*Box 1714 Halfway House 1685*

Phone: +27 (11) 315-1319 Fax: +27 (11) 315-2276

## Guest Editorial

# Information Security – The Family Member Who Came Home

Basie von Solms
*Rand Afrikaans University, Johannesburg, South Africa*
*basie@rkw.rau.ac.za*

The claim that the information society is upon us and that the information superhighway is about to affect us all, is such a cliche that it hardly bears repeating - practically everyone is talking and writing about it. Although the assertion that information *security* is an essential and integral part of the information society might also seem cliched to some, it nevertheless does not get as much attention and exposure as it should. Information security has always been seen in the same light as taxes: nobody really wants it, but everybody (reluctantly) admits that we need it. Following an alternative analogy, it is like an unwanted family member: we realise that he is part of the family, but we really do not want to be bothered by him. In the last few years, the growth of the Internet and the explosive appearance of the World Wide Web (WWW) has brought information security into corporate boardrooms and private lounges. Suddenly everyone wants to get to know this unwanted family member a little better!

This issue of SACJ is dedicated to the unwanted family member and if, by reading this issue, all of us in the IT family are able to learn just a little more about this sibling, then producing the issue would have been worthwhile. In fact, we should welcome him back home as soon as possible.

A little background about this issue is in order. The International Federation for Information Processing (IFIP) is a consortium of about 60 member countries. It provides an umbrella for many international IT activities. Countries are represented by their national IT society, South Africa being represented by the Computer Society of South Africa (CSSA). IFIP has a total of 13 Technical Committees (TCs), each concentrating on a different aspect of IT. TC 11 deals with all aspects of information security. It organises an annual international conference – the so-called IFIP/Sec series, which is widely regarded as one of the major information security conferences.

IFIP/Sec 95, the 11th International Conference on Information Security, took place in Cape Town in May 1995 and IFIP/Sec 96, the 12th International Conference on Information Security, took place on the Greek island of Samos in May 1996. Complete proceedings of the two conferences, together containing more than 80 articles, have been published by the official IFIP publishers, Chapman and Hall. This issue of SACJ consists of a selection of four articles from the IFIP/Sec 95 Proceedings and two from the IFIP/Sec 96 Proceedings. It is intended not only to disseminate relevant information, but also to bring the information security interests of SAICSIT, CSSA, IFIP and TC 11 to the attention of readers. If, after reading this issue, you are interested in the remaining 74 articles, or in the activities of any of these bodies, please feel free to contact the guest editor directly.

The selected articles cover a diverse range of information security issues. Parker extends its theoretical and conceptual understanding, Hoffman addresses several of its non-technical but crucial aspects, Muftic concentrates on its role in open distributed systems, de Ru and Eloff link into the use of biometrics in information security, von Solms investigates the security protocols for the Internet and WWW, and Pangalos and Khair remind us that information security requirements extend even into the medical field.

The first four articles are from the IFIP/Sec 95 Proceedings. The first, by Donn Parker, suggests a framework for information security in order to avoid information anarchy. He argues that the traditional view of the role of information security – to protect the three elements of confidentiality, integrity and availability of information – is dangerously oversimplified. He includes three more elements of information into the equation: authenticity, utility and possession. The article provides a good platform for gaining a better understanding of what information security really ought to be about.

In the second article, Lance Hoffman addresses the important issues of escrow encryption and export controls – specifically, as he clearly points out, from a US standpoint. The article highlights the very important fact that cryptology is not merely a technical issue, but that the political overtones, civil liberties and administrative implications are also extremely relevant. Ignoring these non-technical issues, as many information security specialists tend to do, will have a negative impact on the field becoming a discipline in its own right. Since its first publication, some of the issues raised by the Hoffman article have been receiving attention. For example, the idea of international cryptology policies is being addressed by the European Organisation for Economic Cooperation and Development (OECD). Relevant recent articles on the escrow aspect can be found in [1].

In the next article, Sead Muftic concentrates on aspects of security in open distributed environments. He identifies a number of elements suitable for a secure system in such an environment. These are: smart cards, secure user workstations, integrated security clients, security servers and a global certification system for international networks. This last aspect is becoming more and more important as basically all secure protocols use public key encryption in some form or the other. Using these elements, he also describes a number of security enhanced applications – secure Internet e-mail and secure EDI. Muftic stresses the fact that all these elements are operational, implemented, and already in use.

de Ru and Eloff then discuss the reinforcement of password authentication using typing biometrics. Biometric methods are probably the best means of authentication, but many of these methods are technology-intensive and expensive. Their article tries to use typing characteristics as a cheap and user-transparent way to augment the traditional password.

The last two articles are from the IFIP/Sec 96 Proceedings. von Solm's article gives an overview of two non-payment related and one payment related secure protocol for the Internet and the WWW. The non-payment related protocols are Secure Sockets Layer (SSL) and Secure Hypertext Transport Protocol (SHTTP). SSL is usable in any TCP/IP environment, while SHTTP is specifically for the WWW. Both make use of public key encryption. The payment related protocol, Secure Payment Protocol (SEPP) was superseded by the Secure Transaction Protocol (SET) early in 1996, but only after the article had already been submitted to IFIP/Sec 96. The presentation at the conference however, covered SET and not SEPP. An appendix is attached to this article, giving a brief overview of SET, as SEPP is no longer relevant. SET also uses public key encryption.

The last article by Pangalos and Khair introduces a methodology to improve the security of medical databases in relation to authentication. Though the methodology of the authors is in itself important and interesting, the article was selected for this issue to underline the importance and relevance of information security in medical IT applications – an area where security introduces new problems quite distinct from those traditionally encountered in the financial and other fields. Becuase of the differences, information security in medical applications still requires much research.

It is hoped that readers will find this issue of SACJ useful, and also that they will get involved with the activities of the bodies mentioned above.

## References

1. *Communications of the ACM*, **39**(3):33–53, (March 1996).
2. http://www.visa.com.

## Appendix A

Secure Electronic Protocol (SET) [2] is the secure payment protocol announced by MasterCard and Visa in February 1996. The two previous protocols, SEPP and STT, announced independantly by these two companies, were replaced by SET.

This Appendix gives a brief and oversimplified overview of the SET message flow between the customer and the merchant when a payment is made.

Only the most basic parts of messages are discussed, and many details are left out for the sake of simplicity.

The electronic purchasing process can basically be divided into two phases. Phase 1 is a browsing and negotiation phase in which the customer will decide what to buy. Goods and price will be negotiated and agreed upon between the customer and merchant. The phase will probably end with a completed order form, specifying the goods,with the associated price, the customer is about to buy.

At this point the customer decides to start Phase 2, the payment phase. It is at this point that the SET payment protocol is initiated.

**Step 1:**

A message is sent from the customer to the merchant, requesting, amongst other things, the public key certificates of the merchant and the merchant's acquirer.

**Step 2:**

The merchant sends these certificates back to the customer.

**Step 3:**

- The customer validates the certificates received from the merchant.
- The customer constructs the Order Information (OI).

  OI contains info about the goods and negotiated price.
- The customer generates a DES key K1 and encrypts OI under K1, giving K1(OI).
- The customer encrypts K1 using the public key of the merchant, giving MP(K1).
- The customer constructs the hash of the OI, giving H(OI).
- The customer constructs the Payment Information (PI). The PI contains info about the credit card nr, expiry date etc.
- The customer generates a DES key K2 and encrypts PI under K2, giving K2(PI).
- The customer encrypts K2 using the public key of the acquirer, giving AP(K2).
- The customer constructs the hash of the PI, giving H(PI).
- The customer constructs the hash of OI, the hash of PI and concatenates them, giving H(OI)||H(PI).
- The customer digitally signs this concatenation giving S(H(OI)||H(PI)). The customer performs

this operation to uniquely associate the specific OI with the specific PI.

**Step 4:**

The customer sends the message M1= [K1(OI), MP(K1), H(OI), K2(PI), AP(K2), H(PI), S(H(OI)||H(PI))] to the merchant.

**Step 5:**

The merchant receives M1= [K1(OI), MP(K1), H(OI), K2(PI), AP(K2), H(PI), S(H(OI)||H(PI))]

- The merchant retrieves K1 and then OI.

  Note that the merchant cannot retrieve PI in any way. The customer intended it this way, because he/she does not want the merchant to see his credit card info.

- The merchant creates H(OI), concatenates it with H(PI) received in M1, and compares it with the H(OI)||H(PI) received in M1.

  Note that the merchant can retrieve H(OI)||H(PI) from S(H(OI)||H(PI)) by using the customer's public key.

  If they are equal, then the merchant knows that the OI he/she has retrieved from M1 is the 'correct' OI intended by the customer to accompany the PI provided by the customer.

  Note that this process allows the merchant to associate a specific OI with a specific PI without knowing precisely what PI is.

**Step 6:**

The merchant now requests an authorisation from his acquirer, by sending the following message M2 to the acquirer: [H(OI), K2(PI), AP(K2), H(PI), S(H(OI)||H(PI))]

- The acquirer goes vica versa through the same procedure as the merchant, allowing the acquirer to associate the specific PI with the specific OI without knowing precisely what OI is.

  The customer does not want the acquirer to see what he/she is buying, but wants the acquirer to link the OI to the PI.

**Step 7:**

If necessary, the acquirer gets authorisation from the issuer, and send a digitally signed authorisation back to the merchant.

**Step 8:**

The merchant informs the customer that the transaction is authorised.

SACJ is produced with kind support from
Mosaic Software (Pty) Ltd.

# Functional and Operational Security System for Open Distributed Environments

Sead Muftic
*DSV Department, Stockholm University and Royal Institute of Technology, Sweden*
*sead@dsv.su.se*

## Abstract

*The paper describes the design details and implementation results of the completely integrated, functional and operational security system, suitable for open distributed environments. Functionality means that security system provides all ISO/OSI security services and operability means that the described system is completely operational on various user platforms. The components of the system are smart cards, secure user workstations, integrated security clients, security servers and global certification system for international networks. Several security enhanced applications are also described in the paper: secure PC based on smart cards, secure Internet E-mail (PEM) and secure EDI, all based on the same concept of the security system, common security architecture and integrated security technologies and products.*

**Keywords:** *D.4.6, C.2.4*
**Computing Review Categories:** *Information Security, Distributed Systems, Smart Cards*

## 1 The Properties of the Integrated Security System

The main goal of any security system is to provide security services to users and applications. Security services may be implemented by various security mechanisms. Security mechanisms may be based on alternative algorithms and implementations. Finally, different security services must be combined and integrated in a complete security system, sometimes the result of original development and sometimes available through the existing installed security products.

When analysing security requirements and extensions in various international standards, global applications and operational environments, it may be noticed that in all of them the same or similar services are always required, these services may be implemented by the same or similar security mechanisms, and all standardized security applications may be based on the same or similar concept of a global security architecture.

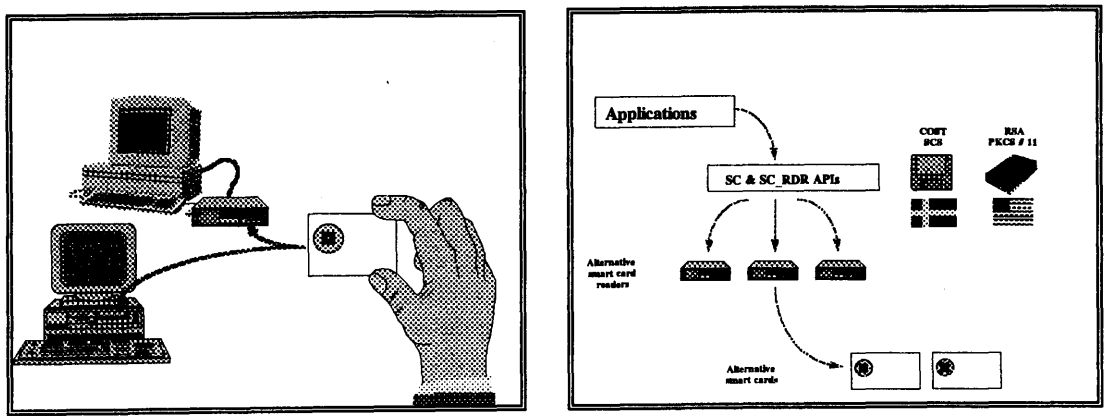
Therefore, the main goal when designing the complete security system, which is described in this paper, was to synthesize and integrate all security requirements, suggested security services, proposed mechanisms and elements of security architectures into the unique concept of a generalized security system. The motives and expectations of such a "synthesized" concept of the security system were that the same concept and security architecture could be used to provide security in various internationally standardized applications and popular operational environments. The objectives of the security system were to enable distributed networks applications to use the same security concepts, infrastructures and basic technologies, despite heterogeneity in installed equipment, registered users, operating systems, networks, or management authorities. The described security system and general concept, with such goals in mind, provides architectural elements, security services and security mechanisms to a variety of users, to different types of computer and communication networks, and to various types of user applications.

In order to meet the described expectations and goals, the security system had to be designed with the two basic principles:

1. The system had to provide an integrated set of security functions, services and mechanisms to different types of network applications in a flexible and functional way. This was achieved by the special internal infrastructure of the security system.
2. In order to hide the heterogeneity of its implementation from users and applications, the system had to be available through appropriate security APIs (Application Programming Interfaces).

These two principles were met by designing the security system based on the object–oriented methodology and with the internal layered infrastructure. The system described in this paper has been designed in the form of small, autonomous object modules, grouped in several layers of functionality and complexity (security object classes). The interfaces between modules (at the same layer or between functional layers) are designed in a form of security APIs. The external aspects of each module (visible behaviour) are given through the name (function) of the module and associated data (input parameters and output results). By combination of elementary security modules, larger and compound modules are created, as components of the global security system.

The results of the design and implementation, described in the remaining sections of this paper are organized from smaller security components (secure user workstation), extended to security systems for local area networks, and finally, completed with solutions and security architectures for global, international networks. The main property of all these security components and individual security systems is their extendibility. This means that not only




(a) External (b) Internal

*Figure 1. Structure and components of the smart cards development system*

the same internal security components have been used for implementation of different security products, but also that one (smaller scale) security product appear as the integral component of the larger security system.

## 2 Smart Cards System and Secure User Workstation

Smart cards are plastic, credit-card size cards with a special cryptographic chip. The chip may perform various symmetric (DES) and asymmetric (RSA, DSS) cryptographic algorithms. In such a way, the primary purpose of smart cards is creation of digital signatures on electronic documents. Since the cryptographic chip has also internal memory, where various user data may be stored, it means that the volume, structure and contents of the internal data values must be determined in advance before starting to use the smart card. Determination of the internal smart card structure is called formatting of a smart card, while the initialization of smart card parameters is called personalization. Smart cards which are delivered to users should be formatted, but not personalized. The internal personal card parameters should not be entered in the card during formatting, i.e. smart cards should be delivered as "blank" formatted smart cards. In such a way each user may generate its own personal smart card parameters and enter them into the smart card.

Before their usage, each smart card must be first personalized. Personalization of smart cards in some security domain (department, company, etc.) must be performed by one user with special security privileges. That user is usually called smart cards administrator. During that process personal parameters are generated and entered into the smart card by smart card administrator in cooperation with smart card owners and therefore the data in the card are protected by owner's Personal Identification Number (PIN), which is always needed in order to activate the smart card. After the personalization process, the parameters and data in the smart card may be changed (updated), but the internal structure of the smart card, created during the formatting process, with some cards may be changed and with some may not be changed.

The described concept of usage of smart cards have been implemented as the *Smart Cards Development System.* It is intended for developers of smart cards applications. One smart cards development system set consists of the following four components:

1. smart cards demo system, which also includes administration software for smart cards,
2. smart cards "engine" software (DLL) with smart cards and reader's functions,
3. one smart cards reader and smart cards drivers for PC or Macintosh workstations, and
4. several pre-personalized smart cards.

With these components and documented smart cards APIs, any user application may be extended with smart cards functions. Specific user applications may be created including the customized internal structure of smart cards and customer's logo on the card.

Figure 1 shows the external and internal structure and components of the smart cards development system. Figure 1a shows a smart card and two types of smart card readers: a desktop and a built-in reader. Figure 1b shows the internal structure of the smart cards system: smart cards applications based on smart cards APIs, which include drivers for various smart card readers and smart cards. There are different types of readers, ranging from very small size readers with "transparent" protocols to sophisticated readers with a keypad (used to activate the card directly at the reader) and with the LCD, where data to be signed by the card are first displayed for visual inspection.

Users who are not interested in development of specific applications, but want to use generalized smart card systems, may consider already available *Secure PC* based on digital signature smart cards. *Secure PC* is the product which provides full protection of all resources at a single PC against intruders and illegal users. Its functioning is based on usage of cryptography and digital signature smart cards for encryption/decryption of files and creation/verification of digital signatures. *Secure PC* may be used for protection of any kind of PC resource: text files, source and executable

programs, documents, etc. It may be activated at the start-up of Windows or invoked separately for each application whose resources must be protected.

Figure 2 shows the activation (Secure Login) screen of the *Secure PC*. *User name* and *PIN* must be given in order to activate the smart card. If activation is skipped by pushing the **Cancel** button, the system may be activated by the **Activate Card** button at the working screen. The system must also be activated if the smart card is pulled out of the reader during the working session.

Figure 3 shows the working screen of the *Secure PC*, where all the functions of that product are invoked by simple push of the corresponding button.

*Secure PC* is intended for end-users of PC/Windows applications. By switching tasks from the current application to the *Secure PC*, the user may very flexibly encrypt/sign or decrypt/verify various documents, files, source and executable programs or E-mail letters.

Functions and security features of the *Secure PC* product are the following:

1. *Authorized usage* of the PC through smart card authentication
2. *Encryption/decryption* of files, programs and other PC resources
3. Creation and verification of *RSA digital signatures*
4. *User-friendly* interface and dialogues
5. *Extendible* with PC boot sequence protection, virus detection and elimination and directory protection

*Secure PC* or other applications based on usage of smart cards, as described, may be treated as the completely secure single user workstation or as the client user workstation in a local area network or global network. As such, they can be components of a larger, network security systems, described further in this paper.

## 3 Secure LAN based on Extended Kerberos System

The best comprehensive security system for client/server distributed environments should be based on the standard MIT Kerberos v5 system which provides authentication and limited authorisation security services for UNIX client and server machines. That system, in order to be used outside of the USA, must be extended with cryptographic routines and libraries which must replace the original MIT crypto modules. The original MIT version needs also some additional security services, mainly access control, GUI based management interfaces and some architectural extensions in case of cross-domain usage.

The structure of the Kerberos system with several registered users and application servers is shown in Figure 4a. Figure 4b shows the extension of the standard Kerberos system with the access control, where access rights of users are defined during their registration and later checked before issuing tickets.

The security system may be used in the following way. After the installation of the system, all users, application servers and applications in a local domain must be registered at the Kerberos Authentication Server. During registration, secret user and application servers cryptographic keys are established, known only by their possessors and the Authentication Server. During the initial login of users, the ticket to the Kerberos Authentication Server is passed to the user and it can be later used to fetch tickets for other application servers. In such a way Kerberos provides single sign-on feature in a local domain. After the initial authentication, users may access other application servers and applications in the authorised way without explicitly authenticating themselves at these servers. Authentication is transparently perform based on the tickets. All users actions may be performed through the standard kerberized applications: *rlogin*, *rcp* and *FTP* or through user created kerberized applications. In addition, users may protect their resources by use of cryptography or verify the correctness of resources and their authentic origin.

The system consists of three components: software for administrative and authentication functions at the Kerberos security server, software for standard security clients at user workstations and security modules for any additional security functions.

With all the described security extensions, the functions of the complete security system for LANs based on the Kerberos system are the following:

- registration and authentication of all users and resources in a single LAN or in a distributed system,
- controlled and authorised usage of distributed system resources, application servers and individual applications through the access control system,
- encryption/decryption of messages (standard Kerberos feature) and additional encryption/decryption of files, programs and documents for extended protection at the application level,
- creation and verification of integrity codes for messages (standard Kerberos feature) and additionally also for files, programs and documents for extended protection at the application level,
- authentic and authorised access and usage of network applications, files and programs based on digital signatures and certificates.

## 4 Global Certificate Management System

The main security technology today for authentic and authorised international electronic transactions is public key cryptography. In order to make that technology available and usable to different types of users and applications, two problems must be effectively solved: *(a)* generation, storage and usage of secret keys, and *(b)* generation, storage, distribution, verification and usage of public keys. The best technology today for protecting and using secret keys are smart cards, described in section 2 of this paper, and the best solution for the second problem are X.509 certificates and global certification infrastructure.

X.509 certificate is a special data structure which con-

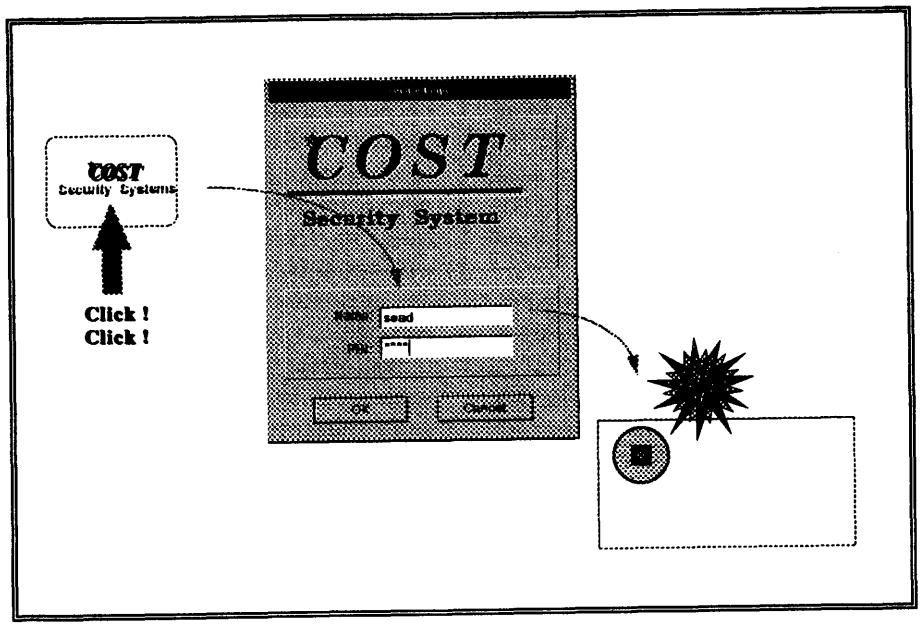



*Figure 2. Activation screen of Secure PC*

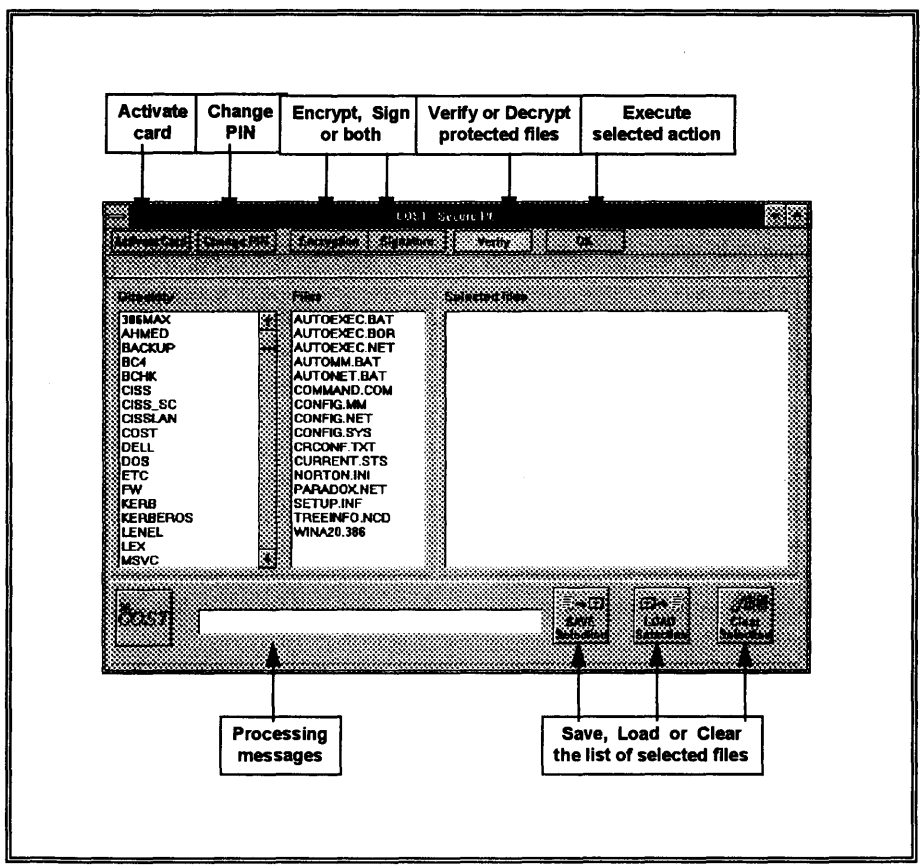
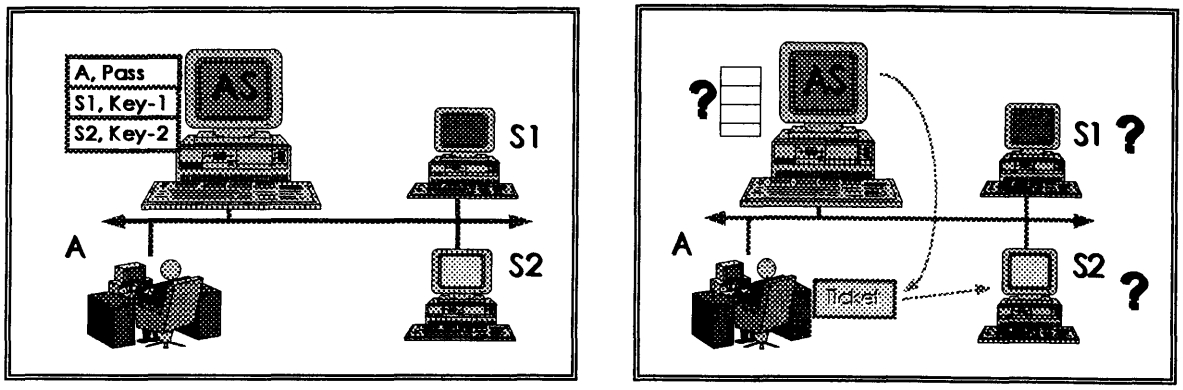



*Figure 3. Working screen of Secure PC*




(a) Structure (b) Extension with access control

*Figure 4. The Kerberos system*

tains user's public key. In addition, the certificate contains also unique user's identification and some additional parameters related to the validity of the certificate. In order to guarantee the integrity, authenticity and originality, each certificate must be issued by the appropriate certification authority. This is reflected in the user's certificate through the identity of the authority, together with the authority's digital signature of the certificate.

The authorities who issue certificates must be trusted with respect to their responsibilities, functions and secrecy. Therefore, they are called *Trusted Third Parties* (TTPs). TTPs constitute the system of special security servers linked into the international computer networks. The main functions of these servers and the main purpose of the TTP system is to provide security, confidence, assistance and support for business electronic transactions in international networks. In order to declare and supervise security policy, to provide strict verification of each certificate and to provide additional functions, TTPs must be organised in a hierarchy with the special, so called *Policy Certification Authority* (PCA) at the top of the hierarchy. Internet Privacy Enhanced Mail (PEM) system suggests that all the PCAs are further linked to a single top level Internet network certification authority, called *Internet Policy Registration Authority* (IPRA).

Each TTP in the certification hierarchy performs four groups of functions: *(a)* registration and authentication of legal entities and individuals, *(b)* storage and international distribution of identification information, *(c)* certification and certificate management functions, and *(d)* various notary services. These four groups of services are performed in the form of various TTP functions and special certification protocols, which are usually based on special E-mail letters.

Registration and authentication of companies and individuals by TTPs should be based on usage of international standardised naming schemes, i.e. X.500 attributes. Each TTP stores and distributes internationally unique names of individuals and business entities. Most of current implementations do not depend on the full X.500 services, they use only a subset of X.500 attributes for identities of the registered entities, certificates are handled separately from X.500 servers and identities are distributed within certificates. However, all existing certification systems may be in the future combined with full X.500 services.

All certificates functions are organized and performed in the form of a global certificate management system. Certificate management system provides certificate management functions: generation, signing, storage, distribution, and verification of certificates. These functions are implemented as special E-mail letters. Notary services may also be used for improved confidence and trust in business electronic transactions.

In order to establish the TTP function, some organisation must first decide on its internal TTP structure. The top level TTP in that structure must be linked to some TTP already existing in the hierarchy. After the top level TTP is installed, its certificate is generated and sent to the higher level TTP for certification. When returned, that TTP may certify lower level TTPs. Finally, at the bottom of the hierarchy, the human users are certified by their local TTPs. Figure 5 shows a global certification management system in the form of a hierarchy and some certificate management letters.

Certificates of individuals may also be stored in their smart cards. In such a way user mobility may be achieved, since people may access and use secure network applications from any security enhanced terminal.

## 5 Security Enhanced Applications

Smart cards systems, extended Kerberos and certificate management system may be used as general security infrastructure for creation and establishment of various standardized or customized security applications. In this section the implementation of two standardized security applications, Internet Privacy Enhanced Mail (PEM) and secure EDIFACT, based on some components of that infrastructure, are described.

**Internet Privacy Enhanced Mail (PEM) System**

PEM is secure E-mail system for the Internet network. It provides confidentiality and integrity of E-mail letters, to-

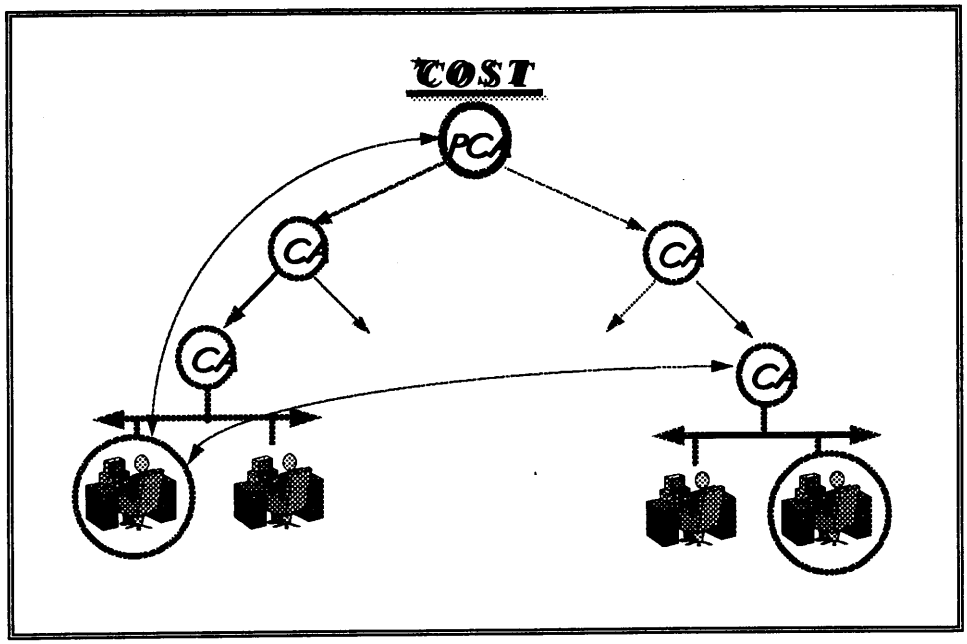



*Figure 5. A global certification management system*

gether with the sender's and receiver's authenticity. Its implementation must be based on public key cryptography, so the system also provides non-repudiation of the sender. PEM may be implemented as a complete autonomous secure E-mail system or as extension of any other existing E-mail system.

Besides basic functions of creating and sending, and subsequently receiving and verifying the E-mail letters, one important component of the PEM system is the certificate management system. It may be implemented as the integral part of the PEM system or separately, as described in section 4, when it may support not only PEM, but also other security enhanced applications.

PEM may be used to protect not only E-mail letters, but also various kinds of important resources during their transfer through the network. Computer files, programs, business reports, industrial documents, and business communication messages may be transferred as E-mail letters in a highly protected form. The sender and receiver of protected letters may be authenticated and verified. This means that

- the sender may be certain that his/her letter will not be read or changed by unauthorised users and that the submission of the letter to the intended receiver may be proved to the third party, and
- the receiver may be certain that the content of the received letter is original, that the letter has not been read by any unauthorised user and the fact that it was sent and confirmed by the indicated sender may be proved to the third party.

PEM may be useful for all users and companies which need to transfer sensitive materials, business documents and files via the E-mail. PEM may efficiently protect E-mail letters from several types of potential problems and through its digital signature capabilities, the system may be used as the platform for electronic commerce transactions. It must be implemented in full compliance with the official PEM standards and user agents may be extended with the smart cards for additional efficiency and protection of users' secret keys.

PEM consists of two subsystems: *(a)* X.509/PEM certificate management system, and *(b)* PEM user functions. All functions of the certificate management system are described in section 4 and the certificate management system is an integral part of the PEM system.

The full implementation of the PEM system should contain four components:

1. The complete set of programs and configuration files for implementation of the PEM Server functions, which must be installed at mail servers. PEM Server consists of two groups of functions: manual actions of the security administrator and automatic actions of the PEM Server, together performing all necessary certificate management functions.
2. The set of PEM User Agent programs and configuration files for the PC/Windows as user workstation. These programs perform all PEM user functions and in addition assist user in performing certificate management functions from user's workstation in co-operation with PEM Server at the mail server.
3. Functionally equivalent set of programs and files for the PEM User Agent as in 2., but for the Macintosh workstation.
4. Functionally equivalent set of programs and files for the PEM User Agent as in 2., but for the UNIX workstation.

Manual actions of the security administrator and automatic actions of the PEM Server at each mail server implement

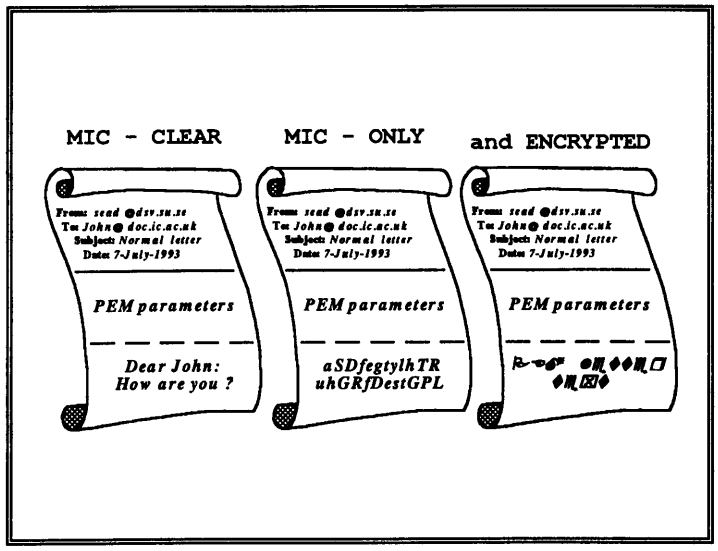



*Figure 6. Three types of PEM letters*

together the functions of the Certificate Authority (CA). Certificate Authorities are organised in a hierarchy, each signing certificates of the lower level CAs, while the lowest level CAs sign the certificates of users. All programs and files used by PEM Server must be installed at each CA mail domain, regardless whether CA serves users or just performs certificate management functions in a hierarchy.

User secure E-mail functions must be implemented on a PC/Windows, Macintosh and UNIX workstations. For PC and Macintosh user secret keys may be stored on protected diskettes or smart cards, on UNIX machines the secret keys must be kept on a disk, encrypted by special user's security password.

Users may register themselves, generate their own certificates, send them for signatures, create and send, also retrieve and verify PEM letters, and retrieve and verify partners' certificates. All user PEM functions and user communications with local CAs should be implemented as PEM User Agents, communicating with users by friendly menu-driven dialogues and communicating with local servers transparently through mail protocols.

After the registration, when users receive back the full certification path, they may create and receive PEM letters from other PEM users. PEM users may also send and receive E- mail letters from users who do not have the PEM system.

There are three types of PEM letters, shown in Figure 6: MIC-CLEAR, MIC-ONLY and ENCRYPTED. MIC-CLEAR is the letter which is in the PEM format, but the text is original, i.e. not filtered. Therefore, this type of the letter may be sent to a user who is not using the PEM system. MIC-ONLY is the letter with the Message Integrity Code (MIC) and filtered, so it provides message integrity security service. ENCRYPTED is the encrypted type of the letter.

The prerequisites for the initial installation of the PEM system is the Internet mail system installed in a domain with a number of user workstations using the mail system over the local TCP/IP network. In such environment, the special person should be nominated as the PEM system administrator. That person will first install the PEM Server software at the mail server and user PEM agents at the local user workstations. The PEM Server together with the running mail server will constitute the local Certificate Authority (CA).

In case of multiple CAs in some large organisation, the system must be initially installed by in the "top-down" approach. First, the central, top level customer's CA must be established. Its certificate will be sent to the higher level CA for signature and returned after signing. The certificate of the higher level CA together with certificates of all CAs up to the top of the hierarchy will be passed to the lower level CA together with its signed certificate. After that, the next lower level customer's CAs may be installed. They must again, as the first step, generate their certificates and send them to the top level customer's CA for signature. They will be signed and returned together with certificates of other CAs above them in the hierarchy. In that way, the certificates of CAs propagate downwards through the hierarchy, during the process of CAs registration. When the CA's certificate is signed and returned by the higher level CA, that CA may further sign the certificates of next lower level CAs.

Finally, when certificates of the lowest level CAs, i.e. CAs serving users in local environments are signed, users may start their own generation of certificates and their submission for signature. The lowest level CAs will return to users their own signed certificate and certificates of all CAs up to the top of the certification hierarchy (certification path). When users receive back the full certification path, they may create and receive the PEM letters from other PEM users. PEM users may also send and receive E-mail letters from users who do not have the PEM system.

However, initially, other then the certification path and his/her own certificate, the user has no certificate of any other partner. So, if he/she wants to send an ENCRYPTED

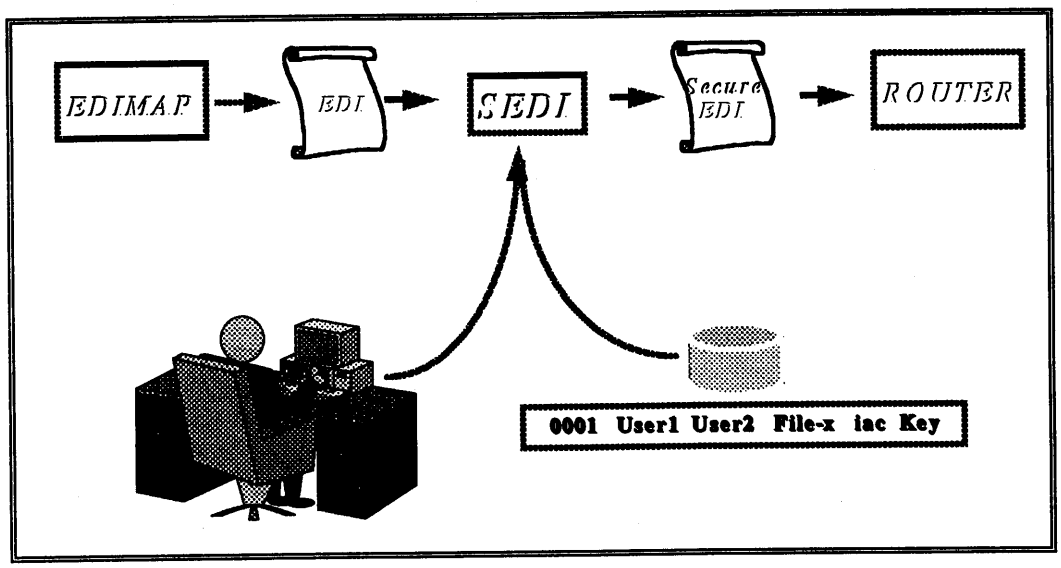



*Figure 7. Activation of Secure EDI*

type of the PEM letter to the partner or if he/she has just received the letter signed by a partner, the user in both cases needs partner's certificate (public key). That key must be fetched from partners CA. This will be achieved by sending a special certificate request letter to that CA. In that way, each user during its usage of the PEM system accumulates certificates of his/her partners at the PEM user's workstation.

Because of the filtering process of PEM letters, PEM may be used for transfer not only of simple ASCII letters, but also of any other type of computer resource: images, documents, source or executable programs, etc.

**Secure EDIFACT**

Secure EDI is the product which provides full protection of EDIFACT documents against illegal reading or accidental modifications. Documents are protected during transfer through communication networks or while stored in local archives. Sender's and receiver's authenticity and sender's non-repudiation are also provided by usage of public key cryptography.

Functions and features of the secure EDI system are the following:

- *Authorized usage* of EDI system through strong user authentication.
- *Standardized* security services for EDIFACT:
  - user authentication,
  - data integrity,
  - data confidentiality,
  - digital signature,
  - sender's and receiver's authenticity.
- Security extensions *for any EDIFACT platform* fully UN/EDIFACT compliant.
- *User-friendly* transparent operations.
- Protection of EDIFACT documents during *transfer* or in *storage*.

The functioning of the secure EDI system is based on usage of public key cryptography and X.509 certificates. For confidentiality and integrity of documents secret key cryptography is used. For private cryptographic keys, security administrator and users may use smart cards or protected diskettes.

Security transformations of EDIFACT documents are performed after EDI transformations and before sending, as illustrated in Figure 7. Subsequently, security verifications are performed immediately after receiving the EDIFACT document. The sender of protected EDI messages may specify security options in two ways: by typing them on the keyboard for each message or by the predefined Transaction Table. Therefore, *Secure EDI* may be activated manually by users for each individual EDI transmission through the dialogue option or transparently, by invoking it through extensions of the EDI platform

*Secure EDI* may be used to enhance with security any EDIFACT platform. It accepts as input the files in EDI format and returns protected documents in the same format, so it may be transparently linked to the EDI platform. It also functions transparently from the user's point of view: only warning and error messages are displayed; otherwise the system is "invisible" for users.

*Secure EDI* requires one security administrator in each domain. Security administrator registers EDI users, generates their certificates and distributes certificates within the domain.

All users must be registered and certified by the security administrator before using the EDI system. In that procedure X.509 certificate is issued to each user, signed by the security administrator. The certificate of each registered user must be distributed to all other users. This is a logical link between the certification system and secure EDI system.

## 6 Conclusions and Future Developments

All the individual components of the described integrated and functional security system have been implemented and they are individually already in use. The implementation has proved the flexibility and correctness of the global concept. Future developments should be oriented towards better integration of individual system components, their portability to various operational platforms and interoperability with existing proprietary security systems.

# Notes for Contributors

The prime purpose of the journal is to publish original research papers in the fields of Computer Science and Information Systems, as well as shorter technical research notes. However, non-refereed review and exploratory articles of interest to the journal's readers will be considered for publication under sections marked as Communications or Viewpoints. While English is the preferred language of the journal, papers in Afrikaans will also be accepted. Typed manuscripts for review should be submitted in triplicate to the editor.

### Form of Manuscript

Manuscripts for *review* should be prepared according to the following guidelines.

- Use wide margins and $1\frac{1}{2}$ or double spacing.
- The first page should include:
  - title (as brief as possible);
  - author's initials and surname;
  - author's affiliation and address;
  - an abstract of less than 200 words;
  - an appropriate keyword list;
  - a list of relevant Computing Review Categories.
- Tables and figures should be numbered and titled.
- References should be listed at the end of the text in alphabetic order of the (first) author's surname, and should be cited in the text in square brackets [1–3]. References should take the form shown at the end of these notes.

Manuscripts accepted for publication should comply with the above guidelines (except for the spacing requirements), and may be provided in one of the following formats (listed in order of preference):

1. As (a) LaTeX file(s), either on a diskette, or via e-mail/ftp – a LaTeX style file is available from the production editor;
2. As an ASCII file accompanied by a hard-copy showing formatting intentions:
   - Tables and figures should be original line drawings/printouts, (not photocopies) on separate sheets of paper, clearly numbered on the back and ready for cutting and pasting. Figure titles should appear in the text where the figures are to be placed.
   - Mathematical and other symbols may be either handwritten or typed. Greek letters and unusual symbols should be identified in the margin, if they are not clear in the text.

   Contact the production editor for markup instructions.
3. In exceptional cases camera-ready format may be accepted – a detailed page specification is available from the production editor;

Authors of accepted papers will be required to sign a copyright transfer form.

### Charges

Charges per final page will be levied on papers accepted for publication. They will be scaled to reflect typesetting, reproduction and other costs. Currently, the minimum rate is R30-00 per final page for LaTeX or camera-ready contributions that require no further attention. The maximum is R120-00 per page (charges include VAT).

These charges may be waived upon request of the author and at the discretion of the editor.

### Proofs

Proofs of accepted papers in category 2 above may be sent to the author to ensure that typesetting is correct, and not for addition of new material or major amendments to the text. Corrected proofs should be returned to the production editor within three days.

Note that, in the case of camera-ready submissions, it is the author's responsibility to ensure that such submissions are error-free. Camera-ready submissions will only be accepted if they are in strict accordance with the detailed guidelines.

### Letters and Communications

Letters to the editor are welcomed. They should be signed, and should be limited to less than about 500 words.

Announcements and communications of interest to the readership will be considered for publication in a separate section of the journal. Communications may also reflect minor research contributions. However, such communications will not be refereed and will not be deemed as fully-fledged publications for state subsidy purposes.

### Book reviews

Contributions in this regard will be welcomed. Views and opinions expressed in such reviews should, however, be regarded as those of the reviewer alone.

### Advertisement

Placement of advertisements at R1000-00 per full page per issue and R500-00 per half page per issue will be considered. These charges exclude specialized production costs which will be borne by the advertiser. Enquiries should be directed to the editor.

## References


1. E Ashcroft and Z Manna. 'The translation of 'goto' programs to 'while' programs'. In *Proceedings of IFIP Congress 71*, pp. 250–255, Amsterdam, (1972). North-Holland.
2. C Bohm and G Jacopini. 'Flow diagrams, turing machines and languages with only two formation rules'. *Communications of the ACM*, **9**:366–371, (1966).
3. S Ginsburg. *Mathematical theory of context free languages*. McGraw Hill, New York, 1966.

South African Computer Journal

Number 17, September 1996
ISSN 1015-7999

Suid-Afrikaanse Rekenaar-tydskrif

Nommer 17, September 1996
ISSN 1015-7999

# Contents



Drukkers
Printers