APM

POSEIDON HOUSE • CASTLE PARK • CAMBRIDGE • CB3 0RD UNITED KINGDOM
+44 1223 515010 • Fax: +44 1223 359779 • Email: apm@ansa.co.uk • URL: http://www.ansa.co.uk

ANSA Phase III

E2S Implementation Architecture

Andrew Herbert

(Project Technical Director)

Abstract

A key objective of the E2S project is to produce a *common technology framework* for large scale secure electronic commerce on the Internet.

This document is the description of the implementation architecture for the common technology framework across the E2S project. It identifies, positions and outlines the function of the main elements of that framework. In addition, it describes the overall security model.

The document is intended to explain the E2S implementation architecture to a technical audience. It assumes the reader has reasonable familiarity with Internet and security technology.

Detailed architectural specifications for the elements of the framework identified in this report will be produced as subsequent E2S deliverables, to be combined with this document in the final E2S Architecture Specification.

APM.1857.00.01 **Draft** 14th October 1996

Architecture Report

Distribution:

Supersedes:

Superseded by:

E2S Implementation Architecture

E2S Implementation Architecture

(Project Technical Director)

APM.1857.00.01

14th October 1996

The material in this Report has been developed as part of the ANSA Architecture for Open Distributed Systems. ANSA is a collaborative initiative, managed by APM Limited on behalf of the companies sponsoring the ANSA Workprogramme.

The ANSA initiative is open to all companies and organisations. Further information on the ANSA Workprogramme, the material in this report, and on other reports can be obtained from the address below.

The authors acknowledge the help and assistance of their colleagues, in sponsoring companies and the ANSA team in Cambridge in the preparation of this report.

APM Limited

Poseidon House
Castle Park
CAMBRIDGE
CB3 0RD
United Kingdom

TELEPHONE UK	(01223) 515010
INTERNATIONAL	+44 1223 515010
FAX	+44 1223 359779
E-MAIL	apm@ansa.co.uk

Contents

1 Introduction

1.1 Scope

This document specifies the implementation architecture for the ESPRIT End-to-end Internet Security Project (E2S), project number 20.563. It is the public version of an internal deliverable (Project Task D1). The specification will be revised as the project continues to reflect:

- changes in user requirements
- changes in available technology
- feedback from the E2S pilot demonstrator projects.

The complete E2S implementation architecture consists of:

- this overall specification
- a set of component specifications
- examples of application of the architecture
- guidelines for implementing systems using the E2S architecture.

This document sets out the architecture - i.e., the common technology framework - for the E2S project pilot demonstrators. The architecture identifies, positions and outlines the function of the main technologies used in the project. Importantly it determines the trust and security model for the demonstrators.

Detailed architectural specifications for the components identified in this report will be produced as deliverables of E2S “infrastructure component” tasks. The examples of application of the architecture and guidelines for implementing further systems using the architecture will be produced as deliverables of E2S “pilot demonstrator” tasks.

1.2 Audience

The document is intended to explain the scope and rationale of the E2S common technology framework to a technical audience.

1.3 Structure of document

This document is divided into the following sections:

1. Introduction (this section)
2. Security model
   - a description of the security functions and security information used in the architecture and the trust placed in users, administrators and security technology.
3. Summary

— a brief description of the framework as a whole to provide an overall picture for the following three sections describing the full detail of the framework

4. Client technology
   — Technology for user interaction with end-to-end secure Internet applications
5. Secure connectivity technology
   — Technology for securing an end-to-end Internet path between users and applications
6. Server technology
   — Technology for supporting securely accessed Internet applications
7. Examples
   — A summary of the choices made from the common technology framework in the E2S project pilot demonstrators
8. Viewpoint analysis
   — Concepts and rules from the architecture categorised in ISO/ITU ODP viewpoints [ISO 10746-3] to enable alignment of the E2S implementation architecture with other standards for open distributed processing.

1.4 Background

The E2S architecture has been derived from a pragmatic assessment of E2S partner requirements for technology appropriate to large scale electronic commerce. These requirements determine both the functionality required by the pilot demonstrators and the constraints on architecture and technology choices dictated by regulatory concerns, availability of standards and market pressures (hence the positioning of the architecture as an “implementation architecture” rather than a “reference model”).

The criteria for building the architecture were:

- **universality** - the security provisions should be widely applicable to as many Internet and Intranet electronic commerce scenarios as possible (i.e., the provisions should not be specific to the E2S pilot demonstrators)
- **security** - the architecture shall embody the high levels of security required to enable businesses to put trust in electronic processes
- **reliability** - the architecture shall embody the high levels of resilience and recovery necessary to support mission critical functions
- **portability** - the architecture shall accommodate multiple computing platforms
- **effectiveness** - the architecture shall be implementable with minimal changes to existing paradigms and APIs
- **performance** - the architecture should not make applications slower or more difficult to use
- **durability** - the architecture should anticipate expected changes in Internet and platform technology

- **end-to-end** - the architecture should provide security for the complete path from a user on the Internet through to the supporting applications and data on the internal networks (“Intranet”) of the organisation delivering an electronic commerce service to the client
- **business-to-business** - in addition to enabling electronic commerce between users and electronic commerce applications, the architecture should allow for business-to-business transactions in which the “user” is a computer application running on behalf of an organisation.

1.5 Application

Within the E2S project, the architecture will be used to develop a common technology framework across four pilot demonstrators:

- Secure telecooperation in administration (Technical University of Berlin)
- Customer support for printer servicing and repair (HP)
- Internet investment banking (Swiss Bank)
- On-line merchant services (Octacon).

2 Security model

The purpose of this chapter is to explain the security philosophy and model that underpins the E2S Implementation Architecture.

2.1 Secure electronic commerce

2.1.1 Security policy

Security is a balance of risk against cost; it is not practical to defend against every possible threat particularly when the risk (e.g., financial loss, bad publicity) associated with the threat is small. This in turn means that there is unlikely to be a single security design which meets all the needs of all applications. For this reason the E2S Implementation Architecture consists of a framework of:

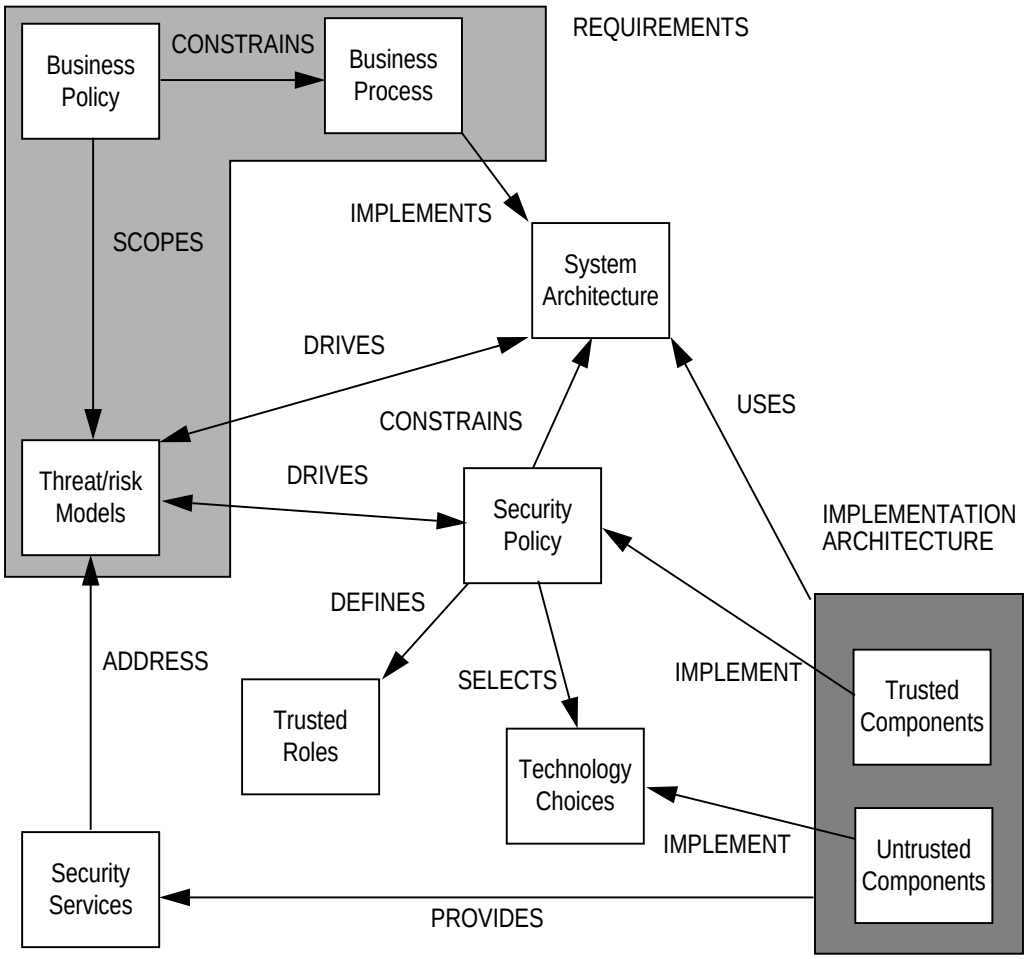
- *system components*
  - **trusted** components that provide the foundations for security
  - **untrusted** components that provide the means of delivering services
- *rules* for combining those components to deliver services securely, end-to-end
- *guidelines* for selecting appropriate components to address specific needs.

The use of the implementation architecture is illustrated in Figure 2.3. The figure shows how the architecture for an E2S system (e.g., one of the pilot demonstrators) is derived from the E2S Implementation Architecture:

- the **system architecture** specialises the E2S architecture by adding components and functions required to support a **business process**
- the business process is constrained by **business policy**
  - some of which may be required by government or regulation
  - some of which defines “corporate practice”
  - business policy scopes the requirement for security in the system
  - **security policy** for the system is an outcome of quantifying the acceptable level of **risk** associated with **security threats** against the business policy
- the security policy defines
  - the level of trust associated with different **user roles**
  - the **trusted components** upon which security is founded
  - acceptable **technology choices** from the E2S Implementation Architecture

- technology selected from the E2S Implementation Architecture provides security functions which reduce the risks associated with security threats to an acceptable level.

Figure 2.1: Security and architecture



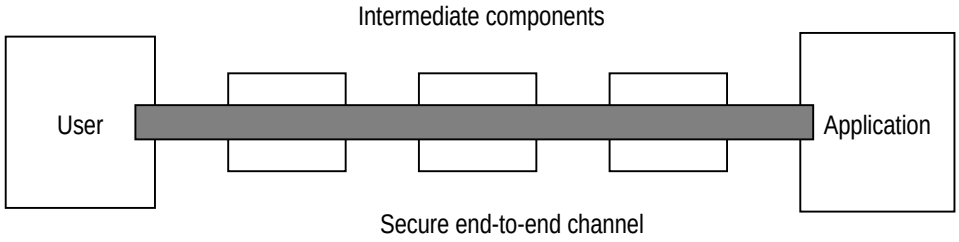
2.1.2 End-to-end security

To argue that a system is secure, the designer must show that business policy, security policy and security services are consistent. For a large system where many components are involved this can be a difficult task. Therefore the E2S Project has focussed on **end-to-end security** which is easier to analyse. In simple terms, “end-to-end” describes the approach in which security functions are divided between the user and the application to provide a “secure channel” between them, without requiring any security guarantees from the intervening networks and computers, as illustrated in Figure 2.3 below.This approach is contrasted with conventional “hop-by-hop” security techniques.

From an analysis of user requirements it is evident that the common security requirement of the secure channel is mutual authentication - the parties at either end are sure of each other's identity. Other requirements such as confidentiality and transaction integrity are application specific. Therefore, the scope of the E2S Implementation Architecture is:

- components required to create a mutually authenticated channel end-to-end, and
- a tool-kit for building application-specific security protocols.

Figure 2.2: End-to-end security



2.1.3 Security technology

From an analysis of user requirements, available standards and technology, the E2S project has chosen the use of **smartcards** and a **public key infrastructure** as the means to achieve authentication:

- public key infrastructures are a widely accepted technology for user authentication, moreover there is an emerging market of public key infrastructure providers (e.g., Verisign Inc., Ice-tel) becoming available
- smartcards overcome many of the weaknesses of password schemes and the risks of storing cryptographic keys on insecure computers; they are physical tokens of security which brings advantages in usability and manageability.

There will often be technology constraints which require parts of a system to use conventional security technology (viz., techniques that are not necessarily end-to-end), for example to created trusted network paths to management interfaces. This is permitted by the E2S Implementation Architecture, but is not included in the security analysis. It is the responsibility of the system designer to show that the introduction of conventional technology has not compromised the integrity of the system.

2.2 Security analysis

This section presents a security analysis of the trusted components in the E2S Implementation Architecture.

2.2.1 Security assumptions

The security of an E2S system depends on:

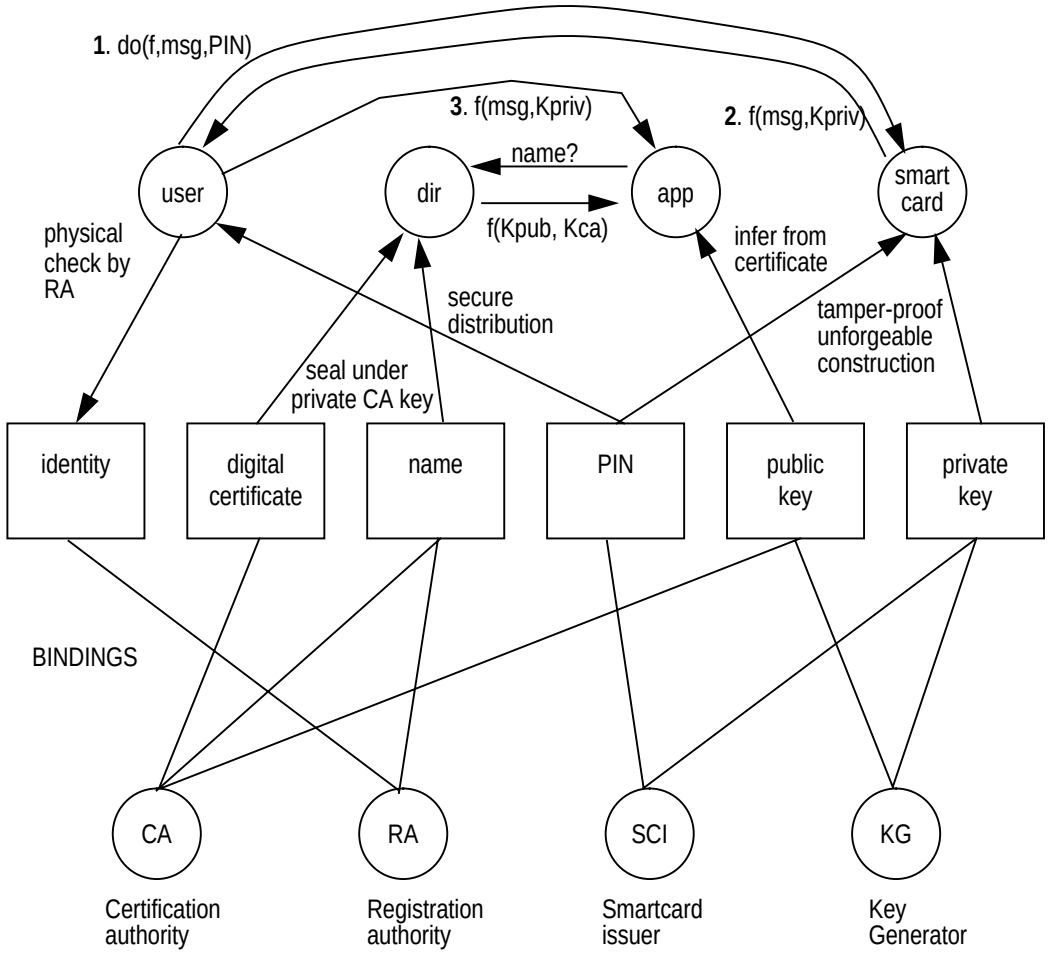
- **public key cryptography**
  - — the private key associated with a public key is a secret
  - — the public key is securely associated with its owner, and this association can be verified globally
  - — the cryptographic functions performed using a private key can only be verified using the corresponding public key (and vice versa)

  - the private key cannot be predicted from knowledge of the public key (and vice versa)
- **controlled access** to interfaces
  - based on controlled checking of a user assertion of identity, role or purpose
- **digital signatures** to confirm origin (a digital signature is a bit pattern that can only have been produced by the owner of a private key, and which can be verified only by using the corresponding public key)
- **digital sealing** to confirm content (a digital seal is a cryptographic digest of content that can only have been produced by the owner of a private key, and which can be verified by using the corresponding public key)
- **digital sealing** of time stamps, sequences numbers etc., to confirm timeliness, prevent replay and tie together transaction steps
- **encryption** to provide confidentiality
- the ability to **identify** individual people and roles by name
  - so that usersand roles can be distinguished within the system in terms of their names
- the ability of **people to keep secrets**
  - so that transaction steps enabled by knowledge of a secret can be undeniably accredited to a person or organisation
- the use of **smartcards**
  - as a tamper-proof, unforgeable means to store data (in particular, private keys)
  - as a means to apply cryptographic functions over both data stored on the smartcard and input data (e.g., encryption of messages under a private key)
- **secure administration**
  - the minimum requirement is to transfer secrets between system administration components
  - use of trusted network communications for on-line distribution of secrets
  - physically secure communication for off-line distribution of secrets (and bootstrap of trusted networks)
- **secure storage**
  - of access control information
- **trusted operational support**
  - assured correct operation of security measures for both operational and backup systems
- trustworthy system **administrators**
  - who control security information.

2.2.2 Security relationships

The E2S security relationships are illustrated in Figure 2.3.

Figure 2.3: Security model



The key to the security model are the relationships between the security resources shown in the centre of diagram (identity, digital certificate, etc.). These relationships are maintained by a set of security agents (certification authority etc.) shown at the bottom of the diagram. On the basis of these relationships, the security resources can be distributed between users, smartcards, applications, and directories in such a way that:

- a user can be issued with an individual smartcard
- the user can use that smartcard as a means of initiating mutual authentication across the Internet with an application
- no other user can achieve authentication (even if that user acquires the smartcard)
- this in turn provides the foundation for further interactions to establish access permissions and/or additional security resources (e.g., session keys) required for transaction integrity, non-repudiation and confidentiality.

The basic steps of the authentication process in the figure show the user requesting the smartcard to perform a cryptographic function on a message. The smartcard returns the result of applying the selected function over the message and a private key held within the card. This information can be sent

across the network and used by the application to authenticate that it has been sent a message by a the particular user associated with the private key.

The enterprise objects in the security model are described below (agents are shown in **bold** type, resources in *italic* type):

- a **user**
  - with a distinct *identity*
  - to be authenticated to an **application**
  - trusted to remember, and keep secret a *personal identification number (PIN)*
  - assigned a *name* by a **registration authority**
  - assigned a *private, public key pair*
  - assigned a *digital certificate* by a **certification authority**
- a **smartcard**
  - securely encapsulating a *PIN*, a *private key, a public key, and public key infrastructure* information
  - securely providing on-board cryptographic functions
  - issued to a **user** by a **smartcard issuer**
  - enabled by input of the *PIN*[1]
- a **directory**
  - of *name* to *digital certificate* mappings (a *digital certificate* is a digitally sealed record containing a ***user name*** and an associated *public key*)
- a server **application**
  - protected by access control based on **user** authentication
  - to be authenticated to a **user**
- a **certification authority**
  - for unambiguously assigning *public keys* to *names*
  - for creating *digital certificates* for valid *public key, name pairs*
- a **registration authority**
  - responsible for identifying the **user** and associating an unambiguous *name* with the **user**
- a **smartcard issuer**
  - responsible for issuing a **smartcard** to the **user**
  - containing the **user**'s *private key*
  - enabled by an unpredictable *PIN*
- a **key generator**[2]
  - responsible for creating an unpredictable *public key, private key pair*

1. A PIN-protected smartcard requires that the user input the correct PIN when the card is inserted into a reader, to prevent inappropriate use of a lost or stolen card. Thus the PIN is only for access control to the card and could be replaced in the future by biometric recognition for example.

— The *private key* is a shared secret between the **smartcard**, the **key generator** and the **smartcard issuer**; communication between these entities must be secured either using trusted networks or safe physical information transfer (e.g., registered post, trusted courier, etc.).

2.2.3 Security rules

The smartcard is not a secret. It is only enabled when inserted in a smartcard reader and with the correct PIN input. The connections between smartcard and the PIN input device and between the application software and the smartcard must be secure (e.g. by using a verified copy of a trusted operating system)[1].

The PIN is a secret shared between the user, the smartcard and the smartcard issuer; communication between these entities must be secured either using trusted networks or safe physical information transfer.

The user's public key is not a secret.

The integrity of the binding between a user's public key and the user's name must be trustworthy. This is achieved by having the binding represented as a digital certificate constructed by a certification authority. The certificate must be globally available (i.e., by replicating it widely).

The private key used by the certification authority to sign and seal the digital certificate is a secret and must be kept confidential to the certification authority (e.g., by creating the certificate off-line in a physically secure location).

The public key corresponding to the certification authority's private key is not a secret. It is required to be available to the application. The public key is trusted and should be made globally available by replicating it widely, and by cross-certifying with other certification authorities.

The registration authority must use physical means to ensure the identity associated with the user and bound to the name is correct (e.g., by reference to legal documents, physical characteristics and so forth).

Since digital certificates are self-describing, directories of certificates need only be protected against denial of service attacks.

The application must include an access control function. Any table of name to privilege rules within this function must be stored securely to prevent tampering, and any communication between an application component and the access control function must be secure if they are not in the same physical location.

---

2. Key generation is also the point at which key escrow may occur to meet government / business regulatory constraints, and at which keys might be securely archived to enable key recovery after accidental destruction of a key.

1. A particular concern here, especially with personal computer operating systems, is the risk of virus or trojan horse attack either via the network, or via infection of software distribution media (i.e., disks). It is anticipated that during the lifetime of the E2S project, operating systems vendors will improve their defences against such attacks. However, a system designer should take such risks into account when using the architecture, for example, by putting less trust in personal computers that are not under the supervision of a trusted IT administrator.

The registration authority, certification authority and smartcard issuer must cooperate to ensure that, for a given valid smartcard, digital certificate, user triple:

(i) the name in the digital certificate corresponds to the user

(ii) the private key in the smartcard corresponds to the public key in the digital certificate

(iii) the PIN known to the user is the PIN known to the smartcard.

From these security assumptions it is possible for:

(i) the user to ask the smartcard to perform a cryptographic function on some data

(ii) the transformed data to be sent to the application

(iii) the application to verify that the transformed data could only have originated from the user associated with the name.

This provides sufficient information:

— to enable authentication and hence access control

— to enable the generation of secure tokens and sequence numbers in business protocols for secure transactions.

A user's right to access an application can be withdrawn:

- by changing the application's access control policy
  - — which does not affect the ability of the user to access other applications
- by the certification authority placing the digital certificate in the directory on a certificate revocation list
  - — which effectively "cancels" the user's smartcard, provided applications check the directory as part of validating a user's key
  - — which requires the directory to be highly available and efficient.

3 Architecture summary

The E2S architecture is summarised diagrammatically in Figure 3.1.

The global areas of technology covered by the E2S implementation architecture are:

- **client technology**, concerned with user interaction
- **secure connectivity technology**, concerned with securing an end-to-end Internet path between users and applications
- **server technology**, concerned with supporting Internet applications

Structurally,

- each technology comprises a set of features from which an E2S implementation can select (such as security management)
- each feature depends upon an underlying set of infrastructures (such as key management)
- each infrastructure is made up of a number of architectural components.

The set of features included in the scope of the architecture has been driven by an analysis of user requirements and a desire to maximise the use of common technology. Thus, the design of the E2S Implementation Architecture is intended to enable the re-use of infrastructure components across a wide set of features and hence application scenarios.

A system conforms to the E2S Architecture if it makes a consistent choice of features from each area and satisfies the security model specified in Chapter 2.

The rules for making consistent choices are specified in the separate reports giving the detailed architecture for each feature and associated infrastructures. The top-level description in this document focuses on the major relationships and the overall management of security in E2S.

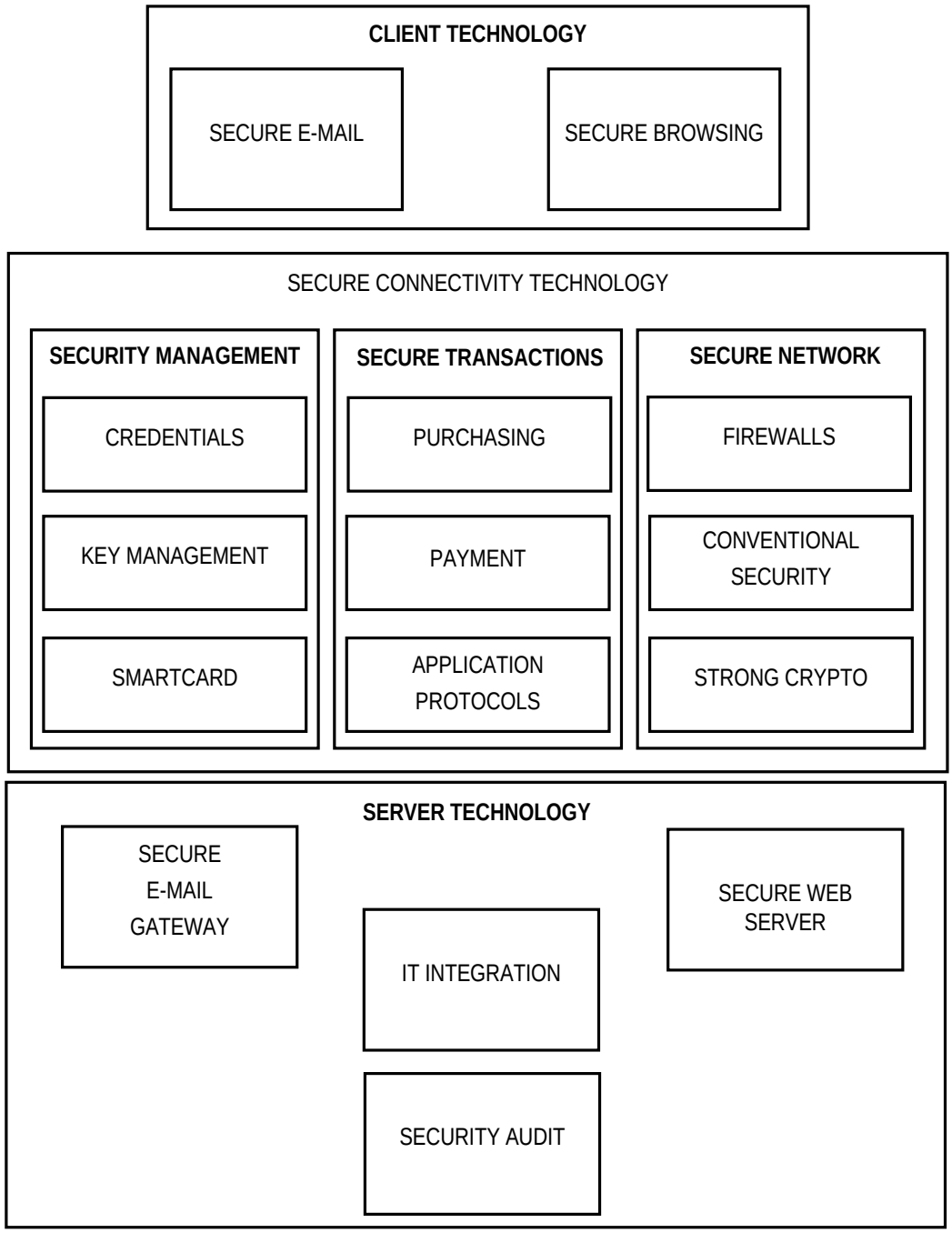
The architecture can be used recursively, in the sense that management functions for E2S components can be implemented themselves as end-to-end secure applications (for example an HTML forms-based interface for updating a registration authority's directory service). In using the architecture recursively the designer must be sure that cyclic dependencies are not created.

3.1 Client technology

Client technology is used to provide the interface between users (i.e., on-line customers) and the servers which are providing electronic services to those users.

The need for two kinds of client features have been identified in the analysis of user requirements:

Figure 3.1: Architecture Summary



- **secure electronic mail** for **telecooperation** involving the secure exchange of messages, documents and instructions.
  - — for use where business processes and data protection regulations are formulated in terms of document handling policies. It is also well suited to applications serving users who may not be on-line continuously (e.g., out-of-office sales staff with laptop computers).
- secure web browsing for transactional **interactive sessions** to enable activities such as searching, selecting, ordering and reporting.
  - — for applications where rapid access to a wide range of information and a fast response is required. It requires that the user remain on-line

for the duration of a session (e.g., to purchase a selection of goods). Most current technology is forms-oriented, but in the future will include the capability to download more intelligent interfaces using scripts and applets.

Associated with both kinds of client technology is the need for **user authentication** based on **smartcards**.

3.2 Secure connectivity

A pre-requisite of end-to-end security is connectivity technology to secure interaction between clients and servers.

Analysis of the end-to-end security aspects of E2S user requirements shows the need for three features to secure connectivity:

- **Secure network infrastructure**
  - technology used to secure individual components, network links and (or) sub-networks where an end-to-end solution is insufficient, unavailable or inappropriate
- **Secure transaction infrastructure**
  - protocols to ensure that electronic business transactions can be secured using unforgeable signing to confirm origin, unforgeable sealing to confirm content, sealed timestamps or sequence numbers to check timeliness and prevent replay, link transaction steps and encryption to provide confidentiality
- **Security management infrastructure**
  - components for making, distributing, verifying and revoking cryptographic keys and for issuing smartcards.

3.2.1 Secure network infrastructure

Secure network infrastructure comprises three components:

- **firewalls** for controlling entry and exit between security domains and to provide a location for security management and auditing
- **conventional security** technology (such as trusted operating systems, secure link-level / transport level protocols, password-based authentication) for inter-operability with other security architectures
- optional **strong cryptography** as a means, where its use is permitted, to increase the resistance of security protocols to attack.

3.2.2 Secure transaction infrastructure

E2S user requirements show the need for three features in secure end-to-end transactions:

- **an application protocol tool-kit** to enable construction of protocols (such as GMD's BaKo) to represent end-to-end procedures and maintain appropriate levels of privacy, obligation and non-repudiation between interacting parties.
- an electronic **payment** infrastructure enabling electronic payments linked via bank card organisations deploying the *Secure Electronic Transactions* (SET) standard (e.g., VISA International).

— bank-card based payment has been selected for E2S since it is international in scope and has a well-understood financial risk model compared to other forms of electronic payment. Moreover pilot SET infrastructures are being created in the time-scale of the E2S project.

- a **purchasing** infrastructure (including a network of supporting banks) for electronic business-to-business corporate purchasing (e.g., of office supplies), based on corporate bank cards.

3.2.3 Security management

To support the E2S security model, three features of security management are required:

- a **credentials management** infrastructure for maintaining relationships between security information belonging to different security domains (e.g., a secure electronic transactions domain and a corporate IT domain)
- a **key management** infrastructure for making, distributing, checking and revoking cryptographic keys used for authentication and access control[1]
- a **smartcard** infrastructure for issuing and verifying smartcards.

3.3 Server technology

Analysis of E2S user requirements shows the need for two different features for delivering secure services to users, one based on electronic mail, the other based on secure user sessions.

Alongside this user-facing functionality is a need to integrate electronic commerce technology with “back office” applications.

To meet a requirement for continued assurance of system security there is additionally a need to monitor and audit server technology.

E2S server technology comprises:

- a **secure e-mail gateway** infrastructure to act as the focus for applications based on secure email. The server provides secure mail boxes and functions such as re-distribution of mail directed to an organisational unit
- a **secure web server** infrastructure acting as the focus for user sessions initiated by client browsers
- **IT integration** infrastructure enabling back office applications to be exported via mail gateways and web servers. It is concerned with the connectivity between the Internet (via which clients access services) and internal “Intranets” on which services are deployed. IT integration includes the capability to download “applets” from the server to the client to enable customisation of the client interface

1. Key generation for secure transport protocols or for confidentiality in business protocols is a separate function from the key management infrastructure for authentication (although algorithms and mechanisms might be shared). E2S has no user requirement for escrow of keys used for secure transport protocols or for confidentiality in business protocols, although this may be forced by regulation or law in some countries.

- **Security audit tools** are to monitor and ensure the integrity of a system implemented using the E2S common technology framework.

4 Client technology

Client technology enables a user (i.e., a **person**) to interact securely across the Internet with an **application**.

It comprises:

- secure electronic mail
- secure browsing.

4.1 Secure electronic mail

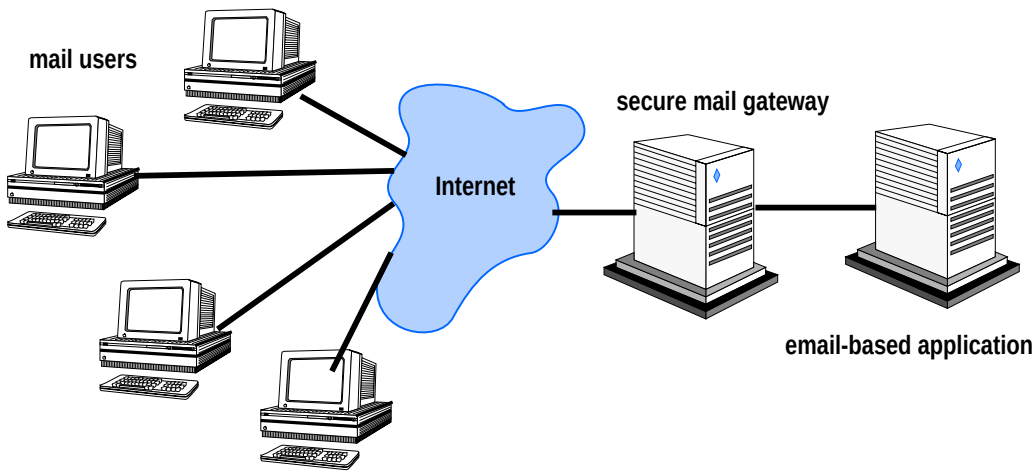
Secure electronic mail is required for applications where business processes and data protection regulations are formulated in terms of document handling policies. It is also well suited to applications serving users who may not be on-line continuously (e.g., out-of-office sales staff with laptop computers).

Secure electronic mail enables **telecooperation** based on the secure exchange of messages, documents and instructions for:

- publication of authentic information
- confirmed delivery of information
- secure access to sensitive information.

Secure electronic mail is illustrated in Figure 4.2. A set of mail users can communicate securely with one another by electronic mail and interact with applications (such as document servers) by sending commands and receiving results as mail messages.

Figure 4.1: Secure email scenario



The secure mail gateway acts as a repository for messages in transit and also provides support for sending mail to **distribution lists** identifying **group**s of users.

Thus secure electronic mail supports user-to-user, user-to-business and (if a program is substituted for a mail user) business-to-business electronic commerce.

Mail users require, in different situations, combinations of

- confirmation of the origin of messages
- confirmation of the content of message
- guarantees that messages will only be delivered to the intended recipients
- confidentiality for messages.

Therefore electronic mail uses **key management** and **business protocols**, in which the protocol will sign, seal and encrypt mail messages appropriately for the transactions taking place.

Two kinds of client email technology are defined in E2S:

- **security enhanced mailer**
- **protected insecure mailer.**

4.1.1 Security enhanced mailer

A security enhanced mailer is one which supports cryptographic sealing and signing of messages using for example, Privacy Enhanced Mail (PEM) [PEM] or Pretty Good Privacy (PGP) [PGP] technology.

A security enhanced mailer requires cryptographical functions using the user's private key. If the mailer runs on a computer accessible to other users, the functions on the user's key should be accessed via a **smartcard**[1].

A security enhanced mailer exchanges mail with

- other security enhanced mailers
- **secure mail gateways**
  - to communicate with users with **protected insecure mailers**
  - to communicate with user **group**s.

4.1.2 Protected insecure mailer

An important requirement for electronic mail-based electronic commerce is to provide access for users with mail packages which have no built in support for security (e.g., Eudora [EUDORA]).

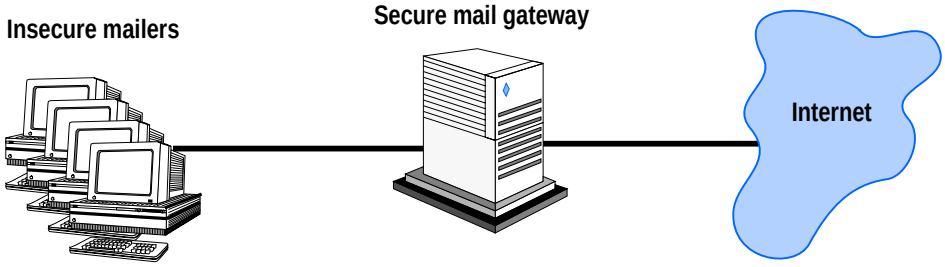
Insecure mailers must be protected by a **secure email gateway**, as illustrated in Figure 4.2.

All mail to and from the workstations to other computers is intercepted by the **secure mail gateway**. This gateway cryptographically signs, seals and possibly encrypts outgoing mail according to a security policy. The gateway checks the seals of incoming mail and decrypts it if necessary, appending to the plaintext contents an explanation of the trust that may be put in the message (e.g., indicating the message was signed by a particular individual and sent confidentially).

---

1. An alternative, when the associated risks are acceptable, is to store user's private keys on the computer's disc, encrypted by a password / pass phrase. Whenever the mailer needs to access the key, the user is requested to input the password and the mailer decrypts a temporary plain text copy of the key which is destroyed after use.

Figure 4.2: Protected insecure mailers



Secure mail gateways are described in more detail in Chapter 6 - *Server technology*.

Insecure mailers must be subject to **security audit** to ensure that:

1. insecure mailers are isolated from the Internet e.g., by positioning behind a **firewall** to avoid either “spoof” mail being accepted or sensitive mail being allowed to escape
2. one user does not masquerade as another (e.g., by associating passwords / pass phrases with names, or putting computers in secure locations associated with named users).

4.2 Secure interactive sessions

Secure interactive sessions meet the user requirement for interactive on-line applications of electronic commerce.

Secure interactive sessions are illustrated in Figure 4.2. A Web browser (e.g. Netscape Navigator 2.0/3.0 [NETSCAPE]) is shown, providing a page-oriented interface to a Web server over which HTML [HTML] and other forms of document can be displayed to the user. HTML includes the provision for the return of filled forms from the user to the server, either for processing within the server, or for hand-off to a back office application through **IT integration** technology.

A web page may include an **applet** which provides a custom user interface, implement a **business protocol** and/or provide other client-side support for the server-side application.

Thus secure sessions provide user-to-business and (where a program is substituted for the browser user) business-to-business electronic commerce.

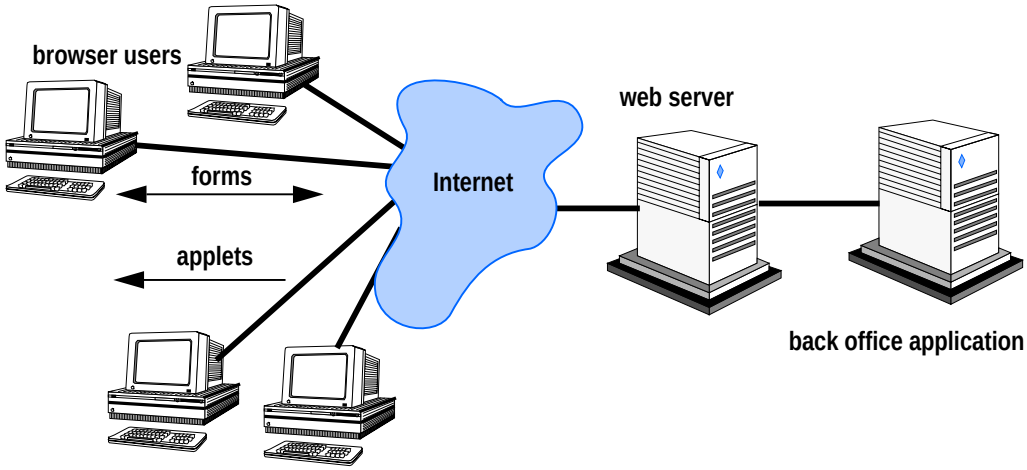
The browser user requires:

- a guarantee that the session is with a server under the control of a trusted business
- the forms downloaded and returned are not tampered with (and in some situations remain confidential)
- that applets downloaded are not malicious or of poor quality.

In order to achieve this, a browser must do one of the following:

- provide a means for **user authentication** to the web server (and then use a **secure transport protocol** to transfer forms between client and server)

Figure 4.3: Web browsing scenario



- be extended by the addition of **helper applications** which implement a secure **business protocol**.
  - The business protocol will perform mutual authentication of user and server, sign, seal and encrypt forms appropriately as they are passed back and forth. (There is a trade-off between security inherent in the business protocol and dependence upon security in the transport protocol)
- download applets that implement business protocols.
  - Before loading the applet, the user and server providing the applet should authenticate themselves to one another to avoid the user down-loading an untrusted applet.[1]

The selected web browser for E2S is Netscape 2.0 (and subsequent versions), which in turn implies HTTP encapsulated in the Secure Sockets Layer protocol (SSL) [SSL] as the secure transport protocol for this scenario.

1. Web “applet architectures” (e.g., SUN’s Java) are proposing mechanisms for checking that an applet has been signed by a trusted supplier, at which point the authentication check can be more specific (i.e., that a particular applet creator, rather than the site that provides it, is trusted).

5 Secure connectivity technology

Secure connectivity technology provides end-to-end security across the Internet between clients and applications. It comprises:

- security management
- secure transactions
- secure networking.

5.1 Security management

Security management is primarily concerned with the management of the cryptographic keys used in secure communications and business protocols.

Security management comprises:

- credentials management infrastructure
- key management infrastructure
- smartcard infrastructure.

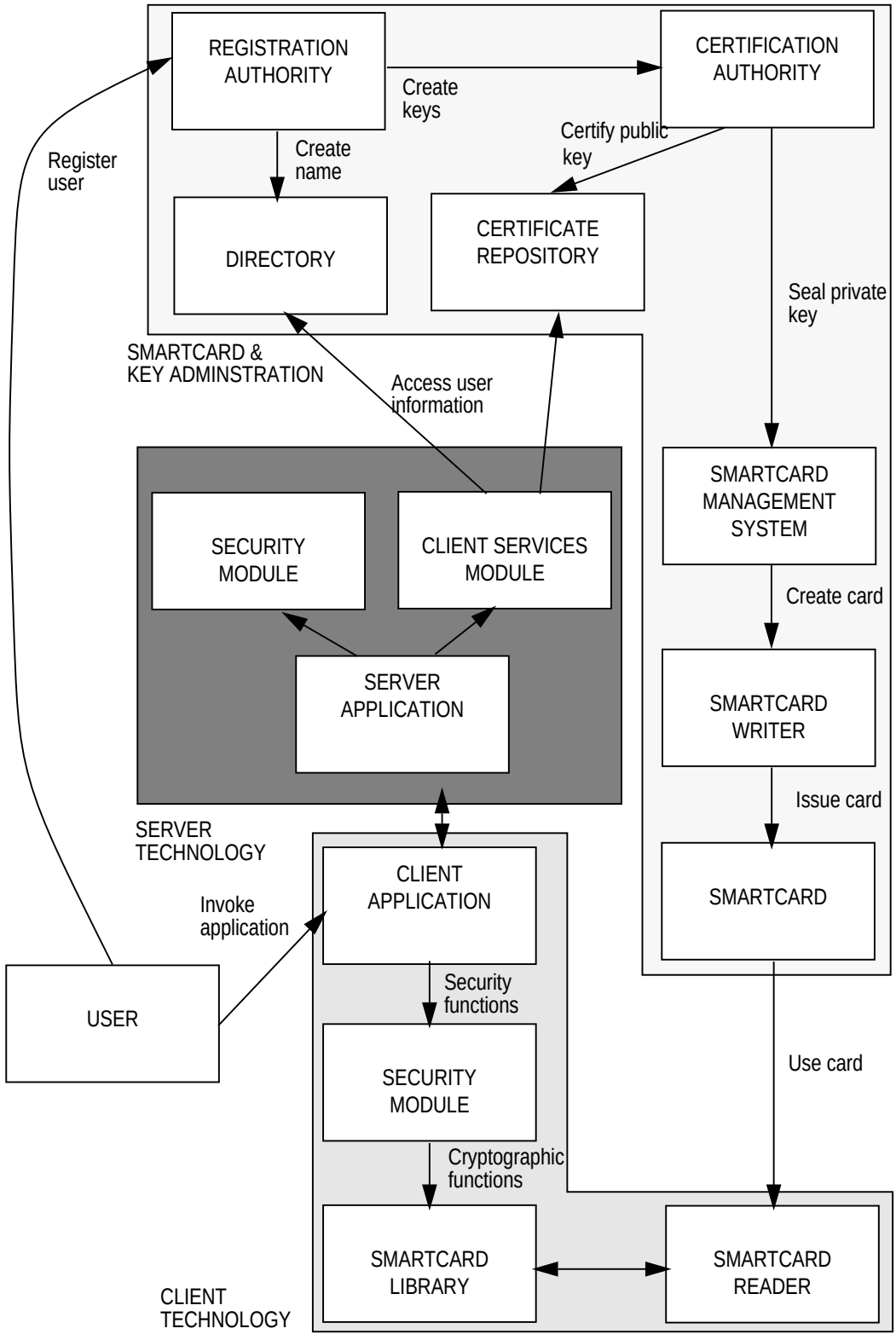
In the interests of clarity, credentials management is described last.

A summary of the major components of the key management and smartcard infrastructures is shown in Figure 5.1. The figure shows how the components divide into three groups:

- those associated with clients (i.e., security enhanced mailers and web browsers)
- those associated with servers (i.e., secure mail gateways and web servers)
- those associated with security administration.

The arrows in the figure denote the principal service requests that occur between the components.

Figure 5.1: Smartcard and key management



5.1.1 Key management infrastructure

Secure communication based on public key cryptography relies on securing the association between cryptographic keys and their owners as explained in Chapter 2.

The functions of the key management infrastructure are:

- **key generation**
  - key generation consists of generating public/private key pairs.
  - key generation can be devolved to the owner of the key (as in PGP) or it can be under the control of the key management infrastructure manager
  - if keys are created by the infrastructure manager they must be distributed securely to the owner (e.g., embedded in a smartcard)
  - if a secret key is distributed other than on a smartcard, the key owner must store it securely
- **certification** - i.e., associating a key with a name in the form of a **digital certificate**
- **registration** - i.e., associating a name with a person
  - registration information is held in on-line **directory services**
- **attribute assignment** - i.e., associating an attribute to a person, for example
  - a **role** (e.g., “head of department”)
  - a **capability** (e.g., “access to sales statistics”)
  - **access privileges** or **encryption restrictions** (e.g., “for bank use only”)
- **verification** - that a purported key can be trusted for the purpose to which it is applied (e.g., “Giles S. Murchiston, acting in the role of repairs budget holder, purchasing printer spares”)

The components of a key management infrastructure are:

- **certification authority**
  - responsible for issuing and revoking digital certificates and certificate revocation lists
- **client services module**
  - providing access to the directory and certificate repository for **secure mail gateways**, **web servers** and **IT integration** components
- **certificate repository**
  - responsible for storage and retrieval of certificates
- **security module**
  - responsible for storing keys in a computer and performing cryptographic functions[1]
  - E2S security modules are built using the SecuDE tool [SECUDE]
- **key administration**
  - an interface to the infrastructure, for policy management and audit.

The inter-relationship and structuring of instances of these components depends upon the needs of the particular application and community of users. In particular two cases are supported:

- **external registration and certification authorities** such as Ice-tel and Verisign Inc. [VERISIGN, ICE-TEL] in systems where access is to be granted to the general public (i.e., users are not pre-registered)
- **internal certification and registration** authorities where the service provider delivers key management alongside a service, either as part of that service, or to control the "branding" of the service, or to restrict use to a registered set of users.

While not necessarily so, internal certification and registration authorities are used when the service provider issues users with **smartcards**. (The service provider could install an external certificate on a smartcard but this is not yet common practice.)

The security of an E2S system depends upon the security of both certificate repositories and security modules. Both should be subject to **security audit**.

Certificate repositories and directories should be secure and the interface for adding and/or changing keys and relationships made available only to trusted **key management infrastructure security manager**s. This can be achieved by locating repositories and directories in physically secure locations, isolated from the Internet by **firewalls**.[1]

Registration authorities are required to take prudent steps to be sure the person they associate with a name is the appropriate individual, for example by checking passports or similar legal identifications against the person claiming to associated with the name being registered.

E2S key management uses X.500 directory services [X.500], X.509 digital certificates [X.509], and both PEM [PEM] and PGP [PGP] technology from the SecuDE [SECUDE] and Osisec [OSISEC] tool-kits.

5.1.2 Smartcard infrastructure

E2S has selected smartcards as the preferred means of issuing keys to users and for key verification because a smartcard is:

- a personal, physical token - to use it requires both the presence of the card and knowledge of a secret personal identification number (PIN)
- a strong, tamper-proof and unique location for a user's private key
  - — because of the card's physical properties the key cannot be duplicated, moved or falsified.

1. The security module associated with a client performs cryptographic functions on keys via access to a smartcard. The security module associated with a secure mail gateway, web server or IT integration component may also be provided with keys and cryptographic functions in the same way. Sometimes (for legacy or cost reasons) it is more convenient to store keys and provide cryptographic functions within the component itself. In this case the component must be subject to security audit to ensure it cannot be compromised.

1. The protection of the certificate repository can be reinforced if the root key is used to create a set of "operating keys", and the repository for the root key then disconnected from any network and stored in a safe. If an operating key is compromised, an alternative operating key can be substituted. If the root key is compromised there is no means of recovery, except to re-issue all keys.

A smartcard infrastructure consists of **smartcard technology** and a **smartcard management system**.

The security of smartcards depends upon the PIN remaining a shared secret between the smartcard issuer and the smartcard user.

5.1.2.1 Smartcard technology

Smartcard technology consists of

- a smartcard architecture
- a smartcard reader/writer device
- a software library for access to smartcards via the reader/writer device.

E2S requires a smartcard architecture in which a smartcard can

- store data (keys, certificates) with a high level of security
- verify digital certificates
- sign and verify blocks of data (e.g., protocol messages)
- generate truly random numbers (e.g., for use as session keys)
- encipher/decipher data.

(An architecture with “on-board” cryptographic processing allows the card to be used with relatively insecure operating systems, since the keys in the card cannot be read, compared to keys on a local disk.)

The E2S project has selected the GEMPlus GPK2000 card to meet these requirements. It supports:

- RSA , DSA and DES algorithms
- true random number generation
- SHA-0, SHA-1 and MD5 hashing
- storage of application data.

5.1.2.2 Smartcard management system

The smartcard management system provides the link between the **smartcard infrastructure** and the **key management infrastructure**.

The smartcard management system is responsible for issuing smartcards to users:

- receiving a set of private keys from a public key infrastructure manager
- creating a smartcard encapsulating the keys and assigning the PIN
- securely delivering the smartcard and knowledge of the PIN to the smartcard user (e.g., by postal mailing in separate packages)

The communication between the smartcard infrastructure and the key management infrastructure must be secure[1] and subject to security audit to avoid the loss or compromise of keys before they are encapsulated within a smartcard.

1. This security can be achieved by co-location of key generation and smartcard issuing, by using secure transport between the two infrastructures or by recursive use of the E2S architecture (i.e., treating card issuing as an application).

5.1.2.3 User authentication with smartcards

Smartcard based authentication is the recommended form of authentication in E2S. The guarantee of mutual authentication between the user and the business providing the server he is using must persist only for the duration of a session. If the user removes his smartcard the client technology should request re-authentication before processing further interactions.

All of the client technology for user authentication (computer, operating system, keyboard, card reader, security software) must be verified to be free of security flaws and immune to virus or trojan horse attack.

5.1.3 Credentials management infrastructure

The purpose of credentials management is to maintain relationships between sets of **digital certificates**. For example some business processes require both proof of identity and proof of ability to pay. Both of these can be represented as certificates, one issued by a **certification authority**, the other by a **payment infrastructure**. Credentials management takes such certificates, **verifies** them individually and then delivers an access control decision (perhaps in the form of another certificate - i.e., a **capability**) based on the results of the verification and an access control policy.

Credentials management is a trusted system component and must be subject to **security audit**, in particular to ensure that:

- only trusted credentials managers can change access control policy
- access policy rules are stored securely.

This can be achieved by putting credentials management in a physically secure location isolated from the Internet by **firewalls**.

5.2 Secure transactions

The secure transactions feature of the E2S architecture consists of protocols that use cryptographic techniques to ensure electronic business processes can be trusted, for example by using cryptographic signing to confirm the origin and content of messages, and cryptographic sealing to protect their confidentiality.

The secure transactions component comprises three layers:

- application protocols
- electronic payment infrastructure
- corporate purchasing infrastructure.

5.2.1 Business protocols

Application protocols are the electronic analogues of the **contracts** entered into every day by **business**es and people e.g., when updating records, ordering goods or buying services.

Business transactions must therefore provide:

- confidentiality - so that transactions are only visible to the participants involved[1]
- integrity - the transaction follows a correct procedure

- authentication - the participants in a transaction are convinced of each other's right to undertake the roles they fulfil in the transaction
- non-repudiation - at the end of a transaction each participant has proof the transaction took place.

E2S business protocols will be based on exchanges of messages in PEM format so the protocols can use either electronic mail or the worldwide web as their transport.

Protocols from the SecuDE [SECUDE] tool-kit will be used for signing, verifying, sealing and unsealing those messages.

5.2.2 Payment system

Means for electronic payment is fundamental to electronic commerce. Given the objectives and time-scales of the E2S project it was important to select a payment system that:

- is convenient for users
- spans national boundaries
- has an accepted status in the financial community
- is available for use immediately.

This led to the choice of electronic bankcards, in particular the *Secure Electronic Transactions* (SET) standard [SET] and its implementation by VISA International.

SET is an open, vendor neutral, non-proprietary, license-free specification for securing on-line transactions. In order to realise the benefits from this specification E2S must develop financial relationships with banks who will act as issuing banks for SET users and acquiring banks for SET merchants.

The payment infrastructure for E2S comprises:

- the existing "**VISANet**" network for settlement of Visa transactions
- a **card-holder SET module**
- a **merchant SET module**.

SET cardholder and SET merchant are particular roles of a **person** or **group**.

A cardholder SET module will be associated with **client technology**, a merchant SET module with **server technology**.

The cardholder SET module interacts with the SET merchant module to initiate and complete a payment transaction.

The merchant SET module transfers notification of completed SET payments from the merchant to the merchant's acquiring bank.

The merchant is not required to be on-line for an SET payment to complete.

The cardholder's issuing bank creates a public/private key pair for each cardholder and securely distributes the private key to the user (e.g., embedded in a "payment" **smartcard**).

---

1. There may be confidentiality constraints within a process. For example a user placing an order and paying by bankcard may not wish the banks to know any details of the transaction apart from its identity and value - i.e., the actual goods ordered are a secret between the user and the merchant.

Since the merchant SET module stores knowledge of completed payments it should be a secured system component (i.e., protected by a firewall and kept in a physically secure location).

5.2.3 Purchasing system

Support for purchasing goods and services is a common requirement of electronic commerce. In addition to the **business protocols** required to protected electronic purchasing transactions, an on-line banking infrastructure is required to handle payment and reconciliation of accounts.

Purchasing involves a customer (i.e., a **person** or **group**) and a merchant (i.e., another **person** or **group**) providing an on-line purchasing **application.**

Customer and merchant are particular roles of a **person** or **group**.

A customer purchasing module will be associated with **client technology**, a merchant purchasing module with **server technology**.

The architecture of the purchasing infrastructure is to be determined during 1997; it will probably include as components:

- **customer purchasing module**
- **merchant purchasing module**
- **banking infrastructure**.

The customer purchasing module interacts with the merchant purchasing model to order goods and/or services and make payment for them.

The banking infrastructure distributes accounts and reports on purchases made and received to users and merchants as appropriate. Additionally it ensures the corresponding payments are made and received using the payment infrastructure.

5.3 Secure networking

Secure networking is required to ensure that electronic commerce and the support infrastructure is safe from network threats such as snooping, replay and other malicious or erroneous events.

Secure networking by itself does not guarantee security. The requirements for trusting people performing critical roles in the architecture and for physical protection of critical components must also be respected.

Secure networking comprises three infrastructures:

- firewalls
- conventional security technology
- strong cryptography.

5.3.1 Firewalls

Firewalls are used to selectively isolate computers from the Internet. A firewall creates a security domain encompassing all the computers connected to it.

Within a security domain it can be assumed that the computers in that domain can only be used by the people with physical access.

On the assumption that the **software** in the computers does not contain **security flaws**, **viruses** or **trojan horses**, and that the people with physical access will not introduce such threats, **user authentication** and access control can be trusted.

A firewall consists of three sets of components:

- **filters** to block and/or audit transmission of certain kinds of message (specified by type, destination or some combination of both)
- **gateways** which forward acceptable messages from one side of the firewall to the other
- **application proxies** which perform application specific access controls, monitoring and auditing.

Trusted system components (i.e., multi-level, compartmentalised operating systems and secure protocols) are recommended as the way to build firewalls:

- using trusted networks to segment internal networks to ensure security policy controlled information flow
- with certified levels of security
- with several individually encapsulated components, co-located on one machine for performance
- to ensure separation of user roles.

Trusted system components can use conventional security technology, provided that the keys used by the components are physically part of the component (e.g., a tamper-proof part of its electronics, or an enabling smartcard and computer in a physically protected location).

Where trusted system components are not appropriate (e.g., on the grounds of cost or needs for legacy integration) physical distribution of firewall components and physically separate network links must be employed to achieve segmentation and encapsulation (see, for example, [CB 94]).

5.3.2 Conventional security

The E2S architecture promotes end-to-end security. From this standpoint E2S has limited needs for conventional security technology, since concerns such as authentication, confidentiality, integrity and non-repudiation are addressed by the application-oriented, secure, end-to-end transactions part of the architecture (viz., smartcard-based user authentication, business protocols).

However in practical systems there will often be a need to inter-work with legacy security infrastructures, or to conform to convention security standards for access to secured data and system management functions.

Where conventional security technology is used, the system designer is responsible for assuring himself that the technology provides the guarantees required by those parts of the system adhering to the end-to-end model. (Moreover there is little point in selecting technologies that offer a stronger guarantee than that required by the end-to-end model, particular if to do so would impose a performance or scaling limit).

The following needs for conventional security have been identified in the E2S pilot demonstrators:

- alternative means of authentication when smartcard technology is not available, e.g., to:

  - gain access to private keys in security enhanced mailers
  - gain access to server technology local administration functions
  - authenticate a computer as operating on behalf of a user or an organisation
- use of PEM and PGP for encryption of mail and form contents in application protocols
- secure transport protocol
  - between web browsers and web servers (i.e., the login and SSL protocol)
  - for network access to system management and auditing functions.

If smartcard based authentication is not possible the alternatives are:

- keeping the machines in a **physically secure location** and associating a user **name** with that location
- a **challenge/response negotiation** with a secure login process.

The guarantee of mutual authentication between a user and a server must persist only for the duration of a session. Therefore the authentication process must provide a means to detect:

- the end of a session (e.g., explicit logout)
- another, perhaps unauthorised person, continuing a session when the authenticated user leaves his terminal unattended (e.g., by timing out idle sessions).

5.3.3 Strong cryptography

The security of secure protocols depends upon the strength of the cryptographic algorithms they use and on the length of keys.

Deployment of cryptography is constrained by export regulations, import regulations and government policy. Consequently the E2S architecture can only make recommendations:

- the greatest strength cryptography permitted should be used[1]
- security protocol implementations should be parameterised by algorithm and key length so that alternatives can be substituted
- specific algorithms and key lengths (except in the form of constraints on minimum size) should not be built into applications
- location information should be associated with system components so that politically correct choices of algorithms and key length can be made.

1. Where the regulatory concern is with (mis)use of cryptography for privacy, strong cryptography should still be used for signing and authentication, even though content is only protected by weak cryptography.

6 Server technology

Server technology provides the means to deliver secure services to users.

It consists of:

- secure email gateway
- secure web server
- IT integration
- security audit.

6.1 Secure email gateway

Secure email gateways are required to support secure electronic mail-based telecooperation.

A secure email gateway acts as a gateway between a secure Intranet (e.g., a LAN) and the open Internet. It guarantees that any mail exchanged with users outside the Intranet is protected from attack (theft, invasion of privacy, modification or forgery) by third parties.

A secure email gateway has both a client and a server mail interface together with a management interface. Insecure mailers access the gateway at the client interface, the mailer connects to the Internet at the server interface. The management interface provides functions to register users and manage **distribution lists** for **groups** of people.

Secure email gateways and **secure mailers** at different locations can cooperate to define a **secure email infrastructure** for the users they protect.

A secure mail gateway provides:

- message origin authentication
- message content integrity
- message content confidentiality
- message non-repudiation
- addressing distribution lists[1]
- addressing recipients by role.

A secure mail gateway provides automatically:

- signing and signature verification for both clear text and confidential messages (driven by the sender's **name**)

1. Directing mail to a distribution list may require mail to be decrypted from a public key associated with the group and redistributed as messages encrypted using the public keys of the recipients. hence the secure mail gateway is sometimes also referred to as a **secure mail exploder**.

- encryption and decryption of confidential messages (driven by the recipient's **name**).

A secure email gateway is a trusted system component and must be subject to **security audit** to ensure that

1. only trusted managers are permitted access the management interface
2. the gateway functions for signing, sealing, verifying and forwarding mail are implemented securely (i.e., are fully protected by the mail gateway's operating system, and the gateway is placed in a physically secure location).

In addition to delivering mail to user mailers, the gateway may also deliver mail to **IT integration** technology which implements a **telecooperation** application (for example, retrieving documents from a database, or driving a **business protocol** for an office procedure such as ordering supplies).

6.2 Secure web server

A web server is required to support interactive sessions as described in Section 4.2 on page 21.

The E2S project has selected the Netscape Web Server because of its wide take-up and extensibility via "helper applications". The Netscape Web Server uses the Secure Sockets Layer protocol (SSL) to mutually authenticate the user and the server at the start of a session, and to maintain the integrity and confidentiality of the session thereafter.

(SSL must be extended to use **strong cryptography** for confidentiality, when the weak cryptography used by default in Netscape[1] is deemed insecure).

6.3 IT Integration technology

IT integration technology enables back office applications to be exported via web servers to users. IT integration must satisfy two key requirements:

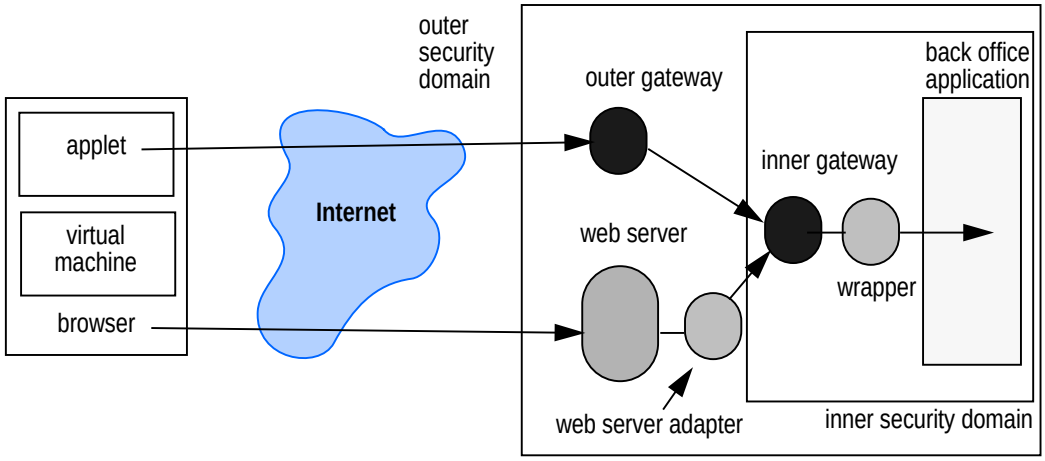
- *controlled access to data and applications in back office systems*
  - — confidential data must not be allowed to leak into the Internet
  - — mission-critical application must be protected from attack via the Internet
- *custom, branded session delivery*
  - — control of presentation to the user
  - — control over division of processing between browser, IT integration component and back office application
  - — user confidence in trustworthiness of system by virtue of trust in the "**brand image**".

The IT integration technology shown in Figure 6.1 supports these both together and separately.

The back office application and associated data is held with an inner security domain protected either by a **firewall** or by use of a **trusted network**. A

1. SSL has a cryptographic strength which is a parameter negotiated between client and server. Its limits are generally imposed by the server implementations.

Figure 6.1: IT Integration Technology



gateway at the boundary of the domain exports controlled access to the application and associated data to an outer security domain.

In the case where HTML and forms are being used for user interaction, a **web server adapter**[1] is used to translate web browsing and form filling actions to requests on the inner gateway. The adapter must enforce access control (e.g., by requiring protected user authentication and a secure user session between the browser and the server).

In the case where interaction is controlled by an applet downloaded into the client browser, the applet connects across the Internet to an outer gateway which maps requests from the applet to the functions exported at the inner gateway. The outer gateway must enforce access control (e.g., by supporting a **business protocol** which includes user authentication).

To protect the user, the **virtual machine** in the client browser must be “safe” in the sense that it prevents the applet from damaging the browser or its environment[2].

The inner gateway ensures that the back office application can only be accessed by either the outer gateway or the web server adapter.

The outer gateway, web server adapter and inner gateway necessarily contain application specific and access control policy specific functionality. To maximise re-use and consistency across applications, they should be constructed using CORBA distributed object technology.[3] In addition to providing support for distribution of these components, standard CORBA

1. The adapter is linked to the Web Server's “Common Gateway Interface (CGI)” or equivalent (e.g., the Netscape “NSAPI” interface).

2. It is assumed the browser will permit controlled access to its environment by trusted applets. This aspect of down-loading applets is still subject to development, a solution based on digital signatures attached to applets is anticipated.

3. At the time of writing Java APIs for distributed object processing (Java remote method invocation - RMI) are being defined which are in the spirit of CORBA, but with simpler APIs and greater integration with the Java language. Java, Java RMI and other Java APIs for wrapping back office applications (e.g. JDBC for database access), my be substituted for CORBA as the E2S project develops.

services provide **wrapper** technology for a wide range of application interfaces (OLTP, remote SQL database, etc.).

The applet to outer gateway path provides a direct means for business-to-business interactions, whereas the Web server route is best suited to supporting user-to-business interaction.

Because of their role in access control both the web server adapter and the gateways require management interfaces for use by security administrators and require access to directories and certificate repositories. Consequently they must be subject to **security audit**.

6.4 Security audit

Security failures are more often attributed to errors in the management and deployment of security technology than in a failure of the technology itself. Additionally in any large organisation there is the risk of an attack by an “insider”. To make a system resilient against such threats security must strengthened by providing logging of security related events (dynamic auditing) and regular checking that physical security and access control policies are correctly implemented (static auditing).

Security audit tools include:

- technology for keeping secure logs of security related events
  - — such logs are critical components and need strong integrity protection
  - — E2S technology can be used to provide the security component of such protection when it is otherwise not available.
- tools for analysis of audit trails kept by critical infrastructure components (e.g., firewalls, key management infrastructure functions)
- tools for ensuring access controls are in place and known security flaws are fixed (e.g., by system probing, review of system configuration files etc.).

7 Viewpoint Analysis

7.1 Viewpoints

This chapter summarises the E2S architecture in terms of the viewpoints of the ISO Reference Model for Distributed Processing (ISO/IEC 10746-3, ITU Recs. X.903).

It consists of a structured list of the architectural concepts and rules defined in chapters 2 to 6 inclusive.

This analysis:

- enables alignment of the E2S architecture to other distributed processing architectures
- shows the separation of concerns in the E2S architecture between
  - roles, objectives and policies (the enterprise viewpoint)
  - information resources and processes (the information viewpoint)
  - functional elements and the interfaces between them (the computational viewpoint)
  - distribution infrastructure (the engineering viewpoint)
  - technology choices (the technology viewpoints).

Given the objectives for the E2S architecture set out on Chapter 1, the design of the architecture has deliberately set out to minimise constraints in the enterprise, engineering and technology viewpoints so as to permit the widest range of applications and implementation freedom, consistent with retaining a common technology framework.

7.2 Enterprise viewpoint

- person
  - group (of people)
- business
  - brand image
- application
- telecooperation
- interactive sessions
- purchasing
- payment

7.2.1 Secure electronic mail

- security enhanced mailer

- insecure mailer
- secure email gateway
- secure email gateway manager

7.2.2 Web browser

- browser
- WWW server
- page-oriented interface
- custom user interface

7.2.3 Key management infrastructure

- key generation
- key escrow
- key recovery
- key certification
- user registration
- key verification
- key revocation
- certification authority
  - internal
  - external
- registration authority
  - internal
  - external
- key management infrastructure security manager

7.2.4 Credentials management

- access control
  - role
  - capability
- credentials manager

7.2.5 Smartcard infrastructure

- smartcard
- smartcard issuer
- smartcard user

7.2.6 Business protocols

- contract

7.2.7 Payment infrastructure

- merchant
- card-holder
- “Visanet”
- issuing bank
- acquiring bank

7.2.8 Purchasing infrastructure

- merchant
- customer
- banking infrastructure

7.2.9 Firewalls

- computer
- Internet (as a “community”)
- security domain
- software
- security flaw
- virus
- trojan horse
- security policy

7.2.10 Security audit

- secure logs
- secure system configuration files

7.3 Information viewpoint

7.3.1 Person

- name
- identity

7.3.2 Secure electronic mail

- email distribution list

7.3.3 World Wide Web (WWW)

- HTML
- page
- form
- applet

7.3.4 Key management infrastructure

- public key, private key pair
- certified public key
- X.509 digital certificate

7.3.5 Smartcard infrastructure

- PIN

7.3.6 User authentication

- challenge / response

7.3.7 Business protocols

- PEM format message

7.3.8 Purchasing infrastructure

- order
- account

7.4 Computational viewpoint

7.4.1 Secure electronic mail

- insecure mailer
- security enhanced mailer
- secure email gateway
  - client interface
  - server interface
  - manager interface

7.4.2 World Wide Web

- Web browser
  - helper application
  - applet
  - virtual machine
- Web server

7.4.3 Key management infrastructure

- certification authority
- registration authority
- client services module
- certificate repository
- directory
- security module

7.4.4 Smartcard infrastructure

- smartcard
- smartcard reader/writer
- smartcard software library

7.4.5 Payment infrastructure

- user SET module
- merchant SET module
- “VISAnet”

7.4.6 Purchasing infrastructure

- user purchasing module
- merchant purchasing module
- purchasing coordinator

7.4.7 Firewalls

- filter
- gateway
- application proxy

7.4.8 IT integration

- “back office” application
- web server adapter
- outer gateway
- inner gateway
- wrapper

7.5 Engineering viewpoint

- Internet (as a network)
- physically secure location

7.5.1 Smartcard architecture

- physical encapsulation of keys and algorithms

7.5.2 Firewalls

- multi-level, compartmentalised operating system
- trusted network
- encapsulated component
- segmented network

7.6 Technology viewpoint

- Internet (as a set of standards defined by IETF etc)
- Eudora
- Privacy Enhanced Mail (PEM)
- Pretty Good Privacy (PGP)
- Netscape 2.0/3.0
- Secure Socket Layer protocol (SSL)
- Verisign
- Ice-tel
- X.500 Directory
- X.509 digital certificate
- SecuDE tool-kit
- Osisec tool-kit
- GEMPlus GPK-2000
- RSA
- DES
- SHA-0, SHA-1
- Secure Electronic Transactions (SET).

References

[C1]

Consolidated User Requirements, E2S Project Deliverable C1, Octacon Ltd., Gateshead, UK, 1996.

[CB 94]

Cheswick, W.R., and Bellovin, S.M., **Firewalls and Internet Security: Repelling the Wily Hacker**, Addison Wesley, Reading MA, USA, 1994.

[D3]

Security Models and Policies, E2S Project Deliverable D3, Gemplus, Gemenos, France, 1996.

[DES]

ANSI X3.92, "American National Standard for Data Encryption Alogorithm (DEA)," ANSI, 1981.

[EUDORA]

http://www.qualcomm.com/

[HTML]

http://www.w3.org/pub/WWW/MarkUp/Activity

[ICE-TEL]

http://www.darmstadt.gmd.de/ice-tel/

[ISO 10746-3]

ISO/IEC 10746-3:1996(E), ITU Rec. X.903, Information Technology - Open Distributed Processing - Reference Model: Architecture, ISO, Geneva, 1996.

[NETSCAPE]

http://www.netscape.com/

[OSISEC]

http://www.cs.ucl.ac.uk/research/ice-tel/osisec/

[PEM]

IETF Privacy Enhanced Mail Documents - Part I: Message Encryption and Authentication Procedures (RFC 1421); Part II: Certificate-Based Key Management (RFC 1422); Part III: Algorithms, Modes and Identifiers (RFC 1423); Part IV: Key Certification and Related Services (RFC 1424).

Available as http://ds.internic.net/rfc/rfc1421.txt etc.

[PGP]

The Official PGP User's Guide, MIT Press, Boston, USA, 1995.

[RSA]

R.L. Rivest, A Shamir, and L.M. Adleman, “A Method for Obtaining Digital Signatures and Public-Key Cryptosystems,” *Communications of the ACM*, **21**, 2, Feb. 1978, pp 120-126.

[SECUDE]

http://saturn.darmstadt.gmd.de/secude/secude.html

[SET]

http://www.visa.com/cgi-bin/vee/sf/set/intro.html?2+0

[SHA]

National Institute of Standards and Technology, NIST FIPS PUB 186, “Digital Signature Standard”, US Dept. of Commerce, May 1994.

[SSL]

http://www.netscape.com/newsref/std/SSL.html

[VERISIGN]

http://www.verisign.com/

[X.500]

ITU Recs X.501-510, The Directory. ITU, Geneva, 1995.

[X.509]

ITU Rec. X.509, The Directory - Authentication Framework, ITU, Geneva, 1989.