T/3522TL/2778/9

4 February 1997

CESG INFOSEC MEMORANDUM NO. 14

AN HMG PUBLIC KEY INFRASTRUCTURE TO SUPPORT CONFIDENTIALITY

Issue 1.0

CESG ELECTRONIC INFORMATION SYSTEMS SECURITY MEMORANDUM NO. 14

AN HMG PUBLIC KEY INFRASTRUCTURE TO SUPPORT CONFIDENTIALITY

Issue 1.0

February 1997

Communications-Electronics Security Group

FOREWORD

This Memorandum is issued by the Communications-Electronics Security Group (CESG) of Government Communications Headquarters as part of its responsibility to advise HMG on Electronic Information Systems Security (Infosec).

It suggests an architecture for a public key infrastructure (PKI) to support confidentiality between communicating systems. The Memorandum will eventually form part of a suite of documents which collectively provide advice on the implementation of a PKI, and the use of the services enabled by such an infrastructure (e.g. electronic mail). The architecture as described in this document is an initial attempt at defining a PKI, and CESG will take into account any comments on its feasibility.

This Memorandum is intended for use by HMG, its contractors and suppliers.

General correspondence in connection with this document, including requests for additional copies, should be addressed to:

Communications-Electronics Security Group (X13)
Government Communications Headquarters
PO Box 144
Cheltenham GL52 5UE
United Kingdom

Technical correspondence in connection with this document should be sent to T27 at the above address.

CONTENTS

FOREWORD ii
CONTENTS iii
REFERENCES iv
ABBREVIATIONS v
I. INTRODUCTION 1
A. Purpose and Scope 1
B. Background 2
II. CONFIDENTIALITY FRAMEWORK 4
A. Key Management 6
B. Key Updates 7
Annex A Key Management Scheme A-1
Annex B A Guide to Public Key Infrastructures B-1

REFERENCES

a. Jeffries, Mitchell, Walker. *A Proposed Architecture for Trusted Third Party Services*. International Conference at Brisbane Aus July 1995, published in Lecture Notes in Computer Science.

b. ITU-T Recommendation X.509 (1993) | ISO/IEC 959 4-8: 1993, Information Technology - Open Systems Interconnection - The Directory : Authentication Framework.

c. Technical Corrigendum 2 to X.509 (‘90 & ‘93) | ISO/IEC 9594-8 (‘90 & ‘93).

d. Draft Amendment 1 to ITU Rec. X.509 (1993) | ISO/IEC 9594-8 : 1993 .

ABBREVIATIONS

ASCII	American Standard Code for Information Interchange
CA	Certification Authority
CESG	Communications-Electronics Security Group
CMA	Certificate Management Authority
DSO	Departmental Security Officer
EDI	Electronic Data Interchange
EDIFACT	EDI for Administration, Commerce and Transport
HMG	Her Majesty's Government
HTTP	Hypertext Transfer Protocol
ITU	International Telecommunications Union
RFC	Request for Comment
SMTP	Simple Mail Transfer Protocol
TK	Token Key

I. INTRODUCTION

A. Purpose and Scope

1. Interdepartmental discussions are currently in progress to determine the general requirement for a Public Key Infrastructure (PKI) within HMG, and the associated issues. In the expectation that these discussions will in due course endorse a need for some level of PKI service, CESG has been developing generic technical recommendations for supporting these services.

2. This document describes CESG's recommended architecture for an HMG PKI to support confidentiality between communicating systems. The main objective of the recommendations is to facilitate pan-government secure communication services (e.g. electronic mail, file transfer, electronic trading and communication with the public). The framework is suitable for communications over both public bearers (e.g. X.400) and Internet technology (e.g. SMTP, HTTP). The objective is met by:

a. simplifying the implementation of such services within government,

b. ensuring secure communication between departments is possible,

c. facilitating future inter-operability with commercial users,

d. maximising the use of commercial technology in a controlled manner,

e. whilst allowing access to keys for data recovery or law enforcement purposes if required, in accordance with official release procedures.

3. The HMG PKI should be capable of interoperating with other PKIs (e.g. those of other governments and commercial PKIs).

4. A companion publication to this document, Draft CESG Infosec Memorandum No. 15, provides an architecture for a PKI to support authentication between communicating entities.

5. Although not a government standard as such, this document forms the current basis upon which government implementation trials in this area are moving forward. Depending upon the experience gained during these trials, it may well contribute to a future formal standard. Future CESG publications will define a data encapsulation protocol, protocols for key management within the PKI, and provide implementation guidance to product developers.

6. An architecture specifically for secure electronic mail is the subject of the CESG Architecture for Secure Messaging (CASM), which will be described in a separate series of publications. The PKI described in this Memorandum can be used to support CASM compliant mail systems.

7. Annex B of this document provides a tutorial on PKI and public key cryptography concepts.

B. Background

8. Over the past few years electronic mail has become the most popular medium for exchanging information within the IT industry. It is used by both people and computer processes to exchange a variety of 'message' formats, from simple ASCII to machine-readable business documents, eg. EDIFACT. Communication systems such as the Internet, which support electronic mail, allow users to send mail over vast distances, and to many different countries.

9. There are currently two main 'standards' for electronic mail in widespread use. The first is based upon the work of the International Telecommunications Union (ITU) and is documented in the X.400 set of recommendations. The second is based upon work carried out by a number of organisations associated with the Internet and documented in a series of 'Request For Comment' papers (RFCs). The standards define the various protocols required to transfer mail from one user to another. For example, the protocol used to transfer mail between mail servers or switches is defined in X.400 as the P1 protocol, and for the Internet as the Simple Mail Transfer Protocol.

10. Security (be this for confidentiality, authentication or other aspects) is not a major feature within current implementations of these standards. The Internet does not provide its users with any confidentiality or integrity services. There are a number of 'add-on' products which provide these services, but none of them are approved for the protection of HMG protectively marked material. The X.400 recommendations do define a set of security services as part of the protocols, however these are specific to X.400 and few implementations are commercially available. The mail systems and the information they carry are therefore susceptible to component failure, user misuse, and malicious attack, which may result in:

a. the loss or disclosure of information,

b. the modification of information,

c. the impersonation of legitimate users,

d. the denial of message generation or receipt,

e. the denial of system services.

11. The popularity of electronic mail within government has meant that it has now been adopted for transferring official material within and between departments, and between government and commerce and other outside bodies. As much of this use became established before the new protective marking scheme was introduced in 1994, we have a de facto situation in which a significant proportion of this information now deserves a protective marking. Even where a protective marking is not formally warranted, users seek certain assurances from their electronic mail systems. These assurances can be provided to the user in the form of a number of user to user security services, such as:

a. data confidentiality,

b. data integrity,

c. proof of origin,

d. proof of delivery/receipt,

e. non-repudiation of origin,

f. non-repudiation of receipt

12. In addition to electronic mail, users will wish to make use of other communication services allowing them to download documents, engage in electronic trading, or access directories and databases for example. Confidentiality is important in these contexts as well, in order to assure the user that his data is not subject to unauthorised disclosure.

13. The security services are implemented in the form of an encapsulation protocol which conveys security information (eg. protective markings, digital signatures, user identities) between users. The services provided by the protocol are based upon public key cryptographic techniques. These techniques require the distribution of key material through trusted channels, and the ability to authenticate material distributed through untrusted channels. This document defines the confidentiality framework recommended to support the confidentiality of data objects within and communicated between computer systems.

14. CESG's policy for the development of products to provide these services is to encourage industry to incorporate CESG approved algorithms into their existing products. This has been done successfully in the past with CESG's RAMBUTAN chip. However it is now recognised that one of the major requirements for communications products is that they should inter-operate with similar products developed by other manufacturers. In order to achieve this, it is no longer practical to provide only the algorithms to industry, and further guidance is now required. The objective of these recommendations is therefore to provide this guidance, and to encourage manufacturers to develop products which will be able to inter-operate with similar products from other sources. This should enable a single architecture for secure communication to be implemented within HMG which maximises the use of commercial products, whilst minimising the infrastructure required to support them.

II. CONFIDENTIALITY FRAMEWORK

15. The confidentiality framework is based upon a proposal by the Royal Holloway College (Reference a) for trusted third party services, using the well known Diffie-Hellman public key protocol. The confidentiality framework relies upon a mechanism to ensure the authenticity of key material. One way to ensure authenticity is to use the authentication framework defined in Memorandum No. 15.

16. The framework has been developed to work with the electronic mail architecture currently evolving within government. The architecture consists of independent domains, corresponding approximately to individual departments. In many cases the domains may be further divided into smaller domains or systems which may be scattered over a wide geographical area. Most domains have a local directory or similar repository for shared information, however this paper does not assume that these are connected. The management of systems within a domain is usually the responsibility of the system administrator. The administrator's role includes the management of the system components, registration of new users, the allocation of user names and addresses, and the management of users' entries within a directory - if one exists.

17. The confidentiality framework is implemented within a domain by a Certificate Management Authority (CMA, analogous to the trusted third party defined in Reference a). The Certificate Management Authority is subordinate to the Departmental Security Officer (DSO), and responsible to the department's crypto custodian. It will be responsible for registering users of the secure communications system (this may be in addition to the registration of users with the system administrator), allocating system privileges, ensuring users have unique distinguished names (but not necessarily allocating them), distributing key material, accounting for key material, and generating key material. As the Certificate Management Authority is responsible for generating the confidentiality keys, it should also take on the role of a certification authority (eg. a Certification Authority or Policy Certification Authority) in order to authenticate them. CESG will initially support the framework through the DSO's and crypto custodians, by generating and signing all the keys used. However in time it is envisaged that this role can be devolved down to those departments who wish to establish their own Certificate Management Authority.

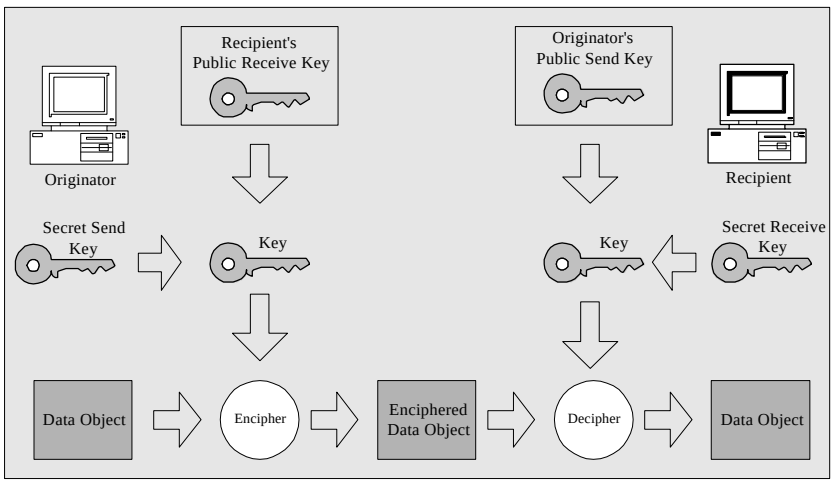

Figure 1 - Key Generation Process

18. An entity (person or process) within the framework is allocated a secret send key and a public receive key as the basis for enciphering data, and a secret receive key and a public send key as the basis for deciphering data. This implementation differs from the more conventional implementations of the Diffie-Hellman protocol which allocates a single pair of keys. A data object is enciphered with a key generated from the originator's secret send key and the recipient's public receive key, and is deciphered with the same key but generated from the originator's public send key and recipient's secret receive key. For the scheme to operate successfully all keys used in the process must be based upon the same parameters (modulus and base value). For efficiency the generated key is not used to encipher the data object directly. Instead the data object is enciphered with a data key which is placed inside a 'token'. It is the token which is enciphered with the Diffie-Hellman derived key or Token Key (TK). The data object is only enciphered once, whereas the data key may be enciphered a number of times depending on the number of recipients.

<table>
<tr><th colspan="10">Certificate Data</th><th colspan="2">Signature Structure</th></tr>
<tr><th rowspan="2">Version</th><th rowspan="2">Serial Number</th><th rowspan="2">Algorithm Identifier</th><th rowspan="2">Issuer</th><th colspan="2">Validity</th><th rowspan="2">Subject</th><th colspan="2">Subject Public Key Information</th><th rowspan="2">Authority Key Identifier (Extension)</th><th rowspan="2">Key Attributes (Extension)</th><th rowspan="2">Algorithm Identifier</th><th rowspan="2">Digital Signature</th></tr>
<tr><th>Valid From</th><th>Valid To</th><th>Public Send or Receive Key</th><th>Algorithm Identifier</th></tr>
</table>

Figure 2 - Confidentiality Certificate

19. The public confidentiality keys are held within confidentiality certificates. The certificates are encoded structures based upon version 3 of the X.509 certificate and selected extensions (References b, c and d). The public send and receive keys are held in Send Certificates and Receive Certificates respectively. The seed keys (see below) are also held within encoded structures, known as secret key tokens. The keys within the tokens may be enciphered with a user code which is distributed to the user separately from the token.

20. Send and receive keys are deterministically generated from top level interoperability keys (IKs) (Figure 3). The Certificate Management Authority first generates a seed key from a function of the user's distinguished name and a top level interoperability key. The secret send and receive keys are generated from a function of this seed key and a datestamp[1]. Public keys can then be generated from the secret keys and the domain parameters. The public receive key is stored with the datestamp in a Receive Certificate, the public send key is stored with the datestamp and the domain's parameters in a Send Certificate, and the seed keys distributed in a secure manner to the user within a Secret Key Token. Receive Certificates are always appended to the enciphered token, to allow the recipient to identify the appropriate seed key required to regenerate the secret receive key. The public confidentiality keys and seed keys are linked by a seed key identifier.

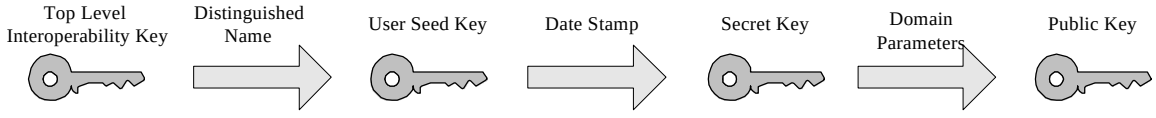

Figure 3 - Generation of Secret and Public Keys

[1] Note that secret send keys may also be randomly generated by the sender.

21. Tokens are enciphered within a domain with receive keys derived from the domain's internal top level interoperability key. Tokens are enciphered between domains with receive keys derived from an interoperability key which has been bilaterally established between the Certificate Management Authorities of the communicating domains. It should be noted that send keys are always derived from the domain's internal interoperability key.

22. Once an interoperability key has been established between two domains, an authority can deterministically generate the public receive keys for members of the other's domain using the distinguished name of that member. The generation (unilateral and bilateral) of these keys and the mechanism by which authorities communicate is beyond the scope of this document. However a detailed description of the generation process is the subject of forthcoming CESG publications.

A. Key Management

23. The following paragraphs explain the key management scheme in outline. Annex A describes the mathematical basis of the scheme in more detail.

Intra-Domain Confidentiality

24. If User A needs to exchange enciphered data objects with User B in the same domain X, they are each allocated the following by Certificate Management Authority X:

a. A secret key token containing the user's seed key, from which secret send and receive keys for Domain X can be generated, and distributed to the user through a secure means.

b. A send certificate containing A's public send key. This certificate may be distributed to users or posted in the domain's directory.

c. A receive certificate containing B's public receive key. This is only posted in the directory.

25. The initial distribution of secret key tokens is outside the scope of this paper, however subsequent distributions of derived keys may be performed using the confidentiality framework.

26. In order for User A to send an enciphered data object to User B, User A enciphers the data object with a data key. User A then generates a token key for the intended recipient, and uses this to encipher the data key. The token key is derived from User A's seed key, User B's public receive key, and some cryptographic parameters contained within A's send certificate.

27. The certificates containing the keys and parameters used to generate the token key are appended to the enciphered data object and token(s) to simplify the recipient's processing.

28. On receipt of the enciphered data object User B extracts their receive certificate. The seed key identifier contained within the certificate indicates which seed key is required to regenerate the token key. The token key, derived from the seed key, A's public send key, and some cryptographic parameters from both send and receive certificates, is used to decipher the data key which was used to encipher the data object.

Inter-Domain Confidentiality

29. If User A needs to exchange enciphered data objects with User C in a different domain Y, the receive certificate of User C will not be available - unless User C has already received enciphered data objects from a user within the first domain. In this case User A passes User C's distinguished name to Certificate Management Authority X which deterministically generates a public receive key for User C, by the following processing.

a. An interoperability key is established (if this has not already been done) between Certificate Management Authority X and Certificate Management Authority Y.

b. User C's distinguished name, a seed key identifier and the top level interoperability key are then used to generate a seed key for User C.

c. A public receive key for User C can now be derived from the seed key.

d. The public receive key and datestamp are incorporated into a receive certificate and passed directly to User A or stored within the local directory for User A and other users to access (or both).

30. User A now enciphers the data object with a data key, which is then itself enciphered with a token key for each of the intended recipient. The token key is derived from User A's seed key, User C's public receive key, and some cryptographic parameters from A's send certificate.

31. The certificates containing the keys used to generate the token key are appended to the enciphered data object and token(s) to enable the recipient's processing.

32. On receipt of the enciphered data object User C extracts their receive certificate. The seed key identifier contained within the certificate indicates which seed is required to regenerate the token key. In this case User C is unlikely to hold the associated seed key, unless User C has previously received enciphered data objects from a user within the Domain X. Therefore User C passes their receive certificate and User A's send certificate to Certificate Management Authority Y. The authority deterministically generates a new seed key for User C, and passes it to User C by some secure means.

33. User C regenerates the token key and uses this to decipher the data key, which in turn is used to decipher the data object. The token key is derived from the seed key, A's public send key, and some cryptographic parameters from both the send and receive certificates.

B. Key Updates

34. There is a requirement to periodically update the various keys used within the system. This will be carried out at different intervals depending upon the type of key in question. The interoperability keys and seed keys will have a similar period, whereas the secret keys and associated public receive key will be considerably shorter. The short lifespan of the keys used does not impose a burden on key distribution because the public certificates do not need to be distributed through secure channels. New secret send keys can be computed by their owners using the datestamp contained within the current send certificate, new secret receive keys can be computed by their owners using the datestamp contained within the receive certificate appended to the enciphered data object and tokens. Both certificates can be posted in a

directory or similar and updated on a regular basis by the local Certificate Management Authority.

35. When an interoperability key is updated a user will no longer have an appropriate seed key. In this case the user (or user's workstation) must pass the receive certificate to their local Certificate Management Authority. The authority can then generate and distribute a new seed key, via a secure channel to the user. This is effectively the same process which is carried out when a user receives an enciphered data object from an unknown domain.

Annex A Key Management Scheme

Intra-Domain Confidentiality

36. If User A (IDA) needs to exchange enciphered data objects with User B (IDB) in the same domain (Domain X), they are each allocated the following by Certificate Management Authority X.

a A secret key token containing the user's seed key $K_{XX}(ID)$ from which secret send and receive keys for Domain X can be generated.[2] This token is distributed to the user through some secure means. The seed key is generated from the following function, where IK_{XX} is the top level interoperability key for Domain X, KID is the seed key identifier, and f1 is some operation such as a hash function.

$$K_{XX}(ID) = f1(IK_{XX}, ID, KID)$$

b A send certificate containing the public send key Spub(ID), datestamp (T_S), and the domain's base (g_X) and modulus (N_X) - written Cert(ID, Spub(ID), T_S, g_X, N_X). This may be distributed to users or posted in the domain's directory. The secret and public send keys are related by the following function, where f2 need not equal f1.

$$Spub(ID) = g_X^{f2(K_{xx}(ID),\ T_S)} \bmod N_X.$$

c A receive certificate containing the public receive key Rpub(ID) and datestamp (T_R) - written Cert(ID, Rpub(ID), T_R). This is only posted in the directory. The secret and public receive keys are related by the following function, where f2 need not equal f1.

$$Rpub(ID) = g_X^{f2(K_{xx}(ID),\ T_R)} \bmod N_X.$$

37. The initial distribution of secret key tokens is outside the scope of this paper, however subsequent distributions may use the electronic mail system and the confidentiality service provided by the security protocol.

38. In order for User A to send an enciphered data object to User B, User A enciphers the data object with a data key. User A then generates a token key (TK) for the intended recipient, and uses this to encipher the data key. The token key is derived from User A's seed key $K_{XX}(A)$, the public receive key from Cert(IDB, Rpub(B), T_R), and the modulus and datestamp from Cert(IDA, Spub(IDA), T_S, g_X, N_X) using the following function.

$$TK = Rpub(B)^{f2(K_{xx}(A),\ T_S)} \bmod N_X$$

[2] Note that an alternative approach is for the secret send key to be generated randomly by the sender .

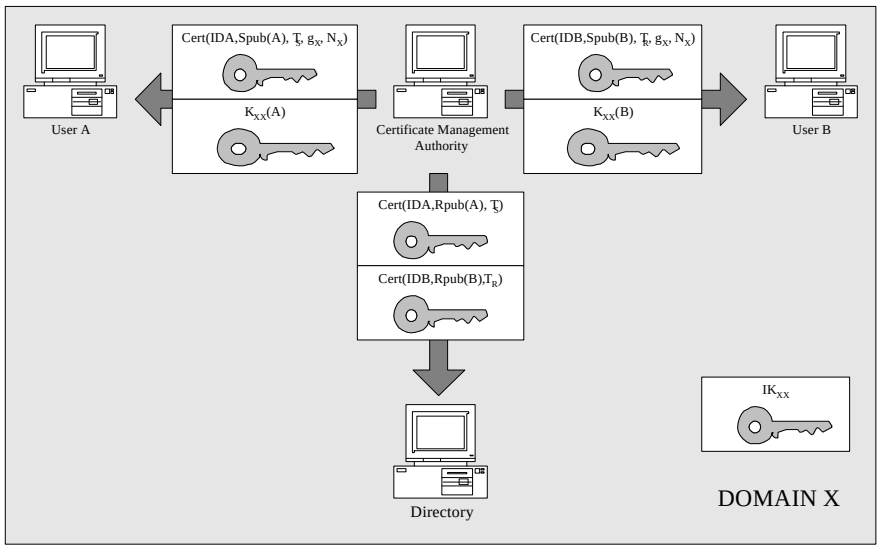

Figure 4 - Example Initial Key Distribution

39. The certificates containing the keys and parameters used to generate the token key, Cert(IDA, Spub(A), T_S, g_X, N_X) and Cert(IDB, Rpub(B), T_R), are appended to the enciphered data object and token(s) to simplify the recipient's processing.

40. On receipt of the enciphered data object User B extracts their receive certificate. The seed key identifier contained within the certificate indicates which seed key $K_{XX}(B)$ is required to regenerate the token key. The token key is used to decipher the data key which was used to encipher the data object. The token key is derived from the seed key, the public send key and modulus from Cert(IDA, Spub(A), T_S, g_X, N_X), and the datestamp from Cert(IDB, Rpub(B), T_R), using the following function.

$$TK = Spub(A)^{f2(K_{XX}(B),\ T_R)} \bmod N_X$$

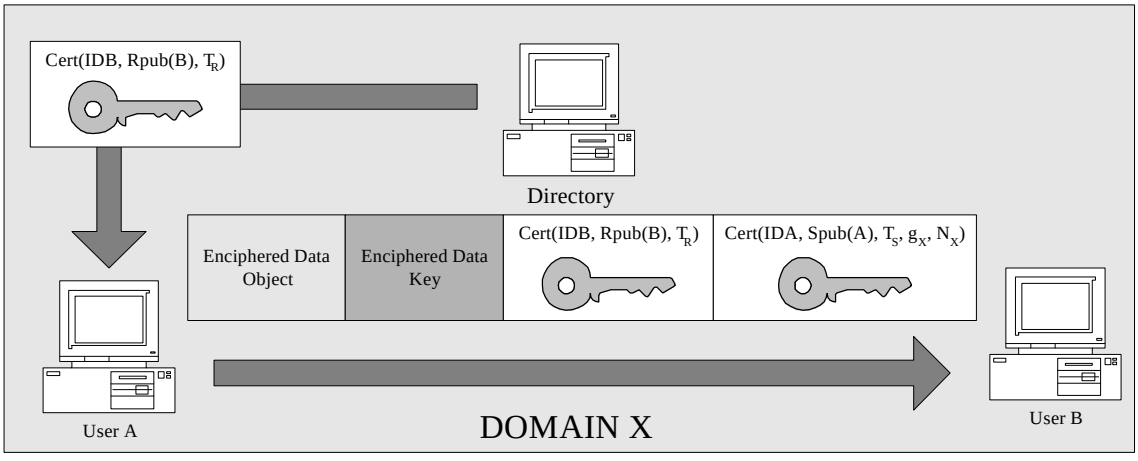

Figure 5 - Intra-Domain Communications

Inter-Domain Confidentiality

41. If User A needs to exchange enciphered data objects with User C (IDC) in a different domain (Domain Y), the receive certificate of User C will not be available - unless User C has already received enciphered data objects from a user within the first domain. In this case

User A passes User C's distinguished name to Certificate Management Authority X which deterministically generates a public receive key for User C, by the following processing.

a. An interoperability key (IK_{XY}) is established (if this has not already been done) between Certificate Management Authority X and Certificate Management Authority Y.

b. User C's distinguished name, the seed identifier KID, and the top level interoperability key are then used to generate a seed key for User C. The seed key is generated from the following function.

$$K_{XY}(C) = f1(IK_{XY}, IDC, KID)$$

c. A public receive key for User C can now be derived from the seed key, a datestamp (T_R), and Domain X's modulus and base value, using the following function.

$$Rpub(C) = g_X^{f2(K_{XY}(C),\ T_R)} \bmod N_X.$$

d The public receive key and datestamp are incorporated into a receive certificate - written Cert(IDC, Rpub(C), T_R), and passed directly to User A or stored within the local directory for User A and other users to access (or both).

42. User A now enciphers the data object with a data key, which is then itself enciphered with a token key for each of the intended recipient. The token key is derived from User A's seed key K_{XX}, the public receive key from Cert(IDC, Rpub(C), T_R), and the modulus and datestamp from Cert(IDA, Spub(A), T_S, g_X, N_X), using the following function.

$$TK = Rpub(C)^{f2(K_{XX},\ T_S)} \bmod N_X$$

43. The certificates containing the keys used to generate the token key, Cert(IDA, Spub(A), T_S, g_X, N_X) and Cert(IDC, Rpub(C), T_R), are appended to the enciphered data object and token(s) to enable the recipient's processing.

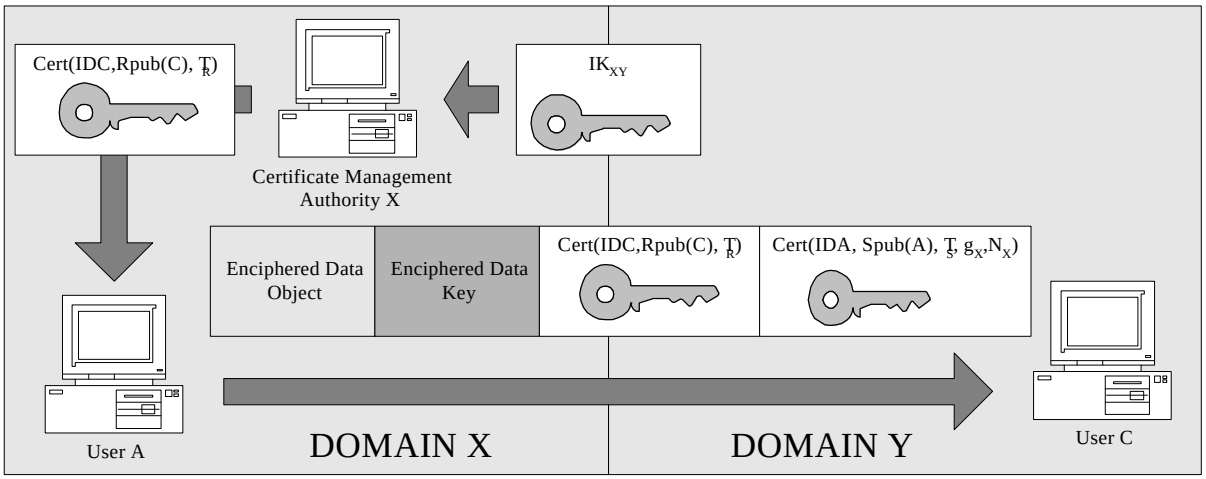

Figure 6 - Domain X Processing

44. On receipt of the enciphered data object User C extracts its receive certificate. The seed key identifier contained within the certificate indicates which seed key $K_{XY}(C)$ is required to regenerate the token key. In this case User C is unlikely to hold the associated seed key, unless User C has previously received enciphered data objects from a user within the Domain X. Therefore User C passes Cert(IDC, Rpub(C), T_R) and Cert(IDA, Spub(A), T_S, g_X, N_X) to Certificate Management Authority Y. The authority deterministically generates a new seed key for User C, by the following processing:

a. Certificate Management Authority Y identifies the appropriate interoperability key (K_{XY}) from the distinguished name (Certificate Management Authority X) of the issuer held within User A's send certificate.

b. User C's distinguished name, the seed key identifier KID, and the top level interoperability key are then used to generate the new seed key using the following function.

$$K_{XY}(C) = f1(IK_{XY}, IDC, KID)$$

c. The seed key is then passed through a secure channel to User C.

45. User C regenerates the token key and uses this to decipher the data key, which in turn is used to decipher the data object. The token key is derived from the seed key, the public send key and modulus from Cert(IDA, Spub(A), T_S, g_X, N_X), and the datestamp from Cert(IDC, Rpub(C), T_R), using the following function.

$$TK = Spub(A)^{f2(K_{XY}(C),\ T_R)} \bmod N_X$$

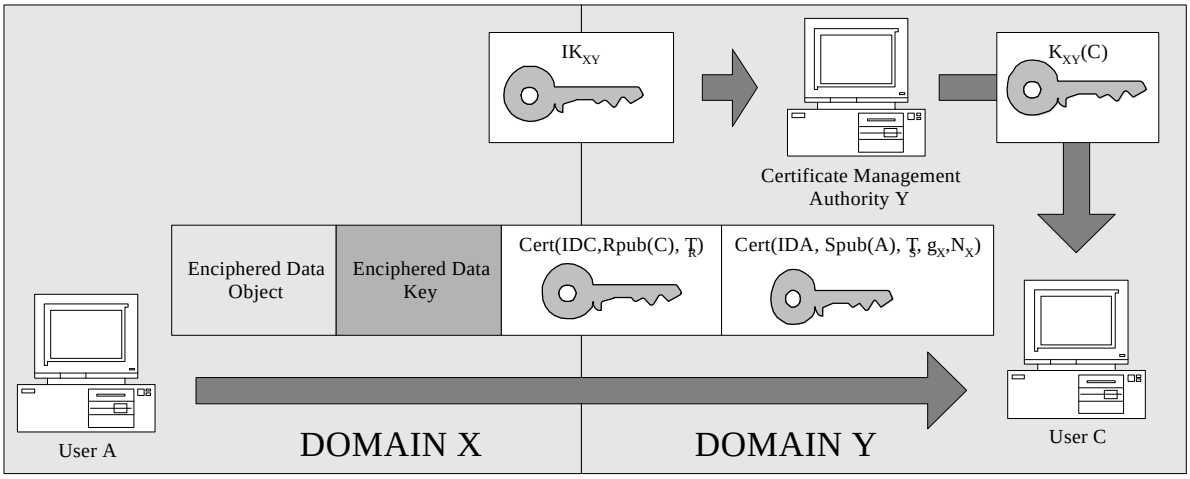

Figure 7 - Domain Y Processing

Annex B A Guide to Public Key Infrastructures

Introduction

46. The following paragraphs provide a brief outline of the basic concepts of Public Key Cryptography and the rationale for a Public Key Infrastructure to support the maintenance and distribution of key material, certificates etc.

Background

47. Traditional cryptography is based upon the use of *symmetric* keys and is characterised by the fact that the same keys are used for both encryption and decryption. Essentially symmetric cryptography provides security services at the link level. Writer-to-reader security services such as authentication and non-repudiation require the use of additional off-line techniques.

48. However the proliferation of communications and networked information systems has resulted in an increasing demand for 'off-line' writer-to-reader services, in particular authentication and non-repudiation, to be provided 'on line'. In addition, there is a requirement for content integrity between originator and recipient. Furthermore, the provision of security services is no longer confined to the network domain and certain functionality is moving to the desktop in the user domain. The additional security functionality required by users can be achieved by the use of *asymmetric* or public key cryptography.

49. Public key cryptography (PKC) has been known as a concept for about twenty years. Stripped of the mathematics, which are complex, it is in essence a means for parties to communicate securely in electronic form *without prior exchange of dedicated secret key material*. This is the basic difference from traditional secret key cryptography. PKC takes a fair amount of computing power to run. This is why it is has only relatively recently started to become a practical proposition for non-specialists. PKC techniques lend themselves to not just the confidentiality aspects of information security, but to other services of increasing relevance to business, both HMG and outside. These services include integrity, authentication and non-repudiation.

50. PKC is characterized by the use of two different but mathematically related keys to perform the cryptographic functions. Each person/entity has a private and public key. The public key is available to the user community in general and is usually posted in an X.500 directory. The private key is known only to the person/entity to whom it is issued. Confidentiality, integrity and authentication (the latter two usually realized in the form of a digital signature) can be achieved by different manipulations of public/private key pairs. Other security functions are supported by additional cryptographic processes. It should be noted that the use of asymmetric cryptography does not exclude the use of symmetric cryptography to support certain functions within the overall scheme. Furthermore, it is normal practice to use one public/private key pair to perform the digital signature, and a different key pair (usually of longer key length) to provide confidentiality through the encryption of the one-time symmetric key. See below for a brief 'tour' of asymmetric cryptography and corresponding security services.

51. An essential feature of PKC is that it must be possible to prove the authenticity of a public key. To achieve this the public key element of those security services derived from the public/private key pair must be 'bound' to the identity of the user in a secure manner. This

binding is achieved by means of a public key certificate using the X.509 format. The management of the certificates, and the provision of user access to both the certificates and the public keys may be carried out using a *Public Key Infrastructure* (PKI). A PKI will enable end-users to obtain authentic key material for those individuals and system components with which they need to exchange data. A PKI will thus support confidentiality, authentication, integrity and non-repudiation services between end-users, between entities, or a combination of both. The PKI entities which allow for the provision of this material are referred to in CESG Infosec Memorandum No. 15 as *Certification Authorities* (CAs).

52. A PKI can also be used to support an extension to the confidentiality service by the creation and distribution of private keying material The PKI entities which provide this service are referred to by CESG Infosec Memorandum No. 14 as *Certificate Management Authorities* (CMAs), but are sometimes denoted in other publications as *Trusted Third Parties* (TTPs). A CMA can also provide an escrow service whereby private keys can be stored and subsequently released either to the registered owner to achieve data recovery should a key be lost, or to intelligence or law enforcement agencies under due process of law. It is certain that a large number of CMAs will exist in the Global Information Society representing various groupings of end users. Hence in order to facilitate the transaction processing across user groups it will necessary to harmonize the workings of CMAs.

53. PKC techniques appear certain to become an economical and favoured means of protecting HMG electronic data transactions, but their potential can only be fully realised as shown above by taking a pan-government approach which establishes a Public Key Infrastructure. This would have applications in e-mail, electronic data interchange (EDI) with business and finance, and formal messaging, all services which are being increasingly used by government departments. The existence of a PKI would also facilitate the interconnection of UK IT networks to other government and commercial networks both national and international.

54. The cost benefit associated with the use of e-mail and EDI for contracting is one factor forcing the pace as government departments strive to reduce their costs. In this electronic environment it is important to replicate the safeguards that have traditionally been applied to paper-based systems to provide adequate confidentiality, integrity and authentication services. Electronic systems will often use cryptographic techniques to provide equivalent services.

An overview of asymmetric cryptography

55. The following paragraphs define the major security services required for Asymmetric Cryptography.

Hashing

> The use of an algorithm to create a unique fingerprint for each document. A small change in the source document changes a large number of bits in hashing code (fingerprint).

Digital Signature

> This requires a public/private key pair. The signature is derived by the sender applying his private key to the data (or a hash of the data). This signature may then be verified by any recipient using sender's public key.

Encryption

This requires public/private key pair. Encryption of one-time symmetric confidentiality key is performed by sender using intended recipient's public key. Decryption of symmetric key performed by recipient using his private key.

56. The following paragraphs describe the stages involved in a secure messaging system using asymmetric public key cryptography.

Definition of Private/Public Key Pairs.

-Bob's key pair for digital signature - KEY SET ALPHA
-Bob's key pair for confidentiality - KEY SET BRAVO

-Alice's key pair for digital signature - KEY SET CHARLIE
-Alice's key pair for confidentiality - KEY SET DELTA

Scenario 1 - Bob Sends a Message to Alice

Bob

- Generates hashing code for document using hashing algorithm.
- Signs hashing code to provide digital signature using ALPHA private key.
- Appends hash code to document.
- Generates a one-time, symmetric key.
- Encrypts document using symmetric key.
- Encrypts the symmetric key using Alice's DELTA public key.
- Transmits message to Alice.

Alice

- Decrypts the one time symmetric key using DELTA private key.
- Decrypts document using one-time symmetric key.
- Strips off the signature from the hash code using Bob's ALPHA public key.
- Regenerates the document fingerprint using the hashing algorithm and compares with the received hash code.

-- If OK message is authentic and is the original.

Scenario 2 - Alice Sends a Message to Bob

Alice

- Generates hashing code for document using hashing algorithm.
- Signs hashing code to provide digital signature using CHARLIE private key.
- Appends hash code to document.
- Generates a one-time, symmetric key.
- Encrypts document using symmetric key.
- Encrypts the symmetric key using Bob's BRAVO public key.
- Transmits message to Bob.

Bob

- Decrypts the one time symmetric key using BRAVO private key.
- Decrypts document using one-time symmetric key.
- Strips off the signature from the hash code using Alice's CHARLIE public key.
- Regenerates the document fingerprint using the hashing algorithm and compares with the received hash code.

-- If OK message is authentic and is the original.