A New Approach of Secret Key Management Lifecycle for Military Applications

NIKOLAOS BARDIS
Adjunct Professor
[1]Hellenic Army Academy,
[2]Hellenic Naval Academy,
[3]Hellenic Air Force Academy
Department of Computer Sciences
[1]Vari - 16673, [2]Terma Hadjikyriakou Avenue, Piraeus - 18539, [3]Dekelia Air Base, Tatoi, Metamorfosi 144 51, Greece
bardis@ieee.org

NIKOLAOS DOUKAS
Adjunct Professor
[1]Hellenic Army Academy
[2]Hellenic Air Force Academy
Department of Computer Sciences
[1]Vari - 16673, [2]Dekelia Air Base, Tatoi, Metamorfosi 144 51, Greece
nd@aeub.gr

KONSTANTINOS NTAIKOS
Second Lieutenant, Air Defence Officer
Hellenic Air Force Academy
Department of Computer Sciences
Dekelia Air Base, Tatoi, Metamorfosi 144 51, Greece
daikoskon@hotmail.com

Abstract: - In this paper a new approach is presented for key management access and sharing secret keys between certified users of a group. Such schemes are referred to as Symmetric Key Management Systems. The concept of information lifecycle management is first presented and analysed in the context of data storage efficiency. This concept is then extended for use with the management of symmetric secret keys. The need for a standard in symmetric secret key management is presented and founded on software engineering principles. A novel scheme contributing in this direction is hence presented. Specifically, access controls processes are presented that are based on passwords. These passwords, with the additional use of the AES cryptographic algorithm and nonces can be used to provide not only authentication for the access control in the system but additionally for the access in the encrypted file that stores all the symmetrical secret keys of each user of certified group. Following this, a new approach for the lifecycle management of secret keys is presented in order to achieve the secure communication based on encryption - decryption of all the messages in real time with the simultaneous use of two symmetrical secret keys for each transmission of information between the users. It is finally concluded that this innovative technology guarantees the automatic password and secret keys management lifecycle irrespective of the actions of the users and provides secure communication between certified group of users in local network and in internet.

Key-Words: - key management lifecycle, key management system, access control, symmetrical secret key

1 Introduction

One of the most important characteristics of military or commercial organizations in nowadays is the possibility of storing and exchanging information securely between a certified team of users [2]. The security of information is a henceforth fundamental objective of military forces and its guarantee is a strengthening factor. Thus there is a continuous development of information systems that aim to provide secure data storage and communication between groups of users. The growth and distribution of the Internet have rendered it one of the most important means for communication not only for large scale organizations (telecommunication companies, banks, ministries and military) but also for simple users [3], [4], [5], [6], [7]. The aim of this work is the development of a management system for passwords and shared symmetrical secret keys used in communication between a certified users group.

Cryptographic key management encompasses the entire lifecycle of cryptographic keys and other keying material. Basic key management guidance is provided in [1].

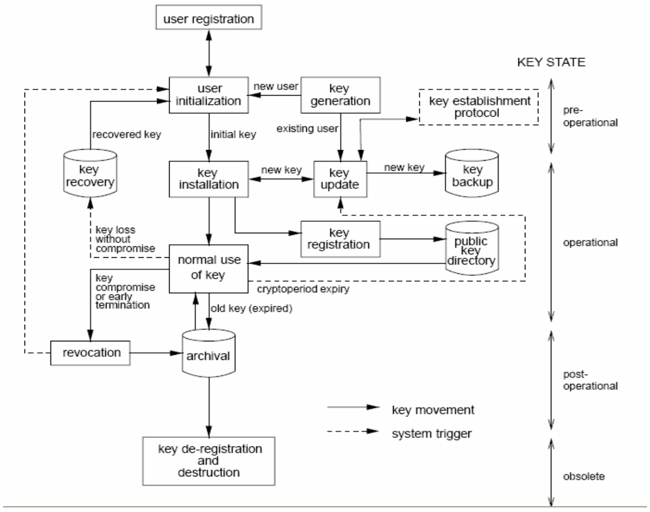
A single item of keying material (e.g., a key) has several states during its life, though some of these states may, in fact, be very short [3], [4]:

- Pre-operational: The keying material is not yet available for normal cryptographic operations.
- Operational: The keying material is available and in normal use.
- Post-operational: The keying material is no longer in normal use, but access to the material is possible.
- Obsolete/destroyed: The keying material is no longer available. All records of its existence may have been deleted.

The next viewgraph identifies the subsections that discuss various stages of key management for a given entity.

(i) User Registration, (ii) System and User Initialization, (iii) Keying Material Installation, (iv) Key Establishment, (v) Key Registration, (vi) Operational Use, (vii) Storage of Keying Material, (viii) Key Update, (ix) Key Recovery, (x) Key De-registration and Destruction, (xi) Key Revocation.

In the next Scheme 1 is shown the structure diagram of the key management life cycle



Scheme 1: Key management life cycle

<u>1. User registration</u> – an entity becomes an authorized member of a security domain. This involves acquisition, or creation and exchange, of initial key ingmaterial such as shared passwords or PINs by a secure, one-time technique (e.g., personal exchange, registered mail, trusted courier).
<u>2. User initialization</u> – an entity initializes its cryptographic application (e.g., installs and initializes software or hardware), involving use or installation (see below) of initial keying material obtained during user registration.
<u>3. Key generation</u> – generation of cryptographic keys should include measures to ensure appropriate properties for the intended application or algorithm and randomness in the sense of being predictable (to adversaries) with negligible probability (see Chapter 5). An entity may generate its own keys, or acquire keys from a trusted system component.
<u>4. Key installation</u> – keying material is installed for operational use within an entity's software or hardware, by a variety of techniques including one or more of the following: manual entry of a password or PIN, transfer of a disk, read-only-memory device, chipcard or other hardware token or device (e.g., key-loader). The initial keying material may serve to establish a secure on-line session throughwhich working keys are established. During subsequent updates, new keying material is installed to replace that in use, ideally through a secure on-line update technique.
<u>5. Key registration</u> – in association with key installation, keying material may be officially recorded (by a registration authority) as associated with a unique name which distinguishes an entity. For public keys, public-key certificates may be created by a certification authority (which serves as guarantor of this association), and made available to others through a public directory or other means (see x13.4).
<u>6. Normal use</u> – the objective of the life cycle is to facilitate operational availability of keying material for standard cryptographic purposes (cf. x13.5 regarding control of keys during usage). Under normal circumstances, this state continues until cryptoperiod expiry; it may also be subdivided – e.g., for encryption public-key pairs, a point may exist at which the public key is no longer deemed valid for encryption, but the private key remains in (normal) use for decryption.
<u>7. Key backup</u> – backup of keying material in independent, secure storage media provides a data source for key recovery (point 11 below). Backup refers to short-term storage during operational use.
<u>8. Key update</u> – prior to cryptoperiod expiry, operational keying material is replaced by new material. This may involve some combination of key generation, key derivation (x13.5.2), execution of two-party key establishment protocols (Chapter 12), or communications with a trusted third party. For public keys, update and registration of new keys typically involves secure communications protocols with certification authorities.
<u>9. Archival</u> – keying material no longer in normal use may be archived to provide a source for key retrieval under special circumstances (e.g., settling disputes involving repudiation). Archival refers to off-line long-term storage of post-operational keys.
<u>10. Key de-registration and destruction</u> – once there are no further requirements for the value of a key or maintaining its association with an entity, the key is de-registered (removed from all official records of existing keys), and all copies of the key are destroyed. In the case of secret keys, all traces are securely erased.
<u>11. Key recovery</u> – if keying material is lost in a manner free of compromise (e.g., due to equipment failure or forgotten passwords), it may be possible to restore the material from a secure backup copy.

12. Key revocation – it may be necessary to remove keys from operational use prior to their originally scheduled expiry, for reasons including key compromise. For public keys distributed by certificates, this involves revoking certificates. Of the above stages, all are regularly scheduled, except key recovery and key revocation which arise under special situations.

2 Information Lifecycle Management

Information Lifecycle Management (ILM) is an innovative concept that has been proposed in the context of improving the efficiency of data backup systems. ILM integrates all procedures included in the comprehensive data storage management program of an organisation. An alternative term for ILM is Data Life Cycle Management (DLM). The aim of ILM is to estimate the value of information over time, deriving measures of the speed at which it needs to be available, the cost that an enterprise should be prepared to face in order to maintain and be able to recover this data and decide on the point in time where the data is no longer necessary and may be deleted. A comprehensive overview of ILM principles can be found for example in [11]. Its most important principles that are necessary for understanding the concept of the secret key lifecycle and developing the proposed scheme for the management of secret keys will be given here.

Modern enterprises are heavily based on information processing and therefore on secure and reliable data storage. Management of the data storage is therefore one of the principle preoccupations of information system managers, who have to ensure that valid and up-to-date data is readily available and easy to recover. Additionally, many organisations, such as those belonging to the fields of the military, financial institutions and healthcare are faced with legal requirements for periodical data retention. All the above considerations imply that the cost of managing the data of a large organisation may rise by up to 30% annually [11]. Such estimates clearly highlight the value of efficient data storage management.

The role of ILM in this context is an automated data archiving process that rearranges the data within the available physical and logical storage locations so as to improve efficiency. This rearrangement is based on predetermined policies about accessibility, security and long-term storage. All the above policies are implemented without any manual intervention and hence possible investments are rapidly reimbursed.

Within the scope of ILM, data can be classified in two categpries:

1. Critical information, which is the data used for the day to day operations, must be located in the primary storage and needs to be quickly recoverable.
2. Important information, which is data that may be located in secondary storage (low cost disks or tapes). This information that needs to be stored due to legal, historical or regulatory considerations.

From the above definitions it is inferred that critical data is accessed frequently while important data is accessed rarely or not at all. ILM systems can hence automatically change the characterisation of data from critical to important via a simple accounting system that measures the frequency of access. The inverse change in data characterisation is also possible.

The function of ILM systems is parallelised with the function of systems that manage the location, recovery and lending of books in public libraries. The cost of books is small compared to the costs incurred in handling the books in such a way that they are easily retrievable. The handling of the books involves moving books around so that frequently used books are more accessible, classifying new books in the correct level of importance and filing low-demand books in less accessible storage. This parallel can give an insight to what an ILM system needs to do, which is to automatically categorise incoming information, automatically and periodically review information categorisation and most importantly, do all this without requiring any human intervention thus alleviating all possible associated costs.

During the introduction of such an automated system, organisations may rightly be inclined to invest heavily in the new technology, based on the hope of recovering the costs by increasing the efficiencies of their data storage. The only underlying danger in this phase is the possibility of data duplication. This danger arises because the operation of the ILM can be confused with the operation of the backup system. Backup is an independent information subsystem that has the responsibility of making copies of critical data on a regular basis (usually daily or weekly) on low cost disk or tape storage space. During the period in which a particular data file is still characterised as being in critical status, regular backups need to continue being taken. In short, backup is a process that protects data whle it is still critical. The ILM data archiving is a completely distinct process by which operational but non-critical data is moved to safe, long term storage. Backup systems hence need to be ILM aware and stop making copies of data that is already in secondary storage, so as to avoid reducing efficiency and increasing the overall cost.

The solution that has been proposed [11] is called the distributed backup. The distributed backup is an ILM aware backup that completely removes the necessity for daily backups of critical data on to costly and difficult to manage magnetic tapes, thereby automatically reducing the costs incurred by organisations. A distributed backup system will store data distributed around network clients onto centralised disk storage in compressed format. When data is required for access, data is recovered and transferred to the local client. This approach has the ability to ensure that ILM controlled data is backed up without the danger of duplicity. The backup system uses the ILM's archive indicators to guarantee that there is only one copy of the data being maintained, either in the backup or the secondary storage. The archive indicators can tell the backup subsystem that certain files have been archived by the ILM. These files can hence be deleted from the backup storage space, thus maintaining high levels of efficiency.

Distributed backup minimises data storage requirements for backup while making the backup operation a faster process that can hence become a more frequent event. Additionally, restoring data from backup becomes simpler and the overall complexity and cost of system administration is reduced. The life of a backup file may also become the subject of study and optimisation. Its handling is different when a file is created, while it is still characterised as critical and when eventually it is archived by ILM and the backup file is deleted. The management of backup files is called the Backup Lifecycle management (BLM) and runs in parallel with the ILM which only deals with the primary critical data [11].

Current military and aerospace computational systems are extremely complex in functionality and large in the scale and extent of application. The amounts of information they handle are challenging, merely because of the volume of traffic they involve. The fact that such defence systems involve both military government organisations and private suppliers and contractors have lead to the development of specialised integrated life cycle information and data management solutions like the system presented in [12].

4 Symmetric key management systems

The success of data lifecycle management systems has lead to the application of similar principles to the management of symmetric encryption keys. Organisations are critically dependent on cryptographic systems for protecting sensitive information in their computing infrastructure. Several standards have been developed in the case where asymmetric key encryption schemes are employed, such as PKCS10, PKCS7, CRMF and CMS [13]. However, symmetric key algorithms have been deployed using non-standard techniques for managing encryption and decryption. There is no standardised architecture or protocol that can provide cross application symmetric key management services [13]. Key management is an issue that large-scale organisations need to address in an integrated way, irrespective of whether the keys are symmetric or non-symmetric.

Symmetric Key Management Systems (SKMS) aim to specify possible application requests for life-cycle services for symmetric keys and to implement responses to these requests and provide services using public key encryption algorithms [13]. Such services include secure generation, escrowing, management, handling and destruction of symmetric encryption keys [14], [15].

5 The necessity of developing a management key system in a symmetrical encryption system

The secure communication model which is developed in the research is based on the symmetrical encryption system [8], [9], [10]. The basic characteristic of a symmetrical encryption system is the use of the same secret key for both the encryption and decryption processes. This means that a user that received a ciphered message cannot recover its initial plain text if he does not know the secret key that was used in order to calculate the received ciphered text [4], [5]. For this reason each user of the information system which is based on a symmetrical encryption system should notify the users that he wishes to communicate with, the secret key that he will use in order to encrypt the data that will promote to transmission. In this paper we denote the above key for each user as personal secret key for encryption.

Each person uses an encryption key in order to decrypt the messages that he receives from the network. The above key is the encryption key (personal secret key for encryption) published by the sender of the ciphered messages and notified to the recipient via secure channel of sharing secret keys.

Let assume that two users belong in a certified team Alice and Peter. Certified group of users means that each user of that group will not distribute the other users key to not certified users. In the symmetrical encryption system described below, Alice publishes a personal secret key of encryption K_A. From the other side Peter publishes his own personal secret key of encryption K_P. In order to acquire possibility of bidirectional communication Peter and Alice exchange via secure communication channel or in personal physical level their personal secret keys. Now the K_A for Peter and the K_P for Alice are the keys of encryption correspondently. Also in this communication system the following elements are denoted:

$$C_X=Enc(P_X,K_X) \tag{1}$$

$$P_X=Dec(C_{X,}K_X) \tag{2}$$

The above expressions present the encryption and decryption functions. Thus the ciphered message C_X is the result of the initial plain text P_X with the personal secret key K_X via the encryption $Enc(P_X, K_X)$ function. In the other side the ciphered text C_X which is calculated with a certain personal secret key can be decrypted only by using the same personal secret key K_X with the decryption function $Dec(C_{X,}\ K_X)$. The result will be the initial pain text P_X.

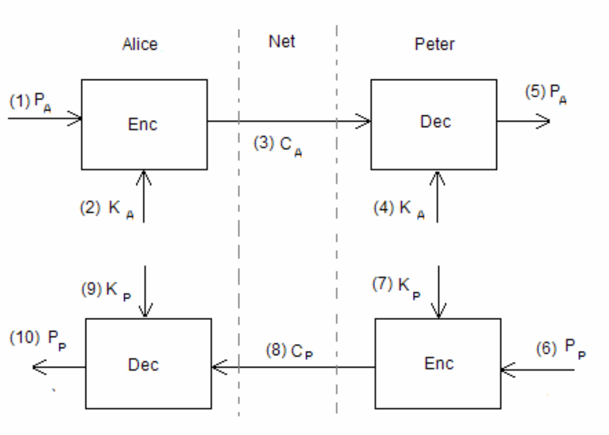
A symmetric cryptographic system in order to achieve secure bidirectional communication works in the following way. Alice in order to send a plain text (P_A) to Peter and to the rest of the certified users, produces a ciphered message C_A using the equation (1) and her personal secret key K_A.

$$C_A=Enc(P_A, K_A) \tag{3}$$

Then Alice transmits via the open network the ciphered text C_A to Peter and to the certified group of users. This C_A text can decrypted using the expression (2) only if the receivers have Alice's personal secret key K_A which was previously shared via secret communication channel as mentioned above. In the following expression is shown the product of the plain text P_A sent by Alice via the decryption function with the parameters C_A and K_A

$$P_A=Dec(C_A, K_A) \tag{4}$$

Corresponding process will be also followed by Peter in order to transmit a text P_P to Alice and to the rest certified users. Concretely it will use the personal secret key K_P and the encryption function Enc in order to produce the ciphered message C_P which will send to Alice and to the rest certified users via the open network. Alice and the rest of the certified users will receive the C_P encrypted message and via the decrypted function Dec and Peter's personal secret key K_P will recover the plain text P_P. In the following scheme 2 the architecture of a secure bidirectional communication of two users of certified group using symmetric cryptographic system is presented.



Scheme 2: Bidirectional user communication based on symmetric cryptographic system

As shown from the above architecture, in the foreword communication is used a secret key and concretely the personal secret key of the transmitter. Thus when Alice transmits the messages she encrypts them with her own personal secret key K_A. On the other side of the communication system Peter and the others certified users must use the same K_A key to decrypt the message. When Peter or any other certified user transmits a message uses his own personal secret key of encryption, i.e., if Peter makes the transmission then he uses his personal secret key K_P to encrypt the new text P_P. Alice or the other certified users, uses the K_P in order to decrypt the received P_P.

From the above it is clear that a bidirectional communication consists of two periods of communication. Each period is established by the prime caller of the communication, i.e., which user become transmitter. When Alice encrypts and transmits the messages the first period of communication is established (1 until 5) and when any of the other certified users, for example Peter, transmits encrypted messages to the other users, for examples to Alice, then the second period of communication is established (6 until 10).

An important characteristic of each period is the secret key that being used each time. The secret key that will be used is the personal secret key of sender. Thus, in the first period of communication where the sender is Alice, the personal secret key K_A is used. In the second period the sender changes and the new sender, Peter, determines now the new personal secret key K_P.

The classical symmetrical cryptographic systems use a common secret key for both the two communication periods. These cryptographic systems are used for the secure communication between two users only and in case of the key break from intruders the communication is open in both the two periods. However in a group of certified users, the use of personal secret key instead of one common secret key for each pair is acceptable while at the same time it offers advantages. Each period of communication is protected by a different key of communication. Thus each hacker should make double computational effort in order to break the two personal secret keys or more keys consequently the total of communication. In order for each user that belongs in the certified group to communicate with the users that he wishes, he should know their personnel secret keys of encryption while at the same time he notifies them his personal secret key. The memorization of each user keys is impossible while the use of on error key makes the communication impossible. For this reason the proposed cryptographic system of communication proposes a key management system. This describes the processes for the secure storage of the communication keys as well as the way of accessing these keys depending on the requirements of communication.

6 Standardisation of symmetric encryption keys

Standardisation of the services required for the effective use of symmetric encryption keys is a domain that has not been sufficiently developed [14]. Surprisingly, this is due to the fact that numerous business applications have been developed in the past that required the use of such keys. The management of the symmetric encryption keys became just another ancillary subsystem of the overall application, safely buried inside the overall product and hence its details were never publicised.

The problem surrounding symmetric key management becomes more apparent when seen from the perspective of the administration of IT operations of e.g. a commercial enterprise that accepts payments via credit cards. In this example, the system would be required to manage:

- A point of sale application communicating with an extended network of point of sales terminals.
- An e-commerce application that handles payments using the received credit card numbers.
- A payment processing application that settles transactions after communication with the credit card network.
- A back office application that handles accounting
- A security application for detecting fraud.

In addition to the above and with the extensive of laptops and PDAs for business purposes, there are even more authentication operations that need monitoring and management. More overheads are added on, due to the existence of databases and operating system specific authentication mechanisms. Overheads are increased furthermore since different applications may coexist within the limits of a particular organisation that are products of different vendors and therefore employ their own different design for symmetric key management. Administration problems are not just problems of operating a particular type of software. Each security subsystem conforms to its own technology and therefore requires its own training, documentation, procedures and audits (such as the audits performed by credit card transaction regulatory authorities or sensitive personal data protection authorities). Apart from increasing cost for companies, all the above factors also increase the risk of an eventual breach of security [14]. Software engineering has been faced with similar problems in the past and the answer has always been to abstract services from applications. Hence it is current practice that all applications use the same Domain Name System service (DNS) for hostname-IP-address resolution, the same Dynamic Host Configuration Protocol service (DHCP) for dynamic IP-address allocation and the same interface (ODBC, JDBC) in order to access a particular Relational Database Management System (RDBMS) for data management. Consequently, the symmetrical key management capability must also be abstracted. Applications need only have access to a key management service that runs independently in its own standardised infrastructure. Encryption and decryption will hence be enabled in a uniform way that can offer a standard and adequate level of security.

Application programs are usually the top end of a layered architecture of services. Since encryption is non-standard, it is general practice in current encryption schemes that data is encrypted in the lowest possible layer. In this way all possible top layer combinations have access to the data protection service. This however leaves more room for attackers to operate. If on the other hand, encryption were standardised and data could be encrypted at the topmost level of its architectural structure, unauthorised users would have to attack the application program itself, in order to gain access to unencrypted data. Consequently, without eliminating the risk of such an attack, the second strategy minimises the possibilities of unauthorised access to unencrypted sensitive information.

Following the analysis in [14], an effective symmetric key management system should possess the following properties:

- Centralized policy-definition and key-management;
- Platform, application and language independence;
- Highly-availability; yet KM client applications must be able to continue functioning - i.e., encrypt and decrypt data - even in the absence of the KM service;
- Scalability;
- Security;
- Leverage for existing standards and security certifications of cryptographic components;

7 Management System of secret keys in symmetric cryptographic systems for communication between certified users

The key management system for the communication (personal secret keys of encryption) consists of the physical means (memories) for the keys storage and the rules that determine the way of recovering them from the storage memory and the way to use them. For example in order for Alice to communicate with Peter or with the other certified user of the group, she should -with the key management system- store with safety in memory her personal secret key K_A and the personal secret keys of all the group users including Peters personal secret key K_P. With the beginning of the communication the key management system will automatically recover the suitable keys, in the particular case the K_A and K_P, and will use them suitably in order to achieve encrypted communication [15].

Thus for the design of the key management system the following apply:

1. A data base for storage of the users personal secret keys of encryption, denoted as Users Data Base - UDB. The UDB contains the keys of users as records that contain personal data like ID number or name user and the corresponding personal secret key.
2. A file used for the storage of personal secret key of encryption of the user, denoted as User Details File – UDF.
3. A Master Key is used for the users' access control in the communication system. It is also used for encrypting the users' personal secret keys that are stored in the memory system and constitute the session keys. The substance of establishing the Master Key that will be mentioned as password is the following: Only the user that knows the password can access the encrypted communication system. Thus the password is used for the encryption via the function *Enc* of a unique number and the result is stored in the UDF in order to use it for the access control with a process which will be described below. Also the password is used for the secure storage of the personal secret keys.

As an example Alice receives from Peter his personal key K_P. Via the passwords management system she should encrypt the K_P key with the password and store it in the UDF_A. At the same time in the User Details File – UDF_A, Alice's password is ciphered and stored. At the beginning of communication with Peter, the key management system should recover the K_A and K_P keys from UDF_A and UDB_A, calculate their initial (decrypted) value and use them according to the requirements of communication.

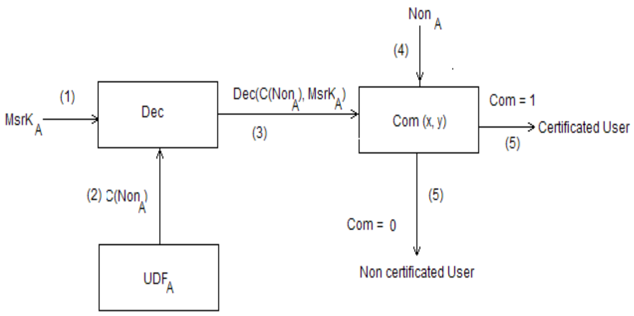
One of the most important operations of the key management system is the access control. Through this process is verified whether the user that attempts to utilize the communication system belongs to the certified group or not. Characteristic sign of user's certification is the knowledge of the password at each access point. The password is not common for the entire system but in each terminal the user's personal password has been set. Thus Alice in her terminal has set her personal password (Master Key of Alice) $MsrK_A$. The system's administrator has determined for the particular terminal a unique number of access control Non_A. The key management system has used the Enc function and will encrypt the Non_A using as key the $MsrK_A$.

The ciphered result $C(Non_A)$ of the function will be stored in the users detail file UDF_A. Through a specific access control process the entering user is certified. The access control process consists of a sequence of actions in order to get the desirable result. The access control process will be reported as access control routine. Concretely for Alice's management system: When Alice attempts access in her terminal the system requires her password. Assume that she enters the correct password $MsrK_A$. The management system will then recall from UDF_A file the ciphered text $C(Non_A)$. Function Dec deciphers the $C(Non_A)$ using Alice's password. Since the $C(Non_A)$ is cipherd with Alice's password $MsrK_{A,}$ if the password entered by Alice is correct the decryption result will the same as the unique number Non_A

In order to check the equality of the decryption result with the Non_A, the comparison function Com is denoted as follows:

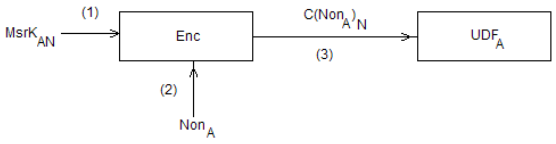
$$Com(x,y) = \begin{cases} 0 & \text{if} \quad \chi \neq y \\ 1 & \text{if} \quad \chi = y \end{cases} \qquad (5)$$

Thus for Alice's terminal applies that $Com(Dec(C(Non_A),MsrK_A),Non_A)=1$. Through the Com function the decryption result is compared to the unique number Non_A and in case of equality the user is considered certified. In the opposite case the entry in the terminal is not allowed. In the following scheme 3 the architecture of the access control routine for Alice's terminal is presented.



Scheme 3: Access control routine for Alice's terminal

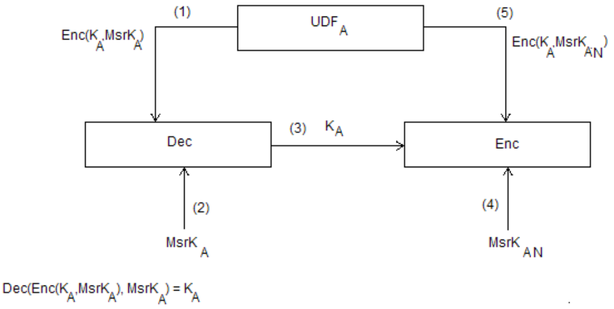
Basic requirement for the information systems safety is the frequent change of the passwords. In order for the users to have this possibility, a routine is determined to define how the keys management system will act in order to change passwords. The objective aim of the routine is to store in the UDF file the unique number ciphered with new password. In this case Alice enters the new password $MsrK_{AN}$. The new password is combined via the Enc function with the unique number Non_A in order to get the ciphered result $C(Non_A)_N$. This result is stored in the UDF_A. So, in Alice's user detail file the new password is stored. In the following scheme 4 the architecture of the password change routine for Alice's terminals is presented.



Scheme 4. Password change routine for Alice's terminals

In a previous paragraph was reported that the password is used as a master key for the rest of the secret keys of communication (session). This means that the personal secret key of encryption in the users detail file (UDF) is ciphered with the password MsrK. In Alice's user detail file are stored the value $C(Non_A)$ that is used for the access control and the value $Enc(K_A, MsrK_A)$ that it is the ciphered version of the personal key of encryption K_A. When changing the password with the above described routine the value $C(Non_A)$ changes but the value $Enc(K_A, MsrK_A)$ still depends on the previous password. For this reason a routine is determined to store in the UDF_A the value $Enc(K_A,MsrK_{AN})$. In this way the system will secure Alice's new password.

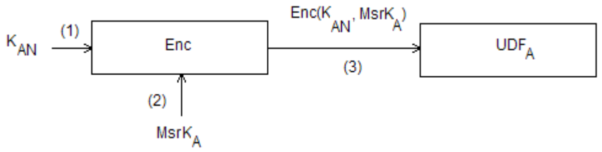
In this routine, the management system initially recovers from the UDF_A the ciphered with the old password key $Enc(K_A, MsrK_A)$ and combines it via the Dec function with old password $MsrK_A$ in order to get Alice's personal key K_A. Then the K_A is used in the Enc function with the new password $MsrK_{AN}$ to get the result $Enc(K_A,MsrK_{AN})$ which is Alice's key of encryption ciphered with the new password. This result is stored in Alice's user detail file UDF_A replacing the old one. In the following scheme 5 the routine of changing the Alice's personal key's encryption is presented



Scheme 5: Routine of changing the Alice's personal key's encryption

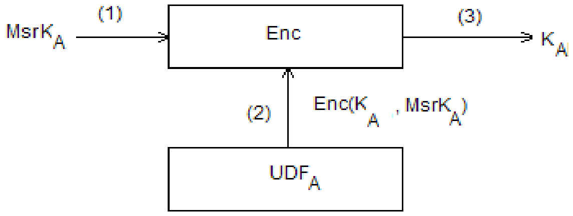
As with the policy of frequent change of passwords for the information system security, the frequent change of the communication (personal secret keys) should be applied. While in the first policy the aim is to protect the stored session keys with the policy of changing the communication keys the aim is to protect the communication with the other users.

When Alice decides to change the personal secret key of encryption $K_{A,}$ then the management system functions is as follows. At the beginning it asks and receives from Alice the new key K_{AN}. With the password it encrypts the new key using the Enc function. The result Enc(K_{AN}, $MsrK_A$) is stored in the users detail file UDF_A where the old key used to be. Thus the new value Enc(K_{AN}, $MsrK_A$) is stored in the place of the old Enc(K_A, $MsrK_A$). With the specific routine the system stores the new personal secret key in a determined position ciphered with the password. In the following scheme 6 the routine of changing the Alice's personal key's encryption is presented.



Scheme 6: Routine of changing Alice's personal encryption key

In the previous paragraphs was analyzed the way the management system handles the data stored in the user detail file UDF. The final aim of the above processes is the secure storage of the communication keys until the beginning of communication for each user. Thus upon request from the user to start communication, the management system has to restore the communication keys (personal secret keys for encryption). In Alice's user detail file is stored her personal secret key ciphered with the password she has set Enc, Enc(K_A,$MsrK_A$). The system restores the above value from the storage and deciphers it using the Dec function with the key having the value $MsrK_{A.}$ The result is the value K_A that is the deciphered version of Alice's personal secret key. The K_A henceforth can be used from the system of the encrypted communication. In the following scheme 7 the routine of restoring Alice's personal key of encryption from the users detail file is presented.

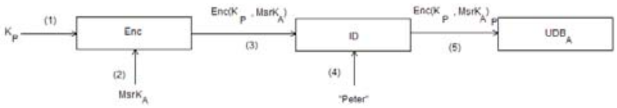


Scheme 7: Routine of restoring Alice's personal key of encryption from the users detail file

In a previous paragraph was mentioned that in order for two users to communicate they should have each others key. When a user receives somebody else's key he should safely store it. The key management system proposed in this article has a routine for entering a new user for this process. In order for Alice to communicate with Peter she receives his personal key K_P either through a secure communication channel or in person. Peter's key is encrypted with Alice's password $MsrK_A$ using the Enc function. At this point a function is defined in order to identify the encrypted key so that it can be searched later on. Thus for Peter's key:

$$ID(\text{“Peter”},Enc(K_B,MsrK_A))=(Enc(K_B,MsrK_A))_P \quad (6)$$

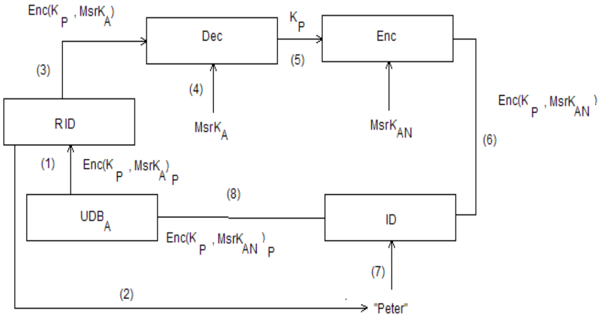
The ciphered result with the ID function is associated with his user and then is stored in the first available position in Alice's user data base UDB_A. In the following scheme 8 the routine of entering user "Peter" in Alice's database is presented.



Scheme 8: Routine of entering user "Peter" in Alice's database

In a previous paragraph the processes were described that is executed by the management system when a password is changed and when a personal secret key is encrypted with a new password and stored. Similar process is executed also for the users database. In this database are stored the users personal secret keys ciphered with the password of the terminal's user. Thus Alice has stored in her database UDB_A Peter's key K_P ciphered with her $MsrK_A$ and storing value (Enc(K_Π, $MsrK_A$))$_{P.}$ When Alice decides to change her password $MsrK_A$ then the key K_P stored in the database has to be ciphered with the new password to get the new value (Enc(K_Π, $MsrK_A$))$_{P.}$ When a new password arises the management system along with the previously described routines also executes the following actions. From Alice's users database Peter's ciphered key with value (Enc(K_Π, $MsrK_A$))$_P$ is restored and used in a reverse function ID named RID. The reverse ID (RID) extracts the user's name that corresponds to the ciphered key and value Enc(K_Π, $MsrK_A$) which is Peter's key K_P ciphered with the old password $MsrK_{A.}$

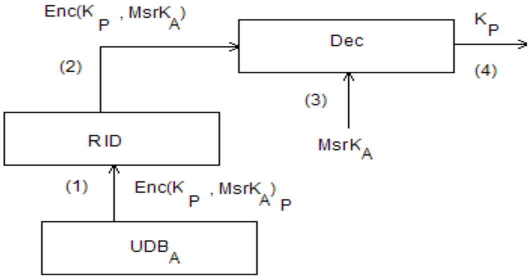
Then the ciphered value is deciphered with the old password and the result (plain key K_B) is ciphered with the new password $MsrK_{AN}$. The result of encryption is associated with the user's name resulted from the RID and is stored in the UDB_A in the place of previous registration. In the following scheme 9 the routine of modifying the data in the “Peter” record in Alice’s database when she changes password is presented.



Scheme 9: Routine of modifying the data in the “Peter” record in Alice’s database when she changes password

Alice however recorded in her database other users too, besides Peter that she wishes to communicate with. It is obvious that the above process is repeated by the management system for every record. Respectively to the routine of restoring the personal secret key of encryption, a routine for forwarding the user’s encryption key is defined. The aim it to restore from the users database UDB the key of the user that the terminal owner wishes to communicate with. Consequently when Alice decides to communicates with Peter the password management system through the routine of forwarding the user’s encryption key will forward the key K_P to the communication system. The routine has as follows:

From Alice’s database UDB_A the value $(Enc(K_B,MsrK_{AN}))_P$ is taken.This is found through the identification given by the ID function. From the restored ciphered value the identity is extracted using the RID function and is forwarded to the Dec function to be deciphered with Alice’s password $MsrK_{A.}$ The result that is Peter’ key in its deciphered version K_B is forwarded to be used in the communication system. In the following scheme 10 the routine of restoring Peter’s encryption key from Alice’s password management system is presented.



Scheme 10: Routine of restoring Peter’s encryption key from Alice’s password management system

8 Conclusion

The new approach of a symmetrical ciphered communication that is presented in this paper is the use of two personal secret keys for bidirectional communication. Which key will be used in each communication period depends on who starts the period. The use of pair of keys is recommended for groups of certified users (e.g. Military environments). With this fact each user exports one and unique personal secret key and shares it with the group and they do not need a common key with each user. The use of two keys in each bidirectional communication instead of one common brings a higher security level. While is determined the way of use of communication keys is resulted the needs for the secure storage and restore of these keys. Thus it was developed a autonomous by the process of communication system a key management system (personal secret keys) for communication This designed in autonomous routines which are executed depending on the user requirements.

Acknowledgments

The authors are grateful to Prof. Antonios Andreatos, PhD, Director of the Department of Computer Sciences of Hellenic Air Force Academy for his continuous support during the research and the writing of this paper. We also want to thank Colonel George Geroulis, MSc, MPhil, PhD, Director of Military Education of Hellenic Air Force Academy who was very helpful with the many insightful discussions and comments on the early drafts of this work.

References:

[1] NIST Special Publication 800-21, Guideline for Implementing Cryptography in the Federal Government, Annabelle Lee, *Security Technology Group -Computer Security Division -National Institute of Standards and Technology Gaithersburg*, MD 20899-8930.

[2] D.W. Davies and W.L. Price, Security for Computer Networks, *JohnWiley&Sons,New York, 2nd edition*, 1989.

[3] W. Fumy and P. Landrock, “Principles of key management”, *IEEE Journal on Selected Areas in Communications, 11* (1993), 785–793.

[4] W. FUMY AND M. LECLERC, “Placement of cryptographic key distribution within OSI:

design alternatives and assessment”, *Computer Networks and ISDN Systems, 26* (1993), 217–225.

[5] M. Abadi and R. Needham, “Prudent engineering practice for cryptographic protocols”, DEC SRC report #125, *Digital Equipment Corporation*, Palo Alto, CA, 1994.

[6] R. Anderson and R. Needham, “Robustness principles for public key protocols”, *Advances in Cryptology–CRYPTO ’95 (LNCS 963)*, 236–247, 1995.

[7] B. Preneel, R. Govaerts, and J. Vandewalle, editors, Computer Security and Industrial Cryptography: *State of the Art and Evolution (LNCS 741)*, 193–210, Springer-Verlag, 1993.

[8] ELECTRONIC INDUSTRIES ASSOCIATION (EIA), “Dual- mode mobile station – base station compatibility standard”, *EIA Interim Standard IS-54 Revision B* (Rev. B), 1992.

[9] ISO 11166-1, “Banking – Key management by means of asymmetric algorithms – *Part 1: Principles, procedures and formats”, International Organization for Standardization*, Geneva, Switzerland, 1994.

[10] —, “Criticism of ISO CD 11166 banking — key management by means of asymmetric algorithms”, W. Wolfowicz, editor, *Proceedings of the 3rd Symposium on State and Progress of Research in Cryptography*, Rome, Italy, 191–198, 1993.

[11] Farajun, Eran, “The Key to Information Lifecycle Management is Cost-Effective Backup”, Computer Technology Review, January 1 2006.

[12] “Integrated Life-Cycle Information and Data Management Solutions”, http://www.xwave.com/files/credentials/integrated_life_cycle_information_management.pdf

[13] Stephen.Wilson, “Symmetric Key Management System (SKMS)”, http://idtrust.xml.org/symmetric-key-management-system-skms

[14] Arshad Noor. “Securing the Core with an Enterprise Key Management Infrastructure (EKMI)”. *ACM IDtrust '08,* March 4-6, 2008.

[15] Nikolaos Doukas, Konstantinos Ntaikos and Nikolaos Bardis, “Integrated Information Life-Cycle, Data Management and Secret Key Lifecycle Management for Military Applications”, 10th WSEAS Int. Conf. on Mathematical Methods, Computational Techniques And Intelligent Systems (MAMECTIS '08), Corfu Island, Greece, October 26-28, 2008.