Design and Development of a Secure Military Communication based on AES Prototype Crypto Algorithm and Advanced Key Management Scheme

NIKOLAOS G. BARDIS
Adjunct Professor
University of Military Education
[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
University of Military Education
[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
doukasn@sse.gr,
nikolaos@doukas.net.gr

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

Abstract: - In this article, a study is presented that aims at the development of a prototype system for the secure real-time exchange of messages between users of workstations connected to the same TCP/IP network. The security is provided based on the AES prototype cryptographic algorithm. An advanced key management scheme is used within this system that enhances the security of the system, reduced the effects of possible security breaches and simultaneously hides from users the unnecessary complexity related to handling multiple encryption keys. The scope of application is military units and is intended to become the basis for the design and development of an integrated framework for the exchange of secure messages between different sites of military or other organizations that are concerned about information security. The present design is limited in its application to local area networks only. There are however no fundamental restrictions and an expansion to wide area networks and the internet is also possible. The design of the application is firstly presented. Problems of security and ease of use that are related to the management of the secret encryption keys are explained. A solution is hence presented for these problems, that is based on an innovative scheme for key storage and management. The design and implementation of the application is presented in detail along with description of its basic functionality. The plans for application and further development of the application are described and conclusions are finally drawn.

Key-Words: - Secure messaging, AES, encryption, key management

1 Introduction

Military organizations, similarly to other organizations that are concerned about the security of information exchanges, have always heavily relied on secure exchanges of short messages. A reliable system for such message exchanges is considered to be a particular strength for such organizations [2]. Information security is a more general concern for officers of the armed forces and armed forces personnel is in constant pursuit of better procedures to ensure data protection and integrity. Consequently, improvements in the technology for providing safe data storage and secure communication and transfer of data between remote military units are continuously being investigated. The development and expansion of the Internet have established it as one of the most important communications channels both at the level of large scale organisations (banks, multinational companies etc) and at the level of simple users.

The purpose of this research was to develop a framework for the provision of secure communications via the exchange of short text

messages between terminals located at remote sites. The communication provided is real – time and based on symmetric cryptographic algorithms for providing the security, namely the AES. The problems and overheads related to the management of the secret keys are considered. The application of an advanced scheme for the automation of the key management is proposed. The application of the key management system averts the danger arising from possible compromises in the secrecy of passwords while also reducing the overheads incurred upon users in order to achieve the desired level of security in their communication.

2 Operation of the symmetric encrypted communication system

The basic operation principle for a system of symmetric cryptographic communication is the use of a shared secret key that is used for both encryption and decryption. The secret key is the most important component of the encryption system, as it is the principle means that transforms clear messages to ciphertexts. The disclosure of the key to malicious users jeopardises the essence of communication. For a group of users of a symmetric cryptography system, the method of a shared secret key is widely used. With this method, if a malicious user were to join forces with enemy cryptanalysts, they would only be capable of disclosing their own secret keys and hence disclose all communication in which they took part. This way, in a group of authenticated users such as the users in a military environment, the use of a shared key for all users entails problems since any disclosure of the key would annihilate security for all communications.

For this reason, instead of using a single key for everyone, a protocol can be designed for which every user is issued a secret key which they distribute via safe communications channels or via personal contact to all the users with whom they are interested of securely communicating. The application presented in this article is developed based on the above protocol. More specifically, a user of the application is assigned a personal key (of their own choice or automatically generated) that they disclose to certified users of the same application that have access to the common network. On the other side, the same user receives the corresponding secret keys from all these users. The above protocol gives the possibility for duplex encrypted communication between users. The application uses the secret key of its owner for sending data to the network and the secret keys of other users for decrypting messages it has received from them.

The operation of the encrypted communication scheme is illustrated in Figure 1 below. The symbol (1, 5) PA denotes the plain text message originating from user Alice, the symbol (2, 4) KA the personal Key of user Alice, the symbol (3) CA the Cipher Text corresponding to Alice's message, the symbol (7, 9) KP the personal Keyof user Peter, the symbol (6, 10) PP is the plain text message originating from Peter and the symbol (9) CP the ciphertext corresponding to Peter's message..

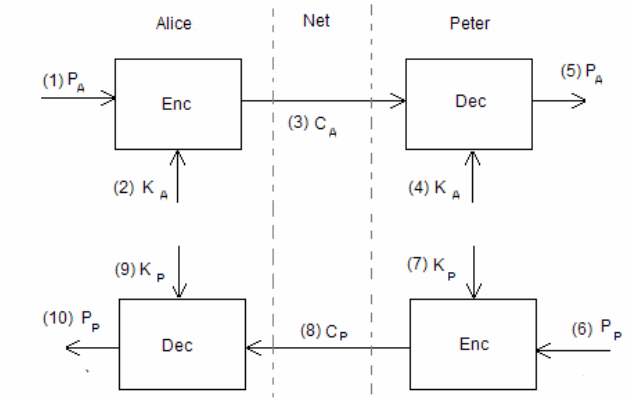

Figure 1: Schematic operation of the system of duplex communication with use of two keys

In this figure it can be seen that in a bidirectional symmetric encrypted communication system, two keys are used. In every epoch of this communication, the sender's personal key is used. When Alice is the sender (1-5), her own personal key KA is used to encrypt her message. On the other side Peter as a receiver uses the sender's key (i.e. Alice's) to recover the original message via decryption. The inverse procedure is performed in the following communication epoch (6-10) when Peter as a sender determines that his own key Kp is in use.

Having defined the protocol for communication, the need for designing a system for handling user passwords and secret keys becomes apparent. Each user receives from the remaining users of the same group their own personal keys, with which they can decrypt the corresponding messages. Memorising all these passwords and entering them in the system whenever necessary, is considered impossible. The secret key management system has as an aim the secure storage and retrieval of passwords and their

use depending on the needs of the communication. The password management system includes various subsystems, to be described later in this paper and acts in conjunction with the secure communication system. In reality, the communication system informs the secret key management system for the requirements of the current exchange. In return the secret key management system retrieves and forwards the required keys so as to achieve successful message exchanges.

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 [14], [15].

2.1 The key management problem

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 [14]:

- A point of sales 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 **Error! Reference source not found.**. 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.

In the pilot phase of the development of this application, a simple approach to solving the key management problem will be taken. This approach will be sufficiently explained in the following section. A more comprehensive approach is under development and will be presented in the near future.

2.2 Key management functionality required

An innovative scheme for the integrated management of security system secret encryption keys and stored data lifecycle has been recently proposed [14]. In this scheme, the various functionalities that the key management system must provide are outlined. These functionalities are reflected in the operation of the secure message exchange system as well and are outlined below.

1. User registration – 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).
2. User initialization – 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.
3. Key generation – 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.
4. Key installation – 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.
5. Key registration – 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).
6. Normal use – 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.
7. Key backup – 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.
8. Key update – 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.
9. Archival – 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.
10. Key de-registration and destruction – 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.
11. Key recovery – 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.

3 Architecture of the application

In a previous paragraph the overall operation of the application was described. This operation is supported using various subsystems that from an implementation point of view can be seen as commands that when properly combined lead to the desired result. The encryption and decryption subsystems can be singled out as two such fundamental subsystems. As autonomous entities, these subsystems have as input the secret key and either the clear message or the ciphertext and as output, either an encrypted or a deciphered result.

The process of calculating the result directly implements the mathematical model of the AES cryptographic algorithm.

After having implemented the encryption functionality and achieved the level of security necessary, the application must be integrated with the subsystems for the handling of the secret keys and passwords. These subsystems are also part of the communication protocol in the key management phase. As it was mentioned before, the purpose of this subsystem is the safe storage of secret keys and passwords for each user and the access control function for the application. The Master Key model is applied in order to achieve these goals. The master key is used by information systems for the secure storage of communication passwords (session keys). User keys are encrypted with the master key before being stored for the purposes of the application, thus ensuring their security. As a means of saving the keys, a database has been designed within the application. This database will be referred to for the purposes of this article as the User Database and will contain entries concerning user personal data and their corresponding secret key, encrypted using the master key. Additionally, the use of a file is defined, with the aim of storing the personal encryption key of the user. The key is saved in the file, after being encrypted with the application master key. This file is called the user data file.

3.1 Access Control

For the access control function, the following procedure is defined. The application uses a unique number hard coded within the application source and encrypts it with the user password. This will be referred to from now on as access code. The encrypted result is saved in the user data file. For access control the application asks the user to enter the password and uses it to decrypt the encrypted unique access code value stored in the user data file. If the result is equal to the unique number stored in the source code then access is allowed, since the access code is correct. The above procedure constitutes the access control routine and is executed at application startup.

The access control routine is a subsystem of the password management system. The unique number is entered in the source code from the system administrator and is changed at regular intervals so as to achieve a high level of security. The principle aim of the management system is to receive the user keys from the application users and safely store them in the user database. This process is executed with the help of the remaining subsystems.

User password update routine

The first one of these subsystems is the user password update routine. This is executed when application users require a change in the password or secret key they use. This subsystem has as input the new user password (or secret key). It hence receives the unique access control number used for access control and encrypts it using the new password. The encrypted result is hence stored in the user data file in the place of the old encrypted number.

This way during the next user access, the new password will need to be entered so that the access control routine allows access to the application. The above functionality is now however sufficient. The access code is used by the application to encrypt the communication secret keys inside the user data file. It is additionally used as a primary key in the user database. For this reason, the following two routines need to be developed.

3.2 User password update routine for the user data file

The user password update routine for the user data file receives as input the encrypted personal secret key decrypts it with the old password and encrypts the result with the new password. The result of the encryption is stored in the user data file in the place of the personal communication key.

3.3 User password update routine for the user database

Similarly, the user password update routine for the user database executes a similar procedure with that of the previous routine for all user keys that are stored within the user database.

It therefore becomes apparent that when a new password is requested from a user, all three routines above will need to be executed. This will result in an update in the encrypted values stored for the unique access code and all the personal secret communication keys, so as to reflect the change in the password.

3.4 Personal encryption key change routine

When the user requires the change of the personal secret encryption key that the application will use for communication, the personal encryption key change routine is executed. This routine receives the new secret key, encrypts it using the password and

stores the result in its correct place inside the user data file

3.5 Communication contact secret key update routine

A similar procedure is followed by the communication contact secret key update routine. This is executed when the user requires an update of the secret communication key stored in the user database for a particular user contact. During this change, the user data for the particular contact are recovered, decrypted, updated with the new key and stored back inside the database, replacing the old entry, after they are once more encrypted.

3.6 New user introduction routine

In order to introduce a new user in the application, with whom communication will be possible, the new user introduction routine is called. This requires as input the new user's personal data, together with their personal secret encryption key. The routine encrypts the key with the password and combines the result with the personal data to formulate a valid entry to be stored in the user database, in the first available position.

When the communication system attempts to start a new duplex communication, it notifies the password management system about the user it needs to connect with. The purpose of this notification is to recover from the appropriate entry in the database the proper communication keys and forward those keys to the communication subsystem. The above operation is completed via the following two routines.

3.7 Personal key recovery routine

The personal key recovery routine receives the encrypted personal communication key from the user data file, decrypts it with the appropriate password and forwards it to the encrypted communication subsystem.

3.8 Contact personal key recovery routine

On the other side, the contact personal key recovery routine receives from the communication subsystem the username with whom communication is going to be established and retrieves the corresponding entry from the user database. When this is recovered, the secret key is decrypted with the password and forwarded to the communications subsystem.

After the two above routines have completed, the application is ready to perform duplex symmetric key encrypted communication as specified.

4. Development of the application interface

The application described in this article is best suited for development based on the object oriented model. The programming language chosen is Microsoft Visual Basic that offers significant capabilities for an efficient window based user interface. The forms (i.e. the windows used) are the means of communication between the user and the application (for the purposes of data entry, function selection and the actual message composition and reception).

The first form in the hierarchy is frmStream. This is essentially an auxiliary form that is invisible to the user and acts as the bridge via which the communications subsystem is connected to the key management subsystem. The form contains text boxes on which the communication keys are stored after they are retrieved by the key recovery routines. Other text boxes on this form store the inputs and outputs of the communications subsystem.

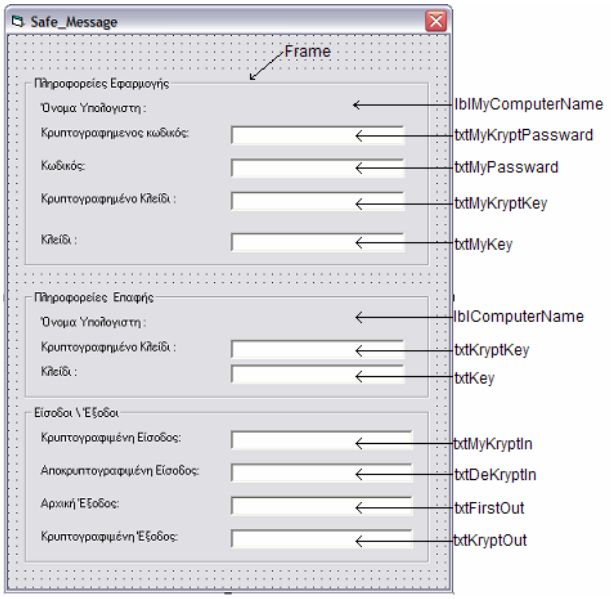

Figure 2: *Auxiliary form frmStream*

At the application startup, an access control procedure is required. For this reason an access control for has been designed, that initiates the access control routine. The users are asked to provide their security credentials. These are verified and after the verification the user is notified of the result. The form will be referred to as frmStart and is the first object that becomes visible at the startup of the application.

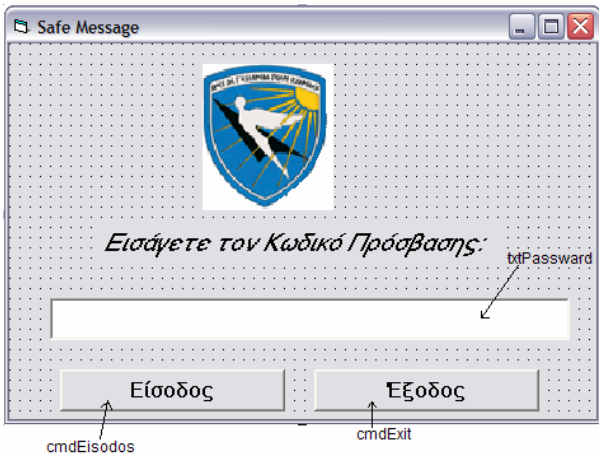

Figure 3: *Access control form (frmStart)*

After the successful authentication, via the access control routine, the principle form of the application becomes visible. This form has been assigned the name Main_Form. This form remains open during the entire period that the application is running. At the same time it is this form that initiates all other functions of the application. It includes the code via which the application accepts requests for exchange of messages from other users. The communication protocol used is TCP/IP and the routine that waits for communication operates via the same interface.

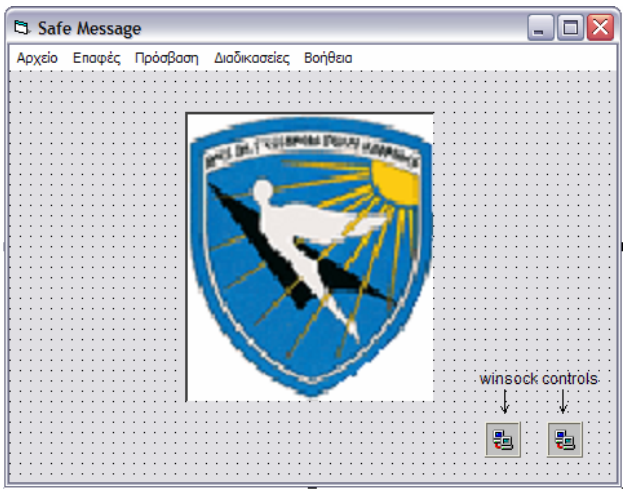

Figure 4*: Main form of application (Main_Form)*

The main form includes a menu of options via which the user is able to call the different communications operations. The visual basic Winsock controls that can be seen on Figure 4 are the VB objects that deal with the communication protocol functions, depending on whether the particular instance of the application is a server or a client.

The new user addition form allows the application administrator to add new users to the application user database. The list of users is available to the user via the "Contacts" option of the menu of the main form.
At the bottom part of this form two command buttons are located. The right hand side button labeled "Εκκαθάριση" (Clear), initializes the values in the text boxes. The right hand side button labeled "Αποθήκευση" (Save), activates the user addition routine by retrieving the data from the text boxes and saving them securely in the database.

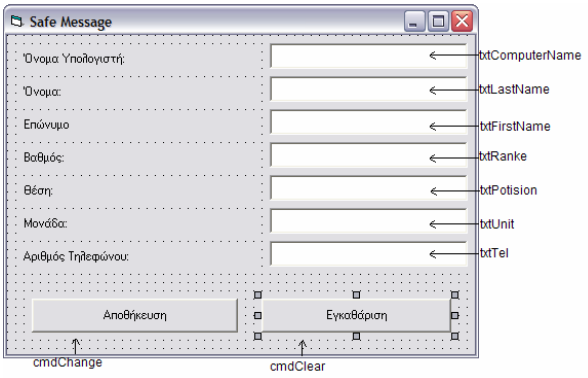

Figure 5*: New user addition form, frmCreate*

The next form is the database access form. It consists of text boxes that are read only, i.e. the user is not allowed to modify the data presented. The form is called frmOpenDataBase and access to it is initiated via the "Contacts" option of the menu of the main form. When the form becomes visible, a query is run that examines the user database, recovers the first entry and displays its fields in the appropriate text boxes.
Via the navigation buttons, the user is able to retrieve all the available contacts. If necessary, the user can call routines for changing contact details or deleting contacts via the frmOpenDataBase form menus. From the "Chat" option of the same menu the user can initiate the communication itself, as well as the routines for changing the user encryption keys.

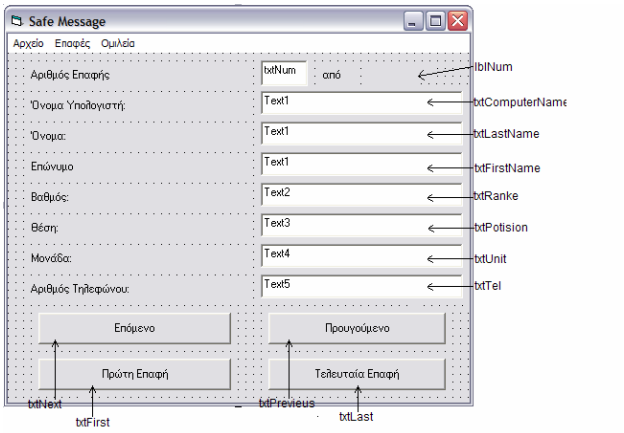

***Figure 6**: Database access form (frmOpenDataBase)*

Another form that may be initiated from the main form menu is the user personal details form (frmMyDetails). It displays the data stored for the user in the database and allows the administrator to change this data via the button labeled "Αλλαγή Στοιχείων" (change data).
There are two more command buttons included in this form. The first one labeled "Αλλαγή κωδικού πρόσβασης" (change password) initiates the routines that have already been described and concern the change the change of the secret access code, while the second one labeled "αλλαγής κλειδιού κρυπτογράφησης" (change encryption key) initiates the routine that changes the encryption key for the particular user.

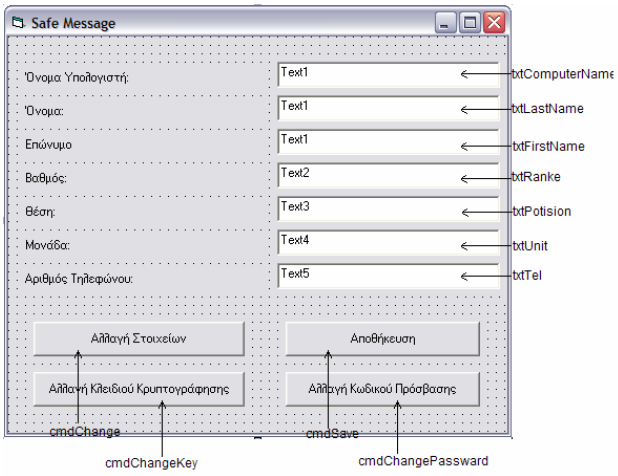

Figure 7: User personal data management form (frmMyDetails)

The above forms are the means via which users and administrators control the system that manages the secret encryption keys. These forms also allow the management of data that is stored in the user database and in the user file. As far as the the communication subsystem is concerned, the application operates as described in the paragraphs that follow.
As it has already been mentioned, the application is based on the TCP/IP protocol for its operation. When in the stand-by mode, the application acts as a server and listens on the network for requests for connection by other users. When a request is received, a separate form is loaded that implements the server functionality and is assigned the name frmTalkingS.
In parallel to the above actions, the encryption key recovery routines are called and inform the auxiliary form about the personal key of the current user, as well as that of the communicating user. The communication form contains a text box in which it displays the messages received from the network, after decrypting them using the appropriate key for the sender (stored in the auxiliary form).
There exists an additional frame on which the local user composes the messages to be sent to the remote user. There is a send button via which the application encrypts the message that has been composed and transmits it via the network. The sending and receiving routines that have been described are controlled by the main form for reasons of simplicity of implementation.
The connection establishment process that is composed of the above steps, presupposes the existence of a request by a remote user. This remote user needs to open the client communication subsystem, controlled by the form named frmTalking. Access to this form is provided by the "Start Communication" option of the main form.
The frmTalking form is similar to the frmTalkingS form, apart from the fact that it contains an extra textbox in which the user enters the username of the remote user with whom they desire to communicate. There exists a command button labeled "Σύνδεση" and sends the request for communication to the remote user. If the response from the remote terminal is affirmative, the appropriate encryption keys are recovered and the encrypted communication procedure is initiated.
In order to interrupt the communication, a separate routine has been designed. This routine is accessible via the corresponding option of the menu in the communication form. The routine is also automatically called when the connection is lost for any reason. The routine deletes the encryption keys and starts the network waiting routine for accepting new communication requests.

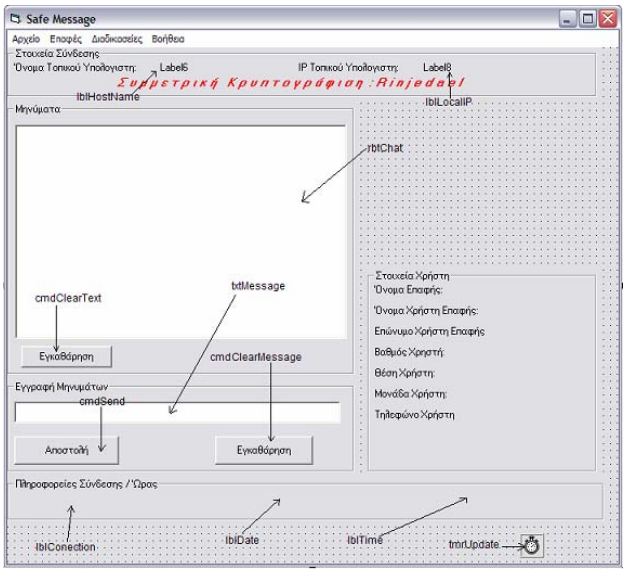

Figure 8: *Client mode communication form (frmTalking)*

5 Further development of the system

As it has already been mentioned, the ultimate aim of the secure message exchange system is to provide a framework for the secure exchange of information between remotely located military units, namely bases of the Hellenic Air Force. This operation, which is already a part of everyday operations of these units, is considered vital and is still currently implemented in many cases based on a legacy telex system.

The functionality of the system described will firstly need to be extended so as to allow the operation over wide area networks or even the internet. This expansion produces some problems concerning the management of IP addresses and the consequent user identification. A more complex protocol will need to be develop that will match users and IP addresses.

Similar problems will occur with stored secret encryption keys. As the extent of the network increases, the number of users will increase accordingly and it will not be viable for all the secret encryption keys to be stored locally on all terminals. A distributed scheme will have to be developed that will handle these secret keys more efficiently. It is expected that this distributed scheme will involve local secret key servers that will be the centre of local communications clusters.

In this way, the extended system will act as a magnified version of the local area network system, where a local cluster replaces the former isolated users and the internet or a wide area network replaces the former local area network. Requests will hence be originating from cluster to cluster and message traffic will be redirected from the cluster server to local authenticated users, via the protocol described in this paper.

6 Conclusion

This article describes the early stages of the design of an application for secure communication for military organizations. The design of this application is based on state of the art encryption technologies, namely AES, and exploits this technology within an environment that promotes and facilitates the use of safe practices on the behalf of users. More specifically, the application takes responsibility for the storage, retrieval and management of the secret keys required for the encryption and proposes a protocol for using keys for users that minimizes the risks for the unit if the secrecy of one or more of the keys is breached and the keys are disclosed to unwanted parties.

Future work includes the possibility for sending encrypted data files, the enhancement of the key management system with new capabilities as well as the improvement of the communication system so as to include security precautions that concern the way in which a group of users is expanded and the control of their authentication procedures. As this application is considered a test prototype, its pilot operation within the limits of a local area network is considered necessary. This operation will discover possible problems or security faults and will lead to an Internet version.

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] Doukas, N., Ntaikos, K. and Bardis, N. "Integrated Information Life-Cycle, Data Management and Secret Key Lifecycle Management for Military Applications". The 10th WSEAS International Conference on Mathematical Methods, Computational Techniques and Intelligent Systems (MAMECTIS '08), Corfu 2008.

[15] Nikolaos Bardis, Konstantinos Ntaikos, "Design of a Secure Chat Application based on AES Cryptographic Algorithm and Key Management". The 10th WSEAS International Conference on Mathematical Methods, Computational Techniques and Intelligent Systems (MAMECTIS '08), Corfu 2008.

.