Design of a Secure Chat Application based on AES Cryptographic Algorithm and Key Management

NIKOLAOS G. BARDIS  
Adjunct Professor  
University of Military Education  
1Hellenic Army Academy, 2Hellenic Naval Academy, 3Hellenic Air Force Academy  
Department of Computer Sciences  
1Vari - 16673, 2Terma Hadjikyriakou Avenue, Piraeus - 18539, 3Dekelia Air Base, Tatoi, Metamorfosi 144 51, Greece

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

Abstract: - This article presents the design and implementation of a software application for the provision of secure real time communication services between workstations, based on the AES prototype cryptographic algorithm and an advanced secret key management system. The application has been designed based on the requirements of a military unit, so as to allow groups of authenticated users to communicate and read the transmitted messages. This application can be used as the basis for the design of an integrated communication system for a military organization. The present design confines its operation within the limits of a local area network, but the possibilities are open for operation in extended networks or the internet. In this article, a concise account of the design of the application is first presented. The way in which a symmetric encryption system uses a pair of secret keys to provide additional capabilities is hence presented. Consequently, the operation of a password management system is presented that achieves secure storage and handling of user secret passwords and access control data. Finally, the application implementation details are presented for a Visual Basic 6 implementation developed for a military unit.

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

1 Introduction

One of the most important factors that determine the efficiency and effectiveness of operation of a contemporary military or commercial organisation is its capability to securely store, retrieve and communicate information between authenticated users [2]. Information security is a primary goal for the armed forces and achieving information security may provide the required leading edge for an army. Consequently, information systems are continuously developed that aim at providing safe data storage and communication between working units of the organisation involved. 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 work is to design and develop a software application that provides secure real time communication based on symmetric cryptographic algorithms and a management system for handling, distributing, safely storing and retrieving user passwords. The ultimate aim of the application is to provide the infrastructure that will allow groups of authenticated users to read messages that they exchange in pairs.

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 Key of 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.

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.

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:

- 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.

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.

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 main window (form) that the user employs in order to compose or read messages is depicted in Figure 2 below. The form contains information about the user from whom the message is originating, connection information like the date and time established and its duration, as well as information about the local computer and user and controls for facilitating the communication. Further details about the actual implementation with simulation and test operation results, will be presented when the design will have reached a more mature stage.

5 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:


