Secure Communication Protocol for Mobile Multimedia Applications

N. Komninos[1], B. Honary[1], M. Darnell[2]

[1]Department of Communication Systems, Lancaster University, Lancaster LA1 4YR
Email: n.komninos@lancaster.ac.uk
[2]HW Communications Limited White Cross Industrial Estate.

***Abstract* – Data transmission over wireless communication networks is increasing rapidly and security has become a very important issue. The main considerations of secure communication systems are authentication, key distribution, and data transfer. This paper describes a secure communication protocol (SCP), which provides privacy and data integrity for end-to-end data transmission over wireless communication systems.**

***Keywords*: Security, Encryption, Protocols, Multimedia**

I. INTRODUCTION

As the use of electronic communications plays an ever-increasing role in business activities, security in communications through networking environments is becoming very important. It is very important to preserve the confidentiality, data integrity, and authenticity of sensitive information to prevent fraud. Therefore, a secure layer (SL) with the above characteristics is required.

The SL, with reference to the OSI (Open System Interconnection) model, can be built between the application and transport layers (Figure 1). SL is also required to be transparent and independent of the underlying layer (i.e. transport layer) to preserve the confidentiality and data integrity of sensitive information.

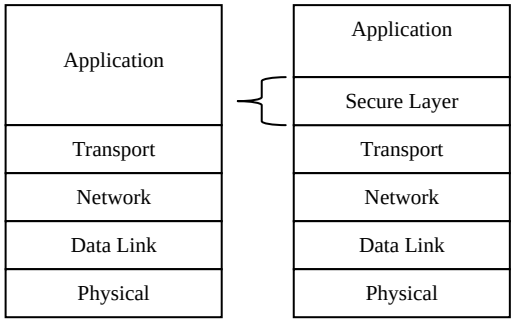

Figure 1. Secure Layer

Therefore, the secure communication protocol (SCP), which is used by SL, is responsible for providing all the necessary encryption mechanisms to the application program to protect sensitive information.

In cryptography all modern algorithms use a **key** to control encryption and decryption; an encrypted message can be decrypted only if the proper key is available to the recipient. The key used for decryption can be different from the encryption key, but for most algorithms they are the same.

There are two classes of key-based algorithms, **symmetric** (or **private-key**) and **asymmetric** (or **public-key**) algorithms. The difference is that symmetric algorithms use the same key for encryption and decryption (or the decryption key is easily derived from the encryption key), whereas asymmetric algorithms use a different key for encryption and decryption, and the decryption key cannot be derived from the encryption key. The **public-key algorithms** permit the encryption key to be published, allowing anyone to encrypt with that key, whereas only the proper recipient (who knows the decryption key) can decrypt the message. The encryption key is called the **public key** and the decryption key is called the **private key** [1].

Public key methods require in complex computations compared to secret key methods. However, since the public key can be distributed through an insecure channel, key management is easier in public key systems. In addition, the public key schemes can be used to generate digital signatures for authentication [1].

When encrypted data is transferred over a noisy mobile channel, there are two synchronisation cases to consider. The first is the initial synchronisation, where the receiver is initially synchronised to an incoming encrypted data stream. The second case is re-synchronisation, where the receiver maintains synchronisation to the incoming encrypted data stream after initial synchronisation [2].

A combination of public and private key schemes is supported by the secure protocol. Public key schemes are used for key distribution and private key schemes are used to encrypt general traffic. Furthermore, end-to-end encryption synchronisation is provided by the secure communication protocol.

II. SCP PROTOCOL OVERVIEW

The SCP protocol allows the transmitter and receiver to authenticate each other and to negotiate an encryption algorithm before data is transmitted. The SCP lays on the application layer and it is protocol independent to the underlying transport protocols in use (i.e. UDP, TCP).

The SCP protocol provides connection security that has three basic properties: privacy, authentication and data integrity. The connection is private because asymmetric key encryption is used to distribute a session key and data is encrypted using symmetric cryptography. The transmitter and receiver can be authenticated using symmetric, or public key, cryptography. One-way hash functions are used to check data integrity.

The protocol is composed by three phases: ***open session***, ***data session***, and ***close session***. Consider a situation where two stations A and B want to communicate securely (Figure 2). When stations A and B first start communicating, they select cryptographic algorithms, authenticate each other, and use public key encryption techniques to distribute keys. These processes can be summarised as follows: In the *open session* phase station A, sends a session request to which the station B must respond with a session response message. Following the session response, station A sends a session indication message to finalise the open session.

The session request, response, and indication, are used to establish security enhancement capabilities and key distribution between the station A and B. The session request, respond, and indication establish the following attributes: authentication of stations A and B, agreement of cipher suite, and generation of session key. Both stations A and B involve in the session key generation, and enable one party to check whether the other party posses the secret key corresponding to the public key it receives from the other party. Once the session key has been distributed secure data transfer can take place.

In the *data session* phase encrypted data is transferred. The protocol identifies the contents of the data (i.e. text, speech, or video) and passes the information to the application program. Encryption synchronization is also supported by the protocol to overcome synchronization problems that arise from dropped data frames due to channel errors [3]. Synchronization frames are included in the data session phase. In addition, data integrity is performed using hash functions.

Finally, the application program that receives the first *close session* request, sends a close session response to acknowledge the close of session. The application program uses the close session phase to shut down a connection when it finishes using it. The same operation is used to abort a session, in which case data transfer in both directions ceases immediately, and resources such as buffers are released.

II.1 *Open Session*

When station A first connects to station B it is required to send the open session request as its first message. The open session request includes a **certificate**, and a **cryptographic algorithm identifier** message. Authentication is done with the use of certificates. A certification authority (CA), which is assumed to be trusted by all users, is responsible for providing these certificates [4]. As shown in Table 1, the *certificate message* contains seven fields; the start delimiter (SD), the signature algorithm (SA), the signer (SG), the validity (VAL), the user identity (UID), the certificate (CER), and the public key (PK).

Certificate Message (Request/Response)	
SD	A start delimiter is used as a frame marker. SD is a fixed 8 bit number
SA	The algorithm used to sign the certificate (3 or 4 bits)
SG	The certification authority that signed the certificate (3 or 4 bits)
VAL	The validity period of the certificate (32 bits)
UID	User identity (32 bits)
CER	Certificate (The size depends on the certificate standards)
PK	Public Key of the certificate authority (1024 or 2048 bits)

Table 1. Certificate Message

In the certificate and cryptographic algorithm identifier (Table 2) messages, SD is mainly used as a frame marker and is an 8-bit fixed number agreed by the certification authority. The SA field indicates which algorithm is used, if multiple algorithms are supported, to sign the certificate. SA can be either 3-bits or 4-bits length depending on the number of the cryptographic algorithms supported by the application program. The SG field defines which certification authority signed the certificate. Similarly, SG can be either 3-bits or 4-bits length depending on the number of the certification authorities. The, VAL is used to indicate the validity period of the certificate. The validity period is a 32-bit number expressed in seconds. UID is a unique 32-bit number that identifies the user to the application program. The CER field contains the certificate itself. Finally, the PK field contains the public key of the certification authority used to validate the certificate.

The *cryptographic algorithm message* immediately follows after the certificate message and contains six fields (Table 2);

Cryptographic Algorithm Identifier Message (Request)	
SD	The start delimiter is used as a frame marker and is a fixed 8 bit number
PKA	If multiple public key algorithms are supported, PKA is the number of the algorithm used to encrypt session keys (8 bits)
SMA	The number of the symmetric algorithm used to encrypt traffic (4 bits)
MCA	Mode of the symmetric algorithm (4 bits)
IV	IV is used to initially synchronised the receiver to an incoming encrypted data stream (128 bits)
PK	Public key of the user (1024 or 2048 bits)
TST	The timestamp used to give limit lifetime to the transmission (32 bits - UNIX format)

Table 2. Cryptographic Algorithm Identifier Message

the public key algorithm (PKA), the symmetric algorithm (SMA), the mode of cryptographic algorithm (MCA), the initialisation vector (IV), the public key of the user (PK) and the timestamp (TST).

The PKA defines which cipher is used, if multiple algorithms are supported, to secure the distribution of the session key, expressed in 8-bit format (i.e. RSA=01100110). Likewise, the SMA indicates which the cipher is used to secure traffic between the two applications. SMA is expressed in 4-bit format (i.e.

SAFER=0101, etc). Furthermore, MCA defines the mode of the cipher, expressed in 4-bit format, that decreases the processing power and time of the application (i.e. ECB=0011, CBC=0101, etc). The application program performs faster by knowing the number of bits to encrypt or decrypt in a block or a stream cipher. Moreover, the IV field contains the initialised input of the block algorithms. The IV used in the algorithms need not be secret and can be transferred in clear. Finally, the TST is used to give a limited lifetime of transmission to prevent replay attacks. The current time and date in standard UNIX 32-bit format (seconds since the midnight starting Jan 1, 1970, GMT) according to the sender's internal clock.

When station B receives the open session request it will first check the timestamp and then authenticate station A. Then, it will agree on the cryptographic algorithms and send an open session response. The open session response includes a **certificate**, and a **cryptographic algorithm** message. The certificate message includes the same fields with the one in the open session request (Table 1). The cryptographic algorithm message, on the other hand, contains one extra field (Table 3); the random bytes (RB). The random bytes (RB) field contains encrypted random bits and is used to certify whether station A and B posses the proper private keys. The RB field is also used for the generation of the session key, which encrypts traffic between the two applications.

Cryptographic Algorithm Identifier Message (Response)	
Same as cryptographic algorithm message request (Table 2)	
RB	Encrypted random bits used to generate one of the parts of the session key and to authenticate parties

Table 3. Cryptographic Algorithm Message Response

Finally, station A sends an open session indication message, which verifies the agreed public and key algorithm. The open session indication message includes also the second part of the random bits used for the generation of the session key. The random bits are encrypted using the public key of station B. Applications can now hash the random bits to create the session key.

II.2 *Data Session*

In the data session phase, the transmitted data is encrypted using the agreed cryptographic algorithm. The **data session message** includes two identifiers used to determine the contents of data and the type of messages transmitted as shown in Table 4.

Data Session Message	
SD	SD is used as a frame marker and also as a session identifier (8 bits)
MTI	Indicates the type of the message (4bits)
FTI	Indicates the contents of the data transmitted (3 bits)
SEQ	SEQ is used to identify lost or duplicated messages (3 bits)
LEN	The number of bytes received (32 bits)
DAT	Bytes Data
DIC	A hash value (64 or 128 bits) of the message received that acts as a CRC
ED	The end delimiter is used for synchronisation purposes and has the same value with SD (8 bits)

Table 4. Data Session Message

The message type identifier (MTI) indicates the type of message carried in the network, expressed in a 4-bit format (i.e. Filename = 0001, Data = 0010, Last Frame = 0100, and Synchronization Frame = 1000). The MTI field gives information to the upper layers to process the data received. Likewise, the file type identifier (FTI), which may be 3-bits such as '001' for text, is used to indicate the type of data carried in a frame (i.e. **T**ext, **S**peech, or **V**ideo).

A start delimiter (SD) is also used to identify the beginning of the message. The value of SD, expressed in 8-bit format, is different in every session in order to avoid an unauthorized user to listen to the session. Similarly, the end delimiter (ED) identifies the end of the data message and mainly used for synchronisation purposes. The receiver is able to re-synchronise by checking the ED for out of sequence entry of a message. ED is the same 8-bit value as SD. The sequence (SEQ) number is mainly used to identify lost or duplicated messages.

SEQ also enables the transmitter to send synchronisation messages after a fixed number of data messages. Furthermore, the length (LEN) field defines the length of the data. LEN is mainly used by the receiver to identify the stream of data received. Moreover, the data (DAT) field contains the actual information, which is encrypted with the agreed symmetric algorithm. Finally, a data integrity check (DIC) is used to indicate if the data received is correct. The hash value of the data session message is included in the DIC field. Thus, the receiver hashes the data retrieved to check the contents of the data session message.

Once the encrypted data is transmitted over a noisy channel, the receiver needs to maintain synchronisation to the incoming data stream after initial synchronisation.

Encryption synchronization is achieved by periodically sending synchronization information to the receiver. This is particularly required by continuous media such as speech or video. Synchronization messages can be transmitted using the data session message format. Therefore, the MTI field defines a synchronisation message and the FTI field is set to the agreed SMA. Furthermore, the DAT field contains the

current IV to update the IV of the receiver and keep the key stream synchronised to the incoming encrypted data stream.

II.3 *Close Session*

In the close session phase the application program that receives the first close session request will then send a close session response to acknowledge the closure. The close session request and respond messages include a SD, and a session release flag (SRF).

Close Session Message (Request/Response)	
SD	SD is used as a session identifier (8 bits)
SRF	SRF is a close session identifier for a request or response message

Table 5. Close Session Message

SD indicates the session ID to close and SRF defines if the message is a close session request or response. The 8-bit value of SRF may be '11110000' or '00001111' for a close session request or response message respectively.

III. CONCLUSION

In this paper a secure communication protocol for mobile multimedia applications was presented. The SCP is an independent secure layer found between the application and transport layers. The transport protocols can be either connectionless or connection oriented. The connection state, in other words, is not affected by the lost or out of order datagrams. The application protocols retrieve information from SCP for the type of media transferred, and the cryptographic algorithms used to encrypt data.

SCP has the following novel properties. Random bits are passed to the application program to generate the session key. Thus, both transmitter and receiver are involved in the generation of the session key. Furthermore, the encrypted random bits exchanged, help the application program to check whether both parties possess the correct private keys. Moreover, the protocol identifies the mode of the cryptographic algorithms (i.e. ECB, OFB etc.) which decreases the processing power and time used by the application. In addition, SCP has data link protocol characteristics such as the message type identifier, which guides the application to download a multimedia file correctly. Finally, the start and end delimiter used in the data session phase act as frame markers and identifiers, which change in every new session. Thus, SD and ED prevent eavesdropping from taking place.

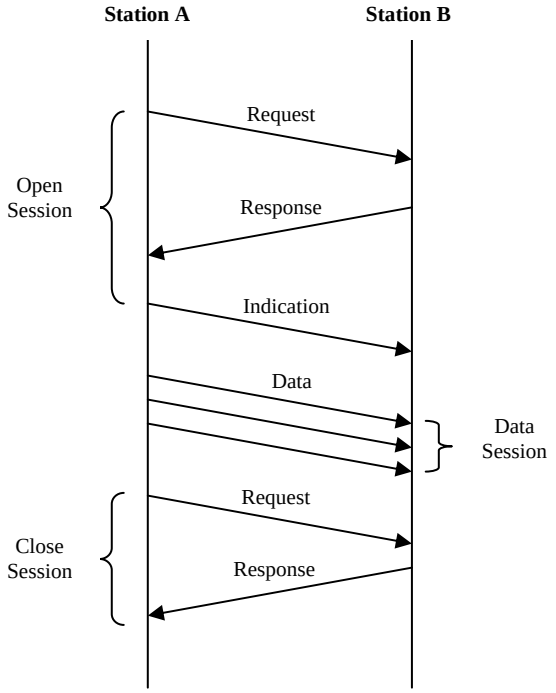

Figure 2. Communication between Station A and B

IV. REFERENCES

[1] B. Schneier, *Applied Cryptography*, Published by John Wiley & Sons Inc., 1996.

[2] M. I. Samarakoon, B. Honary, M. Rayne "End to End Encryption Synchronization Method for Digital Private Mobile Radio Systems", *Fifth International Symposium on Communications Theory and Applications*, Ambleside 1999.

[3] D. Brown, "Techniques for Privacy and Authentication in Personal Communication Systems", IEEE Personal Communications, August 1995.

[4] N. Komninos, M. I. Samarakoon, B. Honary "Authentication and Key Distribution Protocols for Wired and Wireless Systems", *1st Annual Postgraduate Symposium on the Convergence of Telecommunications, Networking and Broadcasting*, June 2000.

[5] A. Tanenbaum, *Computer Networks*, 3rd Edition, Prentice Hall, 1996.