The Use of Plain Text Keys for Encryption of Multimedia Conferences

Draft V1.4, February 27, 1995

Edited by Mark Handley,
Department of Computer Science,
University College London.

Introduction

For reasons of privacy, multimedia conferencing tools are starting to incorporate encryption schemes. When the tools are used for one to many communication using multicast, such encryption schemes must use secret key encryption algorithms such as DES. Keys for the encryption scheme are distributed out of band. Many such key distribution schemes can be envisaged, such as PEM (Privacy Enhanced Mail), secure WWW, and additions to the existing conference control and conference setup schemes. These distribution schemes are not discussed here.

It is important that all compatible tools given a key use it in the same way. It is also desirable that such keys can be plain text, as this aids key distribution in cases where the key must be entered manually, or must be remembered. However, plain text should not be used directly as an encryption key for algorithms such as DES, as the entropy in the key is unevenly distributed amongst the bits of the key, which reduces the potential key space, and thus greatly weakens the encryption scheme.

Another simple but important aspect is that many tools may support multiple encryption schemes. It is important that the encryption scheme being used is specified in a standard way. Distributing this information along with the plain text key simplifies the distribution process.

Specifying the Choice of Encryption Scheme

Keys may consist of any 7 bit printable ASCII text characters (ie., characters with ASCII codes from 32 to 126 inclusive). The restriction to 7 bit printable ASCII is so that an encryption key can be entered on almost any keyboard. We divide the key information into two parts - the encryption scheme, and the encryption key. Any characters up to the first slash ('/') will be taken as the name of an encryption scheme.Clearly encryption format specifiers themselves cannot contain slashes. If no slash is present, DES encryption is assumed as the default and the entire given string is used as the plain text encryption key.

If a key is given which contains one or more slashes, but the string up to the first slash does not match any available encryption scheme, DES is assumed as the encryption scheme, and the entire key including the characters up to the first slash are taken as the encryption key. In such a case, presenting a warning to the user is advisable.

Initially, two encryption format specifier strings are defined:

> **`DES1`** - The DES encryption algorithm in Cipher Block Chaining mode (see [1] and [2])
> **`DES3`** - Triple DES encryption (see [3])
> **`DULL`** - A test "encryption scheme" used for testing of key exchange mechanisms (see Appendix 1).

For example, the following plain text key:

```
DES1/this is a key
```

specifies the DES algorithm and the plain text key string "this is a key"

This is identical in meaning to the plain text key:

```
this is a key
```

which also specifies the plain text key string "this is a key", and chooses DES encryption by default.

Choosing the key:

```
this/is/a/key
```

would specify the key "this/is/a/key" and DES encryption is there was no encryption scheme called "this", and would normally be accompanied by a warning that the encryption scheme "this" is not available, and that DES was being used.

Encryption format specifiers are case sensitive. This is both to reduce the number of keys unavailable (due to starting with strings such as "des1/") and more importantly to serve as a check that the user understands that the encrption key itself is case sensitive. As the application cannot check whether the user understands this, the warning that the specified encryption format is unavailable can also serve to some extent as a warning that keys themselves are case sensitive.

Although no limit as to the size of keys or encryption format specifiers is strictly necessary, implementations are simplyfied if maximum sizes are specified. We specify that encryption format specifiers can be between one and seven characters, with at least three characters recommended. Encryption keys can be between one and 56 characters. Thus a full encyption scheme and encryption key can be represented in 64 characters including the slash ("/") separator.

Entropy Distribution in Encryption Keys

As mentioned above, directly using a plain text string as the key for an encryption scheme has a number of problems. The entropy distribution on a plain text string is non-uniform. A desirable property of an encryption key is uniform entropy distribution. None uniform entropy distribution in an encryption key can make the encryption scheme less secure.

Another desirable property is that all of the plain text key is used by the encryption scheme. An example is DES, which requires a 56 bit key. If a plain text key were used directly, only the first 7 ASCII characters of the key would be used, and those 56 bits would come from a very small subset of the possible combinations those 56 bits could comprise.

To address both these problems, an MD5 checksum is calculated over the plain text key to generate the encryption key that is actually used for encryption. MD5 is described in [3]. MD5 performs a good job of distributing the entropy present in the plain text key throughout the 128 bits of the resulting checksum.

The algorithm used is as follows:

If l_p is the number of bits in the original plain text key, s_p.

1 Let l_k be the first multiple of 128 exceeding l_p, i.e. $l_k = (l_p+127)/128$ (integer division, rounding down)
2 Append s_p to the end of s_p until the length of the resulting string is greater than or equal to l_k. We call this resulting string s_k
3 Truncate s_k to l_k bits (MD5 requires a multiple of 128 bits)
4 Calculate an MD5 checksum over this data. We call the resulting checksum c_k.
5 Use the leftmost bits of c_k to comprise the encryption key to be used. For example DES would use the first 56 bits (first 7 bytes) of the checksum string.

For example, if the plain text key is the string `"foobar!"`, then l_k is 128 and s_k=`"foobar!foobar!fo"`

If the encryption scheme requires more than one encryption key, then subsequent keys take the next leftmost bits. For example DES3 requires two 56 bit keys. The first key should take bytes 1-7, and the second key bytes 8-14.

If an encryption scheme requires more than 128 bits of key, this scheme should would require modification.

DES Keys

A DES encryption key is 56 bits long, and as stated above, when a plain text key is used for multimedia conferencing, the first seven bytes of the output of the MD5 checksum (as calculated above) are used. However, the DES specification also states that when keys have to be entered manually a 64 bit key is entered, of which the most significant bit of each byte is used as a parity bit. DES then reduces the 64 bit key to a 56 bit key by checking and discarding the partity bits. It is clear that these parity bits are of no utility when a plain text key is entered and pre-processed through MD5 as specified above.

If a DES implementation is used which requires 64 bit keys, the first 56 bits of the MD5 checksum will have to be expanded to a 64 bit key by inserting partity bits. However, in general this stage is superfluous as most DES implementations should permit a 56 bit key with no parity bits to be used directly.

Use of Encryption with RTP/RTCP

RTP (see [5]) has no explicit place for encryption data to be inserted. Instead, the entire data packet, including the RTP header should be encrypted. This allows us to perform a header validation check after decryption, and if this check fails, the packet should simply be discarded. When RTP is encapsulated in UDP, clearly the UDP header is not encrypted.

DES in Cipher Block Chaining mode benefits from encrypting the RTP headers, as the sequence number and timestamp in the first 64 bits serve as a randomisation field. Without a randomisation field, fixed format headers would make DES succeptable to plain text attack.

RTCP (see [5]) packets should also be encrypted, but directly encrypting their fixed format would weaken the encryption scheme. To prevent this, 4 random bytes should be added to the start of the RTCP packet header before encryption. These bytes should be discarded upon decryption. No such random bytes are added to RTP packets.

Cipher Block Chained DES normally requires both a key and an initialisation vector (IV). The initialisation vector is to prevent attacks on the first block of a data stream due to its known format. When encrypting RTP packets as described below, a fixed IV of zero is used because either the packets contain sufficiently random data in the first

data block in the case of RTP, or have 32 random bits prepended to them before encryption in the case of RTCP.

Padding RTP & RTCP Packets for Encryption

Many encryption schemes require the encryption of data in fixed size blocks. For example, DES requires data to be in 64 bit blocks. If the data to be encrypted does not end on a 64 bit boundary, then the data requires padding to bring the data up to a 64 bit boundary.

RTP recognises this and has a padding "P" bit in the RTP and RTCP headers. If this bit is set, the payload has been padded to a 64 bit boundary. The last byte in the payload then states the number of padding bytes that must be removed before the original payload is recovered.

DULL "Encryption" Scheme

The DULL encryption scheme is intended only for test purposes - it does not provide a reasonable level of security. However for countries where legal restrictions prevent the use or export of encryption technology, it may be useful to provide a very limited level of privacy to prevent accidental eavesdropping.

After generating an MD5 checksum, as specified above, the first 32 bits of the checksum are taken as the encryption key. Each 32 bits of the source data are then XORed with the encryption key to produce the corresponding 32 bits of encrypted data. If the packet does not finish on a 32 bit boundary, it is padded to a 32 bit boundary as described above before "encryption".

References

[1] National Institute of Standards and Technology (NIST), "*FIPS Publication 46-1: Data Encryption Standard"*, January 22, 1988

[2] National Institute of Standards and Technology (NIST), "*FIPS Publication 81: DES Modes of Operation"*, December 2, 1980

[3] RSA, "*Frequently Asked Questions about Today's Cryptography - DES"*, http://www.rsa.com/faq/faq_des.html, May 1994

[4] Rivest, R., "*The MD5 Message-Digest Algorithm*", RFC 1321, MIT Laboratory for Computer Science and RSA Data Security, Inc., April 1992

[5] Schulzrinne, H., Casner, S., Frederick, R., Jacobson, V., "*RTP: A Transport Protocol for Real Time Applications"*, IETF Internet Draft draft-ietf-avt-rtp-06.txt, November 1994 (working draft)