APPENDIX

5

Encryption Services

What Are Encryption Services? **207**
Terminology **207**
System and Software Requirements **207**
Requirements for SAS Proprietary Encryption Services **208**
Communications Access Methods Support **208**
North American and International Encryption Services Packages **208**
Data Encryption Algorithms **209**
SAS System Options **210**
SAS/CONNECT Example **211**
SAS/CONNECT Local Host **211**
SAS/CONNECT Remote Host **212**
SAS/SHARE Example **212**
SAS/SHARE Client **212**
SAS/SHARE Server **212**

What Are Encryption Services?

Encryption services protect data that is sent between hosts across a network. Encryption services use a reversible algorithm to convert plain-text data into an unintelligible form, thus protecting data from being used by unauthorized parties.

Terminology

This appendix addresses encryption services for both SAS/CONNECT and SAS/SHARE software. The terms *local host* and *remote host* are used to designate local and remote sides for SAS/CONNECT and *client* and *server* sides for SAS/SHARE.

System and Software Requirements

You must purchase a license for SAS/SECURE in order to use the encryption services of these products:

- RSA BSAFE Crypto-C Toolkit
- New in Version 8, RSA BSAFE Crypto-J Toolkit, which supports Java clients that access SAS servers.

The RSA BSAFE Crypto-J Toolkit is supported on these platforms:

- OpenVMS Alpha
- OpenVMS VAX
- OS/2
- AIX
- Compaq Tru64 UNIX (formerly Compaq's DIGITAL UNIX)
- HP-UX
- Solaris 2
- OS/390.

The RSA BSAFE Crypto-J Toolkit runs on any platform that has installed the Java Development Kit, Version 1.1 or later.

CryptoAPI from Microsoft is an application programming interface that provides access to the cryptographic services that are provided by:

- Windows 95 and Windows 98 (as part of Internet Explorer 3.0+)
- Windows NT 4.0+ (as part of the operating system). Service Pack 3 or a subsequent release must be installed.

You must have either of the following packages installed on your Windows host to use CryptoAPI:

- Microsoft Base Cryptographic Service Provider
- Microsoft Enhanced Cryptographic Service Provider.

Requirements for SAS Proprietary Encryption Services

You can use the SAS Proprietary encryption services on all platforms. Encryption services provided by SAS are free of charge and require no additional software license.

Communications Access Methods Support

Encryption services provided by the RSA BSAFE Crypto-C Toolkit and the Microsoft CryptoAPI are available with the following communications access methods on the supported hosts:

- TCP/IP
- DECnet
- NetBIOS.

The RSA BSAFE Crypto-J Toolkit supports the TCP/IP access method only.

For example, you can use encryption services to connect two UNIX hosts when using the TCP/IP access method. Also, you can connect two Windows hosts by using either the TCP/IP, the DECnet, or the NetBIOS access method. See *Communications Access Methods for SAS/CONNECT and SAS/SHARE Software* for a definitive list of supported host connections by access method.

North American and International Encryption Services Packages

United States export regulations on encryption software restrict access to SAS/SECURE software and related technical data as follows:

1. The SAS/SECURE Domestic version may be used or accessed only within the United States or Canada by citizens or lawfully admitted permanent residents of those countries or as otherwise permitted by special export license.
2. The SAS/SECURE International and the SAS/SECURE for Windows versions may not be exported to terrorist supporting or embargoed destinations or parties.

In addition to export regulations, SAS software licensing documents may limit countries of use.

Because of these export key-length restrictions, encryption services are packaged in the following forms:

North American (U. S. and Canada)
: available to North American customers and certain other customers by special export license, supports 1024-bit or 512-bit RSA keys in combination with the following algorithms:

- RC2 using 128-bit or 40-bit keys
- RC4 using 128-bit or 40-bit keys
- DES using 56-bit keys
- Triple DES using 168-bit keys

International
: available to International customers, supports two types of encryption:

- 512-bit RSA keys in combination with the 40-bit key algorithms RC2 and RC4
- new in Version 8, the 1024-bit RSA keys in combination with the 56-bit DES algorithm.

Data Encryption Algorithms

The encryption algorithms as well as the SAS Proprietary algorithm are defined as follows:

RC2
: A proprietary algorithm developed by RSA Data Security, Inc., RC2 is an alternative to DES. The algorithm expands a single message by up to 8 bytes. RC2 is a block cipher that encrypts data in blocks of 64 bits. The size of the output of the algorithm is always a multiple of the block size. The RC2 key size can range from 8 to 256 bits.

RC4
: A proprietary algorithm developed by RSA Data Security, Inc., RC4 is a stream cipher. A stream cipher encrypts one byte at a time. The RC4 key size can range from 8 to 2048 bits.

Note: The term *cipher* means encryption algorithm. △

DES
: An acronym for Data Encryption Standard, DES was developed by IBM. The algorithm expands a single message by up to 8 bytes. DES is a block cipher that encrypts data in blocks of 64 bits by using a 56-bit key.

Triple DES
: Triple DES executes DES three times on the data in order to exploit a key size that is three times that of DES. The algorithm expands a single message by up to 8 bytes. DES is a block cipher that encrypts data in blocks of 64 bits.

SAS Proprietary
This provides basic encryption services on all platforms and requires no additional product licenses. The algorithm expands a single message by approximately one-third. It uses a 32-bit key.

The key sizes that are used are based on the encryption software that is available on your host and the value that is assigned to the NETENCRKEYLEN option (see the next section).

SAS System Options

Note: These options do not apply to hosts that use the RSA BSAFE Crypto-J Toolkit. For Java client options, see other documentation (such as the documentation about the SAS/CONNECT Driver for Java that is provided with SAS/IntrNet software). △

For hosts that use the RSA BSAFE Crypto-C Toolkit or the Microsoft CryptoAPI, here are the SAS options that set encryption services attributes.

NETENCRYPT=YES|NO
or
NETENCRYPT | NONETENCRYPT
Set this option at both the local and remote hosts. At the remote host, this option specifies that encryption is required for each connection from a local host SAS session. At the local side, this option specifies that the local host must connect only to a remote host that supports encryption.

By default, encryption is used if the NETENCRYPTALGORITHM= option is set and if both the local and remote sides are capable of encryption. If encryption algorithms were specified but either the local or the remote side is incapable of encryption, then encryption is not performed.

Encryption may not be supported at the local or at the remote host for these reasons:

- You are running a release of SAS (prior to Version 7) that does not support encryption.
- Your site has not purchased a SAS/SECURE license for a specific platform.
- You specified incompatible encryption algorithms in the local and the remote host SAS sessions.
- You do not have a cryptographic service provider installed.

NETENCRYPTALGORITHM=("*algorithm1*",
"*algorithm2*", ...)
If you specify more than one algorithm, enclose the algorithm names in parenthesis and use commas to separate them. If there are embedded blanks in the algorithm name, enclose each algorithm with quotation marks.

The alias is NETENCRALG.

Set this option at the remote host and, optionally, at the local host to specify one or more encryption algorithms to use in a SAS session. However, the local and remote hosts must share an encryption algorithm in common. If you specify the option in the remote host session only, the local side attempts to select an algorithm that was specified at the remote host. If you also set the option at the

local host and specify an algorithm that is not specified at the remote host, the attempt by the local host to connect to that remote host fails.

Valid values for this option are

RC2

RC4

DES

TripleDES

SAS Proprietary

NETENCRYPTKEYLEN=*n*

Set this option in either the local or the remote host SAS session. It specifies the key length to be used by the encryption algorithm.

The alias is NETENCRKEY.

Valid values for this option are

128	specifies 1024-bit RSA and 128-bit RC2 and RC4 key algorithms.
40	specifies 512-bit RSA and 40-bit RC2 and RC4 key algorithms.
0	no value is set. This is the default.

If you require extra security, set NETENCRYPTKEYLEN=128. If you want to save CPU, set NETENCRYPTKEYLEN=40.

By default, if you try to connect to a host that is capable of only a 40-bit key algorithm with a host that is capable of both 40- bit and 128-bit, the connection using the lesser of the two key lengths is used. If both hosts are capable of 128-bit, then the 128-bit is used. To explicitly set one or the other, set the NETENCRYPTKEYLEN SAS option.

NETMAC | NONETMAC

This option controls the use of Message Authentication Codes (MACs) on network communications. A Message Authentication Code is the equivalent of a checksum that is used to ensure that the original message has not been modified. The MAC integrity checking adds an extra 16 bytes to RC4 encrypted messages and an extra 24 bytes to RC2, DES, and TripleDES encrypted messages.

You set this option at either the local or the remote host. The default is NETMAC.

SAS/CONNECT Example

SAS/CONNECT Local Host

Specify the following statements in a local host session:

```
options netencryptalgorithm=rc4;
options remote=unxnode comamid=tcp;
signon;
```

The NETENCRYPTALGORITHM= option specifies that the RC4 algorithm be used for encryption in the local host session.

SAS/CONNECT Remote Host

The following example illustrates the content of the executable file that a UNIX spawner program uses to start SAS and to specify encryption in a SAS/CONNECT remote host session:

```
#_____________________
# mystartup
# _____________________
#!/bin/ksh
. ~/.profile
sas dmr -noterminal -no$syntaxcheck -comamid tcp -netencryptalgorithm rc4 -nete
```

The NETENCRYPTALGORITHM= option specifies that the RC4 algorithm be used for encryption of all data that is exchanged with connecting local hosts. The NETENCRYPT option specifies that encryption is required by any local host that accesses this remote host.

SAS/SHARE Example

SAS/SHARE Client

Specify the following statements in a client session:

```
options netencryptalgorithm=rc4;
options comamid=tcp;
libname sasdata 'edc.prog2.sasdata' server=rmthost.share1;
```

The NETENCRYPTALGORITHM= option specifies that the RC4 algorithm be used for encryption in the client session.

SAS/SHARE Server

Specify the following statements in a SAS/SHARE server session:

```
options netencrypt netencryptalgorithm=rc4;
options comamid=tcp;
proc server id=share1;
run;
```

The NETENCRYPT option specifies that encryption is required by any client that accesses this server. The NETENCRYPTALGORITHM= option specifies that the RC4 algorithm be used for encryption of all data that is exchanged with connecting clients.

The correct bibliographic citation for this manual is as follows: SAS Institute Inc., *SAS/SHARE User's Guide, Version 8*, Cary, NC: SAS Institute Inc., 1999. pp. 247.

SAS/SHARE User's Guide, Version 8

ISBN 1–58025–478–0

SAS Institute Inc., SAS Campus Drive, Cary, North Carolina 27513.

1st printing, September 1999