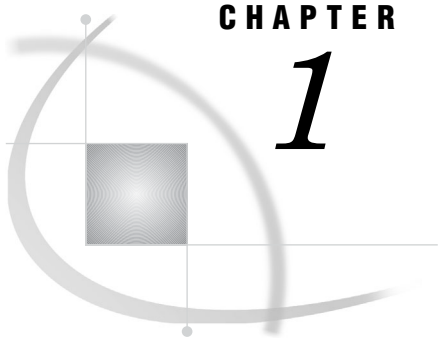


CHAPTER

1

Technologies for Encryption

Encryption: Overview

There is a great need to ensure the confidentiality of business transactions over a network between an enterprise and its consumers, between enterprises, and within an enterprise. SAS products and third-party strategies for protecting data and credentials (user IDs and passwords) are exchanged in a networked environment. This process of protecting data is called *encryption*. Encryption is the transformation of intelligible data (plaintext) into an unintelligible form (ciphertext) by means of a mathematical process. The ciphertext is translated back to plaintext when the appropriate key that is necessary for decrypting (unlocking) the ciphertext is applied.

SAS offers two classes of encryption strength:

- If you don't have SAS/SECURE, only the SASProprietary algorithm is available. SASProprietary uses 32-bit fixed encoding and is appropriate only for preventing accidental exposure of information. SASProprietary is licensed with Base SAS software and is available in all deployments.
- If you have SAS/SECURE, you can use an industry standard encryption algorithm instead of the SASProprietary algorithm. SAS/SECURE is an add-on product that is licensed separately.

Encryption helps to protect information on-disk and in-transit as follows:

- *Over-the-wire* encryption protects SAS data and data while in transit. Passwords in transit to and from SAS servers are encrypted or encoded.
- *On-disk* encryption protects data at rest. Passwords in configuration files and the metadata are encrypted or encoded. Configuration files and metadata repository data sets are also host protected.

Providers of Encryption

SASProprietary

SASProprietary Overview

SASProprietary is a fixed encoding algorithm that is included with Base SAS software. It requires no additional SAS product licenses. The SAS proprietary algorithm is strong enough to protect your data from casual viewing. SASProprietary provides a medium level of security. SAS/SECURE and SSL provide a high level of security.

SASProprietary System Requirements

SAS 9.2 supports SASProprietary under these operating environments:

- OpenVMS
- UNIX
- Windows
- z/OS

SASProprietary Installation and Configuration

SASProprietary is part of Base SAS. Separate installation is not required.

For an example of configuring and using SASProprietary in your environment, see "SASProprietary for SAS/SHARE: Example" on page 38.

SAS/SECURE

SAS/SECURE Overview

SAS/SECURE software is an add-on product that provides industry standard encryption capabilities in addition to the SASProprietary algorithm. SAS/SECURE requires a license, and it must be installed on each computer that runs a Foundation SAS client and a server that will use the encryption algorithms.

Note: SAS/SECURE provides encryption of data in transit. It does not provide authentication or authorization capabilities. △

SAS/SECURE System Requirements

SAS 9.2 supports SAS/SECURE under these operating environments:

- UNIX
- Windows
- z/OS

Export Restrictions for SAS/SECURE

For software licensing and delivery purposes, SAS/SECURE is the product within the SAS System. For U.S. export licensing purposes, SAS designates each product based on the encryption algorithms and the product's functional capability. SAS/SECURE 9.2 is available to most commercial and government users inside and outside the U.S. However, some countries (for example, Russia, China, and France) have import restrictions on products that contain encryption, and the U.S. prohibits the export of encryption software to specific embargoed or restricted destinations.

SAS/SECURE for UNIX and z/OS includes the following encryption algorithms:

- RC2 using up to 128-bit keys
- RC4 using up to 128-bit keys
- DES using up to 56-bit keys
- TripleDES using up to 168-bit keys
- AES using 256-bit keys

SAS/SECURE for Windows uses the encryption algorithms that are available in Microsoft CryptoAPI. The level of the SAS/SECURE encryption algorithms under Windows depends on the level of the encryption support in Microsoft CryptoAPI under Windows.

SAS/SECURE Installation and Configuration

SAS/SECURE must be installed on the SAS server computer, the client computer, and possibly other computers, depending on the SAS software that requires encryption. For installation details, see the SAS documentation for the software that uses encryption.

For examples of configuring and using SAS/SECURE in your environment, see Chapter 4, "Encryption Technologies: Examples," on page 37.

Secure Sockets Layer (SSL)

Secure Sockets Layer (SSL) Overview

SSL is an abbreviation for Secure Sockets Layer, which is a protocol that provides network data privacy, data integrity, and authentication. Developed by Netscape Communications, SSL uses encryption algorithms that include RC2, RC4, DES, TripleDES, AES and others.

In addition to providing encryption services, SSL performs client and server authentication, and it uses message authentication codes to ensure data integrity. SSL is supported by Netscape Navigator, Internet Explorer, and Mozilla Firefox. Many Web sites use the protocol to protect confidential user information, such as credit card numbers. The SSL protocol is application independent and allows protocols such as HTTP, FTP, and Telnet to be transparently layered above it. SSL is optimized for HTTP. SSL includes software that was developed by the OpenSSL Project for use in the OpenSSL Toolkit. For more information see **`www.OpenSSL.org`**.

Note: Transport Layer Security (TLS) is the successor to SSL V3.0. The Internet Engineering Task Force (IETF) took SSL V3.0, which was the *de facto* standard, modified it, renamed it TLS V1.0, and adopted it as a standard. △

SSL System Requirements

SAS 9 and later releases support SSL V2.0, SSL V3.0 and TLS V1.0.
SAS 9.2 supports SSL under these operating environments:

- UNIX
- Windows
- z/OS (new for SAS 9.2)
- OpenVMS

Note: The SAS/SECURE SSL software is included in the SAS installation software only for countries that allow the importation of encryption software. △

SSL Concepts

The following concepts are fundamental to understanding SSL:

Certification Authorities (CAs)
: Cryptography products provide security services by using digital certificates, public-key cryptography, private-key cryptography, and digital signatures. Certification authorities (CAs) create and maintain digital certificates, which also help preserve confidentiality.

  Various commercial CAs, such as VeriSign and Thawte, provide competitive services for the e-commerce market. You can also develop your own CA by using products from companies such as RSA Security and Microsoft or from the Open Source Toolkit OpenSSL.

  Note: z/OS provides the PACDCERT command and PKI Services for implementing a CA. △

From a trusted CA, members of an enterprise can obtain digital certificates to facilitate their e-business needs. The CA provides a variety of ongoing services to the business client that include handling digital certificate requests, issuing digital certificates, and revoking digital certificates.

Public and Private Keys
: Public-key cryptography uses a public and a private key pair. The public key can be known by anyone, so anyone can send a confidential message. The private key is confidential and known only to the owner of the key pair, so only the owner can read the encrypted message. The public key is used primarily for encryption, but it can also be used to verify digital signatures. The private key is used primarily for decryption, but it can also be used to generate a digital signature.

Digital Signatures
: A digital signature affixed to an electronic document or to a network data packet is like a personal signature that concludes a hand-written letter or that validates a credit card transaction. Digital signatures are a safeguard against fraud. A unique digital signature results from using a private key to encrypt a message digest. Receipt of a document that contains a digital signature enables the receiver to verify the source of the document. Electronic documents can be verified if you know where the document came from, who sent it, and when it was sent. Another form of verification comes from Message Authentication Codes (MAC), which ensure that a document has not been changed since it was signed. A MAC is attached to a document to indicate the document's authenticity. Receipt of the document that contains a MAC enables the receiver (who also has the secret key) to know that the document is authentic.

Digital Certificates
: Digital certificates are electronic documents that ensure the binding of a public key to an individual or an organization. Digital certificates provide protection from fraud.

  Usually, a digital certificate contains a public key, a user's name, and an expiration date. It also contains the name of the Certification Authority (CA) that issued the digital certificate and a digital signature that is generated by the CA. The CA's validation of an individual or an organization allows that individual or organization to be accepted at sites that trust the CA.

SSL Installation and Configuration

The instructions that you use to install and configure SSL at your site depend on whether you use UNIX, Windows, or z/OS. See the appropriate details:

- Appendix 1, "Installing and Configuring SSL under UNIX," on page 55
- Appendix 2, "Installing and Configuring SSL under Windows," on page 61
- Appendix 3, "Installing and Configuring SSL under z/OS," on page 67

For examples of configuring and using SSL in your environment, see Chapter 4, "Encryption Technologies: Examples," on page 37.

SSH (Secure Shell)

SSH (Secure Shell) Overview

SSH is an abbreviation for Secure Shell, which is a protocol that enables users to access a remote computer via a secure connection. SSH is available through various commercial products and as freeware. OpenSSH is a free version of the SSH protocol suite of network connectivity tools.

Although SAS software does not directly support SSH functionality, you can use the *tunneling* feature of SSH to enable data to flow between a SAS client and a SAS server. *Port forwarding* is another term for tunneling. The SSH client and SSH server act as agents between the SAS client and the SAS server, tunneling information via the SAS client's port to the SAS server's port.

SSH System Requirements

OpenSSH supports SSH protocol versions 1.3, 1.5, and 2.0.
SAS 9.2 supports SSH under these operating environments:

- UNIX
- Windows
- z/OS

For additional resources, see

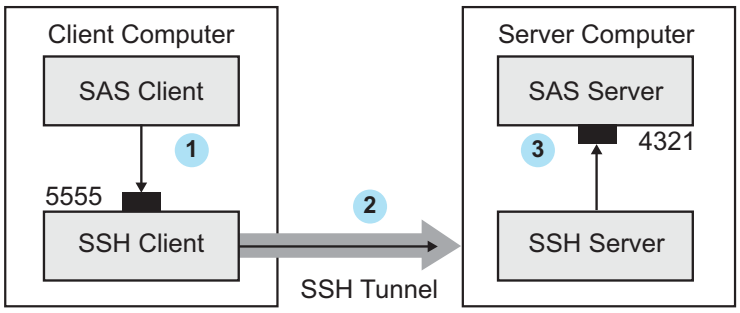
- **`www.openssh.com`**
- **`www.ssh.com`**
- ssh(1) UNIX manual page.

Under z/OS, the IBM Ported Tools for z/OS Program Product must be installed for OpenSSH support. See **`www-03.ibm.com/servers/eserver/zseries/zos/unix/port_tools.html`**.

SSH Tunneling Process

An inbound request from a SAS client to a SAS server is shown as follows:

Figure 1.1 SSH Tunneling Process



❶ The SAS client passes its request to the SSH client's port 5555.
❷ The SSH client forwards the SAS client's request to the SSH server via an encrypted tunnel.
❸ The SSH server forwards the SAS client's request to the SAS server via port 4321.

Outbound, the SAS server's reply to the SAS client's request flows from the SAS server to the SSH server. The SSH server forwards the reply to the SSH client, which passes it to the SAS client.

SSH Tunneling: Process for Installation and Setup

SSH software must be installed on the client and server computers. Exact details about installing SSH software at the client and the server depend on the particular brand and version of the software that is used. See the installation instructions for your SSH software.

The process for setting up an SSH tunnel consists of the following steps:

- SSH tunneling software is installed on the client and server computers. Details about tunnel configuration depend on the specific SSH product that is used.
- The SSH client is started as an agent between the SAS client and the SAS server.
- The components of the tunnel are set up. The components are a "listen" port, a destination computer, and a destination port. The SAS client will access the listen port, which gets forwarded to the destination port on the destination computer. SSH establishes an encrypted tunnel that indirectly connects the SAS client to the SAS server.

For examples of setting up and using a tunnel, see "SSH Tunnel for SAS/CONNECT: Example" on page 47 and "SSH Tunnel for SAS/SHARE: Example" on page 48.

Encryption Algorithms

The following encryption algorithms are used by SASProprietary and SAS/SECURE:

RC2
is a block cipher that encrypts data in blocks of 64 bits. A *block cipher* is an encryption algorithm that divides a message into blocks and encrypts each block. The RC2 key size ranges from 8 to 256 bits. SAS/SECURE uses a configurable key size of 40 or 128 bits. (The NETENCRYPTKEYLEN= system option is used to configure the key length.) The RC2 algorithm expands a single message by a maximum of 8 bytes. RC2 is a proprietary algorithm developed by RSA Data Security, Inc.

Note: RC2 is supported in SAS/SECURE and SSL. △

RC4
is a stream cipher. A *stream cipher* is an encryption algorithm that encrypts data 1 byte at a time. The RC4 key size ranges from 8 to 2048 bits. SAS/SECURE uses a configurable key size of 40 or 128 bits. (The NETENCRYPTKEYLEN= system option is used to configure the key length.) RC4 is a proprietary algorithm developed by RSA Data Security, Inc.

Note: RC4 is supported in SAS/SECURE and SSL. △

DES (Data Encryption Standard)
is a block cipher that encrypts data in blocks of 64 bits by using a 56-bit key. The algorithm expands a single message by a maximum of 8 bytes. DES was originally developed by IBM but is now published as a U.S. Government Federal Information Processing Standard (FIPS 46-3).

Note: DES is supported in SAS/SECURE and SSL. △

TripleDES
is a block cipher that encrypts data in blocks of 64 bits. TripleDES executes the DES algorithm on a data block three times in succession by using a single, 56-bit key. This has the effect of encrypting the data by using a 168-bit key. TripleDES expands a single message by a maximum of 8 bytes. TripleDES is defined in the American National Standards Institute (ANSI) X9.52 specification.

Note: TripleDES is supported in SAS/SECURE and SSL. △

SASProprietary
is a cipher that provides basic fixed encoding encryption services under all operating environments that are supported by SAS. Included in Base SAS, SASProprietary does not require additional SAS product licenses. The algorithm expands a single message to approximately one-third by using 32-bit encoding.

Note: SASProprietary is supported only by the SASProprietary encryption provider. △

AES (Advanced Encryption Standard)
is a block cipher that encrypts data in blocks of 128 bits by using a 256-bit key. AES expands a single message by a maximum of 16 bytes. Based on its DES predecessor, AES has been adopted as the encryption standard by the U.S. Government, and is one of the most popular algorithms that is used in symmetric key cryptography. AES is published as a U.S. Government Federal Information Processing Standard (FIPS 197).

Note: AES is supported in SAS/SECURE and SSL. △

Here is a summary of the encryption algorithms, by operating environment:

Table 1.1 Encryption Algorithms Supported by Operating Environments

	Operating Environments			
Encryption Algorithms	OpenVMS	UNIX	Windows	z/OS
SASProprietary	X	X	X	X
RC2		X	X	X
RC4		X	X	X
DES		X	X	X
TripleDES		X	X	X
SSL	X	X	X	X
AES		X	X	X

Encryption: Comparison

The following table compares the features of the encryption technologies:

Table 1.2 Summary of SASProprietary, SAS/SECURE, SSL, and SSH Features

Features	SASProprietary	SAS/SECURE	SSL	SSH
License required	No	Yes	No	No
Encryption and authentication	Encryption only	Encryption only	Encryption and authentication	Encryption only
Encryption level	Medium	High	High	High
Algorithms supported	SASProprietary fixed encoding	RC2, RC4, DES, TripleDES, AES	RC2, RC4, DES, TripleDES, AES	Product dependent
Installation required	No (part of Base SAS)	Yes	Yes	Yes
Operating environments supported	UNIX Windows z/OS	UNIX Windows z/OS	UNIX Windows z/OS * OpenVMS*	UNIX Windows z/OS
SAS version support	8 and later	8 and later	9 and later	8.2 and later

* SAS 9.2 introduces support for SSL on z/OS and OpenVMS.

Encryption: Implementation

The implementation of the installed encryption technology depends on the environment that you work in. If you work in a SAS enterprise intelligence infrastructure, encryption might be transparent to you because it has already been configured into your site's overall security plan. After the encryption technology has been installed, the site system administrator configures the encryption method (level of encryption) to be used in all client/server data exchanges. All enterprise activity uses the chosen level of encryption, by default. For an example, see "SAS/SECURE for the IOM Bridge: Examples" on page 45.

If you work in a SAS session on a client computer that exchanges data with a SAS server, you will specify SAS system options that implement encryption for the duration of the SAS session. If you connect a SAS/CONNECT client to a spawner, you will specify encryption options in the spawner start-up command. For details about SAS system options, see Chapter 2, "SAS System Options for Encryption," on page 15. For examples, see Chapter 4, "Encryption Technologies: Examples," on page 37.

Accessibility Features in SAS Products

For information about accessibility for any of the products mentioned in this book, see the documentation for that product. If you have questions or concerns about the accessibility of SAS products, send e-mail to accessibility@sas.com

Encrypting ODS Generated PDF Files

You can use ODS to generate PDF output. When these PDF files are not password protected, any user can use Acrobat to view and edit the PDF files. In SAS 9.2, you can encrypt and password-protect your PDF output files by specifying the PDFSECURITY system option. Two levels of security are available: 40-bit (low) and 128-bit (high). With either of these settings, a password will be required to open a PDF file that has been generated with ODS.

You can find information on using the ODS PRINTER and PDF statements in the *SAS Output Delivery System: User's Guide*. The following table lists the PDF system options that are available to restrict or allow users' ability to access, assemble, copy, or modify ODS PDF files. Other SAS system options control whether the user can fill in forms and set the print resolution. These system options are documented in *SAS Language Reference: Dictionary*.

Table 1.3 PDF System Options

Task	System Option
Specifies whether text and graphics from PDF documents can be read by screen readers for the visually impaired	PDFACCESS | NOPDFACCESS
Controls whether PDF documents can be assembled	PDFASSEMBLY | NOPDFASSEMBLY
Controls whether PDF document comments can be modified	PDFCOMMENT | NOPDFCOMMENT
Controls whether the contents of a PDF document can be changed	PDFCONTENT | NOPDFCONTENT
Controls whether text and graphics from a PDF document can be copied	PDFCOPY | NOPDFCOPY
Controls whether PDF forms can be filled in	PDFFILLIN | NOPDFFILLIN
Specifies the password to use to open a PDF document and the password used by a PDF document owner	PDFPASSWORD

Task	System Option
Controls the resolution used to print the PDF document	PDFPRINT
Controls the printing permissions for PDF documents	PDFSECURITY

Note: The SAS/SECURE SSL software is included in the SAS installation software only for countries that allow the importation of encryption software. △