CHAPTER 1

Technologies for Data Security

Data Security Technologies: Overview **3**
Providers of Data Security Technologies **4**
SASProprietary **4**
SASProprietary Overview **4**
SASProprietary System Requirements **4**
SASProprietary Installation and Configuration **4**
SAS/SECURE **4**
SAS/SECURE Overview **4**
SAS/SECURE System Requirements **4**
Export Restrictions for SAS/SECURE **5**
SAS/SECURE Installation and Configuration **5**
Secure Sockets Layer (SSL) **5**
Secure Sockets Layer (SSL) Overview **5**
SSL System Requirements **6**
SSL Concepts **6**
SSL Installation and Configuration **7**
SSH (Secure Shell) **7**
SSH (Secure Shell) Overview **7**
SSH System Requirements **7**
SSH Tunneling Process **7**
SSH Tunneling: Process for Installation and Setup **8**
Data Encryption Algorithms **8**
Data Security Technologies: Comparison **9**
Data Security Technologies: Implementation **10**

Data Security Technologies: Overview

As e-business grows, 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. *Data security technologies* refers to the foundation SAS products and third-party strategies for protecting data and credentials (user IDs and passwords) that are exchanged in a networked environment. Fundamental to these technologies is the use of proven, industry-standard encryption algorithms for data protection.

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. Although encryption increases the protection of data, it does not prevent unauthorized access to data.

Authentication is the act of verifying the identity of an entity (such as a user). Authentication is used to confirm the authority of an entity to access protected resources. For details about authentication, see the documentation for your enterprise.

Providers of Data Security Technologies

- □ “SASProprietary” on page 4
- □ “SAS/SECURE” on page 4
- □ “Secure Sockets Layer (SSL)” on page 5
- □ “SSH (Secure Shell)” on page 7

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 9.1.3 supports SASProprietary under the following operating environments:

- □ OpenVMS Alpha
- □ 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 22.

SAS/SECURE

SAS/SECURE Overview

SAS/SECURE software is an add-on product that provides encryption algorithms in addition to the SASProprietary algorithm. SAS/SECURE requires a license, and it must be installed on each computer that runs a client and a server that will use the encryption algorithms. Although SAS/SECURE increases data security, it cannot completely prevent unauthorized access to your data.

SAS/SECURE System Requirements

SAS 9.1.3 supports SAS/SECURE under the following operating environments:

- □ UNIX

    - Compaq Tru64 UNIX
    - HP UX on Itanium 64-bit platform
    - HP UX on a 64-bit platform
    - Linux for Intel Architecture on a 32-bit platform
    - Solaris on a 64-bit platform
- Windows
- z/OS

Export Restrictions for SAS/SECURE

SAS/SECURE 9.1.3 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 128-bit or 40-bit keys
- RC4 using 128-bit or 40-bit keys
- DES using 56-bit keys
- TripleDES using 168-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. For this reason, SAS/SECURE for Windows has very few export restrictions.

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 3, “Data Security Technologies: Examples,” on page 21.

Secure Sockets Layer (SSL)

Secure Sockets Layer (SSL) Overview

SSL is an abbreviation for Secure Sockets Layer, which is a protocol that provides network security and privacy. Developed by Netscape Communications, SSL uses encryption algorithms that include RC2, RC4, DES, TripleDES, IDEA, MD5, 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 both Netscape Navigator and Internet Explorer. Many Web sites use the protocol to protect confidential user information, such as credit card numbers. By convention, URLs that require an SSL connection begin with https: instead of http:. 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 under the following operating environments:

□ UNIX

□ Windows

CAUTION:

SAS/SECURE SSL is packaged as an add-in product. In order to use the SAS/SECURE SSL software, you must review the licensing terms and download the appropriate Add-in Package from **`www.sas.com/apps/demosdownloads/setupintro.jsp`**. Select **SAS/SECURE Software ▶ SSL Add-in.** △

The SAS/SECURE SSL software is not included on the SAS software CD because some countries do not allow the importation of encryption software. Therefore, SAS/SECURE SSL is provided as an add-in that can be downloaded from the WWW to customers who can import encryption software.

SSL Concepts

Concepts that are fundamental to understanding SSL follow:

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. 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, therefore, anyone can send a confidential message. The private key is confidential and known only to the owner of the key pair, therefore, 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 MACs, which ensure that a document has not been changed since it was signed.

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 or Windows. For complete details, see Appendix 1, "Installing and Configuring SSL under UNIX," on page 35 or Appendix 2, "Installing and Configuring SSL under Windows," on page 41.

For examples of configuring and using SSL in your environment, see Chapter 3, "Data Security Technologies: Examples," on page 21.

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 include a programming interface to SSH functionality, SAS does support the *tunneling* feature of SSH that enables a SAS client to make an encrypted connection to 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

SSH runs under UNIX and Windows operating environments. OpenSSH supports SSH protocol versions 1.3, 1.5, and 2.0.

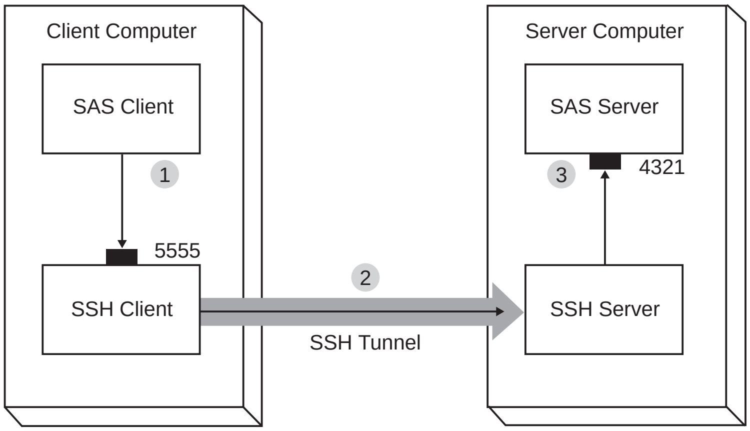
For additional resources, see

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

SSH Tunneling Process

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

Figure 1.1 SSH Tunneling Process



1 The SAS client passes its request to the SSH client's port 5555.

2 The SSH client forwards the SAS client's request to the SSH server via an encrypted tunnel.

3 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 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.
- The SAS server is started.
- The SSH client is started as an "agent" between the SAS client and the SAS server.

For examples of setting up and using a tunnel, see "SSH Tunnel for SAS/CONNECT: Example" on page 30 and "SSH Tunnel for SAS/SHARE: Example" on page 30.

Data Encryption Algorithms

The following encryption algorithms are used by the data security technologies:

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 to 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 to 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 to 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 a 32-bit key.

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

IDEA (International Data Encryption Algorithm)
is a 64-bit iterative block cipher that uses a 128-bit key.

Note: IDEA is supported only in SSL. △

MD5 (Message Digest)
is used for digital signature applications in which a large message must be securely compressed before being signed with a private key. The MD2, MD4, and MD5 family of algorithms share common structures, however, each design is unique. MD2 was optimized for 8-bit computers; MD4 and MD5 were designed for 32-bit computers. MD5 produces a 128-bit message digest from a message of arbitrary length.

Note: MD5 is supported only in SSL. △

Data Security Technologies: Comparison

A comparison of the features of the data security technologies follow:

Table 1.1 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	RC2, RC4, DES, TripleDES, IDEA, MD5, and others	Product dependent
Installation required	No (part of Base SAS)	Yes	Yes	Yes
Operating environments supported	UNIX Windows z/OS OpenVMS Alpha	UNIX Windows z/OS	UNIX Windows	UNIX Windows
SAS version support	8 and later	8 and later	9 and later	8.2 and later

Data Security Technologies: Implementation

The implementation of the installed data security technology depends on the environment that you work in. If you work in a SAS enterprise intelligence infrastructure, data security might be transparent to you because it has already been configured into your site's overall security plan. After the data security 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 28.

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 data security 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 Data Security," on page 11. For examples, see Chapter 3, "Data Security Technologies: Examples," on page 21.