September 13, 2002
databeast, Inc.
1668 Trumansburg Rd.
Ithaca, NY 14850

U.S. Department of Commerce
Bureau of Industry and Security
Office of Strategic Trade and Foreign Policy Controls
Room 2705
14th Street and Pennsylvania Ave., N.W.
Washington, D. C. 20230

Attn: "Application Enclosed"

re: dataComet-Secure VX (dataComet 10.0) Z 255215

Dear Folks,

This cover letter provides information required under the "Guidelines for Submitting a Classification Request for Encryption Items" (Supplement No. 6 to part 742) and supplementary technical information from the dataComet-Secure on-line documentation.

(a) The product requiring classification is "dataComet-Secure" version 10 for native execution under the Mac OS X operating system. An earlier version of this product supporting older MacOS releases was reviewed (Z 254919, 07/02/2001) and was classified under 5D002 as an item covered under ENC under Sections 740.17(B)(3); the encryption functionality in dataComet-Secure version 10 is the same as that provided by the earlier version.

(b) Duplicate copies of all materials submitted with this letter have been sent to the ENC Encryption Request Coordinator.

Sincerely,

Kevin Eric Saunders, President

Enc: Form BXA-748P, Z 255215

Overview of dataComet-Secure Encryption

dataComet-Secure VX for Macintosh OS X offers Telnet/TN3270 (Kerberos 5 authentication) or Secure Shell (SSH1/SSH2) connections. Data may be compressed (ZLIB) and encrypted using 56-bit DES, 168-bit 3-DES, or 128-bit Blowfish algorithms.

In addition to encrypting network data streams using the SSH1, SSH2, or Telnet protocols, dataComet-Secure allows users to save session passwords in an encrypted form using a master passphrase; this passphrase is condensed to an MD5 digest to form a unique key for 3-DES encryption and decryption of the session passwords, which are saved in session configuration documents.

SSH1: Uses standard SSH 1.5 encapsulation of data

Host authentication:

RSA

User authentication:

RSA public key user authentication OR

Password authentication using encryption type and vectors negotiated during RSA host authentication phase.

Encryption:

DES CBC mode
3-DES CBC mode
Blowfish

SSH2: Uses standard SSH2 encapsulation w/ SHA1 or MD5

Host authentication:

DSA

User authentication:

Password authentication using encryption type and vectors negotiated during DSA host authentication phase.

Encryption:

3-DES CBC mode
Blowfish

Telnet/TN3270: Uses Kerberos 5 for mutual authentication.

Encryption:

DES CFB or OFB mode
3-DES CFB or OFB mode

Information required by Supplement No. 6 to part 742

(1) Asymmetric encryption algoritms used:

RSA public key encryption (arbitrary key length and modulus)
DSA public key encryption (arbitrary key length and modulus)

Used for Secure Shell public key authentication.

Symmetric encryption algoritms used:

DES CBC, CFB, or OFB (56-bit)
3-DES CBC, CFB, or OFB (168-bit)
Blowfish CBC (128-bit)

Used for data stream encryption after Telnet or SSH encryption keys have been exchanged in the authentication phase.

(2) dataComet-Secure can generate unique RSA keys for use in SSH user authentication. 1024-bit keys are generated by default. All RSA keys are generated with modulus 37.

(3) dataComet contains no proprietary encryption algorithms.

(4) Pre-processing: Standard Telnet, SSH1, or SSH2 protocols are used to pre-process data prior to encryption. SSH data streams may be compressed using ZLIB prior to encryption. Data streams may also be formatted prior to Session layer processing as required by one of several file transfer protocols (X/Y/Z-MODEM, SCP Secure Copy, or IND$FILE).

(5) Post-processing: Encrypted data transmitted over an SSH connection is framed according to the specifications for the version of SSH in use; no post-processing is performed on encrypted Telnet data prior to transmission over the TCP/IP network.

(6) Protocols supported on TCP/IP connections:

Session	Authentication	Encryption
TN3270	K5	DES, 3-DES CFB/OFB modes RFCs 2946 2947 2952
TELNET	K5	DES, 3-DES CFB/OFB modes RFCs 2946 2947 2952
SSH1	RSA	DES, 3-DES, Blowfish CBC mode
SSH2	DSA	3-DES, Blowfish CBC mode

File transfer over Session protocol connections using:

TN3270	IND$FILE (PC3270)
SSH or TELNET	X/Y/Z-MODEM, SCP (Secure Copy)

(7) The only encryption-related API used is the MIT Kerberos 5 library. No calls are made to the MIT Kerberos 5 driver for encryption or decryption of session data by dataComet-Secure; Kerberos 5 internal encryption capabilities are only used indirectly for authentication of tickets obtained from the Kerberos Key Distribution Center.

(8) All cryptographic functions used by dataComet-Secure are embedded in the application's Mach-O object file; modifying the cryptographic functionality would be difficult (note that ALL native applications running under Mac OS X are dynamically linked). The OS X Kerberos 5 library is accessed as a shared library only for purposes of performing K5 login, mutual user/host authentication, and acquisition of the subkeys used for session encryption.

(9) N.A.: dataComet-Secure is a C-language program which does not use Java.

(10) Since all encryption object code is linked at the application level, it would be difficult for a user to modify any of the cryptographic functions. Users cannot select custom key sizes, except when generating an RSA key.

(11) Retail classification requirements per 740.17(b)(3) are met:

(3) Retail encryption commodities and software. ...

(i) Retail encryption commodities, software and components are products and components:

(A) Generally available to the public by means of any of the following:

YES

(1) Sold in tangible form through retail outlets independent of the manufacturer;

NO

(2) Specifically designed for individual consumer use and sold or transferred through tangible or intangible means; or

YES

(3) Are sold or will be sold in large volume without restriction through mail order transactions, electronic transactions, or telephone call transactions; and

YES

(B) Meeting all of the following:

(1) The cryptographic functionality cannot be easily changed by the user;

YES

Embedding the encryption routines within a single binary object file makes modification difficult.

(2) Substantial support is not required for installation and use;

YES

dataComet-Secure is designed to be a plug-and-play Macintosh Telnet/SSH application requiring minimal user configuration.

(3) The cryptographic functionality has not been modified or customized to customer specification; and

YES

dataComet-Secure uses only standard implementations of common encryption algorithms.

(12) N.A. dataComet-Secure 10 does not incorporate an Open Cryptographic Interface.

---

dataComet-Secure Feature Summary

---

dataComet-Secure adds support for the SSH1 and SSH2 Secure Shell terminal protocols and the SCP "Secure Copy" file transfer protocol. (TCP and X-Window tunnelling are NOT supported.) All features of dataComet, including ZModem file transfers, work with SSH connections. SSH features currently supported include automatic SSH protocol selection, client authentication using user passwords (and under SSH1, using RSA public keys), encryption using Triple-DES or Blowfish, and data compression using zlib. Host public keys are maintained in files using the standard "known hosts" format (e.g., a "NiftyTelnet SSH Known Hosts" document can be copied directly to dataComet's "Security" folder and used as-is). More information on SSH configuration is available in the Help document "3. Dialogs".

Passwords can be stored in session documents in encrypted form to speed logins while maintaining security. You can use one secret Master Passphrase to unlock all your session passwords. In addition, you can use the "Lock" command to prevent use of dataComet until a master passphrase is entered, so that you can leave sessions open on your unattended computer without seriously compromising security. The password encryption is performed using strong encryption (Triple-DES, 168 bits).

---

dataComet-Secure "Security" folder

The "Security" folder in dataComet's home folder and the "{User}/Library/Preferences/dataComet Preferences/Security" folder contain documents which dataComet will use for authentication (Under OS 9, the "System Folder:Preferences:dataComet Preferences:Security" folder is used). Documents containing host public keys placed in these folders will be used for verifying the identity of hosts to which you connect; all documents in these folders are scanned at startup time, and lines that appear to contain valid host keys are added to tables in memory for lookups when SSH host connections are being authenticated.

The public key files must be 'TEXT' type documents, and must use either the standard ssh_known_hosts or ssh_known_hosts2 formats. This allows files to be copied directly from standard SSH "known hosts" files on UNIX hosts.

Host keys added by dataComet will be saved in files named "known_hosts.dataComet" (SSH1) and "known_hosts2.dataComet" (SSH2) using the standard format.

Installing Kerberos Support

---

dataComet-Secure supports Telnet connections authenticated using the Kerberos 5 protocol. This requires that you configure the Kerberos package. (Kerberos support is bundled with Mac OS X; OS 9 versions can be downloaded from <http://web.mit.edu/network/kerberos-form.html>. NOTE: this package is supported only on Macintosh PowerPC machines.)

Telnet "Configure…"

Using the Telnet "Configure…" dialog

This dialog allows you to configure special options for a Telnet session.

"Authentication": Allows you to select a protocol for performing authentication with hosts for a Telnet session. The checkbox can be used to disable Authentication. Currently only Kerberos 5 is supported.

"Encryption": The checkbox can be used to disable Encryption. Telnet allows the selection of a number of different encryption algorithms, which encrypt the plain text of your session so that eavesdroppers cannot (easily) decipher it. "DES3" Triple-DES encryption is the most secure option; generally you should use it rather than DES. Note that if no lock picture appears in the left bottom side of the emulator window after you connect, the session is not being encrypted; some hosts may not support DES3 encryption over Telnet sessions.

"Compression": This option is not yet supported.

"WILL SGA (Berkeley linefeed fix)": Causes dataComet to send WILL SGA, the Telnet Send-Go-Ahead option, in order to get BSD UNIX derivatives to handle carriage returns correctly.

"WILL NAWS (Negotiate Window Size)": Causes dataComet to send WILL NAWS, the Telnet option which allows the host screen size to be adjusted automatically if you change the size of a dataComet emulator screen.

"Display log messages on-screen": Display the Telnet logon messages on the emulator screen. This can help debug connection problems, and shows the Telnet options that are available on the host and which options are actually selected for the session.

SSH "Configure…"

Using the SSH "Configure…" dialog

This dialog allows you to configure a "Secure Shell" (SSH) session. Support for these options is only included in dataComet-Secure.

"SSH Version": There are two different SSH protocols, SSH1 and SSH2. This option allows you to force an SSH2 host to select SSH1 or SSH2 rather than allowing it to make the choice.

"Encryption": SSH allows the selection of a number of different encryption algorithms, which encrypt the plain text of your session so that eavesdroppers cannot (easily) decipher it. Triple-DES is the most secure option; Blowfish is somewhat faster than 3-DES but probably not quite as secure.

"Compression": SSH can compress the data stream, which enhances security and may speed up (or slow down) sessions substantially; the speed increase (or decrease) will be directly proportional to the ratio between the speed of your computer and the speed of the network connection (e.g., if you have a Macintosh G3, a session with compression on is much faster on a slow dialup connection, still significantly faster with an Ethernet connection, and possibly slower if you have a direct Gigabit Ethernet connection to a very fast host).

NOTE that using compression requires an extra 250K bytes of memory per session!

"MAC type": SSH2 helps guarantee communications security by adding a "Message Authentication Code" field to each data packet. The "SHA-1" protocol is used by default.

"Authentication": SSH offers several different methods of "authenticating" your identity to the host. Passwords are the same as your usual host password, with the major difference that under SSH passwords are encrypted so eavesdroppers can't use network "sniffers" to steal your password. RSA and DSS "public key authentication" methods are used by SSH1 and SSH2, respectively; dataComet-Secure only supports RSA public key authentication. RSA keys must be saved in the ":dataComet Preferences:Security:" folder. NOTA BENE: Maintaining security while using a public key REQUIRES that the private key be encrypted using a Passphrase (which can be the same as your Master Passphrase) and should not be shared with other users.

"Keys…": This button brings up a dialog which allows you to create, save, view, and copy public keys, so you can copy and paste them into host key files.

"Use Key…": This button allows you to select the private key file which will be used for an SSH session if you select one of the public key methods in the "Authentication" popup menu.

"Display log messages on-screen": Display the SSH logon messages on the emulator screen. This can help debug connection problems, and shows clearly the SSH options that are available on the host and which options are actually selected for the session.

"Don't allocate PTY (host terminal handler)": This option allows you to skip creating a "pty" on the host, which is used to control the interface to applications on the host which need terminal control information. (You will almost always want this option off!)

"Execute command (rssh)": Allows you to execute a command on the host and then close the session automatically. You can enter the command to be executed in the text field below.

Using the SSH "Keys…" dialog

This dialog allows you to manage RSA keys for user authentication of SSH1 "Secure Shell" sessions. (Use of RSA keys with SSH2 is not yet supported by dataComet.)

The "Key File..." button allows you to open a key file in the dataComet "Security" Folder in the "dataComet Preferences" Folder so you can view and copy the public portion of the key, and save the key with a new name and passphrase if you wish.

The "Public Key" field shows the public key for the key. This is the key which is added to the host file "~/.ssh/authorized_keys" to enable you to log on to the host using your key rather than a password. You can copy the Public Key in this field and then paste it into "authorized_keys" file using a host editor. NOTE that you should make every effort to keep the private portion of the key file private, including using a passphrase to encrypt it. If someone copies your private key, the can log on to your account, just as if you had given them a password for a password-protected account.

The "Fingerprint" field shows the fingerprint of the key, which is a condensed representation of the key useful for verifying (e.g., in a telephone conversation) that a key is valid.

The "Make Key…" button brings up a dialog which allows you to generate a new key.

The "Save Key…" button allows you to save a key.

Using the SSH "Make key…" dialog

This dialog allows you to generate RSA keys.

The "Key Size" field allows you to specify a size other than the usual 1024-bit key length. If you want your encrypted communications to remain secure over a long period of time, you should use more than a size greater than 1024, probably 2048. Note that it takes substantially more time to generate the key and to verify it when making a host connection when you use a larger key.

The "Comment" fields allows you to enter a comment, which will be appended to the newly generated key.

The progress bar indicates roughly the amount of work left in generating the key.
