



85 Grove Street - Peterborough, NH 03458
voice 603-924-6079 • fax 603-924-8668

Payment Application Best Practices Secure Implementation Guide for Trevance®

(Covers PCI, CISP, SDP, PABP)

Version 1.1

9 January 2006

Overview

The Trevance® Transaction Gateway is developed for use in a Payment Card Industry (PCI) Data Security compliant enterprise. Auric Systems International (Auric) follows Visa's Payment Application Best Practices (PABP) and has undergone an independent third-party audit of our development processes as well as the actual Trevance application. At present (January 2006) PABP is a voluntary compliance process for software vendors.

As per the Visa site:

Visa has developed "Payment Application Best Practices" to address security and the risks associated when full magnetic stripe data or CVV2 values are stored after authorization by payment applications. The best practices assist software vendors in creating secure payment applications that help ensure merchant CISP compliance.

This document contains Auric Systems International's installation and configuration recommendations for Trevance. Merchants must make their own determination as to how best to create a PCI-compliant enterprise.

Compliance Status

Is Trevance PCI compliant? Technically, software cannot be PCI-compliant. PCI is a process that applies to merchants – not software. There are 12 basic steps ranging from building and maintaining a secure network, to protecting cardholder data, to maintaining an information security policy. Software, such as Trevance, must be evaluated to see how it fits within a merchant's overall PCI efforts. PCI is for merchants PABP is for software.

So, is Trevance PABP compliant? We think so, and the independent auditor's who have reviewed our processes and software also think so (see accompanying document). The PABP compliance documents have been submitted to Visa and we are informed there may be a wait of as much as six months before the official Visa determination of compliance.

Recommendations

This document contains *recommendations*. Auric cannot say, "If you do this, you will be PCI compliant." Nor can we say, "If you do not do this, you will definitely not be PCI compliant." Merchants are responsible for implementing their own PCI-compliant environment. What we hope to do is to provide sufficient information regarding the installation, configuration, and operation of Trevance to help your PCI compliance efforts.

Section Numbering

The section numbering in the rest of this document corresponds to the number of Visa's Payment Application Best Practices document. It attempts to avoid being yet another tutorial into PCI-compliance implementation. Instead, it focuses on checklists--installation, configuration, and operation--with the occasional detailed foray into specific topics.

The following sections assume you have installed Trevance and can refer to the various configuration screens.

1: Do not retain full magnetic stripe or CVV2 data

From the Configure/Options dialog, check the following:

Security Tab:

- Multi-Pass Overwrite and Delete is checked.
After a batch file is imported it is deleted in a secure manner by being overwritten multiple times before the actual deletion.
If this should cause excessive hard disk activity in your specific installation, the second-best approach is to use the One-Pass Overwrite and Delete.
See Secure File Deletion section in this document for details.

Troubleshooting Tab:

- Turn off all Additional Logs that you are not explicitly using.
Most of these logs are used when trying to isolate a communication problem.
- If you do turn on any Additional Logs, check the Mask Identifying Information in Log Files checkbox. This causes the sensitive data such as account numbers and CVV2/CID values to be masked in the output streams.
Note: Trevance can only mask information appearing in the proper field. If a credit card or CVV2 value should be placed in the wrong field, Trevance has no way to know it should be masked.

From the Configure/Batch Files/Imports dialog, check the following:

Grid:

- CVV/CID field is not imported.

Encryption Tab:

- Import Files are Encrypted is checked.

After Import Tab:

- Delete File.

From the Configure/Batch Files/Exports dialog, check the following:

Grid:

- Account field is either not exported or exported masked.

Options:

- Export Files are Encrypted is checked.

Notes:

- The Trevance real-time web interface accepts transactions containing CVV2/CID, magnetic stripe, and debit card PIN block data. This information is transmitted directly to the processor and never stored.
- The Trevance batch file interface accepts transactions with CVV2/CID data. This feature is provided for integration with legacy systems. Auric recommends that CVV2 data not be transmitted in batch files.
- Import and export file encryption are discussed later in this document.
- If you do not encrypt the import file, Auric strongly recommends you delete the import file after it is read.
- If you do not delete the import file, Auric strongly recommends you mask sensitive data after import.

In this mode, instead of just changing the imported file's extension from .IMP to .DNE, Trevance copies the .IMP file to a temporary file while masking sensitive data such as

account number and CVV2/CID. When the copy is complete, the .IMP file is deleted and the new, masked, copy is given the .DNE extension.

- Trevance currently does not support a real-time drop-file interface. Auric can provide solutions that integrate Trevance with a legacy drop-file system. Such systems can be tuned to meet a specific merchant's PCI needs.
- Avoid, if at all possible, exporting the account code. Instead, use the order number field or an internal tracking ID in one of the four comment fields.

2: Protect stored data

Trevance ships with a default encryption key that must be replaced before running Trevance in Test or Production mode. Select Configure/Set Server Passphrase to set the new encryption key.

The Server Passphrase is divided into two segments for increased security.

People entering the two segments are known as the key custodians.

- _ Each segment must be entered by a different key custodian.
- _ Each segment must be at least seven characters long and contain both alpha and numeric characters.
- _ These segments must be maintained according to your company's policies and procedures for securely handling passwords.
- _ Auric recommends that the key custodians sign a form specifying they understand and accept their key-custodian responsibilities.
Appendix A has a sample key custodian form.

Trevance uses 256-bit AES encryption for all sensitive data.

The passphrase segments should be changed on a schedule in compliance with your company security policy.

The old passphrase segments should be destroyed in compliance with your company security policy.

For additional security, the passphrase segments can only be entered through the Trevance Console when it is running on the same machine as the Trevance Server.

Notes:

The Server Passphrase is used to encrypt sensitive information such as account numbers stored in the database. These passphrases are stored using Microsoft Windows Encryption APIs for the storage of sensitive data. This storage is per-user. When you enter the passphrase, Trevance must currently be running as the same user under which it will run in Test and Production mode. Otherwise, Trevance will be unable to find the passphrase segments at run-time. Only Administrative users can see the Set Server Passphrase dialog.

3: Provide secure password features

Trevance provides a default ADMIN administrative account that needs to be replaced before running Trevance in Test or Production modes. If you are using the real-time Web interface, you must also replace the default WEB user.

From the Configure/Administer Users dialog:

- _ Create a new user.
- _ Set the User Type to Administrator
- _ Enter a strong password consisting of at least seven (7) characters and both alpha and numeric characters.
- _ Repeat, if necessary, for default WEB user.
- _ Create a unique user ID for each person requiring access to the Trevance console.
- _ Provide Administrative access only to those users who must change Trevance configurations. All other users should receive Console access.
- _ Passwords must be maintained according to company policies and procedures. Specifically, PCI recommends that passwords be changed every 90 days.
- _ Never share passwords or let another user have access to your account.

Notes:

Trevance passwords may be as long as 40 characters. This encourages the use of long, easily remembered passwords (sentences, poems, etc.) vs. short cryptic passwords. Spaces and punctuation are acceptable password characters.

Trevance maintains a history of the last four passwords used and does not allow them to be reused.

If a user fails to login after six attempts, they are locked out of the system for 30 minutes. The one exception to this is the Web user accounts for the real-time web transaction interface. A lockout in this instance would lead to a denial of service.

Administrative accounts are automatically logged-out after 15 minutes of inactivity. Console users are not automatically logged out since typically these are used as long-term monitoring accounts.

Auric recommends that Administrative accounts be used solely for administration, and not for monitoring purposes.

4: Log application activity

The Trevance console_user log maintains a running log of Administrative and Console users who connect to Trevance. This log should be regularly monitored for failed log-in attempts.

The Trevance audit log provides a list of activities performed by Administrative and Console users. This log contains both the users log-in name and a date/time stamp at which the activity occurred.

These logs are stored as simple text files that are easily reviewed.

From the Configure/E-Mail Notification dialog:

- _ check All Logs to have the daily logs automatically emailed to you.
- _ configure the settings for your SMTP mail server.
- _ select a time at which the logs should be emailed to you.
- _ check Login Report to receive an email whenever anyone logs into Trevance.

In general:

- _ Use a Network Time Protocol service to ensure the time on the Trevance server is properly synchronized.
- _ Check the timezone and Daylight Savings/Standard Time flag is set properly on the Trevance server.

Check all logs on a daily basis.

Implement automated audit trails to reconstruct the following events, for all system components:

- _ All individual user accesses to cardholder data
- _ All actions taken by any individual with root or administrative privileges
- _ Access to all audit trails
- _ Invalid logical access attempts
- _ Use of identification and authentication mechanisms
- _ Initialization of the audit logs
- _ Creation and deletion of system-level objects.

Record at least the following audit trail entries for each event, for all system components:

- _ User identification
- _ Type of event
- _ Date and time
- _ Success or failure indication
- _ Origination of event
- _ Identity or name of affected data, system component, or resource.

5: Develop secure applications

This section of the PABP standard is heavily focused on the development of secure web (Internet-based) applications.

Although Trevance has a Web interface, it is not a Web application and is not designed to be implemented directly on the public Internet. See section 8 below for recommendations on secure network implementation.

Where applicable, Auric has followed the Open Web Application Security Project guidelines available at <http://www.owasp.org>. Auric recommends any merchant integrating Trevance with a Web site also follow the OWASP guidelines.

6: Protect wireless transmissions

The use of wireless networks is outside the scope of the Trevance implementation.

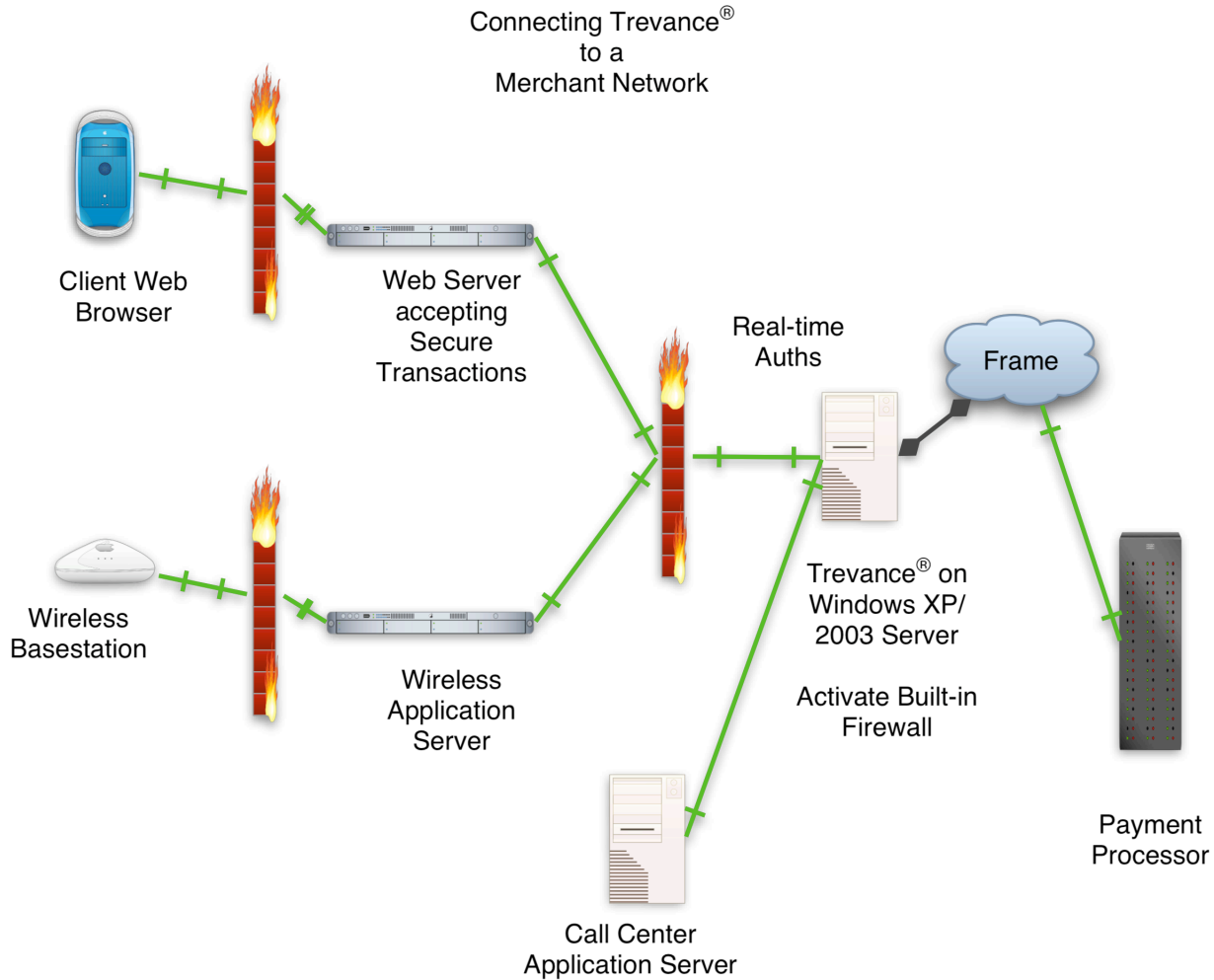
If Trevance is integrated into a merchant system using Wireless payment applications, the merchant must address the PCI compliance requirements including:

- use of appropriate encryption technologies such as VPN, 128-bit or greater SSL/TLS, 128-bit WEP (Wired Equivalency Protocol), and/or WPA.
- proper key rotation
- proper use of firewalls (see section 8).
- removal of all default keys from wireless equipment

7: Test applications to address vulnerabilities

In addition to on-going internal testing, Auric monitors outside security sources and product-specific mailing lists to check for product vulnerabilities. If a vulnerability is found in the Trevance system, merchants will be so informed and a timely correction will be provided.

8: Facilitate secure network implementation



- Operate Trevance on a separate server.
- Isolate the Trevance server from the public Internet.
- Maintain your web server in a DMZ as shown in the above diagram.
- Do not run Trevance in the DMZ (where the Web Server or Wireless Application Server are shown in the above diagram).
- If your application must use wireless, provide wireless access through a separate firewall and isolate the application server.

9: Cardholder data must never be stored on a server connected to the Internet

Trevance runs on the local, private network and not in either the DMZ or on a server directly connected to the Internet.

10: Facilitate secure remote software updates

Auric does not force automatic Trevance updates. The latest update is always available for immediate download from the Auric web site. Both MD5 and SHA-128 hashes are provided on the Auric web site. After downloading the release or update, merchants should perform their own MD5 and/or SHA-128 calculation on the downloaded file to check the hashes before installing. These hashes are also available via Email from Auric. Please call tech support for details.

Auric provides tools to perform these calculations but recommends merchants use third-party tools to ensure integrity.

11: Facilitate secure remote access to application

All remote access to the Trevance server is via the Trevance Console (TrevCon.exe). Communications between the Trevance Console and Trevance Server is encrypted using 128-bit AES. Keys are dynamically created and exchanged at the start of the communication.

12: Encrypt sensitive traffic over public networks

Trevance is designed for installation on a private network – not a public network. As such, sensitive traffic is not communicated over the public network.

Trevance has no facility for emailing credit card information.
Never email sensitive credit card information in an unencrypted form.

13: Encrypt all non-console administrative access

All administrative access to Trevance is through the Trevance Console.

All configuration changes must occur through the Trevance Console.

Communications between the Trevance Console and the Trevance Server are encrypted using 128-bit AES encryption. The encryption keys are dynamically generated and exchanged using RSA protocols.

Trevance Secure File Deletion

Trevance offers the option to securely delete files. Normally, files deleted using the standard services provided by the operating system do not erase the actual data in the file. Files deleted this way can be easily recovered using software "undelete" tools. Even files that have been overwritten can sometimes be recovered using additional hardware and sophisticated forensic techniques.

Trevance supports secure deletion methods. Secure delete is primarily intended for batch import files, but Trevance also applies the secure delete option to any external file that it handles that may contain sensitive information, including temporary encryption files created during the upload and download process

Trevance offers three deletion choices ranging from the quick (but not secure) standard operating system delete to a multi-pass secure deletion. Because the multi-pass secure deletion requires 35 write passes over the file, some sites may determine this consumes too much time or causes too much hard disk activity and interferes with other services. To address this, Trevance provides a one-pass secure delete that simply overwrites the file data with 0's before deleting.

These are the three options:

- Quick Delete
- One-Pass Overwrite and Delete
- Multi-Pass Overwrite and Delete

Quick Delete

This uses standard operating system calls. It doesn't overwrite any of the file (typically only the directory entry is updated) and so is very fast. However, file data is easily recovered if this option is used.

One-Pass Overwrite

Trevance first overwrites the file data with a single pass of binary zeros. This makes it difficult to recover the file using "undelete" tools, but the file data might still be recoverable using sophisticated forensic tools.

Multi-Pass Overwrite and Delete

This method overwrites file data with 35 passes using various data patterns. The 35 overwrite patterns, though possibly considered excessive for modern drives, is specifically designed to make data recovery extremely difficult. The pattern was developed by Peter Gutmann, and is often the pattern used by secure deletion utilities. Gutmann's paper describing the pattern can be found at:

<http://www.cs.auckland.ac.nz/~pgut001/pubs/secure_del.html>

Trevance Internal Encryption

Trevance uses a variety of encryption techniques, both to follow industry rules regarding the storage of sensitive information and to help reduce the exposure of cardholder data to unauthorized access.

Trevance uses encryption in the following areas:

- Communicating with the Processor over the Internet.
- VPN encrypted HTTP traffic.
- Batch Import/Export files.
- Stored data.

Communicating with the Processor over the Internet

Trevance's method of communication with each processor is based on the protocols provided by that processor. When communicating with processors over the Internet, Trevance uses the encryption mechanisms provided by each processor. Typical communications include S-FTP and VPN.

VPN Encryption of HTTP Traffic

Trevance contains an embedded HTTP web server through which real-time transactions can be processed. Since Trevance is implemented on a company's private, and not public, network, use of HTTPS security is not required by either the PABP or PCI standards.

Auric recommends it is a prudent practice to implement either a VPN or Secure Tunnel between the source of the HTTP traffic and Trevance. VPN and Secure Tunnels have several advantages over HTTPS including slightly lower processing requirements (which can be important in a high-speed environment).

If an HTTPS interface is important for your application, Auric has several solutions to address this need or Trevance can be placed behind a standard HTTPS-supporting server such as Apache.

Batch Import/Export Files

Trevance can import delimited text files that are externally encrypted using 256-bit AES encryption. Trevance can also export encrypted delimited text files. Import and export file encryption is recommended to ensure that transaction data is not exposed while the files reside on the filesystem.

Since AES is a symmetric algorithm, both Trevance and the external encryption program must have access to the same key. A key consists of any series of 256 bits.

Trevance can:

- **Generate keys.**
Trevance generates a random import/export encryption key then encrypts and stores the key in its database. A copy of the key is written to an external file for use by the external encrypting application. Treat this key in compliance with your company security policy.
- **Import keys.**
Trevance can read a file containing the encryption key and use that key for future import/export file decryption/encryption. The key may be one previously exported from Trevance, or one created externally.
- **Export keys.**
The import/export encryption key may be exported at any time.

The Trevance key file format is as follows:

- The file must contain a single key.
- The file must contain the key encoded using Base64 (<http://www.ietf.org/rfc/rfc3548.txt>).
- The raw key must be 256-bits, or 32 bytes. Because Base64-encoded text has a 4:3 expansion ratio, the encoded key is a single line of text, 44 characters in length.

Batch files are encrypted line-by-line. Each line in the encrypted batch file represents a line in the plaintext batch file.

The line-by-line approach is taken to ensure both Trevance and the merchant's external encryption routines can better handle the data in a secure manner. Import and export files can be quite large (10s or 100s of thousands of lines). If the file was encrypted as one item, it would be difficult to decrypt it at import time without creating an intermediate plain-text version. Since the goal of the encrypted batch file is to have end-to-end encrypted file handling, Auric selected the line-by-line approach.

The end-of-line characters (CR/LF) are not part of the encrypted line. End-of-line characters separate each line in the encrypted file.

Each line must be encrypted using AES with an 8-bit cipher feedback-chaining mode. The initialization vector must be set to 128 '0' bits. After encryption, each encrypted line is encoded using Base64 and written to the file.

The Auric encryption format adds a 16-character randomization factor to the beginning of each line. This ensures that plain-text import lines that start with identical values (e.g., Merchant

Identifiers, Order numbers with leading 0s, etc.) do not generate encrypted text that starts with identical values. Before encryption, each plaintext line must be prefixed with a 16-character string in the following format:

00SSMMHHddmmYYYY

Where:

00 A two-digit random number.

Other fields represent the time at which the information was encrypted:

SS Seconds.

MM Minutes.

HH Hour.

dd Day.

mm Month.

YYYY Four-digit year.

This same 16-character pattern is prefixed to each exported plaintext line before exports are encrypted.

Import and Export file encryption is controlled separately.

Stored Data

Sensitive fields stored in Trevance's internal embedded database are encrypted using 256-bit AES encryption. The following fields are encrypted:

Account

CVV/CID

Customer Social Security Number

Customer Drivers License Number

Customer Date of Birth

CVV/CID is stored (always encrypted) only on batch authorization requests, and only until the authorization process is complete. It is then cleared from the database. Auric recommends avoiding the use of CVV/CID values on batch imports whenever possible.

Each of these fields is encrypted using a single 256-bit key. In addition, processor-specific configuration information (such as an S-FTP login password) is stored encrypted in the database. These configuration fields are protected using a second 256-bit key. Trevance maintains two 256-bit keys, but for the purpose of further discussion, we'll speak as if it were a single key. If we need to distinguish between these two keys, we will call the key used to protect transaction information the "master key," and the key used to protect configuration data the "admin key."

Because Trevance is an automated server, it is not practical to require that an operator provide an encryption key each time the software is started. Therefore, Trevance must store the encryption key in a secure manner, without requiring user input in day-to-day operations.

It is also desirable to be able to rebuild or restore the database from backup, without storing the key with the backup copy. To meet this requirement, an operator must be able to provide the key from an external source upon restore.

The key management design meets the following requirements:

- Store the key securely so that the system can run in an automated fashion.
- Store the key outside of the database so that it cannot be recovered from backup media.
- Provide a way for authorized users (who have some secret information) to recover the key if the database is restored.
- Allow demo-mode use without key management.

Here is how Trevance meets these criteria:

Before Trevance can be run in test or production (non-demo) modes, two operators must each provide a part of a two-segment passphrase. Each operator is expected to use a passphrase segment that he or she can remember. These operators must be trusted to maintain this information securely. Because the passphrase is in two parts, both operators are needed to provide the complete passphrase.

When a passphrase is initially entered, Trevance generates a new master key and a new admin key. Each key contains 256 cryptographically random bits (generated using the Windows `CryptGenRandom` API).

Each of these keys is then AES encrypted using a hash of the passphrase provided. These encrypted keys are stored in the database. Because the passphrase is not stored in the database, these keys cannot be recovered from backup media unless the passphrase is provided.

The passphrase hash is itself encrypted using the Windows Data Protection API (DPAPI), in user mode. Using DPAPI encryption ties the encrypted value to the login credentials of the active user. That means that another Windows user cannot decrypt the passphrase; nor can the passphrase be recovered from another machine. For this reason, when the passphrases are set, the Trevance server **must** be running under the Windows account that it will run under for test and production. The encrypted passphrase hash is then stored in the database (again, it cannot be decrypted except by the current Windows user on the current machine).

At this point the setup process is complete. When run, Trevance decrypts the passphrase hash using DPAPI. It then uses the passphrase hash to decrypt the actual master and admin keys. This allows operators to change the passphrase at any time without requiring that database fields be re-encrypted.

If the database is restored on another machine or under another Windows user account, Trevance will fail to decrypt the passphrase. It will then enter a recovery mode and refuse to run unless the passphrase is provided again (from the console). If the passphrase is re-entered correctly, it is again hashed and re-encrypted using DPAPI; the hash is used to decrypt the master and admin keys. If the passphrase is not re-entered correctly, encrypted information in the database is not recoverable.

In demo mode, Trevance handles master and admin keys differently. This is because demo transactions never leave demo mode, but demo configuration information (protected by the admin key) may need to be consistent between demo and non-demo modes. (This is mostly

true if the user enters production configuration information in a production mode, switches to demo mode, and then switches back to production.)

In demo mode, Trevance uses a less-secure, hard-coded master key, whether a real passphrase has been entered or not. This key is used for demo transaction data only. If a real admin key is available (a passphrase has been entered), Trevance will use it even in demo mode. This means that production configuration information will migrate properly (and securely) from mode to mode. If a real admin key has not been generated, Trevance will simply substitute blanks for encrypted data when writing encrypted fields. This allows users to experiment in demo mode without providing a passphrase, and it keeps production configuration information from being stored insecurely. This also means that some configuration information (e.g. FTP passwords) is not stored in demo mode until the key has been set. These configuration entries are not used in demo mode.

Appendix A – Sample Authorized Key Custodian Form

All Company staff that hold responsible authorized positions where they manage or handle encryption keys must sign the following document.

As a condition of continued employment with Company, and as an employee that has access to key management tools and equipment, you are obligated to sign the following to indicate acceptance of your responsibility.

The signatory of this document is in full employment with Company on the date shown below and has been afforded access to key management devices, software and equipment, and hereby agrees that, he or she

- Has read and understood the policies and procedures associated with key management and agrees to comply with them to the best of his/her ability, and has been trained in security awareness and has had the ability to raise questions and has had those questions answered satisfactory.
- Understands that non-compliance with the key management procedures can lead to disciplinary action including termination and prosecution.
Exceptions to compliance only occur where such compliance would violate local, state, or federal law, or where a senior officer of the company or law enforcement officer has given prior authorization.
- Agrees to never divulge to any third party any key management or related security systems, passwords, processes, security hardware or secrets associated with the Company systems, unless authorized by an officer of the Company or required to do so by law enforcement officers
- Agrees to report promptly and in full to the correct personnel, any suspicious activity including but not limited to key compromise or suspected key compromise. Suspicious activity can include: signs of unauthorized equipment usage during evenings and weekends, phone requests from unidentifiable callers for access to secure information, unidentifiable files found on file servers, and unusual activity recorded in log files.

I agree to the above and understand that this original copy will be held on my personnel record and kept by the company indefinitely.

Signed: [_____] Witnessed:[_____]

Print Name: [_____] [_____]

Date: [_____]