Encrypting Software for Transmission to NIST

1. Scope

NIST requires that all software submitted by the participants be signed and encrypted. Signing is done with the participant's private key, and encrypting is done with the NIST project public key, which is published at http://www.nist.gov/itl/iad/ig/encrypt.cfm. NIST will validate all submitted materials using the participant's public key, and the authenticity of that key will be verified using the key fingerprint. This fingerprint must be submitted to NIST as part of the signed participant agreement.

By encrypting the submissions, we ensure privacy; by signing the submission, we ensure authenticity (the software actually belongs to the submitter). NIST will not take ownership of any submissions that are not signed and encrypted.

All cryptographic operations (signing and encrypting) shall be performed with software that implements the OpenPGP standard, as described in Internet RFC 4880. The freely available Gnu Privacy Guard (GPG) software, available at www.gnupg.org, is one such implementation.

2. Submission of software to NIST

NIST requires that all software submitted by the participants be signed and encrypted. Two keys pairs are needed:

- Signing is done with the software provider's private key, and
- Encryption is done with the NIST project public key, which is available at http://www.nist.gov/itl/iad/ig/encrypt.cfm

2.1. Project Specific Parameters

The values for the project specific parameters (*ProjectName*, *ProjectPublicKey*, and *ProjectEmail*) mentioned in this document are found at http://www.nist.gov/itl/iad/ig/encrypt.cfm

2.2. Creating participant cryptographic key pair

The steps below show how to create a public/private key pair and fingerprint using the GPG software.

1	**Generate your key pair**	gpg --gen-key <press Enter for the default key type> <Choose a key size of 2048> <Choose a non-expiring key> <Press 'y'> <Enter Real Name> <Enter **Participant email address**; this is the key identity> <Enter an optional comment> <Press 'O' to continue> <Enter a passphrase for the secret (private) key> Once the key pair is generated, the public key must be exported in the proper format to be sent to NIST. It is crucial that the applicant protect the private key by choosing a strong password that is not shared.
2	**Export your public key**	gpg --armor --output for<*ProjectName*>.gpg --export <*ParticipantEmail*> Where <*ParticipantEmail*> is the address used in step 1. This address is the key identity. The participant public key will be saved into the file named 'for<*ProjectName*>.gpg'.
3	**Email your public key**	The file containing the participant public key must be sent to the NIST Project Test Liaison at <*ProjectEmail*>
4	**Generate the key fingerprint**	gpg --fingerprint <*ParticipantEmail*> The key fingerprint will be shown in the output as a set of hex digits. The fingerprint must be copied onto the project participant agreement sent to NIST.

2.3. Importing the NIST Project Public Key

The next series of step show how the participant will import the NIST project public key and authenticate it using the key fingerprint. The NIST project specific public key is available at http://www.nist.gov/itl/iad/ig/encrypt.cfm. The following example assumes the NIST key is saved into a file named <*ProjectPublicKey*>.gpg.

5	Import the NIST project public key, contained in the file called <*ProjectPublicKey*>.gpg	gpg --import <*ProjectPublicKey*>.gpg The output should be similar to: key 856B9B28: public key "Project Test Liaison (Project Test Liaison Key) <*ProjectEmail*> imported
6	Authenticate the NIST key	gpg --fingerprint <*ProjectEmail*> The key fingerprint will be shown in the output as a set of hex digits. These digits must be the same as our project public key fingerprint which is printed on the participant agreement. If the fingerprints do not match, contact NIST and do not use the key for encrypting.
7	Optionally, the participant may want to assign a level of trust to the NIST public key.	gpg --edit-key <*ProjectEmail*> <Enter 'trust' at the Command prompt> <Choose a trust level; 3 is a good choice> <Enter 'y' to approve the trust selection, if asked> <Enter 'q' to quit>

2.4. Encryption and Signing

By following the instructions in Sections 2.2 and 2.3, the keys have been generated and exchanged between NIST and the participant. From this point forward, all software submissions must be signed and encrypted. In addition, general email communication can be encrypted and signed, if desired. This section shows how to encrypt and sign a file to be sent to NIST.

8	Encrypt and sign the file to be submitted to NIST	gpg --default-key <*ParticipantEmail*> --output <*filename*>.gpg --encrypt --recipient <*ProjectEmail*> --sign <*filename*> <*ParticipantEmail*> is the key identity chosen when the key pair was created <*filename*> is the file to be submitted to NIST <Enter the passphrase chosen for the private key>

NIST accepts no responsibility for unencrypted materials sent to NIST.