ifRSA

Implemented By

- roRSA

Supported Methods

- SetPrivateKey(keyFileName as String) as Integer
- SetPublicKey(keyFileName as String) as Integer
- SetDigestAlgorithm(digestAlgorithm as String) as Boolean
- Sign(digest as Object) as Object
- Verify(digest as Object, signature as Object) as Integer

Description of Methods

SetPrivateKey(keyFileName as String) as Integer

Specify the private key to use for signing. The file name should specify a path, either in the package or a temp path.

Returns 1 if the key is valid, or 0 if the file does not contain a valid key, or -1 if the file was not found.

SetPublicKey(keyFileName as String) as Integer

Specify the public key to use for verification. The file name should specify a path, either in the package or a temp path.

Returns 1 if the key is valid, or 0 if the file does not contain a valid key, or -1 if the file was not found.

SetDigestAlgorithm(digestAlgorithm as String) as Boolean

Specify the digest algorithm to use for signing and verification. This should be an openssl string. Common digest algorithms are "sha1", "ripemd160", and "md5".

Returns true if the algorithm was set, false if the string was not recognized.

Sign(digest as Object) as Object

digest should be a roByteArray to be signed.

Returns a roByteArray containing the signature, or invalid if an error occurred. Errors will be printed in the BrightScript console. Some possible errors:

- digest is empty
- SetPrivateKey() was not yet called
- out of memory
- the digest could not be signed

If the digest algorithm is not set (using SetDigestAlgorithm) before calling Sign(), the digest is not encapsulated. This would be equivalent to simply calling the openssl function RSA_private_encrypt().

Verify(digest as Object, signature as Object) as Integer

Verifies the given digest and signature. Both digest and signature should be roByteArrays.

Returns 1 if the signature matches. Errors:

- Returns -1 if SetPublicKey() was not yet called
- Returns -2 if digest is empty
- Returns -3 if there is not enough memory
- Returns 0 if the signature does not match

If the digest algorithm is not set (using SetDigestAlgorithm) before calling Verify(), the digest associated with the signature is not expected to be encapsulated. This would be equivalent to simply calling the openssl function RSA_public_decrypt(signature) and then comparing the result with the digest.