RE002 Requirements for the Module codecs_lib.vigenere

Conventions

Requirements listed in this document are constructed according to the following structure:

Requirement ID: REQ-UVW-XYZ

Title: Title / name of the requirement

Description: Descriprion / definition of the requirement

Verification Method: I / A / T / D

The requirement ID starts with the fixed prefix 'REQ'. The prefix is followed by 3 letters abbreviation (in here 'UVW'), which defines the requiement type - e.g. 'FUN' for a functional and capability requirement, 'AWM' for an alarm, warnings and operator messages, etc. The last part of the ID is a 3-digits *hexadecimal* number (0..9|A..F), with the first digit identifing the module, the second digit identifing a class / function, and the last digit - the requirement ordering number for this object. E.g. 'REQ-FUN-112'. Each requirement type has its own counter, thus 'REQ-FUN-112' and 'REQ-AWN-112' requirements are different entities, but they refer to the same object (class or function) within the same module.

The verification method for a requirement is given by a single letter according to the table below:

Term	Definition
Inspection (I)	Control or visual verification
Analysis (A)	Verification based upon analytical evidences
Test (T)	Verification of quantitative characteristics with quantitative measurement
Demonstration (D)	Verification of operational characteristics without quantitative measurement

Functional and capability requirements

Requirement ID: REQ-FUN-200

Title: Vignere codec implementation

Description: The module must provide a proper implementation of the algorithm for encoding and decoding based on the Vigenere substitution algorithm.

Verification Method: A

Requirement ID: REQ-FUN-201

Title: Module interface

Description: The module should provide four class instance methods:

- For setting the passphrase
- For instructing the codec to 'reset' the internal index to the start of the stored passphrase
- For encoding an arbitrary Unicode string into a bytestring
- For decoding an arbitrary bytestring or bytes array into an Unicode string

Verification Method: A

Requirement ID: REQ-FUN-202

Title: Support for a continuous data feed.

Description: Each encoded or decoded byte should shift the internal index within the byte-encoded passphrase, and that index shouldn't be reset to the beginning of the passphrase unless requested directly via the specific method call, or by setting a new passphrase. Thus, this codec can be used for the encoding and decoding of a continuous data feed, e.g. of an input stream.

Verification Method: T

Requirement ID: REQ-FUN-210

Title: Data encoding

Description: The method for encoding of the data should accept an arbitrary Unicode string as its argument, encode it into a bytestring using the specified Unicode codec (optonal, default is UTF-8), and apply the Vignere substitution algorithm modulo 256 to each consecutive byte in the bytestring, and return the result as a bytestring.

Verification Method: T

Requirement ID: REQ-FUN-220

Title: Data decoding

Description: The method for decoding of the data should accept an arbitrary byte string (type **bytes**) or byte array (type **bytearray**) object as its argument, to which the Vignere substitution algorithm modulo 256 should be applied on per byte manner, and the result should be converted into a Unicode string using the specified Unicode codec (optonal, default is UTF-8).

Verification Method: T

Requirement ID: REQ-FUN-230

Title: Setting a passphrase

Description: The passphrase for encoding and decoding can be set as a bytestring, a bytes array or as an Unicode string, in which case the UTF-8 encoding should be used to convert the Unicode passphrase into a stored bytestring or bytes array value.

Verification Method: T

Alarms, warnings and operator messages

Requirement ID: REQ-AWM-200

Title: Improper input value of a proper type raises an exception

Description: A passed string (Unicode) argument cannot be encoded into a byte string using the specified codec, OR a decoded byte string / bytes array cannot be decoded into a string using the specificed codec, OR such codec is not registered. **ValueError** exception or its sub-class must be raised.

Verification Method: T

Requirement ID: REQ-AWM-201

Title: Improper type of the optional codec argument

Description: A **TypeError** or its sub-class exception should be raised if the optional codec agrument for the encoding or decoding method is not a string.

Verification Method: T

Requirement ID: REQ-AWM-202

Title: Encoding or decoding without a passphrase set

Description: An attempt to encode or decode data without a set passphrase should result in a run-time exception.

Verification Method: T

Requirement ID: REQ-AWM-210

Title: Improper input type for encoding

Description: A non-string type passed as the mandatory value into the encoding method should result in an exception of **TypeError** or its sub-class.

Verification Method: T

Requirement ID: REQ-AWM-220

Title: Improper input type for decoding

Description: Any type except for bytestring or bytesarray passed as the mandatory value into the decoding method should result in an exception of **TypeError** or its sub-class.

Verification Method: T

Requirement ID: REQ-AWM-230

Title: Improper input type for passphrase

Description: Any type except for string, bytestring or bytesarray passed into the 'set passphrase'' method should result in an exception of **TypeError** or its sub-class.

Verification Method: T