aws-encryption-sdk-python

Release 1.4.1

Aug 26, 2020

Contents

The AWS Encryption SDK for Python provides a fully compliant, native Python implementation of the AWS Encryption SDK.

The latest full documentation can be found at Read the Docs.

Find us on GitHub.

Security issue notifications

CHAPTER 1

Getting Started

1.1 Required Prerequisites

- Python 2.7 or 3.5+
- cryptography >= 1.8.1
- boto3
- attrs

1.2 Installation

Note: If you have not already installed cryptography, you might need to install additional prerequisites as detailed in the cryptography installation guide for your operating system.

```
$ pip install aws-encryption-sdk
```

1.3 Concepts

There are three main concepts that are helpful to understand when using the AWS Encryption SDK.

For further information, see the AWS Encryption SDK developer guide concepts.

1.3.1 Cryptographic Materials Managers

The cryptographic materials manager (CMM) assembles the cryptographic materials that are used to encrypt and decrypt data.

For more details, see the AWS Encryption SDK developer guide cryptographic materials manager concept.

1.3.2 Keyrings

A keyring generates, encrypts, and decrypts data keys.

For more details, see the AWS Encryption SDK developer guide keyring concept.

1.3.3 Data Keys

A data key is an encryption key that the AWS Encryption SDK uses to encrypt your data.

For more details, see the AWS Encryption SDK developer guide data key concept.

CHAPTER 2

Usage

For examples of how to use these concepts to accomplish different tasks, see our examples.

2.1 Performance Considerations

Adjusting the frame size can significantly improve the performance of encrypt/decrypt operations with this library.

Processing each frame in a framed message involves a certain amount of overhead. If you are encrypting a large file, increasing the frame size can offer potentially significant performance gains. We recommend that you tune these values to your use-case in order to obtain peak performance.

CHAPTER 3

Modules

aws_encryption_sdk	High level AWS Encryption SDK client functions.
aws_encryption_sdk.exceptions	Contains exception classes for AWS Encryption SDK.
aws_encryption_sdk.identifiers	AWS Encryption SDK native data structures for defining implementation-specific characteristics.
aws_encryption_sdk.caches	Common functions and structures for use in cryptographic materials caches.
aws_encryption_sdk.caches.base	Base class interface for caches for use with caching crypto material managers.
aws_encryption_sdk.caches.local	Local, in-memory, LRU, cryptographic materials cache for use with caching cryptographic materials providers.
aws_encryption_sdk.caches.null	Null cache: a cache which does not cache.
aws_encryption_sdk.keyrings.base	Base class interface for Keyrings.
aws_encryption_sdk.keyrings.aws_kms	Keyring for use with AWS Key Management Service (KMS).
aws_encryption_sdk.keyrings.aws_kms.client_suppliers	AWS KMS client suppliers for use with AWS KMS keyring.
aws_encryption_sdk.keyrings.multi	Resources required for Multi Keyrings.
aws_encryption_sdk.keyrings.raw	Resources required for Raw Keyrings.
aws_encryption_sdk.key_providers.base	Base class interface for Master Key Providers.
aws_encryption_sdk.key_providers.kms	Master Key Providers for use with AWS KMS
aws_encryption_sdk.key_providers.raw	Resources required for Raw Master Keys.
aws_encryption_sdk.materials_managers	Primitive structures for use when interacting with crypto material managers.
aws_encryption_sdk.materials_managers.base	Base class interface for crypto material managers.
aws_encryption_sdk.materials_managers.caching	Caching crypto material manager.
aws_encryption_sdk.materials_managers.default	Default crypto material manager class.

Continued on next page

Table 1 – continued from previous page

aws_encryption_sdk.streaming_client	High level AWS Encryption SDK client for streaming objects.
aws_encryption_sdk.structures	Public data structures for aws_encryption_sdk.
aws_encryption_sdk.internal	Internal Implementation Details
aws_encryption_sdk.internal.crypto.authentication	Contains authentication primitives.
aws_encryption_sdk.internal.crypto.data_keys	Contains data key helper functions.
aws_encryption_sdk.internal.crypto.elliptic_curve	Contains elliptic curve functionality.
aws_encryption_sdk.internal.crypto.encryption	Contains encryption primitives and helper functions.
aws_encryption_sdk.internal.crypto.iv	Helper functions used for generating deterministic initialization vectors (IVs).
aws_encryption_sdk.internal.crypto.wrapping_keys	Contains wrapping key primitives.
aws_encryption_sdk.internal.defaults	Default values for AWS Encryption SDK.
aws_encryption_sdk.internal.formatting	Formatting functions for aws_encryption_sdk.
aws_encryption_sdk.internal.formatting.deserialize	Components for handling AWS Encryption SDK message deserialization.
aws_encryption_sdk.internal.formatting.encryption_context	Components for handling serialization and deserialization of encryption context data in AWS Encryption SDK messages.
aws_encryption_sdk.internal.formatting.serialize	Components for handling AWS Encryption SDK message serialization.
aws_encryption_sdk.internal.str_ops	Helper functions for consistently obtaining str and bytes objects in both Python2 and Python3.
aws_encryption_sdk.internal.structures	Public data structures for aws_encryption_sdk.
aws_encryption_sdk.internal.validators	Common `attrs` validators.
aws_encryption_sdk.internal.utils	Helper utility functions for AWS Encryption SDK.
aws_encryption_sdk.keyrings.aws_kms._client_cache	boto3 client cache for use by client suppliers.

3.1 aws_encryption_sdk

High level AWS Encryption SDK client functions.

Functions

decrypt(**kwargs)	Deserializes and decrypts provided ciphertext.
encrypt(**kwargs)	Encrypts and serializes provided plaintext.
stream(**kwargs)	Provides an `open()`-like interface to the streaming encryptor/decryptor classes.

aws_encryption_sdk.**encrypt**(***kwargs*)
Encrypts and serializes provided plaintext.

Note: When using this function, the entire ciphertext message is encrypted into memory before returning any data. If streaming is desired, see *aws_encryption_sdk.stream*.

New in version 2.0.0: The *keyring* parameter.

New in version 2.0.0: For backwards compatibility, the new `CryptoResult` return value also unpacks like a 2-member tuple. This allows for backwards compatibility with the previous outputs so this change should not break any existing consumers.

```
>>> import aws_encryption_sdk
>>> from aws_encryption_sdk.keyrings.aws_kms import AwsKmsKeyring
>>> keyring = AwsKmsKeyring(
...     generator_key_id="arn:aws:kms:us-east-1:2222222222222:key/22222222-2222-
↪2222-2222-222222222222",
...     key_ids=["arn:aws:kms:us-east-1:3333333333333:key/33333333-3333-3333-3333-
↪333333333333"],
... )
>>> my_ciphertext, encryptor_header = aws_encryption_sdk.encrypt(
...     source=my_plaintext,
...     keyring=keyring,
>>> )
```

Parameters

- **config** (*aws_encryption_sdk.streaming_client.EncryptorConfig*) – Client configuration object (config or individual parameters required)
- **source** (*str, bytes, io.IOBase, or file*) – Source data to encrypt or decrypt
- **materials_manager** (*CryptoMaterialsManager*) – Cryptographic materials manager to use for encryption (either `materials_manager`, `keyring`, `key_provider` required)
- **keyring** (*Keyring*) – Keyring to use for encryption (either `materials_manager`, `keyring`, `key_provider` required)
- **key_provider** (*MasterKeyProvider*) – Master key provider to use for encryption (either `materials_manager`, `keyring`, `key_provider` required)
- **source_length** (*int*) – Length of source data (optional)

  Note: If source_length is not provided and unframed message is being written or read() is called, will attempt to seek() to the end of the stream and tell() to find the length of source data.

  Note: New in version 1.3.0.

  If *source_length* and *materials_manager* are both provided, the total plaintext bytes encrypted will not be allowed to exceed *source_length*. To maintain backwards compatibility, this is not enforced if a *key_provider* is provided.

- **encryption_context** (*dict*) – Dictionary defining encryption context
- **algorithm** (*aws_encryption_sdk.identifiers.Algorithm*) – Algorithm to use for encryption

- **frame_length** (*int*) – Frame length in bytes

Returns Encrypted message and message metadata (header)

Return type *CryptoResult*

aws_encryption_sdk.**decrypt**(***kwargs*)

Deserializes and decrypts provided ciphertext.

Note: When using this function, the entire ciphertext message is decrypted into memory before returning any data. If streaming is desired, see *aws_encryption_sdk.stream*.

New in version 2.0.0: The *keyring* parameter.

New in version 2.0.0: For backwards compatibility, the new `CryptoResult` return value also unpacks like a 2-member tuple. This allows for backwards compatibility with the previous outputs so this change should not break any existing consumers.

```
>>> import aws_encryption_sdk
>>> from aws_encryption_sdk.keyrings.aws_kms import AwsKmsKeyring
>>> keyring = AwsKmsKeyring(
...     generator_key_id="arn:aws:kms:us-east-1:2222222222222:key/22222222-2222-
↪2222-2222-222222222222",
...     key_ids=["arn:aws:kms:us-east-1:3333333333333:key/33333333-3333-3333-3333-
↪333333333333"],
... )
>>> my_ciphertext, decryptor_header = aws_encryption_sdk.decrypt(
...     source=my_ciphertext,
...     keyring=keyring,
... )
```

Parameters

- **config** (*aws_encryption_sdk.streaming_client.DecryptorConfig*) – Client configuration object (config or individual parameters required)
- **source** (*str, bytes, io.IOBase, or file*) – Source data to encrypt or decrypt
- **materials_manager** (*CryptoMaterialsManager*) – Cryptographic materials manager to use for encryption (either `materials_manager`, `keyring`, `key_provider` required)
- **keyring** (*Keyring*) – Keyring to use for encryption (either `materials_manager`, `keyring`, `key_provider` required)
- **key_provider** (*MasterKeyProvider*) – Master key provider to use for encryption (either `materials_manager`, `keyring`, `key_provider` required)
- **source_length** (*int*) – Length of source data (optional)

  Note: If source_length is not provided and read() is called, will attempt to seek() to the end of the stream and tell() to find the length of source data.

- **max_body_length** (*int*) – Maximum frame size (or content length for non-framed messages) in bytes to read from ciphertext message.

Returns Decrypted plaintext and message metadata (header)

Return type *CryptoResult*

aws_encryption_sdk.**stream**(***kwargs*)

Provides an `open()`-like interface to the streaming encryptor/decryptor classes.

> **Warning:** Take care when decrypting framed messages with large frame length and large non-framed messages. In order to protect the authenticity of the encrypted data, no plaintext is returned until it has been authenticated. Because of this, potentially large amounts of data may be read into memory. In the case of framed messages, the entire contents of each frame are read into memory and authenticated before returning any plaintext. In the case of non-framed messages, the entire message is read into memory and authenticated before returning any plaintext. The authenticated plaintext is held in memory until it is requested.

Note: Consequently, keep the above decrypting consideration in mind when encrypting messages to ensure that issues are not encountered when decrypting those messages.

```
>>> import aws_encryption_sdk
>>> from aws_encryption_sdk.keyrings.aws_kms import AwsKmsKeyring
>>> keyring = AwsKmsKeyring(
...     generator_key_id="arn:aws:kms:us-east-1:2222222222222:key/22222222-2222-
↪2222-2222-222222222222",
...     key_ids=["arn:aws:kms:us-east-1:3333333333333:key/33333333-3333-3333-3333-
↪333333333333"],
... )
>>> plaintext_filename = 'my-secret-data.dat'
>>> ciphertext_filename = 'my-encrypted-data.ct'
>>> with open(plaintext_filename, 'rb') as pt_file, open(ciphertext_filename, 'wb
↪') as ct_file:
...      with aws_encryption_sdk.stream(
...         mode='e',
...         source=pt_file,
...         keyring=keyring,
...     ) as encryptor:
...         for chunk in encryptor:
...              ct_file.write(chunk)
>>> new_plaintext_filename = 'my-decrypted-data.dat'
>>> with open(ciphertext_filename, 'rb') as ct_file, open(new_plaintext_filename,
↪'wb') as pt_file:
...     with aws_encryption_sdk.stream(
...         mode='d',
...         source=ct_file,
...         keyring=keyring,
...     ) as decryptor:
...         for chunk in decryptor:
...             pt_file.write(chunk)
```

Parameters

- **mode** (*str*) – Type of streaming client to return (e/encrypt: encryptor, d/decrypt: decryptor)
- ****kwargs** – All other parameters provided are passed to the appropriate Streaming client

Returns Streaming Encryptor or Decryptor, as requested

Return type *aws_encryption_sdk.streaming_client.StreamEncryptor* or *aws_encryption_sdk.streaming_client.StreamDecryptor*

Raises **ValueError** – if supplied with an unsupported mode value

3.2 aws_encryption_sdk.exceptions

Contains exception classes for AWS Encryption SDK.

Exceptions

AWSEncryptionSDKClientError	General exception class for AWS Encryption SDK.
ActionNotAllowedError	Exception class for errors encountered when attempting to perform unallowed actions.
CacheError	General exception class for materials caches.
CacheKeyError	Exception class for *CryptoCache* key misses.
ConfigMismatchError	Exception class for errors encountered when the wrong type of config is passed to an object.
CustomMaximumValueExceeded	Exception class for use when values are found which exceed user-defined custom maximum values.
DecryptKeyError	Exception class for errors encountered when MasterKeys try to decrypt data keys.
EncryptKeyError	Exception class for errors encountered when MasterKeys try to encrypt data keys.
GenerateKeyError	Exception class for errors encountered when MasterKeys try to generate data keys.
IncorrectMasterKeyError	Exception class for operations attempted against the incorrect Master Key.
InvalidAlgorithmError	Exception class for Invalid Algorithm definitions.
InvalidCryptographicMaterialsError	Exception class for errors encountered when attempting to validate cryptographic materials.
InvalidDataKeyError	Exception class for Invalid Data Keys.
InvalidKeyIdError	Exception class for Invalid Key IDs.
InvalidProviderIdError	Exception class for Invalid Provider IDs.
MasterKeyError	Exception class for Master Keys.
MasterKeyProviderError	Exception class for Master Key Providers.
NotSupportedError	Exception class for unsupported identities or operations.
SerializationError	Exception class for serialization/deserialization errors.
SignatureKeyError	Exception class for errors encountered with signing or verification keys.
UnknownIdentityError	Exception class for unknown identity errors.
UnknownRegionError	Exception class for errors encountered when attempting to process unknown regions or region names.

exception aws_encryption_sdk.exceptions.**AWSEncryptionSDKClientError**

Bases: exceptions.Exception

General exception class for AWS Encryption SDK.

exception aws_encryption_sdk.exceptions.**ActionNotAllowedError**

Bases: *aws_encryption_sdk.exceptions.AWSEncryptionSDKClientError*

Exception class for errors encountered when attempting to perform unallowed actions.

exception aws_encryption_sdk.exceptions.**CacheError**
Bases: *aws_encryption_sdk.exceptions.AWSEncryptionSDKClientError*

General exception class for materials caches.

New in version 1.3.0.

exception aws_encryption_sdk.exceptions.**CacheKeyError**
Bases: *aws_encryption_sdk.exceptions.CacheError*

Exception class for *CryptoCache* key misses.

New in version 1.3.0.

This exception is meant to mirror *KeyError* but in the context of a *CryptoCache*.

exception aws_encryption_sdk.exceptions.**ConfigMismatchError**
Bases: *aws_encryption_sdk.exceptions.AWSEncryptionSDKClientError*

Exception class for errors encountered when the wrong type of config is passed to an object.

exception aws_encryption_sdk.exceptions.**CustomMaximumValueExceeded**
Bases: *aws_encryption_sdk.exceptions.SerializationError*

Exception class for use when values are found which exceed user-defined custom maximum values.

exception aws_encryption_sdk.exceptions.**DecryptKeyError**
Bases: *aws_encryption_sdk.exceptions.AWSEncryptionSDKClientError*

Exception class for errors encountered when MasterKeys try to decrypt data keys.

exception aws_encryption_sdk.exceptions.**EncryptKeyError**
Bases: *aws_encryption_sdk.exceptions.AWSEncryptionSDKClientError*

Exception class for errors encountered when MasterKeys try to encrypt data keys.

exception aws_encryption_sdk.exceptions.**GenerateKeyError**
Bases: *aws_encryption_sdk.exceptions.AWSEncryptionSDKClientError*

Exception class for errors encountered when MasterKeys try to generate data keys.

exception aws_encryption_sdk.exceptions.**IncorrectMasterKeyError**
Bases: *aws_encryption_sdk.exceptions.AWSEncryptionSDKClientError*

Exception class for operations attempted against the incorrect Master Key.

exception aws_encryption_sdk.exceptions.**InvalidAlgorithmError**
Bases: *aws_encryption_sdk.exceptions.AWSEncryptionSDKClientError*

Exception class for Invalid Algorithm definitions.

exception aws_encryption_sdk.exceptions.**InvalidCryptographicMaterialsError**
Bases: *aws_encryption_sdk.exceptions.AWSEncryptionSDKClientError*

Exception class for errors encountered when attempting to validate cryptographic materials.

New in version 1.5.0.

exception aws_encryption_sdk.exceptions.**InvalidDataKeyError**
Bases: *aws_encryption_sdk.exceptions.AWSEncryptionSDKClientError*

Exception class for Invalid Data Keys.

exception aws_encryption_sdk.exceptions.**InvalidKeyIdError**
Bases: *aws_encryption_sdk.exceptions.AWSEncryptionSDKClientError*

Exception class for Invalid Key IDs.

exception aws_encryption_sdk.exceptions.**InvalidProviderIdError**
Bases: *aws_encryption_sdk.exceptions.AWSEncryptionSDKClientError*

Exception class for Invalid Provider IDs.

exception aws_encryption_sdk.exceptions.**MasterKeyError**
Bases: *aws_encryption_sdk.exceptions.AWSEncryptionSDKClientError*

Exception class for Master Keys.

exception aws_encryption_sdk.exceptions.**MasterKeyProviderError**
Bases: *aws_encryption_sdk.exceptions.AWSEncryptionSDKClientError*

Exception class for Master Key Providers.

exception aws_encryption_sdk.exceptions.**NotSupportedError**
Bases: *aws_encryption_sdk.exceptions.AWSEncryptionSDKClientError*

Exception class for unsupported identities or operations.

exception aws_encryption_sdk.exceptions.**SerializationError**
Bases: *aws_encryption_sdk.exceptions.AWSEncryptionSDKClientError*

Exception class for serialization/deserialization errors.

exception aws_encryption_sdk.exceptions.**SignatureKeyError**
Bases: *aws_encryption_sdk.exceptions.AWSEncryptionSDKClientError*

Exception class for errors encountered with signing or verification keys.

New in version 1.5.0.

exception aws_encryption_sdk.exceptions.**UnknownIdentityError**
Bases: *aws_encryption_sdk.exceptions.AWSEncryptionSDKClientError*

Exception class for unknown identity errors.

exception aws_encryption_sdk.exceptions.**UnknownRegionError**
Bases: *aws_encryption_sdk.exceptions.AWSEncryptionSDKClientError*

Exception class for errors encountered when attempting to process unknown regions or region names.

3.3 aws_encryption_sdk.identifiers

AWS Encryption SDK native data structures for defining implementation-specific characteristics.

Classes

Algorithm	alias of *aws_encryption_sdk.identifiers.AlgorithmSuite*
AlgorithmSuite(algorithm_id, encryption[, ...])	Static combinations of encryption, KDF, and authentication algorithms.
AuthenticationSuite(algorithm, ...)	Static definition of authentication algorithm details.

Continued on next page

Table 4 – continued from previous page

ContentAADString	Body Additional Authenticated Data values for building the AAD for a message body.
ContentType	Type of content framing contained in message.
EncryptionKeyType	Identifies raw encryption key type.
EncryptionSuite(algorithm, mode, ... [, ...])	Static definition of encryption algorithm details.
EncryptionType	Identifies symmetric vs asymmetric encryption.
KDFSuite(algorithm, input_length, hash_algorithm)	Static definition of key derivation algorithm details.
ObjectType	Valid Type values per the AWS Encryption SDK message format.
SequenceIdentifier	Identifiers for specific sequence frames.
SerializationVersion	Valid Versions of AWS Encryption SDK message format.
WrappingAlgorithm(encryption_type, ...)	Wrapping Algorithms for use by RawMasterKey objects.

aws_encryption_sdk.identifiers.**Algorithm**
alias of *aws_encryption_sdk.identifiers.AlgorithmSuite*

class aws_encryption_sdk.identifiers.**AlgorithmSuite**(*algorithm_id*, *encryption*, *kdf=<KDFSuite.NONE: (None, None, None)>*, *authentication=<AuthenticationSuite.NONE: (None, None, 0)>*, *allowed=True*)

Bases: enum.Enum

Static combinations of encryption, KDF, and authentication algorithms.

Warning: No AlgorithmSuites except those defined here are supported.

Parameters

- **algorithm_id** (*int*) – KMS Encryption Algorithm ID
- **encryption_suite** (aws_encryption_sdk.identifiers.EncryptionSuite) – EncryptionSuite to use with this AlgorithmSuite
- **kdf_suite** (aws_encryption_sdk.identifiers.KDFSuite) – KDFSuite to use with this AlgorithmSuite
- **authentication_suite** (aws_encryption_sdk.identifiers.AuthenticationSuite) – AuthenticationSuite to use with this AlgorithmSuite

Prepare a new AlgorithmSuite.

id_as_bytes()
Return the algorithm suite ID as a 2-byte array

kdf_input_len
Determine the correct KDF input value length for this algorithm suite.

safe_to_cache()
Determine whether encryption materials for this algorithm suite should be cached.

class aws_encryption_sdk.identifiers.**AuthenticationSuite**(*algorithm*, *hash_algorithm*, *signature_length*)

Bases: enum.Enum

Static definition of authentication algorithm details.

> **Warning:** These members must only be used as part of an AlgorithmSuite.

Parameters

- **algorithm** (*may vary (currently only ECC curve object)*) – Information needed by signing algorithm to define behavior
- **hash_algorithm** (*cryptography.io hashes object*) – Hash algorithm to use in signature
- **signature_lenth** (*int*) – Number of bytes in signature

Prepare a new AuthenticationSuite.

class aws_encryption_sdk.identifiers.**ContentAADString**

Bases: enum.Enum

Body Additional Authenticated Data values for building the AAD for a message body.

class aws_encryption_sdk.identifiers.**ContentType**

Bases: enum.Enum

Type of content framing contained in message.

class aws_encryption_sdk.identifiers.**EncryptionKeyType**

Bases: enum.Enum

Identifies raw encryption key type. Used to identify key capabilities for WrappingAlgorithm.

class aws_encryption_sdk.identifiers.**EncryptionSuite**(*algorithm*, *mode*, *data_key_length*, *iv_length*, *auth_length*, *auth_key_length=0*)

Bases: enum.Enum

Static definition of encryption algorithm details.

> **Warning:** These members must only be used as part of an AlgorithmSuite.

Parameters

- **algorithm** (*cryptography.io ciphers algorithm object*) – Encryption algorithm to use
- **mode** (*cryptography.io ciphers modes object*) – Encryption mode in which to operate
- **data_key_length** (*int*) – Number of bytes in envelope encryption data key
- **iv_length** (*int*) – Number of bytes in IV
- **auth_length** (*int*) – Number of bytes in auth data (tag)

- **auth_key_length** (*int*) – Number of bytes in auth key (not currently supported by any algorithms)

Prepare a new EncryptionSuite.

valid_kdf(*kdf*)
Determine whether a KDFSuite can be used with this EncryptionSuite.

Parameters **kdf** (aws_encryption_sdk.identifiers.KDFSuite) – KDFSuite to evaluate

Return type bool

class aws_encryption_sdk.identifiers.**EncryptionType**
Bases: enum.Enum

Identifies symmetric vs asymmetric encryption. Used to identify encryption type for WrappingAlgorithm.

class aws_encryption_sdk.identifiers.**KDFSuite**(*algorithm*, *input_length*, *hash_algorithm*)
Bases: enum.Enum

Static definition of key derivation algorithm details.

Warning: These members must only be used as part of an AlgorithmSuite.

Parameters

- **algorithm** (*cryptography.io KDF object*) – KDF algorithm to use
- **input_length** (*int*) – Number of bytes of input data to feed into KDF function
- **hash_algorithm** (*cryptography.io hashes object*) – Hash algorithm to use in KDF

Prepare a new KDFSuite.

input_length(*encryption*)
Determine the correct KDF input value length for this KDFSuite when used with a specific EncryptionSuite.

Parameters **encryption** (aws_encryption_sdk.identifiers.EncryptionSuite) – EncryptionSuite to use

Return type int

class aws_encryption_sdk.identifiers.**ObjectType**
Bases: enum.Enum

Valid Type values per the AWS Encryption SDK message format.

class aws_encryption_sdk.identifiers.**SequenceIdentifier**
Bases: enum.Enum

Identifiers for specific sequence frames.

class aws_encryption_sdk.identifiers.**SerializationVersion**
Bases: enum.Enum

Valid Versions of AWS Encryption SDK message format.

class aws_encryption_sdk.identifiers.**WrappingAlgorithm**(*encryption_type*, *algorithm*, *padding_type*, *padding_algorithm*, *padding_mgf*)

Bases: enum.Enum

Wrapping Algorithms for use by RawMasterKey objects.

Parameters

- **algorithm** (*aws_encryption_sdk.identifiers.Algorithm*) – Encryption algorithm to use for encryption of data keys
- **padding_type** – Padding type to use for encryption of data keys
- **padding_algorithm** – Padding algorithm to use for encryption of data keys
- **padding_mgf** – Padding MGF to use for encryption of data keys

Prepares new WrappingAlgorithm.

3.4 aws_encryption_sdk.caches

Common functions and structures for use in cryptographic materials caches.

New in version 1.3.0.

Functions

build_decryption_materials_cache_key(...)	Generates a cache key for a decrypt request.
build_encryption_materials_cache_key(...)	Generates a cache key for an encrypt request.

Classes

CryptoMaterialsCacheEntry(cache_key, value)	Value and metadata store for cryptographic materials cache entries.
CryptoMaterialsCacheEntryHints([lifetime])	Optional metadata to associate with cryptographic materials cache entries.

class aws_encryption_sdk.caches.**CryptoMaterialsCacheEntry**(*cache_key*, *value*, *hints=NOTHING*)

Bases: object

Value and metadata store for cryptographic materials cache entries.

Parameters

- **cache_key** (*bytes*) – Identifier for entries in cache
- **value** – Value to store in cache entry
- **hints** (aws_encryption_sdk.caches.CryptoMaterialsCacheEntryHints) – Metadata to associate with entry (optional)

age

Returns this entry's current age in seconds.

Return type float

invalidate()
Marks a cache entry as invalidated.

is_too_old()
Determines if if this entry's lifetime has passed.

Return type bool

class aws_encryption_sdk.caches.**CryptoMaterialsCacheEntryHints**(*lifetime=None*)
Bases: object

Optional metadata to associate with cryptographic materials cache entries.

Parameters **lifetime** (*float*) – Number of seconds to retain entry in cache (optional)

aws_encryption_sdk.caches.**build_decryption_materials_cache_key**(*partition*, *request*)
Generates a cache key for a decrypt request.

Parameters

- **partition** (*bytes*) – Partition name for which to generate key
- **request** (aws_encryption_sdk.materials_managers.DecryptionMaterialsRequest) – Request for which to generate key

Returns cache key

Return type bytes

aws_encryption_sdk.caches.**build_encryption_materials_cache_key**(*partition*, *request*)
Generates a cache key for an encrypt request.

Parameters

- **partition** (*bytes*) – Partition name for which to generate key
- **request** (aws_encryption_sdk.materials_managers.EncryptionMaterialsRequest) – Request for which to generate key

Returns cache key

Return type bytes

3.5 aws_encryption_sdk.caches.base

Base class interface for caches for use with caching crypto material managers.

Classes

CryptoMaterialsCache	Parent interface for crypto materials caches.

class aws_encryption_sdk.caches.base.**CryptoMaterialsCache**
Bases: object

Parent interface for crypto materials caches.

New in version 1.3.0.

get_decryption_materials(*cache_key*)
Locates exactly one available decryption materials cache entry for the specified cache_key.

Parameters **cache_key** (*bytes*) – Cache ID for which to locate cache entries

Return type aws_encryption_sdk.caches.CryptoCacheEntry

Raises ***CacheKeyError*** – if no values found in cache for cache_key

get_encryption_materials(*cache_key*, *plaintext_length*)
Locates exactly one available encryption materials cache entry for the specified cache_key, incrementing the entry's usage stats prior to returning it to the caller.

Parameters

- **cache_key** (*bytes*) – Cache ID for which to locate cache entries
- **plaintext_length** (*int*) – Bytes to be encrypted by the encryption materials

Return type aws_encryption_sdk.caches.CryptoCacheEntry

Raises ***CacheKeyError*** – if no values found in cache for cache_key

put_decryption_materials(*cache_key*, *decryption_materials*)
Adds decryption materials to the cache

Parameters

- **cache_key** (*bytes*) – Identifier for entries in cache
- **decryption_materials** (aws_encryption_sdk.materials_managers.DecryptionMaterials) – Decryption materials to add to cache

Return type aws_encryption_sdk.caches.CryptoCacheEntry

put_encryption_materials(*cache_key*, *encryption_materials*, *plaintext_length*, *entry_hints=None*)
Adds encryption materials to the cache.

Parameters

- **cache_key** (*bytes*) – Identifier for entries in cache
- **encryption_materials** (aws_encryption_sdk.materials_managers.EncryptionMaterials) – Encryption materials to add to cache
- **plaintext_length** (*int*) – Length of plaintext associated with this request to the cache
- **entry_hints** (*aws_encryption_sdk.caches.CryptoCacheEntryHints*) – Metadata to associate with entry (optional)

Return type aws_encryption_sdk.caches.CryptoCacheEntry

3.6 aws_encryption_sdk.caches.local

Local, in-memory, LRU, cryptographic materials cache for use with caching cryptographic materials providers.

Classes

LocalCryptoMaterialsCache(capacity)	Local, in-memory, LRU, cache for use with caching cryptographic materials providers.

class aws_encryption_sdk.caches.local.**LocalCryptoMaterialsCache**(*capacity*)

Bases: *aws_encryption_sdk.caches.base.CryptoMaterialsCache*

Local, in-memory, LRU, cache for use with caching cryptographic materials providers.

New in version 1.3.0.

Parameters **capacity** (*int*) – Maximum number of entries to retain in cache at once

clear()

Clears the cache.

get_decryption_materials(*cache_key*)

Locates exactly one available decryption materials cache entry for the specified cache_key.

Parameters **cache_key** (*bytes*) – Cache ID for which to locate cache entries

Return type *aws_encryption_sdk.caches.CryptoMaterialsCacheEntry*

Raises ***CacheKeyError*** – if no values found in cache for cache_key

get_encryption_materials(*cache_key*, *plaintext_length*)

Locates exactly one available encryption materials cache entry for the specified cache_key, incrementing the entry's usage stats prior to returning it to the caller.

Parameters

- **cache_key** (*bytes*) – Cache ID for which to locate cache entries
- **plaintext_length** (*int*) – Length of plaintext associated with this request to the cache

Return type *aws_encryption_sdk.caches.CryptoMaterialsCacheEntry*

Raises ***CacheKeyError*** – if no values found in cache for cache_key

put_decryption_materials(*cache_key*, *decryption_materials*)

Adds decryption materials to the cache

Parameters

- **cache_key** (*bytes*) – Identifier for entries in cache
- **decryption_materials** (aws_encryption_sdk.materials_managers.DecryptionMaterials) – Decryption materials to add to cache

Return type *aws_encryption_sdk.caches.CryptoMaterialsCacheEntry*

put_encryption_materials(*cache_key*, *encryption_materials*, *plaintext_length*, *entry_hints=None*)

Adds encryption materials to the cache.

Parameters

- **cache_key** (*bytes*) – Identifier for entries in cache
- **encryption_materials** (aws_encryption_sdk.materials_managers.EncryptionMaterials) – Encryption materials to add to cache
- **plaintext_length** (*int*) – Length of plaintext associated with this request to the cache

- **entry_hints** (*aws_encryption_sdk.caches.CryptoCacheEntryHints*) – Metadata to associate with entry (optional)

Return type *aws_encryption_sdk.caches.CryptoMaterialsCacheEntry*

remove(*value*)
Removes a value from the cache.

Parameters **value** (aws_encryption_sdk.caches.CryptoMaterialsCacheEntry) – Value to add to cache

Raises ***CacheKeyError*** – if value not found in cache

3.7 aws_encryption_sdk.caches.null

Null cache: a cache which does not cache.

Classes

NullCryptoMaterialsCache	Null cache: a cache which does not cache.

class aws_encryption_sdk.caches.null.**NullCryptoMaterialsCache**
Bases: *aws_encryption_sdk.caches.base.CryptoMaterialsCache*

Null cache: a cache which does not cache.

New in version 1.3.0.

get_decryption_materials(*cache_key*)
Always raises a CacheKeyError.

Parameters **cache_key** (*bytes*) – Cache ID for which to locate cache entries

Return type aws_encryption_sdk.caches.CryptoCacheEntry

Raises ***CacheKeyError*** – when called

get_encryption_materials(*cache_key*, *plaintext_length*)
Always raises a CacheKeyError.

Parameters

- **cache_key** (*bytes*) – Cache ID for which to locate cache entries
- **plaintext_length** (*int*) – Bytes to be encrypted by the encryption materials

Return type aws_encryption_sdk.caches.CryptoCacheEntry

Raises ***CacheKeyError*** – when called

put_decryption_materials(*cache_key*, *decryption_materials*)
Does not add decryption materials to the cache since there is no cache to which to add them.

Parameters

- **cache_key** (*bytes*) – Identifier for entries in cache
- **decryption_materials** (aws_encryption_sdk.materials_managers.DecryptionMaterials) – Decryption materials to add to cache

Return type *aws_encryption_sdk.caches.CryptoMaterialsCacheEntry*

put_encryption_materials(*cache_key*, *encryption_materials*, *plaintext_length*, *entry_hints=None*)

Does not add encryption materials to the cache since there is no cache to which to add them.

Parameters

- **cache_key** (*bytes*) – Identifier for entries in cache
- **encryption_materials** (aws_encryption_sdk.materials_managers.EncryptionMaterials) – Encryption materials to add to cache
- **plaintext_length** (*int*) – Length of plaintext associated with this request to the cache
- **entry_hints** (*aws_encryption_sdk.caches.CryptoCacheEntryHints*) – Metadata to associate with entry (optional)

Return type *aws_encryption_sdk.caches.CryptoMaterialsCacheEntry*

3.8 aws_encryption_sdk.keyrings.base

Base class interface for Keyrings.

Classes

Keyring	Parent interface for Keyring classes.

class aws_encryption_sdk.keyrings.base.**Keyring**

Bases: object

Parent interface for Keyring classes.

New in version 1.5.0.

on_decrypt(*decryption_materials*, *encrypted_data_keys*)

Attempt to decrypt the encrypted data keys.

Parameters

- **decryption_materials** (DecryptionMaterials) – Decryption materials for keyring to modify.
- **encrypted_data_keys** (*List[EncryptedDataKey]*) – List of encrypted data keys.

Returns Optionally modified decryption materials.

Return type *DecryptionMaterials*

Raises **NotImplementedError** – if method is not implemented

on_encrypt(*encryption_materials*)

Generate a data key if not present and encrypt it using any available wrapping key.

Parameters **encryption_materials** (EncryptionMaterials) – Encryption materials for keyring to modify.

Returns Optionally modified encryption materials.

Return type *EncryptionMaterials*

Raises `NotImplementedError` – if method is not implemented

3.9 aws_encryption_sdk.keyrings.aws_kms

Keyring for use with AWS Key Management Service (KMS).

New in version 2.0.0.

Classes

AwsKmsKeyring([client_supplier, ...])	Keyring that uses AWS Key Management Service (KMS) Customer Master Keys (CMKs) to manage wrapping keys.

class aws_encryption_sdk.keyrings.aws_kms.**AwsKmsKeyring**(*client_supplier=NOTHING*, *is_discovery=False*, *generator_key_id=None*, *key_ids=NOTHING*, *grant_tokens=NOTHING*)

Bases: *aws_encryption_sdk.keyrings.base.Keyring*

Keyring that uses AWS Key Management Service (KMS) Customer Master Keys (CMKs) to manage wrapping keys.

Set `generator_key_id` to require that the keyring use that CMK to generate the data key. If you do not set `generator_key_id`, the keyring will not generate a data key.

Set `key_ids` to specify additional CMKs that the keyring will use to encrypt the data key.

The keyring will attempt to use any CMKs identified by CMK ARN in either `generator_key_id` or `key_ids` on decrypt.

You can identify CMKs by any valid key ID for the keyring to use on encrypt, but for the keyring to attempt to use them on decrypt you MUST specify the CMK ARN.

If you specify `is_discovery=True` the keyring will be a KMS discovery keyring, doing nothing on encrypt and attempting to decrypt any AWS KMS-encrypted data key on decrypt.

Note: You must either set `is_discovery=True` or provide key IDs.

You can use the `ClientSupplier` to customize behavior further, such as to provide different credentials for different regions or to restrict which regions are allowed.

See the AWS KMS Keyring specification for more details.

New in version 2.0.0.

Parameters

- **client_supplier** (`ClientSupplier`) – Client supplier that provides AWS KMS clients (optional)
- **is_discovery** (*bool*) – Should this be a discovery keyring (optional)
- **generator_key_id** (*str*) – Key ID of AWS KMS CMK to use when generating data keys (optional)

- **key_ids** (*List[str]*) – Key IDs that will be used to encrypt and decrypt data keys (optional)
- **grant_tokens** (*List[str]*) – AWS KMS grant tokens to include in requests (optional)

on_decrypt(*decryption_materials*, *encrypted_data_keys*)
Attempt to decrypt the encrypted data keys.

Parameters

- **decryption_materials** (DecryptionMaterials) – Decryption materials for keyring to modify.
- **encrypted_data_keys** (*List[EncryptedDataKey]*) – List of encrypted data keys.

Returns Optionally modified decryption materials.

Return type *DecryptionMaterials*

on_encrypt(*encryption_materials*)
Generate a data key using generator keyring and encrypt it using any available wrapping key in any child keyring.

Parameters **encryption_materials** (EncryptionMaterials) – Encryption materials for keyring to modify.

Returns Optionally modified encryption materials.

Return type *EncryptionMaterials*

Raises ***EncryptKeyError*** – if unable to encrypt data key.

3.10 aws_encryption_sdk.keyrings.aws_kms.client_suppliers

AWS KMS client suppliers for use with AWS KMS keyring.

New in version 1.5.0.

Classes

AllowRegionsClientSupplier(allowed_regions)	AWS KMS client supplier that only supplies clients for the specified regions.
ClientSupplier	Base class for client suppliers.
DefaultClientSupplier([botocore_session, ...])	The default AWS KMS client supplier.
DenyRegionsClientSupplier(denied_regions[, ...])	AWS KMS client supplier that supplies clients for any region except for the specified regions.

class aws_encryption_sdk.keyrings.aws_kms.client_suppliers.**ClientSupplier**
Bases: object

Base class for client suppliers.

New in version 1.5.0.

aws_encryption_sdk.keyrings.aws_kms.client_suppliers.**ClientSupplierType**

alias of typing.Callable

class aws_encryption_sdk.keyrings.aws_kms.client_suppliers.**DefaultClientSupplier**(*botocore_sess*
client_config=

Bases: *aws_encryption_sdk.keyrings.aws_kms.client_suppliers.ClientSupplier*

The default AWS KMS client supplier. Creates and caches clients for any region.

New in version 1.5.0.

If you want clients to have special credentials or other configuration, you can provide those with custom botocore Session and/or Config instances.

```
from aws_encryption_sdk.keyrings.aws_kms.client_supplier import␣
↪DefaultClientSupplier
from botocore.session import Session
from botocore.config import Config

my_client_supplier = DefaultClientSupplier(
    botocore_session=Session(**_get_custom_credentials()),
    client_config=Config(connect_timeout=10),
)
```

Parameters

- **botocore_session** (*botocore.session.Session*) – Botocore session to use when creating clients (optional)
- **client_config** (*botocore.config.Config*) – Config to use when creating client (optional)

class aws_encryption_sdk.keyrings.aws_kms.client_suppliers.**AllowRegionsClientSupplier**(*allowe*
*client_*

Bases: *aws_encryption_sdk.keyrings.aws_kms.client_suppliers.ClientSupplier*

AWS KMS client supplier that only supplies clients for the specified regions.

New in version 1.5.0.

Parameters

- **allowed_regions** (*List[str]*) – Regions to allow
- **client_supplier** (ClientSupplier) – Client supplier to wrap (optional)

class aws_encryption_sdk.keyrings.aws_kms.client_suppliers.**DenyRegionsClientSupplier**(*denied_*
client_s

Bases: *aws_encryption_sdk.keyrings.aws_kms.client_suppliers.ClientSupplier*

AWS KMS client supplier that supplies clients for any region except for the specified regions.

New in version 1.5.0.

Parameters

- **denied_regions** (*List[str]*) – Regions to deny
- **client_supplier** (ClientSupplier) – Client supplier to wrap (optional)

3.11 aws_encryption_sdk.keyrings.multi

Resources required for Multi Keyrings.

Classes

MultiKeyring([generator, children])	Public class for Multi Keyring.

class aws_encryption_sdk.keyrings.multi.**MultiKeyring**(*generator=None*, *children=NOTHING*)

Bases: *aws_encryption_sdk.keyrings.base.Keyring*

Public class for Multi Keyring.

New in version 1.5.0.

Parameters

- **generator** (Keyring) – Generator keyring used to generate data encryption key (optional)
- **children** (*List[Keyring]*) – List of keyrings used to encrypt the data encryption key (optional)

Raises ***EncryptKeyError*** – if encryption of data key fails for any reason

on_decrypt(*decryption_materials*, *encrypted_data_keys*)

Attempt to decrypt the encrypted data keys.

Parameters

- **decryption_materials** (DecryptionMaterials) – Decryption materials for keyring to modify.
- **encrypted_data_keys** (*List[EncryptedDataKey]*) – List of encrypted data keys.

Returns Optionally modified decryption materials.

Return type *DecryptionMaterials*

on_encrypt(*encryption_materials*)

Generate a data key using generator keyring and encrypt it using any available wrapping key in any child keyring.

Parameters **encryption_materials** (EncryptionMaterials) – Encryption materials for keyring to modify.

Returns Optionally modified encryption materials.

Return type *EncryptionMaterials*

Raises ***EncryptKeyError*** – if unable to encrypt data key.

3.12 aws_encryption_sdk.keyrings.raw

Resources required for Raw Keyrings.

Classes

RawAESKeyring(key_namespace, key_name, ...)	Generate an instance of Raw AES Keyring which encrypts using AES-GCM algorithm using wrapping key provided as a byte array
RawRSAKeyring(key_namespace, key_name, ...)	Generate an instance of Raw RSA Keyring which performs asymmetric encryption and decryption using public and private keys provided

class aws_encryption_sdk.keyrings.raw.**RawAESKeyring**(*key_namespace*, *key_name*, *wrapping_key*)

Bases: *aws_encryption_sdk.keyrings.base.Keyring*

Generate an instance of Raw AES Keyring which encrypts using AES-GCM algorithm using wrapping key provided as a byte array

New in version 2.0.0.

Parameters **key_namespace** (*str*) – String defining the keyring.

Note: key_namespace MUST NOT equal "aws-kms".

Parameters

- **key_name** (*str*) – Key ID
- **wrapping_key** (*bytes*) – Encryption key with which to wrap plaintext data key.

Note: Only one wrapping key can be specified in a Raw AES Keyring

on_decrypt(*decryption_materials*, *encrypted_data_keys*)

Attempt to decrypt the encrypted data keys.

Parameters

- **decryption_materials** (DecryptionMaterials) – Decryption materials for the keyring to modify
- **encrypted_data_keys** (*List[EncryptedDataKey]*) – List of encrypted data keys

Returns Decryption materials that MAY include a plaintext data key

Return type *DecryptionMaterials*

on_encrypt(*encryption_materials*)

Generate a data key if not present and encrypt it using any available wrapping key

Parameters **encryption_materials** (EncryptionMaterials) – Encryption materials for the keyring to modify

Returns Encryption materials containing data key and encrypted data key

Return type *EncryptionMaterials*

class aws_encryption_sdk.keyrings.raw.**RawRSAKeyring**(*key_namespace*, *key_name*, *wrapping_algorithm*, *private_wrapping_key=None*, *public_wrapping_key=None*)

Bases: *aws_encryption_sdk.keyrings.base.Keyring*

Generate an instance of Raw RSA Keyring which performs asymmetric encryption and decryption using public and private keys provided

New in version 2.0.0.

Parameters **key_namespace** (*str*) – String defining the keyring ID

Note: key_namespace MUST NOT equal "aws-kms".

Parameters

- **key_name** (*str*) – Key ID
- **private_wrapping_key** (*cryptography.hazmat.primitives.asymmetric.rsa.RSAPrivateKey*) – Private encryption key with which to wrap plaintext data key (optional)
- **public_wrapping_key** (*cryptography.hazmat.primitives.asymmetric.rsa.RSAPublicKey*) – Public encryption key with which to wrap plaintext data key (optional)
- **wrapping_algorithm** (WrappingAlgorithm) – Wrapping Algorithm with which to wrap plaintext data key
- **key_provider** (MasterKeyInfo) – Complete information about the key in the keyring

---

Note: At least one of public wrapping key or private wrapping key must be provided.

---

classmethod from_der_encoding(*key_namespace*, *key_name*, *wrapping_algorithm*, *public_encoded_key=None*, *private_encoded_key=None*, *password=None*)

Generate a raw RSA keyring using DER Encoded public and private keys

Parameters

- **key_namespace** (*str*) – String defining the keyring ID
- **key_name** (*bytes*) – Key ID
- **wrapping_algorithm** (WrappingAlgorithm) – Wrapping Algorithm with which to wrap plaintext data key
- **public_encoded_key** (*bytes*) – DER encoded public key (optional)
- **private_encoded_key** (*bytes*) – DER encoded private key (optional)
- **password** (*bytes*) – Password to load private key (optional)

Returns *RawRSAKeyring* constructed using required parameters

classmethod from_pem_encoding(*key_namespace*, *key_name*, *wrapping_algorithm*, *public_encoded_key=None*, *private_encoded_key=None*, *password=None*)

Generate a Raw RSA keyring using PEM Encoded public and private keys

Parameters

- **key_namespace** (*str*) – String defining the keyring ID
- **key_name** (*bytes*) – Key ID

- **wrapping_algorithm** (*WrappingAlgorithm*) – Wrapping Algorithm with which to wrap plaintext data key
- **public_encoded_key** (*bytes*) – PEM encoded public key (optional)
- **private_encoded_key** (*bytes*) – PEM encoded private key (optional)
- **password** (*bytes*) – Password to load private key (optional)

Returns *RawRSAKeyring* constructed using required parameters

on_decrypt(*decryption_materials*, *encrypted_data_keys*)
Attempt to decrypt the encrypted data keys.

Parameters

- **decryption_materials** (*DecryptionMaterials*) – Decryption materials for keyring to modify.
- **encrypted_data_keys** – List of encrypted data keys.

Type List[*EncryptedDataKey*]

Returns Decryption materials that MAY include a plaintext data key

Return type *DecryptionMaterials*

on_encrypt(*encryption_materials*)
Generate a data key using generator keyring and encrypt it using any available wrapping key in any child keyring.

Parameters **encryption_materials** (*EncryptionMaterials*) – Encryption materials for keyring to modify.

Returns Encryption materials containing data key and encrypted data key

Return type *EncryptionMaterials*

3.13 aws_encryption_sdk.key_providers.base

Base class interface for Master Key Providers.

Classes

MasterKey	Parent interface for Master Key classes.
MasterKeyConfig(key_id)	Configuration object for MasterKey objects.
MasterKeyProvider	Parent interface for Master Key Provider classes.
MasterKeyProviderConfig()	Provides a common ancestor for MasterKeyProvider configuration objects and a stand-in point if common params are needed later.

class aws_encryption_sdk.key_providers.base.**MasterKey**
Bases: *aws_encryption_sdk.key_providers.base.MasterKeyProvider*

Parent interface for Master Key classes.

New in version 1.5.0: Master key providers are deprecated. Use *aws_encryption_sdk.keyrings.base.Keyring* instead.

Parameters

- **key_id** (*bytes*) – Key ID for Master Key
- **config** (aws_encryption_sdk.key_providers.base.MasterKeyConfig) – Configuration object

Performs universal prep work for all MasterKeys.

decrypt_data_key(*encrypted_data_key*, *algorithm*, *encryption_context*)
Decrypts an encrypted data key and returns the plaintext.

Parameters

- **data_key** (aws_encryption_sdk.structures.EncryptedDataKey) – Encrypted data key
- **algorithm** (*aws_encryption_sdk.identifiers.Algorithm*) – Algorithm object which directs how this Master Key will encrypt the data key
- **encryption_context** (*dict*) – Encryption context to use in decryption

Returns Decrypted data key

Return type *aws_encryption_sdk.structures.DataKey*

Raises ***IncorrectMasterKeyError*** – if Data Key's key provider does not match this Master Key

encrypt_data_key(*data_key*, *algorithm*, *encryption_context*)
Encrypts a supplied data key.

Parameters

- **data_key** (*aws_encryption_sdk.structures.RawDataKey* or *aws_encryption_sdk.structures.DataKey*) – Unencrypted data key
- **algorithm** (*aws_encryption_sdk.identifiers.Algorithm*) – Algorithm object which directs how this Master Key will encrypt the data key
- **encryption_context** (*dict*) – Encryption context to use in encryption

Returns Data key containing encrypted data key

Return type *aws_encryption_sdk.structures.EncryptedDataKey*

Raises ***IncorrectMasterKeyError*** – if Data Key's key provider does not match this Master Key

generate_data_key(*algorithm*, *encryption_context*)
Generates and returns data key for use encrypting message.

Parameters

- **algorithm** (*aws_encryption_sdk.identifiers.Algorithm*) – Algorithm on which to base data key
- **encryption_context** (*dict*) – Encryption context to use in encryption

Returns Generated data key

Return type *aws_encryption_sdk.structures.DataKey*

key_provider
Provides the MasterKeyInfo object identifying this MasterKey.

Returns This MasterKey's Identifying Information

Return type *aws_encryption_sdk.structures.MasterKeyInfo*

master_keys_for_encryption(*encryption_context*, *plaintext_rostream*, *plaintext_length=None*)
Returns self and a list containing self, to match the format of output for a Master Key Provider.

> **Warning:** If plaintext_stream seek position is modified, it must be returned before leaving method.

Parameters

- **encryption_context** (*dict*) – Encryption context passed to client
- **plaintext_rostream** (*aws_encryption_sdk.internal.utils.streams.ROStream*) – Source plaintext read-only stream
- **plaintext_length** (*int*) – Length of source plaintext (optional)

Returns Tuple containing self and a list of self

Return type tuple containing *aws_encryption_sdk.key_providers.base.MasterKey* and list of *aws_encryption_sdk.key_providers.base.MasterKey*

owns_data_key(*data_key*)
Determines if data_key object is owned by this MasterKey.

Parameters **data_key** (*aws_encryption_sdk.structures.DataKey*, *aws_encryption_sdk.structures.RawDataKey*, or *aws_encryption_sdk.structures.EncryptedDataKey*) – Data key to evaluate

Returns Boolean statement of ownership

Return type bool

class aws_encryption_sdk.key_providers.base.**MasterKeyConfig**(*key_id*)
Bases: object

Configuration object for MasterKey objects.

Parameters **key_id** (*bytes*) – Key ID for Master Key

class aws_encryption_sdk.key_providers.base.**MasterKeyProvider**
Bases: object

Parent interface for Master Key Provider classes.

New in version 1.5.0: Master key providers are deprecated. Use *aws_encryption_sdk.keyrings.base.Keyring* instead.

Parameters **config** (aws_encryption_sdk.key_providers.base.MasterKeyProviderConfig) – Configuration object

Set key index and member set for all new instances here to avoid requiring child classes to call super init.

add_master_key(*key_id*)
Adds a single Master Key to this provider.

Parameters **key_id** (*bytes*) – Key ID with which to create MasterKey

add_master_key_provider(*key_provider*)
Adds a single Master Key Provider to this provider.

Parameters **key_provider** (aws_encryption_sdk.key_providers.base.MasterKeyProvider) – Master Key Provider to add to this provider

add_master_key_providers_from_list(*key_providers*)
Adds multiple Master Key Providers to this provider.

Parameters **key_provider** (list of *aws_encryption_sdk.key_providers.base.MasterKeyProvider*) – List of Master Key Providers to add to this provider

add_master_keys_from_list(*key_ids*)
Adds multiple Master Keys to this provider.

Parameters **key_ids** (*list*) – List of Master Key IDs

decrypt_data_key(*encrypted_data_key*, *algorithm*, *encryption_context*)
Iterates through all currently added Master Keys and Master Key Providers to attempt to decrypt data key.

Parameters

- **encrypted_data_key** (aws_encryption_sdk.structures.EncryptedDataKey) – Encrypted data key to decrypt
- **algorithm** (*aws_encryption_sdk.identifiers.Algorithm*) – Algorithm object which directs how this Master Key will encrypt the data key
- **encryption_context** (*dict*) – Encryption context to use in encryption

Returns Decrypted data key

Return type *aws_encryption_sdk.structures.DataKey*

Raises ***DecryptKeyError*** – if unable to decrypt encrypted data key

decrypt_data_key_from_list(*encrypted_data_keys*, *algorithm*, *encryption_context*)
Receives a list of encrypted data keys and returns the first one which this provider is able to decrypt.

Parameters

- **encrypted_data_keys** (list of *aws_encryption_sdk.structures.EncryptedDataKey*) – List of encrypted data keys
- **algorithm** (*aws_encryption_sdk.identifiers.Algorithm*) – Algorithm object which directs how this Master Key will encrypt the data key
- **encryption_context** (*dict*) – Encryption context to use in encryption

Returns Decrypted data key

Return type *aws_encryption_sdk.structures.DataKey*

Raises ***DecryptKeyError*** – if unable to decrypt any of the supplied encrypted data keys

master_key(*key_id*)
Returns a master key for encrypt based on the specified key_id, adding it to this provider if not already present.

Parameters **key_id** (*bytes*) – Key ID with which to find or create Master Key

Returns Master Key based on key_id

Return type *aws_encryption_sdk.key_providers.base.MasterKey*

master_key_for_decrypt(*key_info*)
Returns a master key for decrypt based on the specified key_info. This is only added to this master key provider for the decrypt path.

Parameters **key_info** (*bytes*) – Key info from encrypted data key

Returns Master Key based on key_info

Return type *aws_encryption_sdk.key_providers.base.MasterKey*

master_key_for_encrypt(*key_id*)
Returns a master key for encrypt based on the specified key_id, adding it to this provider if not already present.

Parameters **key_id** (*bytes*) – Key ID with which to find or create Master Key

Returns Master Key based on key_id

Return type *aws_encryption_sdk.key_providers.base.MasterKey*

master_keys_for_data_key(*data_key*)
Locates the correct master keys from children for the specified data key.

Parameters **data_key** (EncryptedDataKey, RawDataKey, or DataKey) – Data key for which to locate owning master keys

Returns Masters key that own data key

Return type iterator of *MasterKey*

Raises ***UnknownIdentityError*** – if unable to locate the correct master key

master_keys_for_encryption(*encryption_context*, *plaintext_rostream*, *plaintext_length=None*)
Returns a set containing all Master Keys added to this Provider, or any member Providers, which should be used to encrypt data keys for the specified data.

Note: This does not necessarily include all Master Keys accessible from this Provider.

Note: The Primary Master Key is the first Master Key added to this Master Key Provider and is the Master Key which will be used to generate the data key.

Warning: If plaintext_rostream seek position is modified, it must be returned before leaving method.

Parameters

- **encryption_context** (*dict*) – Encryption context passed to client
- **plaintext_rostream** (*aws_encryption_sdk.internal.utils.streams.ROStream*) – Source plaintext read-only stream
- **plaintext_length** (*int*) – Length of source plaintext (optional)

Returns Tuple containing Primary Master Key and List of all Master Keys added to this Provider and any member Providers

Return type tuple containing *aws_encryption_sdk.key_providers.base.MasterKey* and list of *aws_encryption_sdk.key_providers.base.MasterKey*

provider_id
String defining provider ID.

Note: Must be implemented by specific MasterKeyProvider implementations.

class aws_encryption_sdk.key_providers.base.**MasterKeyProviderConfig**
Bases: object

Provides a common ancestor for MasterKeyProvider configuration objects and a stand-in point if common params are needed later.

3.14 aws_encryption_sdk.key_providers.kms

Master Key Providers for use with AWS KMS

Classes

KMSMasterKey(**kwargs)	Master Key class for KMS CMKs.
KMSMasterKeyConfig(key_id[, client, ...])	Configuration object for MasterKey objects.
KMSMasterKeyProvider(**kwargs)	Master Key Provider for KMS.
KMSMasterKeyProviderConfig([...])	Configuration object for KMSMasterKeyProvider objects.

class aws_encryption_sdk.key_providers.kms.**KMSMasterKey**(**kwargs*)
Bases: *aws_encryption_sdk.key_providers.base.MasterKey*

Master Key class for KMS CMKs.

New in version 1.5.0: Master key providers are deprecated. Use *aws_encryption_sdk.keyrings.aws_kms.AwsKmsKeyring* instead.

Parameters

- **config** (aws_encryption_sdk.key_providers.kms.KMSMasterKeyConfig) – Configuration object (config or individual parameters required)
- **key_id** (*bytes*) – KMS CMK ID
- **client** (*botocore.client.KMS*) – Boto3 KMS client
- **grant_tokens** (*list*) – List of grant tokens to pass to KMS on CMK operations

Performs transformations needed for KMS.

class aws_encryption_sdk.key_providers.kms.**KMSMasterKeyConfig**(*key_id*, *client=NOTHING*, *grant_tokens=NOTHING*)
Bases: *aws_encryption_sdk.key_providers.base.MasterKeyConfig*

Configuration object for MasterKey objects.

Parameters

- **key_id** (*str*) – KMS CMK ID
- **client** (*botocore.client.KMS*) – Boto3 KMS client
- **grant_tokens** (*list*) – List of grant tokens to pass to KMS on CMK operations

client_default()
Create a client if one was not provided.

class aws_encryption_sdk.key_providers.kms.**KMSMasterKeyProvider**(**kwargs)
Bases: *aws_encryption_sdk.key_providers.base.MasterKeyProvider*

Master Key Provider for KMS.

New in version 1.5.0: Master key providers are deprecated. Use *aws_encryption_sdk.keyrings.aws_kms.AwsKmsKeyring* instead.

To encrypt data, you must configure *KMSMasterKeyProvider* with at least one CMK. If you configure *KMSMasterKeyProvider* with multiple CMKs, it generates the data key using the first CMK and encrypts that data key using the rest, so that the encrypted message includes a copy of the data key encrypted under each configured CMK.

```
>>> from aws_encryption_sdk.key_providers.kms import KMSMasterKeyProvider
>>> kms_key_provider = KMSMasterKeyProvider(key_ids=[
...     "arn:aws:kms:us-east-1:2222222222222:key/22222222-2222-2222-2222-
↪222222222222",
...     "arn:aws:kms:us-east-1:3333333333333:key/33333333-3333-3333-3333-
↪333333333333",
... ])
```

You can also configure *KMSMasterKeyProvider* with CMKs in multiple regions:

```
>>> from aws_encryption_sdk.key_providers.kms import KMSMasterKeyProvider
>>> kms_key_provider = KMSMasterKeyProvider(key_ids=[
...     "arn:aws:kms:us-east-1:2222222222222:key/22222222-2222-2222-2222-
↪222222222222",
...     "arn:aws:kms:us-west-2:3333333333333:key/33333333-3333-3333-3333-
↪333333333333",
...     "arn:aws:kms:ap-northeast-1:4444444444444:key/44444444-4444-4444-4444-
↪444444444444",
... ])
```

KMSMasterKeyProvider needs AWS credentials in order to interact with AWS KMS. There are two ways that you can provide these credentials:

1. Provide your AWS credentials in one of the standard AWS credential discovery locations and the *KMSMasterKeyProvider* instance automatically discovers those credentials.

```
>>> from aws_encryption_sdk.key_providers.kms import KMSMasterKeyProvider
>>> import botocore.session
>>> kms_key_provider = KMSMasterKeyProvider()
```

2. Provide an existing botocore session to *KMSMasterKeyProvider*. This option can be useful if you want to use specific credentials or if you want to reuse an existing botocore session instance to decrease startup costs.

```
>>> from aws_encryption_sdk.key_providers.kms import KMSMasterKeyProvider
>>> import botocore.session
>>> existing_botocore_session = botocore.session.Session(profile="custom")
>>> kms_key_provider = KMSMasterKeyProvider(botocore_session=existing_botocore_
↪session)
```

If you need different credentials to use different CMKs, you can combine multiple *KMSMasterKeyProvider* or *KMSMasterKey* instances, each with their own credentials. However, we recommend that you use

aws_encryption_sdk.keyrings.aws_kms.AwsKmsKeyring and client suppliers for a simpler user experience.

Parameters

- **config** (*aws_encryption_sdk.key_providers.kms.KMSMasterKeyProviderConfig*) – Configuration object (optional)
- **botocore_session** (*botocore.session.Session*) – botocore session object (optional)
- **key_ids** (*list*) – List of KMS CMK IDs with which to pre-populate provider (optional)
- **region_names** (*list*) – List of regions for which to pre-populate clients (optional)

Prepares mutable attributes.

add_regional_client(*region_name*)
Adds a regional client for the specified region if it does not already exist.

Parameters **region_name** (*str*) – AWS Region ID (ex: us-east-1)

add_regional_clients_from_list(*region_names*)
Adds multiple regional clients for the specified regions if they do not already exist.

Parameters **region_names** (*list*) – List of regions for which to pre-populate clients

class aws_encryption_sdk.key_providers.kms.**KMSMasterKeyProviderConfig**(*botocore_session=NOTHING*, *key_ids=NOTHING*, *region_names=NOTHING*)
Bases: *aws_encryption_sdk.key_providers.base.MasterKeyProviderConfig*

Configuration object for KMSMasterKeyProvider objects.

Parameters

- **botocore_session** (*botocore.session.Session*) – botocore session object (optional)
- **key_ids** (*list*) – List of KMS CMK IDs with which to pre-populate provider (optional)
- **region_names** (*list*) – List of regions for which to pre-populate clients (optional)

3.15 aws_encryption_sdk.key_providers.raw

Resources required for Raw Master Keys.

Classes

RawMasterKey	Raw Master Key.
RawMasterKeyConfig(key_id, provider_id, ...)	Configuration object for RawMasterKey objects.
RawMasterKeyProvider	Raw Master Key Provider.

class aws_encryption_sdk.key_providers.raw.**RawMasterKey**
Bases: *aws_encryption_sdk.key_providers.base.MasterKey*

Raw Master Key.

New in version 1.5.0: Master key providers are deprecated. Use *aws_encryption_sdk.keyrings.raw.*

RawAESKeyring or *aws_encryption_sdk.keyrings.raw.RawRSAKeyring* instead.

Parameters

- **config** (aws_encryption_sdk.key_providers.raw.RawMasterKeyConfig) – Configuration object (config or individual parameters required)
- **key_id** (*bytes*) – Key ID for Master Key
- **provider_id** (*str*) – String defining provider ID
- **wrapping_key** (*aws_encryption_sdk.internal.crypto.WrappingKey*) – Encryption key with which to wrap plaintext_data_key

Inject registration of the new Raw Master Key Provider into the creation of each instance.

Note: Overloaded here to allow definition of _key_info_prefix on instantiation.

---

owns_data_key(*data_key*)

Determines if data_key object is owned by this RawMasterKey.

Parameters **data_key** (*aws_encryption_sdk.structures.DataKey*, *aws_encryption_sdk.structures.RawDataKey*, or *aws_encryption_sdk.structures.EncryptedDataKey*) – Data key to evaluate

Returns Boolean statement of ownership

Return type bool

class aws_encryption_sdk.key_providers.raw.**RawMasterKeyConfig**(*key_id*, *provider_id*, *wrapping_key*)

Bases: *aws_encryption_sdk.key_providers.base.MasterKeyConfig*

Configuration object for RawMasterKey objects.

Parameters

- **key_id** (*bytes*) – Key ID for Master Key
- **provider_id** (*str*) – String defining provider ID
- **wrapping_key** (*aws_encryption_sdk.internal.crypto.WrappingKey*) – Encryption key with which to wrap plaintext_data_key

class aws_encryption_sdk.key_providers.raw.**RawMasterKeyProvider**

Bases: *aws_encryption_sdk.key_providers.base.MasterKeyProvider*

Raw Master Key Provider.

New in version 1.5.0: Master key providers are deprecated. Use *aws_encryption_sdk.keyrings.raw.RawAESKeyring* or *aws_encryption_sdk.keyrings.raw.RawRSAKeyring* instead.

Parameters **config** (aws_encryption_sdk.key_providers.base.MasterKeyProviderConfig) – Configuration object (optional)

Set key index and member set for all new instances here to avoid requiring child classes to call super init.

3.16 aws_encryption_sdk.materials_managers

Primitive structures for use when interacting with crypto material managers.

New in version 1.3.0.

Classes

CryptographicMaterials(algorithm, . . . [, . . .])	Cryptographic materials core.
DecryptionMaterials([data_key, verification_key])	Decryption materials returned by a crypto material manager's *decrypt_materials* method.
DecryptionMaterialsRequest(algorithm, . . .)	Request object to provide to a crypto material manager's *decrypt_materials* method.
EncryptionMaterials([algorithm, . . .])	Encryption materials returned by a crypto material manager's *get_encryption_materials* method.
EncryptionMaterialsRequest(. . . [, . . .])	Request object to provide to a crypto material manager's *get_encryption_materials* method.

class aws_encryption_sdk.materials_managers.**CryptographicMaterials**(*algorithm*, *encryption_context*, *data_encryption_key=None*)

Bases: object

Cryptographic materials core.

New in version 2.0.0.

Parameters

- **algorithm** (*Algorithm*) – Algorithm to use for encrypting message
- **encryption_context** (*dict*) – Encryption context tied to *encrypted_data_keys*
- **data_encryption_key** (RawDataKey) – Plaintext data key to use for encrypting message

class aws_encryption_sdk.materials_managers.**DecryptionMaterials**(*data_key=<object object>*, *verification_key=None*, ***kwargs*)

Bases: *aws_encryption_sdk.materials_managers.CryptographicMaterials*

Decryption materials returned by a crypto material manager's *decrypt_materials* method.

New in version 1.3.0.

New in version 2.0.0: The **algorithm**, **data_encryption_key**, and **encryption_context** parameters.

New in version 2.0.0: All parameters are now optional.

Parameters

- **algorithm** (*Algorithm*) – Algorithm to use for encrypting message (optional)
- **data_encryption_key** (RawDataKey) – Plaintext data key to use for encrypting message (optional)

- **encryption_context** (*dict*) – Encryption context tied to *encrypted_data_keys* (optional)
- **verification_key** (*bytes*) – Raw signature verification key (optional)

data_key
Backwards-compatible shim for access to data key.

is_complete
Determine whether these materials are sufficiently complete for use as decryption materials.

Return type bool

with_data_encryption_key(*data_encryption_key*)
Get new decryption materials that also include this data encryption key.

New in version 2.0.0.

Parameters **data_encryption_key** (RawDataKey) – Data encryption key

Return type *DecryptionMaterials*

Raises

- **AttributeError** – if data encryption key is already set
- ***InvalidDataKeyError*** – if data key length does not match algorithm suite

with_verification_key(*verification_key*)
Get new decryption materials that also include this verification key.

New in version 2.0.0.

Parameters **verification_key** (*bytes*) – Verification key

Return type *DecryptionMaterials*

class aws_encryption_sdk.materials_managers.**DecryptionMaterialsRequest**(*algorithm*, *encrypted_data_keys*, *encryption_context*)

Bases: object

Request object to provide to a crypto material manager's *decrypt_materials* method.

New in version 1.3.0.

Parameters

- **algorithm** (*aws_encryption_sdk.identifiers.Algorithm*) – Algorithm to provide to master keys for underlying decrypt requests
- **encrypted_data_keys** (set of *aws_encryption_sdk.structures.EncryptedDataKey*) – Set of encrypted data keys
- **encryption_context** (*dict*) – Encryption context to provide to master keys for underlying decrypt requests

class aws_encryption_sdk.materials_managers.**EncryptionMaterials**(*algorithm=None*, *data_encryption_key=None*, *encrypted_data_keys=None*, *encryption_context=None*, *signing_key=None*, ***kwargs*)

Bases: *aws_encryption_sdk.materials_managers.CryptographicMaterials*

Encryption materials returned by a crypto material manager's *get_encryption_materials* method.

New in version 1.3.0.

New in version 2.0.0: Most parameters are now optional.

Parameters

- **algorithm** (*Algorithm*) – Algorithm to use for encrypting message
- **data_encryption_key** (RawDataKey) – Plaintext data key to use for encrypting message (optional)
- **encrypted_data_keys** (list of EncryptedDataKey) – List of encrypted data keys (optional)
- **encryption_context** (*dict*) – Encryption context tied to *encrypted_data_keys*
- **signing_key** (*bytes*) – Encoded signing key (optional)

encrypted_data_keys

Return a read-only version of the encrypted data keys.

Return type Tuple[*EncryptedDataKey*]

is_complete

Determine whether these materials are sufficiently complete for use as encryption materials.

Return type bool

with_data_encryption_key(*data_encryption_key*)

Get new encryption materials that also include this data encryption key.

New in version 2.0.0.

Parameters **data_encryption_key** (RawDataKey) – Data encryption key

Return type *EncryptionMaterials*

Raises

- **AttributeError** – if data encryption key is already set
- ***InvalidDataKeyError*** – if data key length does not match algorithm suite

with_encrypted_data_key(*encrypted_data_key*)

Get new encryption materials that also include this encrypted data key.

New in version 2.0.0.

Parameters **encrypted_data_key** (EncryptedDataKey) – Encrypted data key to add

Return type *EncryptionMaterials*

Raises **AttributeError** – if data encryption key is not set

with_signing_key(*signing_key*)
Get new encryption materials that also include this signing key.

New in version 2.0.0.

Parameters **signing_key** (*bytes*) – Signing key

Return type *EncryptionMaterials*

Raises

- **AttributeError** – if signing key is already set
- ***SignatureKeyError*** – if algorithm suite does not support signing keys

class aws_encryption_sdk.materials_managers.**EncryptionMaterialsRequest**(*encryption_context*, *frame_length*, *plaintext_rostream=None*, *algorithm=None*, *plaintext_length=None*)

Bases: object

Request object to provide to a crypto material manager's *get_encryption_materials* method.

New in version 1.3.0.

Warning: If plaintext_rostream seek position is modified, it must be returned before leaving method.

Parameters

- **encryption_context** (*dict*) – Encryption context passed to underlying master key provider and master keys
- **frame_length** (*int*) – Frame length to be used while encrypting stream
- **plaintext_rostream** (*aws_encryption_sdk.internal.utils.streams.ROStream*) – Source plaintext read-only stream (optional)
- **algorithm** (*aws_encryption_sdk.identifiers.Algorithm*) – Algorithm passed to underlying master key provider and master keys (optional)
- **plaintext_length** (*int*) – Length of source plaintext (optional)

3.17 aws_encryption_sdk.materials_managers.base

Base class interface for crypto material managers.

Classes

CryptoMaterialsManager	Parent interface for crypto material manager classes.

class aws_encryption_sdk.materials_managers.base.**CryptoMaterialsManager**
Bases: object

Parent interface for crypto material manager classes.

New in version 1.3.0.

decrypt_materials(*request*)
Provides decryption materials appropriate for the request.

Note: Must be implemented by specific CryptoMaterialsManager implementations.

Parameters **request** (aws_encryption_sdk.materials_managers.DecryptionMaterialsRequest) – decrypt materials request

Returns decryption materials

Return type *aws_encryption_sdk.materials_managers.DecryptionMaterials*

get_encryption_materials(*request*)
Provides encryption materials appropriate for the request.

Note: Must be implemented by specific CryptoMaterialsManager implementations.

Parameters **request** (aws_encryption_sdk.materials_managers.EncryptionMaterialsRequest) – encryption materials request

Returns encryption materials

Return type *aws_encryption_sdk.materials_managers.EncryptionMaterials*

3.18 aws_encryption_sdk.materials_managers.caching

Caching crypto material manager.

Classes

CachingCryptoMaterialsManager(cache, max_age)	Crypto material manager which caches results from an underlying material manager.

class aws_encryption_sdk.materials_managers.caching.**CachingCryptoMaterialsManager**(*cache*, *max_age*, *max_messag* *max_bytes_e* *par-* *ti-* *tion_name=* *mas-* *ter_key_prov* *back-* *ing_materia* *keyring=No*

Bases: *aws_encryption_sdk.materials_managers.base.CryptoMaterialsManager*

Crypto material manager which caches results from an underlying material manager.

New in version 1.3.0.

New in version 1.5.0: The *keyring* parameter.

```
>>> import aws_encryption_sdk
>>> kms_key_provider = aws_encryption_sdk.KMSMasterKeyProvider(key_ids=[
...     'arn:aws:kms:us-east-1:2222222222222:key/22222222-2222-2222-2222-
↪222222222222',
...     'arn:aws:kms:us-east-1:3333333333333:key/33333333-3333-3333-3333-
↪333333333333'
... ])
>>> local_cache = aws_encryption_sdk.LocalCryptoMaterialsCache(capacity=100)
>>> caching_materials_manager = aws_encryption_sdk.CachingCryptoMaterialsManager(
...     master_key_provider=kms_key_provider,
...     cache=local_cache,
...     max_age=600.0,
...     max_messages_encrypted=10
... )
```

Note: The partition name is used to enable a single cache instance to be used by multiple material manager instances by partitioning the entries in that cache based on this value. If no partition name is provided, a random UUID will be used.

Note: Exactly one of `backing_materials_manager`, `keyring`, or `master_key_provider` must be provided.

Parameters

- **cache** (`CryptoMaterialsCache`) – Crypto cache to use with material manager
- **backing_materials_manager** (`CryptoMaterialsManager`) – Crypto material manager to back this caching material manager (either `backing_materials_manager`, `keyring`, or `master_key_provider` required)
- **master_key_provider** (`MasterKeyProvider`) – Master key provider to use (either `backing_materials_manager`, `keyring`, or `master_key_provider` required)

- **keyring** (Keyring) – Keyring to use (either backing_materials_manager, keyring, or master_key_provider required)
- **max_age** (*float*) – Maximum time in seconds that a cache entry may be kept in the cache
- **max_messages_encrypted** (*int*) – Maximum number of messages that may be encrypted under a cache entry (optional)
- **max_bytes_encrypted** (*int*) – Maximum number of bytes that a cache entry may be used to process (optional)
- **partition_name** (*bytes*) – Partition name to use for this instance (optional)

decrypt_materials(*request*)
Provides decryption materials appropriate for the request.

Parameters **request** (aws_encryption_sdk.materials_managers.DecryptionMaterialsRequest) – decrypt materials request

Returns decryption materials

Return type *aws_encryption_sdk.materials_managers.DecryptionMaterials*

get_encryption_materials(*request*)
Provides encryption materials appropriate for the request.

Parameters **request** (aws_encryption_sdk.materials_managers.EncryptionMaterialsRequest) – Encryption materials request

Returns encryption materials

Return type *aws_encryption_sdk.materials_managers.EncryptionMaterials*

3.19 aws_encryption_sdk.materials_managers.default

Default crypto material manager class.

Classes

DefaultCryptoMaterialsManager([...])	Default crypto material manager.

class aws_encryption_sdk.materials_managers.default.**DefaultCryptoMaterialsManager**(*master_key_provider=None*, *keyring=None*)

Bases: *aws_encryption_sdk.materials_managers.base.CryptoMaterialsManager*

Default crypto material manager.

New in version 1.3.0.

New in version 1.5.0: The *keyring* parameter.

Parameters

- **master_key_provider** (MasterKeyProvider) – Master key provider to use (either keyring or master_key_provider is required)
- **keyring** (Keyring) – Keyring to use (either keyring or master_key_provider is required)

decrypt_materials(*request*)
Obtains a plaintext data key from one or more encrypted data keys using underlying master key provider.

Parameters **request** (aws_encryption_sdk.materials_managers.DecryptionMaterialsRequest) – decrypt materials request

Returns decryption materials

Return type *aws_encryption_sdk.materials_managers.DecryptionMaterials*

get_encryption_materials(*request*)
Creates encryption materials using underlying master key provider.

Parameters **request** (aws_encryption_sdk.materials_managers.EncryptionMaterialsRequest) – encryption materials request

Returns encryption materials

Return type *aws_encryption_sdk.materials_managers.EncryptionMaterials*

Raises

- ***InvalidCryptographicMaterialsError*** – if keyring cannot complete encryption materials
- ***InvalidCryptographicMaterialsError*** – if encryption materials received from keyring do not match request
- ***MasterKeyProviderError*** – if no master keys are available from the underlying master key provider
- ***MasterKeyProviderError*** – if the primary master key provided by the underlying master key provider is not included in the full set of master keys provided by that provider

3.20 aws_encryption_sdk.streaming_client

High level AWS Encryption SDK client for streaming objects.

Classes

DecryptorConfig(source[, materials_manager, ...])		Configuration object for StreamDecryptor class.
EncryptorConfig(source[, materials_manager, ...])		Configuration object for StreamEncryptor class.
StreamDecryptor(**kwargs)		Provides a streaming encryptor for encrypting a stream source.
StreamEncryptor(**kwargs)		Provides a streaming encryptor for encrypting a stream source.

class aws_encryption_sdk.streaming_client.**DecryptorConfig**(*source*, *materials_manager=None*, *keyring=None*, *key_provider=None*, *source_length=None*, *line_length=8192*, *max_body_length=None*)
Bases: aws_encryption_sdk.streaming_client._ClientConfig

Configuration object for StreamDecryptor class.

New in version 1.5.0: The *keyring* parameter.

Parameters

- **source** (*str, bytes, io.IOBase, or file*) – Source data to encrypt or decrypt
- **materials_manager** (CryptoMaterialsManager) – Cryptographic materials manager to use for encryption (either materials_manager, keyring, key_provider required)
- **keyring** (Keyring) – Keyring to use for encryption (either materials_manager, keyring, key_provider required)
- **key_provider** (MasterKeyProvider) – Master key provider to use for encryption (either materials_manager, keyring, key_provider required)
- **source_length** (*int*) – Length of source data (optional)

  Note: If source_length is not provided and read() is called, will attempt to seek() to the end of the stream and tell() to find the length of source data.

- **max_body_length** (*int*) – Maximum frame size (or content length for non-framed messages) in bytes to read from ciphertext message.

class aws_encryption_sdk.streaming_client.**EncryptorConfig**(*source*, *materials_manager=None*, *keyring=None*, *key_provider=None*, *source_length=None*, *line_length=8192*, *encryption_context=NOTHING*, *algorithm=None*, *frame_length=4096*)

Bases: aws_encryption_sdk.streaming_client._ClientConfig

Configuration object for StreamEncryptor class.

New in version 1.5.0: The *keyring* parameter.

Parameters

- **source** (*str, bytes, io.IOBase, or file*) – Source data to encrypt or decrypt
- **materials_manager** (CryptoMaterialsManager) – Cryptographic materials manager to use for encryption (either materials_manager, keyring, key_provider required)
- **keyring** (Keyring) – Keyring to use for encryption (either materials_manager, keyring, key_provider required)
- **key_provider** (MasterKeyProvider) – Master key provider to use for encryption (either materials_manager, keyring, key_provider required)
- **source_length** (*int*) – Length of source data (optional)

  Note: If source_length is not provided and unframed message is being written or read() is called, will attempt to seek() to the end of the stream and tell() to find the length of source

data.

Note: New in version 1.3.0.

If *source_length* and *materials_manager* are both provided, the total plaintext bytes encrypted will not be allowed to exceed *source_length*. To maintain backwards compatibility, this is not enforced if a *key_provider* is provided.

- **encryption_context** (*dict*) – Dictionary defining encryption context
- **algorithm** (*aws_encryption_sdk.identifiers.Algorithm*) – Algorithm to use for encryption (optional)
- **frame_length** (*int*) – Frame length in bytes (optional)

class aws_encryption_sdk.streaming_client.**StreamDecryptor**(***kwargs*)

Bases: aws_encryption_sdk.streaming_client._EncryptionStream

Provides a streaming encryptor for encrypting a stream source. Behaves as a standard file-like object.

Note: Take care when decrypting framed messages with large frame length and large non-framed messages. See *aws_encryption_sdk.stream* for more details.

Note: If config is provided, all other parameters are ignored.

New in version 1.5.0: The *keyring* parameter.

Parameters

- **config** (aws_encryption_sdk.streaming_client.DecryptorConfig) – Client configuration object (config or individual parameters required)
- **source** (*str, bytes, io.IOBase, or file*) – Source data to encrypt or decrypt
- **materials_manager** (CryptoMaterialsManager) – Cryptographic materials manager to use for encryption (either materials_manager, keyring, key_provider required)
- **keyring** (Keyring) – Keyring to use for encryption (either materials_manager, keyring, key_provider required)
- **key_provider** (MasterKeyProvider) – Master key provider to use for encryption (either materials_manager, keyring, key_provider required)
- **source_length** (*int*) – Length of source data (optional)

Note: If source_length is not provided and read() is called, will attempt to seek() to the end of the stream and tell() to find the length of source data.

- **max_body_length** (*int*) – Maximum frame size (or content length for non-framed messages) in bytes to read from ciphertext message.

Prepares necessary initial values.

close()
Closes out the stream.

class aws_encryption_sdk.streaming_client.**StreamEncryptor**(***kwargs*)
Bases: aws_encryption_sdk.streaming_client._EncryptionStream

Provides a streaming encryptor for encrypting a stream source. Behaves as a standard file-like object.

Note: Take care when encrypting framed messages with large frame length and large non-framed messages. See *aws_encryption_sdk.stream* for more details.

Note: If config is provided, all other parameters are ignored.

New in version 1.5.0: The *keyring* parameter.

Parameters

- **config** (aws_encryption_sdk.streaming_client.EncryptorConfig) – Client configuration object (config or individual parameters required)
- **source** (*str, bytes, io.IOBase, or file*) – Source data to encrypt or decrypt
- **materials_manager** (CryptoMaterialsManager) – Cryptographic materials manager to use for encryption (either materials_manager, keyring, key_provider required)
- **keyring** (Keyring) – Keyring to use for encryption (either materials_manager, keyring, key_provider required)
- **key_provider** (MasterKeyProvider) – Master key provider to use for encryption (either materials_manager, keyring, key_provider required)
- **source_length** (*int*) – Length of source data (optional)

  Note: If source_length is not provided and unframed message is being written or read() is called, will attempt to seek() to the end of the stream and tell() to find the length of source data.

  Note: New in version 1.3.0.

  If *source_length* and *materials_manager* are both provided, the total plaintext bytes encrypted will not be allowed to exceed *source_length*. To maintain backwards compatibility, this is not enforced if a *key_provider* is provided.

- **encryption_context** (*dict*) – Dictionary defining encryption context
- **algorithm** (*aws_encryption_sdk.identifiers.Algorithm*) – Algorithm to use for encryption
- **frame_length** (*int*) – Frame length in bytes

Prepares necessary initial values.

ciphertext_length()
Returns the length of the resulting ciphertext message in bytes.

Return type int

close()
Closes out the stream.

3.21 aws_encryption_sdk.structures

Public data structures for aws_encryption_sdk.

Classes

CryptoResult(result, header)	Result container for one-shot cryptographic API results.
DataKey(key_provider, data_key, ...)	Holds both the encrypted and unencrypted copies of a data key.
EncryptedDataKey(key_provider, ...)	Holds only the encrypted copy of a data key.
MasterKeyInfo(provider_id, key_info[, key_name])	Contains information necessary to identify a Master Key.
MessageHeader(version, type, algorithm, ...)	Deserialized message header object.
RawDataKey(key_provider, data_key)	Hold only the unencrypted copy of a data key.

class aws_encryption_sdk.structures.**CryptoResult**(*result*, *header*)
Bases: object

Result container for one-shot cryptographic API results.

New in version 2.0.0.

Note: For backwards compatibility, this container also unpacks like a 2-member tuple. This allows for backwards compatibility with the previous outputs.

Parameters

- **result** (*bytes*) – Binary results of the cryptographic operation
- **header** (MessageHeader) – Encrypted message metadata

class aws_encryption_sdk.structures.**DataKey**(*key_provider*, *data_key*, *encrypted_data_key*)
Bases: object

Holds both the encrypted and unencrypted copies of a data key.

Parameters

- **key_provider** (aws_encryption_sdk.structures.MasterKeyInfo) – Key Provider information
- **data_key** (*bytes*) – Plaintext data key
- **encrypted_data_key** (*bytes*) – Encrypted data key

class aws_encryption_sdk.structures.**EncryptedDataKey**(*key_provider*, *encrypted_data_key*)
Bases: object

Holds only the encrypted copy of a data key.

Parameters

- **key_provider** (aws_encryption_sdk.structures.MasterKeyInfo) – Key Provider information
- **encrypted_data_key** (*bytes*) – Encrypted data key

classmethod from_data_key(*data_key*)
Build an *EncryptedDataKey* from a *DataKey*.

New in version 2.0.0.

class aws_encryption_sdk.structures.**MasterKeyInfo**(*provider_id*, *key_info*, *key_name=None*)

Bases: object

Contains information necessary to identify a Master Key.

Note: The only keyring or master key that should need to set `key_name` is the Raw AES keyring/master key. For all other keyrings and master keys, `key_info` and `key_name` should always be the same.

New in version 2.0.0: `key_name`

Parameters

- **provider_id** (*str*) – MasterKey provider_id value
- **key_info** (*bytes*) – MasterKey key_info value
- **key_name** (*bytes*) – Key name if different than key_info (optional)

key_namespace
Access the key namespace value (previously, provider ID).

New in version 2.0.0.

class aws_encryption_sdk.structures.**MessageHeader**(*version*, *type*, *algorithm*, *message_id*, *encryption_context*, *encrypted_data_keys*, *content_type*, *content_aad_length*, *header_iv_length*, *frame_length*)

Bases: object

Deserialized message header object.

Parameters

- **version** (SerializationVersion) – Message format version, per spec
- **type** (ObjectType) – Message content type, per spec
- **algorithm** (AlgorithmSuite) – Algorithm to use for encryption
- **message_id** (*bytes*) – Message ID
- **encryption_context** (*Dict[str, str]*) – Dictionary defining encryption context
- **encrypted_data_keys** (*Sequence[EncryptedDataKey]*) – Encrypted data keys
- **content_type** (ContentType) – Message content framing type (framed/non-framed)
- **content_aad_length** (*int*) – empty

- **header_iv_length** (*int*) – Bytes in Initialization Vector value found in header
- **frame_length** (*int*) – Length of message frame in bytes

class aws_encryption_sdk.structures.**RawDataKey**(*key_provider*, *data_key*)
Bases: object

Hold only the unencrypted copy of a data key.

Parameters

- **key_provider** (aws_encryption_sdk.structures.MasterKeyInfo) – Key Provider information
- **data_key** (*bytes*) – Plaintext data key

classmethod from_data_key(*data_key*)
Build an *RawDataKey* from a *DataKey*.

New in version 2.0.0.

3.22 aws_encryption_sdk.internal

Internal Implementation Details

Warning: No guarantee is provided on the modules and APIs within this namespace staying consistent. Directly reference at your own risk.

3.23 aws_encryption_sdk.internal.crypto.authentication

Contains authentication primitives.

Classes

Signer(algorithm, key)	Abstract signing handler.
Verifier(algorithm, key)	Abstract signature verification handler.

class aws_encryption_sdk.internal.crypto.authentication.**Signer**(*algorithm*, *key*)
Bases: aws_encryption_sdk.internal.crypto.authentication._PrehashingAuthenticator

Abstract signing handler.

Parameters

- **algorithm** (*aws_encryption_sdk.identifiers.Algorithm*) – Algorithm on which to base signer
- **key** (*currently only Elliptic Curve Private Keys are supported*) – Private key from which a signer can be generated

Prepares initial values.

encoded_public_key()

Returns the encoded public key.

Note: For ECC curves, this will return the encoded compressed public point.

Returns Encoded public key from signer

Return type bytes

finalize()
Finalizes the signer and returns the signature.

Returns Calculated signer signature

Return type bytes

classmethod from_key_bytes(*algorithm*, *key_bytes*)
Builds a *Signer* from an algorithm suite and a raw signing key.

Parameters

- **algorithm** (*aws_encryption_sdk.identifiers.Algorithm*) – Algorithm on which to base signer
- **key_bytes** (*bytes*) – Raw signing key

Return type aws_encryption_sdk.internal.crypto.Signer

key_bytes()
Returns the raw signing key.

Return type bytes

update(*data*)
Updates the cryptographic signer with the supplied data.

Parameters **data** (*bytes*) – Data to be signed

class aws_encryption_sdk.internal.crypto.authentication.**Verifier**(*algorithm*, *key*)

Bases: aws_encryption_sdk.internal.crypto.authentication._PrehashingAuthenticator

Abstract signature verification handler.

Note: For ECC curves, the signature must be DER encoded as specified in RFC 3279.

Parameters

- **algorithm** (*aws_encryption_sdk.identifiers.Algorithm*) – Algorithm on which to base verifier
- **public_key** (*may vary*) – Appropriate public key object for algorithm

Prepares initial values.

classmethod from_encoded_point(*algorithm*, *encoded_point*)
Creates a Verifier object based on the supplied algorithm and encoded compressed ECC curve point.

Parameters

- **algorithm** (*aws_encryption_sdk.identifiers.Algorithm*) – Algorithm on which to base verifier
- **encoded_point** (*bytes*) – ECC public point compressed and encoded with _ecc_encode_compressed_point

Returns Instance of Verifier generated from encoded point

Return type aws_encryption_sdk.internal.crypto.Verifier

classmethod from_key_bytes(*algorithm*, *key_bytes*)
Creates a *Verifier* object based on the supplied algorithm and raw verification key.

Parameters

- **algorithm** (*aws_encryption_sdk.identifiers.Algorithm*) – Algorithm on which to base verifier
- **encoded_point** (*bytes*) – Raw verification key

Returns Instance of Verifier generated from encoded point

Return type aws_encryption_sdk.internal.crypto.Verifier

key_bytes()
Returns the raw verification key.

Return type bytes

update(*data*)
Updates the cryptographic verifier with the supplied data.

Parameters **data** (*bytes*) – Data to verify using the signature

verify(*signature*)
Verifies the signature against the current cryptographic verifier state.

Parameters **signature** (*bytes*) – The signature to verify

3.24 aws_encryption_sdk.internal.crypto.data_keys

Contains data key helper functions.

Functions

derive_data_encryption_key(source_key, ...)	Derives the data encryption key using the defined algorithm.

aws_encryption_sdk.internal.crypto.data_keys.**derive_data_encryption_key**(*source_key*, *algorithm*, *message_id*)
Derives the data encryption key using the defined algorithm.

Parameters

- **source_key** (*bytes*) – Raw source key

- **algorithm** (*aws_encryption_sdk.identifiers.Algorithm*) – Algorithm used to encrypt this body
- **message_id** (*bytes*) – Message ID

Returns Derived data encryption key

Return type bytes

3.25 aws_encryption_sdk.internal.crypto.elliptic_curve

Contains elliptic curve functionality.

Functions

generate_ecc_signing_key(algorithm)	Returns an ECC signing key.

aws_encryption_sdk.internal.crypto.elliptic_curve.**generate_ecc_signing_key**(*algorithm*)
Returns an ECC signing key.

Parameters **algorithm** (*aws_encryption_sdk.identifiers.Algorithm*) – Algorithm object which determines what signature to generate

Returns Generated signing key

Raises ***NotSupportedError*** – if signing algorithm is not supported on this platform

3.26 aws_encryption_sdk.internal.crypto.encryption

Contains encryption primitives and helper functions.

Functions

decrypt(algorithm, key, encrypted_data, ...)	Decrypts a frame body.
encrypt(algorithm, key, plaintext, ...)	Encrypts a frame body.

Classes

Decryptor(algorithm, key, associated_data, ...)	Abstract decryption handler.
Encryptor(algorithm, key, associated_data, iv)	Abstract encryption handler.

class aws_encryption_sdk.internal.crypto.encryption.**Decryptor**(*algorithm*, *key*, *associated_data*, *iv*, *tag*)

Bases: object

Abstract decryption handler.

Parameters

- **algorithm** (*aws_encryption_sdk.identifiers.Algorithm*) – Algorithm used to encrypt this body
- **key** (*bytes*) – Raw source key
- **associated_data** (*bytes*) – Associated Data to send to decryption subsystem
- **iv** (*bytes*) – IV value with which to initialize decryption subsystem
- **tag** (*bytes*) – Tag with which to validate ciphertext

Prepares initial values.

finalize()
Finalizes and closes _decryptor.

Returns Final decrypted plaintext

Return type bytes

update(*ciphertext*)
Updates _decryptor with provided ciphertext.

Parameters **ciphertext** (*bytes*) – Ciphertext to decrypt

Returns Decrypted plaintext

Return type bytes

class aws_encryption_sdk.internal.crypto.encryption.**Encryptor**(*algorithm*, *key*, *associated_data*, *iv*)

Bases: object

Abstract encryption handler.

Parameters

- **algorithm** (*aws_encryption_sdk.identifiers.Algorithm*) – Algorithm used to encrypt this body
- **key** (*bytes*) – Encryption key
- **associated_data** (*bytes*) – Associated Data to send to encryption subsystem
- **iv** (*bytes*) – IV to use when encrypting message

Prepares initial values.

finalize()
Finalizes and closes _encryptor.

Returns Final encrypted ciphertext

Return type bytes

tag
Returns the _encryptor tag from the encryption subsystem.

Returns Encryptor tag

Return type bytes

update(*plaintext*)
Updates _encryptor with provided plaintext.

Parameters **plaintext** (*bytes*) – Plaintext to encrypt

Returns Encrypted ciphertext

Return type bytes

aws_encryption_sdk.internal.crypto.encryption.**decrypt**(*algorithm*, *key*, *encrypted_data*, *associated_data*)

Decrypts a frame body.

Parameters

- **algorithm** (*aws_encryption_sdk.identifiers.Algorithm*) – Algorithm used to encrypt this body
- **key** (*bytes*) – Plaintext data key
- **encrypted_data** (*aws_encryption_sdk.internal.structures.EncryptedData*, aws_encryption_sdk.internal.structures.FrameBody, or *aws_encryption_sdk.internal.structures.MessageNoFrameBody*) – EncryptedData containing body data
- **associated_data** (*bytes*) – AAD string generated for body

Returns Plaintext of body

Return type bytes

aws_encryption_sdk.internal.crypto.encryption.**encrypt**(*algorithm*, *key*, *plaintext*, *associated_data*, *iv*)

Encrypts a frame body.

Parameters

- **algorithm** (*aws_encryption_sdk.identifiers.Algorithm*) – Algorithm used to encrypt this body
- **key** (*bytes*) – Encryption key
- **plaintext** (*bytes*) – Body plaintext
- **associated_data** (*bytes*) – Body AAD Data
- **iv** (*bytes*) – IV to use when encrypting message

Returns Deserialized object containing encrypted body

Return type *aws_encryption_sdk.internal.structures.EncryptedData*

3.27 aws_encryption_sdk.internal.crypto.iv

Helper functions used for generating deterministic initialization vectors (IVs).

Deterministic IVs are used to reduce the probability of IV/message-key pair collisions when caching data keys.

Prior to introducing caching, a statement could safely be made that every encrypt call resulted in a new data key which would only be used with a single message. With the introduction of caching, this statement by definition becomes false.

This is a problem because there are cryptographic limits on the number of times AES can be safely invoked using the same key (or using keys derived from the same key) and a random IV. In framed messages, this manifests as the total number of frames which can be safely encrypted under the same data key across all messages for which the data key is reused.

By using a random IV for each frame, we actually decrease the number of frames which can be safely encrypted under the same data key. Rather than attempting to track the number of frames across messages, we decided to move to a

deterministic IV constructed in such a way that it is guaranteed to never conflict within the same message. This means that we can consider only the likelihood of KDF collisions, which raises the limit sufficiently that we can assume that every message contains the maximum 2^32 invocations (2^32 - 1 frames + header auth).

Each IV is constructed from two big-endian byte arrays concatenated in the following order:

1. **64 bytes** : 0 (reserved space for possible future use)
2. **32 bytes** : frame sequence number (0 for the header auth calculation)

Functions

frame_iv(algorithm, sequence_number)	Builds the deterministic IV for a body frame.
header_auth_iv(algorithm)	Builds the deterministic IV for header authentication.
non_framed_body_iv(algorithm)	Builds the deterministic IV for a non-framed body.

aws_encryption_sdk.internal.crypto.iv.**frame_iv**(*algorithm*, *sequence_number*)
Builds the deterministic IV for a body frame.

Parameters

- **algorithm** (*aws_encryption_sdk.identifiers.Algorithm*) – Algorithm for which to build IV
- **sequence_number** (*int*) – Frame sequence number

Returns Generated IV

Return type bytes

Raises ***ActionNotAllowedError*** – if sequence number of out bounds

aws_encryption_sdk.internal.crypto.iv.**header_auth_iv**(*algorithm*)
Builds the deterministic IV for header authentication.

Parameters **algorithm** (*aws_encryption_sdk.identifiers.Algorithm*) – Algorithm for which to build IV

Returns Generated IV

Return type bytes

aws_encryption_sdk.internal.crypto.iv.**non_framed_body_iv**(*algorithm*)
Builds the deterministic IV for a non-framed body.

Parameters **algorithm** (*aws_encryption_sdk.identifiers.Algorithm*) – Algorithm for which to build IV

Returns Generated IV

Return type bytes

3.28 aws_encryption_sdk.internal.crypto.wrapping_keys

Contains wrapping key primitives.

Classes

WrappingKey(wrapping_algorithm, ... [, password])	Creates a wrapping encryption key object to encrypt and decrypt data keys.

class aws_encryption_sdk.internal.crypto.wrapping_keys.**WrappingKey**(*wrapping_algorithm*, *wrapping_key*, *wrapping_key_type*, *password=None*)

Bases: object

Creates a wrapping encryption key object to encrypt and decrypt data keys.

For use inside *aws_encryption_sdk.key_providers.raw.RawMasterKeyProvider* objects.

Parameters

- **wrapping_algorithm** (aws_encryption_sdk.identifiers.WrappingAlgorithm) – Wrapping Algorithm with which to wrap plaintext_data_key
- **wrapping_key** (*bytes*) – Encryption key with which to wrap plaintext_data_key
- **wrapping_key_type** (aws_encryption_sdk.identifiers.EncryptionKeyType) – Type of encryption key with which to wrap plaintext_data_key
- **password** (*bytes*) – Password to decrypt wrapping_key (optional, currently only relevant for RSA)

Prepares initial values.

decrypt(*encrypted_wrapped_data_key*, *encryption_context*)

Decrypts a wrapped, encrypted, data key.

Parameters

- **encrypted_wrapped_data_key** (aws_encryption_sdk.internal.structures.EncryptedData) – Encrypted, wrapped, data key
- **encryption_context** (*dict*) – Encryption context to use in decryption

Returns Plaintext of data key

Return type bytes

encrypt(*plaintext_data_key*, *encryption_context*)

Encrypts a data key using a direct wrapping key.

Parameters

- **plaintext_data_key** (*bytes*) – Data key to encrypt
- **encryption_context** (*dict*) – Encryption context to use in encryption

Returns Deserialized object containing encrypted key

Return type *aws_encryption_sdk.internal.structures.EncryptedData*

3.29 aws_encryption_sdk.internal.defaults

Default values for AWS Encryption SDK.

3.30 aws_encryption_sdk.internal.formatting

Formatting functions for aws_encryption_sdk.

Functions

body_length(header, plaintext_length)	Calculates the ciphertext message body length, given a complete header.
ciphertext_length(header, plaintext_length)	Calculates the complete ciphertext message length, given a complete header.
footer_length(header)	Calculates the ciphertext message footer length, given a complete header.
header_length(header)	Calculates the ciphertext message header length, given a complete header.

aws_encryption_sdk.internal.formatting.**body_length**(*header*, *plaintext_length*)
Calculates the ciphertext message body length, given a complete header.

Parameters

- **header** (aws_encryption_sdk.structures.MessageHeader) – Complete message header object
- **plaintext_length** (*int*) – Length of plaintext in bytes

Return type int

aws_encryption_sdk.internal.formatting.**ciphertext_length**(*header*, *plaintext_length*)
Calculates the complete ciphertext message length, given a complete header.

Parameters

- **header** (aws_encryption_sdk.structures.MessageHeader) – Complete message header object
- **plaintext_length** (*int*) – Length of plaintext in bytes

Return type int

aws_encryption_sdk.internal.formatting.**footer_length**(*header*)
Calculates the ciphertext message footer length, given a complete header.

Parameters **header** (aws_encryption_sdk.structures.MessageHeader) – Complete message header object

Return type int

aws_encryption_sdk.internal.formatting.**header_length**(*header*)
Calculates the ciphertext message header length, given a complete header.

Parameters **header** (aws_encryption_sdk.structures.MessageHeader) – Complete message header object

Return type int

3.31 aws_encryption_sdk.internal.formatting.deserialize

Components for handling AWS Encryption SDK message deserialization.

Functions

deserialize_footer(stream[, verifier])	Deserializes a footer.
deserialize_frame(stream, header[, verifier])	Deserializes a frame from a body.
deserialize_header(stream)	Deserializes the header from a source stream
deserialize_header_auth(stream, algorithm[, ...])	Deserializes a MessageHeaderAuthentication object from a source stream.
deserialize_non_framed_values(stream, header)	Deserializes the IV and body length from a non-framed stream.
deserialize_tag(stream, header[, verifier])	Deserialize the Tag value from a non-framed stream.
deserialize_wrapped_key(wrapping_algorithm, ...)	Extracts and deserializes EncryptedData from a Wrapped EncryptedDataKey.
unpack_values(format_string, stream[, verifier])	Helper function to unpack struct data from a stream and update the signature verifier.
validate_header(header, header_auth, ...)	Validates the header using the header authentication data.

aws_encryption_sdk.internal.formatting.deserialize.**deserialize_footer**(*stream*, *verifier=None*)

Deserializes a footer.

Parameters

- **stream** (*io.BytesIO*) – Source data stream
- **verifier** (*aws_encryption_sdk.internal.crypto.Verifier*) – Signature verifier object (optional)

Returns Deserialized footer

Return type *aws_encryption_sdk.internal.structures.MessageFooter*

Raises ***SerializationError*** – if verifier supplied and no footer found

aws_encryption_sdk.internal.formatting.deserialize.**deserialize_frame**(*stream*, *header*, *verifier=None*)

Deserializes a frame from a body.

Parameters

- **stream** (*io.BytesIO*) – Source data stream
- **header** (aws_encryption_sdk.structures.MessageHeader) – Deserialized header
- **verifier** (*aws_encryption_sdk.internal.crypto.Verifier*) – Signature verifier object (optional)

Returns Deserialized frame and a boolean stating if this is the final frame

Return type *aws_encryption_sdk.internal.structures.MessageFrameBody* and bool

aws_encryption_sdk.internal.formatting.deserialize.**deserialize_header**(*stream*)

Deserializes the header from a source stream

Parameters **stream** (*io.BytesIO*) – Source data stream

Returns Deserialized MessageHeader object

Return type *aws_encryption_sdk.structures.MessageHeader* and bytes

Raises

- ***NotSupportedError*** – if unsupported data types are found
- ***UnknownIdentityError*** – if unknown data types are found
- ***SerializationError*** – if IV length does not match algorithm

aws_encryption_sdk.internal.formatting.deserialize.**deserialize_header_auth**(*stream*, *algorithm*, *verifier=None*)

Deserializes a MessageHeaderAuthentication object from a source stream.

Parameters

- **stream** (*io.BytesIO*) – Source data stream
- **algorithm** – The AlgorithmSuite object type contained in the header
- **verifier** (*aws_encryption_sdk.internal.crypto.Verifier*) – Signature verifier object (optional)

Returns Deserialized MessageHeaderAuthentication object

Return type *aws_encryption_sdk.internal.structures.MessageHeaderAuthentication*

aws_encryption_sdk.internal.formatting.deserialize.**deserialize_non_framed_values**(*stream*, *header*, *verifier=None*)

Deserializes the IV and body length from a non-framed stream.

Parameters

- **stream** (*io.BytesIO*) – Source data stream
- **header** (aws_encryption_sdk.structures.MessageHeader) – Deserialized header
- **verifier** (*aws_encryption_sdk.internal.crypto.Verifier*) – Signature verifier object (optional)

Returns IV and Data Length values for body

Return type tuple of bytes and int

aws_encryption_sdk.internal.formatting.deserialize.**deserialize_tag**(*stream*, *header*, *verifier=None*)

Deserialize the Tag value from a non-framed stream.

Parameters

- **stream** (*io.BytesIO*) – Source data stream
- **header** (aws_encryption_sdk.structures.MessageHeader) – Deserialized header
- **verifier** (*aws_encryption_sdk.internal.crypto.Verifier*) – Signature verifier object (optional)

Returns Tag value for body

Return type bytes

aws_encryption_sdk.internal.formatting.deserialize.**deserialize_wrapped_key**(*wrapping_algorithm*, *wrapping_key_id*, *wrapped_encrypted_ke*

Extracts and deserializes EncryptedData from a Wrapped EncryptedDataKey.

Parameters

- **wrapping_algorithm** (aws_encryption_sdk.identifiers.WrappingAlgorithm) – Wrapping Algorithm with which to wrap plaintext_data_key
- **wrapping_key_id** (*bytes*) – Key ID of wrapping MasterKey
- **wrapped_encrypted_key** (aws_encryption_sdk.structures.EncryptedDataKey) – Raw Wrapped EncryptedKey

Returns EncryptedData of deserialized Wrapped EncryptedKey

Return type *aws_encryption_sdk.internal.structures.EncryptedData*

Raises

- ***SerializationError*** – if wrapping_key_id does not match deserialized wrapping key id
- ***SerializationError*** – if wrapping_algorithm IV length does not match deserialized IV length

aws_encryption_sdk.internal.formatting.deserialize.**unpack_values**(*format_string*, *stream*, *verifier=None*)

Helper function to unpack struct data from a stream and update the signature verifier.

Parameters

- **format_string** (*str*) – Struct format string
- **stream** (*io.BytesIO*) – Source data stream
- **verifier** (*aws_encryption_sdk.internal.crypto.Verifier*) – Signature verifier object

Returns Unpacked values

Return type tuple

aws_encryption_sdk.internal.formatting.deserialize.**validate_header**(*header*, *header_auth*, *raw_header*, *data_key*)

Validates the header using the header authentication data.

Parameters

- **header** (aws_encryption_sdk.structures.MessageHeader) – Deserialized header
- **header_auth** (aws_encryption_sdk.internal.structures.MessageHeaderAuthentication) – Deserialized header auth
- **raw_header** (*bytes*) – Raw header bytes
- **data_key** (*bytes*) – Data key with which to perform validation

Raises ***SerializationError*** – if header authorization fails

3.32 aws_encryption_sdk.internal.formatting.encryption_context

Components for handling serialization and deserialization of encryption context data in AWS Encryption SDK messages.

Functions

assemble_content_aad(message_id, ...)	Assembles the Body AAD string for a message body structure.
deserialize_encryption_context(...)	Deserializes the contents of a byte string into a dictionary.
read_short(source, offset)	Reads a number from a byte array.
read_string(source, offset, length)	Reads a string from a byte string.
serialize_encryption_context(encryption_context)	Serializes the contents of a dictionary into a byte string.

aws_encryption_sdk.internal.formatting.encryption_context.**assemble_content_aad**(*message_id*, *aad_content_stri*, *seq_num*, *length*)

Assembles the Body AAD string for a message body structure.

Parameters

- **message_id** (*str*) – Message ID
- **aad_content_string** (aws_encryption_sdk.identifiers.ContentAADString) – ContentAADString object for frame type
- **seq_num** (*int*) – Sequence number of frame
- **length** (*int*) – Content Length

Returns Properly formatted AAD bytes for message body structure.

Return type bytes

Raises ***SerializationError*** – if aad_content_string is not known

aws_encryption_sdk.internal.formatting.encryption_context.**deserialize_encryption_context**(*se*

Deserializes the contents of a byte string into a dictionary.

Parameters **serialized_encryption_context** (*bytes*) – Source byte string containing serialized dictionary

Returns Deserialized encryption context

Return type dict

Raises

- ***SerializationError*** – if serialized encryption context is too large
- ***SerializationError*** – if duplicate key found in serialized encryption context
- ***SerializationError*** – if malformed data found in serialized encryption context

aws_encryption_sdk.internal.formatting.encryption_context.**read_short**(*source*, *offset*)

Reads a number from a byte array.

Parameters

- **source** (*bytes*) – Source byte string
- **offset** (*int*) – Point in byte string to start reading

Returns Read number and offset at point after read data

Return type tuple of ints

Raises SerializationError if unable to unpack

aws_encryption_sdk.internal.formatting.encryption_context.**read_string**(*source*, *offset*, *length*)

Reads a string from a byte string.

Parameters

- **source** (*bytes*) – Source byte string
- **offset** (*int*) – Point in byte string to start reading
- **length** (*int*) – Length of string to read

Returns Read string and offset at point after read data

Return type tuple of str and int

Raises ***SerializationError*** – if unable to unpack

aws_encryption_sdk.internal.formatting.encryption_context.**serialize_encryption_context**(*encry*

Serializes the contents of a dictionary into a byte string.

Parameters **encryption_context** (*dict*) – Dictionary of encrytion context keys/values.

Returns Serialized encryption context

Return type bytes

3.33 aws_encryption_sdk.internal.formatting.serialize

Components for handling AWS Encryption SDK message serialization.

Functions

serialize_encrypted_data_key(encrypted_data_key)	Serializes an encrypted data key.
serialize_footer(signer)	Uses the signer object which has been used to sign the message to generate the signature, then serializes that signature.
serialize_frame(algorithm, plaintext, ...[, ...])	Receives a message plaintext, breaks off a frame, encrypts and serializes the frame, and returns the encrypted frame and the remaining plaintext.
serialize_header(header[, signer])	Serializes a header object.
serialize_header_auth(algorithm, header, ...)	Creates serialized header authentication data.
serialize_non_framed_close(tag[, signer])	Serializes the closing block for a non-framed message body.
serialize_non_framed_open(algorithm, iv, ...)	Serializes the opening block for a non-framed message body.
serialize_raw_master_key_prefix(raw_master_key)	Produces the prefix that a RawMasterKey will always use for the key_info value of keys which require additional information.
serialize_wrapped_key(key_provider, ...)	Serializes EncryptedData into a Wrapped EncryptedDataKey.

aws_encryption_sdk.internal.formatting.serialize.**serialize_encrypted_data_key**(*encrypted_data_ke*

Serializes an encrypted data key.

New in version 1.3.0.

Parameters **encrypted_data_key** (aws_encryption_sdk.structures.EncryptedDataKey) – Encrypted data key to serialize

Returns Serialized encrypted data key

Return type bytes

aws_encryption_sdk.internal.formatting.serialize.**serialize_footer**(*signer*)

Uses the signer object which has been used to sign the message to generate the signature, then serializes that signature.

Parameters **signer** (*aws_encryption_sdk.internal.crypto.Signer*) – Cryptographic signer object

Returns Serialized footer

Return type bytes

aws_encryption_sdk.internal.formatting.serialize.**serialize_frame**(*algorithm*, *plaintext*, *message_id*, *data_encryption_key*, *frame_length*, *sequence_number*, *is_final_frame*, *signer=None*)

Receives a message plaintext, breaks off a frame, encrypts and serializes the frame, and returns the encrypted frame and the remaining plaintext.

Parameters

- **algorithm** (*aws_encryption_sdk.identifiers.Algorithm*) – Algorithm to use for encryption
- **plaintext** (*bytes*) – Source plaintext to encrypt and serialize
- **message_id** (*bytes*) – Message ID
- **data_encryption_key** (*bytes*) – Data key with which to encrypt message
- **frame_length** (*int*) – Length of the framed data
- **sequence_number** (*int*) – Sequence number for frame to be generated
- **is_final_frame** (*bool*) – Boolean stating whether or not this frame is a final frame
- **signer** (*aws_encryption_sdk.Signer*) – Cryptographic signer object (optional)

Returns Serialized frame and remaining plaintext

Return type tuple of bytes

Raises ***SerializationError*** – if number of frames is too large

aws_encryption_sdk.internal.formatting.serialize.**serialize_header**(*header*, *signer=None*)

Serializes a header object.

Parameters

- **header** (aws_encryption_sdk.structures.MessageHeader) – Header to serialize
- **signer** (*aws_encryption_sdk.internal.crypto.Signer*) – Cryptographic signer object (optional)

Returns Serialized header

Return type bytes

aws_encryption_sdk.internal.formatting.serialize.**serialize_header_auth**(*algorithm*, *header*, *data_encryption_key*, *signer=None*)

Creates serialized header authentication data.

Parameters

- **algorithm** (*aws_encryption_sdk.identifiers.Algorithm*) – Algorithm to use for encryption
- **header** (*bytes*) – Serialized message header
- **data_encryption_key** (*bytes*) – Data key with which to encrypt message
- **signer** (*aws_encryption_sdk.Signer*) – Cryptographic signer object (optional)

Returns Serialized header authentication data

Return type bytes

aws_encryption_sdk.internal.formatting.serialize.**serialize_non_framed_close**(*tag*, *signer=None*)

Serializes the closing block for a non-framed message body.

Parameters

- **tag** (*bytes*) – Auth tag value from body encryptor

- **signer** (*aws_encryption_sdk.internal.crypto.Signer*) – Cryptographic signer object (optional)

Returns Serialized body close block

Return type bytes

aws_encryption_sdk.internal.formatting.serialize.**serialize_non_framed_open**(*algorithm*, *iv*, *plaintext_length*, *signer=None*)

Serializes the opening block for a non-framed message body.

Parameters

- **algorithm** (*aws_encryption_sdk.identifiers.Algorithm*) – Algorithm to use for encryption
- **iv** (*bytes*) – IV value used to encrypt body
- **plaintext_length** (*int*) – Length of plaintext (and thus ciphertext) in body
- **signer** (*aws_encryption_sdk.internal.crypto.Signer*) – Cryptographic signer object (optional)

Returns Serialized body start block

Return type bytes

aws_encryption_sdk.internal.formatting.serialize.**serialize_raw_master_key_prefix**(*raw_master_k*

Produces the prefix that a RawMasterKey will always use for the key_info value of keys which require additional information.

Parameters **raw_master_key** (aws_encryption_sdk.key_providers.raw.RawMasterKey) – RawMasterKey for which to produce a prefix

Returns Serialized key_info prefix

Return type bytes

aws_encryption_sdk.internal.formatting.serialize.**serialize_wrapped_key**(*key_provider*, *wrapping_algorithm*, *wrapping_key_id*, *encrypted_wrapped_key*)

Serializes EncryptedData into a Wrapped EncryptedDataKey.

Parameters

- **key_provider** (aws_encryption_sdk.structures.MasterKeyInfo) – Info for Wrapping MasterKey
- **wrapping_algorithm** (aws_encryption_sdk.identifiers.WrappingAlgorithm) – Wrapping Algorithm with which to wrap plaintext_data_key
- **wrapping_key_id** (*bytes*) – Key ID of wrapping MasterKey
- **encrypted_wrapped_key** (aws_encryption_sdk.internal.structures.EncryptedData) – Encrypted data key

Returns Wrapped EncryptedDataKey

Return type *aws_encryption_sdk.structures.EncryptedDataKey*

3.34 aws_encryption_sdk.internal.str_ops

Helper functions for consistently obtaining str and bytes objects in both Python2 and Python3.

Functions

to_bytes(data)	Takes an input str or bytes object and returns an equivalent bytes object.
to_str(data)	Takes an input str or bytes object and returns an equivalent str object.

aws_encryption_sdk.internal.str_ops.**to_bytes**(*data*)
Takes an input str or bytes object and returns an equivalent bytes object.

Parameters **data** (*str or bytes*) – Input data

Returns Data normalized to bytes

Return type bytes

aws_encryption_sdk.internal.str_ops.**to_str**(*data*)
Takes an input str or bytes object and returns an equivalent str object.

Parameters **data** (*str or bytes*) – Input data

Returns Data normalized to str

Return type str

3.35 aws_encryption_sdk.internal.structures

Public data structures for aws_encryption_sdk.

Classes

EncryptedData(iv, ciphertext, tag)	Holds encrypted data.
MessageFooter(signature)	Deserialized message footer
MessageFrameBody(iv, ciphertext, tag, ...)	Deserialized message frame
MessageHeaderAuthentication(iv, tag)	Deserialized message header authentication
MessageNoFrameBody(iv, ciphertext, tag)	Deserialized message body with no framing

class aws_encryption_sdk.internal.structures.**EncryptedData**(*iv*, *ciphertext*, *tag*)
Bases: object

Holds encrypted data.

Parameters

- **iv** (*bytes*) – Initialization Vector

- **ciphertext** (*bytes*) – Ciphertext
- **tag** (*bytes*) – Encryption tag

class aws_encryption_sdk.internal.structures.**MessageFooter**(*signature*)

Bases: object

Deserialized message footer

Parameters **signature** (*bytes*) – Message signature

class aws_encryption_sdk.internal.structures.**MessageFrameBody**(*iv*, *ciphertext*, *tag*, *sequence_number*, *final_frame*)

Bases: object

Deserialized message frame

Parameters

- **iv** (*bytes*) – Initialization Vector
- **ciphertext** (*bytes*) – Ciphertext
- **tag** (*bytes*) – Encryption Tag
- **sequence_number** (*int*) – Frame sequence number
- **final_frame** (*bool*) – Identifies final frames

class aws_encryption_sdk.internal.structures.**MessageHeaderAuthentication**(*iv*, *tag*)

Bases: object

Deserialized message header authentication

Parameters

- **iv** (*bytes*) – Initialization Vector
- **tag** (*bytes*) – Encryption Tag

class aws_encryption_sdk.internal.structures.**MessageNoFrameBody**(*iv*, *ciphertext*, *tag*)

Bases: object

Deserialized message body with no framing

Parameters

- **iv** (*bytes*) – Initialization Vector
- **ciphertext** (*bytes*) – Ciphertext
- **tag** (*bytes*) – Encryption Tag

3.36 aws_encryption_sdk.internal.validators

Common attrs validators.

Functions

value_is_not_a_string(instance, attribute, value)	Technically a string is an iterable containing strings.

aws_encryption_sdk.internal.validators.**value_is_not_a_string**(*instance*, *attribute*, *value*)

Technically a string is an iterable containing strings.

This validator lets you accept other iterators but not strings.

3.37 aws_encryption_sdk.internal.utils

Helper utility functions for AWS Encryption SDK.

Functions

content_type(frame_length)	Returns the appropriate content type based on the frame length.
get_aad_content_string(content_type, ...)	Prepares the appropriate Body AAD Value for a message body.
message_id()	Generates a new message ID.
prep_stream_data(data)	Take an input and prepare it for use as a stream.
prepare_data_keys(primary_master_key, ...)	Prepares a DataKey to be used for encrypting message and list of EncryptedDataKey objects to be serialized into header.
source_data_key_length_check(...)	Validates that the supplied source_data_key's data_key is the correct length for the supplied algorithm's kdf_input_len value.
validate_frame_length(frame_length, algorithm)	Validates that frame length is within the defined limits and is compatible with the selected algorithm.

aws_encryption_sdk.internal.utils.**content_type**(*frame_length*)

Returns the appropriate content type based on the frame length.

Parameters **frame_length** (*int*) – Message frame length

Returns Appropriate content type based on frame length

Return type *aws_encryption_sdk.identifiers.ContentType*

aws_encryption_sdk.internal.utils.**get_aad_content_string**(*content_type*, *is_final_frame*)

Prepares the appropriate Body AAD Value for a message body.

Parameters

- **content_type** (aws_encryption_sdk.identifiers.ContentType) – Defines the type of content for which to prepare AAD String
- **is_final_frame** (*bool*) – Boolean stating whether this is the final frame in a body

Returns Appropriate AAD Content String

Return type bytes

Raises ***UnknownIdentityError*** – if unknown content type

aws_encryption_sdk.internal.utils.**message_id**()
Generates a new message ID.

Returns Message ID

Return type bytes

aws_encryption_sdk.internal.utils.**prep_stream_data**(*data*)
Take an input and prepare it for use as a stream.

Parameters **data** – Input data

Returns Prepared stream

Return type InsistentReaderBytesIO

aws_encryption_sdk.internal.utils.**prepare_data_keys**(*primary_master_key*, *master_keys*, *algorithm*, *encryption_context*)
Prepares a DataKey to be used for encrypting message and list of EncryptedDataKey objects to be serialized into header.

Parameters

- **primary_master_key** (aws_encryption_sdk.key_providers.base.MasterKey) – Master key with which to generate the encryption data key
- **master_keys** (list of *aws_encryption_sdk.key_providers.base.MasterKey*) – All master keys with which to encrypt data keys
- **algorithm** (*aws_encryption_sdk.identifiers.Algorithm*) – Algorithm to use for encryption
- **encryption_context** (*dict*) – Encryption context to use when generating data key

Return type tuple containing *aws_encryption_sdk.structures.DataKey* and set of *aws_encryption_sdk.structures.EncryptedDataKey*

aws_encryption_sdk.internal.utils.**source_data_key_length_check**(*source_data_key*, *algorithm*)
Validates that the supplied source_data_key's data_key is the correct length for the supplied algorithm's kdf_input_len value.

Parameters

- **source_data_key** (*aws_encryption_sdk.structures.RawDataKey* or *aws_encryption_sdk.structures.DataKey*) – Source data key object received from MasterKey decrypt or generate data_key methods
- **algorithm** (*aws_encryption_sdk.identifiers.Algorithm*) – Algorithm object which directs how this data key will be used

Raises ***InvalidDataKeyError*** – if data key length does not match required kdf input length

aws_encryption_sdk.internal.utils.**validate_frame_length**(*frame_length*, *algorithm*)
Validates that frame length is within the defined limits and is compatible with the selected algorithm.

Parameters

- **frame_length** (*int*) – Frame size in bytes
- **algorithm** (*aws_encryption_sdk.identifiers.Algorithm*) – Algorithm to use for encryption

Raises

- ***SerializationError*** – if frame size is negative or not a multiple of the algorithm block size
- ***SerializationError*** – if frame size is larger than the maximum allowed frame size

3.38 aws_encryption_sdk.keyrings.aws_kms._client_cache

boto3 client cache for use by client suppliers.

New in version 1.5.0.

Warning: No guarantee is provided on the modules and APIs within this namespace staying consistent. Directly reference at your own risk.

Classes

ClientCache(botocore_session, client_config)	Provide boto3 clients regional clients, caching by region.

class aws_encryption_sdk.keyrings.aws_kms._client_cache.**ClientCache**(*botocore_session*, *client_config*)

Bases: object

Provide boto3 clients regional clients, caching by region.

Any clients that throw an error when used are immediately removed from the cache.

New in version 1.5.0.

Parameters

- **botocore_session** (*botocore.session.Session*) – Botocore session to use when creating clients
- **client_config** (*botocore.config.Config*) – Config to use when creating client

client(*region_name*, *service*)

Get a client for the specified region and service.

Generate a new client if needed. Otherwise, retrieve an existing client from the internal cache.

Parameters

- **region_name** (*str*) – Client region
- **service** (*str*) – Client service

Return type botocore.client.BaseClient

CHAPTER 4

Changelog

4.1 1.5.0 – 2020-xx-xx

4.1.1 Major Features

- Add keyrings.
- Change one-step APIs to return a `CryptoResult` rather than a tuple.
  - Modified APIs: `aws_encryption_sdk.encrypt` and `aws_encryption_sdk.decrypt`.

Note: For backwards compatibility, `CryptoResult` also unpacks like a 2-member tuple. This allows for backwards compatibility with the previous outputs so this change should not break any existing consumers unless you are specifically relying on the output being an instance of `tuple`.

4.1.2 Deprecations

- Deprecate master key providers in favor of keyrings.
  - We still support using master key providers and are not removing them yet. When we decide to remove them, we will communicate that as defined in our versioning policy.
- Deprecate support for Python 3.4.
  - This does not mean that this library will no longer work or install with 3.4, but we are no longer testing against or advertising support for 3.4.

4.1.3 Documentation

- Added new examples demonstrating how to use APIs, keyrings, cryptographic materials managers, and master key providers. #221 #236 #239

4.2 1.4.1 – 2019-09-20

4.2.1 Bugfixes

- Fix region configuration override in botocore sessions. #190 #193

4.2.2 Minor

- Caching CMM must require that max age configuration value is greater than 0. #147 #172

4.3 1.4.0 – 2019-05-23

4.3.1 Minor

- Remove dependence on all `source_stream` APIs except for `read()`. #103

Potentially Backwards Incompatible

- Encryption streams no longer close the `source_stream` when they themselves close. If you are using context managers for all of your stream handling, this change will not affect you. However, if you have been relying on the `StreamDecryptor` or `StreamEncryptor` to close your `source_stream` for you, you will now need to close those streams yourself.
- `StreamDecryptor.body_start` and `StreamDecryptor.body_end`, deprecated in a prior release, have now been removed.

4.3.2 Maintenance

- Move all remaining `unittest` tests to `pytest`. #99

4.3.3 Bugfixes

- Fix `MasterKeyprovider.decrypt_data_key_from_list` error handling. #150

4.4 1.3.8 – 2018-11-15

4.4.1 Bugfixes

- Remove debug logging that may contain input data when encrypting non-default unframed messages. #105

4.4.2 Minor

- Add support to remove clients from `KMSMasterKeyProvider` client cache if they fail to connect to endpoint. #86
- Add support for SHA384 and SHA512 for use with RSA OAEP wrapping algorithms. #56
- Fix `streaming_client` classes to properly interpret short reads in source streams. #24

4.5 1.3.7 – 2018-09-20

4.5.1 Bugfixes

- Fix KMSMasterKeyProvider to determine the default region before trying to create the requested master keys. #83

4.6 1.3.6 – 2018-09-04

4.6.1 Bugfixes

- `StreamEncryptor` and `StreamDecryptor` should always report as readable if they are open. #73
- Allow duck-typing of source streams. #75

4.7 1.3.5 – 2018-08-01

- Move the `aws-encryption-sdk-python` repository from `awslabs` to `aws`.

4.8 1.3.4 – 2018-04-12

4.8.1 Bugfixes

- AWS KMS master key/provider user agent extension fixed. #47

4.8.2 Maintenance

- New minimum pytest version 3.3.1 to avoid bugs in 3.3.0 #32
- New minimum attrs version 17.4.0 to allow use of `converter` rather than `convert` #39
- Algorithm Suites are modeled as collections of sub-suites now #36
- Selecting test suites is more sane now, with pytest markers. #41

4.9 1.3.3 – 2017-12-05

4.9.1 Bugfixes

- Remove use of attrs functionality deprecated in 17.3.0 #29

4.9.2 Maintenance

- Blacklisted pytest 3.3.0 #32 pytest-dev/pytest#2957

4.10 1.3.2 – 2017-09-28

- Addressed issue #13 to properly handle non-seekable source streams.

4.11 1.3.1 – 2017-09-12

4.11.1 Reorganization

- Moved source into `src`.
- Moved examples into `examples`.
- Broke out `internal.crypto` into smaller, feature-oriented, modules.

4.11.2 Tooling

- Added tox configuration to support automation and development tooling.
- Added pylint, flake8, and doc8 configuration to enforce style rules.

4.11.3 Maintenance

- Updated `internal.crypto.authentication.Verifier` to use `Prehashed`.
- Addressed docstring issue #7.
- Addressed docstring issue #8.
- Addressed logging issue #10.
- Addressed assorted linting issues to bring source, tests, examples, and docs up to configured linting standards.

4.12 1.3.0 – 2017-08-04

4.12.1 Major

- Added cryptographic materials managers as a concept
- Added data key caching

- Moved to deterministic IV generation

4.12.2 Minor

- Added changelog
- Fixed attrs usage to provide consistent behavior with 16.3.0 and 17.x
- Fixed performance bug which caused KDF calculations to be performed too frequently
- Removed `line_length` as a configurable parameter of `EncryptingStream` and `DecryptingStream` objects to simplify class APIs after it was found in further testing to have no measurable impact on performance
- Added deterministic length eliptic curve signature generation
- Added support for calculating ciphertext message length from header
- Migrated README from md to rst

4.13 1.2.2 – 2017-05-23

- Fixed `attrs` version to 16.3.0 to avoid breaking changes in attrs 17.1.0

4.14 1.2.0 – 2017-03-21

- Initial public release

Python Module Index

a

Index

A

B

C

D

E

F

L

M

N

O

P

R

S

T

U

V

W