
dynamodb-encryption-sdk-python

Release 3.0.0

Jul 15, 2021

Encryption Helpers

1	Getting Started	3
1.1	Required Prerequisites	3
1.2	Installation	3
1.3	Concepts	3
2	Usage	5
2.1	Helper Clients	5
2.2	Item Encryptor	6
2.3	Advanced Use	6
3	API	7
3.1	Cryptographic Configuration	7
3.2	Helper Clients	8
3.3	Item Encryptor	12
3.4	Cryptographic Materials Providers	14
3.5	Provider Stores	18
3.6	Cryptographic Materials	20
3.7	Delegated Keys	23
3.8	Tools	24
3.9	Exceptions	27
3.10	Internal Resources	29
4	Changelog	51
4.1	3.0.0 – 2021-07-15	51
4.2	2.1.0 – 2021-07-15	51
4.3	2.0.0 – 2021-02-04	51
4.4	1.3.0 – 2021-02-04	52
4.5	1.2.0 – 2019-10-10	52
4.6	1.1.1 – 2019-08-29	52
4.7	1.1.0 – 2019-03-13	52
4.8	1.0.7 – 2018-01-16	52
4.9	1.0.6 – 2018-01-15	53
4.10	1.0.5 – 2018-08-01	53
4.11	1.0.4 – 2018-05-22	53
4.12	1.0.3 – 2018-05-03	53
4.13	1.0.2 – 2018-05-03	53
4.14	1.0.1 – 2018-05-02	53

4.15 1.0.0 – 2018-05-02	53
Python Module Index	55
Index	57

The [Amazon DynamoDB Encryption Client for Python](#) provides client-side encryption of [Amazon DynamoDB](#) items to help you to protect your table data before you send it to DynamoDB. It provides an implementation of the [Amazon DynamoDB Encryption Client](#) that is fully compatible with the [Amazon DynamoDB Encryption Client for Java](#).

You can find the latest Python documentation at [Read the Docs](#) and you can find the latest full documents in our [primary documents](#).

You can find our source on [GitHub](#).

[Security issue notifications](#)

See [Support Policy](#) for details on the current support status of all major versions of this library.

1.1 Required Prerequisites

- Python 3.5+

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 dynamodb-encryption-sdk
```

1.3 Concepts

For a detailed description of the concepts that are important to understand when using this client, please review our [Concepts Guide](#).

2.1 Helper Clients

We provide helper clients that look and feel like the low level client (`EncryptedClient`), service resource (`EncryptedResource`), and table resource (`EncryptedTable`) available from the `boto3` library. For most uses, once configured, these clients can be used exactly as you would a standard client from `boto3`, and your items will be transparently encrypted on write and decrypted on read.

2.1.1 What can't I do with the helper clients?

For most uses, the helper clients (once configured) can be used as drop-in replacements for the `boto3` clients. However, there are a couple cases where this is not the case.

Update Item

Because we can't know that a partial update you might be making to an item covers all of the signed attributes in your item, we do not allow `update_item` on the helper clients.

This is because if you update only some of the signed attributes, then next time you try to read that item the signature validation will fail.

Attribute Filtering

Because we can't know what attributes in an item are signed, the helper clients do not allow any attribute filtering.

For `get_item`, `batch_get_item`, and `scan`, this includes the use of `AttributesToGet` and `ProjectionExpression`.

For `scan`, this also includes the use of `Select` values `SPECIFIC_ATTRIBUTES` and `ALL_PROJECTED_ATTRIBUTES`.

This is because if you do not retrieve all signed attributes, the signature validation will fail.

2.2 Item Encryptor

The helper clients provide a familiar interface but the actual item encryption and decryption is handled by a low-level item encryptor. You usually will not need to interact with these low-level functions, but for certain advanced use cases it can be useful.

If you do choose to use the item encryptor functions directly, you will need to provide a `CryptoConfig` for each call.

```
>>> from dynamodb_encryption_sdk.encrypted.item import decrypt_python_item, encrypt_
↳python_item
>>> plaintext_item = {
...     'some': 'data',
...     'more': 5
... }
>>> encrypted_item = encrypt_python_item(
...     item=plaintext_item,
...     crypto_config=my_crypto_config
... )
>>> decrypted_item = decrypt_python_item(
...     item=encrypted_item,
...     crypto_config=my_crypto_config
... )
```

2.2.1 When should I use the item encryptor?

One example of a use case where you might want to use the item encryptor directly is when processing items in a DynamoDB Stream. Since you receive the items data directly, and in DynamoDB JSON format, you can use the `decrypt_dynamodb_item` function to decrypt the item in the stream. We also provide helper transformation functions

2.3 Advanced Use

By default, the helper clients use your attribute actions and cryptographic materials provider to build the `CryptoConfig` that is provided to the item encryptor. For some advanced use cases, you might want to provide a custom `CryptoConfig` for specific operations.

All data plane operations (get item, put item, etc) on helper clients accept a `crypto_config` parameter in addition to all of the parameters that the underlying `boto3` client accepts.

If this parameter is supplied, that `CryptoConfig` will be used for that operation instead of the one that the client would normally construct for you.

```
>>> from dynamodb_encryption_sdk.encrypted.table import EncryptedTable
>>> encrypted_table = EncryptedTable(
...     table=table,
...     materials_provider=my_crypto_materials_provider
... )
>>> encrypted_table.put_item(
...     Item=my_standard_item
... ) # this uses the crypto config built by the helper
>>> encrypted_table.put_item(
...     Item=my_special_item,
...     crypto_config=my_special_crypto_config
... ) # this uses my_special_crypto_config
```

3.1 Cryptographic Configuration

Resources for encrypting items.

```
class dynamodb_encryption_sdk.encrypted.CryptoConfig (materials_provider,          en-  
                                                    crypton_context,          at-  
                                                    tribute_actions)
```

Bases: `object`

Container for all configuration needed to encrypt or decrypt an item using the item encryptor functions in `dynamodb_encryption_sdk.encrypted.item`.

Parameters

- **materials_provider** (`CryptographicMaterialsProvider`) – Cryptographic materials provider to use
- **encryption_context** (`EncryptionContext`) – Context data describing what is being encrypted or decrypted
- **attribute_actions** (`AttributeActions`) – Description of what action should be taken for each attribute

decryption_materials ()

Load decryption materials from instance resources.

Returns Decryption materials

Return type `CryptographicMaterials`

encryption_materials ()

Load encryption materials from instance resources.

Returns Encryption materials

Return type `CryptographicMaterials`

copy()

Return a copy of this instance with a copied instance of its encryption context.

Returns New *CryptoConfig* identical to this one**Return type** *CryptoConfig***with_item(item)**

Return a copy of this instance with an encryption context that includes the provided item attributes.

Parameters *item* (*dict*) – DynamoDB item in DynamnoDB JSON format**Returns** New *CryptoConfig* identical to this one**Return type** *CryptoConfig*

3.2 Helper Clients

3.2.1 Encrypted Table Resource

High-level helper class to provide a familiar interface to encrypted tables.

```
class dynamodb_encryption_sdk.encrypted.table.EncryptedTable(table, materials_provider, table_info=None, attribute_actions=None, auto_refresh_table_indexes=True)
```

Bases: *object*

High-level helper class to provide a familiar interface to encrypted tables.

```
>>> import boto3
>>> from dynamodb_encryption_sdk.encrypted.table import EncryptedTable
>>> from dynamodb_encryption_sdk.material_providers.aws_kms import _
    ↪ AwsKmsCryptographicMaterialsProvider
>>> table = boto3.resource('dynamodb').Table('my_table')
>>> aws_kms_cmp = AwsKmsCryptographicMaterialsProvider('alias/MyKmsAlias')
>>> encrypted_table = EncryptedTable(
...     table=table,
...     materials_provider=aws_kms_cmp
... )
```

Note: This class provides a superset of the boto3 DynamoDB Table API, so should work as a drop-in replacement once configured.<https://boto3.readthedocs.io/en/latest/reference/services/dynamodb.html#table>If you want to provide per-request cryptographic details, the `put_item`, `get_item`, `query`, and `scan` methods will also accept a `crypto_config` parameter, defining a custom `CryptoConfig` instance for this request.**Warning:** We do not currently support the `update_item` method.

Parameters

- **table** (*boto3.resources.base.ServiceResource*) – Pre-configured boto3 DynamoDB Table object
- **materials_provider** (*CryptographicMaterialsProvider*) – Cryptographic materials provider to use
- **table_info** (*TableInfo*) – Information about the target DynamoDB table
- **attribute_actions** (*AttributeActions*) – Table-level configuration of how to encrypt/sign attributes
- **auto_refresh_table_indexes** (*bool*) – Should we attempt to refresh information about table indexes? Requires `dynamodb:DescribeTable` permissions on each table. (default: True)

update_item (***kwargs*)

Update item is not yet supported.

batch_writer (*overwrite_by_pkeys=None*)

Create a batch writer object.

https://boto3.readthedocs.io/en/latest/reference/services/dynamodb.html#DynamoDB.Table.batch_writer

Parameters `overwrite_by_pkeys` (*list(string)*) – De-duplicate request items in buffer if match new request item on specified primary keys. i.e ["partition_key1", "sort_key2", "sort_key3"]

3.2.2 Encrypted Service Resource

High-level helper class to provide a familiar interface to encrypted tables.

class `dynamodb_encryption_sdk.encrypted.resource.EncryptedTablesCollectionManager` (*collection, materials_provider, attribute_actions, table_info_cache*)

Bases: `object`

Tables collection manager that provides `EncryptedTable` objects.

<https://boto3.readthedocs.io/en/latest/reference/services/dynamodb.html#DynamoDB.ServiceResource.tables>

Parameters

- **collection** (*boto3.resources.collection.CollectionManager*) – Pre-configured boto3 DynamoDB table collection manager
- **materials_provider** (*CryptographicMaterialsProvider*) – Cryptographic materials provider to use
- **attribute_actions** (*AttributeActions*) – Table-level configuration of how to encrypt/sign attributes
- **table_info_cache** (*TableInfoCache*) – Local cache from which to obtain Table-Info data

```
class dynamodb_encryption_sdk.encrypted.resource.EncryptedResource (resource,
                                                                    materi-
                                                                    als_provider,
                                                                    at-
                                                                    tribute_actions=None,
                                                                    auto_refresh_table_indexes=True)
```

Bases: `object`

High-level helper class to provide a familiar interface to encrypted tables.

```
>>> import boto3
>>> from dynamodb_encryption_sdk.encrypted.resource import EncryptedResource
>>> from dynamodb_encryption_sdk.material_providers.aws_kms import _
↳AwsKmsCryptographicMaterialsProvider
>>> resource = boto3.resource('dynamodb')
>>> aws_kms_cmp = AwsKmsCryptographicMaterialsProvider('alias/MyKmsAlias')
>>> encrypted_resource = EncryptedResource(
...     resource=resource,
...     materials_provider=aws_kms_cmp
... )
```

Note: This class provides a superset of the boto3 DynamoDB service resource API, so should work as a drop-in replacement once configured.

<https://boto3.readthedocs.io/en/latest/reference/services/dynamodb.html#service-resource>

If you want to provide per-request cryptographic details, the `batch_write_item` and `batch_get_item` methods will also accept a `crypto_config` parameter, defining a custom `CryptoConfig` instance for this request.

Parameters

- **resource** (*boto3.resources.base.ServiceResource*) – Pre-configured boto3 DynamoDB service resource object
- **materials_provider** (*CryptographicMaterialsProvider*) – Cryptographic materials provider to use
- **attribute_actions** (*AttributeActions*) – Table-level configuration of how to encrypt/sign attributes
- **auto_refresh_table_indexes** (*bool*) – Should we attempt to refresh information about table indexes? Requires `dynamodb:DescribeTable` permissions on each table. (default: `True`)

Table (*name, **kwargs*)

Creates an `EncryptedTable` resource.

If any of the optional configuration values are not provided, the corresponding values for this `EncryptedResource` will be used.

<https://boto3.readthedocs.io/en/latest/reference/services/dynamodb.html#DynamoDB.ServiceResource.Table>

Parameters

- **name** – The table name.

- **materials_provider** (*CryptographicMaterialsProvider*) – Cryptographic materials provider to use (optional)
- **table_info** (*TableInfo*) – Information about the target DynamoDB table (optional)
- **attribute_actions** (*AttributeActions*) – Table-level configuration of how to encrypt/sign attributes (optional)

3.2.3 Encrypted Client

High-level helper class to provide a familiar interface to encrypted tables.

```
class dynamodb_encryption_sdk.encrypted.client.EncryptedPaginator (paginator,  
de-  
crypt_method,  
crypto_config_method)
```

Bases: `object`

Paginator that decrypts returned items before returning them.

Parameters

- **paginator** (*botocore.paginate.Paginator*) – Pre-configured boto3 DynamoDB paginator object
- **decrypt_method** – Item decryptor method from *dynamodb_encryption_sdk.encrypted.item*
- **crypto_config_method** (*callable*) – Callable that returns a *CryptoConfig*

```
validate_decrypt_method (attribute, value)  
Validate that _decrypt_method is one of the item encryptors.
```

```
paginate (**kwargs)  
Create an iterator that will paginate through responses from the underlying paginator, transparently decrypting any returned items.
```

```
class dynamodb_encryption_sdk.encrypted.client.EncryptedClient (client, materials_provider,  
attribute_actions=None,  
auto_refresh_table_indexes=True,  
expect_standard_dictionaries=False)
```

Bases: `object`

High-level helper class to provide a familiar interface to encrypted tables.

```
>>> import boto3  
>>> from dynamodb_encryption_sdk.encrypted.client import EncryptedClient  
>>> from dynamodb_encryption_sdk.material_providers.aws_kms import _  
↳ AwsKmsCryptographicMaterialsProvider  
>>> client = boto3.client('dynamodb')  
>>> aws_kms_cmp = AwsKmsCryptographicMaterialsProvider('alias/MyKmsAlias')  
>>> encrypted_client = EncryptedClient(  
...     client=client,  
...     materials_provider=aws_kms_cmp  
... )
```

Note: This class provides a superset of the boto3 DynamoDB client API, so should work as a drop-in replacement once configured.

<https://boto3.readthedocs.io/en/latest/reference/services/dynamodb.html#client>

If you want to provide per-request cryptographic details, the `put_item`, `get_item`, `query`, `scan`, `batch_write_item`, and `batch_get_item` methods will also accept a `crypto_config` parameter, defining a custom `CryptoConfig` instance for this request.

Warning: We do not currently support the `update_item` method.

Parameters

- **client** (*boto3.resources.base.BaseClient*) – Pre-configured boto3 DynamoDB client object
- **materials_provider** (*CryptographicMaterialsProvider*) – Cryptographic materials provider to use
- **attribute_actions** (*AttributeActions*) – Table-level configuration of how to encrypt/sign attributes
- **auto_refresh_table_indexes** (*bool*) – Should we attempt to refresh information about table indexes? Requires `dynamodb:DescribeTable` permissions on each table. (default: True)
- **expect_standard_dictionaries** (*bool*) – Should we expect items to be standard Python dictionaries? This should only be set to True if you are using a client obtained from a service resource or table resource (ex: `table.meta.client`). (default: False)

`update_item` (**kwargs)

Update item is not yet supported.

Raises `NotImplementedError` – if called

`get_paginator` (*operation_name*)

Get a paginator from the underlying client. If the paginator requested is for “scan” or “query”, the paginator returned will transparently decrypt the returned items.

Parameters `operation_name` (*str*) – Name of operation for which to get paginator

Returns Paginator for name

Return type `botocore.paginate.Paginator` or *EncryptedPaginator*

<code>dynamodb_encryption_sdk.encrypted.table</code>	High-level helper class to provide a familiar interface to encrypted tables.
<code>dynamodb_encryption_sdk.encrypted.resource</code>	High-level helper class to provide a familiar interface to encrypted tables.
<code>dynamodb_encryption_sdk.encrypted.client</code>	High-level helper class to provide a familiar interface to encrypted tables.

3.3 Item Encryptor

Top-level functions for encrypting and decrypting DynamoDB items.

`dynamodb_encryption_sdk.encrypted.item.encrypt_dynamodb_item` (*item*,
crypto_config)

Encrypt a DynamoDB item.

```
>>> from dynamodb_encryption_sdk.encrypted.item import encrypt_dynamodb_item
>>> plaintext_item = {
...     'some': {'S': 'data'},
...     'more': {'N': '5'}
... }
>>> encrypted_item = encrypt_dynamodb_item(
...     item=plaintext_item,
...     crypto_config=my_crypto_config
... )
```

Note: This handles DynamoDB-formatted items and is for use with the boto3 DynamoDB client.

Parameters

- **item** (*dict*) – Plaintext DynamoDB item
- **crypto_config** (*CryptoConfig*) – Cryptographic configuration

Returns Encrypted and signed DynamoDB item

Return type *dict*

`dynamodb_encryption_sdk.encrypted.item.encrypt_python_item(item, crypto_config)`

Encrypt a dictionary for DynamoDB.

```
>>> from dynamodb_encryption_sdk.encrypted.item import encrypt_python_item
>>> plaintext_item = {
...     'some': 'data',
...     'more': 5
... }
>>> encrypted_item = encrypt_python_item(
...     item=plaintext_item,
...     crypto_config=my_crypto_config
... )
```

Note: This handles human-friendly dictionaries and is for use with the boto3 DynamoDB service or table resource.

Parameters

- **item** (*dict*) – Plaintext dictionary
- **crypto_config** (*CryptoConfig*) – Cryptographic configuration

Returns Encrypted and signed dictionary

Return type *dict*

`dynamodb_encryption_sdk.encrypted.item.decrypt_dynamodb_item(item, crypto_config)`

Decrypt a DynamoDB item.

```
>>> from dynamodb_encryption_sdk.encrypted.item import decrypt_python_item
>>> encrypted_item = {
...     'some': {'B': b'ENCRYPTED_DATA'},
...     'more': {'B': b'ENCRYPTED_DATA'}
... }
>>> decrypted_item = decrypt_python_item(
...     item=encrypted_item,
...     crypto_config=my_crypto_config
... )
```

Note: This handles DynamoDB-formatted items and is for use with the boto3 DynamoDB client.

Parameters

- **item** (*dict*) – Encrypted and signed DynamoDB item
- **crypto_config** (*CryptoConfig*) – Cryptographic configuration

Returns Plaintext DynamoDB item

Return type *dict*

`dynamodb_encryption_sdk.encrypted.item.decrypt_python_item(item, crypto_config)`
Decrypt a dictionary for DynamoDB.

```
>>> from dynamodb_encryption_sdk.encrypted.item import decrypt_python_item
>>> encrypted_item = {
...     'some': Binary(b'ENCRYPTED_DATA'),
...     'more': Binary(b'ENCRYPTED_DATA')
... }
>>> decrypted_item = decrypt_python_item(
...     item=encrypted_item,
...     crypto_config=my_crypto_config
... )
```

Note: This handles human-friendly dictionaries and is for use with the boto3 DynamoDB service or table resource.

Parameters

- **item** (*dict*) – Encrypted and signed dictionary
- **crypto_config** (*CryptoConfig*) – Cryptographic configuration

Returns Plaintext dictionary

Return type *dict*

3.4 Cryptographic Materials Providers

3.4.1 AWS KMS Provider

Cryptographic materials provider for use with the AWS Key Management Service (KMS).

```
class dynamodb_encryption_sdk.material_providers.aws_kms.AwsKmsCryptographicMaterialsProvider
```

Bases: `dynamodb_encryption_sdk.material_providers.CryptographicMaterialsProvider`
 Cryptographic materials provider for use with the AWS Key Management Service (KMS).

Note: This cryptographic materials provider makes one AWS KMS API call each time encryption or decryption materials are requested. This means that one request will be made for each item that you read or write.

Parameters

- **key_id** (*str*) – ID of AWS KMS CMK to use
- **botocore_session** (*botocore.session.Session*) – botocore session object (optional)
- **grant_tokens** (*list*) – List of grant tokens to pass to KMS on CMK operations (optional)
- **material_description** (*dict*) – Material description to use as default state for this CMP (optional)
- **regional_clients** (*dict*) – Dictionary mapping AWS region names to pre-configured boto3 KMS clients (optional)

decryption_materials (*encryption_context*)

Provide decryption materials.

Parameters **encryption_context** (*EncryptionContext*) – Encryption context for request

Returns Encryption materials

Return type *RawDecryptionMaterials*

encryption_materials (*encryption_context*)

Provide encryption materials.

Parameters **encryption_context** (*EncryptionContext*) – Encryption context for request

Returns Encryption materials

Return type *RawEncryptionMaterials*

3.4.2 Wrapped Provider

Cryptographic materials provider to use ephemeral content encryption keys wrapped by delegated keys.

class dynamodb_encryption_sdk.material_providers.wrapped.**WrappedCryptographicMaterialsProvider**

Bases: dynamodb_encryption_sdk.material_providers.CryptographicMaterialsProvider
Cryptographic materials provider to use ephemeral content encryption keys wrapped by delegated keys.

Parameters

- **signing_key** (*DelegatedKey*) – Delegated key used as signing and verification key
- **wrapping_key** (*DelegatedKey*) – Delegated key used to wrap content key

Note: `wrapping_key` must be provided if providing encryption materials

Parameters **unwrapping_key** (*DelegatedKey*) – Delegated key used to unwrap content key

Note: `unwrapping_key` must be provided if providing decryption materials or loading materials from material description

encryption_materials (*encryption_context*)

Provide encryption materials.

Parameters **encryption_context** (*EncryptionContext*) – Encryption context for request

Returns Encryption materials

Return type *WrappedCryptographicMaterials*

Raises *WrappingError* – if no wrapping key is available

decryption_materials (*encryption_context*)

Provide decryption materials.

Parameters **encryption_context** (*EncryptionContext*) – Encryption context for request

Returns Decryption materials

Return type *WrappedCryptographicMaterials*

Raises *UnwrappingError* – if no unwrapping key is available

3.4.3 Most Recent Provider

Cryptographic materials provider that uses a provider store to obtain cryptographic materials.

class dynamodb_encryption_sdk.material_providers.most_recent.CachingMostRecentProvider (prov

ma-
te-
rial_
ver-
sion_
cach

Bases: dynamodb_encryption_sdk.material_providers.CryptographicMaterialsProvider

Cryptographic materials provider that uses a provider store to obtain cryptography materials. Materials obtained from the store are cached for a user-defined amount of time, then removed from the cache and re-retrieved from the store.

When encrypting, the most recent provider that the provider store knows about will always be used.

Parameters

- **provider_store** (*ProviderStore*) – Provider store to use
- **material_name** (*str*) – Name of materials for which to ask the provider store
- **version_ttl** (*float*) – Max time in seconds to go until checking with provider store for a more recent version
- **cache_size** (*int*) – The maximum number of entries that the cache can hold

decryption_materials (*encryption_context*)

Return decryption materials.

Parameters **encryption_context** (*EncryptionContext*) – Encryption context for request

Raises **AttributeError** – if no decryption materials are available

encryption_materials (*encryption_context*)

Return encryption materials.

Parameters **encryption_context** (*EncryptionContext*) – Encryption context for request

Raises **AttributeError** – if no encryption materials are available

refresh ()

Clear all local caches for this provider.

3.4.4 Static Provider

Cryptographic materials provider for use with pre-configured encryption and decryption materials.

class dynamodb_encryption_sdk.material_providers.static.StaticCryptographicMaterialsProvider

Bases: dynamodb_encryption_sdk.material_providers.CryptographicMaterialsProvider

Manually combine encryption and decryption materials for use as a cryptographic materials provider.

Parameters

- **decryption_materials** (*DecryptionMaterials*) – Decryption materials to provide (optional)

- **encryption_materials** (*EncryptionMaterials*) – Encryption materials to provide (optional)

decryption_materials (*encryption_context*)

Return the static decryption materials.

Parameters **encryption_context** (*EncryptionContext*) – Encryption context for request (not used by *StaticCryptographicMaterialsProvider*)

Raises **AttributeError** – if no decryption materials are available

encryption_materials (*encryption_context*)

Return the static encryption materials.

Parameters **encryption_context** (*EncryptionContext*) – Encryption context for request (not used by *StaticCryptographicMaterialsProvider*)

Raises **AttributeError** – if no encryption materials are available

<code>dynamodb_encryption_sdk. material_providers</code>	Cryptographic materials providers.
<code>dynamodb_encryption_sdk. material_providers.wrapped</code>	Cryptographic materials provider to use ephemeral content encryption keys wrapped by delegated keys.
<code>dynamodb_encryption_sdk. material_providers.most_recent</code>	Cryptographic materials provider that uses a provider store to obtain cryptographic materials.
<code>dynamodb_encryption_sdk. material_providers.static</code>	Cryptographic materials provider for use with pre-configured encryption and decryption materials.

3.5 Provider Stores

3.5.1 MetaStore

Meta cryptographic provider store.

class `dynamodb_encryption_sdk.material_providers.store.meta`.**MetaStore** (*table*,
materials_provider)

Bases: `dynamodb_encryption_sdk.material_providers.store.ProviderStore`

Create and retrieve wrapped cryptographic materials providers, storing their cryptographic materials using the provided encrypted table.

Parameters

- **table** (*boto3.resources.base.ServiceResource*) – Pre-configured boto3 DynamoDB Table object
- **materials_provider** (*CryptographicMaterialsProvider*) – Cryptographic materials provider to use

classmethod **create_table** (*client*, *table_name*, *read_units*, *write_units*)

Create the table for this MetaStore.

Parameters

- **table** (*boto3.resources.base.BaseClient*) – Pre-configured boto3 DynamoDB client object

- **table_name** (*str*) – Name of table to create
- **read_units** (*int*) – Read capacity units to provision
- **write_units** (*int*) – Write capacity units to provision

get_or_create_provider (*material_name, version*)

Obtain a cryptographic materials provider identified by a name and version.

If the requested version does not exist, a new one will be created.

Parameters

- **material_name** (*str*) – Material to locate
- **version** (*int*) – Version of material to locate

Returns cryptographic materials provider

Return type CryptographicMaterialsProvider

Raises *InvalidVersionError* – if the requested version is not available and cannot be created

provider (*material_name, version=None*)

Obtain a cryptographic materials provider identified by a name and version.

If the version is provided, an error will be raised if that version is not found.

If the version is not provided, the maximum version will be used.

Parameters

- **material_name** (*str*) – Material to locate
- **version** (*int*) – Version of material to locate (optional)

Returns cryptographic materials provider

Return type CryptographicMaterialsProvider

Raises *InvalidVersionError* – if the requested version is not found

version_from_material_description (*material_description*)

Determine the version from the provided material description.

Parameters **material_description** (*dict*) – Material description to use with this request

Returns version to use

Return type *int*

max_version (*material_name*)

Find the maximum known version of the specified material.

Parameters **material_name** (*str*) – Material to locate

Returns Maximum known version

Return type *int*

Raises *NoKnownVersion* – if no version can be found

dynamodb_encryption_sdk.
material_providers.store

Cryptographic materials provider stores.

Continued on next page

Table 3 – continued from previous page

<code>dynamodb_encryption_sdk.</code>	Meta cryptographic provider store.
<code>material_providers.store.meta</code>	

3.6 Cryptographic Materials

3.6.1 Wrapped Cryptographic Materials

Cryptographic materials to use ephemeral content encryption keys wrapped by delegated keys.

class `dynamodb_encryption_sdk.materials.wrapped.WrappedCryptographicMaterials` (*signing_key*, *wrapping_key=None*, *unwrapping_key=None*, *material_description=None*)

Bases: `dynamodb_encryption_sdk.materials.CryptographicMaterials`

Encryption/decryption key is a content key stored in the material description, wrapped by the wrapping key.

Parameters

- **signing_key** (*DelegatedKey*) – Delegated key used as signing and verification key
- **wrapping_key** (*DelegatedKey*) – Delegated key used to wrap content key

Note: `wrapping_key` must be provided if material description contains a wrapped content key

Parameters **unwrapping_key** (*DelegatedKey*) – Delegated key used to unwrap content key

Note: `unwrapping_key` must be provided if material description does not contain a wrapped content key

Parameters **material_description** (*dict*) – Material description to use with these cryptographic materials

material_description

Material description to use with these cryptographic materials.

Returns Material description

Return type `dict`

encryption_key

Content key used for encrypting attributes.

Returns Encryption key

Return type `DelegatedKey`

decryption_key

Content key used for decrypting attributes.

Returns Decryption key

Return type DelegatedKey

signing_key

Delegated key used for calculating digital signatures.

Returns Signing key

Return type DelegatedKey

verification_key

Delegated key used for verifying digital signatures.

Returns Verification key

Return type DelegatedKey

3.6.2 Raw Cryptographic Materials

Cryptographic materials classes for use directly with delegated keys.

Warning: Using raw cryptographic materials can be very dangerous because you are likely to be encrypting many items using the same encryption key material. This can have some unexpected and difficult to detect side effects that weaken the security of your encrypted data.

Unless you have specific reasons for using raw cryptographic materials, we highly recommend that you use wrapped cryptographic materials instead.

```
class dynamodb_encryption_sdk.materials.raw.RawEncryptionMaterials (signing_key,
                                                                    encryp-
                                                                    tion_key=None,
                                                                    mate-
                                                                    rial_description=None)
```

Bases: dynamodb_encryption_sdk.materials.EncryptionMaterials

Encryption materials for use directly with delegated keys.

Note: Not all delegated keys allow use with raw cryptographic materials.

Parameters

- **signing_key** (*DelegatedKey*) – Delegated key used as signing key
- **encryption_key** (*DelegatedKey*) – Delegated key used as encryption key
- **material_description** (*dict*) – Material description to use with these cryptographic materials

material_description

Material description to use with these cryptographic materials.

Returns Material description

Return type *dict*

signing_key

Delegated key used for calculating digital signatures.

Returns Signing key

Return type DelegatedKey

encryption_key

Delegated key used for encrypting attributes.

Returns Encryption key

Return type DelegatedKey

class dynamodb_encryption_sdk.materials.raw.RawDecryptionMaterials (*verification_key*,
*decryp-
tion_key=None*,
*mate-
rial_description=None*)

Bases: dynamodb_encryption_sdk.materials.DecryptionMaterials

Encryption materials for use directly with delegated keys.

Note: Not all delegated keys allow use with raw cryptographic materials.

Parameters

- **verification_key** (*DelegatedKey*) – Delegated key used as verification key
- **decryption_key** (*DelegatedKey*) – Delegated key used as decryption key
- **material_description** (*dict*) – Material description to use with these cryptographic materials

material_description

Material description to use with these cryptographic materials.

Returns Material description

Return type dict

verification_key

Delegated key used for verifying digital signatures.

Returns Verification key

Return type DelegatedKey

decryption_key

Delegated key used for decrypting attributes.

Returns Decryption key

Return type DelegatedKey

<code>dynamodb_encryption_sdk.materials</code>	Cryptographic materials are containers that provide delegated keys for cryptographic operations.
<code>dynamodb_encryption_sdk.materials.raw</code>	Cryptographic materials classes for use directly with delegated keys.
<code>dynamodb_encryption_sdk.materials.wrapped</code>	Cryptographic materials to use ephemeral content encryption keys wrapped by delegated keys.

3.7 Delegated Keys

3.7.1 JCE Standard Name Delegated Key

Delegated key that uses JCE StandardName values to determine behavior.

```
class dynamodb_encryption_sdk.delegated_keys.jce.JceNameLocalDelegatedKey (key,
                                                                    al-
                                                                    go-
                                                                    rithm,
                                                                    key_type,
                                                                    key_encoding)
```

Bases: `dynamodb_encryption_sdk.delegated_keys.DelegatedKey`

Delegated key that uses JCE StandardName values to determine behavior.

Accepted algorithm names for this include:

- JCE Mac names (for a signing key)
 - HmacSHA512
 - HmacSHA256
 - HmacSHA384
 - HmacSHA224
- JCE Signature names (for a signing key)
 - SHA512withRSA
 - SHA256withRSA
 - SHA384withRSA
 - SHA224withRSA
- JCE Cipher names (for an encryption key)
 - RSA
 - AES
 - AESWrap

Parameters

- **key** (*bytes*) – Raw key bytes
- **algorithm** (*str*) – JCE Standard Algorithm Name
- **key_type** (`EncryptionKeyType`) – Identifies what type of key is being provided
- **key_encoding** (`KeyEncodingType`) – Identifies how the provided key is encoded

algorithm

Text description of algorithm used by this delegated key.

classmethod generate (*algorithm*, *key_length=None*)

Generate an instance of this `DelegatedKey` using the specified algorithm and key length.

Parameters

- **algorithm** (*str*) – Text description of algorithm to be used

- **key_length** (*int*) – Size in bits of key to generate

Returns Generated delegated key

Return type DelegatedKey

allowed_for_raw_materials

Only *JceNameLocalDelegatedKey* backed by AES keys are allowed to be used with *RawDecryptionMaterials* or *RawEncryptionMaterials*.

Returns decision

Return type bool

<code>dynamodb_encryption_sdk.delegated_keys</code>	Delegated keys.
<code>dynamodb_encryption_sdk.delegated_keys.jce</code>	Delegated key that uses JCE StandardName values to determine behavior.

3.8 Tools

3.8.1 Identifiers

Unique identifiers used by the DynamoDB Encryption Client.

class `dynamodb_encryption_sdk.identifiers.CryptoAction`

Bases: `enum.Enum`

Possible actions to take on an item attribute.

class `dynamodb_encryption_sdk.identifiers.EncryptionKeyType`

Bases: `enum.Enum`

Supported types of encryption keys.

class `dynamodb_encryption_sdk.identifiers.KeyEncodingType`

Bases: `enum.Enum`

Supported key encoding schemes.

3.8.2 Structures

Common structures used by the DynamoDB Encryption Client.

class `dynamodb_encryption_sdk.structures.EncryptionContext` (*table_name=None, partition_key_name=None, sort_key_name=None, attributes=None, material_description=None*)

Bases: `object`

Additional information about an encryption request.

Parameters

- **table_name** (*str*) – Table name

- **partition_key_name** (*str*) – Name of primary index partition attribute
- **sort_key_name** (*str*) – Name of primary index sort attribute
- **attributes** (*dict*) – Plaintext item attributes as a DynamoDB JSON dictionary
- **material_description** (*dict*) – Material description to use with this request

```
class dynamodb_encryption_sdk.structures.AttributeActions (default_action=<CryptoAction.ENCRYPT_AND_SIGN>, attribute_actions=None)
```

Bases: `object`

Configuration resource used to determine what action should be taken for a specific attribute.

Parameters

- **default_action** (`CryptoAction`) – Action to take if no specific action is defined in `attribute_actions`
- **attribute_actions** (*dict*) – Dictionary mapping attribute names to specific actions

action (*attribute_name*)

Determine the correct `CryptoAction` to apply to a supplied attribute based on this config.

Parameters **attribute_name** (*str*) – Attribute for which to determine action

copy ()

Return a new copy of this object.

set_index_keys (**keys*)

Set the appropriate action for the specified indexed attribute names.

Warning: If you have already set a custom action for any of these attributes, this will raise an error.

```
Default Action  -> Index Key Action
DO_NOTHING      -> DO_NOTHING
SIGN_ONLY       -> SIGN_ONLY
ENCRYPT_AND_SIGN -> SIGN_ONLY
```

Parameters ***keys** (*str*) – Attribute names to treat as indexed

Raises `InvalidArgumentError` – if a custom action was previously set for any specified attributes

contains_action (*action*)

Determine if the specified action is a possible action from this configuration.

Parameters **action** (`CryptoAction`) – Action to look for

```
class dynamodb_encryption_sdk.structures.TableIndex (partition, sort=None)
```

Bases: `object`

Describes a table index.

Parameters

- **partition** (*str*) – Name of the partition attribute
- **sort** (*str*) – Name of the sort attribute (optional)

classmethod `from_key_schema` (*key_schema*)
Build a `TableIndex` from the key schema returned by `DescribeTable`.

```
[
  {
    "KeyType": "HASH"|"RANGE",
    "AttributeName": ""
  },
]
```

Parameters `key_schema` (*list*) – `KeySchema` from `DescribeTable` response

Returns New `TableIndex` that describes the provided schema

Return type `TableIndex`

class `dynamodb_encryption_sdk.structures.TableInfo` (*name*, *primary_index=None*, *secondary_indexes=None*)

Bases: `object`

Describes a DynamoDB table.

Parameters

- **name** (*str*) – Table name
- **all_encrypting_secondary_indexes** (*bool*) – Should we allow secondary index attributes to be encrypted?
- **primary_index** (`TableIndex`) – Description of primary index
- **secondary_indexes** (*list* (`TableIndex`)) – Set of `TableIndex` objects describing any secondary indexes

`primary_index`

Return the primary `TableIndex`.

Returns primary index description

Return type `TableIndex`

Raises `AttributeError` – if primary index is unknown

`secondary_indexes`

Return the primary `TableIndex`.

Returns secondary index descriptions

Return type `TableIndex`

Raises `AttributeError` – if secondary indexes are unknown

`protected_index_keys` ()

Provide a set containing the names of all indexed attributes that must not be encrypted.

`encryption_context_values`

Build parameters needed to inform an `EncryptionContext` constructor about this table.

Return type `dict`

`refresh_indexed_attributes` (*client*)

Use the provided boto3 DynamoDB client to determine all indexes for this table.

Parameters `client` (`botocore.client.BaseClient`) – Pre-configured boto3 DynamoDB client

3.8.3 Transformers

Helper tools for translating between native and DynamoDB items.

`dynamodb_encryption_sdk.transform.dict_to_ddb(item)`

Converts a native Python dictionary to a raw DynamoDB item.

Parameters `item(dict)` – Native item

Returns DynamoDB item

Return type `dict`

`dynamodb_encryption_sdk.transform.ddb_to_dict(item)`

Converts a raw DynamoDB item to a native Python dictionary.

Parameters `item(dict)` – DynamoDB item

Returns Native item

Return type `dict`

<code>dynamodb_encryption_sdk.identifiers</code>	Unique identifiers used by the DynamoDB Encryption Client.
<code>dynamodb_encryption_sdk.structures</code>	Common structures used by the DynamoDB Encryption Client.
<code>dynamodb_encryption_sdk.transform</code>	Helper tools for translating between native and DynamoDB items.

3.9 Exceptions

Exception classed for use in the DynamoDB Encryption Client.

exception `dynamodb_encryption_sdk.exceptions.DynamodbEncryptionSdkError`

Bases: `Exception`

Base class for all custom exceptions.

exception `dynamodb_encryption_sdk.exceptions.InvalidArgumentError`

Bases: `dynamodb_encryption_sdk.exceptions.DynamodbEncryptionSdkError`

Raised when a general invalid argument is provided.

exception `dynamodb_encryption_sdk.exceptions.SerializationError`

Bases: `dynamodb_encryption_sdk.exceptions.DynamodbEncryptionSdkError`

Otherwise undifferentiated errors encountered while serializing data.

exception `dynamodb_encryption_sdk.exceptions.DeserializationError`

Bases: `dynamodb_encryption_sdk.exceptions.DynamodbEncryptionSdkError`

Otherwise undifferentiated errors encountered while deserializing data.

exception `dynamodb_encryption_sdk.exceptions.InvalidMaterialDescriptionError`

Bases: `dynamodb_encryption_sdk.exceptions.DeserializationError`

Raised when errors are encountered processing a material description.

exception `dynamodb_encryption_sdk.exceptions.InvalidMaterialDescriptionVersionError`

Bases: `dynamodb_encryption_sdk.exceptions.DeserializationError`

Raised when a material description is encountered with an invalid version.

exception `dynamodb_encryption_sdk.exceptions.InvalidAlgorithmError`
Bases: `dynamodb_encryption_sdk.exceptions.InvalidArgumentError`

Raised when an invalid algorithm identifier is encountered.

exception `dynamodb_encryption_sdk.exceptions.JceTransformationError`
Bases: `dynamodb_encryption_sdk.exceptions.DynamodbEncryptionSdkError`

Otherwise undifferentiated errors encountered when attempting to read a JCE transformation.

exception `dynamodb_encryption_sdk.exceptions.DelegatedKeyError`
Bases: `dynamodb_encryption_sdk.exceptions.DynamodbEncryptionSdkError`

Otherwise undifferentiated errors encountered by a DelegatedKey.

exception `dynamodb_encryption_sdk.exceptions.DelegatedKeyEncryptionError`
Bases: `dynamodb_encryption_sdk.exceptions.DelegatedKeyError`

Raised when a DelegatedKey encounters an error during encryption.

exception `dynamodb_encryption_sdk.exceptions.DelegatedKeyDecryptionError`
Bases: `dynamodb_encryption_sdk.exceptions.DelegatedKeyError`

Raised when a DelegatedKey encounters an error during decryption.

exception `dynamodb_encryption_sdk.exceptions.AwsKmsMaterialsProviderError`
Bases: `dynamodb_encryption_sdk.exceptions.DynamodbEncryptionSdkError`

Otherwise undifferentiated errors encountered by the AwsKmsCryptographicMaterialsProvider.

exception `dynamodb_encryption_sdk.exceptions.UnknownRegionError`
Bases: `dynamodb_encryption_sdk.exceptions.AwsKmsMaterialsProviderError`

Raised when the AwsKmsCryptographicMaterialsProvider is asked for an unknown region.

exception `dynamodb_encryption_sdk.exceptions.DecryptionError`
Bases: `dynamodb_encryption_sdk.exceptions.DynamodbEncryptionSdkError`

Otherwise undifferentiated error encountered while decrypting data.

exception `dynamodb_encryption_sdk.exceptions.UnwrappingError`
Bases: `dynamodb_encryption_sdk.exceptions.DynamodbEncryptionSdkError`

Otherwise undifferentiated error encountered while unwrapping a key.

exception `dynamodb_encryption_sdk.exceptions.EncryptionError`
Bases: `dynamodb_encryption_sdk.exceptions.DynamodbEncryptionSdkError`

Otherwise undifferentiated error encountered while encrypting data.

exception `dynamodb_encryption_sdk.exceptions.WrappingError`
Bases: `dynamodb_encryption_sdk.exceptions.DynamodbEncryptionSdkError`

Otherwise undifferentiated error encountered while wrapping a key.

exception `dynamodb_encryption_sdk.exceptions.SigningError`
Bases: `dynamodb_encryption_sdk.exceptions.DynamodbEncryptionSdkError`

Otherwise undifferentiated error encountered while signing data.

exception `dynamodb_encryption_sdk.exceptions.SignatureVerificationError`
Bases: `dynamodb_encryption_sdk.exceptions.DynamodbEncryptionSdkError`

Otherwise undifferentiated error encountered while verifying a signature.

- exception** `dynamodb_encryption_sdk.exceptions.ProviderStoreError`
 Bases: `dynamodb_encryption_sdk.exceptions.DynamodbEncryptionSdkError`
 Otherwise undifferentiated error encountered by a provider store.
- exception** `dynamodb_encryption_sdk.exceptions.NoKnownVersionError`
 Bases: `dynamodb_encryption_sdk.exceptions.ProviderStoreError`
 Raised if a provider store cannot locate any version of the requested material.
- exception** `dynamodb_encryption_sdk.exceptions.InvalidVersionError`
 Bases: `dynamodb_encryption_sdk.exceptions.ProviderStoreError`
 Raised if an invalid version of a material is requested.
- exception** `dynamodb_encryption_sdk.exceptions.VersionAlreadyExistsError`
 Bases: `dynamodb_encryption_sdk.exceptions.ProviderStoreError`
 Raised if a version that is being added to a provider store already exists.

3.10 Internal Resources

Warning: These are provided for informational purposes only. No guarantee is provided on the modules and APIs described here remaining consistent. Directly reference at your own risk.

<code>dynamodb_encryption_sdk.internal.identifiers</code>	Unique identifiers for internal use only.
<code>dynamodb_encryption_sdk.internal.str_ops</code>	Helper functions for consistently obtaining str and bytes objects in both Python2 and Python3.
<code>dynamodb_encryption_sdk.internal.utils</code>	Otherwise undifferentiated utility resources.
<code>dynamodb_encryption_sdk.internal.validators</code>	Custom validators for attrs.
<code>dynamodb_encryption_sdk.internal.crypto</code>	Inner cryptographic components.
<code>dynamodb_encryption_sdk.internal.crypto.authentication</code>	Functions to handle calculating and verifying signatures of encrypted items.
<code>dynamodb_encryption_sdk.internal.crypto.encryption</code>	Functions to handle encrypting and decrypting DynamoDB attributes.
<code>dynamodb_encryption_sdk.internal.crypto.jce_bridge</code>	Components to provide cryptographic primitives based on JCE Standard Names.
<code>dynamodb_encryption_sdk.internal.crypto.jce_bridge.authentication</code>	Cryptographic authentication resources for JCE bridge.
<code>dynamodb_encryption_sdk.internal.crypto.jce_bridge.encryption</code>	Cipher resource for JCE bridge.
<code>dynamodb_encryption_sdk.internal.crypto.jce_bridge.primitives</code>	Cryptographic primitive resources for JCE bridge.
<code>dynamodb_encryption_sdk.internal.formatting</code>	Data formatting utilities for the DynamoDB Encryption Client.
<code>dynamodb_encryption_sdk.internal.formatting.material_description</code>	Tools for serializing and deserializing material descriptions.

Continued on next page

Table 7 – continued from previous page

<code>dynamodb_encryption_sdk.internal.formatting.deserialize</code>	Helper functions for deserializing values.
<code>dynamodb_encryption_sdk.internal.formatting.deserialize.attribute</code>	Tooling for deserializing attributes.
<code>dynamodb_encryption_sdk.internal.formatting.serialize</code>	Helper functions for serializing values.
<code>dynamodb_encryption_sdk.internal.formatting.serialize.attribute</code>	Tooling for serializing attributes.

3.10.1 dynamodb_encryption_sdk.internal.identifiers

Unique identifiers for internal use only.

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

Classes

<code>MaterialDescriptionKeys</code>	Static keys for use when building and reading material descriptions.
<code>MaterialDescriptionValues</code>	Static default values for use when building material descriptions.
<code>MinimumKeySizes</code>	Minimum safe key sizes for algorithms.
<code>ReservedAttributes</code>	Item attributes reserved for use by DynamoDBEncryptionClient
<code>SignatureValues(raw, sha256)</code>	Values used when building the string to sign.
<code>Tag(tag, dynamodb_tag[, element_tag])</code>	Attribute data type identifiers used for serialization and deserialization of attributes.
<code>TagValues</code>	Static values to use when serializing attribute values.

`dynamodb_encryption_sdk.internal.identifiers.TEXT_ENCODING = 'utf-8'`
 Encoding to use for all text values. This is noted here for consistency but should not be changed.

class `dynamodb_encryption_sdk.internal.identifiers.MinimumKeySizes`
 Bases: `enum.Enum`

Minimum safe key sizes for algorithms.

class `dynamodb_encryption_sdk.internal.identifiers.ReservedAttributes`
 Bases: `enum.Enum`

Item attributes reserved for use by DynamoDBEncryptionClient

class `dynamodb_encryption_sdk.internal.identifiers.Tag(tag, dynamodb_tag, element_tag=None)`

Bases: `enum.Enum`

Attribute data type identifiers used for serialization and deserialization of attributes.

Sets up new Tag object.

Parameters

- **tag** (*bytes*) – DynamoDB Encryption SDK tag

- **dynamodb_tag** (*str*) – DynamoDB tag
- **element_tag** (*bytes*) – The type of tag contained within attributes of this type

class dynamodb_encryption_sdk.internal.identifiers.**TagValues**

Bases: `enum.Enum`

Static values to use when serializing attribute values.

class dynamodb_encryption_sdk.internal.identifiers.**SignatureValues** (*raw*,
sha256)

Bases: `enum.Enum`

Values used when building the string to sign.

Note: The only time we actually use these values, we use the SHA256 hash of the value, so we pre-compute these hashes here.

Set up a new *SignatureValues* object.

Parameters

- **raw** (*bytes*) – Raw value
- **sha256** (*bytes*) – SHA256 hash of raw value

class dynamodb_encryption_sdk.internal.identifiers.**MaterialDescriptionKeys**

Bases: `enum.Enum`

Static keys for use when building and reading material descriptions.

class dynamodb_encryption_sdk.internal.identifiers.**MaterialDescriptionValues**

Bases: `enum.Enum`

Static default values for use when building material descriptions.

3.10.2 dynamodb_encryption_sdk.internal.str_ops

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

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

Functions

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

`dynamodb_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`

`dynamodb_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`

3.10.3 dynamodb_encryption_sdk.internal.utils

Otherwise undifferentiated utility resources.

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

Functions

<code>crypto_config_from_cache(materials_provider, ...)</code>	Build a crypto config from the provided values, loading the table info from the provided cache.
<code>crypto_config_from_kwargs(fallback, **kwargs)</code>	Pull all encryption-specific parameters from the request and use them to build a crypto config.
<code>crypto_config_from_table_info(...)</code>	Build a crypto config from the provided values and table info.
<code>decrypt_batch_get_item(decrypt_method, ...)</code>	Transparently decrypt multiple items after getting them in a batch request.
<code>decrypt_get_item(decrypt_method, ...)</code>	Transparently decrypt an item after getting it from the table.
<code>decrypt_list_of_items(crypto_config, ...)</code>	Iterate through a list of encrypted items, decrypting each item and yielding the plaintext item.
<code>decrypt_multi_get(decrypt_method, ...)</code>	Transparently decrypt multiple items after getting them from the table with a scan or query method.
<code>encrypt_batch_write_item(encrypt_method, ...)</code>	Transparently encrypt multiple items before putting them in a batch request.
<code>encrypt_put_item(encrypt_method, ...)</code>	Transparently encrypt an item before putting it to the table.
<code>validate_get_arguments(kwargs)</code>	Verify that attribute filtering parameters are not found in the request.

Classes

<code>TableInfoCache(client, ...)</code>	Very simple cache of <code>TableInfo</code> objects, providing configuration information about DynamoDB tables.
--	---

class `dynamodb_encryption_sdk.internal.utils.TableInfoCache` (*client*, *auto_refresh_table_indexes*)

Bases: `object`

Very simple cache of `TableInfo` objects, providing configuration information about DynamoDB tables.

Parameters

- **client** (*botocore.client.BaseClient*) – Boto3 DynamoDB client
- **auto_refresh_table_indexes** (*bool*) – Should we attempt to refresh information about table indexes? Requires `dynamodb:DescribeTable` permissions on each table.

table_info (*table_name*)

Collect a `TableInfo` object for the specified table, creating and adding it to the cache if not already present.

Parameters **table_name** (*str*) – Name of table

Returns `TableInfo` describing the requested table

Return type *TableInfo*

`dynamodb_encryption_sdk.internal.utils.validate_get_arguments` (*kwargs*)

Verify that attribute filtering parameters are not found in the request.

Raises *InvalidArgumentError* – if banned parameters are found

`dynamodb_encryption_sdk.internal.utils.crypto_config_from_kwargs` (*fallback*,
***kwargs*)

Pull all encryption-specific parameters from the request and use them to build a crypto config.

Returns crypto config and updated kwargs

Return type `dynamodb_encryption_sdk.encrypted.CryptoConfig` and dict

`dynamodb_encryption_sdk.internal.utils.crypto_config_from_table_info` (*materials_provider*,
at-tribute_actions,
table_info)

Build a crypto config from the provided values and table info.

Returns crypto config and updated kwargs

Return type *tuple(CryptoConfig, dict)*

`dynamodb_encryption_sdk.internal.utils.crypto_config_from_cache` (*materials_provider*,
at-tribute_actions,
table_info_cache,
table_name)

Build a crypto config from the provided values, loading the table info from the provided cache.

Returns crypto config and updated kwargs

Return type *tuple(CryptoConfig, dict)*

`dynamodb_encryption_sdk.internal.utils.decrypt_list_of_items` (*crypto_config*,
decrypt_method,
items)

Iterate through a list of encrypted items, decrypting each item and yielding the plaintext item.

Parameters

- **crypto_config** (*CryptoConfig*) – `CryptoConfig` to use
- **decrypt_method** (*callable*) – Method to use to decrypt items
- **items** – Iterable of encrypted items

Returns Iterable of plaintext items

`dynamodb_encryption_sdk.internal.utils.decrypt_multi_get` (*decrypt_method*,
crypto_config_method,
read_method, ****kwargs**)

Transparently decrypt multiple items after getting them from the table with a scan or query method.

Parameters

- **decrypt_method** (*callable*) – Method to use to decrypt items
- **crypto_config_method** (*callable*) – Method that accepts kwargs and provides a `CryptoConfig`
- **read_method** (*callable*) – Method that reads from the table
- ****kwargs** – Keyword arguments to pass to `read_method`

Returns DynamoDB response

Return type `dict`

`dynamodb_encryption_sdk.internal.utils.decrypt_get_item` (*decrypt_method*,
crypto_config_method,
read_method, ****kwargs**)

Transparently decrypt an item after getting it from the table.

Parameters

- **decrypt_method** (*callable*) – Method to use to decrypt item
- **crypto_config_method** (*callable*) – Method that accepts kwargs and provides a `CryptoConfig`
- **read_method** (*callable*) – Method that reads from the table
- ****kwargs** – Keyword arguments to pass to `read_method`

Returns DynamoDB response

Return type `dict`

`dynamodb_encryption_sdk.internal.utils.decrypt_batch_get_item` (*decrypt_method*,
crypto_config_method,
read_method,
****kwargs**)

Transparently decrypt multiple items after getting them in a batch request.

Parameters

- **decrypt_method** (*callable*) – Method to use to decrypt items
- **crypto_config_method** (*callable*) – Method that accepts kwargs and provides a `CryptoConfig`
- **read_method** (*callable*) – Method that reads from the table
- ****kwargs** – Keyword arguments to pass to `read_method`

Returns DynamoDB response

Return type `dict`

`dynamodb_encryption_sdk.internal.utils.encrypt_put_item` (*encrypt_method*,
crypto_config_method,
write_method, ****kwargs**)

Transparently encrypt an item before putting it to the table.

Parameters

- **encrypt_method** (*callable*) – Method to use to encrypt items
- **crypto_config_method** (*callable*) – Method that accepts kwargs and provides a `CryptoConfig`
- **write_method** (*callable*) – Method that writes to the table
- ****kwargs** – Keyword arguments to pass to `write_method`

Returns DynamoDB response

Return type `dict`

`dynamodb_encryption_sdk.internal.utils.encrypt_batch_write_item` (*encrypt_method*,
crypto_config_method,
write_method,
***kwargs*)

Transparently encrypt multiple items before putting them in a batch request.

Parameters

- **encrypt_method** (*callable*) – Method to use to encrypt items
- **crypto_config_method** (*callable*) – Method that accepts a table name string and provides a `CryptoConfig`
- **write_method** (*callable*) – Method that writes to the table
- ****kwargs** – Keyword arguments to pass to `write_method`

Returns DynamoDB response

Return type `dict`

3.10.4 dynamodb_encryption_sdk.internal.validators

Custom validators for `attrs`.

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

Functions

<code>callable_validator</code> (<i>instance</i> , <i>attribute</i> , <i>value</i>)	Validate that an attribute value is callable.
<code>dictionary_validator</code> (<i>key_type</i> , <i>value_type</i>)	Validator for <code>attrs</code> that performs deep type checking of dictionaries.
<code>iterable_validator</code> (<i>iterable_type</i> , <i>member_type</i>)	Validator for <code>attrs</code> that performs deep type checking of iterables.

`dynamodb_encryption_sdk.internal.validators.dictionary_validator` (*key_type*,
value_type)
Validator for `attrs` that performs deep type checking of dictionaries.

`dynamodb_encryption_sdk.internal.validators.iterable_validator` (*iterable_type*,
member_type)
Validator for `attrs` that performs deep type checking of iterables.

3.10.5 dynamodb_encryption_sdk.internal.crypto

Inner cryptographic components.

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

3.10.6 dynamodb_encryption_sdk.internal.crypto.authentication

Functions to handle calculating and verifying signatures of encrypted items.

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

Functions

<code>sign_item(encrypted_item, signing_key, ...)</code>	Generate the signature DynamoDB attribute.
<code>verify_item_signature(signature_attribute, ...)</code>	Verify the item signature.

`dynamodb_encryption_sdk.internal.crypto.authentication.sign_item(encrypted_item, signing_key, crypto_config)`

Generate the signature DynamoDB attribute.

Parameters

- **encrypted_item** (*dict*) – Encrypted DynamoDB item
- **signing_key** (*DelegatedKey*) – DelegatedKey to use to calculate the signature
- **crypto_config** (*CryptoConfig*) – Cryptographic configuration

Returns Item signature DynamoDB attribute value

Return type `dict`

`dynamodb_encryption_sdk.internal.crypto.authentication.verify_item_signature(signature_attribute, encrypted_item, verification_key, crypto_config)`

Verify the item signature.

Parameters

- **signature_attribute** (*dict*) – Item signature DynamoDB attribute value
- **encrypted_item** (*dict*) – Encrypted DynamoDB item

- **verification_key** (*DelegatedKey*) – DelegatedKey to use to calculate the signature
- **crypto_config** (*CryptoConfig*) – Cryptographic configuration

3.10.7 dynamodb_encryption_sdk.internal.crypto.encryption

Functions to handle encrypting and decrypting DynamoDB attributes.

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

Functions

decrypt_attribute(attribute_name, attribute, Decrypt a single DynamoDB attribute.
...)

encrypt_attribute(attribute_name, attribute, Encrypt a single DynamoDB attribute.
...)

dynamodb_encryption_sdk.internal.crypto.encryption.**encrypt_attribute** (*attribute_name*,
at-tribute,
encryp-tion_key,
algo-rithm)

Encrypt a single DynamoDB attribute.

Parameters

- **attribute_name** (*str*) – DynamoDB attribute name
- **attribute** (*dict*) – Plaintext DynamoDB attribute
- **encryption_key** (*DelegatedKey*) – DelegatedKey to use to encrypt the attribute
- **algorithm** (*str*) – Encryption algorithm descriptor (passed to encryption_key as algorithm)

Returns Encrypted DynamoDB binary attribute

Return type `dict`

dynamodb_encryption_sdk.internal.crypto.encryption.**decrypt_attribute** (*attribute_name*,
at-tribute,
decryp-tion_key,
algo-rithm)

Decrypt a single DynamoDB attribute.

Parameters

- **attribute_name** (*str*) – DynamoDB attribute name

- **attribute** (*dict*) – Encrypted DynamoDB attribute
- **encryption_key** (*DelegatedKey*) – DelegatedKey to use to encrypt the attribute
- **algorithm** (*str*) – Decryption algorithm descriptor (passed to encryption_key as algorithm)

Returns Plaintext DynamoDB attribute

Return type `dict`

3.10.8 dynamodb_encryption_sdk.internal.crypto.jce_bridge

Components to provide cryptographic primitives based on JCE Standard Names. <https://docs.oracle.com/javase/8/docs/technotes/guides/security/StandardNames.html>

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

3.10.9 dynamodb_encryption_sdk.internal.crypto.jce_bridge.authentication

Cryptographic authentication resources for JCE bridge.

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

Classes

<i>JavaAuthenticator</i>	Parent class for all Java bridges that provide authentication characteristics.
<i>JavaMac</i> (java_name, algorithm_type, hash_type)	Symmetric MAC authenticators.
<i>JavaSignature</i> (java_name, algorithm_type, ...)	Asymmetric signature authenticators.

class dynamodb_encryption_sdk.internal.crypto.jce_bridge.authentication.**JavaAuthenticator**

Bases: `object`

Parent class for all Java bridges that provide authentication characteristics.

load_key (*key, key_type, key_encoding*)

Load a key from bytes.

Parameters

- **key** (*bytes*) – Raw key bytes to load
- **key_type** (`EncryptionKeyType`) – Type of key to load
- **key_encoding** (`KeyEncodingType`) – Encoding used to serialize key

Returns Loaded key

Return type `bytes`

validate_algorithm (*algorithm*)

Determine whether the requested algorithm name is compatible with this authenticator.

Parameters **algorithm** (*str*) – Algorithm name

Raises *InvalidAlgorithmError* – if specified algorithm name is not compatible with this authenticator

sign (*key, data*)

Sign data using loaded key.

Parameters

- **key** – Loaded key
- **data** (*bytes*) – Data to sign

Returns Calculated signature

Return type *bytes*

Raises *SigningError* – if unable to sign data with key

verify (*key, signature, data*)

Verify signature over data using key.

Parameters

- **key** – Loaded key
- **signature** (*bytes*) – Signature to verify
- **data** (*bytes*) – Data over which to verify signature

Raises *SignatureVerificationError* – if unable to verify signature

class `dynamodb_encryption_sdk.internal.crypto.jce_bridge.authentication.JavaMac` (*java_name, algorithm_type, hash_type*)

Bases: `dynamodb_encryption_sdk.internal.crypto.jce_bridge.authentication.JavaAuthenticator`

Symmetric MAC authenticators.

<https://docs.oracle.com/javase/8/docs/api/javax/crypto/Mac.html>

<https://docs.oracle.com/javase/8/docs/technotes/guides/security/StandardNames.html#Mac>

load_key (*key, key_type, key_encoding*)

Load a raw key from bytes.

Parameters

- **key** (*bytes*) – Raw key bytes to load
- **key_type** (*EncryptionKeyType*) – Type of key to load
- **key_encoding** (*KeyEncodingType*) – Encoding used to serialize key

Returns Loaded key

Return type *bytes*

Raises *ValueError* – if *key_type* is not symmetric or *key_encoding* is not raw

validate_algorithm (*algorithm*)

Determine whether the requested algorithm name is compatible with this authenticator.

Parameters `algorithm` (*str*) – Algorithm name

Raises `InvalidAlgorithmError` – if specified algorithm name is not compatible with this authenticator

sign (*key*, *data*)

Sign data using loaded key.

Parameters

- **key** (*bytes*) – Loaded key
- **data** (*bytes*) – Data to sign

Returns Calculated signature

Return type `bytes`

Raises `SigningError` – if unable to sign data with key

verify (*key*, *signature*, *data*)

Verify signature over data using key.

Parameters

- **key** (*bytes*) – Loaded key
- **signature** (*bytes*) – Signature to verify
- **data** (*bytes*) – Data over which to verify signature

Raises `SignatureVerificationError` – if unable to verify signature

class `dynamodb_encryption_sdk.internal.crypto.jce_bridge.authentication.JavaSignature` (*java_n*
al-
go-
rithm_
hash_t
padding

Bases: `dynamodb_encryption_sdk.internal.crypto.jce_bridge.authentication.JavaAuthenticator`

Asymmetric signature authenticators.

<https://docs.oracle.com/javase/8/docs/api/java/security/Signature.html> <https://docs.oracle.com/javase/8/docs/technotes/guides/security/StandardNames.html#Signature>

validate_algorithm (*algorithm*)

Determine whether the requested algorithm name is compatible with this authenticator.

Parameters `algorithm` (*str*) – Algorithm name

Raises `InvalidAlgorithmError` – if specified algorithm name is not compatible with this authenticator

load_key (*key*, *key_type*, *key_encoding*)

Load a key object from the provided raw key bytes.

Parameters

- **key** (*bytes*) – Raw key bytes to load
- **key_type** (`EncryptionKeyType`) – Type of key to load
- **key_encoding** (`KeyEncodingType`) – Encoding used to serialize key

Returns Loaded key

Raises `ValueError` – if `key_type` and `key_encoding` are not a valid pairing

sign (*key, data*)

Sign data using loaded key.

Parameters

- **key** – Loaded key
- **data** (*bytes*) – Data to sign

Returns Calculated signature

Return type `bytes`

Raises `SigningError` – if unable to sign data with key

verify (*key, signature, data*)

Verify signature over data using key.

Parameters

- **key** – Loaded key
- **signature** (*bytes*) – Signature to verify
- **data** (*bytes*) – Data over which to verify signature

Raises `SignatureVerificationError` – if unable to verify signature

3.10.10 dynamodb_encryption_sdk.internal.crypto.jce_bridge.encryption

Cipher resource for JCE bridge.

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

Classes

<i>JavaCipher</i> (cipher, mode, padding)	Defines the encryption cipher, mode, and padding type to use for encryption.
---	--

class dynamodb_encryption_sdk.internal.crypto.jce_bridge.encryption.**JavaCipher** (*cipher, mode, padding*)

Bases: `object`

Defines the encryption cipher, mode, and padding type to use for encryption.

<https://docs.oracle.com/javase/8/docs/api/javax/crypto/Cipher.html>

Parameters

- **cipher** (`JavaEncryptionAlgorithm`) – Encryption algorithm to use
- **mode** (`JavaMode`) – Encryption mode to use
- **padding** (`JavaPadding`) – Encryption padding to use

encrypt (*key, data*)

Encrypt data using loaded key.

Parameters

- **key** – Key loaded by cipher
- **data** (*bytes*) – Data to encrypt

Returns Encrypted data

Return type *bytes*

decrypt (*key, data*)

Decrypt data using loaded key.

Parameters

- **key** – Key loaded by cipher
- **data** (*bytes*) – Data to decrypt

Returns Decrypted data

Return type *bytes*

wrap (*wrapping_key, key_to_wrap*)

Wrap key using loaded key.

Parameters

- **wrapping_key** – Key loaded by cipher
- **key_to_wrap** (*bytes*) – Key to wrap

Returns Wrapped key

Return type *bytes*

unwrap (*wrapping_key, wrapped_key*)

Wrap key using loaded key.

Parameters

- **wrapping_key** – Key loaded by cipher
- **wrapped_key** (*bytes*) – Wrapped key

Returns Unwrapped key

Return type *bytes*

transformation

Returns the Java transformation describing this `JavaCipher`. <https://docs.oracle.com/javase/8/docs/api/javax/crypto/Cipher.html> <https://docs.oracle.com/javase/8/docs/technotes/guides/security/StandardNames.html#Cipher>

Returns Formatted transformation

Return type *str*

classmethod from_transformation (*cipher_transformation*)

Generates an `JavaCipher` object from the Java transformation. <https://docs.oracle.com/javase/8/docs/api/javax/crypto/Cipher.html> <https://docs.oracle.com/javase/8/docs/technotes/guides/security/StandardNames.html#Cipher>

Parameters **cipher_transformation** (*str*) – Formatted transformation

Returns JavaCipher instance

Return type *JavaCipher*

3.10.11 dynamodb_encryption_sdk.internal.crypto.jce_bridge.primitives

Cryptographic primitive resources for JCE bridge.

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

Functions

<code>load_rsa_key(key, key_type, key_encoding)</code>	Load an RSA key object from the provided raw key bytes.
--	---

Classes

<code>BlockSizePadding(java_name, padding)</code>	Padding types that require a block size input.
<code>JavaAsymmetricEncryptionAlgorithm(java_name, ...)</code>	JavaEncryptionAlgorithm for asymmetric algorithms.
<code>JavaEncryptionAlgorithm(java_name, cipher)</code>	Bridge the gap from the Java encryption algorithm names and Python resources.
<code>JavaMode(java_name, mode)</code>	Bridge the gap from the Java encryption mode names and Python resources.
<code>JavaPadding</code>	Bridge the gap from the Java padding names and Python resources.
<code>JavaSymmetricEncryptionAlgorithm(java_name, ...)</code>	JavaEncryptionAlgorithm for symmetric algorithms.
<code>OaepPadding(java_name, padding, digest, mgf, ...)</code>	OAEP padding types.
<code>SimplePadding(java_name, padding)</code>	Padding types that do not require any preparation.

class `dynamodb_encryption_sdk.internal.crypto.jce_bridge.primitives.JavaPadding`
 Bases: `object`

Bridge the gap from the Java padding names and Python resources. <https://docs.oracle.com/javase/8/docs/technotes/guides/security/StandardNames.html#Cipher>

build (*block_size*)

Build an instance of this padding type.

class `dynamodb_encryption_sdk.internal.crypto.jce_bridge.primitives.SimplePadding` (*java_name, padding*)

Bases: `dynamodb_encryption_sdk.internal.crypto.jce_bridge.primitives.JavaPadding`

Padding types that do not require any preparation.

build (*block_size=None*)

Build an instance of this padding type.

Parameters `block_size` (*int*) – Not used by SimplePadding. Ignored and not required.

Returns Padding instance

class dynamodb_encryption_sdk.internal.crypto.jce_bridge.primitives.**BlockSizePadding** (*java_name, padding*)

Bases: *dynamodb_encryption_sdk.internal.crypto.jce_bridge.primitives.JavaPadding*

Padding types that require a block size input.

build (*block_size*)

Build an instance of this padding type.

Parameters **block_size** (*int*) – Block size of algorithm for which to build padder.

Returns Padding instance

class dynamodb_encryption_sdk.internal.crypto.jce_bridge.primitives.**OaepPadding** (*java_name, padding, digest, mgf, mgf_digest*)

Bases: *dynamodb_encryption_sdk.internal.crypto.jce_bridge.primitives.JavaPadding*

OAEP padding types. These require more complex setup.

Warning: By default, Java incorrectly implements RSA OAEP for all hash functions besides SHA1. The same hashing algorithm should be used by both OAEP and the MGF, but by default Java always uses SHA1 for the MGF.

Because we need to match this behavior, all *OaepPadding* instances should be created with MGF1-SHA1.

build (*block_size=None*)

Build an instance of this padding type.

Parameters **block_size** (*int*) – Not used by OaepPadding. Ignored and not required.

Returns Padding instance

class dynamodb_encryption_sdk.internal.crypto.jce_bridge.primitives.**JavaMode** (*java_name, mode*)

Bases: *object*

Bridge the gap from the Java encryption mode names and Python resources. <https://docs.oracle.com/javase/8/docs/technotes/guides/security/StandardNames.html#Cipher>

build (*iv*)

Build an instance of this mode type.

Parameters **iv** (*bytes*) – Initialization vector bytes

Returns Mode instance

class dynamodb_encryption_sdk.internal.crypto.jce_bridge.primitives.**JavaEncryptionAlgorithm**

Bases: *object*

Bridge the gap from the Java encryption algorithm names and Python resources. <https://docs.oracle.com/javase/8/docs/technotes/guides/security/StandardNames.html#Cipher>

validate_algorithm (*algorithm*)

Determine whether the requested algorithm name is compatible with this cipher

class dynamodb_encryption_sdk.internal.crypto.jce_bridge.primitives.**JavaSymmetricEncryption**

Bases: `dynamodb_encryption_sdk.internal.crypto.jce_bridge.primitives.
JavaEncryptionAlgorithm`

JavaEncryptionAlgorithm for symmetric algorithms. <https://docs.oracle.com/javase/8/docs/technotes/guides/security/StandardNames.html#Cipher>

load_key (*key, key_type, key_encoding*)

Load a key from bytes.

Parameters

- **key** (*bytes*) – Key bytes
- **key_type** (`EncryptionKeyType`) – Type of key
- **key_encoding** (`KeyEncodingType`) – Encoding used to serialize key

Returns Loaded key

wrap (*wrapping_key, key_to_wrap*)

Wrap key using AES keywrap.

Parameters

- **wrapping_key** (*bytes*) – Loaded key with which to wrap
- **key_to_wrap** (*bytes*) – Raw key to wrap

Returns Wrapped key

Return type `bytes`

unwrap (*wrapping_key, wrapped_key*)

Unwrap key using AES keywrap.

Parameters

- **wrapping_key** (*bytes*) – Loaded key with which to unwrap
- **wrapped_key** (*bytes*) – Wrapped key to unwrap

Returns Unwrapped key

Return type `bytes`

encrypt (*key, data, mode, padding*)

Encrypt data using the supplied values.

Parameters

- **key** (*bytes*) – Loaded encryption key
- **data** (*bytes*) – Data to encrypt
- **mode** (`JavaMode`) – Encryption mode to use
- **padding** (`JavaPadding`) – Padding mode to use

Returns IV prepended to encrypted data

Return type `bytes`

decrypt (*key, data, mode, padding*)

Decrypt data using the supplied values.

Parameters

- **key** (*bytes*) – Loaded decryption key
- **data** (*bytes*) – IV prepended to encrypted data
- **mode** (*JavaMode*) – Decryption mode to use
- **padding** (*JavaPadding*) – Padding mode to use

Returns Decrypted data

Return type *bytes*

class `dynamodb_encryption_sdk.internal.crypto.jce_bridge.primitives.JavaAsymmetricEncryptionAlgorithm`

Bases: `dynamodb_encryption_sdk.internal.crypto.jce_bridge.primitives.JavaEncryptionAlgorithm`

JavaEncryptionAlgorithm for asymmetric algorithms.

<https://docs.oracle.com/javase/8/docs/technotes/guides/security/StandardNames.html#Cipher>

load_key (*key, key_type, key_encoding*)

Load a key from bytes.

Parameters

- **key** (*bytes*) – Key bytes
- **key_type** (*EncryptionKeyType*) – Type of key
- **key_encoding** (*KeyEncodingType*) – Encoding used to serialize key

Returns Loaded key

encrypt (*key, data, mode, padding*)

Encrypt data using the supplied values.

Parameters

- **key** (*bytes*) – Loaded encryption key
- **data** (*bytes*) – Data to encrypt
- **mode** (*JavaMode*) – Encryption mode to use (not used by `JavaAsymmetricEncryptionAlgorithm`)
- **padding** (*JavaPadding*) – Padding mode to use

Returns Encrypted data

Return type *bytes*

decrypt (*key, data, mode, padding*)

Decrypt data using the supplied values.

Parameters

- **key** (*bytes*) – Loaded decryption key
- **data** (*bytes*) – IV prepended to encrypted data
- **mode** (*JavaMode*) – Decryption mode to use (not used by `JavaAsymmetricEncryptionAlgorithm`)

- **padding** (`JavaPadding`) – Padding mode to use

Returns Decrypted data

Return type `bytes`

3.10.12 `dynamodb_encryption_sdk.internal.formatting`

Data formatting utilities for the DynamoDB Encryption Client.

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

3.10.13 `dynamodb_encryption_sdk.internal.formatting.material_description`

Tools for serializing and deserializing material descriptions.

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

Functions

<code>deserialize(serialized_material_description)</code>	Deserialize a serialized material description attribute into a material description dictionary.
<code>serialize(material_description)</code>	Serialize a material description dictionary into a DynamoDB attribute.

`dynamodb_encryption_sdk.internal.formatting.material_description.serialize(material_description)`
Serialize a material description dictionary into a DynamoDB attribute.

Parameters `material_description` (*dict*) – Material description dictionary

Returns Serialized material description as a DynamoDB binary attribute value

Return type `dict`

Raises `InvalidMaterialDescriptionError` – if invalid name or value found in material description

`dynamodb_encryption_sdk.internal.formatting.material_description.deserialize(serialized_material_description)`
Deserialize a serialized material description attribute into a material description dictionary.

Parameters `serialized_material_description` (*dict*) – DynamoDB attribute value containing serialized material description.

Returns Material description dictionary

Return type `dict`

Raises

- `InvalidMaterialDescriptionError` – if malformed version
- `InvalidMaterialDescriptionVersionError` – if unknown version is found

3.10.14 dynamodb_encryption_sdk.internal.formatting.deserialize

Helper functions for deserializing values.

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

Functions

<code>decode_byte(stream)</code>	Decode a single raw byte from a serialized stream (used for deserialize bool).
<code>decode_length(stream)</code>	Decode the length of a value from a serialized stream.
<code>decode_tag(stream)</code>	Decode a tag value from a serialized stream.
<code>decode_value(stream)</code>	Decode the contents of a value from a serialized stream.
<code>unpack_value(format_string, stream)</code>	Helper function to unpack struct data from a stream and update the signature verifier.

`dynamodb_encryption_sdk.internal.formatting.deserialize.unpack_value` (*format_string*, *stream*)

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

Returns Unpacked values

Return type `tuple`

`dynamodb_encryption_sdk.internal.formatting.deserialize.decode_length` (*stream*)

Decode the length of a value from a serialized stream.

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

Returns Decoded length

Return type `int`

`dynamodb_encryption_sdk.internal.formatting.deserialize.decode_value` (*stream*)

Decode the contents of a value from a serialized stream.

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

Returns Decoded value

Return type `bytes`

`dynamodb_encryption_sdk.internal.formatting.deserialize.decode_tag` (*stream*)

Decode a tag value from a serialized stream.

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

Returns Decoded tag

Return type `bytes`

3.10.15 dynamodb_encryption_sdk.internal.formatting.deserialize.attribute

Tooling for deserializing attributes.

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

Functions

<code><i>deserialize_attribute</i>(serialized_attribute)</code>	Deserializes serialized attributes for decryption.
---	--

`dynamodb_encryption_sdk.internal.formatting.deserialize.attribute.deserialize_attribute` (*serialized_attribute*)
Deserializes serialized attributes for decryption.

Parameters `serialized_attribute` (*bytes*) – Serialized attribute bytes

Returns Deserialized attribute

Return type `dict`

3.10.16 dynamodb_encryption_sdk.internal.formatting.serialize

Helper functions for serializing values.

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

Functions

<code><i>encode_length</i>(attribute)</code>	Encodes the length of the attribute as an unsigned int.
<code><i>encode_value</i>(value)</code>	Encodes the value in Length-Value format.

`dynamodb_encryption_sdk.internal.formatting.serialize.encode_length` (*attribute*)
Encodes the length of the attribute as an unsigned int.

Parameters `attribute` – Attribute with length value

Returns Encoded value

Return type `bytes`

`dynamodb_encryption_sdk.internal.formatting.serialize.encode_value` (*value*)
Encodes the value in Length-Value format.

Parameters `value` (`six.string_types` or `boto3.dynamodb_encryption_sdk.types.Binary`) – Value to encode

Returns Length-Value encoded value

Return type `bytes`

3.10.17 dynamodb_encryption_sdk.internal.formatting.serialize.attribute

Tooling for serializing attributes.

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

Functions

<code>serialize_attribute(attribute)</code>	Serializes a raw attribute to a byte string as defined for the DynamoDB Client-Side Encryption Standard.
---	--

`dynamodb_encryption_sdk.internal.formatting.serialize.attribute.serialize_attribute(attribute)`
Serializes a raw attribute to a byte string as defined for the DynamoDB Client-Side Encryption Standard.

Parameters `attribute` (*dict*) – Item attribute value

Returns Serialized attribute

Return type `bytes`

4.1 3.0.0 – 2021-07-15

4.1.1 Deprecation

The AWS DynamoDB Encryption Client for Python no longer supports Python 2 or Python 3.4 as of major version 3.x; only Python 3.5+ is supported. Customers using Python 2 or Python 3.4 can still use the 2.x line of the DynamoDB Encryption Client, which will continue to receive security updates for the next 12 months, in accordance with our [Support Policy](#).

4.2 2.1.0 – 2021-07-15

4.2.1 Deprecation Announcement

The AWS DynamoDB Encryption Client for Python is discontinuing support for Python 2. Future major versions of this library will drop support for Python 2 and begin to adopt changes that are known to break Python 2.

Support for Python 3.4 will be removed at the same time. Moving forward, we will support Python 3.5+.

Security updates will still be available for the DynamoDB Encryption Client 2.x line for the next 12 months, in accordance with our [Support Policy](#).

4.3 2.0.0 – 2021-02-04

4.3.1 Breaking Changes

Removes MostRecentProvider. MostRecentProvider is replaced by CachingMostRecentProvider as of 1.3.0.

4.4 1.3.0 – 2021-02-04

Adds the `CachingMostRecentProvider` and deprecates `MostRecentProvider`.

Time-based key reauthorization logic in `MostRecentProvider` did not reauthorize the use of the key after key usage permissions were changed at the key provider (for example AWS Key Management Service). This created the potential for keys to be used in the DynamoDB Encryption Client after permissions to do so were revoked.

`CachingMostRecentProvider` replaces `MostRecentProvider` and provides a cache entry TTL to reauthorize the key with the key provider.

`MostRecentProvider` is now deprecated, and is removed in 2.0.0. See <https://docs.aws.amazon.com/dynamodb-encryption-client/latest/devguide/most-recent-provider.html> for more details.

4.5 1.2.0 – 2019-10-10

4.5.1 Bugfixes

- Fix `AwsKmsCryptographicMaterialsProvider` regional clients override bug #124 **NOTE: It is possible that this is a breaking change for you, depending on how you are re-using any custom botocore sessions that you provide to `AwsKmsCryptographicMaterialsProvider`.**
- Remove `attributes` attribute from `EncryptionContext` `str` and `repr` values. #127

4.6 1.1.1 – 2019-08-29

4.6.1 Bugfixes

- Fix `EncryptedPaginator` to successfully decrypt when using `AwsKmsCryptographicMaterialsProvider` #118

4.7 1.1.0 – 2019-03-13

4.7.1 Features

- **Batch write operations via the high-level helper clients now return plaintext items in `UnprocessedItems`.** #107

4.8 1.0.7 – 2018-01-16

4.8.1 Bugfixes

- Fix `MostRecentProvider` cache reuse bug. #105

4.9 1.0.6 – 2018-01-15

4.9.1 Bugfixes

- Fix `MostRecentProvider` bug in providing invalid cached results. #102

4.10 1.0.5 – 2018-08-01

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

4.11 1.0.4 – 2018-05-22

4.11.1 Bugfixes

- Fix `MostRecentProvider` behavior when lock cannot be acquired. #72
- Fix `MostRecentProvider` lock acquisition for Python 2.7. #74
- Fix `TableInfo` secondary index storage. #75

4.12 1.0.3 – 2018-05-03

4.12.1 Bugfixes

- Finish fixing `MANIFEST.in`.

4.13 1.0.2 – 2018-05-03

4.13.1 Bugfixes

- Fill out `MANIFEST.in` to correctly include necessary files in source build.

4.14 1.0.1 – 2018-05-02

- Add version convenience import to base namespace.

4.15 1.0.0 – 2018-05-02

- Initial public release

Python Module Index

d

dynamodb_encryption_sdk.delegated_keys.jce, 23

dynamodb_encryption_sdk.encrypted, 7

dynamodb_encryption_sdk.encrypted.client, 11

dynamodb_encryption_sdk.encrypted.item, 12

dynamodb_encryption_sdk.encrypted.resource, 9

dynamodb_encryption_sdk.encrypted.table, 8

dynamodb_encryption_sdk.exceptions, 27

dynamodb_encryption_sdk.identifiers, 24

dynamodb_encryption_sdk.internal.crypto, 36

dynamodb_encryption_sdk.internal.crypto.authentication, 36

dynamodb_encryption_sdk.internal.crypto.encryption, 37

dynamodb_encryption_sdk.internal.crypto.jce_bridge, 38

dynamodb_encryption_sdk.internal.crypto.jce_bridge.authentication, 38

dynamodb_encryption_sdk.internal.crypto.jce_bridge.encryption, 41

dynamodb_encryption_sdk.internal.crypto.jce_bridge.primitives, 43

dynamodb_encryption_sdk.internal.formatting, 47

dynamodb_encryption_sdk.internal.formatting.deserialize, 48

dynamodb_encryption_sdk.internal.formatting.deserialize.attribute, 49

dynamodb_encryption_sdk.internal.formatting.material_description, 47

dynamodb_encryption_sdk.internal.formatting.serialize, 49

dynamodb_encryption_sdk.internal.formatting.serialize.attribute, 50

dynamodb_encryption_sdk.internal.identifiers, 30

dynamodb_encryption_sdk.internal.str_ops, 31

dynamodb_encryption_sdk.internal.utils, 32

dynamodb_encryption_sdk.internal.validators, 35

dynamodb_encryption_sdk.material_providers.aws_kms, 14

dynamodb_encryption_sdk.material_providers.most_recommended, 16

dynamodb_encryption_sdk.material_providers.static, 17

dynamodb_encryption_sdk.material_providers.store.most_recommended, 18

dynamodb_encryption_sdk.material_providers.wrapped, 15

dynamodb_encryption_sdk.materials.raw, 21

dynamodb_encryption_sdk.materials.wrapped, 20

dynamodb_encryption_sdk.structures, 24

dynamodb_encryption_sdk.transform, 27

A

action() (*dynamodb_encryption_sdk.structures.AttributeActions* class method), 25
 algorithm(*dynamodb_encryption_sdk.delegated_keys.jce.JceNameLocalDelegatedKey* attribute), 23
 allowed_for_raw_materials (*dynamodb_encryption_sdk.delegated_keys.jce.JceNameLocalDelegatedKey* attribute), 24
 AttributeActions (class in *dynamodb_encryption_sdk.structures*), 25
 AwsKmsCryptographicMaterialsProvider (class in *dynamodb_encryption_sdk.material_providers.aws_kms*), 14
 AwsKmsMaterialsProviderError, 28
 contains_action() (*dynamodb_encryption_sdk.structures.AttributeActions* method), 25
 copy() (*dynamodb_encryption_sdk.structures.AttributeActions* method), 25
 create_table() (*dynamodb_encryption_sdk.material_providers.store.meta.MetaStore* class method), 18
 crypto_config_from_cache() (in module *dynamodb_encryption_sdk.internal.utils*), 33
 crypto_config_from_kwargs() (in module *dynamodb_encryption_sdk.internal.utils*), 33
 crypto_config_from_table_info() (in module *dynamodb_encryption_sdk.internal.utils*), 33

B

batch_writer() (*dynamodb_encryption_sdk.encrypted.table.EncryptedTable* method), 9
 BlockSizePadding (class in *dynamodb_encryption_sdk.internal.crypto.jce_bridge.primitives*), 44
 build() (*dynamodb_encryption_sdk.internal.crypto.jce_bridge.primitives.BlockSizePadding* method), 44
 build() (*dynamodb_encryption_sdk.internal.crypto.jce_bridge.primitives.JavaMode* method), 44
 build() (*dynamodb_encryption_sdk.internal.crypto.jce_bridge.primitives.JavaPadding* method), 43
 build() (*dynamodb_encryption_sdk.internal.crypto.jce_bridge.primitives.OverPadding* method), 44
 build() (*dynamodb_encryption_sdk.internal.crypto.jce_bridge.primitives.SimplePadding* method), 43
 CryptoAction (class in *dynamodb_encryption_sdk.identifiers*), 24
 CryptoConfig (class in *dynamodb_encryption_sdk.encrypted*), 7
 decode_length() (in module *dynamodb_encryption_sdk.internal.formatting.deserialize*), 48
 decode_tag() (in module *dynamodb_encryption_sdk.internal.formatting.deserialize*), 48
 decode_value() (in module *dynamodb_encryption_sdk.internal.formatting.deserialize*), 48

C

CachingMostRecentProvider (class in *dynamodb_encryption_sdk.material_providers.most_recent*), 16
 decrypt() (*dynamodb_encryption_sdk.internal.crypto.jce_bridge.primitives* method), 42
 decrypt() (*dynamodb_encryption_sdk.internal.crypto.jce_bridge.primitives* method), 46
 decrypt() (*dynamodb_encryption_sdk.internal.crypto.jce_bridge.primitives* method), 45

decrypt_attribute()	(in module dynamodb_encryption_sdk.internal.crypto.encryption), 37	dynamodb_encryption_sdk.delegated_keys.jce (module), 23
decrypt_batch_get_item()	(in module dynamodb_encryption_sdk.internal.utils), 34	dynamodb_encryption_sdk.encrypted (module), 7
decrypt_dynamodb_item()	(in module dynamodb_encryption_sdk.encrypted.item), 13	dynamodb_encryption_sdk.encrypted.client (module), 11
decrypt_get_item()	(in module dynamodb_encryption_sdk.internal.utils), 34	dynamodb_encryption_sdk.encrypted.item (module), 12
decrypt_list_of_items()	(in module dynamodb_encryption_sdk.internal.utils), 33	dynamodb_encryption_sdk.encrypted.resource (module), 9
decrypt_multi_get()	(in module dynamodb_encryption_sdk.internal.utils), 33	dynamodb_encryption_sdk.encrypted.table (module), 8
decrypt_python_item()	(in module dynamodb_encryption_sdk.encrypted.item), 14	dynamodb_encryption_sdk.exceptions (module), 27
decryption_key	(dynamodb_encryption_sdk.materials.raw.RawDecryptionMaterials attribute), 22	dynamodb_encryption_sdk.identifiers (module), 24
decryption_key	(dynamodb_encryption_sdk.materials.wrapped.WrappedCryptographicMaterials attribute), 20	dynamodb_encryption_sdk.internal.crypto (module), 36
decryption_materials()	(dynamodb_encryption_sdk.encrypted.CryptoConfig method), 7	dynamodb_encryption_sdk.internal.crypto.authentication (module), 36
decryption_materials()	(dynamodb_encryption_sdk.material_providers.aws_kms.AwsKmsCryptographicMaterialsProvider method), 15	dynamodb_encryption_sdk.internal.crypto.encryption (module), 36
decryption_materials()	(dynamodb_encryption_sdk.material_providers.most_recent.CachingMostRecentProvider method), 17	dynamodb_encryption_sdk.internal.crypto.jce_bridge (module), 38
decryption_materials()	(dynamodb_encryption_sdk.material_providers.static.StaticCryptographicMaterialsProvider method), 18	dynamodb_encryption_sdk.internal.crypto.jce_bridge (module), 38
decryption_materials()	(dynamodb_encryption_sdk.material_providers.wrapped.WrappedCryptographicMaterialsProvider method), 16	dynamodb_encryption_sdk.internal.crypto.jce_bridge (module), 43
DecryptionError, 28		dynamodb_encryption_sdk.internal.formatting (module), 47
DelegatedKeyDecryptionError, 28		dynamodb_encryption_sdk.internal.formatting.deseriali (module), 47
DelegatedKeyEncryptionError, 28		dynamodb_encryption_sdk.internal.formatting.deseriali (module), 47
DelegatedKeyError, 28		dynamodb_encryption_sdk.internal.formatting.material (module), 47
DeserializationError, 27		dynamodb_encryption_sdk.internal.formatting.seriali (module), 49
deserialize()	(in module dynamodb_encryption_sdk.internal.formatting.material_description), 47	dynamodb_encryption_sdk.internal.formatting.seriali (module), 49
deserialize_attribute()	(in module dynamodb_encryption_sdk.internal.formatting.deserialize.attri (module), 49	dynamodb_encryption_sdk.internal.formatting.seriali (module), 50
dict_to_ddb()	(in module dynamodb_encryption_sdk.transform), 27	dynamodb_encryption_sdk.internal.identifiers (module), 30
dictionary_validator()	(in module dynamodb_encryption_sdk.internal.validators), 35	dynamodb_encryption_sdk.internal.str_ops (module), 31
		dynamodb_encryption_sdk.internal.utils (module), 32
		dynamodb_encryption_sdk.internal.validators (module), 35
		dynamodb_encryption_sdk.material_providers.aws_kms (module), 14
		dynamodb_encryption_sdk.material_providers.most_re (module), 16

dynamodb_encryption_sdk.material_providers.encrypted_tables_collection_manager (module), 17

dynamodb_encryption_sdk.material_providers.store (module), 18

dynamodb_encryption_sdk.material_providers.encrypted_context_values (module), 15

dynamodb_encryption_sdk.materials.raw (module), 21

dynamodb_encryption_sdk.materials.wrapped (module), 20

dynamodb_encryption_sdk.structures (module), 24

dynamodb_encryption_sdk.transform (module), 27

DynamodbEncryptionSdkError, 27

E

encode_length() (in module dynamodb_encryption_sdk.internal.formatting.serialize), 49

encode_value() (in module dynamodb_encryption_sdk.internal.formatting.serialize), 49

encrypt() (dynamodb_encryption_sdk.internal.crypto.jce_bridge.encrypt_java_cipher method), 41

encrypt() (dynamodb_encryption_sdk.internal.crypto.jce_bridge.primitives.java_symmetric_encryption_algorithm method), 46

encrypt() (dynamodb_encryption_sdk.internal.crypto.jce_bridge.primitives.java_symmetric_encryption_algorithm method), 45

encrypt_attribute() (in module dynamodb_encryption_sdk.internal.crypto.encryption), 37

encrypt_batch_write_item() (in module dynamodb_encryption_sdk.internal.utils), 35

encrypt_dynamodb_item() (in module dynamodb_encryption_sdk.encrypted.item), 12

encrypt_put_item() (in module dynamodb_encryption_sdk.internal.utils), 34

encrypt_python_item() (in module dynamodb_encryption_sdk.encrypted.item), 13

EncryptedClient (class in dynamodb_encryption_sdk.encrypted.client), 11

EncryptedPaginator (class in dynamodb_encryption_sdk.encrypted.client), 11

EncryptedResource (class in dynamodb_encryption_sdk.encrypted.resource), 9

EncryptedTable (class in dynamodb_encryption_sdk.encrypted.table), 8

EncryptedTablesCollectionManager (class in dynamodb_encryption_sdk.encrypted.resource), 9

encrypted_context_values (dynamodb_encryption_sdk.structures.TableInfo attribute), 26

encryption_key (dynamodb_encryption_sdk.materials.raw.RawEncryptionMaterials attribute), 22

encryption_key (dynamodb_encryption_sdk.materials.wrapped.WrappedCryptogram attribute), 20

encryption_materials() (dynamodb_encryption_sdk.encrypted.CryptoConfig method), 7

encryption_materials() (dynamodb_encryption_sdk.material_providers.aws_kms.AwsKmsCryptoConfig method), 15

encryption_materials() (dynamodb_encryption_sdk.material_providers.static.StaticCryptoConfig method), 18

EncryptionContext (class in dynamodb_encryption_sdk.structures), 24

EncryptionError, 28

EncryptionKeyType (class in dynamodb_encryption_sdk.identifiers), 24

F

from_key_schema() (dynamodb_encryption_sdk.structures.TableIndex class method), 25

from_transformation() (dynamodb_encryption_sdk.internal.crypto.jce_bridge.encryption.java method), 42

G

generate() (dynamodb_encryption_sdk.delegated_keys.jce.JceNameLocation class method), 23

get_or_create_provider() (dynamodb_encryption_sdk.material_providers.store.meta.MetaStore method), 19

get_paginator() (dynamodb_encryption_sdk.encrypted.client.EncryptedClient method), 12

I

InvalidAlgorithmError, 28

InvalidArgumentError, 27
 InvalidMaterialDescriptionError, 27
 InvalidMaterialDescriptionVersionError, 27
 InvalidVersionError, 29
 iterable_validator() (in module dynamodb_encryption_sdk.internal.validators), 35

J

JavaAsymmetricEncryptionAlgorithm (class in dynamodb_encryption_sdk.internal.crypto.jce_bridge.primitives), 46
 JavaAuthenticator (class in dynamodb_encryption_sdk.internal.crypto.jce_bridge.authentication), 38
 JavaCipher (class in dynamodb_encryption_sdk.internal.crypto.jce_bridge.encryption), 41
 JavaEncryptionAlgorithm (class in dynamodb_encryption_sdk.internal.crypto.jce_bridge.primitives), 44
 JavaMac (class in dynamodb_encryption_sdk.internal.crypto.jce_bridge.authentication), 39
 JavaMode (class in dynamodb_encryption_sdk.internal.crypto.jce_bridge.primitives), 44
 JavaPadding (class in dynamodb_encryption_sdk.internal.crypto.jce_bridge.primitives), 43
 JavaSignature (class in dynamodb_encryption_sdk.internal.crypto.jce_bridge.authentication), 40
 JavaSymmetricEncryptionAlgorithm (class in dynamodb_encryption_sdk.internal.crypto.jce_bridge.primitives), 45
 JceNameLocalDelegatedKey (class in dynamodb_encryption_sdk.delegated_keys.jce), 23
 JceTransformationError, 28

K

KeyEncodingType (class in dynamodb_encryption_sdk.identifiers), 24

L

load_key() (dynamodb_encryption_sdk.internal.crypto.jce_bridge.authentication), 38
 load_key() (dynamodb_encryption_sdk.internal.crypto.jce_bridge.authentication), 39

load_key() (dynamodb_encryption_sdk.internal.crypto.jce_bridge.authentication), 40
 load_key() (dynamodb_encryption_sdk.internal.crypto.jce_bridge.primitives), 46
 load_key() (dynamodb_encryption_sdk.internal.crypto.jce_bridge.primitives), 45

M

material_description (dynamodb_encryption_sdk.materials.raw.RawDecryptionMaterials attribute), 22
 material_description (dynamodb_encryption_sdk.materials.raw.RawEncryptionMaterials attribute), 21
 material_description (dynamodb_encryption_sdk.materials.wrapped.WrappedCryptogram attribute), 20
 MaterialDescriptionKeys (class in dynamodb_encryption_sdk.internal.identifiers), 31
 MaterialDescriptionValues (class in dynamodb_encryption_sdk.internal.identifiers), 31
 max_version() (dynamodb_encryption_sdk.material_providers.store.meta.MetaStore method), 19
 MetaStore (class in dynamodb_encryption_sdk.material_providers.store.meta), 18
 MinimumKeySizes (class in dynamodb_encryption_sdk.internal.identifiers), 30

N

NoKnownVersionError, 29

O

OnePadding (class in dynamodb_encryption_sdk.internal.crypto.jce_bridge.primitives), 44

P

paginate() (dynamodb_encryption_sdk.encrypted.client.EncryptedPagination method), 11
 primary_index (dynamodb_encryption_sdk.structures.TableInfo attribute), 26
 protected_index_keys() (dynamodb_encryption_sdk.structures.TableInfo attribute), 26
 provider() (dynamodb_encryption_sdk.material_providers.store.meta.MetaStore method), 19
 ProviderStoreError, 28

R

RawDecryptionMaterials (class in *dy-*
namodb_encryption_sdk.materials.raw),
22

RawEncryptionMaterials (class in *dy-*
namodb_encryption_sdk.materials.raw),
21

refresh() (*dynamodb_encryption_sdk.material_providers*
most_recent), 17

refresh_indexed_attributes() (*dy-*
namodb_encryption_sdk.structures.TableInfo
method), 26

ReservedAttributes (class in *dy-*
namodb_encryption_sdk.internal.identifiers),
30

S

secondary_indexes (dy-
namodb_encryption_sdk.structures.TableInfo
attribute), 26

SerializationError, 27

serialize() (in module *dy-*
namodb_encryption_sdk.internal.formatting.material_description),
47

serialize_attribute() (in module *dy-*
namodb_encryption_sdk.internal.formatting.serialize_attribute),
50

set_index_keys() (dy-
namodb_encryption_sdk.structures.AttributeActions
method), 25

sign() (*dynamodb_encryption_sdk.internal.crypto.jce_bridge.authentication.JavaAuthenticator*
method), 39

sign() (*dynamodb_encryption_sdk.internal.crypto.jce_bridge.authentication.JavaMac*
method), 40

sign() (*dynamodb_encryption_sdk.internal.crypto.jce_bridge.authentication.JavaSignature*
method), 41

sign_item() (in module *dy-*
namodb_encryption_sdk.internal.crypto.authentication),
36

SignatureValues (class in *dy-*
namodb_encryption_sdk.internal.identifiers),
31

SignatureVerificationError, 28

signing_key(*dynamodb_encryption_sdk.materials.raw.RawEncryptionMaterials*
attribute), 21

signing_key(*dynamodb_encryption_sdk.materials.wrapped.WrappedEncryptionMaterials*
attribute), 21

SigningError, 28

SimplePadding (class in *dy-*
namodb_encryption_sdk.internal.crypto.jce_bridge.primitives),
43

StaticCryptographicMaterialsProvider
(class in *dy-*
namodb_encryption_sdk.material_providers.static),

17

T

Table() (*dynamodb_encryption_sdk.encrypted.resource.EncryptedResource*
method), 10

table_info() (*dynamodb_encryption_sdk.internal.utils.TableInfoCache*
method), 33

TableIndexCachingMostRecentProvider (dy-
namodb_encryption_sdk.structures), 25

TableInfo (class in *dy-*
namodb_encryption_sdk.structures), 26

TableInfoCache (class in *dy-*
namodb_encryption_sdk.internal.utils), 32

Tag (class in *dynamodb_encryption_sdk.internal.identifiers*),
30

TagValues (class in *dy-*
namodb_encryption_sdk.internal.identifiers),
31

TEXT_ENCODING (in module *dy-*
namodb_encryption_sdk.internal.identifiers),
30

to_bytes() (in module *dy-*
namodb_encryption_sdk.internal.str_ops),
32

to_str() (in module *dy-*
namodb_encryption_sdk.internal.str_ops),
31

transformation (dy-
namodb_encryption_sdk.internal.crypto.jce_bridge.encryption.Java
attribute), 42

unpack_value() (in module *dy-*
namodb_encryption_sdk.internal.formatting.deserialize),
48

unwrap() (*dynamodb_encryption_sdk.internal.crypto.jce_bridge.encrypted.primitives*
method), 42

unwrap() (*dynamodb_encryption_sdk.internal.crypto.jce_bridge.primitives*
method), 45

UnwrappingError, 28

update_item() (dy-
namodb_encryption_sdk.encrypted.client.EncryptedClient
method), 9

update_item() (dy-
namodb_encryption_sdk.encrypted.table.EncryptedTable
method), 9

V

algorithm() (dy-
namodb_encryption_sdk.internal.crypto.jce_bridge.authentication
method), 38

`validate_algorithm()` (*dy-*
namodb_encryption_sdk.internal.crypto.jce_bridge.authentication.JavaMac
method), 39

`validate_algorithm()` (*dy-*
namodb_encryption_sdk.internal.crypto.jce_bridge.authentication.JavaSignature
method), 40

`validate_algorithm()` (*dy-*
namodb_encryption_sdk.internal.crypto.jce_bridge.primitives.JavaEncryptionAlgorithm
method), 44

`validate_decrypt_method()` (*dy-*
namodb_encryption_sdk.encrypted.client.EncryptedPaginator
method), 11

`validate_get_arguments()` (*in module dy-*
namodb_encryption_sdk.internal.utils), 33

`verification_key` (*dy-*
namodb_encryption_sdk.materials.raw.RawDecryptionMaterials
attribute), 22

`verification_key` (*dy-*
namodb_encryption_sdk.materials.wrapped.WrappedCryptographicMaterials
attribute), 21

`verify()` (*dynamodb_encryption_sdk.internal.crypto.jce_bridge.authentication.JavaAuthenticator*
method), 39

`verify()` (*dynamodb_encryption_sdk.internal.crypto.jce_bridge.authentication.JavaMac*
method), 40

`verify()` (*dynamodb_encryption_sdk.internal.crypto.jce_bridge.authentication.JavaSignature*
method), 41

`verify_item_signature()` (*in module dy-*
namodb_encryption_sdk.internal.crypto.authentication),
36

`version_from_material_description()` (*dy-*
namodb_encryption_sdk.material_providers.store.meta.MetaStore
method), 19

`VersionAlreadyExistsError`, 29

W

`with_item()` (*dynamodb_encryption_sdk.encrypted.CryptoConfig*
method), 8

`wrap()` (*dynamodb_encryption_sdk.internal.crypto.jce_bridge.encryption.JavaCipher*
method), 42

`wrap()` (*dynamodb_encryption_sdk.internal.crypto.jce_bridge.primitives.JavaSymmetricEncryptionAlgorithm*
method), 45

`WrappedCryptographicMaterials` (*class in dy-*
namodb_encryption_sdk.materials.wrapped),
20

`WrappedCryptographicMaterialsProvider`
(*class in dy-*
namodb_encryption_sdk.material_providers.wrapped),
15

`WrappingError`, 28