| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/EncryptedPrivateKeyInfo.html) | [**Tree**](http://docs.google.com/package-tree.html) | [**Deprecated**](http://docs.google.com/deprecated-list.html) | [**Index**](http://docs.google.com/index-files/index-1.html) | [**Help**](http://docs.google.com/help-doc.html) | | --- | --- | --- | --- | --- | --- | --- | --- | | | ***Java™ Platform***  ***Standard Ed. 6*** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| [**PREV CLASS**](http://docs.google.com/javax/crypto/CipherSpi.html)   [**NEXT CLASS**](http://docs.google.com/javax/crypto/ExemptionMechanism.html) | [**FRAMES**](http://docs.google.com/index.html?javax/crypto/EncryptedPrivateKeyInfo.html)    [**NO FRAMES**](http://docs.google.com/EncryptedPrivateKeyInfo.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | [CONSTR](#3znysh7) | [METHOD](#2et92p0) | DETAIL: FIELD | [CONSTR](#3dy6vkm) | [METHOD](#17dp8vu) |

## **javax.crypto**

Class EncryptedPrivateKeyInfo

[java.lang.Object](http://docs.google.com/java/lang/Object.html)  
 **javax.crypto.EncryptedPrivateKeyInfo**

public class **EncryptedPrivateKeyInfo**extends [Object](http://docs.google.com/java/lang/Object.html)

This class implements the EncryptedPrivateKeyInfo type as defined in PKCS #8.

Its ASN.1 definition is as follows:

EncryptedPrivateKeyInfo ::= SEQUENCE {  
 encryptionAlgorithm AlgorithmIdentifier,  
 encryptedData OCTET STRING }  
  
 AlgorithmIdentifier ::= SEQUENCE {  
 algorithm OBJECT IDENTIFIER,  
 parameters ANY DEFINED BY algorithm OPTIONAL }

**Since:** 1.4 **See Also:**[PKCS8EncodedKeySpec](http://docs.google.com/java/security/spec/PKCS8EncodedKeySpec.html)

| **Constructor Summary** | |
| --- | --- |
| [**EncryptedPrivateKeyInfo**](http://docs.google.com/javax/crypto/EncryptedPrivateKeyInfo.html#EncryptedPrivateKeyInfo(java.security.AlgorithmParameters,%20byte%5B%5D))([AlgorithmParameters](http://docs.google.com/java/security/AlgorithmParameters.html) algParams, byte[] encryptedData)            Constructs an EncryptedPrivateKeyInfo from the encryption algorithm parameters and the encrypted data. |
| [**EncryptedPrivateKeyInfo**](http://docs.google.com/javax/crypto/EncryptedPrivateKeyInfo.html#EncryptedPrivateKeyInfo(byte%5B%5D))(byte[] encoded)            Constructs (i.e., parses) an EncryptedPrivateKeyInfo from its ASN.1 encoding. |
| [**EncryptedPrivateKeyInfo**](http://docs.google.com/javax/crypto/EncryptedPrivateKeyInfo.html#EncryptedPrivateKeyInfo(java.lang.String,%20byte%5B%5D))([String](http://docs.google.com/java/lang/String.html) algName, byte[] encryptedData)            Constructs an EncryptedPrivateKeyInfo from the encryption algorithm name and the encrypted data. |

| **Method Summary** | |
| --- | --- |
| [String](http://docs.google.com/java/lang/String.html) | [**getAlgName**](http://docs.google.com/javax/crypto/EncryptedPrivateKeyInfo.html#getAlgName())()            Returns the encryption algorithm. |
| [AlgorithmParameters](http://docs.google.com/java/security/AlgorithmParameters.html) | [**getAlgParameters**](http://docs.google.com/javax/crypto/EncryptedPrivateKeyInfo.html#getAlgParameters())()            Returns the algorithm parameters used by the encryption algorithm. |
| byte[] | [**getEncoded**](http://docs.google.com/javax/crypto/EncryptedPrivateKeyInfo.html#getEncoded())()            Returns the ASN.1 encoding of this object. |
| byte[] | [**getEncryptedData**](http://docs.google.com/javax/crypto/EncryptedPrivateKeyInfo.html#getEncryptedData())()            Returns the encrypted data. |
| [PKCS8EncodedKeySpec](http://docs.google.com/java/security/spec/PKCS8EncodedKeySpec.html) | [**getKeySpec**](http://docs.google.com/javax/crypto/EncryptedPrivateKeyInfo.html#getKeySpec(javax.crypto.Cipher))([Cipher](http://docs.google.com/javax/crypto/Cipher.html) cipher)            Extract the enclosed PKCS8EncodedKeySpec object from the encrypted data and return it. |
| [PKCS8EncodedKeySpec](http://docs.google.com/java/security/spec/PKCS8EncodedKeySpec.html) | [**getKeySpec**](http://docs.google.com/javax/crypto/EncryptedPrivateKeyInfo.html#getKeySpec(java.security.Key))([Key](http://docs.google.com/java/security/Key.html) decryptKey)            Extract the enclosed PKCS8EncodedKeySpec object from the encrypted data and return it. |
| [PKCS8EncodedKeySpec](http://docs.google.com/java/security/spec/PKCS8EncodedKeySpec.html) | [**getKeySpec**](http://docs.google.com/javax/crypto/EncryptedPrivateKeyInfo.html#getKeySpec(java.security.Key,%20java.security.Provider))([Key](http://docs.google.com/java/security/Key.html) decryptKey, [Provider](http://docs.google.com/java/security/Provider.html) provider)            Extract the enclosed PKCS8EncodedKeySpec object from the encrypted data and return it. |
| [PKCS8EncodedKeySpec](http://docs.google.com/java/security/spec/PKCS8EncodedKeySpec.html) | [**getKeySpec**](http://docs.google.com/javax/crypto/EncryptedPrivateKeyInfo.html#getKeySpec(java.security.Key,%20java.lang.String))([Key](http://docs.google.com/java/security/Key.html) decryptKey, [String](http://docs.google.com/java/lang/String.html) providerName)            Extract the enclosed PKCS8EncodedKeySpec object from the encrypted data and return it. |

| **Methods inherited from class java.lang.**[**Object**](http://docs.google.com/java/lang/Object.html) |
| --- |
| [clone](http://docs.google.com/java/lang/Object.html#clone()), [equals](http://docs.google.com/java/lang/Object.html#equals(java.lang.Object)), [finalize](http://docs.google.com/java/lang/Object.html#finalize()), [getClass](http://docs.google.com/java/lang/Object.html#getClass()), [hashCode](http://docs.google.com/java/lang/Object.html#hashCode()), [notify](http://docs.google.com/java/lang/Object.html#notify()), [notifyAll](http://docs.google.com/java/lang/Object.html#notifyAll()), [toString](http://docs.google.com/java/lang/Object.html#toString()), [wait](http://docs.google.com/java/lang/Object.html#wait()), [wait](http://docs.google.com/java/lang/Object.html#wait(long)), [wait](http://docs.google.com/java/lang/Object.html#wait(long,%20int)) |

| **Constructor Detail** |
| --- |

### EncryptedPrivateKeyInfo

public **EncryptedPrivateKeyInfo**(byte[] encoded)  
 throws [IOException](http://docs.google.com/java/io/IOException.html)

Constructs (i.e., parses) an EncryptedPrivateKeyInfo from its ASN.1 encoding.

**Parameters:**encoded - the ASN.1 encoding of this object. The contents of the array are copied to protect against subsequent modification. **Throws:** [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if the encoded is null. [IOException](http://docs.google.com/java/io/IOException.html) - if error occurs when parsing the ASN.1 encoding.

### EncryptedPrivateKeyInfo

public **EncryptedPrivateKeyInfo**([String](http://docs.google.com/java/lang/String.html) algName,  
 byte[] encryptedData)  
 throws [NoSuchAlgorithmException](http://docs.google.com/java/security/NoSuchAlgorithmException.html)

Constructs an EncryptedPrivateKeyInfo from the encryption algorithm name and the encrypted data.

Note: This constructor will use null as the value of the algorithm parameters. If the encryption algorithm has parameters whose value is not null, a different constructor, e.g. EncryptedPrivateKeyInfo(AlgorithmParameters, byte[]), should be used.

**Parameters:**algName - encryption algorithm name. See Appendix A in the  [Java Cryptography Architecture Reference Guide](http://docs.google.com/technotes/guides/security/crypto/CryptoSpec.html#AppA) for information about standard Cipher algorithm names.encryptedData - encrypted data. The contents of encrypedData are copied to protect against subsequent modification when constructing this object. **Throws:** [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if algName or encryptedData is null. [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if encryptedData is empty, i.e. 0-length. [NoSuchAlgorithmException](http://docs.google.com/java/security/NoSuchAlgorithmException.html) - if the specified algName is not supported.

### EncryptedPrivateKeyInfo

public **EncryptedPrivateKeyInfo**([AlgorithmParameters](http://docs.google.com/java/security/AlgorithmParameters.html) algParams,  
 byte[] encryptedData)  
 throws [NoSuchAlgorithmException](http://docs.google.com/java/security/NoSuchAlgorithmException.html)

Constructs an EncryptedPrivateKeyInfo from the encryption algorithm parameters and the encrypted data.

**Parameters:**algParams - the algorithm parameters for the encryption algorithm. algParams.getEncoded() should return the ASN.1 encoded bytes of the parameters field of the AlgorithmIdentifer component of the EncryptedPrivateKeyInfo type.encryptedData - encrypted data. The contents of encrypedData are copied to protect against subsequent modification when constructing this object. **Throws:** [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if algParams or encryptedData is null. [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if encryptedData is empty, i.e. 0-length. [NoSuchAlgorithmException](http://docs.google.com/java/security/NoSuchAlgorithmException.html) - if the specified algName of the specified algParams parameter is not supported.

| **Method Detail** |
| --- |

### getAlgName

public [String](http://docs.google.com/java/lang/String.html) **getAlgName**()

Returns the encryption algorithm.

Note: Standard name is returned instead of the specified one in the constructor when such mapping is available. See Appendix A in the  [Java Cryptography Architecture Reference Guide](http://docs.google.com/technotes/guides/security/crypto/CryptoSpec.html#AppA) for information about standard Cipher algorithm names.

**Returns:**the encryption algorithm name.

### getAlgParameters

public [AlgorithmParameters](http://docs.google.com/java/security/AlgorithmParameters.html) **getAlgParameters**()

Returns the algorithm parameters used by the encryption algorithm.

**Returns:**the algorithm parameters.

### getEncryptedData

public byte[] **getEncryptedData**()

Returns the encrypted data.

**Returns:**the encrypted data. Returns a new array each time this method is called.

### getKeySpec

public [PKCS8EncodedKeySpec](http://docs.google.com/java/security/spec/PKCS8EncodedKeySpec.html) **getKeySpec**([Cipher](http://docs.google.com/javax/crypto/Cipher.html) cipher)  
 throws [InvalidKeySpecException](http://docs.google.com/java/security/spec/InvalidKeySpecException.html)

Extract the enclosed PKCS8EncodedKeySpec object from the encrypted data and return it.

Note: In order to successfully retrieve the enclosed PKCS8EncodedKeySpec object, cipher needs to be initialized to either Cipher.DECRYPT\_MODE or Cipher.UNWRAP\_MODE, with the same key and parameters used for generating the encrypted data.

**Parameters:**cipher - the initialized cipher object which will be used for decrypting the encrypted data. **Returns:**the PKCS8EncodedKeySpec object. **Throws:** [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if cipher is null. [InvalidKeySpecException](http://docs.google.com/java/security/spec/InvalidKeySpecException.html) - if the given cipher is inappropriate for the encrypted data or the encrypted data is corrupted and cannot be decrypted.

### getKeySpec

public [PKCS8EncodedKeySpec](http://docs.google.com/java/security/spec/PKCS8EncodedKeySpec.html) **getKeySpec**([Key](http://docs.google.com/java/security/Key.html) decryptKey)  
 throws [NoSuchAlgorithmException](http://docs.google.com/java/security/NoSuchAlgorithmException.html),  
 [InvalidKeyException](http://docs.google.com/java/security/InvalidKeyException.html)

Extract the enclosed PKCS8EncodedKeySpec object from the encrypted data and return it.

**Parameters:**decryptKey - key used for decrypting the encrypted data. **Returns:**the PKCS8EncodedKeySpec object. **Throws:** [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if decryptKey is null. [NoSuchAlgorithmException](http://docs.google.com/java/security/NoSuchAlgorithmException.html) - if cannot find appropriate cipher to decrypt the encrypted data. [InvalidKeyException](http://docs.google.com/java/security/InvalidKeyException.html) - if decryptKey cannot be used to decrypt the encrypted data or the decryption result is not a valid PKCS8KeySpec.**Since:** 1.5

### getKeySpec

public [PKCS8EncodedKeySpec](http://docs.google.com/java/security/spec/PKCS8EncodedKeySpec.html) **getKeySpec**([Key](http://docs.google.com/java/security/Key.html) decryptKey,  
 [String](http://docs.google.com/java/lang/String.html) providerName)  
 throws [NoSuchProviderException](http://docs.google.com/java/security/NoSuchProviderException.html),  
 [NoSuchAlgorithmException](http://docs.google.com/java/security/NoSuchAlgorithmException.html),  
 [InvalidKeyException](http://docs.google.com/java/security/InvalidKeyException.html)

Extract the enclosed PKCS8EncodedKeySpec object from the encrypted data and return it.

**Parameters:**decryptKey - key used for decrypting the encrypted data.providerName - the name of provider whose Cipher implementation will be used. **Returns:**the PKCS8EncodedKeySpec object. **Throws:** [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if decryptKey or providerName is null. [NoSuchProviderException](http://docs.google.com/java/security/NoSuchProviderException.html) - if no provider providerName is registered. [NoSuchAlgorithmException](http://docs.google.com/java/security/NoSuchAlgorithmException.html) - if cannot find appropriate cipher to decrypt the encrypted data. [InvalidKeyException](http://docs.google.com/java/security/InvalidKeyException.html) - if decryptKey cannot be used to decrypt the encrypted data or the decryption result is not a valid PKCS8KeySpec.**Since:** 1.5

### getKeySpec

public [PKCS8EncodedKeySpec](http://docs.google.com/java/security/spec/PKCS8EncodedKeySpec.html) **getKeySpec**([Key](http://docs.google.com/java/security/Key.html) decryptKey,  
 [Provider](http://docs.google.com/java/security/Provider.html) provider)  
 throws [NoSuchAlgorithmException](http://docs.google.com/java/security/NoSuchAlgorithmException.html),  
 [InvalidKeyException](http://docs.google.com/java/security/InvalidKeyException.html)

Extract the enclosed PKCS8EncodedKeySpec object from the encrypted data and return it.

**Parameters:**decryptKey - key used for decrypting the encrypted data.provider - the name of provider whose Cipher implementation will be used. **Returns:**the PKCS8EncodedKeySpec object. **Throws:** [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if decryptKey or provider is null. [NoSuchAlgorithmException](http://docs.google.com/java/security/NoSuchAlgorithmException.html) - if cannot find appropriate cipher to decrypt the encrypted data in provider. [InvalidKeyException](http://docs.google.com/java/security/InvalidKeyException.html) - if decryptKey cannot be used to decrypt the encrypted data or the decryption result is not a valid PKCS8KeySpec.**Since:** 1.5

### getEncoded

public byte[] **getEncoded**()  
 throws [IOException](http://docs.google.com/java/io/IOException.html)

Returns the ASN.1 encoding of this object.

**Returns:**the ASN.1 encoding. Returns a new array each time this method is called. **Throws:** [IOException](http://docs.google.com/java/io/IOException.html) - if error occurs when constructing its ASN.1 encoding.

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/EncryptedPrivateKeyInfo.html) | [**Tree**](http://docs.google.com/package-tree.html) | [**Deprecated**](http://docs.google.com/deprecated-list.html) | [**Index**](http://docs.google.com/index-files/index-1.html) | [**Help**](http://docs.google.com/help-doc.html) | | --- | --- | --- | --- | --- | --- | --- | --- | | | ***Java™ Platform***  ***Standard Ed. 6*** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| [**PREV CLASS**](http://docs.google.com/javax/crypto/CipherSpi.html)   [**NEXT CLASS**](http://docs.google.com/javax/crypto/ExemptionMechanism.html) | [**FRAMES**](http://docs.google.com/index.html?javax/crypto/EncryptedPrivateKeyInfo.html)    [**NO FRAMES**](http://docs.google.com/EncryptedPrivateKeyInfo.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | [CONSTR](#3znysh7) | [METHOD](#2et92p0) | DETAIL: FIELD | [CONSTR](#3dy6vkm) | [METHOD](#17dp8vu) |

[Submit a bug or feature](http://bugs.sun.com/services/bugreport/index.jsp)

For further API reference and developer documentation, see [Java SE Developer Documentation](http://docs.google.com/webnotes/devdocs-vs-specs.html). That documentation contains more detailed, developer-targeted descriptions, with conceptual overviews, definitions of terms, workarounds, and working code examples.

Copyright 2006 Sun Microsystems, Inc. All rights reserved. Use is subject to [license terms](http://docs.google.com/legal/license.html). Also see the [documentation redistribution policy](http://java.sun.com/docs/redist.html).