| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/Security.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/java/security/SecureRandomSpi.html)   [**NEXT CLASS**](http://docs.google.com/java/security/SecurityPermission.html) | [**FRAMES**](http://docs.google.com/index.html?java/security/Security.html)    [**NO FRAMES**](http://docs.google.com/Security.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | CONSTR | [METHOD](#3znysh7) | DETAIL: FIELD | CONSTR | [METHOD](#tyjcwt) |

## **java.security**

Class Security

[java.lang.Object](http://docs.google.com/java/lang/Object.html)  
 **java.security.Security**

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

This class centralizes all security properties and common security methods. One of its primary uses is to manage providers.

| **Method Summary** | |
| --- | --- |
| static int | [**addProvider**](http://docs.google.com/java/security/Security.html#addProvider(java.security.Provider))([Provider](http://docs.google.com/java/security/Provider.html) provider)            Adds a provider to the next position available. |
| static [String](http://docs.google.com/java/lang/String.html) | [**getAlgorithmProperty**](http://docs.google.com/java/security/Security.html#getAlgorithmProperty(java.lang.String,%20java.lang.String))([String](http://docs.google.com/java/lang/String.html) algName, [String](http://docs.google.com/java/lang/String.html) propName)  **Deprecated.** *This method used to return the value of a proprietary property in the master file of the "SUN" Cryptographic Service Provider in order to determine how to parse algorithm-specific parameters. Use the new provider-based and algorithm-independent AlgorithmParameters and KeyFactory engine classes (introduced in the J2SE version 1.2 platform) instead.* |
| static [Set](http://docs.google.com/java/util/Set.html)<[String](http://docs.google.com/java/lang/String.html)> | [**getAlgorithms**](http://docs.google.com/java/security/Security.html#getAlgorithms(java.lang.String))([String](http://docs.google.com/java/lang/String.html) serviceName)            Returns a Set of Strings containing the names of all available algorithms or types for the specified Java cryptographic service (e.g., Signature, MessageDigest, Cipher, Mac, KeyStore). |
| static [String](http://docs.google.com/java/lang/String.html) | [**getProperty**](http://docs.google.com/java/security/Security.html#getProperty(java.lang.String))([String](http://docs.google.com/java/lang/String.html) key)            Gets a security property value. |
| static [Provider](http://docs.google.com/java/security/Provider.html) | [**getProvider**](http://docs.google.com/java/security/Security.html#getProvider(java.lang.String))([String](http://docs.google.com/java/lang/String.html) name)            Returns the provider installed with the specified name, if any. |
| static [Provider](http://docs.google.com/java/security/Provider.html)[] | [**getProviders**](http://docs.google.com/java/security/Security.html#getProviders())()            Returns an array containing all the installed providers. |
| static [Provider](http://docs.google.com/java/security/Provider.html)[] | [**getProviders**](http://docs.google.com/java/security/Security.html#getProviders(java.util.Map))([Map](http://docs.google.com/java/util/Map.html)<[String](http://docs.google.com/java/lang/String.html),[String](http://docs.google.com/java/lang/String.html)> filter)            Returns an array containing all installed providers that satisfy the specified\* selection criteria, or null if no such providers have been installed. |
| static [Provider](http://docs.google.com/java/security/Provider.html)[] | [**getProviders**](http://docs.google.com/java/security/Security.html#getProviders(java.lang.String))([String](http://docs.google.com/java/lang/String.html) filter)            Returns an array containing all installed providers that satisfy the specified selection criterion, or null if no such providers have been installed. |
| static int | [**insertProviderAt**](http://docs.google.com/java/security/Security.html#insertProviderAt(java.security.Provider,%20int))([Provider](http://docs.google.com/java/security/Provider.html) provider, int position)            Adds a new provider, at a specified position. |
| static void | [**removeProvider**](http://docs.google.com/java/security/Security.html#removeProvider(java.lang.String))([String](http://docs.google.com/java/lang/String.html) name)            Removes the provider with the specified name. |
| static void | [**setProperty**](http://docs.google.com/java/security/Security.html#setProperty(java.lang.String,%20java.lang.String))([String](http://docs.google.com/java/lang/String.html) key, [String](http://docs.google.com/java/lang/String.html) datum)            Sets a security property value. |

| **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)) |

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

### getAlgorithmProperty

[@Deprecated](http://docs.google.com/java/lang/Deprecated.html)  
public static [String](http://docs.google.com/java/lang/String.html) **getAlgorithmProperty**([String](http://docs.google.com/java/lang/String.html) algName,  
 [String](http://docs.google.com/java/lang/String.html) propName)

**Deprecated.** *This method used to return the value of a proprietary property in the master file of the "SUN" Cryptographic Service Provider in order to determine how to parse algorithm-specific parameters. Use the new provider-based and algorithm-independent AlgorithmParameters and KeyFactory engine classes (introduced in the J2SE version 1.2 platform) instead.*

Gets a specified property for an algorithm. The algorithm name should be a standard name. See Appendix A in the  [Java Cryptography Architecture API Specification & Reference](http://docs.google.com/technotes/guides/security/crypto/CryptoSpec.html#AppA)  for information about standard algorithm names. One possible use is by specialized algorithm parsers, which may map classes to algorithms which they understand (much like Key parsers do).

**Parameters:**algName - the algorithm name.propName - the name of the property to get. **Returns:**the value of the specified property.

### insertProviderAt

public static int **insertProviderAt**([Provider](http://docs.google.com/java/security/Provider.html) provider,  
 int position)

Adds a new provider, at a specified position. The position is the preference order in which providers are searched for requested algorithms. The position is 1-based, that is, 1 is most preferred, followed by 2, and so on.

If the given provider is installed at the requested position, the provider that used to be at that position, and all providers with a position greater than position, are shifted up one position (towards the end of the list of installed providers).

A provider cannot be added if it is already installed.

First, if there is a security manager, its checkSecurityAccess method is called with the string "insertProvider."+provider.getName() to see if it's ok to add a new provider. If the default implementation of checkSecurityAccess is used (i.e., that method is not overriden), then this will result in a call to the security manager's checkPermission method with a SecurityPermission("insertProvider."+provider.getName()) permission.

**Parameters:**provider - the provider to be added.position - the preference position that the caller would like for this provider. **Returns:**the actual preference position in which the provider was added, or -1 if the provider was not added because it is already installed. **Throws:** [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if provider is null [SecurityException](http://docs.google.com/java/lang/SecurityException.html) - if a security manager exists and its [SecurityManager.checkSecurityAccess(java.lang.String)](http://docs.google.com/java/lang/SecurityManager.html#checkSecurityAccess(java.lang.String)) method denies access to add a new provider**See Also:**[getProvider(java.lang.String)](http://docs.google.com/java/security/Security.html#getProvider(java.lang.String)), [removeProvider(java.lang.String)](http://docs.google.com/java/security/Security.html#removeProvider(java.lang.String)), [SecurityPermission](http://docs.google.com/java/security/SecurityPermission.html)

### addProvider

public static int **addProvider**([Provider](http://docs.google.com/java/security/Provider.html) provider)

Adds a provider to the next position available.

First, if there is a security manager, its checkSecurityAccess method is called with the string "insertProvider."+provider.getName() to see if it's ok to add a new provider. If the default implementation of checkSecurityAccess is used (i.e., that method is not overriden), then this will result in a call to the security manager's checkPermission method with a SecurityPermission("insertProvider."+provider.getName()) permission.

**Parameters:**provider - the provider to be added. **Returns:**the preference position in which the provider was added, or -1 if the provider was not added because it is already installed. **Throws:** [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if provider is null [SecurityException](http://docs.google.com/java/lang/SecurityException.html) - if a security manager exists and its [SecurityManager.checkSecurityAccess(java.lang.String)](http://docs.google.com/java/lang/SecurityManager.html#checkSecurityAccess(java.lang.String)) method denies access to add a new provider**See Also:**[getProvider(java.lang.String)](http://docs.google.com/java/security/Security.html#getProvider(java.lang.String)), [removeProvider(java.lang.String)](http://docs.google.com/java/security/Security.html#removeProvider(java.lang.String)), [SecurityPermission](http://docs.google.com/java/security/SecurityPermission.html)

### removeProvider

public static void **removeProvider**([String](http://docs.google.com/java/lang/String.html) name)

Removes the provider with the specified name.

When the specified provider is removed, all providers located at a position greater than where the specified provider was are shifted down one position (towards the head of the list of installed providers).

This method returns silently if the provider is not installed or if name is null.

First, if there is a security manager, its checkSecurityAccess method is called with the string "removeProvider."+name to see if it's ok to remove the provider. If the default implementation of checkSecurityAccess is used (i.e., that method is not overriden), then this will result in a call to the security manager's checkPermission method with a SecurityPermission("removeProvider."+name) permission.

**Parameters:**name - the name of the provider to remove. **Throws:** [SecurityException](http://docs.google.com/java/lang/SecurityException.html) - if a security manager exists and its [SecurityManager.checkSecurityAccess(java.lang.String)](http://docs.google.com/java/lang/SecurityManager.html#checkSecurityAccess(java.lang.String)) method denies access to remove the provider**See Also:**[getProvider(java.lang.String)](http://docs.google.com/java/security/Security.html#getProvider(java.lang.String)), [addProvider(java.security.Provider)](http://docs.google.com/java/security/Security.html#addProvider(java.security.Provider))

### getProviders

public static [Provider](http://docs.google.com/java/security/Provider.html)[] **getProviders**()

Returns an array containing all the installed providers. The order of the providers in the array is their preference order.

**Returns:**an array of all the installed providers.

### getProvider

public static [Provider](http://docs.google.com/java/security/Provider.html) **getProvider**([String](http://docs.google.com/java/lang/String.html) name)

Returns the provider installed with the specified name, if any. Returns null if no provider with the specified name is installed or if name is null.

**Parameters:**name - the name of the provider to get. **Returns:**the provider of the specified name.**See Also:**[removeProvider(java.lang.String)](http://docs.google.com/java/security/Security.html#removeProvider(java.lang.String)), [addProvider(java.security.Provider)](http://docs.google.com/java/security/Security.html#addProvider(java.security.Provider))

### getProviders

public static [Provider](http://docs.google.com/java/security/Provider.html)[] **getProviders**([String](http://docs.google.com/java/lang/String.html) filter)

Returns an array containing all installed providers that satisfy the specified selection criterion, or null if no such providers have been installed. The returned providers are ordered according to their [preference order](#1t3h5sf).

A cryptographic service is always associated with a particular algorithm or type. For example, a digital signature service is always associated with a particular algorithm (e.g., DSA), and a CertificateFactory service is always associated with a particular certificate type (e.g., X.509).

The selection criterion must be specified in one of the following two formats:

* *<crypto\_service>.<algorithm\_or\_type>*  
  The cryptographic service name must not contain any dots.  
  A provider satisfies the specified selection criterion iff the provider implements the specified algorithm or type for the specified cryptographic service.  
  For example, "CertificateFactory.X.509" would be satisfied by any provider that supplied a CertificateFactory implementation for X.509 certificates.
* *<crypto\_service>.<algorithm\_or\_type> <attribute\_name>:< attribute\_value>*  
  The cryptographic service name must not contain any dots. There must be one or more space charaters between the the *<algorithm\_or\_type>* and the *<attribute\_name>*.  
  A provider satisfies this selection criterion iff the provider implements the specified algorithm or type for the specified cryptographic service and its implementation meets the constraint expressed by the specified attribute name/value pair.  
  For example, "Signature.SHA1withDSA KeySize:1024" would be satisfied by any provider that implemented the SHA1withDSA signature algorithm with a keysize of 1024 (or larger).

See Appendix A in the  [Java Cryptography Architecture API Specification & Reference](http://docs.google.com/technotes/guides/security/crypto/CryptoSpec.html#AppA)  for information about standard cryptographic service names, standard algorithm names and standard attribute names.

**Parameters:**filter - the criterion for selecting providers. The filter is case-insensitive. **Returns:**all the installed providers that satisfy the selection criterion, or null if no such providers have been installed. **Throws:** [InvalidParameterException](http://docs.google.com/java/security/InvalidParameterException.html) - if the filter is not in the required format [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if filter is null**Since:** 1.3 **See Also:**[getProviders(java.util.Map)](http://docs.google.com/java/security/Security.html#getProviders(java.util.Map))

### getProviders

public static [Provider](http://docs.google.com/java/security/Provider.html)[] **getProviders**([Map](http://docs.google.com/java/util/Map.html)<[String](http://docs.google.com/java/lang/String.html),[String](http://docs.google.com/java/lang/String.html)> filter)

Returns an array containing all installed providers that satisfy the specified\* selection criteria, or null if no such providers have been installed. The returned providers are ordered according to their [preference order](#1t3h5sf).

The selection criteria are represented by a map. Each map entry represents a selection criterion. A provider is selected iff it satisfies all selection criteria. The key for any entry in such a map must be in one of the following two formats:

* *<crypto\_service>.<algorithm\_or\_type>*  
  The cryptographic service name must not contain any dots.  
  The value associated with the key must be an empty string.  
  A provider satisfies this selection criterion iff the provider implements the specified algorithm or type for the specified cryptographic service.
* *<crypto\_service>.<algorithm\_or\_type> <attribute\_name>*  
  The cryptographic service name must not contain any dots. There must be one or more space charaters between the *<algorithm\_or\_type>* and the *<attribute\_name>*.  
  The value associated with the key must be a non-empty string. A provider satisfies this selection criterion iff the provider implements the specified algorithm or type for the specified cryptographic service and its implementation meets the constraint expressed by the specified attribute name/value pair.

See Appendix A in the  [Java Cryptography Architecture API Specification & Reference](http://docs.google.com/technotes/guides/security/crypto/CryptoSpec.html#AppA)  for information about standard cryptographic service names, standard algorithm names and standard attribute names.

**Parameters:**filter - the criteria for selecting providers. The filter is case-insensitive. **Returns:**all the installed providers that satisfy the selection criteria, or null if no such providers have been installed. **Throws:** [InvalidParameterException](http://docs.google.com/java/security/InvalidParameterException.html) - if the filter is not in the required format [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if filter is null**Since:** 1.3 **See Also:**[getProviders(java.lang.String)](http://docs.google.com/java/security/Security.html#getProviders(java.lang.String))

### getProperty

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

Gets a security property value.

First, if there is a security manager, its checkPermission method is called with a java.security.SecurityPermission("getProperty."+key) permission to see if it's ok to retrieve the specified security property value..

**Parameters:**key - the key of the property being retrieved. **Returns:**the value of the security property corresponding to key. **Throws:** [SecurityException](http://docs.google.com/java/lang/SecurityException.html) - if a security manager exists and its [SecurityManager.checkPermission(java.security.Permission)](http://docs.google.com/java/lang/SecurityManager.html#checkPermission(java.security.Permission)) method denies access to retrieve the specified security property value [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - is key is null**See Also:**[setProperty(java.lang.String, java.lang.String)](http://docs.google.com/java/security/Security.html#setProperty(java.lang.String,%20java.lang.String)), [SecurityPermission](http://docs.google.com/java/security/SecurityPermission.html)

### setProperty

public static void **setProperty**([String](http://docs.google.com/java/lang/String.html) key,  
 [String](http://docs.google.com/java/lang/String.html) datum)

Sets a security property value.

First, if there is a security manager, its checkPermission method is called with a java.security.SecurityPermission("setProperty."+key) permission to see if it's ok to set the specified security property value.

**Parameters:**key - the name of the property to be set.datum - the value of the property to be set. **Throws:** [SecurityException](http://docs.google.com/java/lang/SecurityException.html) - if a security manager exists and its [SecurityManager.checkPermission(java.security.Permission)](http://docs.google.com/java/lang/SecurityManager.html#checkPermission(java.security.Permission)) method denies access to set the specified security property value [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if key or datum is null**See Also:**[getProperty(java.lang.String)](http://docs.google.com/java/security/Security.html#getProperty(java.lang.String)), [SecurityPermission](http://docs.google.com/java/security/SecurityPermission.html)

### getAlgorithms

public static [Set](http://docs.google.com/java/util/Set.html)<[String](http://docs.google.com/java/lang/String.html)> **getAlgorithms**([String](http://docs.google.com/java/lang/String.html) serviceName)

Returns a Set of Strings containing the names of all available algorithms or types for the specified Java cryptographic service (e.g., Signature, MessageDigest, Cipher, Mac, KeyStore). Returns an empty Set if there is no provider that supports the specified service or if serviceName is null. For a complete list of Java cryptographic services, please see the [Java Cryptography Architecture API Specification & Reference](http://docs.google.com/technotes/guides/security/crypto/CryptoSpec.html). Note: the returned set is immutable.

**Parameters:**serviceName - the name of the Java cryptographic service (e.g., Signature, MessageDigest, Cipher, Mac, KeyStore). Note: this parameter is case-insensitive. **Returns:**a Set of Strings containing the names of all available algorithms or types for the specified Java cryptographic service or an empty set if no provider supports the specified service.**Since:** 1.4

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/Security.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/java/security/SecureRandomSpi.html)   [**NEXT CLASS**](http://docs.google.com/java/security/SecurityPermission.html) | [**FRAMES**](http://docs.google.com/index.html?java/security/Security.html)    [**NO FRAMES**](http://docs.google.com/Security.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | CONSTR | [METHOD](#3znysh7) | DETAIL: FIELD | CONSTR | [METHOD](#tyjcwt) |

[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).