MET56-J. Do not use Object.equals() to compare cryptographic keys

The method `java.lang.Object.equals()`, by default, is unable to compare composite objects such as cryptographic keys. Most `Key` classes fail to provide an `equals()` implementation that overrides `Object.equals()`. In such cases, the components of the composite object must be compared individually to ensure correctness.

Noncompliant Code Example

This noncompliant code example compares two keys using the `equals()` method. The keys may compare unequal even when they represent the same value.

```
private static boolean keysEqual(Key key1, Key key2) {
  if (key1.equals(key2)) {
    return true;
  }
  return false;
}
```

Compliant Solution

This compliant solution uses the `equals()` method as a first test and then compares the encoded version of the keys to facilitate provider-independent behavior. It checks whether an `RSAPrivateKey` and an `RSAPrivateCrtKey` represent equivalent private keys [Oracle 2011b].

```
private static boolean keysEqual(Key key1, Key key2) {
  if (key1.equals(key2)) {
    return true;
  }

  if (Arrays.equals(key1.getEncoded(), key2.getEncoded())) {
    return true;
  }

  // More code for different types of keys here
  // For example, the following code can check whether
  // an RSAPrivateKey and an RSAPrivateCrtKey are equal
  if ((key1 instanceof RSAPrivateKey) &&
      (key2 instanceof RSAPrivateKey)) {

    if ((((RSAKey) key1).getModulus().equals(((RSAKey) key2).getModulus()))
       && (((RSAPrivateKey) key1).getPrivateExponent().equals(
           ((RSAPrivateKey) key2).getPrivateExponent()))) {
      return true;
    }
  }
  return false;
}
```

Automated Detection

Tool	Version	Checker	Description
The Checker Framework	2.1.3	**Interning Checker**	Errors in equality testing and interning (see Chapter 5)

Bibliography

[API 2013]	`java.lang.Object.equals()`, `Object.equals()`
[Oracle 2011b]	Determining If Two Keys Are Equal (JCA Reference Guide)