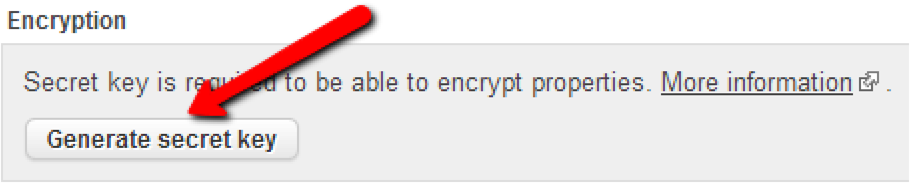
Settings Encryption

Encryption is mostly used to remove clear passwords from settings (database or SCM credentials for instance). The implemented solution is based on a symetric key algorithm. The key point is that the secret key is stored in a secured file on disk. This file must owned by and readable only by the system account that runs the SonarQube server, the analysis with SonarQube Runner, SonarQube Ant Task, Maven or from the Continuous Integration server.

The algorithm is AES 128 bits. Note that 256 bits cipher is not used because it's not supported by default on all Java Virtual Machines (see this article).

1. Generate the secret key

A unique secret key must be shared between all parts of the SonarQube infrastructure (server and analyzers). To generate it, go to **Settings > General Settings > Security > Encryption** and click on **Generate secret key***:*



Secret key is required to be able to encrypt properties. More information .

2. Store the secret key on the SonarQube server

1. Copy the generated secret key to a file:

sonar-secret.txt

```
bIOVA1TybepjqLH+uYxuNh==
```

2. Store this file on the machine hosting the SonarQube server (default location: `~/.sonar/sonar-secret.txt`). If you want to store it somewhere else, set its path through the `sonar.secretKeyPath` property in *SONARQUBE_HOME/conf/sonar.properties*:

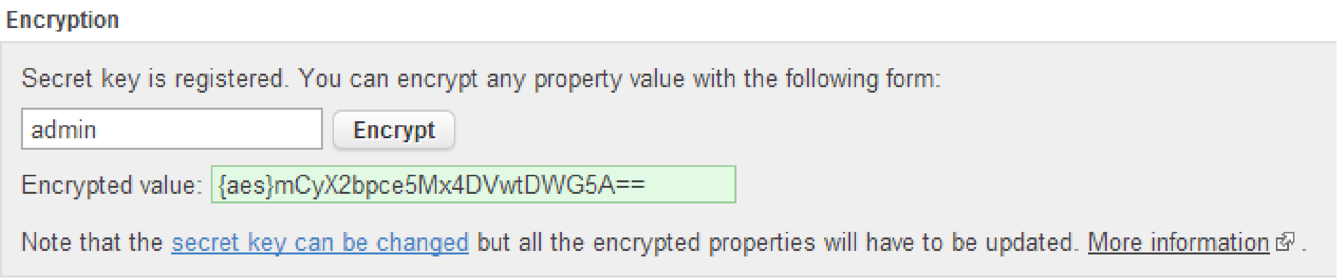
SONARQUBE_HOME/conf/sonar.properties

```
...
# On Linux
sonar.secretKeyPath=/path/to/my/secure/location/my_secret_key.txt
# On Windows
# (use / or \\ as directory separator)
sonar.secretKeyPath=C:/path/to/my/secure/location/my_secret_key.txt
...
```

3. Restrict its access to the system account running the SonarQube server (ownership and read-access only).
4. Restart your SonarQube server.

3. Generate the encrypted values of your settings

Go back to **Settings > General Settings > Security > Encryption** and generate the encrypted values of your settings:



Secret key is registered. You can encrypt any property value with the following form:

admin

Encrypted value: {aes}mCyX2bpce5Mx4DVwtDWG5A==

Note that the secret key can be changed but all the encrypted properties will have to be updated. More information .

4. Use these encrypted values

Server side

Simply copy these encrypted values into *SONARQUBE_HOME/conf/sonar.properties*:

SONARQUBE_HOME/conf/sonar.properties

```
sonar.jdbc.url=jdbc:oracle:thin:@172.16.199.130/XE
sonar.jdbc.username=sonar
sonar.jdbc.password={aes}CCGCFg4Xpm6r+PiJb1Swfg==     # Encrypted password for the database
...
sonar.secretKeyPath=C:/path/to/my/secure/location/my_secret_key.txt
```

Restart your SonarQube server.

Scanner side

To use an encrypted value on the scanner side,

1. Copy the secret key file to the machine running the analysis.
2. Use the encrypted values where needed.
3. Configure the scanner with the location of the secret key file.

sonar-runner.properties

```
...
sonar.secretKeyPath=C:/path/to/my/secure/location/my_secret_key.txt
...
```

Maven settings.xml

```
...
<profile>
  <id>sonar</id>
  <properties>
    ...
    <sonar.secretKeyPath>C:/path/to/my/secure/location/my_secret_key.txt</sonar.secretKeyPath>
  </properties>
</profile>
...
```

SonarQube Scanner for MSBuild's SonarQube.Analysis.xml

```
...
<Property Name="sonar.secretKeyPath">C:/path/to/my/secure/location/my_secret_key.txt</Property>
...
```

Note that the SonarQube Scanner for MSBuild does not yet support encrypting some properties such as "`sonar.host.url`", "`sonar.login`" or "`sonar.password`" (see ~~SONARMSBRU-192~~ - Support encrypting "sonar.login", "sonar.password" properties CLOSED).

⚠ Note also that the Maven encryption mechanism can be used to encrypt password properties.