Шифрование разделов файлов конфигурации в приложениях .NET

1. Пишем свой криптопровайдер файлов конфигурации, наследуя абстрактный [ProtectedConfigurationProvider](http://msdn.microsoft.com/ru-ru/library/system.configuration.protectedconfigurationprovider.aspx), для реализации любого желаемого алгоритма шифрования, либо используем приложенную библиотеку TripleDESProtectedConfigurationProvider.dll (библиотека и исходный код приложены). Важным условием является возможность провайдером экспорта-импорта контейнера с ключами в файл, прикладываемый к приложению при переносе на другую машину. Это даёт нам возможность пользоваться зашифрованным файлом конфигурации на любой другой машине.
2. Кидаем библиотеку с провайдером рядом с исполняемым файлом проекта, конфиги которого планируется шифровать. Следующим кодом формируем контейнер с ключами.

string keyContainer = "ks";

TripleDESProtectedConfigurationProvider provider = new TripleDESProtectedConfigurationProvider();

provider.CreateKey(keyContainer);

, где keyContainer – путь к файлу с ключами.

1. Добавляем в app.config секцию описывающую наш криптопровайдер.

<configProtectedData>

<providers>

<add name="TripleDESProvider" type="MyConfigurationProvider.TripleDESProtectedConfigurationProvider, TripleDESProtectedConfigurationProvider, Version=0.0.0.0, CultureInfo=neutral, PublicKeyToken=4cd79b2bacb58c8b, processorArchitecture=MSIL" keyFilePath="ks" />

</providers>

</configProtectedData>

, где keyFilePath – путь к файлу с ключами, name – имя провайдера, по которому будем подключаться к нему из проекта.

1. Добавляем в проект методы шифрования и расшифровки желаемой секции файла конфигурации.

/// <summary>

/// Protect the connectionStrings section.

/// </summary>

/// <param name="SectionName">sectionName for protect</param>

public static void ProtectConfiguration(String sectionName)

{

// Get the application configuration file.

System.Configuration.Configuration config = ConfigurationManager.OpenExeConfiguration(ConfigurationUserLevel.None);

// Define the provider name.

string provider = "TripleDESProvider";

// Get the section to protect.

ConfigurationSection connStrings = config.GetSection(sectionName) as ConfigurationSection;

if (connStrings != null)

{

if (!connStrings.SectionInformation.IsProtected)

{

if (!connStrings.ElementInformation.IsLocked)

{

// Protect the section. connStrings.SectionInformation.ProtectSection(provider);

connStrings.SectionInformation.ForceSave = true;

config.Save(ConfigurationSaveMode.Full);

}

}

}

}

/// <summary>

/// Unprotect the connectionStrings section.

/// </summary>

/// <param name="SectionName">sectionName for unprotect</param>

public static void UnProtectConfiguration(String sectionName)

{

// Get the application configuration file.

System.Configuration.Configuration config = ConfigurationManager.OpenExeConfiguration(ConfigurationUserLevel.None);

// Get the section to unprotect.

ConfigurationSection connStrings = config.GetSection(sectionName) as ConfigurationSection;

if (connStrings != null)

{

if (connStrings.SectionInformation.IsProtected)

{

if (!connStrings.ElementInformation.IsLocked)

{

// Unprotect the section.

connStrings.SectionInformation.UnprotectSection();

connStrings.SectionInformation.ForceSave = true;

config.Save(ConfigurationSaveMode.Full);

}

}

}

}

, где sectionName – наименование шифруемой секции. Разово каким-либо образом отрабатываем метод ProtectConfiguration для шифрования заданной секции в файле конфигурации. В случае необходимости дешифруем её методом UnProtectConfiguration.

После этого мы сможем пользоваться приложением с зашифрованным файлом конфигурации, на любой машине, без каких-то дополнительных вставок в код по дешифровке данных. Необходимо только наличие файла с ключами и библиотеки с провайдером.