Here's a list of Frequently Asked Questions. I have compiled some of the obvious ones too, though that wasn't the aim; hope it helps to clear up a few things.

1. **Why do I need to use this tool?**

   Using this encryption tool is not a mandatory requirement, Period. It is an option. It is a feature that you may choose to use when required. There are best practices you can adhere to. However, being a developer, aren't we sometimes fallible? How many of us have a clear understanding of an Access Control List and Built-in Windows groups, and how many of us restrict access to a package configuration on a file system? Even if we did everything right, our solutions might not have fit in some scenarios or restrictions from the domain/customer. In some rare scenarios, there might be requirements to run an SSIS package from a web page that runs on a public facing site. Nonetheless, if you are looking for an effortless way to encrypt information in your Sql server configuration file, there you have it.

2. **What are my other options if I don't encrypt? What are the best practices?**

   Read this blog:
   http://consultingblogs.emc.com/jamiethomson/archive/2007/04/26/SSIS_3A00_-Storing-passwords.aspx

   If it is not clear, please read it again. This tool is to encrypt information in xml configuration files.

3. **What if I use other package config option apart from xml?**

   This tool supports only xml configuration files.

4. **Does this tool support configuration values in registry, environments variables, Sql server?**

   No.

5. **What is sensitive information?**

   1. User name
   2. Password
   3. Server name or IP addresses
   4. Port number
   5. Email addresses
   6. Credit card number

7. Web service authentication tokens/keys
8. Anything that your domain/customer deems to protect, etc.

6. Is server name a sensitive information?

For all those who think, a connection string is not sensitive if it does not contain a password; I wish I could refer you to tales of brute force attacks. Attacking a system just does not happen overnight. An intruder uses all possible ways to collect information about a system and its associated systems to find a hole. For instance, when you have a server name, you could run a simple freely available port query tool to scan for open ports and possibly send malicious software. While we have strong Intrusion Prevention systems, the answer is just a rationale to help you decide if a server name is sensitive information. (Don't *EncryptAllWithPassword* and *EncryptAllWithUserKey* from *DTSProtectionLevel* encrypt everything in the package including the server name?)

7. We are protecting information. Agreed. But who are we protecting it from?

The answer depends on many things. Some of them are below. You might want to protect information from

1. A casual observer or a trespasser
2. Someone who is not supposed to have that information (for instance a team member)
3. An attacker
4. Sniffing programs, etc.

There have been rare instances, where a colleague got a sample package from a friend and tried running it, resulting in truncation of database values. If Microsoft did not have security in mind, we wouldn't have DTSProtectionLevel. All these best practices are not just to defend ourselves from just an enemy; rather, they are the recommended ways and options aiming at protecting information that is meant to be protected.

8. Why do I need to export and import Rsa key pair?

During encryption/decryption, cryptographic key management is a very crucial consideration. At some point of time you would find yourself requiring to hard code/save some kind of value to arrive at the cryptographic key when decrypting information, which means anyone with the cryptographic key can decrypt an information (though it is not that simple). In SSISCipherUtil.dll, the cryptographic key management is handed over to Windows, and it is not hardcoded in the source code. Windows APIs expose ways to export and import the cryptographic key. Rest is the basics – when you want to decrypt data, you would require the cryptographic key. That is why we are importing and exporting the Rsa key pair.

9. **Why would I want to protect my server name when I don't use Sql Server Authentication?**

   Refer to Question # 5.

10. **Why would I want to protect my server name, when I am dynamically using an expression to build a connection string on the fly?**

    Refer to Question # 5.

11. **When I use a DSN connection, why would I want to encrypt a connection string?**

    If you think that when using a DSN connection, you would not be required to provide a password in the connection string, I'd say that you'd have to recheck. The reason being, when you create File DSN or System DSN or User DSN and when you have Sql server authentication, it might appear that it is enough to only pass the DSN connection name. Sometimes it might seem to work because it could have been using Windows Authentication. However, when you are using Sql Server authentication, it is required to provide the Password (and User ID), along with the DSN connection name.

12. **When my password changes what are the steps I need perform to update the config file?**

    Follow the steps that would normally do when you want to change a password or a connection string. Once done, use the tool to encrypt it (No, there is no automatic encryption).

13. **Where will store my original config file in the Source control?**

    The RSA key pair used to encrypt and decrypt, along with the SSISCipherUtil.exe should be saved at a location in the source control with restricted access (say a few members of the team). Optionally you may choose to save an unencrypted copy of the config file also.