[fancy logo]

CrypTool 2.0

DEVELOPER INFORMATION

Authors:
Sebastian Przybylski
etc.

July 10, 2009

Version 0.1

Contents

Part I. Developer Guidelines

CrypTool 2.0 uses state-of-the-art technologies like .NET 3.5 and WPF. In order to make your first steps towards developing something in the context of this project a few things need to be considered. In order to not get stuck, please follow the instructions on this page. If you encouter a problem/error, which is not described here, please let us know, so we can add this information to this guide.

In the following we describe all steps necessary in order to compile CrypTool 2.0 on your own. This is always the first thing you need to do before you go on developing own plugins and extensions. The basic steps are

- Getting all pre-requisites and installing them
- Accessing and downloading the source code with SVN
- Compiling the source-code for the first time

1. Pre-requisites

Since CrypTool 2.0 is based on Microsoft .NET 3.5, you first need a Microsoft Windows environment. (Right now no plans exist for porting this project to mono and therefore other platforms.) We successfully tested with **Windows XP** and **Windows Vista**.

Since you're reading the developer guidlines, you probably want to develop something. Hence you need a developer environment. In order to compile our sources you need **Microsoft Visual Studio 2008 Professional**. Please make sure you always install the latest service packs for Visual Studio too. Unfortunately it does not work (smoothly) with the freely available Visual Studio Express (C#) versions. This is due to the fact, that CrypWin uses a commercial component, and is therefore distributed only as binary. The C# Express version unfortunately cannot handle a binary as a start project, hence debugging becomes cumbersome.

Usually the installation of Visual Studio also installs the .NET framework. In order to run/compile our source code you need (at the time of this writing) at least **Microsoft .NET 3.5 with Service Pack 1 (SP1)**. You can get this freely from Microsofts webpage.

After the last step, your development environment should be ready for our source-code. Hence, now you need a way of accessing and downloading the entire sources. In the CrypTool 2.0 project we use SVN (subversion control) for version control, hence you need an **SVN client** of you're choice, e.g. **TortoiseSVN** or the **svn commandline**

from cygwin. If you never worked with SVN before, we suggest to download and install TortoiseSVN, since it offers a nice windows explorer integration of SVN and any windows user should feel right away at home.

2. Accessing Subversion Control (SVN)

This section describes how to access our SVN and the basic settings you need.

SVN URL

Our code repository is accessable under the following url:

`https://www.cryptool.org/svn/CrypTool2/`

If you are a guest and just want to download our source code you can use "anonymous" as the username and an empty password. If you are a registered developer, just use your provided username and password (which is the same as for this wiki).

Accessing the SVN with TortoiseSVN

As already mentioned, in order to use the SVN repository one of the best options is TortoiseSVN. Please install TortoiseSVN (unfortunately it will ask you to reboot, which you need to do) and then create somewhere on your computer a directory (for instance "Cryptool2") for storing the local working files. Right click on this directory and select "SVN Checkout" from the context menu. In the new appearing window you must enter the URL of the repository as given above. The "Checkout directory" should be filled in correctly. After that, just hit ok, accept the certificate (if necessary) and enter your user credentials or "anonymous" for guests. Also mark the checkbox for saving your crendentials otherwise you will be asked about them for every single file. Then hit ok, and now the whole CrypTool2 repository should be checked out into the given directory.

Later on, if you just want to update (if there where changes in the repository) you can do this with right click on any directory within the working files and choose "SVN Update" from the context menu. If you changed a file you should choose "SVN Commit" from the context menu in order to upload your changes. Please always provide *meaningfull descriptions* of your updates.

A TortoiseSVN Tutorial can be found here.

Ignore Patterns

In order to checkin only clean code, please use the following **ignore patterns**: *"obj bin debug release *.pdb *.suo *.exe *.dll"*

This basically means that you should never check-in compiled and user-generated files. As an example please do not check-in the entire bin/ and obj/ directories which Visual Studio generates. If you want to submit a component (binary file) despite the ignore patterns you can still add *.dll files by using the context menu and add that file explicitely – but please be absolutely sure, that you know what you are doing.

3. Compiling the sources

At this point you should have checked out the entire CrypTool repository. Then compiling is pretty easy, you just go to the directory *trunk/* and open the **CrypTool 2.0.sln** Visual Studio solution. Now Visual Studio should open with all working plugins and all components nicely arranged. In case you started Visual Studio now for the very first time, you must choose a certain settings – just select either "most common" or "C#" – you can change this at any time later. In the right hand you get the project explorer, where you see all the subprojects included in the solution. You have to look for the project **CrypWin.exe** there. When you found it, you need to right-click it and select **"Set as startup-project"** from the context menu. After you have done this, just go to the menu *Build* and select *Build solution* (clearly you can also use the hotkeys if you memorized them). Then go to *Debug* and click *Start debugging* – now CrypTool 2.0 should start for the first time with your own compiled code – clearly you did not change yet anything, however, you have now an own build of all components (with the exception of CrypWin and AnotherEditor, since they are available only as binary). In case it does not compile or start, please consult our F.A.Q. and let us know if you found a bug.

As a core-developer, hence somebody who can also compile CryWin and AnotherEditor, you should use the **CrypTool 2.0.sln** solution from the trunk/CoreDeveloper/ directory (this directory is **not** visible to you if you are not a core developer). As a core developer you should know, that when compiling you **change** the CryWin.exe which is visible to everybody else. Hence, when doing a checkin, please make sure you *really* want to checkin a new binary.

Part II. Create an Encryption-Plugin using Visual Studio 2008

4. Create a new project in VS2008 for your plugin

Open Visual Studio 2008 and create a new project:

Select "‘.NET-Framework 3.5"‘ as the target framework (the Visual Studio Express edition don't provide this selection because it automatically chooses the actual target framework), and "‘Class Library"‘ as default template to create a DLL file. Give the project a unique and significant name (here: "‘Caesar"‘), and choose a location where to save (the Express edition will ask later for a save location when you close your project or your environment). Finally confirm by pressing the "‘OK"‘ button.

Now your Visual Studio solution should look like this:

5. Select the interface, your plugin wants to serve

First we have to add a reference to the Cryptool library called "‘CrypPluginBase.dll"‘ where all necessary Cryptool plugin interfaces are declared.

and select the library by double clicking the file or pressing the "‘OK"‘ button.

6. Create the classes for the algorithm and for its settings

6.1. Create the class for the algorithm (MD5)

6.2. Create the class for the settings (MD5Settings)

6.3. Add namespace for the class MD5 and the place from where to inherit

6.4. Add the interface functions for the class MD5

6.5. Add namespace and interfaces for the class MD5Settings

6.6. Add controls for the class MD5Settings (if needed)

7. Select and add an image as icon for the class MD5

8. Set the attributes for the class MD5

9. Set the private variables for the settings in the class MD5

10. Define the code of the class MD5 to fit the interface

11. Complete the actual code for the class MD5

12. Sign the created plugin

13. Import the plugin to Cryptool and test it

14. Source code and source template