

Im Herbst 2007 begann eine kleine Gruppe erfahrener Software-Entwickler der Universitäten Duisburg-Essen und Siegen, den Grundstein für einen adäquaten Nachfolger des inzwischen weltweit bekannten E-Learning-Programms CrypTool 1.x zu legen. Im Frühjahr 2008 wurde die dann fertig gestellte Architektur und das dazu gehörende Framework der Open-Source-Gemeinde zur Verfügung gestellt. Auf diese Weise soll CrypTool 2.0 zügig auf einen Funktionsumfang gebracht werden, der einer Bezeichnung als *Nachfolger* des äußerst umfangreichen CrypTool 1.x gerecht wird. Bis August 2008 erschienen bereits erste Alpha- und Beta-Versionen von CrypTool 2.0.

**Die Architektur** von CrypTool 2.0 wurde so entworfen, dass sie flexibel ist und Entwicklern gleichzeitig ein schlankes Framework zur Verfügung stellt. Der Aufwand zur Integration neuer Funktionalität, insbesondere zusätzlicher Algorithmen, wird deutlich geringer ausfallen, als das bei CrypTool 1.x der Fall war. Ermöglicht wird dies durch ein vollkommen modulares Design, in dem fast jede Komponente ein Plugin ist. Die Abbildung rechts gibt einen groben Überblick über die Architektur von CrypTool 2.0:

In der oberen Schicht sind zwei unterschiedliche Benutzer-Schnittstellen zu sehen — eine leicht zu automatisierende Schnittstelle für die Kommandozeile (*CrypConsole*) und eine auf der *Windows Presentation Foundation (WPF)* basierende GUI (*CrypWin*). Die GUI besteht aus zwei Komponenten: Zum einen das Programm-Fenster mit seinen Haupt-Steuerelementen. Es wurde an den von Microsoft veröffentlichten *User Interface Design Guidelines* zu Office 2007 ausgerichtet. Zum anderen die im Programm-Fenster angezeigte Benutzer-Schnittstelle zur Verwendung der Algorithmen. Sie wird durch ein *Editor-Plugin* realisiert. Dieses *Editor-Plugin* bietet eine einfache Form der visuellen Programmierung, also der grafischen Verkettung von Einzelverfahren zu einem Protokoll-Ablauf.

In der mittleren Schicht findet sich der Kern (*CrypCore*) von CrypTool 2.0, dessen Hauptaufgabe in der Verwaltung aller Plugins liegt. Hier wurden insbesondere Microsofts Regeln für Mehrbenutzer-Systeme beachtet. So wird neben dem globalen Plugin-Speicher auch jeweils ein Plugin-Speicher für normale Benutzer verwaltet. Der globale Speicher ist lediglich durch den Administrator modifizierbar. Er enthält die für alle Nutzer verfügbaren Standardkomponenten. Zusätzlich kann jeder Nutzer eigene, nur ihm zugängliche Plugins laden, die in seinem persönlichen Speicher verwaltet werden.

In der *CrypPluginBase* findet sich eine Sammlung von Schnittstellen und Attributen. Der Entwickler eines Plugins implementiert die zu seinem Algorithmus oder zu seiner Visualisierung entsprechende Schnittstelle und beschreibt die neu implementierten Klassen mithilfe der in der *CrypPluginBase* definierten Attribute. Diese Form der deklarativen Programmierung ermöglicht es dem Kern von CrypTool 2.0 (*CrypCore*), jedes Plugin anhand seiner Metadaten zu klassifizieren und zu verwalten.

Mit dem *CrypDownloadService* wird ein global verfügbarer Webservice bereit gestellt, der alle durch das CrypTool-Projekt zertifizierten Plugins auflistet und zum Herunterladen anbietet. CrypTool 2.0 enthält eine Client-Schnittstelle, über die der Webservice für den Benutzer transparent angesprochen wird. Damit in CrypTool 2.0 nur durch das CrypTool-Projekt zertifizierte Plugins verwendet werden, lassen sich im normalen Benutzer-Modus lediglich digital signierte Plugins nachladen. Darüber hinaus wird es in einem Entwickler-Modus von CrypTool 2.0 möglich sein, nicht zertifizierte Plugins zu integrieren.

**Mehrsprachigkeit** wurde von Anfang an berücksichtigt. Im Gegensatz zu CrypTool 1.x lässt sich die Sprache in CrypTool 2.0 zur Laufzeit umschalten. Um eine weltweit nutzbare Standard-Version von CrypTool 2.0 bereitstellen zu können, muss jeder Entwickler eine englische Benutzer-Schnittstelle samt Dokumentation bereit stellen. Übersetzungen lassen sich dann leicht mit Hilfe des separaten Programms *CrypResource* durch eine andere Person erstellen.

Mit Hilfe des **Inline Code Editor** lässt sich zur Laufzeit des Programms C#-Code kompilieren und ausführen. Damit können insbesondere Lehrende wertvolle Übungsaufgaben für ihre Studenten erstellen. So kann man den Studenten das Code-Gerüst eines Algorithmus bereitstellen, in dem sie dann eine bestimmte Funktion selbst programmieren müssen. Das Ergebnis lässt sich anschließend gegen eine in CrypTool 2.0 vorliegende Referenzimplementierung prüfen.

