# Besprechungsprotokoll

|  |  |
| --- | --- |
| Titel | Zusammenfassung der CT2-Evaluation von Herrn Christian Meyer und Herleitung notwendiger Arbeiten |
| Datum | 12. und 13. November 2009 |
| Teilnehmer | Dr. Arno Wacker, Matthäus Wander |

## Einleitende Informationen

Art = **A**ction/**I**nformation/**D**ecision/**O**pen Issue

| # | Art | Beschreibung | Referenz (Kap.) |
| --- | --- | --- | --- |
| 1 | I | Grundsätzlich fehlen noch einige wichtige Programm-Features, etwa:   * Online-Hilfe * Deutschsprachige Oberfläche | 2.2 |
| 2 | I | Es fehlen Visualisierungen bei den meisten Algorithmen. | 2.2 |
| 3 | I | Es fehlen bedeutsame Algorithmen und Verfahren, etwa:   * ECC * MARS * RC6 * Serpent * DSA * Schnorr-Signatur * Merkle-Signatur | 2.2 |
| 4 | I | Es fehlen weitere kryptoanalytische Werkzeuge. | 2.2 |
| 5 | I | Es gibt keine Lerneinheiten in CT2, die für Personen ohne Krypto-Vorwissen hilfreich wären. | 3.3.4 |
| 6 | I | Es gibt keine motivierende Lernimpulse, beispielsweise:   * Geheimnisse, Rätsel * Alltagsbezug, z. B. „typische Bankverschlüsselung“ * Erläuterung der Sicherheitsziele Vertraulichkeit, Integrität und Authentizität | 3.3.5 |
| 7 | I | Bei der Vorbereitung der Probanden-Befragung ergaben sich die folgenden Schwierigkeiten:   * Keine asymmetrischen Kryptosysteme (RSA, El Gamal) * Fehlerhafte/unvollständige Implementierungen bzgl. Enigma, AES, KasiskiTest und FrequencyTest * Schwache Usability für unerfahrene Benutzer   Die genannten Punkte sind teilweise bereits gelöst (RSA, AES, FrequencyTest), teilweise tauchen sie weiter unten als konkrete Aufgaben (A) auf. | 3.4 |
| 8 | I | Von den 25 Probanden hatten 5 Personen (20%) kein kompatibles Betriebssystem (Mac, Linux). | 3.4 |
| 9 | I | Entscheidend zum erfolgreichen Umgang mit CT2 scheint „die Vertrautheit mit allgemeinen Konzepten von Programmierung“ zu sein. | 5 |
| 10 | I | Entscheidend für den didaktischen Erfolg ist eine konsistente Darstellung und Bedienung. | 5.6 |

## Verbesserung vorhandener Funktionen

Art = **A**ction/**I**nformation/**D**ecision/**O**pen Issue

| # | Art | Beschreibung | Referenz (Kap.) |
| --- | --- | --- | --- |
| 11 | A | Weiße Schrift im Splash/About-Fenster durch schwarze Schrift ersetzen. | 4.1.2 |
| 12 | A | Für den Benutzer irrelevante Log-Meldungen im Splash-Fenster dezenter darstellen, etwa in hellgrauer Schrift. | 4.1.2 |
| 13 | I | Das Editor-Fenster ist beim Programmstart zu klein. | 4.2 |
| 14 | A | ToolTips in der RibbonBar durch mehrzeilige ScreenTips ersetzen (vgl. Word 2007). | 4.2.1 |
| 15 | A | Funktionslose Dialog Box Launcher (Ecke unten rechts) in der RibbonBar entfernen. | 4.2.1 |
| 16 | A | Icons für die Ribbon-Gruppen hinzufügen, die beim Verkleinern der Fensterbreite angezeigt werden (zur Zeit leeres Icon). | 4.2.1 |
| 17 | D | Zusätzlich sollte eine minimale Fenstergröße für CrypWin eingestellt sein, sowie für alle visuellen Element mit variabler Größe (Panes, Panels). |  |
| 18 | I | Für den Benutzer ist verwirrend, dass im Play-Mode einige Felder geändert werden können, andere jedoch nicht (Settings). | 4.2.1 |
| 19 | A | Funktion Ausschneiden, Kopieren, Einfügen für Plugin-Instanzen einbauen (Settings sollten mitkopiert werden). | 4.2.1 |
| 20 | A | Die Verwaltung von Sub-Workspaces ist fehlerhaft und sollte überarbeitet werden. SWS sollten sich wie normale Projekte öffnen, bearbeiten und speichern lassen. | 4.2.1 |
| 21 | A | Die Option „Detail Level“ sollte ersatzlos entfernt werden, da sie   * „einen Zusammenhang zu den Vorerfahrungen nahe legt, der in keiner Weise gegeben ist“ (S. 41) * fehlerhaft ist und bei den meisten Plugins ohnehin keine Verwendung findet (S. 43f) * „unnatürlich, irreführend und kompliziert ist“ (S. 65) | 4.2.1 |
| 22 | D | Die Toggle-Funktion des Buttons „Reopen last file“ ist nicht intuitiv. Sollte zu Checkbox geändert werden. | 4.2.1 |
| 23 | D | Die Ribbon-Gruppe „Plugins“ sollte unter „Home“ eingeordnet sein, nicht unter „Settings“. | 4.2.1 |
| 24 | A | Das Ribbon-Tab „Algorithms“ streichen, da es redundant ist und den Benutzer verwirrt. | 4.2.1 |
| 25 | I | Das Fenster „Algorithm Settings“ ist ebenfalls aus verschiedenen Gründen nachteilig und sollte entfernt werden, was jedoch nicht ersatzlos möglich ist. Hierzu müsste zuerst #53 umgesetzt werden. | 4.2.3 |
| 26 | D | Beim Öffnen eines Projekts sollte grundsätzlich die Funktion „Fit2Screen“ (Autoscale) durchgeführt werden.  Ggf. kann dies durch ein Setting, welches in der Projektdatei gespeichert wird (nicht CT2-global), abgeschaltet werden. |  |
| 27 | D | Der Editor zeigt die vertikale Scrollbar, obwohl noch Platz vorhanden ist. Dies sollte korrigiert werden. |  |
| 28 | D | Ein Klick auf das „Minimize“-Icon des LogWindow lagert das Tab an einer ungünstigen Stelle ein (unter der Algorithms-Navigation). Es sollte stattdessen direkt unter dem Editor eingelagert werden. |  |
| 29 | D | Die Fenstereinstellungen von CrypWin sollten speicherbar sein.  Ggf. als Checkbox-Option im Settings-Ribbon-Tab hinzufügen.  Ggf. sollte es einen Button zum Reset geben. |  |
| 30 | I | Die Notwendigkeit Workspaces starten und stoppen zu müssen war für die meisten Probanden verwirrend. | 4.2.5 |
| 31 | D | Für das vorgeschlagene Streichen des Play-Buttons (automatisches Play) sehen wir keine sinnvolle Umsetzungsmöglichkeit. |  |
| 32 | O | Für das vogeschlagene Streichen des Stop-Buttons (automatisches Stop) sehen wir die folgenden Umsetzungsmöglichkeiten:   * Editor-Feature einer „Stop-Senke“. Die Stop-Senke verhält sich wie ein Plugin, das in den Workspace gezogen werden kann und das einen Eingang hat. Sobald am Eingang ein Trigger anliegt, wird der Workspace gestoppt. Es stellt somit ein „exit“ oder „return“ aus dem visuellen Programm dar. * Aufploppende Meldung. Wenn der Benutzer im Play-Mode eine Aktion durchführt, die nicht möglich ist (Reinziehen eines Plugins, Ändern von Verbindungen o.ä.), erscheint eine Meldung (Dialogbox, Bubble etc.), die den Benutzer auf den noch laufenden Workspace hinweist (und evtl. ihm anbietet den Workspace zu stoppen). |  |
| 33 | O | Evtl. kann man die Play/Stop Buttons auffälliger in die Oberfläche einbauen? Z.B. im Editor-Fenster unten mitte (vergleichbar mit einem Video-Player) |  |
| 34 | I | Das Konzept der „farbigen Datentypen“ ist sehr hilfreich und sollte auch in Zukunft weiter verfolgt werden. | 4.2.5 |
| 35 | O | Die dynamischen Properties beim Eingang des TextOutput sind leicht zu übersehen. Vorschlag von Herrn Meyer ist ein „Multi-Datentyp-Eingang“, der mehrfarbig oder transparent ist und mehrere unterschiedliche Datentypen akzeptiert.  Äquivalent dazu kann man den TextInput mit einem „Multi-Datentyp-Ausgang“ versehen, wobei dort die unterschiedliche Semantik der Ausgänge Schwierigkeiten bereitet (byte[] ist anders als bool[]). |  |
| 36 | A | Notwendige und optionale Eingänge sollen sich visuell unterscheiden, etwa durch unterschiedliche Größe der Dock-Punkte. | 4.2.5 |
| 37 | A | Der ToolTip der Ein- und Ausgänge ist „zu technisch“ gehalten. Er sollte zu einem ScreenTip mit leichter verständlichen Hinweisen umgebaut werden. | 4.2.5 |
| 38 | A | Abzweigungen und Kreuzungen der Linien im Workspace sollten sich visuell unterscheiden, Beispiel: | 4.2.5 |
| 39 | A | Der Algorithmus zur Anordnung der Linien sollte überarbeitet werden, und/oder der Benutzer sollte manuell die Anordnung anpassen können (vgl. mit MS Robotics Developer Studio). | 4.2.5 |
| 40 | A | Plugin-Fehlermeldungen und Warnungen sollten direkt am Plugin gezeigt werden, Beispiel: | 4.2.5 |
| 41 | A | Es sollten mehrere Projekte als Tabs gleichzeitig geöffnet werden können. | 4.2.5 |
| 42 | A | Die Funktion Multi-Workspace soll nach Umsetzung von #41 entfernt werden. | 4.2.5 |
| 43 | A | Alle Tabs sollten sich einheitlich verhalten. Jedes Tab sollte eine Schließen-Funktion haben, möglichst auch eine Funktion zum Verändern der Größe.  Vgl. auch #53. | 4.2.5 |
| 44 | A | Im Kontextmenü eines Plugins sollten keine Settings enthalten sein. Jedes Plugin sollte das gleiche Kontextmenü haben. | 4.2.6 |
| 45 | I | Bei den Wikipedia-Links in den Descriptions wird nicht deutlich, ob diese eine ergänzende Lektüre oder eine Quellenangabe darstellen. | 4.2.6 |
| 46 | D | Alle Descriptions sollten anhand einer zu erstellenden Checkliste überarbeitet werden. Die Checkliste sollte im Wiki abgelegt werden. |  |
| 47 | A | Die Funktionen Suche und Skalierung im Description-Viewer sollten sich unten rechts im jeweiligen Fenster befinden. | 4.2.6 |
| 48 | A | Die Spalten- und Blätterfunktion sollten entfernt werden. Stattdessen sollte eine scrollbare Seite wie in einem Webbrowser präsentiert werden. | 4.2.6 |
| 49 | D | Ein erwähnter optischer Mangel im FrequencyTest ist bereits behoben, aber gleichermaßen im KasiskiTest vorhanden. Dies sollte dort ebenfalls behoben werden, oder alternativ sollte die generische Funktionalität zur Anzeige eines Histogramms in eine gemeinsame Basisklasse gezogen werden. | 4.2.6 |
| 50 | D | Der Vorschlag die Hintergrundfarbe eines Comparator-Plugins entsprechend dem Ergebnis (true=grün, false=rot) anzupassen, erscheint problematisch. | 4.2.6 |
| 51 | D | Der Vorschlag alle Input- und Output-Plugins in ein universelles Plugin zu ziehen, erscheint problematisch. Einzelne Plugins lassen sich durchaus zusammenfassen (TextInput+FileInput), aber nicht Input und Output gemischt. | 4.2.6 |

## Entwicklung neuer Funktionen

Art = **A**ction/**I**nformation/**D**ecision/**O**pen Issue

| # | Art | Beschreibung | Referenz (Kap.) |
| --- | --- | --- | --- |
| 52 | O | Der Editor sollte um typische Funktionen eines Zeichenprogramms ergänzt werden:   * Einfügen von Rechtecken, Kreisen, Linien, Pfeilen, Wolken * Einfügen von kurzen Zeichenketten (vgl. Memo, aber auch geeignet für Einzeiler und mit transparentem Grund) * Einfügen von Grafiken/Bildern | 4.2.5 |
| 53 | O | Es sollten mehr Funktionen direkt in die Plugin-Instanz gezogen werden. So sollte der Benutzer über Buttons zwischen den folgenden Dingen umschalten können:   * Settings * Werte der Ein/Ausgänge * Visualisierung * Description   Da die Plugin-Instanz hierfür mehr Platz braucht, müssen die folgenden Funktionen zum schnellen Ändern der Größe eingebaut werden:   * Zum Icon minimieren * Zur ursprünglichen Größe zurückkehren * Stark vergrößern/maximieren   Diese sollten über kleine Buttons oder über das Kontextmenü anwählbar sein.  Es entfallen die folgenden Funktionen:   * Fenster „Algorithm Settings“ (vgl. #25) * Trennung zwischen QuickWatchPresentation und Presentation * Anzeige der Description in einem eigenen Tab   Das ganze kann folgendermaßen aussehen: | 4.2.6 |
| 54 | O | Ein **Startcenter** soll den Benutzer beim ersten Programmstart begrüßen. Darin sollen Links/Buttons zu Hilfeseiten führen und elementare Funktionen (Samples laden, Play/Stop) erklären. | 5.1 |
| 55 | O | Eine **Guided Tour** wäre hilfreich, um den Benutzer in das Konzept von CrypTool einzuführen. Da CT2 diesem Konzept bislang jedoch keinerlei Grundlage liefert, ist die Umsetzung sehr aufwendig. | 5.2 |
| 56 | O | Herr Meyer schlägt die Einführung von neun verschiedenen Lernebenen für jedes Plugin vor:   * Geschichte * Einführung * Skript * Tutorial * Visuelles Programm * Programmcode * Visualisierung * Übungen * Kurzweiliges   Diese sollten etwa über ein „Context Tab“ in der Ribbon Bar angezeigt werden:    *Geschichte*, *Einführung*, *Skript* und *Übungen* können durch eine entsprechend erweiterte Description abgedeckt werden (vgl. #46).  *Tutorial* ist entweder statisch als Teil der Description (#46) oder dynamisch als Unterpunkt der Guided Tour (#55).  *Visuelles Programm* meint eine grafische Implementation des C# Codes, was nicht umsetzbar ist. Man kann hingegen auf ein Sample verweisen.  Der C# *Programmcode* könnte beim Build-Prozess automatisch eingebunden. Dieses Feature ist mit vertretbarem Aufwand umsetzbar, solange der Code vom Benutzer nur gelesen und nicht zur Laufzeit modifiziert und kompiliert werden kann.  *Visualisierung* ist bereits in der Plugin-Instanz auswählbar, vgl. (#5).  *Kurzweiliges* ist nicht bei jedem Plugin sinnvoll. Stattdessen könnte man eine eigene Plugin-Kategorie mit Lernspielen erstellen. | 5.3 |