Nils Kopal, Olga Kieselmann, Arno Wacker, Bernhard Esslinger

CrypTool 2.0

Open-Source Kryptologie für Jedermann

Die Lernsoftware CrypTool 2.0 bietet Schülern, Studierenden und Kryptologie-Begeisterten einen einfachen Zugang in die Welt der Kryptographie und der Kryptoanalyse. Nach siebenjähriger Entwicklungszeit hat das CrypTool-2-Team die erste „Release-Version" veröffentlicht. CrypTool 2.0 wird schon jetzt in Schulen, Hochschulen, Unternehmen und Behörden zur Ausbildung, zum Selbststudium und für Awareness-Maßnahmen eingesetzt. Es ist Open-Source und kann von jedermann frei genutzt werden. CrypTool 2.0 beinhaltet eine Vielzahl von Verschlüsselungs- und Analyseverfahren. Innerhalb dieses Artikels erhält der Leser einen kurzen Einblick in das CrypTool-Projekt und einen tiefergehenden in die Software CrypTool 2.0.

Nils Kopal, M.Sc.

Wissenschaftlicher Mitarbeiter, Fachgebiet Angewandte Informationssicherheit, Universität Kassel, Technischer Leiter CrypTool 2.0

E-Mail: nils.kopal@uni-kassel.de

Dipl.-Inf. Olga Kieselmann

Wissenschaftliche Mitarbeiterin, Fachgebiet Angewandte Informationssicherheit, Universität Kassel

E-Mail: olga.kieselmann@uni-kassel.de

Prof. Dr. Arno Wacker

Leiter des Fachgebiets Angewandte Informationssicherheit, Universität Kassel, Projektleiter CrypTool 2.0

E-Mail: arno.wacker@uni-kassel.de

Prof. Bernhard Esslinger

Professor für IT Security und Kryptologie, Universität Siegen, Leiter des CrypTool-Projekts

E-Mail: bernhard.esslinger@uni-siegen.de

1 Einleitung

Insbesondere nach den Veröffentlichungen von Edward Snowden, ist der Bedarf an (verlässlicher) IT-Sicherheit innerhalb der Bevölkerung deutlich gestiegen. Zeitgleich ist das Vertrauen in viele der eingesetzten Verfahren gesunken. Viele Menschen nutzen IT-Sicherheitsverfahren (kryptographische Protokolle und Algorithmen) in ihrem täglichen Leben, ohne zu wissen, wie diese richtig und vor allem sicher einzusetzen sind. Sie nutzen schwache Passwörter für die Authentifizierung, das unsichere WEP-Protokoll im heimischen Netzwerk, Mail-Clients ohne die integrierte S/MIME-Funktion [1], unvertraute TLS-Verbindungen während des Online-Einkaufs – und viele weitere unsichere oder (eigentlich) sichere Verfahren falsch. Aber nicht nur der durchschnittliche Benutzer verwendet solche Verfahren falsch. Auch (angehende) IT-Experten und Firmen machen häufig dieselben Fehler – aus Unwissenheit oder Bequemlichkeit oder weil die genutzten Anwendungen nicht benutzerfreundlich genug sind.

Aus den oben genannten Erfahrungen mit Mitarbeitern, Schülern und Studierenden, welche das CrypTool-Team während vieler Workshops, Unterrichtseinheiten und Vorlesungen sammeln konnte, entstand ab 1998, zunächst in der Deutschen Bank unter der Schirmherrschaft von Bernhard Esslinger, die Lernsoftware CrypTool als internes Projekt [2]. Im Jahr 2000 wurde vom IT-Vorstand die Entscheidung getroffen, CrypTool auch der Allgemeinheit zur Verfügung zu stellen. CrypTool v1 (CT1) wurde zunächst als Freeware zum Herunterladen angeboten, dann 2003 als Open-Source-Software veröffentlicht und von einer Open-Source-Community, zu der inzwischen über 70 Personen weltweit gehören, weiter entwickelt. In CT1 wurden viele kryptographische Verfahren („Wie funktionieren Verschlüsselungen?"), kryptoanalytische Verfahren („Wie bricht man Verschlüsselungen ohne

Kenntnis des geheimen Schlüssels?") und viele Sicherheitsprotokolle („Wie werden kryptographische Verfahren richtig eingesetzt?") implementiert und für den Anwender visuell aufbereitet.

Der Benutzer von CT1 gibt seinen Klartext in ein Fenster ein und wählt dann über die Menüleiste den zu verwendenden Algorithmus aus. Nach der Auswahl öffnet sich ein weiteres Fenster, in dem Parameter für den Algorithmus eingegeben werden können. Danach erscheint ein weiteres Fenster mit dem fertig verarbeiteten Text. Für den Benutzer ist die vollständige Verarbeitung nach dem EVA-Prinzip (Eingabe, Verarbeitung und Ausgabe) nur in seinen einzelnen Schritten ersichtlich, allerdings nicht als Ganzes. Um in CT1 weiteren Text ver- oder entschlüsseln zu können, muss dieses Vorgehen wieder von vorne wiederholt werden. Außerdem ist die Visualisierung der inneren Abläufe einzelner Algorithmen oder Verfahren in CT1 losgelöst von der Durchführung der Algorithmen. So kann ein Benutzer zwar einen Text mit Hilfe des Caesar-Algorithmus ver- und entschlüsseln, die Visualisierung des Caesar-Algorithmus ist allerdings nur beispielhaft als eigenes Fenster realisiert und unabhängig von der Eingabe des Benutzers. CT1 bietet somit seinen Benutzern gute Möglichkeiten, um Texte zu ver- und entschlüsseln oder auch zu analysieren. Eine Kombination oder Verkettung von mehreren Verfahren, z.B. eine Caesar-Verschlüsselung gefolgt von einer Transpositionschiffre (Buchstabenpermutation), ist nur sukzessive möglich. Aus technischer Sicht wurde CT1 monolithisch in C++ mit Hilfe der Microsoft-Foundation-Classes (MFC) entwickelt. Das heißt, dass CT1 aus „einem großen Programm" besteht und Erweiterungen und Fehlerbehebung immer das ganze Programm betreffen.

Aus diesen Gründen (Verfahren nicht als Ganzes ersichtlich, getrennte Visualisierung/Durchführung von Algorithmen, monolithische Architektur sowie mittlerweile veraltete Technik) begann das CrypTool-Team[1] im Jahr 2007 mit der Entwicklung von CrypTool 2.0 (CT2.0), dem offiziellen Nachfolgers von CT1: Zunächst an der Universität Duisburg-Essen am Fachgebiet für „Verteilte Systeme" von Torben Weis unter technischer Leitung von Arno Wacker, und ab Ende 2011 am neugegründeten Fachgebiet für „Angewandte Informationssicherheit" von Arno Wacker an der Universität Kassel. Bei der technischen Konzeption von CT2.0 wurden neue Ideen entwickelt, um die vorher angesprochenen Probleme von vornherein auszuschließen. Zusätzlich wurden von den Hochschulen in Aachen und Koblenz Vorschläge für das Oberflächendesign beigesteuert.

Aktuell (Stand Juli 2014) besteht CT2.0 aus etwas mehr als 390.000 Zeilen Code (Geschrieben in C#, C++ sowie OpenCL [3]. Gezählt wird hierbei nur eigener Code, ohne die inkludierten Bibliotheken wie NTL oder BouncyCastle). Implementiert sind mehr als 200 verschiedene Kryptographieverfahren [4]. Insgesamt haben über 50 freiwillige Mitarbeiter an CT2.0 mitgewirkt. Aktuell gibt es vier aktive Kernentwickler, die CT2.0 pflegen. Regelmäßig werden Studierendenprojekte und Abschlussarbeiten unter anderem an den Universitäten Kassel, Bochum, Mannheim und Siegen durchgeführt, die CT2.0 mit neuen Funktionen erweitern. Im Jahr 2013 wurde CT2.0 insgesamt über 40.000 Mal von der CT2.0-Webseite [6] heruntergeladen (ca. 60 % davon luden die englische Version herunter).

1 Neben CT2.0 wurden im CrypTool-Projekt auch die Software-Projekte JCT und CTO ins Leben gerufen. JCT ist eine Plattform-unabhängige Eclipse-Java-Anwendung. CTO ist eine Browser-basierte Anwendung. Beide werden ebenfalls aktiv weiter entwickelt. Bei CT1 finden dagegen nur Pflege und Fehlerbehebung und eine Portierung ins Französische statt.

2 Design-Kriterien von CrypTool 2.0

Die erste grundlegende Neuerung in CT2.0 war der Umstieg auf eine .NET, C# und Windows-Presentation-Foundation (WPF) [7] basierte Plattform. Hierdurch konnte für CT2.0 gleich von Beginn an eine moderne Oberfläche entwickelt werden, wie sie z.B. alle Microsoft-Programme ab Office 2007 besitzen. Die Idee hierbei war, dass sich Benutzer, die die Bedienung von Microsoft-Anwendungen gewohnt sind, gleich in CT2.0 zurechtfinden.

Die nächste grundlegende Anforderung an CT2.0 war der Umstieg von einer manuellen Verarbeitung zu einer automatisierten und interaktiven Verarbeitung. Konkret soll es für den Benutzer in CT2.0 möglich sein, einen Text einzugeben und „live" verfolgen zu können, wie das jeweilige Verfahren diesen ver- bzw. entschlüsselt. Diese Anforderung macht das Fensterkonzept, wie es noch in CT1 verfolgt wurde, für CT2.0 unmöglich. Die Lösung für diese grundsätzlich neue Anforderung ist die Entwicklung einer grafischen „Programmiersprache", die alle Algorithmen und Verfahren als sogenannte Komponenten visualisiert, die in (nahezu) beliebiger Art und Weise miteinander kombiniert werden können. Diese visuelle Programmierung wurde mit dem sogenannten *Arbeitsplatz-Manager* umgesetzt, auf den in Kapitel 3.1 genauer eingegangen wird.

Eine weitere Anforderung an CT2.0 ist die einfache Erweiterbarkeit. Es soll sowohl CrypTool-Projektmitgliedern als auch Außenstehenden einfach möglich sein, neue Algorithmen und Verfahren für CT2.0 zu entwickeln und CT2.0 als Framework für eigene, notfalls auch nicht-öffentliche Entwicklungen zu benutzen. Dafür wurde von Beginn an eine komponentenbasierte Architektur mit knappen Schnittstellen entwickelt, die eine einfache Erweiterung unabhängig vom „Kern" von CT2.0 ermöglicht. Weitere CT2.0-Komponenten können einfach als .NET-Assembly (eine .NET-Programmbibliothek) in das CT2.0-Verzeichnis gelegt und geladen werden. Die Assemblies, die die Kernkomponenten von CT2.0 beinhalten, bleiben dabei unangetastet. Auch die eigenen Assemblies profitieren über diese Schnittstellen von allen Funktionen des CT2.0-Frameworks, z.B. Auflistung im Startcenter, in den Suchfeldern oder in der Online-Hilfe.

Die Trennung von Algorithmus-Ausführung und Algorithmus-Visualisierung, wie es in CT1 der Fall ist, ist in CT2.0 nicht mehr vorhanden. Jede Komponente, die einen Algorithmus für CT2.0 implementiert, kann über eine sogenannte *Präsentation*, d.h. grafische Visualisierung des Algorithmus, verfügen. Diese Präsentationen können in der visuellen Programmierung von CT2.0 einfach eingeblendet werden, und visualisieren den inneren Ablauf wichtiger kryptographischer Algorithmen direkt innerhalb des Arbeitsplatz-Managers. Ein Beispiel für eine Präsentation von einem einfachen modernen symmetrischen Verschlüsselungsverfahren (SDES [8]) ist in Abb. 1 abgebildet.

Evaluierungen bei den CT2.0-Nutzern ergaben, dass Informatik-affine Benutzer sehr gut mit der visuellen Programmierung innerhalb von CT2.0 zurechtkommen, jedoch gerade Anfänger oder Benutzer ohne Programmiererfahrung Probleme mit der Konstruktion von visuellen Programmen haben. Um auch Anfängern den Einstieg in die CT2.0-Bedienung und in die Kryptologie zu erleichtern, ist CT2.0 mit einer Vielzahl von fertigen Workflows, sogenannten *Vorlagen* ausgestattet, die in Kapitel 3.2 näher beschrieben werden.

Für Umsteiger von CT1 und für Einsteiger, die nur „schnell" etwas ver- oder entschlüsseln möchten, ist der *Wizard* für CT2.0

entwickelt worden. Im Wizard kann man zunächst das Verfahren auswählen, danach einfach einen Text eingeben und abschließend das Ergebnis erhalten. Die Funktionsweise des Wizards ist im Kapitel 3.3 dieses Artikels genauer beschrieben.

Eine abschließende Anforderung ist die Integration einer HTML-Hilfe, die jede Komponente von CT2.0 textuell beschreibt und dem Benutzer die Funktionalität sowie die Benutzung verdeutlicht.

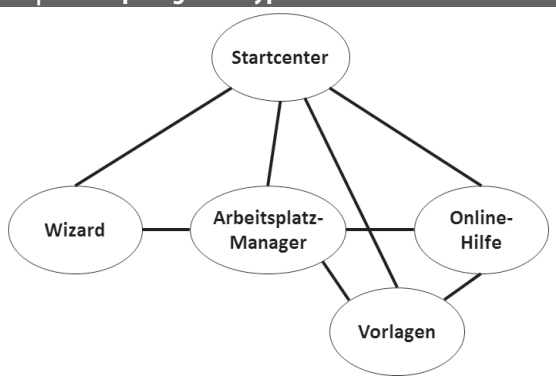
3 Die CrypTool-2.0-Komponenten

CrypTool 2.0 (CT2.0) besteht aus folgenden fünf Kernbestandteilen:

1. *Arbeitsplatz-Manager*,
2. *Startcenter,*
3. die (mit ausgelieferten)*Vorlagen*,
4. *Wizard* und
5. *Online-Hilfe*.

In Abb. 2 ist dargestellt, wie die einzelnen Kernbestandteile in ihrer Benutzung miteinander verknüpft sind. Außerdem existieren über 140 weitere, von verschiedenen Entwicklern entwickelte Komponenten, die kryptographische, kryptoanalytische oder Hilfsverfahren implementieren. Diese Komponenten lassen sich aus allen Kernbestandteilen (außer der Online-Hilfe) heraus aufrufen. Im Folgenden werden die ersten vier Kernbestandteile erläutert und abschließend in Kapitel 3.4 ausgesuchte Beispiel-Komponenten präsentiert.

Abb. 2 | Verknüpfung der CrypTool-2.0-Kernbestandteile



3.1 Der Arbeitsplatz-Manager

Der *Arbeitsplatz-Manager* ist der wichtigste Kernbestandteil von CT2.0, da er die grafische Programmiersprache von CT2.0 beinhaltet. Er ermöglicht dem Benutzer, die sogenannten Komponenten auf der Arbeitsfläche zu platzieren und diese mittels Verbindungslinien miteinander zu verknüpfen. Hierfür können Komponenten mittels Drag&Drop aus einem Menü ausgewählt und auf die Arbeitsfläche gelegt werden. Komponenten werden als Icons dargestellt, die über verschiedene Ein- und Ausgänge, die Konnektoren, verfügen. Jeder Konnektor besitzt einen eigenen Datentyp (String, Byte Array, Integer, BigInteger, Datenstrom und viele weitere), der durch seine Farbe identifiziert werden kann. Konnektoren mit gleicher Farbe können einfach mittels gedrückter Maustaste, was die Erstellung einer Verbindungslinie auslöst, miteinander verbunden werden. Auch die Verbindung von Datentypen unterschiedlicher Art ist möglich. Diese werden einerseits automatisch implizit konvertiert, andererseits müssen spezielle Konverter-Komponenten genutzt werden, um „inkompatible" Datentypen explizit ineinander umzuwandeln. Ob Datentypen implizit ineinander konvertiert werden können, zeigt der Arbeitsplatz-Manager mittels eines gelben Tooltip-Fensters auf dem Konnektor an. Verbindungen, die problemlos möglich sind, werden durch einen grünen Tooltip angezeigt.

Abb. 1 | Beispiel für eine Präsentation in CrypTool 2.0

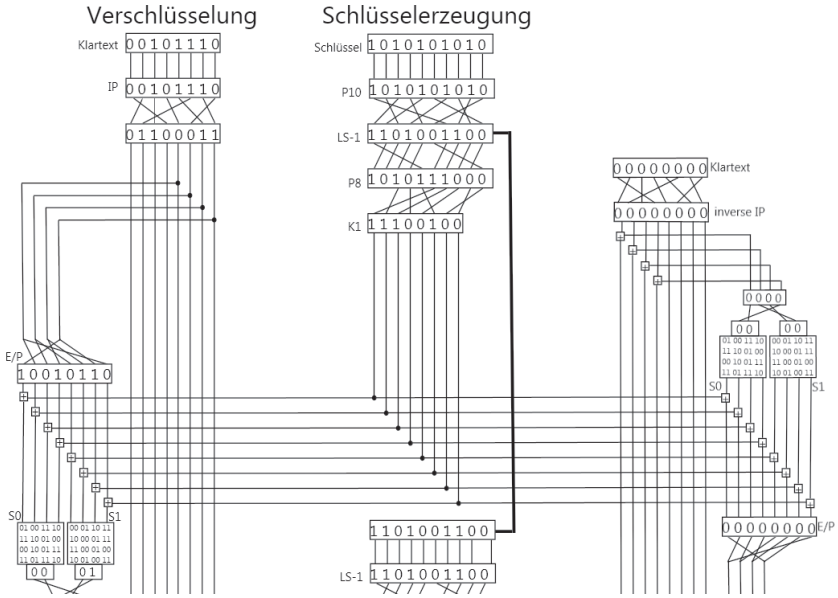
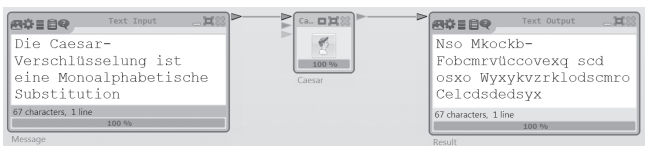


Abb. 3 | Caesar-Chiffre auf einem CrypTool-2.0-Arbeitsplatz



Dank des Arbeitsplatz-Managers kann eine einfache Caesar-Verschlüsselung, wie in Abb. 3 dargestellt, schon mit nur drei Einzel-Komponenten erstellt werden.[2] Hierzu werden je eine Texteingabe-, Caesar- und Textausgabe-Komponente auf die Arbeitsfläche platziert. Abschließend werden die Texteingabe mit dem Caesar und der Caesar mit der Textausgabe verbunden.

Der Schlüssel für die Caesar-Verschlüsselung kann über eine Parameterleiste eingegeben werden. Zu jeder Einzelkomponente auf dem Arbeitsplatz kann man sich rechts eine Parameterleiste einblenden lassen (siehe Abb. 4). Alternativ kann jede Komponente auf die gesamte Arbeitsfläche maximiert und die Parameter können direkt in der Komponente eingestellt werden.

Nun kann das erstellte visuelle Programm ausgeführt werden. Diese Ausführung der Vorlagen wird von der Ausführungsmaschi-

2 Ein Video von Arno Wacker zur Einführung in die Benutzung von CrypTool 1 und CrypTool 2.0 kann man auf YouTube [25] ansehen.

Abb. 4 | Eingeblendete Parameterleiste für die Caesar-Komponente

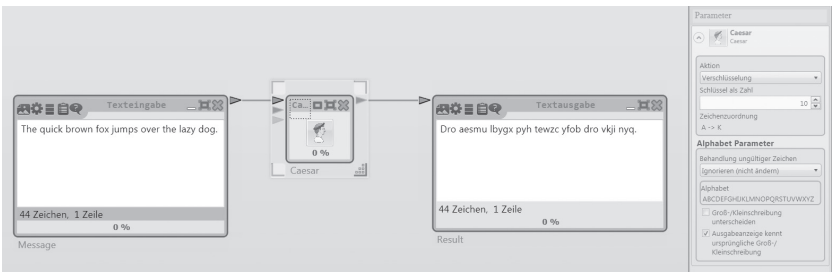


Abb. 5 | Das CrypTool-2.0-Startcenter

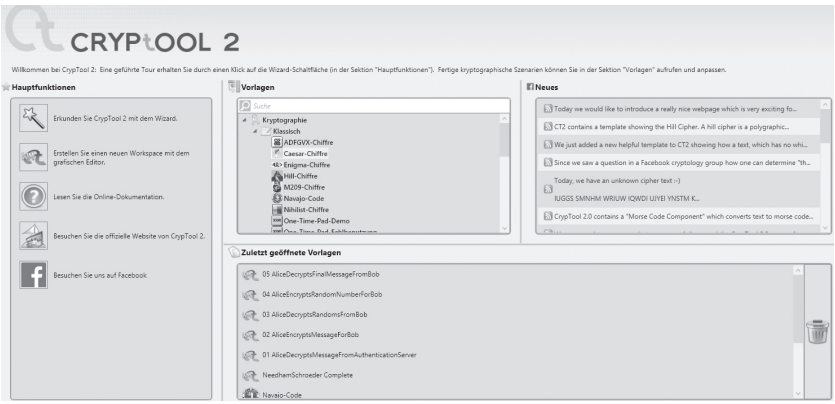


Abb. 6 | Das BB84-Protokoll visualisiert in CrypTool 2.0

ne im Arbeitsplatz-Manager realisiert. Dafür muss der Benutzer nur auf einen Abspiel-Button (Play) klicken, der sich oben links in der Ribbon-Leiste von CT2.0 befindet. Gibt der Benutzer nun einen Text in die Texteingabe-Komponente ein, so wird dieser zu der Caesar-Komponente geleitet. Diese wiederum verschlüsselt den übergebenen Text und leitet diesen an die Textausgabe-Komponente weiter. Die Textausgabe-Komponente zeigt daraufhin den verschlüsselten Text an. Hierbei kann der Benutzer schon beim Drücken der ersten Taste verfolgen, wie der ausgegebene, verschlüsselte Text live erstellt wird. Außerdem ist es möglich, die aktuellen Daten eines jeden Konnektors aller Komponenten, mittels Tooltip-Text, anzusehen. Die automatische Live-Verarbeitung ist mit nahezu jeder CT2.0-Komponente möglich. Bei der Ausführung der einzelnen Komponenten setzt der Arbeitsplatz-Manager mehrere Threads ein, um so die Verarbeitung innerhalb der Arbeitsfläche zu parallelisieren und zu beschleunigen.

Da selbst einfache visuelle Programme schnell recht viel Platz auf der Arbeitsfläche einnehmen, kann man diese einfach vergrößern. Mit den Tasten F11 und F12 lässt sich die Arbeitsfläche so erweitern, dass von der „Umgebung" nur noch die Titelleiste, das Menü und die Leiste für die Arbeitsflächen-Reiter übrig bleiben. Die „Entwicklungsumgebung" von CT2.0 wird dadurch schnell zur Ausführungsdarstellung.

3.2 Das Startcenter und die Vorlagen

Beim Start von CT2.0 erscheint als Erstes das Startcenter (siehe Abb. 5). Vom Startcenter aus kann der Benutzer einfach auf die weiteren Kernbestandteile und alle Einzel-Komponenten von CT2.0 zugreifen. Außerdem bietet das Startcenter eine Übersicht aller *Vorlagen*, die CT2.0 beigefügt sind.

Eine Vorlage ist ein, von den CT2.0-Entwicklern erstelltes, visuelles Programm, das eine oder mehrere kryptographische oder kryptoanalytische Komponenten von CT2.0 und deren Handhabung demonstriert. So verfügt CT2.0 über mehr als 150 Vorlagen, die in unterschiedliche Kategorien aufgeteilt sind. Neben den Kategorien „Kryptographie" und „Kryptoanalyse" gibt es „Hash-Funktionen", „Mathematik", „Codes", „Protokolle", „Steganographie" und „Werkzeuge".

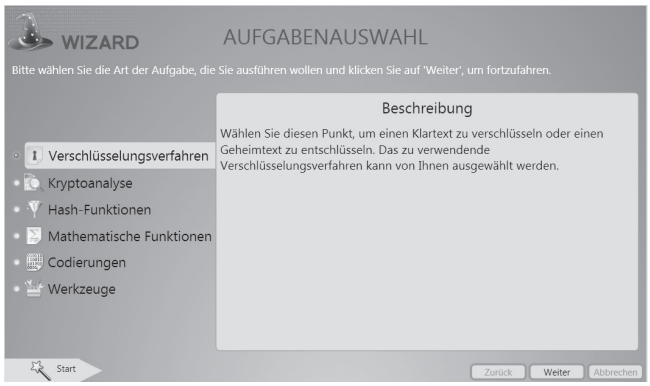
Ein Beispiel für eine Vorlage, welche ein modernes kryptographisches Protokoll visualisiert, ist das in Abb. 6 dargestellte BB84-Quantenkryptographie-Protokoll [9]. Das BB84-Protokoll dient dem sicheren Schlüsselaustausch und wurde 1984 von Charles Bennett und Gilles Brassard entwickelt. Die CT2.0-Vorlage visualisiert die „Übertragung" von Quanten mittels verschiedener Animationen und verdeutlicht, wie ein Schlüssel mithilfe dieses Protokolls vom Sender (Alice) zum Empfänger (Bob) sicher übertragen werden kann.

Des Weiteren bietet das Startcenter eine Übersicht über wichtige Neuerungen in CT2.0. Diese werden regelmäßig auf der CT2.0-Facebook-Seite [10] veröffentlicht und sind als RSS-Feed in Form von News innerhalb des Startcenters einsehbar.

3.3 Der Wizard

Die einfachste Möglichkeit, um als Anfänger mit CT2.0 kryptographische oder kryptoanalytische Verfahren zu nutzen, bietet der *Wizard* (Abb. 7). Der Wizard ist, ähnlich wie die Vorlagen, in

Abb. 7 | Der Wizard von CrypTool 2.0



verschiedene Themenbereiche gegliedert, wie z.B. klassische und moderne Verschlüsselungsverfahren. Der Benutzer wird von einer Übersichtsseite des Wizards zur nächsten geleitet, indem er im linken Menü des Wizards immer präziser auswählt, was er tun möchte. Der Wizard bzw. die enthaltenen Themen sind baumartig strukturiert. Um z.B. einen Text mit der monoalphabetischen Substitution (Buchstabenersetzung) zu verschlüsseln, wählt der Benutzer zunächst „Verschlüsselungsverfahren" aus. Diese präzisiert er mit der Auswahl von „Klassische Verschlüsselungsverfahren", in denen er dann die „Substitution" auswählen kann. Jedes Verfahren besitzt eine eigene Seite für die Parameter, die zumeist die vorletzte Auswahlseite im Wizard ist. Hier kann der Benutzer nun das konkrete Verfahren parametrisieren, z.B. den geheimen Schlüssel auswählen.

Die letzte Seite des Wizards zeigt dann das Ergebnis des ausgewählten Verfahrens. Auch im Wizard ist es in vielen Fällen möglich, Text einzugeben und gleichzeitig das Ergebnis der Verschlüsselung quasi in Echtzeit zu betrachten. Da der Wizard im Hintergrund dieselbe Ausführungsmaschine wie der Arbeitsplatz-Manager nutzt, kann man auf der letzten Seite des Wizards auch direkt zu einer dazu passenden Vorlage im Arbeitsplatz-Manager wechseln. Der Nutzer kann dadurch erkennen mit welchem visuellen Programm der Wizard die Aktionen (z.B. Verschlüsselung) durchgeführt hat.

3.4 Beispiel-Komponenten und Verfahren

Die CT2.0-Entwickler sind bemüht, die besten und neuesten Verfahren der Kryptographie und Kryptoanalyse in CT2.0 zu integrieren.

So verfügt CT2.0 z.B. im Bereich der klassischen Kryptographie über sehr schnelle Verfahren zum Brechen der monoalphabetischen Substitution und der einfachen Spaltentransposition, sowie über ein sehr schnelles Analyseverfahren für die Rotormaschine Enigma, das von James J. Gillogly [11] entwickelt wurde. Aktuell wurde ein, von George Lasry [12] entwickeltes, sehr schnelles Analyseverfahren für die doppelte Spaltentransposition („Doppelwürfel") von dem Kernentwickler Armin Krauß in CT2.0 implementiert. Es bietet die Möglichkeit, Geheimtexte, die mittels des Doppelwürfels verschlüsselt sind und bisher als „unknackbar" galten, ohne Kenntnis der geheimen Schlüssel, zu brechen.[3]

Im Bereich der modernen Kryptographie sind unter anderem Brute-force-Verfahren (d.h. vollständige Schlüsselsuche) in CT2.0 effizient implementiert. So können mit dem „Schlüsselsucher" reduzierte Suchräume, sowohl des Advanced-Encryption-Standards (AES) [13] als auch des Data-Encryption-Standards (DES) [14], mittels Grafikkarte und OpenCL sehr schnell durchsucht werden. Auf einem Notebook mit einem Intel i7-Prozessor mit 8 Kernen zu je 2,4 GHz und einer Nvidia Quadro 1000M schafft der Schlüsselsucher ca. drei Millionen AES-Schlüssel pro Sekunde.

Für die Analyse moderner asymmetrischer Verfahren enthält CT2.0 ein sehr schnelles Verfahren für die Faktorisierung von sehr großen Zahlen, das auf der Open-Source-Implementierung „Msieve" [15] basiert.

CT2.0 bietet außerdem Analysemöglichkeiten für moderne Protokolle. Ein Beispiel hierfür ist der Angriff auf das (unsichere) Wired-Equivalent-Privacy-Protokoll (WEP) [16], welcher das Kennwort eines Access-Points mit WEP innerhalb weniger Sekunden bricht. Andere moderne Protokolle wie Coin-Flipping, Oblivious Transfer, Yao's Millionärs-Problem und die Dining-Philosophers-Frage (Abb. 8) sind ebenfalls in CT2.0 als Vorlagen eingepflegt.

Im Bereich der kryptographischen Hashfunktionen ist die CT2.0-Vorlage zum Auffinden von Kollisionen für die bekannte und weitverbreitete (unsichere) Hashfunktion Message-Digest 5 (MD5) [17] verfügbar. Der hierfür genutzte Algorithmus findet Kollisionen (d.h. unterschiedliche Nachrichten mit dem gleichen Hash-Wert) schon innerhalb weniger Sekunden.

Auch praktische Verfahren, wie eine sichere Verschlüsselung mittels Passwort, für die Kommunikation mit Freunden und Bekannten, können mit CT2.0 genutzt werden. Eine spezielle Vorlage ermöglicht die sichere Verschlüsselung von Texten, die dann z.B. in eine E-Mail kopiert werden können. Diese Texte können dann beim Empfänger wieder mittels CT2.0 entschlüsselt werden (sofern er im Besitz des richtigen Passworts ist). Darüber hinaus kann man die Android-App CrypDroid [18] nutzen, um die gleichen sicheren Nachrichten mittels Smartphone zu ver- und entschlüsseln.

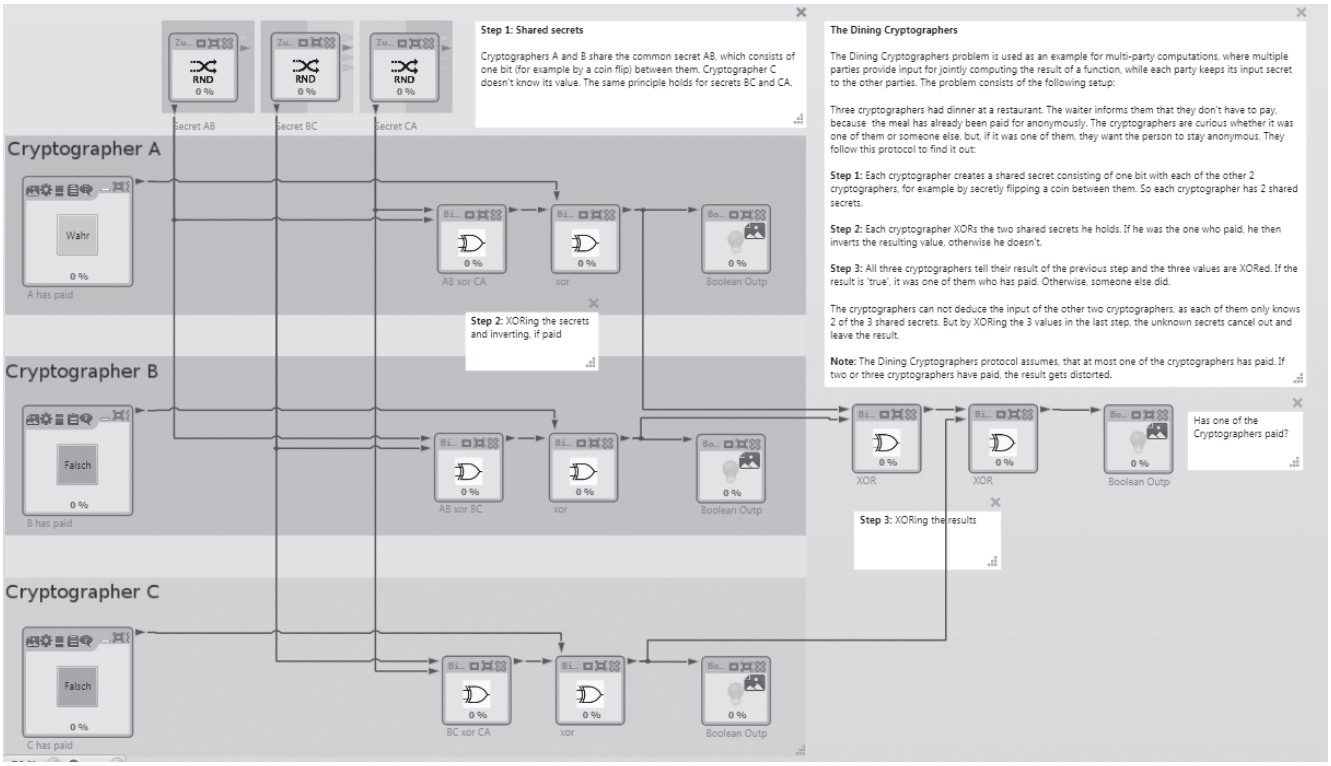
Aktuelle Ereignisse im Bereich der IT-Sicherheit oder der Kryptographie finden schnell Einzug in CT2.0. So gibt es in CT2.0 die Möglichkeit, den Heartbleed-Bug [19] zu analysieren, der im April 2014 zu einer großflächigen Bedrohung des Internets führte. In Abb. 9 ist die Vorlage mit der Analyse abgebildet. Hier kann ein CT2.0-Benutzer eine beliebige Adresse eines Servers eingeben und überprüfen, ob dieser sicher oder unsicher ist.

Beim Heartbleed-Bug führte ein manipuliertes Paket im Transport-Layer-Security-Protokoll (TLS) [20] sowie ein Buffer-Overflow dazu, dass Webserver, die OpenSSL einsetzten, einem Angreifer große Teile ihres Programmspeichers zurücklieferten. Angreifern war somit das Auslesen von Passwörtern und Zertifikaten möglich. Dies kann mit der Vorlage in CT2.0 und einem angreifbaren Webserver [21], der vom CT2.0-Team bereitgestellt wird, vom Benutzer nachgestellt werden.

Gerade die Visualisierung von Angriffen und Verfahren ist ein wichtiger Bestandteil von CT2.0. So kann mittels grafischer Beispiele und Vorlagen relativ einfach gezeigt werden, warum Verkettungsmodi von modernen Verschlüsselungsverfahren, trotz gutem, zugrunde liegenden Algorithmus, Informationen durch

3 Die entsprechende Doppelwürfel-Aufgabe war eine auf Otto Leiberich zurückgehende Level-X-Challenge von Klaus Schmeh in dem internationalen Krypto-Wettbewerb MysteryTwister C3 (MTC3) [24]. Die Software-Projekte von CrypTool und der MTC3-Wettbewerb befruchten sich gegenseitig sehr positiv (neue Lösungsverfahren aufgrund der Challenges fließen in CT2.0 ein; MTC3-Autoren nutzen CT2.0 und MTC3 für Lehre und Übungen).

Abb. 8 | Wer zahlte? – Das Dining-Cryptographers-Problem



die Verschlüsselung „hindurch scheinen" lassen. Ein Verkettungsmodus oder Blockmodus verknüpft Blöcke von Block-Chiffren, damit man mit diesen auch Daten verschlüsseln kann, die länger als die Blockgröße der Chiffre sind. Das sogenannte Electronic-Code-Book-Verfahren (ECB) ist unsicherer, da gleiche Klartextblöcke auf gleiche Geheimtextblöcke abgebildet werden (siehe in Abb. 10 oben rechts den noch sichtbaren Smiley). Dagegen ist das Cipher-Block-Chaining (CBC) in diesem Fall sicher, da jeder Geheimtextblock mit seinem Vorgänger verknüpft wird und so jeder Block anders verschlüsselt ist (siehe in Abb. 10 unten rechts das verrauschte Bild). Diese Eigenschaft ist dabei unabhängig vom eingesetzten Verschlüsselungsverfahren und stellt damit ein Beispiel dar, wie es zu einer unsicheren Übertragung von Daten kommen kann, trotz der Verwendung eines sicheren Verschlüsselungsalgorithmus. Dies kann mit der abgebildeten CT2.0-Vorlage auch Anfängern leicht vermittelt werden.

4 Einsatz, Wartung und Aktualität

CrypTool 2.0 (CT2.0) wird sowohl in der Lehre als auch im Selbststudium eingesetzt. Zum Beispiel wird CT2.0 in Universitätskursen (z.B. „Grundlagen der Kryptologie" oder „Sicherheit in Kommunikationsnetzen" an der Universität Kassel) eingesetzt. Zum anderen werden über das Jahr verteilt verschiedene „Schülerkryptos" [22] (unter anderem in Kassel, Siegen und Kelkheim bei Frankfurt) durchgeführt. Ziel der Schülerkryptos ist es, Oberstufenschüler für die Welt der Kryptologie zu begeistern und für ein Studium in den MINT-Fächern (Mathematik, Informatik, Naturwissenschaften und Technik) zu motivieren. Innerhalb dieser Schülerkryptos werden Aufgaben ausgegeben, die mit Hilfe von CT2.0 zu lösen sind. Für die schnellsten Lösungen bekommen die Schüler Preise, z.B. Bücher und DVDs. Zeitgleich sind die Schülerkryptos eine Plattform, um direkt von Schülern Feedback bezüglich der Benutzbarkeit von CT2.0 einzuholen. So konnte CT2.0 gerade durch diese Veranstaltungen stetig verbessert werden. Aussagekräftig sind die Ergebnisse von zwei Fragen, die bei der Schülerkrypto 2014 in Kassel 46 Schülerinnen und Schülern gestellt worden sind. So sind 95% aller Schüler der Meinung, dass CT2.0 anwenderfreundlich gestaltet ist. Außerdem fanden 83% aller Schüler, dass CT2.0 ihnen mindestens weitestgehend geholfen hat, Kryptologie besser zu verstehen. Diese Ergebnisse zeigen, dass CT2.0 sein Ziel erfüllt, und motivieren, die Arbeit an CT2.0 weiter zu führen.

Um CT2.0 für den „Heimgebrauch" nutzbar zu machen und die Verbreitung der Software zu fördern, verfügt CT2.0 über einen automatischen und einfachen Updatemechanismus, der die drei verschiedenen CT2.0-Versions-Arten („Nightly Builds", Betas, Releases) automatisch aktualisieren kann. In den Nightly Builds erreichen neue Funktionen und Fehlerkorrekturen die Benutzer schon innerhalb eines Tages. Um die Qualität neuer Funktionen zu gewährleisten, setzt das CT2.0-Team auf einen leichtgewichtigen Entwicklungsprozess. So werden neue Komponenten zunächst als „experimentell" markiert und sind zunächst auch noch nicht im Nightly Build verfügbar. Sofern die Komponenten einen ausreichenden Grad der Qualität erreicht haben (fehlerfreie Funktionalität, zumindest englische Hilfe vorhanden, mindestens eine Vorlage verfügbar und, falls möglich, ein Unit-Test vorhanden) werden diese von einem Kernentwickler vom „experimentellen" Status in den Nightly Build aufgenommen.

Ab diesem Zeitpunkt sind diese Komponenten für jeden CT2.0-Nutzer sichtbar. Damit Fehler von Benutzern einfach gemeldet werden können, verfügt CT2.0 außerdem über einen Report-Mechanismus, der das Einstellen von Tickets ermöglicht. Sollte CT2.0 trotzdem einmal abstürzen, kann der Benutzer, sofern er

Abb. 9 | Heartbleed-Angriff in CrypTool 2.0

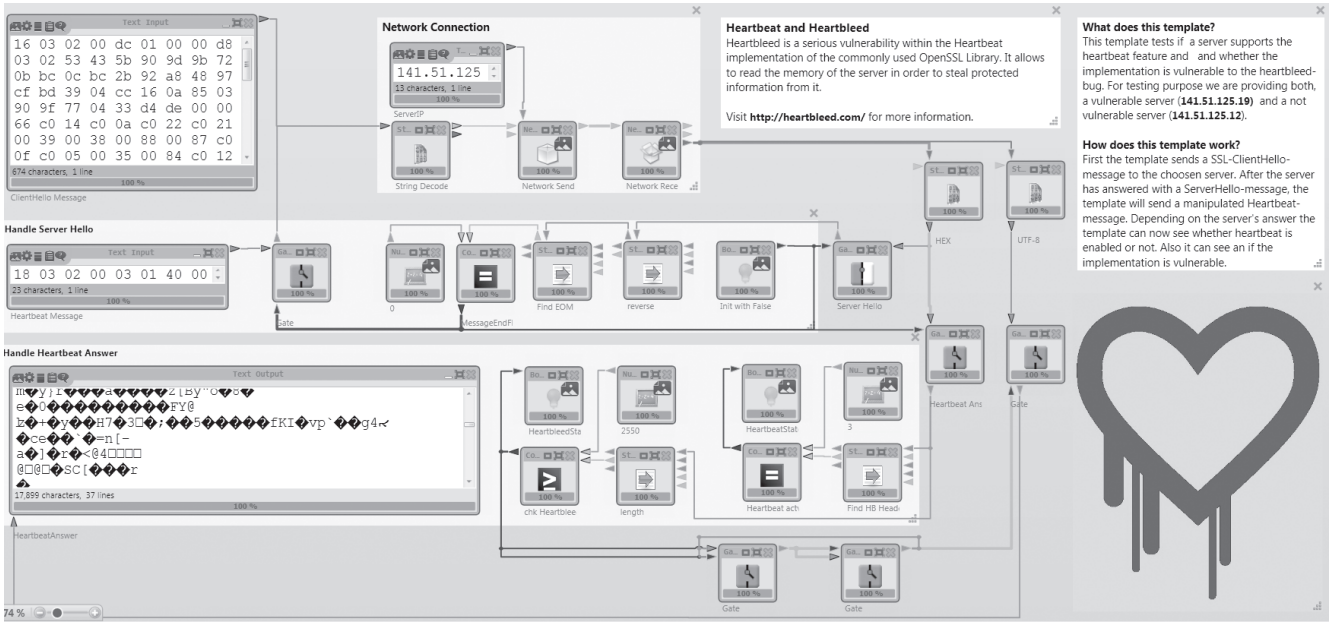
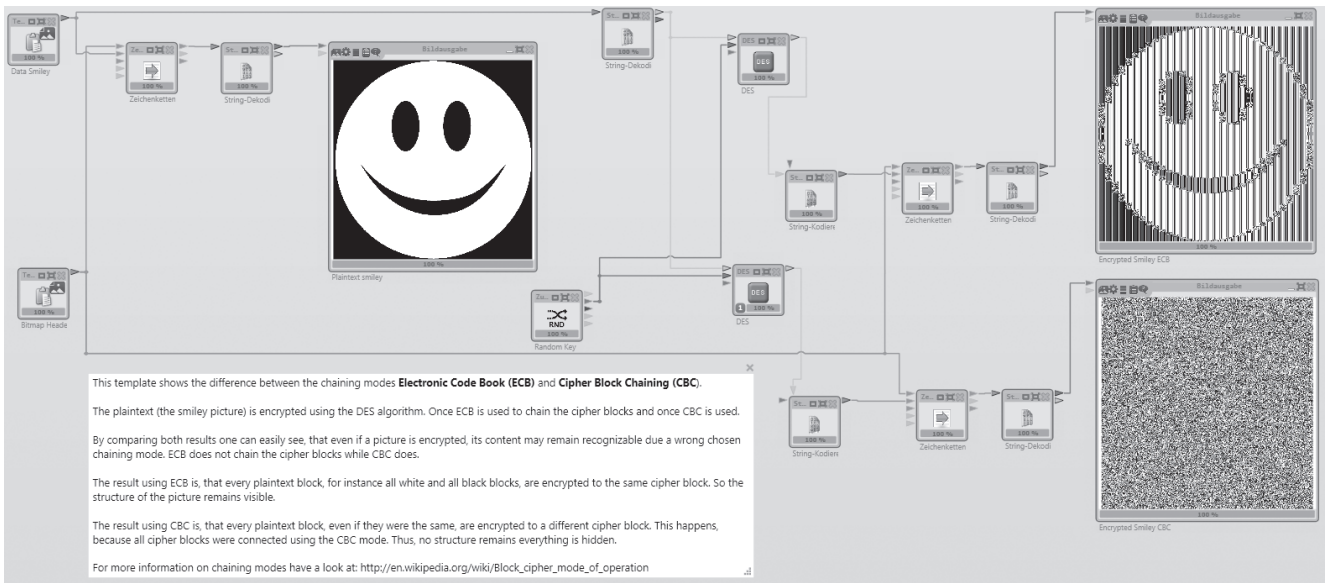


Abb. 10 | Darstellung von Blockmodi in CrypTool 2.0



dies wünscht, einen automatisch generierten Fehlerbericht an das CT2.0-Ticketsystem schicken lassen. Hier wird automatisch ein sogenannter „Crash-Report" erstellt, der detaillierte Fehlerbeschreibungen für das CT2.0-Team beinhaltet. So gehen aufgetretene Fehler nicht „einfach verloren", sondern können zeitnah und effizient behoben werden. Die Entwicklung selbst erfolgt verteilt mit Hilfe des Source-Code-Verwaltungssystems Subversion. Studierende, die länger in diesem Open-Source-Projekt mitarbeiten, lernen all die modernen Techniken und Softskills, die man auch im Berufsleben als Software-Entwickler und Software-Architekt braucht (das Feedback ehemaliger Diplomanden ergab, dass sich das auch gleich in ihren Bewerbungen positiv auswirkte).

5 Fazit und Ausblick

Die Open-Source-Software CrypTool 2.0 (CT2.0) bietet sowohl Schülern als auch Studierenden und Kryptologie-Begeisterten eine Möglichkeit, um Kryptologie und ihre Verfahren zu erleben und zu erlernen. Ob als Mittel innerhalb von Vorlesungen oder im Selbststudium, CT2.0 bietet ausreichend Vorlagen, die Kryptographie und Kryptoanalyse erlebbar und erlernbar machen. Ebenso erhöht CT2.0 in Schulungen von Mitarbeitern in Firmen und Behörden deren Verständnis, da sie spielerisch experimentieren können. Nach siebenjähriger Entwicklungszeit ist die Software an einem Punkt angekommen, den das CT2.0-Team als „erstes Release" bezeichnen kann. CT2.0 ist für ein erstes Release „rund", zweisprachig und „Feature complete", wenn man dies für eine Software, die sich mit Kryptologie und IT-Sicherheit beschäftigt, sagen kann.

Natürlich ist nach dem Release das Projekt „CrypTool 2.0" noch lange nicht abgeschlossen. In den Köpfen der CT2.0-Entwickler und ihrer Community existiert eine Vielzahl an Ideen und Wünschen. So ist für die CrypTool-2.1-Version geplant, Programminstanzen stärker miteinander vernetzbar zu machen. Konkret soll das Volunteer-Computing [23] Einzug in die CrypTool-2.1-Welt finden. Als Volunteer-Computing bezeichnet man die Möglichkeit, die Rechner von vielen Freiwilligen miteinander vernetzen zu können, um so die Rechenkraft aller Rechner zu vereinen und eine Art „Supercomputer" zu bilden. Dies soll ermöglichen, dass mit Hilfe von CrypTool 2.1 sowohl die verteilte Faktorisierung als auch die verteilte Schlüsselsuche möglich ist.

Ein weiterer Meilenstein im zukünftigen CrypTool 2.1 ist die Bildverarbeitung. So soll CrypTool 2.1 die Möglichkeit bieten, klassische Chiffren einfacher zu transkribieren (Transkription = Umwandlung von unleserlichen Buchstaben und Zeichen in ein Maschinen-lesbares Format) und zu analysieren. Dies hilft dann auch Historikern, die z.B. Texte aus dem Spanischen Bürgerkrieg mit der vor kurzem implementierten „Spanish-Strip-Cipher" entschlüsseln wollen. Darüber hinaus sollen auch moderne Bild-Hashes, sogenannte robuste Hash-Verfahren [5], integriert werden.

Des Weiteren ist das Projektteam bestrebt, alle (wichtigen) Neuerungen, die sich sowohl in klassischer als auch moderner Kryptologie ergeben, in zukünftige CrypTool-2-Versionen einfließen zu lassen. Bis zum Release der CrypTool-2.1-Version sind wieder mehrere Beta-Versionen geplant, die aktuelle stabile Zwischenstände der Entwicklung darstellen. Außerdem ist weiterhin jede Nacht ein Nightly Build verfügbar, das die neuesten Entwicklungen enthält.

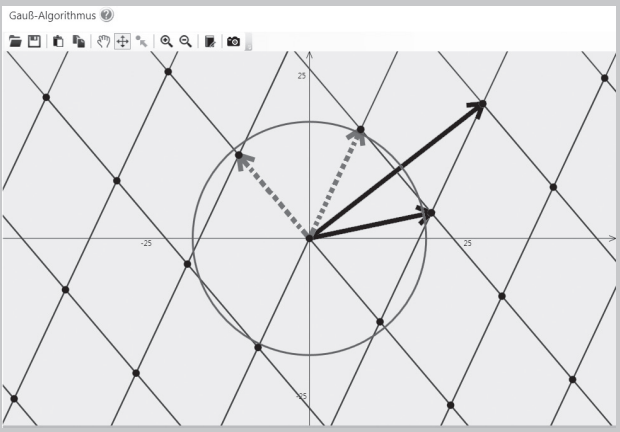
Unterstützung für dieses Open-Source-Projekt ist sehr willkommen: Benötigt werden weitere freiwillige Entwickler und Tester, Lehrstühle und Fachgebiete (die ihre spezifische Kompetenz einbringen und entsprechende Arbeiten zur Weiterentwicklung von CT2 vergeben) und Sponsoren (die z.B. die Infrastruktur mit finanzieren).

Krypto-Tutorien

Anhand der im Artikel beschriebenen CT2-Philosophie können Komponenten im Arbeitsplatz-Manager beliebig kombiniert werden, um Workflows darzustellen. Zusätzlich wurde mit dem Menü „Kryptotutorien" die Möglichkeit geschaffen, einzelne Themen in einem größeren Zusammenhang darzustellen bzw. mit mehr didaktischer Führung zu versehen. Momentan sind darin drei Einträge, die in das jeweilige Thema mit sehr vielen interaktiven Visualisierungen einführen:

- Die Welt der Primzahlen
- Gitterbasierte Kryptographie
- Angriff auf PKCS#1

Abb. 11 | Visualisierung des Algorithmus von Gauß in der Einführung zu Gittern



Literatur

[1] B. Esslinger, „Sichere E-Mail mit S/MIME – Eine Anleitung aus Anwenderperspektive", *Datenschutz und Datensicherheit (DuD),* Bd. 38, Nr. 5, pp. 305-313, 2014.

[2] B. Esslinger, „CrypTool – Ein Open-Source-Projekt in der Praxis", *Datenschutz und Datensicherheit-DuD,* Bd. 33, Nr. 3, p. 167–173, 2009.

[3] Khronos OpenCL Working Group, „The OpenCL Specification", 2009.

[4] CrypTool 2.0 Team, „Kryptologische Funktionen in verschiedenen CrypTool-Versionen", 2014, http://www.cryptool.org/de/ctpdokumentation-de/ctp-functions-de, abgerufen am 28.06.2014.

[5] F. Jiri, M. Goljan, „Robust Hash Functions for Digital Watermarking", in *International Conference on Information Technology: Coding and Computing*, 2000.

[6] CrypTool 2.0 Team, „CrypTool Portal – Cryptography for Everybody", 2014, http://www.cryptool.org/, abgerufen am 28.06.2014.

[7] A. Nathan, „Windows Presentation Foundation Unleashed", Pearson Education, 2006.

[8] E. Schaefer, „A Simplified Data Encryption Standard Algorithm", *Cryptologia,* Bd. 20, Nr. 1, pp. 77-84, 1996.

[9] G. B. C. Bennett, „BB84", in *IEEE International Conference on Computers, Systems, and Signal Processing*, Los Alamitos, 1984.

[10] N. Kopal, „CrypTool 2.0 Facebook-Seite", 2014, https://de.facebook.com/CrypTool20, abgerufen am 28.06.2014.

[11] J. J. Gillogly, „Ciphertext-Only Cryptanalysis of Enigma", *Cryptologia,* Bd. 19, Nr. 4, pp. 404-413, 1995.

[12] G. Lasry, N. Kopal, A. Wacker, „Solving the Double Transposition Challenge with a Divide-and-Conquer Approach", *Cryptologia,* Bd. 38, Nr. 3, pp. 197-214, 2014.

[13] National Institute for Standards NIST, „Advanced Encryption Standard", 2001.

[14] National Institute for Standards NIST, „Data Encryption Standard (DES)", 1999.

[15] J. Papadopoulos, „MSieve", 2014, http://sourceforge.net/projects/msieve/, abgerufen am 28.06.2014.

[16] IEEE Computer Society LAN MAN Standards Committee, „Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications", 1997.

[17] R. Rivest, „The MD5 Message-Digest Algorithm", 1992.

[18] ZweiPunktFuenf, „CrypDroid", 2014, https://play.google.com/store/apps/details?id=de.zweipunktfuenf.crypdroid&hl=de, abgerufen am 28.06.2014.

[19] CODENOMICON, „The Heartbleed Bug", 2014, http://heartbleed.com/, abgerufen am 28.06.2014.

[20] T. Dierks, C. Allen, „RFC 2246: The TLS Protocol", IETF, 1999.

[21] Fachgebiet Angewandte Informationssicherheit, „AIS Heartbleed Challenge Server", 2014, https://heartbleed.ais.uni-kassel.de/, abgerufen am 30.06.2014.

[22] A. Wacker, „Schülerkrypto2014", 2014, http://www.cryptool.org/schuelerkrypto/, abgerufen am 28.06.2014.

[23] D. P. Anderson, G. Fedak, „The Computational and Storage Potential of Volunteer Computing", in *CCGRID 06, Cluster Computing and the Grid*, 2006.

[24] MTC3 Team, „MysteryTwister C3 – The Cryptography Cipher Contest", 2014, https://www.mysterytwisterc3.org/, abgerufen am 28.06.2014.

[25] A. Wacker, „Einführung in CrypTool und CrypTool 2.0 – Schülerkrypto 2013", 2013, https://www.youtube.com/watch?v=KsBhU5Pi3Nc, abgerufen am 07.07.2014.