CrypTool 2 – Ein Open-Source-Projekt zur Kryptologie für Lehre, Forschung, Selbststudium und Experimentieren

Nils Kopal · Bernhard Esslinger

CrypTool-Projekt
https://www.cryptool.org
{kopal | esslinger}@cryptool.org

Zusammenfassung

Dieser Artikel stellt das CrypTool-Projekt und insbesondere die E-Learning-Software CrypTool 2 (CT2) vor. Er erläutert die Motivation, eine Software zur Kryptologie zu entwickeln, die die Qualität professioneller Software hat, einem agilen Entwicklungsprozess folgt und weltweit eingesetzt wird. CT2 wird sowohl zum Lernen, Lehren, Forschen als auch für Awareness-Maßnahmen benutzt. Die zukünftige Planung des CT2-Projekts bildet den Abschluss des Artikels.

1 Einleitung

Kryptographie (also das Ver- und Entschlüsseln geheimer Botschaften) und Kryptoanalyse (die Wissenschaft des Brechens geheimer Botschaften, ohne Kenntnis des geheimen Schlüssels) bilden zusammen das Forschungs- und Anwendungsgebiet der Kryptologie. Bis zum Ende des 20. Jahrhunderts wurde die Kryptologie hauptsächlich nicht-öffentlich (von Geheimdiensten und militärischen Organisationen) erforscht und genutzt. In der öffentlichen Lehre an Universitäten fand sich die Kryptologie erst ab den 1970er Jahren. Heute ist Kryptologie Teil eines jeden Informatikstudiums, mindestens als Kapitel innerhalb einer IT-Sicherheitsvorlesung, häufig sogar als eigenständiger Kurs und mittlerweile sogar als ganze Studienvertiefung.

Kryptologie begegnet jedermann täglich in den unterschiedlichsten Anwendungen: Beim Online-Shopping als gesicherte TLS/SSL-Verbindung, bei W-LAN, das durch WPA2 gesichert wird, beim Pay-TV, das mithilfe von Smartcards entschlüsselt wird, bei Ende-zu-Ende verschlüsselnden Messengern etc. Die Sicherheit einer jeden gesicherten Anwendung steigt und fällt nicht nur mit der Güte der eingesetzten Kryptographie, sondern auch mit deren korrekter Anwendung. Erfolgreiche Angriffe auf IT-Systeme nutzen selten Schwächen in eigentlichen Basiskomponenten (Primitive) der Krypto-Verfahren, aber häufig fehlerhafte Implementierungen oder falsches Benutzen der Kryptographie. So sind schwache Passwörter häufig das einfachste Einfallstor für Angreifer. Aus diesem Grund ist die Schulung von IT- und insbesondere von IT-Security-Fachkräften wichtig. Um deren Ausbildung und Schulung zu fördern, arbeitet das CrypTool-Team (CT-Team) seit nunmehr 20 Jahren an kostenloser Software für die Unterstützung der Lehre und des Selbststudiums der Kryptologie. Weltweit arbeiten dabei rund 100 Freiwillige mit.

P. Schartner · N. Pohlmann (Hrsg.) · D•A•CH Security 2018 · syssec (2018) 13-25.

1.1 Entstehung des CrypTool-Projekts

Die Entwicklung der CrypTool-Software wurde 1998 als Awareness-Projekt für die Mitarbeiter der Deutschen Bank gestartet. Das Projekt wurde intern zu einem so großen Erfolg, dass der IT-Vorstand 2000 erlaubte, die Software als Freeware zum Download anzubieten. 2003 wurde CrypTool (CT) in ein Open-Source-Projekt umgewandelt. Nun stand sowohl das Programm CT als auch dessen Source-Code der Allgemeinheit zur Verfügung. CT wird von einer breiten Community stetig weiter entwickelt und gepflegt. Alle CT-Programme können von [CT18a] kostenlos heruntergeladen werden.

1.2 Entstehung des CrypTool 2-Projekts

Fast 10 Jahre nach dem Start der Entwicklung der ersten CrypTool-Variante (CT1) wurde 2007 entschieden, einen Nachfolger von CT1 namens CrypTool 2 (CT2) zu entwickeln [KKWE14]. Dies war aus mehreren Gründen notwendig geworden: Zum einen war CT1 monolithisch entwickelt und Erweiterungen waren und sind für neue Entwickler schwer in die Software zu integrieren. Zum anderen basiert CT1 auf einem Fenster-Prinzip, das für die didaktische Nachverfolgbarkeit von kryptographischen Verfahren nicht optimal ist: Der zu verschlüsselnde Text wird zunächst in ein eigenes Fenster eingegeben. Danach wird über einen Menüpunkt und eine Dialogbox die Kryptofunktion ausgewählt, parametrisiert und ausgeführt. Daraufhin erscheint ein neues Fenster in CT1, das den verschlüsselten Text anzeigt. Möchte der Benutzer nun etwas am Eingabetext (und indirekt am Ausgabetext) ändern, so muss er diese Schritte alle erneut durchführen.

Für CT2 wurde mithilfe von Mediendesign- und Computervisualistik-Lehrstühlen in Aachen und Koblenz ein modernes Benutzungskonzept entwickelt, das hier Abhilfe schafft. Zudem lässt sich CT2 deutlich einfacher erweitern: So können neue Kryptofunktionen mithilfe standardisierter Interfaces automatisch geladen und genutzt werden. Alle Funktionen können als grafische Komponente auf einem virtuellen Arbeitsplatz, dem CT2-Workspace, angeordnet und miteinander verknüpft werden. So kann bspw. eine Texteingabe-Komponente mit einer Verschlüsselungs-Komponente und diese wiederum mit einer Textausgabe-Komponente verknüpft werden. Sobald dann in die Texteingabe-Komponente Text eingegeben wird, wird dieser automatisch von der Verschlüsselungs-Komponente verschlüsselt und in der Textausgabe-Komponente angezeigt. Dadurch werden Änderungen an der Eingabe sofort auch in der Ausgabe ersichtlich. Außerdem sollten alle grafischen Elemente zoombar sein, weswegen innerhalb von .NET auf WPF [The12] gesetzt wurde. Diese Entscheidung sorgt für ein attraktives und flexibles Aussehen unter Windows, verhindert aber auch, dass CT2 unter Mac und Linux läuft, da das inzwischen auch von Microsoft geförderte Mono-Projekt wohl .NET auf andere Betriebssysteme portiert, aber nicht den WPF-Part.

Im weiteren Verlauf des Artikels wird zunächst auf das CT-Gesamtprojekt eingegangen. Darauf folgt in Kapitel 3 der Aufbau und die Bedienung von CT2. Dann wird die Nutzung von CT2 an Schulen und in der universitären Lehre beispielhaft beschrieben. In Kapitel 5 werden der Einsatz in und der Nutzen für die Forschung aufgezeigt. Auf den Einsatz zur Ausbildung und in Awareness-Maßnahmen in Firmen und Behörden (bspw. bei PWC, Boeing, Microsoft, BKA) wird hier nicht eingegangen. Kapitel 6 diskutiert, wie die mittelfristige Zukunft von CT2 aussieht. Kapitel 7 fasst den Artikel kurz zusammen.

2 Das CrypTool-Gesamtprojekt

Das CT-Projekt besteht nicht nur aus den bereits erwähnten zwei Software-Projekten CrypTool 1 (CT1) und CrypTool 2 (CT2). Neben diesen beiden gibt es noch die Projekte: CrypTool-Online (CTO), JCrypTool (JCT), die Schülerkrypto-Veranstaltungen [CT18c], die Krypto-Rätsel-Webseite MysteryTwister C3 (MTC3) [MT18] und das CrypTool-Portal (die Einstiegs-Webseite) [CT18a].

CrypTool-Online (CTO) führt klassische und moderne Verschlüsselungsalgorithmen als Anwendung im Browser (für PCs und Smartphones) aus. Auch Kryptoanalyseverfahren oder Awareness-Tools wie Passwort-Qualitätsmesser sind implementiert. Die Hauptidee für CTO war, ein CT zu bauen, das als Einstieg dienen und auch unterwegs auf dem Mobiltelefon einfach genutzt werden kann. So wird CTO häufig von Geo-Cachern eingesetzt, um "Mystery Caches" (verschlüsselte Nachrichten) zu brechen.

Bei **JCrypTool** (JCT) handelt es sich – wie bei CT1 und CT2 – um ein eigenständiges, lokal ausgeführtes Programm. JCT ist vollständig in Java entwickelt und setzt auf Eclipse auf. Konzeptionell ist es ähnlich zu CT1 und baut ebenfalls auf Fenstern und Menüs auf, enthält jedoch auch Kaskaden und sehr viele Visualisierungen (bspw. zu Elliptischen Kurven oder Quantencomputer-resistenten Signaturverfahren). Demnächst wird JCT seine Oberfläche dynamisch an Erweiterungen der Kryptobibliothek BouncyCastle [Leg18] anpassen können.

Die **Schülerkrypto** wird mehrmals im Jahr durchgeführt. Ziel ist es, mit Hilfe einer "Agentenausbildung" Schüler für ein Studium der MINT-Fächer (Mathematik, Informatik, Naturwissenschaften und Technik) zu begeistern. Weitere Details zur Schülerkrypto folgen in Kapitel 4.1.

Auf der Webseite **MysteryTwister C3** (MTC3) kann man kryptographische Rätsel (Challenges) lösen und Punkte für eine Bestenliste (Hall-of-Fame) sammeln. Neben den Leveln 1, 2 und 3 gibt es auch das Level X. Level-1-Challenges können noch von Hand gelöst werden, bei Level 2 benötigt man Programmierkenntnisse für die Lösung, und Level-3-Challenges können nur mit erheblicher Rechenleistung und/oder neuen Ansätzen gelöst werden. Bei Level-X-Challenges kennen auch die Autoren die Lösung nicht. Eine Level-X-Challenge kann bspw. eine historische Chiffre sein, von der noch niemand weiß, mit welchem Verfahren sie verschlüsselt wurde. Schon mehrfach wurden bei MTC3 historische Level-X-Challenges gelöst. Die meisten Punkte bringt das Lösen von Level-3- und Level-X-Challenges. Zu MTC3 gehört auch ein moderiertes Forum, in dem die Teilnehmer Unterstützung erhalten. Inzwischen gibt es über 250 Aufgaben (jeweils auf deutsch und englisch) und über 8.000 angemeldete Benutzer. Im Vordergrund steht für die Teilnehmer der Spaß am Lösen der Rätsel.

Eine Übersicht über die CT-Programme und die Anzahl der darin enthaltenen Kryptofunktionen enthält Tabelle 1.

Tab. 1: Übersicht zu den CrypTool-Programmen

Programm	ab	Sprache	Betriebssystem	Akt. Version	Released	#Funktionen
CT1	1998	C, C++	Windows	1.4.41	März 2018	ca. 130
CT2	2008	C#	Windows	2.1 (2018.2)	April 2018	ca. 250
JCT	2008	Java	Win, Linux, MacOS	0.9.9	Okt. 2016	ca. 150
CTO	2010	JS	alle (im Browser)	laufend	laufend	ca. 70

Zur Philosophie des CT-Projekts gehören der Open-Source-Gedanke, das respektvolle und lernende Miteinander der Entwickler und eine hohe Responsiveness für die Anliegen der Benutzer. Alle angebotenen Programme sind frei und der Source-Code ist kostenlos. Das Projekt baut auf die Mitarbeit von Freiwilligen. In den CT-Teilprojekten arbeiten Studierende verschiedener nationaler und internationaler Hochschulen. Dies geschieht unter anderem in Seminaren, Projekten und Abschlussarbeiten sowohl von Diplom-, Master- als auch Bachelor-Studenten – und das mit sehr großem Erfolg, sowohl für die Studierenden, als auch für das CT-Projekt. Koordiniert wird das CT-Projekt von einem Kernteam aus erfahrenen Entwicklern und Kryptologen, so dass die Studenten und neue freiwillige Mitentwickler viel über moderne, verteilte Softwareentwicklung lernen.

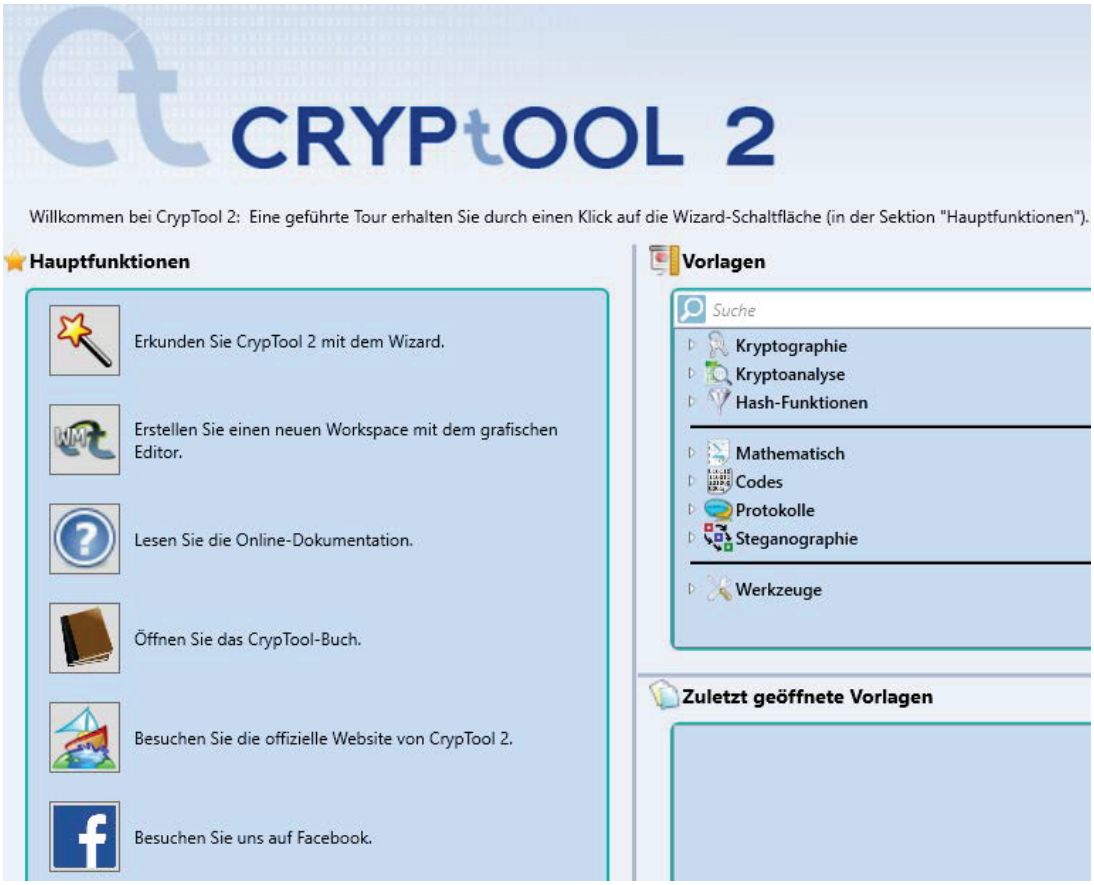

Abb. 1: CrypTool 2 – Startcenter (Ausschnitt)

3 Aufbau und Benutzung von CrypTool 2

CT2 wurde nahezu vollständig in C# [AA17] entwickelt. Nur Performance-kritische Stellen sind in anderen Sprachen, u.a. C++ und OpenCL [NFHS11], geschrieben. CT2 kann aufgrund seiner Plugin-basierten Softwarearchitektur einfach erweitert werden. So besteht CT2 aus vielen einzelnen .NET-Assemblies (Programmbibliotheken), die zur Laufzeit dynamisch nachgeladen werden können. Aufgrund des Netzwerks der Koordinatoren waren zu Beginn, also während der Design-Phase des Produkts, auch zwei C#- und VisualStudio-Evangelisten von Microsoft für mehrere Wochen kostenlos dabei, um eine nachhaltige Basis für die Architektur zu legen. CT2

wird multilingual entwickelt. Aktuell kann man zwischen Deutsch und Englisch umschalten. Weitere Sprachen, unter anderem Chinesisch und Russisch, sind geplant.

Aus Benutzersicht besteht CT2 aus fünf "Kern-Komponenten": dem Startcenter, dem Wizard, dem virtuellen Arbeitsplatz (Workspace Manager), der Online-Hilfe und den Vorlagen.

Das **Startcenter**, ausschnittsweise zu sehen in Abbildung 1, ist der Einstiegsbildschirm von CT2. Er ermöglicht dem Benutzer beim Starten von CT2 den Zugriff auf alle weiteren Kern-Komponenten. Auf der linken Seite des Startcenters sind unter anderem der Wizard, der Workspace-Manager und die **Online-Hilfe** erreichbar. In der Mitte können alle Vorlagen durchsucht und geöffnet werden. Eine **Vorlage** (Template) enthält bereits vorgefertigte kryptographische Szenarien, beispielsweise eine Caesar-Verschlüsselung oder eine dynamische Simulation des Padding-Oracle-Angriffs gegen SSL. Jeder Benutzer kann aber auch eigene Szenarien erstellen und als CWM-Datei abspeichern und verteilen.

Der **Wizard** basiert im Kern auf derselben Idee wie CT1 und soll den Übergang von CT1 auf CT2 erleichtern. Der Benutzer wählt zunächst über verschiedene Auswahlmenüs aus, was er gerne tun möchte. Z.B. wählt er zunächst "Verschlüsselungsverfahren", dann "Klassische Verschlüsselungsverfahren" und daraufhin "Caesar". Hier kann nun ein Klartext eingegeben werden. Dieser wird dann final ver- oder entschlüsselt und im letzten Schritt angezeigt.

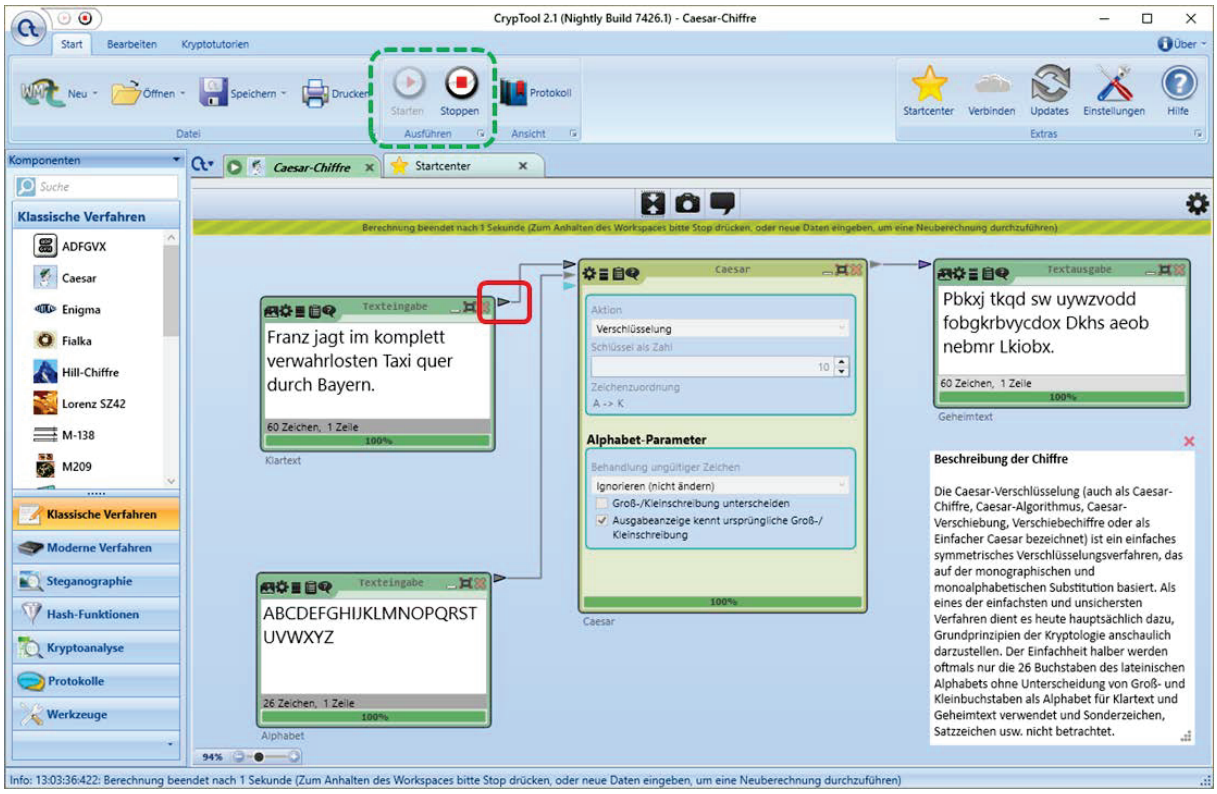

Abb. 2: CrypTool 2 – Workspace-Manager mit Caesar-Chiffre

Der **Workspace-Manager** ist das Herzstück von CT2, er beinhaltet die grafische Programmiersprache sowie die zugehörige Ausführungsmaschine. Abbildung 2 zeigt, wie darin eine Caesar-Chiffre ausgeführt wird. Im Workspace-Manager kann der Benutzer per Drag&Drop verschiedene Komponenten als grafische Symbole (Icons) auf die Arbeitsfläche legen.

Jede Komponente kann mehrere Ein- und Ausgänge besitzen, über die von den Komponenten Daten ein- und ausgegeben werden. Datentypen wie Text, Zahl etc. werden durch unterschiedliche Farben der Ein- und Ausgangspfeile dargestellt (siehe rote Umrahmung in Abbildung 2). Gleiche Farben können immer miteinander verbunden werden. Bei unterschiedlichen Farben sind implizite oder explizite Konvertierungen, durch die beigelegten Konvertierungs-Komponenten, nötig. Das grafische Programm in Abbildung 2 implementiert eine Caesar-Chiffre (Substitutions-Verschlüsselung mit Verschiebung des Alphabets um einen fixen Wert (Schlüssel)). Hierfür wurden zwei Texteingabe-Komponenten mit der Caesar-Komponente und diese mit einer Textausgabe-Komponente verbunden. Die erste Texteingabe enthält den Klartext und die zweite das zu nutzende Alphabet. Die Textausgabe stellt den Geheimtext dar. Die eigentliche Verschlüsselungs-Komponente braucht "von außen" nur den Klartext. Alphabet und Schlüssel sind optionale Eingänge – sie können auch innerhalb der Caesar-Komponente eingestellt werden (für optionale Eingänge sind initial sinnvolle Defaultwerte vorgegeben). Mit Shortcuts wie F11 und F12 kann man bspw. für Präsentationen das Aussehen des Workspace-Managers einfach maximieren.

Damit ein grafisches Programm in CT2 ausgeführt wird, muss die Ausführung explizit gestartet werden. Dies geschieht mit dem **Play**-Button (Starten), der sich in der oberen Leiste befindet (siehe grün-gestrichelte Umrahmung in Abbildung 2). Gestoppt wird mit dem Stop-Button (Stoppen).

Die mitgelieferte Online-Hilfe enthält Informationen zu allen CT2-Komponenten und -Vorlagen sowie weitere Hinweise zu den Verfahren, zu Literatur- und Internetquellen.

4 CrypTool 2 in der Lehre

Die CT-Programme werden vielfach innerhalb der Lehre eingesetzt – sowohl im Unterricht an Realschulen und Gymnasien als auch in der universitären Ausbildung. Die folgenden Kapitel beschreiben zunächst den Einsatz in der Schule, danach in der universitären Vorlesung und zum Schluss während studentischer Projekte und Abschlussarbeiten.

4.1 CrypTool 2 in der Schule

Neben Vertiefungen und Arbeitsgemeinschaften (AGs) kommt CT2 bei der Schülerkrypto zum Einsatz. In der Schülerkrypto erhalten Oberstufenschüler spielerisch einen Einblick in die Welt der Kryptologie. Die Schüler lernen in dem ganztägigen Kurs sowohl klassische Chiffren (Skytale, Caesar, einfache monoalphabetische Substitution, einfache Spaltentransposition), die Enigma-Verschlüsselungsmaschine als auch moderne symmetrische Chiffren (DES [NIS99], AES [DR13]) und asymmetrische Kryptographie (Diffie-Hellman-Schlüsselaustausch [DH76], RSA-Verschlüsselung [RSA78]) kennen. Neben den eigentlichen Chiffren ist auch deren Kryptoanalyse Thema der Schülerkrypto. Zumindest die einfachen Chiffren können mit Hilfe der in CT1 und CT2 integrierten Analyseverfahren von den Schülern in den praktischen Übungseinheiten gebrochen werden. Auch die modernen Verfahren werden, in abgeschwächter Form (verringerte Schlüsselraumgröße), von den Schülern gebrochen. Als zusätzliche Motivation können die schnellsten Schüler einen Sachpreis gewinnen, z.B. Bücher oder Filme über Kryptologie, Geheimdienste oder Digitalisierung.

Alle Aufgaben der Schülerkrypto können mit Hilfe von CT1 und CT2 gelöst werden. Dazu wird innerhalb der Vorlesungsteile der Schülerkrypto in die Bedienung der Tools eingeführt.

Hauptziel der Schülerkrypto ist es, die teilnehmenden Schüler zum Studium eines MINT-Fachs zu motivieren, und zu zeigen, dass Mathematik nicht nur Theorie ist, sondern dass es spannende und konkrete Anwendungen gibt, mit denen sie auch selbst gleich experimentieren können.

Erst der Einsatz von CT1 und CT2 ermöglicht es, ein solch umfangreiches Tages-Programm durchzuarbeiten, da die Verfahren nun nicht mehr mit Stift und Papier manuell ausgeführt werden müssen. Aus didaktischen Gründen werden die einfachsten Verschlüsselungsverfahren von den Schülern dennoch jeweils einmal während der Vorlesungsteile manuell durchgeführt, damit die Chiffren vollständig verstanden werden.

Ergänzend wird eine vertiefende 2-tägige Lehrerveranstaltung angeboten, die bisher jedoch deutlich seltener in Anspruch genommen wurde. Weiterhin wurden zusammen mit Lehrern und Informatik-Didaktikern auf Konferenzen (wie der INFOS) oder bei Workshops (wie denen der GI-Fachgruppe “Informatik-Bildung”) oder in Didaktik-Fachzeitschriften (wie der “LOG IN”) Angebote gemacht, die Unterrichtsthemen mit CrypTool, Python und SageMath [Sag18] ausarbeiteten [LOG18].

4.2 CrypTool 2 in der universitären Vorlesung und Übung

Mit den Visualisierungen in CT2 können bestimmte Aspekte eines Verfahrens gut verdeutlicht werden. Ein Beispiel aus einer aktuellen Kryptologievorlesung ist die in Abbildung 3 dargestellte Visualisierung. Um mehr Daten als die sogenannte Blocklänge (z.B. 128 Bytes) mit modernen symmetrischen Verfahren zu verschlüsseln, gibt es verschiedene kryptographische Standards – die Block-Modi. Ein veralteter und unsicherer Modus ist der Electronic Codebook Modus (ECB-Modus). Besser und sicherer ist der Cipher Block Chaining Modus (CBC-Modus). In Abbildung 3 wird die zu verschlüsselnde Nachricht (in diesem Fall das Bild des Smileys auf der linken Seite) mittels beider Modi und jeweils derselben modernen Blockchiffre verschlüsselt. Einmal ist der Smiley im Chiffrat noch in seinen Grundzügen zu erkennen (ECB-Modus, oben rechts), einmal nicht (CBC-Modus). So wird den Studierenden auch visuell nahegebracht, warum der eine Modus auf gar keinen Fall mehr eingesetzt werden darf.

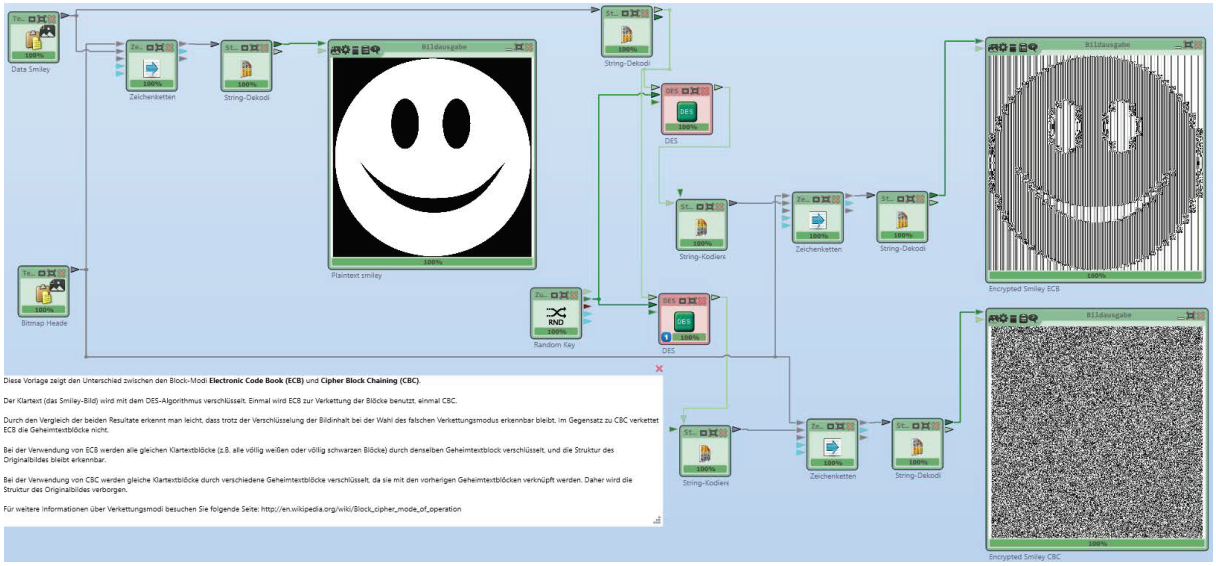

Abb. 3: CrypTool 2 – Visualisierung von Block-Modi moderner Chiffren

Neben dem oben gezeigten Beispiel bietet CT2 viele weitere Vorlagen an. Auch innerhalb der Übungen wird CT2 von Studierenden genutzt, um sowohl klassische als auch moderne kryptographische und kryptoanalytische Verfahren auszuprobieren und zu verstehen (z.B. Padding und seine Auswirkungen). Zudem dient CT2 hier auch als Referenzimplementierung, wenn Studenten Kryptoverfahren selbst programmieren sollen.

4.3 Bachelor- und Master-Arbeiten und -Projekte in CT2

CT2 wäre heute nicht so umfangreich, wenn nicht kontinuierlich neue Komponenten durch studentische Projekte und Abschlussarbeiten hinzugefügt worden wären. Bisher flossen weit über 100 Arbeiten von in- und ausländischen Hochschulen direkt ein. CT bietet aus mehrerlei Hinsicht für Studierende der Mathematik, Informatik und Wirtschaftsinformatik eine interessante Plattform: 1) Sie können sich praktisch mit kryptologischen Themen beschäftigen. 2) An CT2 arbeiten viele Entwickler gleichzeitig. So lernen Studierende im Team und im Großprojekt zu arbeiten. 3) CT2 basiert auf modernen Programmierkonzepten (bspw. C#, .NET, WPF [The12], C++, OpenCL, CUDA [NFHS11]). 4) Alle Entwicklungen werden, sofern sie den Qualitätsansprüchen des CT2-Teams genügen, innerhalb des CT2-Projekts veröffentlicht. So wird "Arbeit für die Schublade" oder nur für die Credit Points vermieden und die Studierenden können ihre Komponenten, z.B. in Bewerbungsunterlagen, referenzieren. 5) CT2 versucht, möglichst aktuell zu sein. Dadurch können Studierende auch an neuen "Hypes" mitarbeiten – z.B. an Blockchain-Technologie, homomorpher Verschlüsselung und verteiltem Rechnen.

Da sich die Kryptologie stetig weiter entwickelt, wird auch CT2 niemals "fertig" sein. Deswegen können stets Projekte und Abschlussarbeiten mit und für CT2 durchgeführt werden, wobei Mitarbeiter des CT-Projektes bei Bedarf auch gern als Zweitbetreuer fungieren. So wird versucht, immer die besten Kryptoanalyseverfahren sowohl für klassische als auch für moderne Chiffren in CT2 einzubauen und Studierenden spannende Arbeiten zu ermöglichen.

Erfolgreiche und sehr gute Abschlussarbeiten sind z.B. der CT2-Workspace-Manager, der CT2-Wizard, die Visualisierung der Enigma, die Kryptoanalyse der M-138-Chiffre, die Visualisierung des BB84-Protokolls [BB84], die verschiedenen Ausprägungen der Keccak-Hashfunktion [BDPVA09], visuelle Codierungen, robuste Hash-Funktionen, digitale Wasserzeichen, und der Diffie-Hellman-AES-Video- und Audio-Chat.

5 CrypTool 2 in der Forschung

Neben dem Einsatz in der Lehre, für den CT2 als E-Learning-Tool primär entwickelt wird, bietet es auch Einsatzmöglichkeiten in der Forschung. Neue Ergebnisse aus der aktuellen Forschung fließen ein und sind dann dauerhaft sichtbar und nutzbar.

5.1 CrypTool 2 als Sammelstelle für State-of-the-Art-Krypto

CT2 wird weniger selbst als Forschungstool benutzt, sondern als Testbett, in dem Forschungsergebnisse ausprobiert werden. Die Ergebnisse werden dann vom CT-Team weiter gepflegt und verfügbar gehalten, auch wenn die Forscher sich schon mit anderen Dingen befassen und ihre Sourcen nicht mehr betreuen.

Wie bereits erwähnt, ist das CT2-Team bemüht, stets neueste Chiffren und die aktuell besten Analyseverfahren zu implementieren. Dadurch kann CT2 auch als Sammelstelle des "State-of-the-Art" angesehen werden. Dabei wird nicht unbedingt jeder Kryptoalgorithmus in CT2 einge-

baut. Ziel ist, alle relevanten modernen Chiffren in CT2 anzubieten. Schwache oder Amateur-Chiffren werden nur dann integriert, wenn dies entweder einen didaktischen Nutzen hat oder die Chiffren geschichtlich relevant sind. Natürlich hält das CT-Team niemand davon ab, eigene Verfahren zu erproben und bspw. über den CT-Store anzubieten.

Bei den Analyseverfahren enthält CT2 mittlerweile ein umfangreiches Repertoire an Kryptoanalyse-Komponenten – insbesondere für klassische Chiffren und Verschlüsselungsmaschinen. Diese basieren unter anderem auf aktuellen Forschungsergebnissen, z.B. aus der Dissertation von Lasry [Las18].

Zur verteilten Kryptoanalyse (siehe Kapitel 6.2) flossen in CT2 bspw. die Ergebnisse der Dissertation von Kopal [Kop18] und weitere Vorarbeiten der Uni Duisburg ein.

Implementierungen moderner Forschungsergebnisse finden sich auch auf vielen anderen Webseiten wie GitHub, in SageMath, bei Bernstein [Ber18] oder bei ECRYPT [ECR18]. Etablierte Programme wie OpenSSL oder Bibliotheken wie BouncyCastle enthalten jedoch fast nie kryptoanalytische Verfahren. Das Alleinstellungsmerkmal von CrypTool ist, dass Kryptographie und Kryptoanalyse an einer Stelle verfügbar sind – und das nicht nur für Einzelverfahren, sondern für die ganze Bandbreite (wenn auch nicht so tief wie in Spezialimplementierungen).

5.2 Kryptoanalyse historischer Chiffren mit CrypTool 2

Mitglieder im CT2-Team sind aktiv an Forschung und Entwicklung von neuen Kryptoanalyseverfahren (Optimierungsverfahren) für klassische Verschlüsselungen beteiligt. Beispiele sind die ADFGVX-Chiffre [LNKW17], die Spaltentransposition [LKW16b], die Enigma [Gil95], die M-209-Verschlüsselungsmaschine [LKW16a] und Multiplexchiffren [Kop17]. Ergebnisse dieser Forschungen fließen direkt, als Komponenten, in CT2 ein. So enthält CT2 z.B. die aktuell besten Algorithmen für die Analyse der einfachen und doppelten Spaltentransposition, welche von Lasry [LKW14, LKW16b] entwickelt wurden. Auch zur Analyse der monoalphabetischen Substitution ist einer der besten bekannten Algorithmen enthalten. Mit der implementierten Vigenère-Analyse lassen sich auch die entsprechenden Beispiele der American Cryptogram Association (ACA) [ACA18] lösen, die sie selbst als schwer einstufte.

In CT2 basieren die meisten Analyseverfahren für klassische Chiffren auf Heuristiken wie Hill-Climbing, Simulated Annealing oder genetischen Algorithmen.

So dient CT2 zum einen zum Bereitstellen der entwickelten Prototypen, zum anderen profitieren Nutzer von CT2 direkt von den implementierten Verfahren. Dazu gehören auch Historiker, die “Historische Chiffrate” entschlüsseln möchten, z.B. wie im DECODE-Projekt [PM18], das verschlüsselte historische Manuskripte sammelt, indiziert und analysiert. Hier besteht eine Kooperation mit der Uni Uppsala.

6 Weitere Pläne für CrypTool 2

In diesem Kapitel werden die aktuellen, mittelfristigen Pläne des CT2-Teams kurz vorgestellt. Neben den laufenden Arbeiten soll CT2 als “das” Tool der Wahl für klassische Kryptoanalyse etabliert werden, verteiltes Rechnen ermöglichen und einen “CrypTool-Store” anbieten.

6.1 “Erste Wahl” für die Kryptoanalyse klassischer Chiffren

CT2 beinhaltet eine Vielzahl von Kryptoanalyseverfahren, insbesondere für klassische Chiffren. Allerdings sind die vorhandenen Chiffren sowie Analyse-Komponenten noch nicht vollständig

und robust genug für historische Forscher.

Ziel des CT2-Teams ist es daher, CT2 attraktiver für die interdisziplinäre Nutzung zu machen. Es gibt einige historische Texte und Bücher, die verschlüsselt wurden. Derartige Verschlüsselungen sind bspw. homophone Buchstaben- und Wort-Ersetzungen.

Um CT2 zum "Tool der Wahl" für die Kryptoanalyse klassischer Chiffren zu machen, arbeitet das CT2-Team eng mit Historikern, (Computer-)Linguisten und Forschern aus dem Bereich des Image Processing zusammen. Dabei wird versucht, CT2 noch besser an die Nutzer-Bedürfnisse anzupassen und den Einstieg in die Software einfacher zu machen.

Alternativen zu CT2 bezüglich klassischer Verfahren bieten bspw. die Webseiten von ACA [ACA18], von Simon Singh, und insbesondere von Phil Pilcrow [Pil18], der auch bei CT mitwirkt und das Ziel hat, für alle von ACA angebotenen klassischen Verschlüsselungsverfahren Cipher-text-only-Angriffe zu erstellen. Einige (graphische) Crack-Tools für klassische Verfahren werden leider nicht mehr weiter entwickelt: EverCrack (bis 2006), Ganzua (bis 2004) und Crank (bis 2001).

Eine **Alternative zu CT2 bezüglich moderner Verfahren** ist bspw. das Kommandozeilen-Programm John the Ripper zum Passwort-Knacken, das sehr gut ist und ständig aktuell gehalten wird. Ein anderes alternatives modernes Programm aus der Forschung ist Lineartrails, ein Tool der TU Graz zur automatischen linearen Kryptoanalyse von Substitution-Permutation-Networks, das die linearen Charakteristiken von Kryptofunktionen identifiziert. Es wurde 2015 für ein Paper auf der AsiaCrypt erstellt und unseres Wissens seitdem aber nicht weiter entwickelt. Ein weiteres modernes und spannendes Programm ist CipherCAD aus 2011, mit dem man kryptographische Funktionen und Protokolle graphisch modellieren kann. Leider scheint das Programm nicht für die Öffentlichkeit verfügbar zu sein.

6.2 Verteilte Kryptoanalyse mit CrypTool 2

Etliche kryptographische Verfahren können mit einem Computer allein nur in sehr langer Zeit oder gar nicht gebrochen werden, sind jedoch durch einen Rechnerverbund durchaus angreifbar.[1] Ein Beispiel aus der klassischen Kryptographie sind historische Enigma-Nachrichten, die mit Hilfe von vielen Computern simultan in wochen-, monate- oder jahrelanger Rechenarbeit gebrochen werden. Ein Beispiel aus der modernen Kryptographie ist das Finden von Passwörtern. Moderne kryptographische Passwort-Hashing-Verfahren wie bcrypt oder scrypt gelten als sicher. Jedoch ist es möglich, per Vorwärtssuche die Passwörter zu finden, sofern die Passwörter "schwach" sind. Schwache Passwörter sind z.B. kurz oder stehen in einem Wörterbuch. Mit einem Rechnerverbund können auch stärkere Passwörter gebrochen werden.

CT2 verfügt seit einiger Zeit über die sogenannte "CrypCloud". CrypCloud basiert auf einem unstrukturierten Peer-to-Peer-Netzwerk [MS07] ohne zentrale Instanz und ermöglicht die verteilte Kryptoanalyse von modernen symmetrischen Chiffren, z.B. AES und DES. Mittels des KeySearchers können alle Teilräume des Schlüsselraums durchsucht werden, um so den kryptographischen Schlüssel zu finden. Hierfür können in der CrypCloud Jobs erstellt werden, an denen dann viele CT2-Instanzen auf verschiedenen Rechnern mitarbeiten können (Volunteer Computing). Aktuell durchläuft die CrypCloud ein Redesign, um sie robuster zu machen. Wenn dies abgeschlossen ist, wird die CrypCloud auch in den Release-Versionen von CT2 erscheinen.

[1] Anmerkung: Werden die aktuellen modernen kryptographischen Verfahren und Protokolle (z.B. AES, SSL) richtig eingesetzt, gelten sie heute als unbrechbar – deshalb sind diese hier nicht gemeint.

Für die CrypCloud ist neben der verteilten Schlüsselsuche geplant, auch verteilte heuristische Analysen, z.B. für die Enigma, zu implementieren. Auch verteiltes Brechen von Passwörtern und verteilte Faktorisierung sind angedacht.

6.3 Der CrypTool-Store

Aktuell müssen sich externe Entwickler, die dem CT2-Projekt eigene Komponenten beisteuern möchten, an das CT2-Team wenden. Dies ist notwendig, um einen Entwickler-Account zu erhalten und damit neuen Programmcode in das CT2-SVN (Source-Code-Verwaltung) einzuchecken. Danach wird vom CT2-Team geprüft, ob das Beigesteuerte den Qualitätsansprüchen genügt. Auch das Einfügen in den "Nightly Build" sowie in die Beta- und Release-Versionen muss von einem Kernentwickler durchgeführt werden. Erst danach wird der beigesteuerte Code in den ausgelieferten Versionen enthalten sein. (Natürlich ist ein direkter Austausch untereinander auch heute schon möglich, wenn man seinen Code in eine DLL packt und versendet. Der Empfänger muss die DLL dazu in das CT2-Unterverzeichnis "CrypPlugins" kopieren.)

Um diesen Prozess in seiner Gänze zu vereinfachen, ist der sogenannte "CrypTool-Store" geplant. Auch wenn der Name suggeriert, dass hier etwas über CT gekauft werden soll, ist dem nicht so. Der Name orientiert sich lediglich an bekannten App-Stores. Alle im CT-Store angebotenen "Apps" bzw. CT2-Komponenten müssen kostenlos angeboten werden und Open-Source sein. Der Store soll es externen Entwicklern einfach machen, ihre Komponenten (und deren Source-Code) hochzuladen und den CT2-Benutzern anzubieten. Um im Store etwas zu veröffentlichen, wird man sich als Entwickler registrieren können und ein eigenes Zertifikat erhalten. Mit diesem Zertifikat wird es möglich sein, im CT-Store eigenen Source-Code und CT2-Komponenten zu veröffentlichen. Damit kein schadhafter Code im CT-Store hochgeladen wird, wird dieser vollständig einsehbar sein. CT-Store-Komponenten können, falls qualitativ gut, auch in den Kern aufgenommen werden. Wie bei klassischen "Apps" auf Mobilgeräten wird man auch CT-Store-Komponenten einfach deinstallieren können, falls diese nicht (mehr) gefallen oder nicht mehr notwendig sind. Außerdem können über den Store große Dateien nachgeladen werden, z.B. 5- und 6-Gramm-Statistiken, die man für die fortgeschrittene Parametrisierung der klassischen Kryptoanalyse nutzen kann und die mehrere GB groß werden können.

7 Resümee

Dieser Artikel gab eine Übersicht über das CrypTool-Projekt und über die Software CrypTool 2 (CT2). Zunächst wurde die zwanzigjährige Geschichte des Gesamtprojekts dargestellt. Danach wurden, neben CT1 und CT2, kurz die anderen Teilprojekte (CrypTool-Online, JCrypTool, MysteryTwister C3 und die Schülerkrypto) beschrieben. Im dritten Kapitel wurden die Kern-Komponenten von CT2 (Startcenter, Wizard, Workspace-Manager, Online-Hilfe und Templates) vorgestellt. Kapitel 4 stellte beispielhaft den Einsatz von CT2 an Schulen und Hochschulen vor. Kapitel 5 ging auf den derzeitigen Nutzen von CT2 innerhalb der Forschung ein. CT2 dient als "Sammelstelle" des State-of-the-Art klassischer und moderner Verschlüsselungsalgorithmen und Kryptoanalyseverfahren. Das sechste Kapitel erläuterte drei zukünftige, mittelfristige Ziele für CT2: 1) Etablierung als Standardtool für die Kryptoanalyse historischer Chiffren, 2) Integration der verteilten Kryptoanalyse, und 3) Entwicklung des CrypTool-Stores.

Für weitere Informationen zur Kryptologie und ihrer Anwendung in CT und SageMath kann man das CT-Buch nutzen, das kostenlos unter [Eea18] auf der CT-Webseite erhältlich ist. Für interessierte Entwickler lohnt sich ein Besuch des CT2-Developer-Wikis unter [CT18b]. Dort fin-

den sich Links zum SVN, zu Beispiel-Komponenten und generelle Infos für die Komponenten-Entwicklung sowie der CT2-Developer-Guide.

Die CT-Programme werden pro Monat rund 10.000 mal herunter geladen, wobei über 50 % die englische Version nutzen. Die Downloads kommen vor allem aus Deutschland, USA, Indien, Schweiz, Türkei, Spanien, Russland, Australien, Vietnam, Polen und UK.

Das CT2-Team freut sich immer über konstruktives Feedback und beantwortet Fragen. Falls Interesse besteht, an CT2 mitzuarbeiten (oder auch an den anderen Teilprojekten), können Sie sich gerne bei den Autoren dieses Artikels melden. Gesucht werden immer freiwillige und lernbegierige Software-Entwickler, Krypto-Experten, aber auch Tester, Benutzer und Kryptologiebegeisterte, die wertvolles Feedback geben können.

Literatur

[AA17] J. Albahari, B. Albahari: *C# 7.0 in a Nutshell: The Definitive Reference*. O'Reilly Media, Inc., 2017.

[ACA18] ACA. http://www.cryptogram.org/, 2018.

[BB84] C.H. Bennett, G. Brassard: Bb84. In *Proc. IEEE International Conference on Computers, Systems, and Signal Processing, IEEE Press, Los Alamitos, Calif*, volume 175, 1984.

[BDPVA09] G. Bertoni, J. Daemen, M. Peeters, G. Van Assche: Keccak sponge function family main document. *Submission to NIST (Round 2)*, 3(30), 2009.

[Ber18] D. Bernstein: https://cr.yp.to/, 2018.

[CT18a] CT-Team: Cryptool-projekt/cryptool-portal, https://www.cryptool.org/, 2018.

[CT18b] CT-Team: Cryptool-wiki, https://www.cryptool.org/trac/CrypTool2/, 2018.

[CT18c] CT-Team: Schülerkrypto, https://www.cryptool.org/schuelerkrypto/, 2018.

[DH76] W. Diffie, M. Hellman: New directions in cryptography. *IEEE transactions on Information Theory*, 22(6):644–654, 1976.

[DR13] J. Daemen, V. Rijmen: *The design of Rijndael: AES – the advanced encryption standard*. Springer Science & Business Media, 2013.

[ECR18] ECRYPT: http://www.ecrypt.eu.org/csa/, 2018.

[Eea18] B. Esslinger et al: Cryptool-buch, 12. auflage, https://www.cryptool.org/de/ctp-dokumentation/ctbuch, 2018.

[Gil95] J.J. Gillogly: Ciphertext-only cryptanalysis of enigma. *Cryptologia*, 19(4):405–413, 1995.

[KKWE14] N. Kopal, O. Kieselmann, A. Wacker, B. Esslinger: Cryptool 2.0. *Datenschutz und Datensicherheit-DuD*, 38(10):701–708, 2014.

[Kop17] N. Kopal: A general solution for the m-94 cylinder cipher. *Proceedings of the 3rd Euro-HCC (European Historic Ciphers Colloquium) 2017*, 2017.

[Kop18] N. Kopal: *Secure Volunteer Computing for Distributed Cryptanalysis*. kassel university press GmbH, 2018.

[Las18] G. Lasry: *A Methodology for the Cryptanalysis of Classical Ciphers with Search Metaheuristics*. kassel university press GmbH, 2018.

[Leg18] Legion_of_the_Bouncy_Castle_Inc: Bouncy castle crypto apis, https://www.bouncycastle.org/, 2018.

[LKW14] G. Lasry, N. Kopal, A. Wacker. Solving the double transposition challenge with a divide-and-conquer approach. *Cryptologia*, 38(3):197–214, 2014.

[LKW16a] G. Lasry, N. Kopal, A. Wacker: Ciphertext-only cryptanalysis of hagelin m-209 pins and lugs. *Cryptologia*, 40(2):141–176, 2016.

[LKW16b] G. Lasry, N. Kopal, A. Wacker: Cryptanalysis of columnar transposition cipher with long keys. *Cryptologia*, 40(4):374–398, 2016.

[LNKW17] G. Lasry, I. Niebel, N. Kopal, A. Wacker: Deciphering adfgvx messages from the eastern front of world war i. *Cryptologia*, 41(2):101–136, 2017.

[LOG18] LOGIN: Zeitschrift für die informatische bildung, http://www.log-in-verlag.de/, http://ods2.schule.de/pub/bscw.cgi/159132, 2018.

[MS07] P. Mahlmann, C. Schindelhauer: *Peer-to-Peer-Netzwerke: Algorithmen und Methoden*. Springer-Verlag, 2007.

[MT18] MTC3-Team: Mysterytwister c3 – the crypto challenge contest, https://www.mysterytwisterc3.org/, 2018.

[NFHS11] A. Nischwitz, M. Fischer, P. Haberäcker, G. Socher: Gpu programmierung mit cuda und opencl. In *Computergrafik und Bildverarbeitung*, pages 481–505. Springer, 2011.

[NIS99] NIST: Data encryption standard. *Federal Information Processing Standards Publication*, 46, 1999.

[Pil18] P. Pilcrow: Cryptoprograms und cryptocrack, http://www.cryptoprograms.com/, https://sites.google.com/site/cryptocrackprogram/, 2018.

[PM18] E. Pettersson, B. Megyesi: The histcorp collection of historical corpora and resources. In *DHN 2018 The Third Conference on Digital Humanities in the Nordic Countries, March 7-9 2018, Helsinki, Finland*, pages 306–320. University of Helsinki, 2018.

[RSA78] R.L. Rivest, A. Shamir, L. Adleman: A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21(2):120–126, 1978.

[Sag18] SageMath: Freies und sehr umfangreiches computer-algebra-programm, http://www.sagemath.org/, 2018.

[The12] T. Theis: *Einstieg in WPF: Grundlagen und Praxis*. Galileo Press, 2012.