CrypTool

An Open-Source E-Learning Project for Cryptography and Cryptanalysis

Professor Bernhard Esslinger, University of Siegen

September 22nd, 2016

25th Crypto Day in Walldorf, Germany

UNIVERSITÄT SIEGEN

Abbreviations used

CT	CrypTool (means the project)
CT1	CrypTool v1
CT2	CrypTool v2
JCT	JavaCrypTool
CTO	CrypTool-Online (apply crypto in a browser)
MTC3	MysteryTwister C3 (international cipher contest)
CTP	CrypTool Portal (main website www.cryptool.org)
CTB	CT Book (free and open-source too)

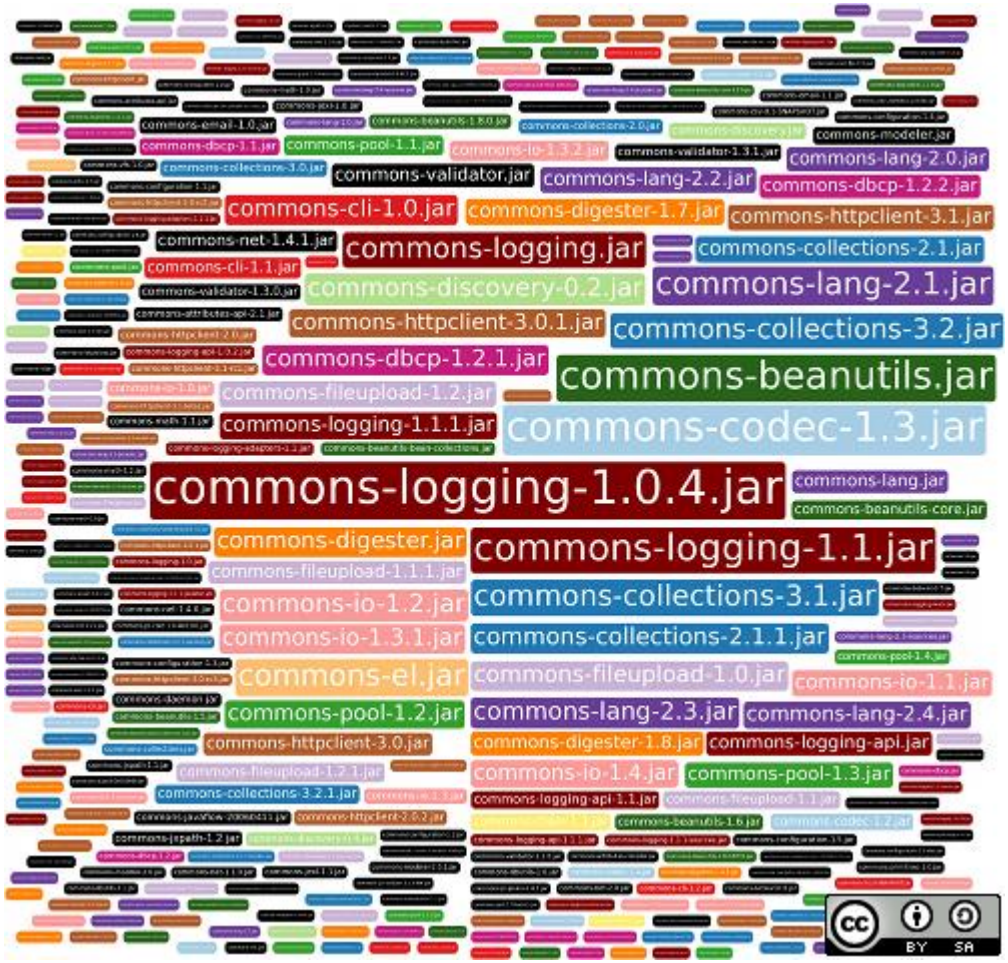
Agenda

Agenda

What Happens with the Implementations of Research Results?

http://commons.wikimedia.org/wiki/File:Universit%C3%A4t_Bonn.jpg
http://commons.wikimedia.org/wiki/File:Bin.JPG

How to Set up an Open-Source Project – 99 % of them are Dead?



http://commons.wikimedia.org/wiki/File:Opensource.svg Logo of the Open Source Initiative
http://commons.wikimedia.org/wiki/File:Project_reuse_ranking_apache_commons_library.png

What Makes an OS Project Successful? → Make Many People Benefit, Make Many People Contribute, Spread the Word, and Start Again

Contributing universities (contributing with crypto plugins): > 20

— Belgrad, Berlin, Bochum, Bonn, Brisbane, Brno, Darmstadt, Dubai, Duisburg-Essen, Eindhoven, Frankfurt, Hagenberg, Jena, Karlsruhe, Kassel, Klagenfurth, Koblenz, London, Madrid, Mannheim, Osnabrück, San Jose, Siegen, Thessaloniki, Utrecht, Warsaw, …

Contributing people

— 70 volunteers, both experts and beginners from all over the world

— Keep the main contributors and the core team happy

High responsiveness; Administrators to run the website securely and stable

— We try to answer each mail within 2 days (we are getting circa 3 mails from users per day)

— Some effort is needed to keep Linux, PHP, Joomla, and all other tools up-to-date

Target Users – Audience

The CrypTool project exists since more than 15 years !

Audience – heterogeneous by will

— Students

— Pupils

— Teachers

— Post Docs

— Lecturers

Mission

— Raise the number of pupils and students to study a MINT subject, and

— Offer a modern e-learning tool to help them succeed when studying information security / cryptography

— Continue to maintain the “good things” (framework instead of bin)

— Support lecturers with an open framework containing the results in cryptology

Agenda

Overview of CrypTool: Three Offline Programs plus Websites

CRYPtOOL version 1.x http://www.cryptool.org/en/cryptool1

CRYPtOOL 2 http://www.cryptool.org/en/ct2-documentation

JCrypTool The cryptography e-learning platform. https://github.com/jcryptool/

CRYPtOOL-ONLINE http://www.cryptool-online.org

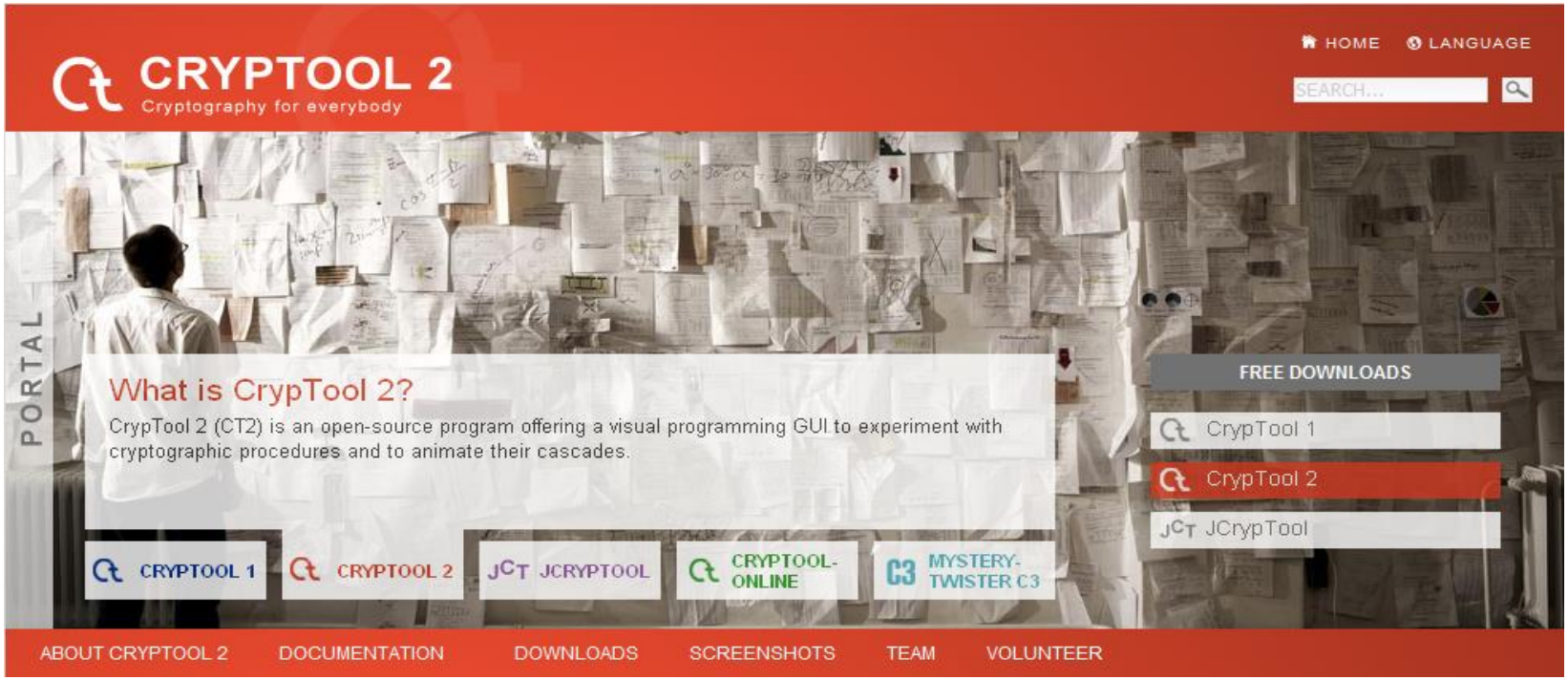
MysteryTwister C3 THE CRYPTO CHALLENGE CONTEST http://www.mysterytwisterc3.org/

CT1
www.cryptool.org/en

- CrypTool 1 [1.4.30 (released); 1.4.31 (stable); 1.5.00 (planned for 2017)]
  - C++ under VS 2015, for Win32
  - Runs under Windows 7, 8, and 10; mature and still broadly used
  - Available in English, German, Spanish, Polish, Serbian, Greek, and (soon) in French

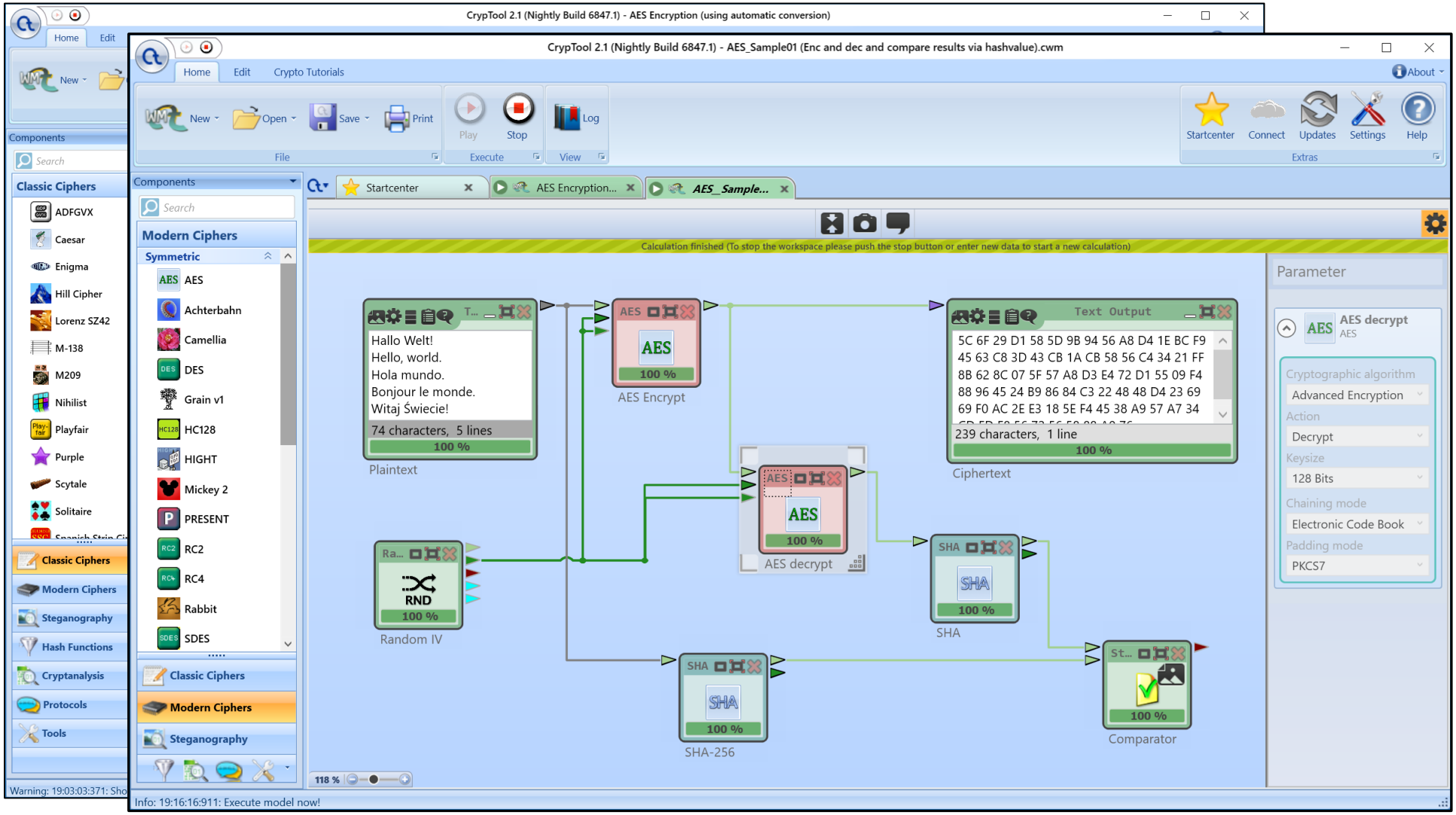
CT2
www.cryptool.org/en

- CrypTool 2 [CT 2.0 released 2014; beta 1 and nightly builds of CT 2.1 are both stable]
  - C# under Visual Studio 2015 (free Express Edition) and WPF
  - Runs under Windows 8 and 10 (requires the .NET framework v 4.0)
  - Available in English and German. Build-in automatic upgrade mechanism



CT2

Example of modern symmetric encryption (AES) in CT2



CT2 Features (1)

- Visual programming (concept developed by universities of Koblenz and Aachen)
  - Allows the combination of cryptographic and cryptanalytic components
  - Implicit data conversion (plus explicit conversion using converters)
  - CT2 learns which links are used more frequently when connecting the components
  - IControl links the components directly (much faster than via the GUI).
    In addition, there is an embedded components interface used e.g. in the KeySearcher.
  - There is a kind of sub routines (calling component chains via the VARIABLE mechanism)
  - Components can include a visualization which is shown within a window directly on the workplace. This allows to visualize several algorithms in "parallel“.
- Classical and modern primitives, and protocols
  - Some have nice visuals like Enigma, PRESENT, Keccak, MD5, transposition, frequency analysis, (N)LFSR, Quadratic Sieve, Key Searcher, QR codes, Padding Oracle Attack
- Video tutorials
  - Further people are needed to create videos we want to show directly within CT2
- Link with information for developing new plugins:
  www.cryptool.org/en/ct2-documentation

CT2 Features (2)

- Networking components supporting TCP / UDP
  - Components allow different participants at different computers to perform a protocol
  - Webcam encryption with and without DH
- Framework for research (to embed your research topic)
  - E-learning / didactics: How to use the new mechanisms, how to try new things?
  - Use the existing tools with all its elements (editor, interfaces, …) to test and discuss new methods (ciphers or attacks)
  - Volunteer Computing for distributed cryptanalysis
    - A multicast P2P network for brute-force attacks comes with CT 2.1
- Work in progress: e.g. Visualization of AES, DES, and Avalance, encrypted virtual machine using SEAL, automated SAT analyzer using CBMC, port of Dieharder, …
- Teaching
  - Used in schools (pupils crypto courses, maths and computer science) and universities
  - Crypto presented more accessible and easier to understand

Demo

CrypTool 2.1 (Nightly Build 6843.1) - Startcenter

Home | Edit | Crypto Tutorials | About

New | Open | Save | Print | Play | Stop | Log | Startcenter | Connect | Updates | Settings | Help

File | Execute | View | Extras

Startcenter | AES Cipher (Te... | Keccak Cipher | Padding Oracle... | Heartbleed Test | CrypCloudMan... | Simple AES Chat

CRYPTOOL 2

Filter

Welcome to CrypTool 2. There are two ways for a quick start: Click on the wizard button in the section "Main Functions" to g tour, or load one of the pre-defined workflows in the section "Templates" which demonstrate the program functionality in cryptographic scenarios.

Main Functions

- Use the wizard to easily try some CrypTool 2 features.
- Create a new workspace with the graphical editor.
- Watch videos about CrypTool 2.
- Read the online documentation.
- Open the CrypTool Book.
- Visit the official CrypTool 2 website.
- Visit us on Facebook.

Templates

Search

- Cryptography
- Cryptanalysis
- Hash Functions
- Mathematics
- Codes
- Protocols
- Steganography
- Tools

News

- Ticket #907 (Crash report 2016-09-16)
- Changeset [6843]: small text change
- Changeset [6842]: some text fixes/char
- Changeset [6841]: some text changes
- Changeset [6840]: fixed connection line
- Changeset [6839]: small typo fix in wiza

Recently Opened Templates

- Simple AES Chat
- Simple Video and Audio Chat using AES encryption
- Heartbleed Test
- Padding Oracle Attack on AES
- Keccak Cipher
- AES Cipher (Text Input)

Show this welcome screen at startup.

Info: 14:17:55:939: AutoUpdate: No updates found - you are on the most current version.

CT2

Fit with one click to workspace size

Add text field (memo) to workspace

CrypTool 2.0 (Nightly Build 5474.1) - Caesar Analysis

Home Edit Crypto Tutorials About

Caesar Analysis | Keccak Hash (SHA-...

Text Input

Ns hwduytlwfumd, f Hfjxfw hnumjw, fqxt pstbs fx Hfjxfw'x hnumjw, ymj xmnky hnumjw, Hfjxfw'x htij tw Hfjxfw xmnky, nx tsj tk ymj xnruqjxy fsi rtxy bnijqd pstbs jshwduynts yjhmsnvzjx. Ny nx f yduj tk xzgxynyzynts hnumjw ns bmnhm jfhm qjyyjw ns ymj uqfnsyjcy nx wjuqfhji gd f qjyyjw xtrj kncji szrgjw tk utxnyntsx itbs ymj fqumfgjy. Ktw jcfruqj, bnym f xmnky tk 3, F btzqi gj wjuqfhji gd I, G btzqi gjhtrj J, fsi xt ts. Ymj rjymti nx sfrji fkyjw Ozqnzx Hfjxfw, bmt zxji ny ns mnx uwnafyj htwwjxutsijshj. [xtzwhj myyu://js.bnpnujinf.twl/bnpn/ Hfjxfw_hnumjw]

553 characters, 2 lines

0 %

Ciphertext

Frequen...

0 %

FrequencyTest

Ca...

0 %

CaesarAnalysis

Ca...

0 %

Caesar Decryp

T...

5

1 character, 1 line

0 %

Key

Text Output

In cryptography, a Caesar cipher, also known as Caesar's cipher, the shift cipher, Caesar's code or Caesar shift, is one of the simplest and most widely known encryption techniques. It is a type of substitution cipher in which each letter in the plaintext is replaced by a letter some fixed number of positions down the alphabet. For example, with a shift of 3, A would be replaced by D, B would become E, and so on. The method is named after Julius Caesar, who used it in his private correspondence. [source http://en.wikipedia.org/wiki/ Caesar_cipher]

553 characters, 2 lines

0 %

Deciphered Ciphertext

Parameter

FrequencyTest

Frequency Test

Enter the length of the gramms to be investigated.

1

Handling of unknown characters

Don't count

Alphabet case sensitivity

Case insensitive

Word boundary fragments

No fragments at bounda

Presentation

Autozoom

Chart height 160

Zoom

Caesar - statistical analysis

This sample performs a statistical analysis attack on the Caesar cipher. The character frequencies are analyzed and -- based on it -- the substitution done by the Caesar cipher is reverted.

How it works:

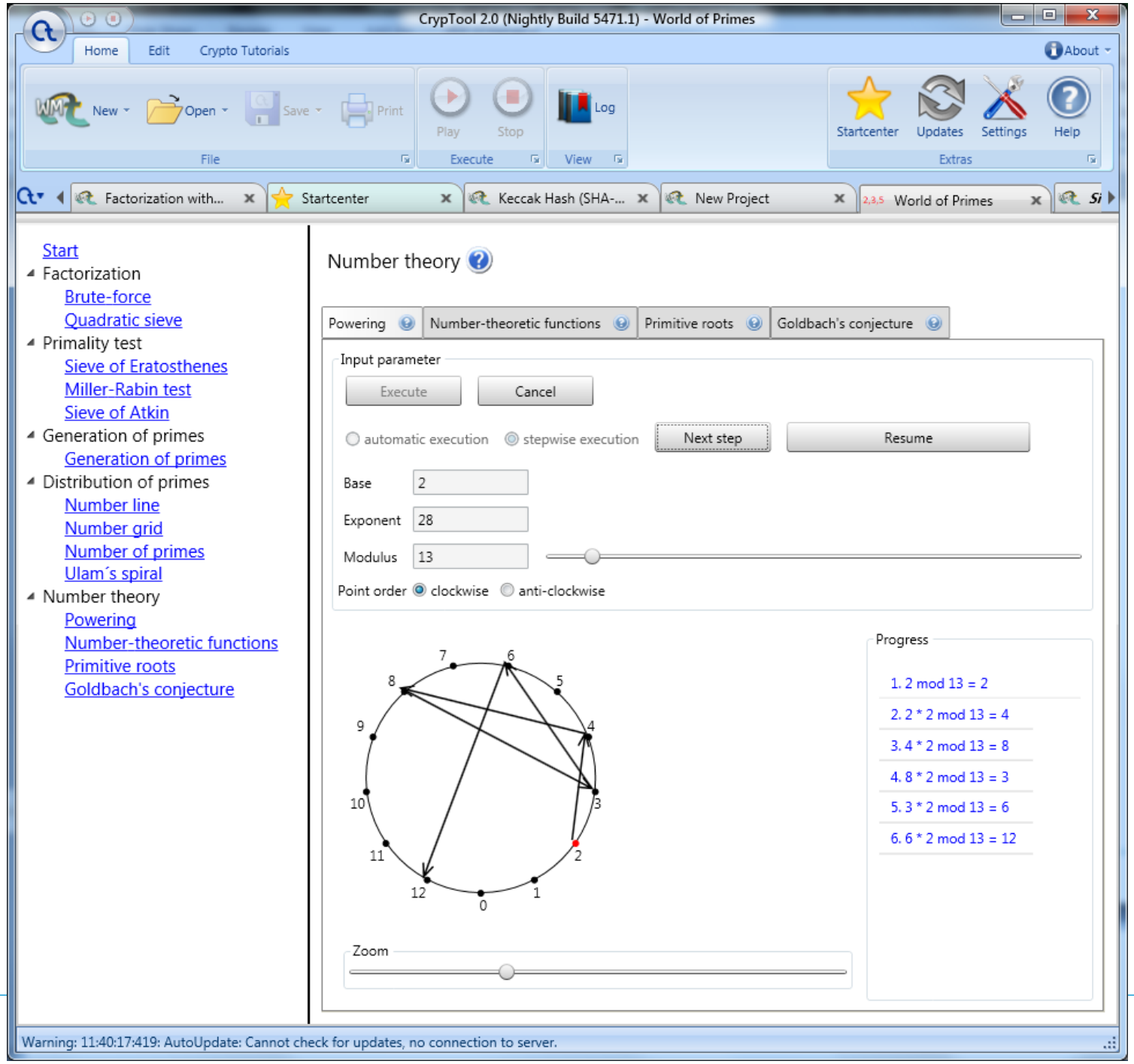
The encrypted text is forwarded to the FrequencyTest component. This component generates a bar chart of the character frequencies of the encrypted text and sends it to the CaesarAnalysisHelper component. This component performs the cryptanalysis of a Caesar cipher using the frequency of unigrams and bigrams in the encrypted text. The calculated shift key is finally given to a Caesar component to decrypt the encrypted text. The key may be also seen in the TextOutput "Key".

98 %

Info: 00:24:10:113: ExecutionEngine successfully stopped

Additionally: Quickly adapt the CT2 GUI with the keyboard using F11 and F12 by fading-in or fading-out parts outside the actual workspace

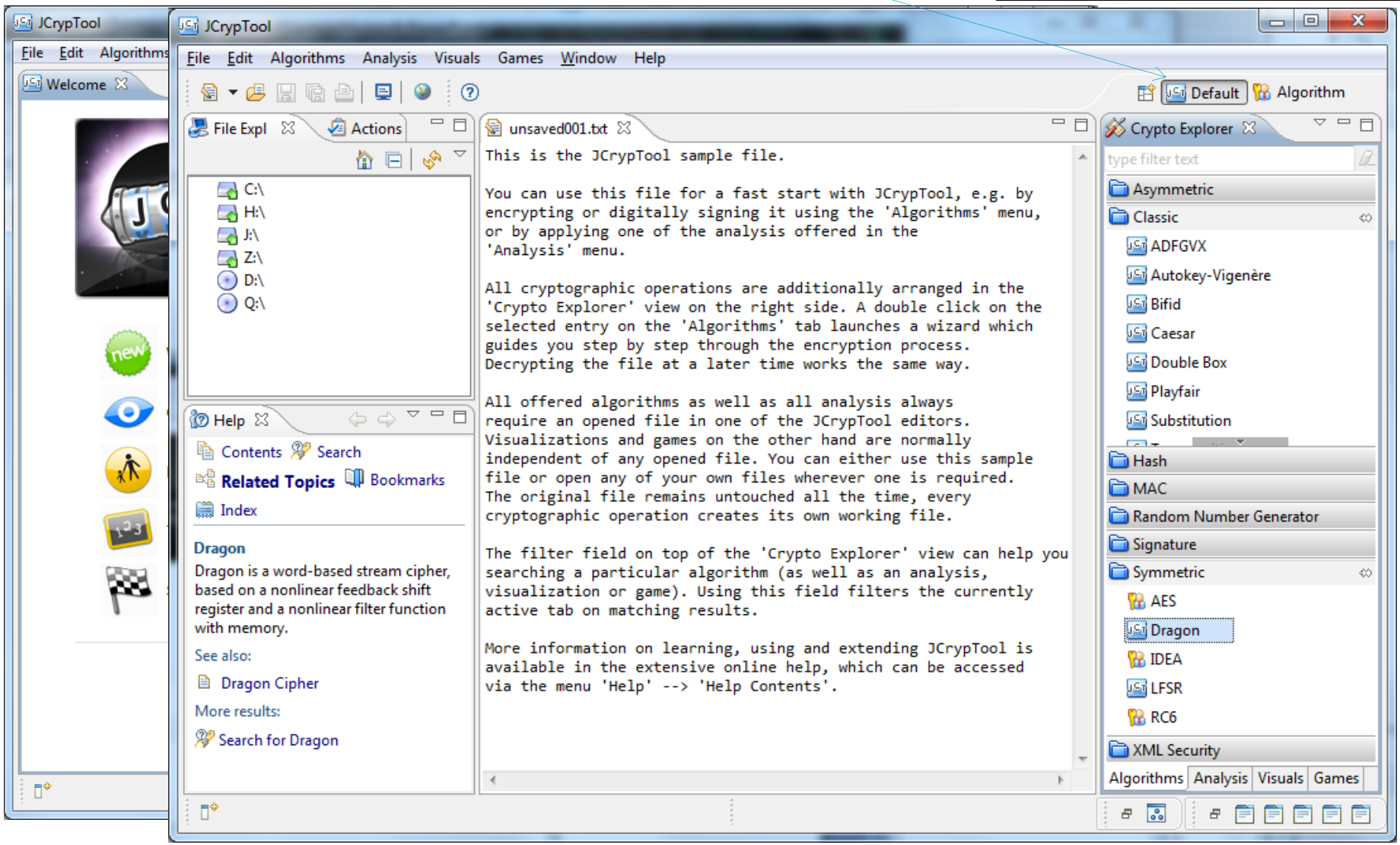
CT2


CrypTool 2.0 (Nightly Build 5471.1) - World of Primes
Home
Edit
Crypto Tutorials
About
New
Open
Save
Print
Play
Stop
Log
Startcenter
Updates
Settings
Help
File
Execute
View
Extras
Factorization with...
Startcenter
Keccak Hash (SHA-...
New Project
World of Primes
Start
Factorization
Brute-force
Quadratic sieve
Primality test
Sieve of Eratosthenes
Miller-Rabin test
Sieve of Atkin
Generation of primes
Generation of primes
Distribution of primes
Number line
Number grid
Number of primes
Ulam´s spiral
Number theory
Powering
Number-theoretic functions
Primitive roots
Goldbach's conjecture
Number theory
Powering
Number-theoretic functions
Primitive roots
Goldbach's conjecture
Input parameter
Execute
Cancel
automatic execution
stepwise execution
Next step
Resume
Base
2
Exponent
28
Modulus
13
Point order
clockwise
anti-clockwise
Progress
1. 2 mod 13 = 2
2. 2 * 2 mod 13 = 4
3. 4 * 2 mod 13 = 8
4. 8 * 2 mod 13 = 3
5. 3 * 2 mod 13 = 6
6. 6 * 2 mod 13 = 12
Zoom
Warning: 11:40:17:419: AutoUpdate: Cannot check for updates, no connection to server.

JCT
www.cryptool.org/en

- JCrypTool [RC 8 (stable) and weekly builds, Release JCT 1.0 planned for end 2016]
  - Build with the free IDE Eclipse, RCP, and SWT, and using Java 8 as JRE
  - Available in English and German; runs on Windows, MacOS, and Linux
  - Build-in automatic upgrade mechanism

JCT – Welcome and **Default Perspective**



JCT – **Algorithm Perspective**

JCrypTool

Default | Algorithm

Keystore

JCrypTool Keystore

- Alice Whitehat
  - Certificates (Public Keys)
  - Key Pairs
    - CMSSwithSHA1andWinternitzOTS_1 (OI
    - CMSSwithSHA384andWinternitzOTS_1 (
    - CMSSwithSHA384andWinternitzOTS_4 (
    - DSA (OID: 1.3.14.3.2.12)Key Pair (key s
    - ElGamal (OID: 1.3.14.7.2.1.1)Key Pair (l
    - RSA (OID: 1.2.840.113549.1.1.1)Key Pa
    - RSA (OID: 1.2.840.113549.1.1.1)Key Pa
  - Secret Keys
    - AES, Rijndael (OID: 2.16.840.1.101.3.4
    - HmacMD5 (OID: 1.3.6.1.5.5.8.1.1) (key
    - IDEA (OID: 1.3.6.1.4.1.188.7.1.1) (key
    - MARS (key strength: 128)
    - RC6 (key strength: 128)

Help

Contents | Search | Related Topics | Bookmarks | Index

Operations

After selecting an algorithm in the **Algorithms** view, the corresponding entry will appear in the **Operations** view. This view allows you to enter the input source (either the text editor or a file) and the output destination (a file). Via drag and drop from the **Keystore** view, a key can be added to a specific operation. It is necessary to add a matching key before executing an algorithm.

To execute an algorithm, after all parameters are set correctly, click on the execute icon in the upper right corner of the Operations view.

More results:

Search for Keystore view

unsaved001.txt

```
This is the JCrypTool sample file.

You can use this file for a fast start with JCrypTool, e.g. by
encrypting or digitally signing it using the 'Algorithms' menu,
or by applying one of the analysis offered in the
'Analysis' menu.

All cryptographic operations are additionally arranged in the
'Crypto Explorer' view on the right side. A double click on the
selected entry on the 'Algorithms' tab launches a wizard which
guides you step by step through the encryption process.
Decrypting the file at a later time works the same way.

All offered algorithms as well as all analysis always
require an opened file in one of the JCrypTool editors.
Visualizations and games on the other hand are normally
independent of any opened file. You can either use this sample
file or open any of your own files wherever one is required.
The original file remains untouched all the time, every
cryptographic operation creates its own working file.

The filter field on top of the 'Crypto Explorer' view can help you
searching a particular algorithm (as well as an analysis,
visualization or game). Using this field filters the currently
active tab on matching results.

More information on learning, using and extending JCrypTool is
available in the extensive online help, which can be accessed
via the menu 'Help' --> 'Help Contents'.
```

Operations

Current Entry: (Shacal) no name Created on: Wed Apr 10 23:00:08 CEST 2013

- (Shacal) no name Created on: Wed Apr 10 23:00:08 CEST 2013
  - Algorithm: Shacal
    - Mode: CBC
    - Padding: PKCS5Padding
  - Input/Output
    - Input: <Editor>
    - Output: <Editor>
  - Key:
  - Operation: <not specified>

Algorithms

- Asymmetric Block Ciphers
  - ElGamal (OID: 1.3.14.7.2.1.1)
  - McElieceCPKCS (OID: 1.3.6.1.4.1.8301.3.1.3.4.1)
  - MeRSA
  - MpRSA
  - Niederreiter
  - RSA_PKCS1_v1_5 (OID: 1.2.840.113549.1.1.1)
  - RSA_PKCS1_v2_1 (OID: 1.2.840.113549.1.1.7)
- Block Ciphers
  - Camellia
  - DESede
  - IDEA (OID: 1.3.6.1.4.1.188.7.1.1)
  - MARS
  - Misty1
  - RC2
  - RC5
  - RC6
  - Rijndael
  - SAFER+
  - SAFER++
  - Serpent (OID: 1.3.6.1.4.1.11591.13.2)
  - Shacal
  - Shacal2
  - Twofish
- Hybrid Ciphers
  - ECIES
  - McElieceFujisakiCipher (OID: 1.3.6.1.4.1.8301.3.1.3.4.2.1)
  - McElieceKobaraImaiCipher (OID: 1.3.6.1.4.1.8301.3.1.3.4.2.3)
  - McEliecePointchevalCipher (OID: 1.3.6.1.4.1.8301.3.1.3.4.2.2)
- Message Authentication Codes
  - CBCMac
  - CMac
    - CmacAES128
    - CmacAES192
    - CmacAES256
    - CmacDESede
  - HMac
    - Hmac (OID: 1.3.6.1.5.5.8.1.2)
    - HmacDHA256
    - HmacFORK256
    - HmacMD4
    - HmacMD5 (OID: 1.3.6.1.5.5.8.1.1)

JCT Features

- Platform independent
  - A crypto plugin developer doesn't have to care for the other operating systems. He/she just develops on his platform.
- Two perspectives (Default and Algorithm)
- Two crypto providers (FP and Bouncy Castle)
- Text and hex editor
- Cascading of ciphers
- Action history
- Common key store used by all modern plugins.
  It stores secret and public keys, certificates and some meta data.

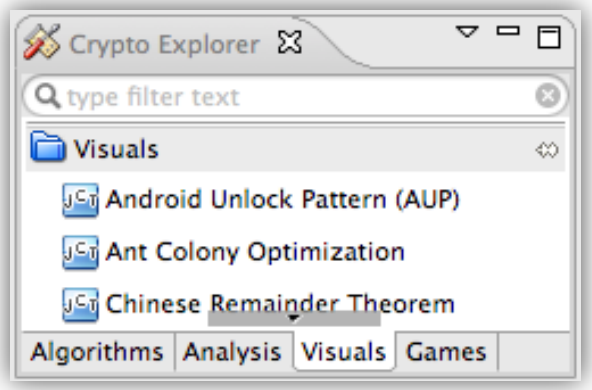
These features plus a modern GUI are offered by JCT.

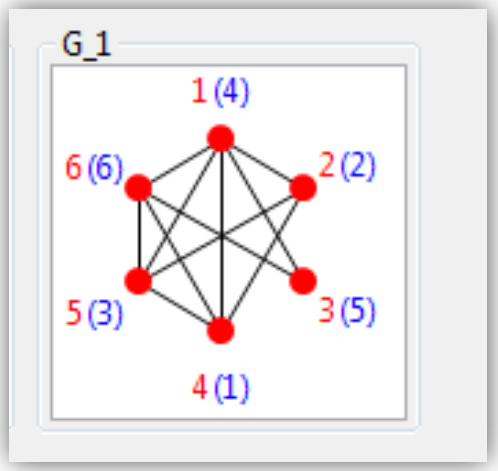
The crypto-plugin developer decides what to use.

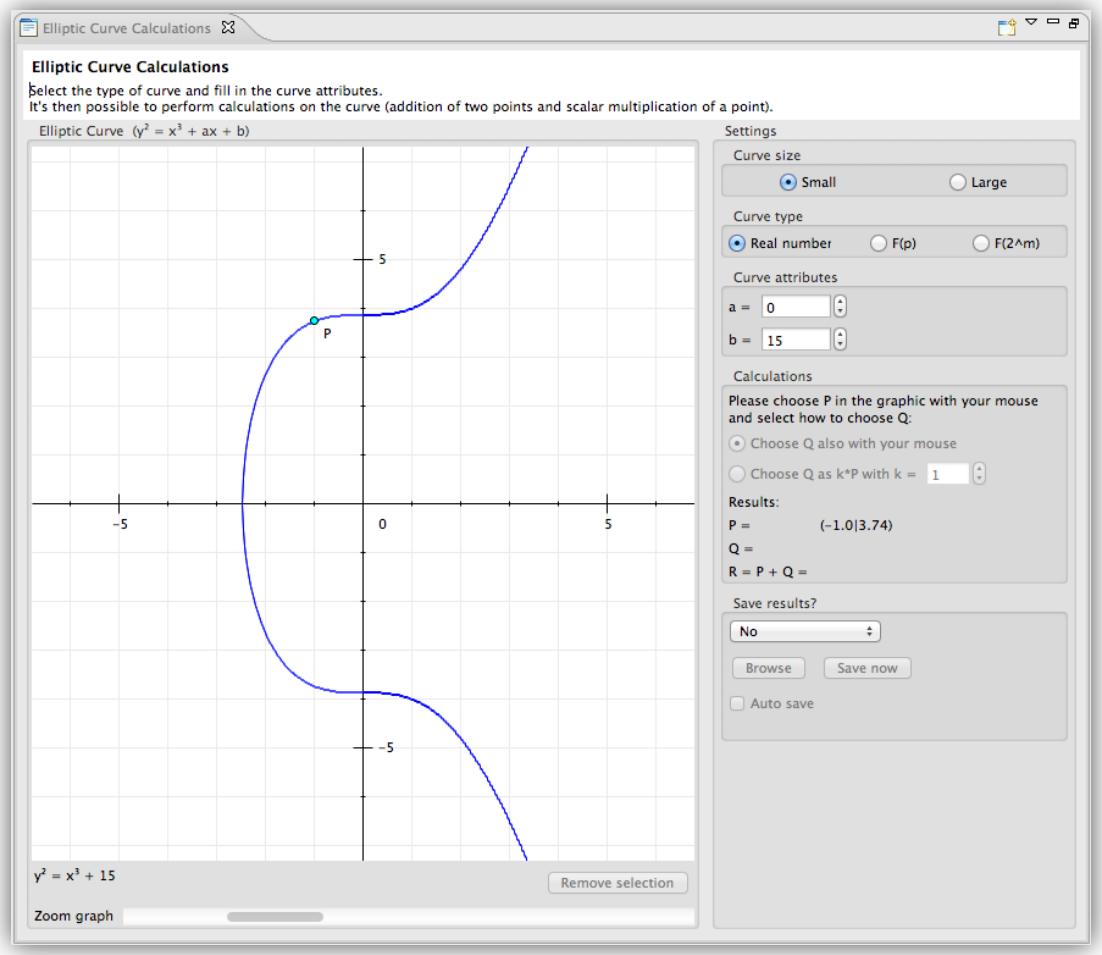
- Work in progress or planned: build multi-tree variant of MSS, connect existing plugins, brute-force search, graphical interface for Bouncy Castle in JCT, …
- Link with information for developing new plugins:
  https://github.com/jcryptool/core/wiki

Demo

JCT


Crypto Explorer
type filter text
Visuals
Android Unlock Pattern (AUP)
Ant Colony Optimization
Chinese Remainder Theorem
Algorithms
Analysis
Visuals
Games


G_1
1 (4)
2 (2)
3 (5)
4 (1)
5 (3)
6 (6)


Elliptic Curve Calculations
Elliptic Curve Calculations
Select the type of curve and fill in the curve attributes.
It's then possible to perform calculations on the curve (addition of two points and scalar multiplication of a point).
Elliptic Curve (y² = x³ + ax + b)
Settings
Curve size
Small
Large
Curve type
Real number
F(p)
F(2^m)
Curve attributes
a = 0
b = 15
Calculations
Please choose P in the graphic with your mouse and select how to choose Q:
Choose Q also with your mouse
Choose Q as k*P with k = 1
Results:
P = (-1.0|3.74)
Q =
R = P + Q =
Save results?
No
Browse
Save now
Auto save
y² = x³ + 15
Remove selection
Zoom graph

JCT Information for Developers

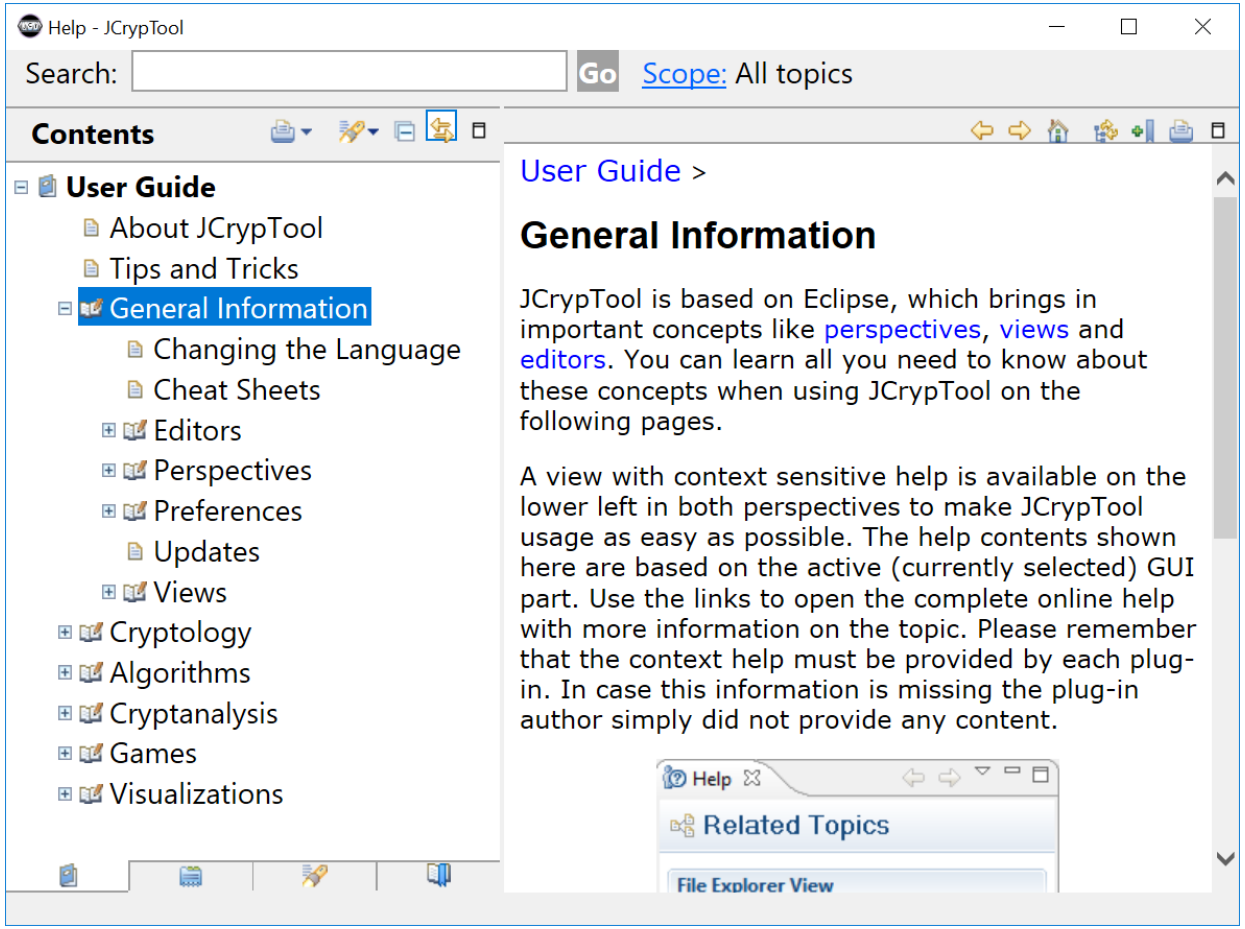
Wiki: https://github.com/jcryptool/core/wiki

Style guide: https://github.com/jcryptool/doc/blob/master/Guidelines/JCrypTool-GUI-Guidelines.pdf

Information for both, the **core** and the **plugin** developers, is provided in the **Wiki**.

JCT-plugin developers should not need any projects from the JCT repository but need to run JCT as a target platform and develop for it.

Included in JCT itself is the **User Guide (**see screenshot here) – in analogy to Eclipse.



Agenda

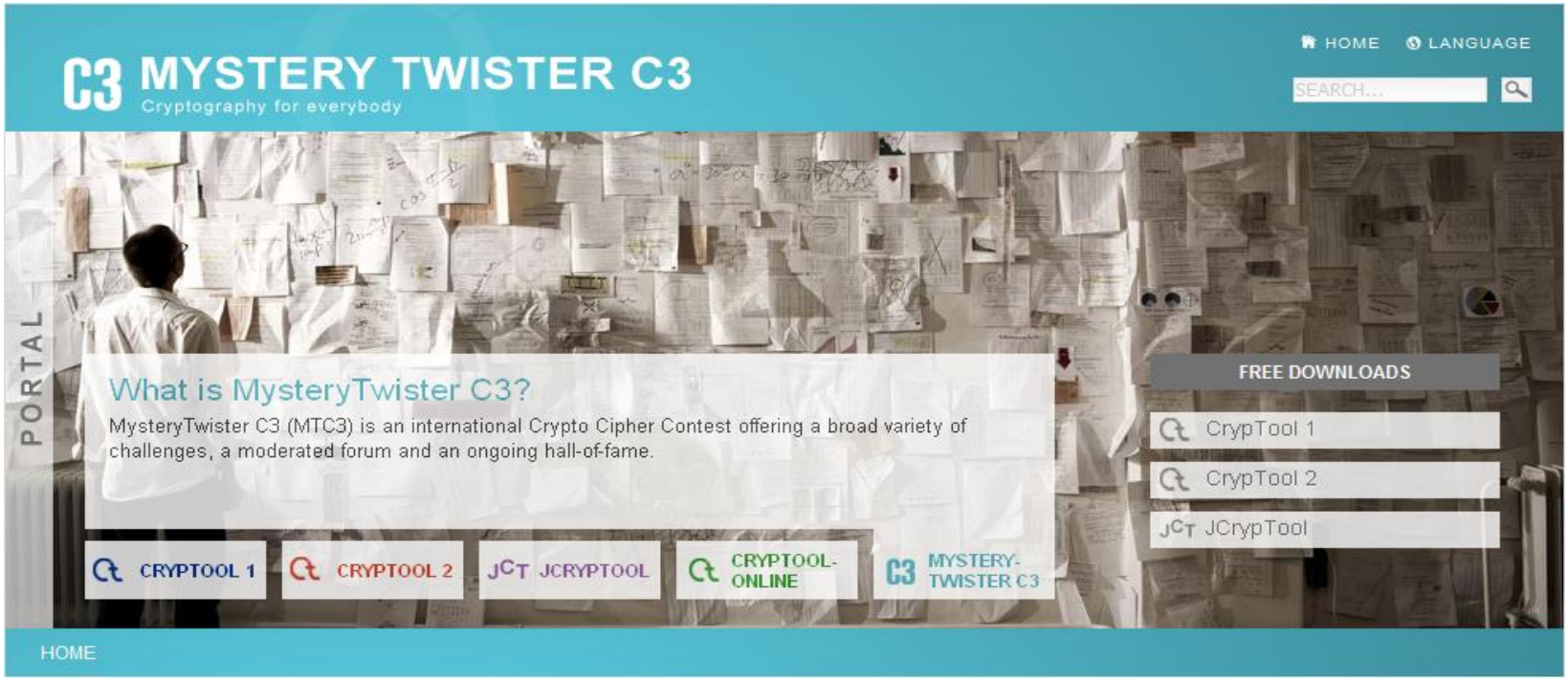
Online Resource: CTO
www.cryptool.org/en/cryptool-online

- CrypTool-Online
  - CrypTool within a browser (running on a PC or on a smart phone)
  - Currently, backend gets new infrastructure with Angular 2, Bootstrap, and Joomla 3.6
  - Available in English and German

CTO: http://www.cryptool-online.org

Online Resource: MTC3 – The Cipher Contest
www.cryptool.org/en/mtc3

- MysteryTwister C3 (MTC3)
  - International Crypto Cipher Contest
  - Available in English and German
  - Currently more than 200 challenges, created by more than 40 different authors



MTC3: http://www.mysterytwisterc3.org/

CrypTool 1 - CrypToo... | Start | CrypTool - Wikipedia,... | JCrypTool · GitHub | actual - Englisch - Deutsc...

https://www.mysterytwisterc3.org/en/

MysteryTwister C3

THE CRYPTO CHALLENGE CONTEST

NUMBER OF ACTIVE MEMBERS:

7535

Register here

MTC3 PARTNERS

Follow us:

Search... All Search!

Start | Challenges | Forum | MysteryTwister I

Login DE EN

About MTC3 | Partners | News

FOUR LEVELS OF CHALLENGES

MysteryTwister C3 offers something for everybody by featuring four levels of challenges, from pen-and-paper riddles to highly sophisticated mathematical mysteries.

Register here

Welcome to MTC3 — The Cipher Contest

You like riddles? You always loved to solve the crosswords in your newspaper? Or maybe you are just curious and want to find out about some of the

+++ [08:09 - 20.09.2016] Norbert solved the Level II challenge 'Smartcard RSA' +++ [16:40 - 19.09.2016] Bakuz solved the Level I challenge 'Number Sequence

MTC3:

leonhard euler - Google-S | CRYPTO 2013 | Finse winter school 2013 | Overall Hall-of-Fame

www.mysterytwisterc3.org/en/challenges/overall-hall-of-fame?time=month&complete=0&end=current

Start | Challenges | Forum | MysteryTwister I | Login | DE | EN

The four levels | Level I | Level II | Level III | Level X | Challenges Hall-of-Fame | Overall Hall-of-Fame | Submit a challenge

Level I	Level II	Level III	Level X
35 / 50 solved	14 / 51 solved	0 / 41 solved	0 / 11 solved

Overall Hall-of-Fame (This month)

The Overall Hall-of-Fame contains the sum of all achieved points of all solved challenges for all users. You will get at least 100 points for a level I challenge, 1,000 points for a level II challenge, and 10,000 points for a level III challenge (minimum points per challenge). As closer to the date it was published you solve it as more points you'll get: The maximum is the double of the minimum points when you send in the correct solution within a day after the publishing date. If you solve a challenge some weeks after it was published you will only get about 110 % of the minimum points. The points will be fewer every day, but will never fall below 100 % of the minimum points.

If you want to know more on how the points are calculated, take a look at the formula shown at the end of the Overall Hall-of-Fame table.

Using the drop-down list at the right side on top of the following table you can select the displayed time frame of the Overall Hall-of-Fame.

Time frame: view from 2013-04-01 to 2013-04-16

This month

Rank	User (#57)	#Level I (#172)	#Level II (#26)	#Level III (#0)	#Level X (#0)	Points (46,875)
	Velko Nikolov (staafl)	15	5	0	0	6,511
	mk (bilbobeutlin)	6	3	0	0	3,600

[18:08 - 15.04.2013] rocsci solved the Level I challenge 'Letter to the Templars — Part 1' +++ [18:08 - 15.04.2013] snk solved the Level I challenge 'One-Time Pad with Flaws' +++

MTC3:

Level I - Mozilla Firefox

Datei Bearbeiten Ansicht Chronik Lesezeichen Extras Hilfe

Overall Hall... | Welcome to... | MTC3 Statis... | Level I | Level II | Level III | M... | Level X | My... | Forum | My... | Partners

http://www.mysterytwisterc3.org/index.php/en/crypto-challenges/level-1-cryptography-challe

Level I challenges (19)

All challenges in Level I, ordered by date posted (the most recent appear first).

Music Code - Part 1

[wolter-01] - 4 users already solved this challenge, 3 users are working on it.

A piece of music is given as mp3 file and in music notation. The notes represent a secret message, which was encrypted with two classic methods. In part 1 of this challenge you have to solve the first of these methods. Thereafter, you can set out to solve part 2 (which is in level 2). Read more...

Click here to get to the corresponding forum topic to share your opinion.
Click here to download the challenge.
Click here to download the additional file of the challenge.

Revealed the secret of the challenge? Check if it's correct: Submit

Post-Quantum Cryptography: Unbalanced Oil and Vinegar System - Part 1

[wolf-01] - 5 users already solved this challenge, 1 user is trying it.

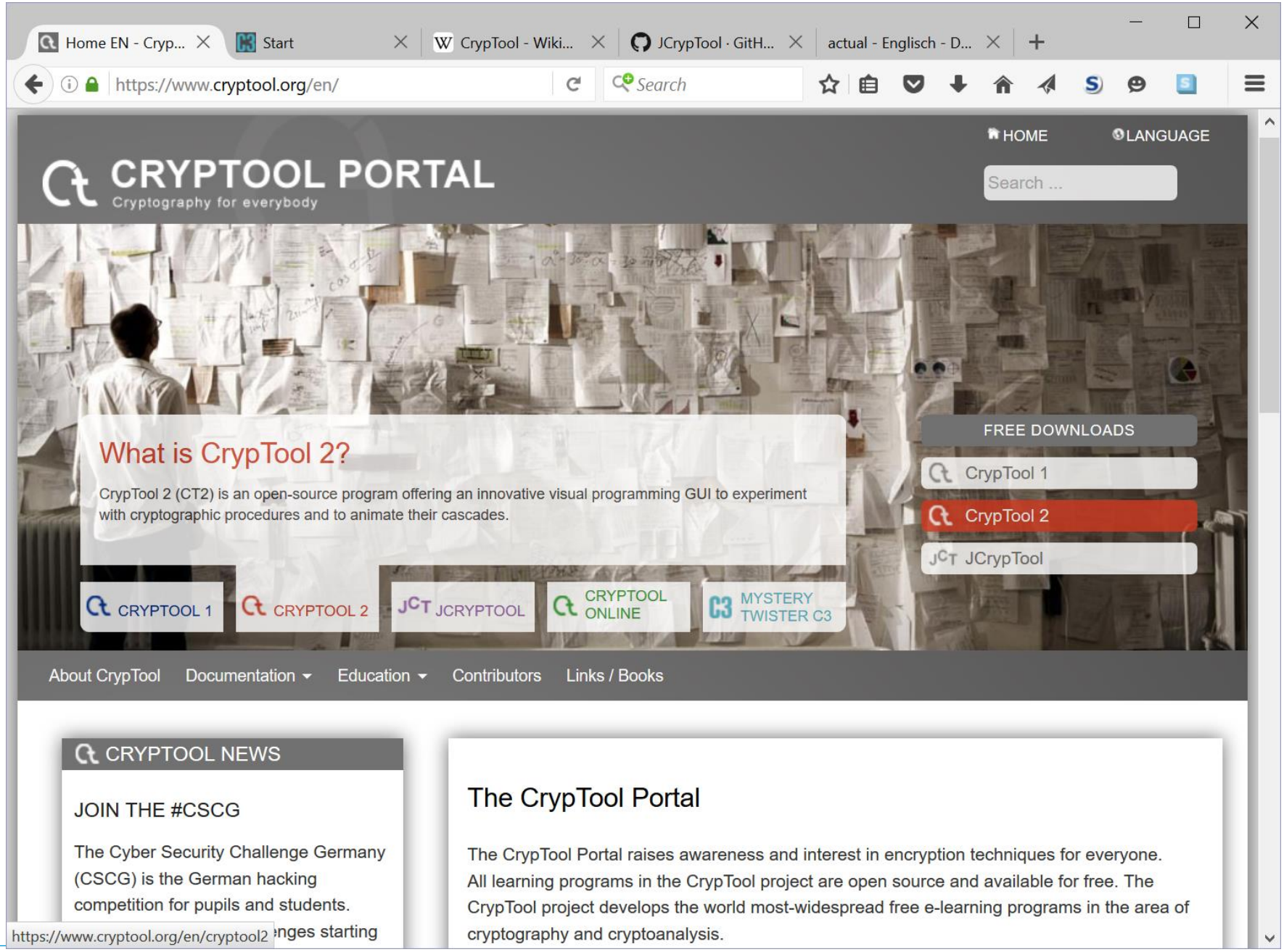
By the time quantum computers exist, the actual signing algorithms are not secure anymore. Then one can switch e.g. to so-called "Unbalanced Oil and Vinegar" systems. It is your challenge to break a simplified version of such a system today already. Read more...

Click here to get to the corresponding forum topic to share your opinion.
Click here to download the challenge.

11.06.2011] Wombat has solved the Level II challenge 'Music Code - Part 2' +++

Suchen: romleo Abwärts Aufwärts Hervorheben Groß-/Kleinschreibung

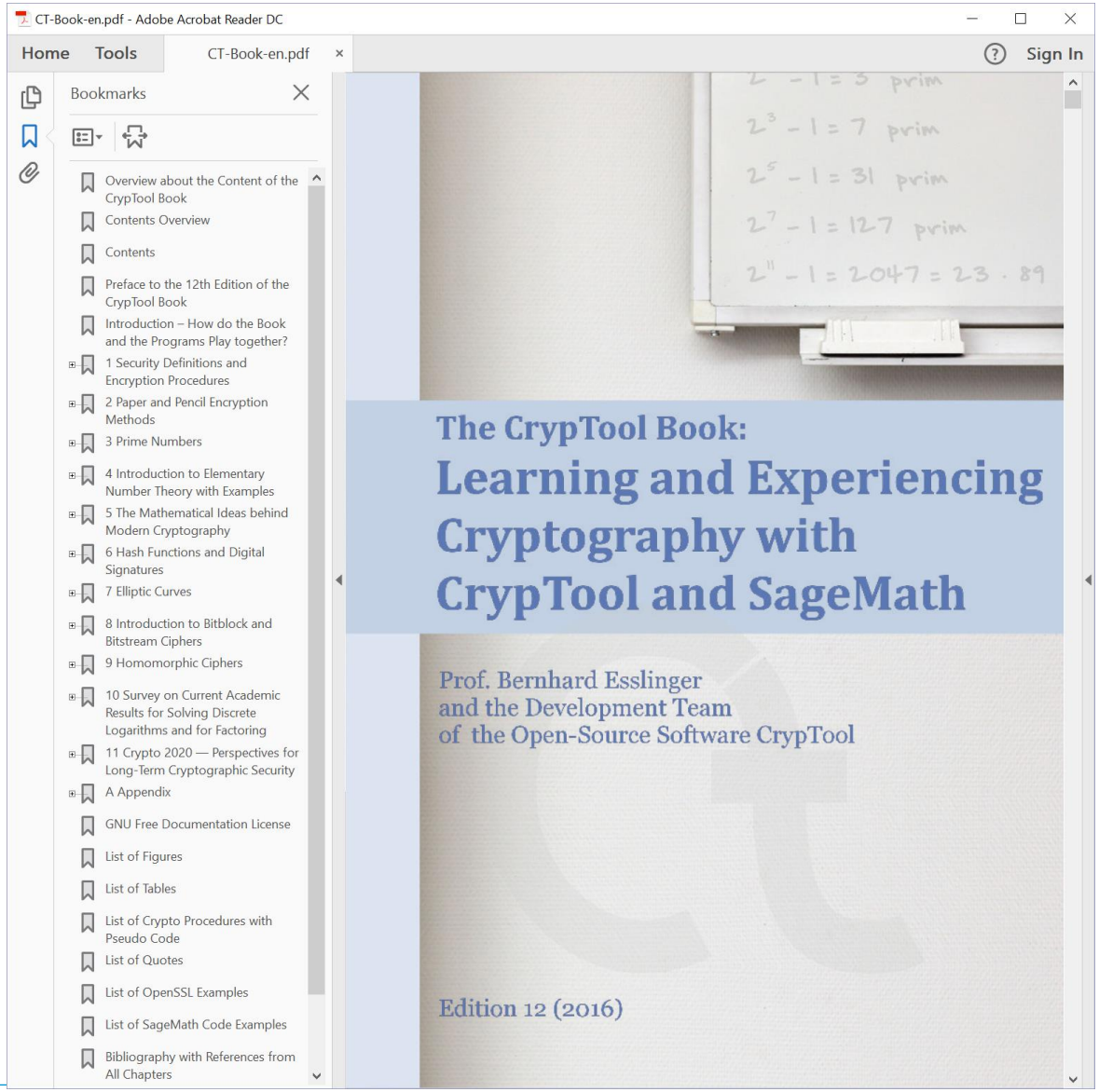
CTP:


Home EN - Cryp...
Start
CrypTool - Wiki...
JCrypTool · GitH...
actual - Englisch - D...
https://www.cryptool.org/en/
Search
HOME
LANGUAGE
CRYPTOOL PORTAL
Cryptography for everybody
Search ...
What is CrypTool 2?
CrypTool 2 (CT2) is an open-source program offering an innovative visual programming GUI to experiment with cryptographic procedures and to animate their cascades.
FREE DOWNLOADS
CrypTool 1
CrypTool 2
JCrypTool
CRYPTOOL 1
CRYPTOOL 2
JCRYPTOOL
CRYPTOOL ONLINE
MYSTERY TWISTER C3
About CrypTool
Documentation
Education
Contributors
Links / Books
CRYPTOOL NEWS
JOIN THE #CSCG
The Cyber Security Challenge Germany (CSCG) is the German hacking competition for pupils and students.
https://www.cryptool.org/en/cryptool2
The CrypTool Portal
The CrypTool Portal raises awareness and interest in encryption techniques for everyone. All learning programs in the CrypTool project are open source and available for free. The CrypTool project develops the world most-widespread free e-learning programs in the area of cryptography and cryptoanalysis.

CrypTool Book: Background Information (some more maths)

On the CTP website:

- Menu path:
  "Documentation" → "CT Book"
- http://www.cryptool.org/en/ctp-documentation/ctbook
- Edition 12 from 2016



How to Search for a Specific Crypto Functionality within CrypTool
http://www.cryptool.org/en/ctp-documentation/ctp-functions

On the CTP website:

- Filter on the CrypTool portal for the currently around 500 different functions (from all CT versions)
- Menu path: "Documentation" → "CT Functions"

Plus within the offline programs:

- Online help search
- Filters within CT2 and JCT

Cryptological functions in different CrypTool versions

Selection

Cryptographic category: No filter applied

Additional search phrase: fac

CrypTool 1 (CT1) ☑ CrypTool 2 (CT2) ☑ JCrypTool (JCT) ☐ CrypTool Online (CTO) ☐

4 rows found according to the selection criteria.

Function	CT1	CT2	CT 1 Path	CT 2 Path
Factorization of a Number	X	C\T\W\N	Indiv. Procedures\ RSA Cryptosystem\ Factorization of a Number...\ Brute-force Indiv. Procedures\ RSA Cryptosystem\ Factorization of a Number...\ Brent Indiv. Procedures\ RSA Cryptosystem\ Factorization of a Number...\ Pollard Indiv. Procedures\ RSA Cryptosystem\ Factorization of a Number...\ Williams Indiv. Procedures\ RSA Cryptosystem\ Factorization of a Number...\ Lenstra Indiv. Procedures\ RSA Cryptosystem\ Factorization of a Number...\ Quadratic Sieve	[C] Cryptanalysis\ Generic\ Factorizer [C] Cryptanalysis\ Generic\ Quadratic Sieve [T] Mathematics\ Factorization with Trial Division (brute-force) [T] Mathematics\ Factorization with General Number Field Sieve [W] Start\ Mathematical Functions\ Prime Factorization [N] Crypto Tutorials\ World of Primes\ Factorization\ Brute-force [N] Crypto Tutorials\ World of Primes\ Quadratic Sieve
General Number Field Sieve (GNFS)		C\T		[C] Cryptanalysis\ Generic\ General Number Field Sieve [T] Mathematics\ Factorization with General Number Field Sieve (GNFS)
Prime Number Tutorial	X	N	Indiv. Procedures\ RSA Cryptosystem\ Prime Number Test... Indiv. Procedures\ RSA Cryptosystem\ Generate Prime Numbers... Indiv. Procedures\ RSA Cryptosystem\ Factorization of a Number...	[N] Crypto Tutorials\ World of Primes
Quadratic Sieve (QS)		C\T		[C] Cryptanalysis\ Generic\ Quadratic Sieve (QS) [T] Mathematics\ Factorization with Quadratic Sieve

Agenda

Sample Contributions

University	CT	Plugin
Hagenberg, Eindhoven	JCT	Post-quantum series: WOTS, Merkle signature schemes
Kassel	CT2	Quantum key-exchange protocol BB84
Utrecht	JCT	Elliptic curve calculations over R, F(p), and F(2^m)
Hagen	JCT	Inner states of DES
Frankfurt, Darmstadt	JCT	Kleptography (4 attacks implemented)
Kassel, Belgrade	CT2	Network communication, Chat
Bochum	CT2	Keccak for hashing (SHA3), as PRNG and as stream cipher
Frankfurt	CT2	Padding-oracle attack
Kassel	CT2	Heartbleed attack against 2 life servers
Kassel, Duisburg	CT2	CrypCloud – distributed computing
Bochum	CT2	SAT solver (analyzer at work, problem with port from Unix)
Brno	CT2	Protocols like oblivious transfer

JCrypTool

File Edit Algorithms Analysis Visuals Games Window Help

Default Algorithm

Winternitz OT-Signature (WOTS | MerkleTree | Extended Euclidean / Reciproca | Certificate Verification | Shanks Babystep-Giantstep | Huffman Coding

Seed and Key Generation | MerkleTree | Message Signing | Verification

Merkle Signature Scheme (MSS)

- Merkle Signature Scheme (MSS)
- eXtended Merkle Signature Scheme (XMSS)
- XMSS Multi Tree (XMSS^MT)

eXtended Merkle Signature Scheme

The eXtended Merkle Signature Scheme (XMSS) is a Scheme, that extends the Merkle Signature Scheme. One Advantage are smaller as in the MSS. An other Difference to the MSS is the Usage of the Bitmask Q and the L-Tree. With this two Extensions, the used collision resistant hash function can be replaced by a second preimage resistant hash function.

Seed 444402438794322819 Generate new Seed

Generate Key Pairs

Merkle Signature Scheme uses the Winternitz One Time Signature (WOTS) Scheme in the Seedmode, which generates the Keys with the help of a Seed. One must use 2^n keys because else there are problems with the verification of Merkle signatures. For more information please consult the help menu.

Number of Keys 8 Generate Keys

JCrypTool

File Edit Algorithms Analysis Visuals Games Window Help

Default Algorithm

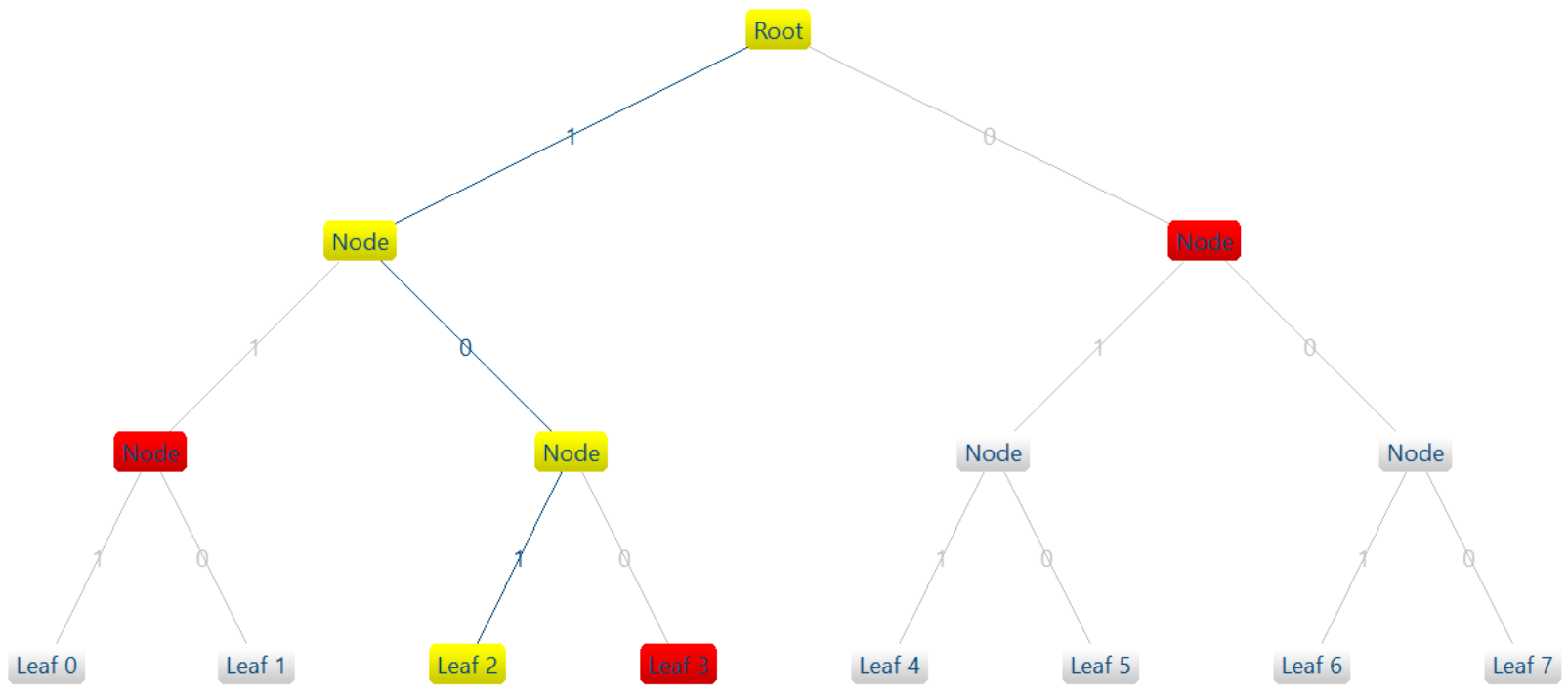
Winternitz OT-Signature (WOTS | MerkleTree | Extended Euclidean / Reciproca | Certificate Verification | Shanks Babystep-Giantstep | Huffman Coding

Seed and Key Generation | MerkleTree | Message Signing | Verification

Here you can see the Signature. The signature will only be verified correct, when the correct Node is chosen. Otherwise the verification will fail.

Please click Verify to check the chosen Node.

Verify



Test

JCrypTool

File Edit Algorithms Analysis Visuals Games Window Help

Default Algorithm

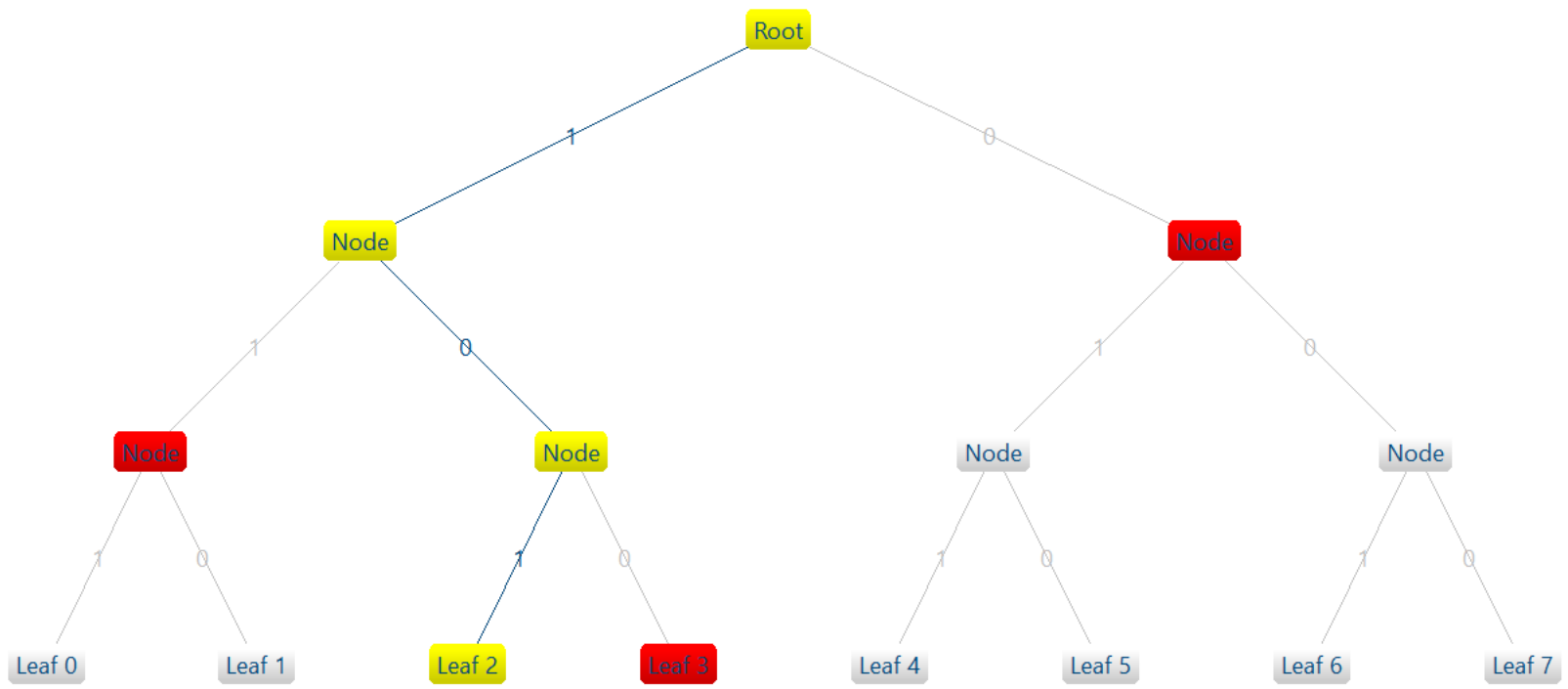
Winternitz OT-Signature (WOTS | MerkleTree | Extended Euclidean / Reciproca | Certificate Verification | Shanks Babystep-Giantstep | Huffman Coding

Seed and Key Generation | MerkleTree | Message Signing | Verification

Here you can see the Signature. The signature will only be verified correct, when the correct Node is chosen. Otherwise the verification will fail.

The verification of the signature with the chosen Node was successful.

Verify



Hash = (82D6AE28F907BC7BDF1D31C44CD3EAB5384A6DE768A9DB1584D2E5C4CF15964C)

Test

JCT

JCrypTool

File Edit Algorithms Analysis Visuals Games Window Help

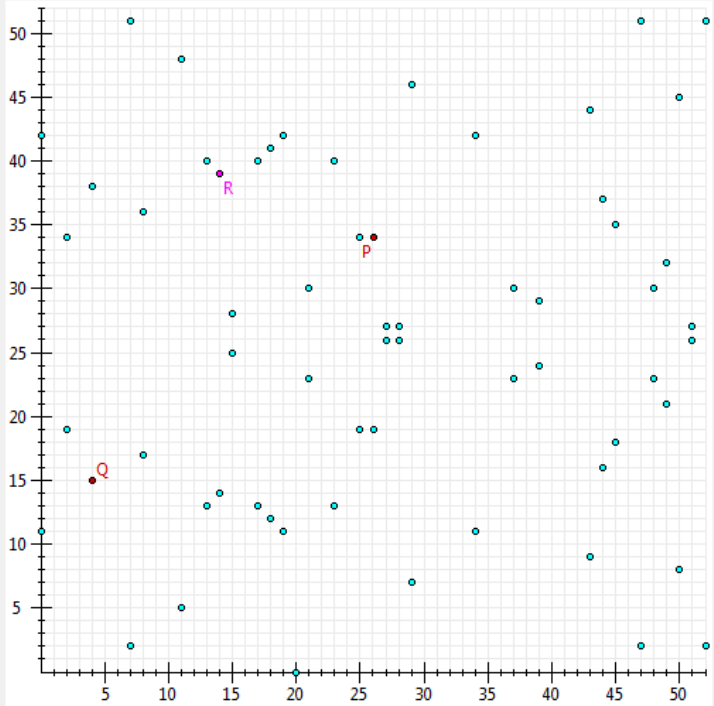
Default Algorithm

Diffie-Hellman Key Exchang | Magic Door | Extended RSA Cryptosystem | Verifiable Secret Sharing | Inner States of the Data Enc | Elliptic Curve Calculations

Elliptic Curve Calculations

Select the type of curve and fill in the curve attributes.
It's then possible to perform calculations on the curve (addition of two points and scalar multiplication of a point).

Elliptic Curve (y^2 mod p = (x^3 + ax + b) mod p)



y^2 mod 53 = (x^3 + 10x + 15) mod 53

Remove selection

Zoom graph

Points (64)

O	R(14|39)	(26|19)	(44|16)
(0|11)	(15|25)	P(26|34)	(44|37)
(0|42)	(15|28)	(27|26)	(45|18)
(2|19)	(17|13)	(27|27)	(45|35)
(2|34)	(17|40)	(28|26)	(47|2)
Q(4|15)	(18|12)	(28|27)	(47|51)

Settings

Curve size: Small / Large

Curve type: Real numbers / F(p) / F(2^m)

Curve attributes: a = 10, b = 15, p = 53

Calculations

Please choose P in the graphic with your mouse and select how to choose Q:

Choose Q also with your mouse

Choose Q as k*P with k = 1

Results:

P = (26|34)
Q = (4|15)
R = P + Q = (14|39)

Save results?

No

Browse Save now

Auto save

JCrypTool

File Edit Algorithms Analysis Visuals Games Window Help

Default Algorithm

Android Unlock Pattern | Extended RSA Cryptosystem | Elliptic Curve Calculations | Inner States of the Data Encryption Standard (DES)

Algorithm Study | Anti- / Fixed Point Study | S-Box Study

Information

Different aspects of the encryption or decryption process of DES are visualized.

Key:
The key k is used to encrypt or decrypt the data.

Output table "Roundciphers":
The table shows the intermediate round ciphers m[0]-m[17] for the process (en-/decryption).
For each column: Adjacent bit-colors change if adjacent bit-values change.

Output table "DES(k,p+e_i)":
For i = 1, ..., 64: Plaintexts p and p+e_i differ at position i by one bit.
Each DES(k; p+e_i) is presented and compared with DES(k,p) using the Hamming distance DIST as measure.

Output table "Distance Matrix":
Two matrices visualize Hamming distances
More information can be found on the tab.

Output Table "Roundkeys":
The table shows the 16 round keys.

Output table "CD Matrix":
Round key k_i is generated from C[i], D[i] by cyclic operations combined with specific bit-elections.

For more information please consult the documentation.

Input

Mode: (•) Encrypt ( ) Decrypt

Key: (•) k[0] ( ) k[3] ( ) k[5] ( ) k[6] ( ) Manual key (16 hexdigits): ( ) k[9] ( ) k[10] ( ) k[12] ( ) k[15] (0)

Data: Plaintext (16 hexdigits): 1111111111111111 (16)

Ciphertext: 89B07B35A1B3F47E

Output

Roundciphers | DES(k,p+e_i) | Distance Matrix | Roundkeys | CD Matrix

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	DIST
m[0]	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	
m[1]	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	16
m[2]	1	1	0	1	1	0	0	0	0	0	1	0	0	1	1	1	1	1	0	1	1	0	1	1	0	1	0	0	0	0	1	1	17
m[3]	1	0	0	1	1	1	0	0	0	1	1	1	0	1	0	0	0	1	0	0	0	0	0	1	0	0	0	1	0	0	0	1	13
m[4]	1	1	1	1	1	1	0	0	1	0	1	1	1	0	1	0	1	0	1	0	0	0	0	1	1	0	0	0	0	0	0	1	12
m[5]	0	0	1	0	0	1	0	0	1	1	0	1	0	1	1	1	1	1	1	1	1	1	0	0	1	0	1	1	1	1	1	0	20
m[6]	1	0	0	1	1	1	0	0	0	0	0	0	0	0	1	1	0	1	1	0	1	0	0	1	0	1	1	1	1	0	1	0	15
m[7]	1	0	1	1	1	1	1	0	1	0	0	0	1	1	1	1	0	0	0	1	0	0	1	1	0	0	1	1	0	0	1	0	12
m[8]	0	1	0	1	0	1	1	0	1	1	0	1	1	0	0	1	0	1	0	1	1	0	1	1	1	0	0	0	0	1	0	1	16
m[9]	0	0	1	1	0	0	1	1	1	0	1	0	1	0	0	1	1	0	1	1	1	1	0	0	1	0	0	0	1	1	1	1	15
m[10]	1	1	1	1	0	1	1	1	0	0	1	0	1	0	1	0	0	0	0	0	0	1	1	0	1	0	1	0	1	0	0	0	15
m[11]	0	0	1	0	1	1	0	1	1	0	1	1	1	0	0	0	0	1	0	1	1	1	1	1	0	1	1	0	1	1	0	1	16
m[12]	0	1	0	1	0	0	1	1	1	1	0	0	0	1	1	0	0	1	0	1	0	0	1	0	0	1	1	0	0	1	0	1	16
m[13]	1	0	1	1	0	0	0	0	1	0	0	1	1	0	1	1	1	0	1	1	0	1	0	1	1	0	0	0	0	1	0	0	20
m[14]	1	1	1	1	1	0	1	1	0	0	0	0	1	0	0	0	1	0	1	1	0	1	1	1	0	0	1	1	1	1	0	0	13
m[15]	1	1	0	1	0	1	0	1	0	0	1	1	1	1	0	1	1	0	0	0	1	1	0	1	0	0	0	1	0	1	0	1	15
m[16]	0	1	1	1	0	0	1	1	1	1	1	1	1	1	1	0	1	0	0	0	0	1	0	1	1	0	1	0	0	1	0	0	13

Status

2013-04-24 00:58:54 Data Evaluation: Mode=Encrypt, Key=K[0], Data=1111111111111111

Reset | Evaluate

JCT

JCrypTool

File Edit Algorithms Analysis Visuals Games Window Help

Default Algorithm

ARC4 / Spritz | Kleptography

Key generation and encryption | SETUP attack

Kleptographic attacks on RSA

Kleptography is the technique of stealing information securely and subliminally by introducing an asymmetric backdoor in a cryptographic system. This can be accomplished in an RSA cryptosystem by using a subliminal channel to leak encrypted data that will allow an attacker to easily factor a given RSA modulus. When executed correctly in a black-box device, the encrypted output will appear perfectly normal and even a successful reverse-engineering of the system will not reveal the leaked data. It may reveal the presence of the attack, but the private key of the attacker will not be revealed. This implementation uses RSA and includes four different malicious methods of generating keys that an attacker could use to compromise a device. This topic shows how important it is to use truly randomly generated keys.

Key generation

Settings

Method: Attack 4: SETUP

Binary Decimal Hexadecimal

Key bit length: 64 (in decimal)

Additional cryptosystem values

Generate new attacker keys

Attacker's N: e4643b83

Attacker's E: 8bab9abd

Encrypted P: b356b562

N' (temporary composite): b356b5625cf9e581

Standard cryptosystem values

Generate all at once

Generate primes P and Q — P (prime): c9134767 — Q (prime): e45370b9

Calculate N — N = P * Q: b356b561a090a96f

E (public exponent): 67960daebc661227 — Generate random E — Restore default E

Calculate D — D (private exponent): 88270e838fcd1ba7

Dishonest key generation in RSA: SETUP attack

Step 1: A SETUP (secretly embedded trapdoor with universal protection) attack is designed to leak encrypted data out of a cryptographic device. In this example, it is assumed that the attacker has compromised the device to contain his or her own public key.

Step 2: The prime P will be generated randomly. It will then be encrypted with the attacker's public key. A temporary composite N' will be designed so that the upper half of the bits are equal to the encrypted P, and the rest is random. The second prime Q is calculated by division from the equation P*Q + R = N', where R is an unused remainder. Warning: this may take several seconds or even minutes for large key sizes!

Step 3: Multiply P and Q together to calculate the actual composite N. N is also equal to N' - R. Note that if R is large enough, the upper half of the bits in N may be equal to one less than those in N'. The attacker will later have to take this into account.

Step 4: Calculate Phi, select the public exponent E, and calculate the private exponent D by finding the modular inverse of E mod Phi. E and N together form the public key, and D and N form the private key.

Step 5: Enter a message below and encrypt it with the public key.

Plain- and ciphertexts

Original message (plaintext): this is a test message

Encrypted message (ciphertext): 7247fb1344b4742e2dd0c21024b7e3e49cea0fcfed03c84411 2065b879a254984e190ad3da1e496069ed5c5e6fb8f965

Decrypted ciphertext (plaintext): this is a test message

Encrypt | Decrypt | Save public key and ciphertext

Attack 4: SETUP

- Generate honest (random) primes
- Attack 1: Use fixed P
- Attack 2: Generate P via pseudo-random function
- Attack 3: Generate P via pseudo-random generator
- Attack 4: SETUP

JCT

JCrypTool

File Edit Algorithms Analysis Visuals Games Window Help

Default Algorithm

ARC4 / Spritz | Kleptography

Key generation and encryption | SETUP attack

The attacker's perspective

The goal of an attacker is to use publicly available information, i.e. the public keys and ciphertexts, to obtain the information that he or she needs to factor the key composite N and thus recalculate the private exponent D.

Decrypt encrypted P

Calculate private keys

Decrypt ciphertexts

Back to Key Generation and Encryption

Carrying out a SETUP attac

Step 7: Extract the encrypted prime P from the upper bits of the composite N. Decrypting this value with the attacker's private key yields the prime P, unless a borrow bit was taken in the earlier division. Since the attacker can't know if this occurred, he or she must decrypt both the encrypted P and that value plus one.

Step 8: Using P and P' along with the public key allows the attacker to reproduce the second prime Q and then the private key. The attacker must calculate Q and Q', although only one of them will be prime and divide N evenly.

Step 9: You now have two possible keys to decrypt the ciphertext with. An attacker should already know which is correct, but one will always reproduce the original plaintext message and the other nonsense.

Public key

The public key is by definition publicly accessible and thus visible to an attacker. This key was copied directly from the first tab.

N (composite)	E (public exponent)
b356b561a090a96f	67960daebc661227

Additional data

The encrypted P is read from the upper bits of the public key composite N, and to decrypt it the attacker will need his or her own private key.

Encrypted P	Attacker's D (private exponent)
b356b561	331a3165

Calculations

The attacker will find the prime P encrypted with his or her public key stored in the upper bits of the composite N. Because of a potential carry bit in division in the prime generation algorithm, the attacker must also calculate P' by adding one to the encrypted P before decrypting. The attacker can use P to find Q and then D but must also use P' to find Q' and D'.

Decrypted P	Decrypted P' (P + 1)
8782402b	c9134767
Q = N / decrypted P	Q' = N / decrypted P'
152cd6c5b	e45370b9
D (private exponent)	D' (private exponent)
0	88270e838fcd1ba7

Ciphertext

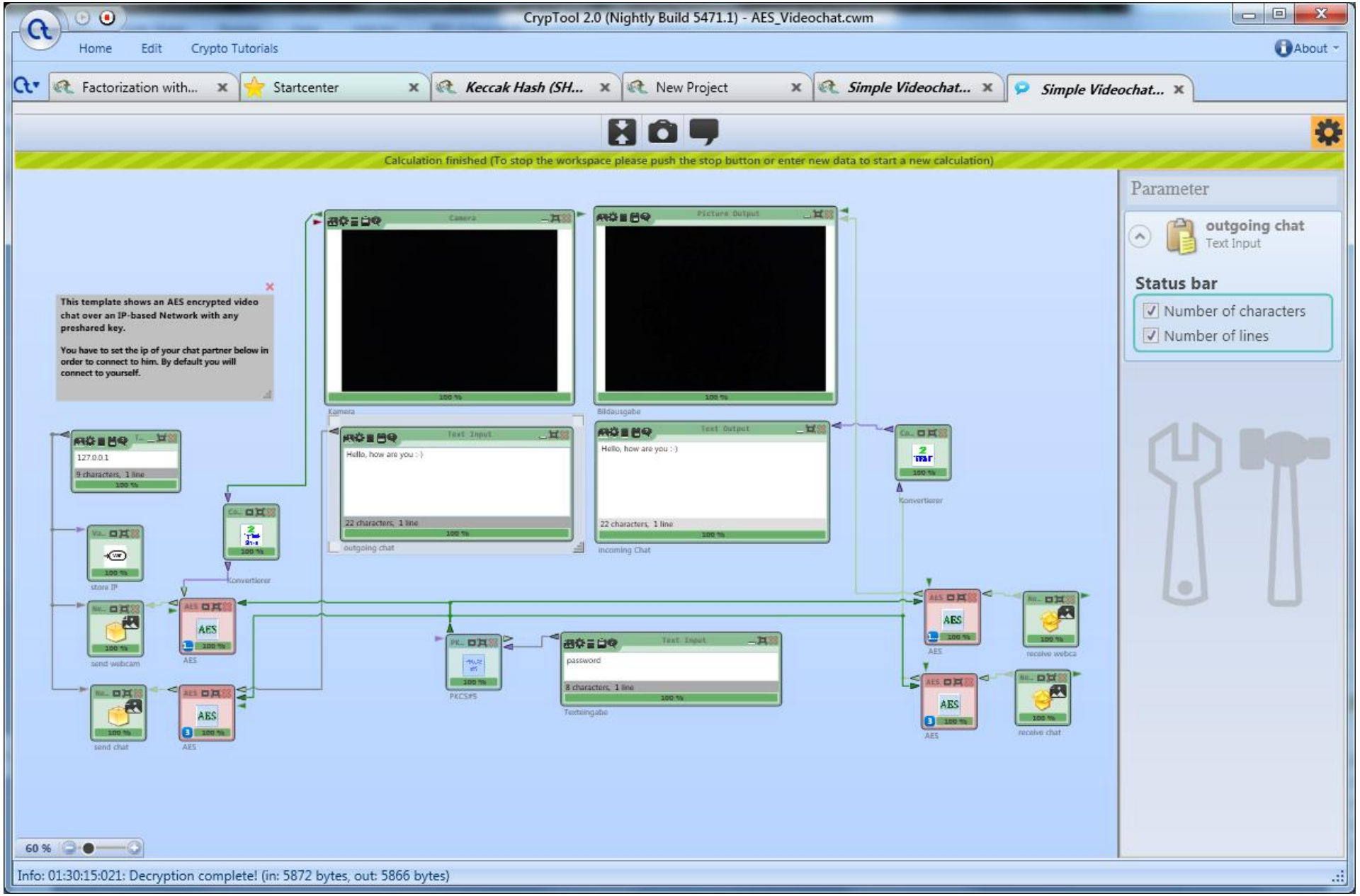
The ciphertext is transmitted publicly, so it is entirely visible to an attacker or any other party monitoring the communication channel.

```
7247fb1344b4742e2dd0c21024b7e3e49cea0fcfed03c844112065b879a
254984e190ad3da1e496069ed5c5e6fb8f965
```

Decrypted texts

The attacker can obtain the plaintext by decrypting with the a recalculated private key, but he or she will not know whether D or D' is the correct private exponent and thus must use both.

Ciphertext decrypted with D	Ciphertext decrypted with D'
	this is a test message


CrypTool 2.0 (Nightly Build 5471.1) - AES_Videochat.cwm
Home
Edit
Crypto Tutorials
About
Factorization with...
Startcenter
Keccak Hash (SH...
New Project
Simple Videochat...
Simple Videochat...
Calculation finished (To stop the workspace please push the stop button or enter new data to start a new calculation)
This template shows an AES encrypted video chat over an IP-based Network with any preshared key.
You have to set the ip of your chat partner below in order to connect to him. By default you will connect to yourself.
Camera
Picture Output
Text Input
Hello, how are you :)
22 characters, 1 line
outgoing chat
Text Output
Hello, how are you :)
incoming Chat
127.0.0.1
9 characters, 1 line
store IP
send webcam
send chat
AES
PKCS#5
password
8 characters, 1 line
Texteingabe
receive webca
receive chat
Parameter
outgoing chat
Text Input
Status bar
Number of characters
Number of lines
60 %
Info: 01:30:15:021: Decryption complete! (in: 5872 bytes, out: 5866 bytes)

CrypTool 2.0 (Nightly Build 5471.1) - Keccak Hash (SHA-3)

Home Edit Crypto Tutorials About

Factorization with... Startcenter Keccak Hash (SH... New Project Simple Videochat...

Calculation finished (To stop the workspace please push the stop button or enter new data to start a new calculation)

Text Input

SHA-3, originally known as Keccak (pronounced [krtʃak], like "ketchak"), is a cryptographic hash function designed by Guido Bertoni, Joan Daemen, Michaël Peeters, and Gilles Van Assche, building upon RadioGatún. On October 2, 2012, Keccak was selected as the winner of the NIST hash function competition. SHA-3 is not meant to replace SHA-2, as no significant attack on SHA-2 has been demonstrated. Because of the successful attacks on MD5, SHA-0 and theoretical attacks on SHA-1, NIST perceived a need for an alternative, dissimilar cryptographic hash, which became SHA-3.

[Source: http://en.wikipedia.org/wiki/SHA-3]

621 characters, 3 lines

100 %

Input

Text Output

CE 1A 15 4F CF A9 9D 4F C8 12 B7 24 F4 B3 3C BA A3 A4 9D 9D 23 13 30 F4 C9 B1 12 EC AA 71 D3 50

95 characters, 1 line

100 %

Hash

Keccak

Squeezing Phase

The 256-bit hash value is extracted from the bit rate part (upper part of the state).

State

Hash Output

100 %

Keccak

Converter

Text Output

#Keccak: running Keccak with the following parameters:
#Keccak: output length 256 bits
#Keccak: state size 1600 bits
#Keccak: bit rate 1088 bits
#Keccak: capacity 512 bits

#Sponge: the input of length 5032 bits is padded to 5440 bits
#Sponge: the padded input is splitted into 5 block(s) of size 1088 bit

#Sponge: begin absorbing phase
#Sponge: XORing input block #1 on state

#Keccak-f: start Keccak-f[1600] with 24 rounds
#Keccak-f: state before permutation:

00: 53 48 41 2D 33 2C 20 6F
01: 72 69 67 69 6E 61 6C 6C
02: 79 20 6B 6E 6F 77 6E 20
03: 61 73 20 4B 65 63 63 61
04: 6B 20 28 70 72 6F 6E 6F
05: 75 6E 63 65 64 20 5B 6B
06: C9 9B 74 CA 83 61 6B 5D
07: 2C 20 6C 69 6B 65 20 E2
08: 80 9C 6B 65 74 63 68 61
09: 6B E2 80 9D 29 2C 20 69
10: 73 20 61 20 63 72 79 70
11: 74 6F 67 72 61 70 68 69
12: 63 20 6B 61 73 68 20 66
13: 75 6E 63 74 69 6F 6E 20
14: 64 65 73 69 67 6E 65 64
15: 20 62 79 20 47 75 69 64

8,743 characters, 311 lines

100 %

Debug Information

69 %

Info: 01:23:56:173: Execute model now!

CrypTool 2.1 (Nightly Build 6843.1) - Padding Oracle Attack on AES

Home | Edit | Crypto Tutorials | About

New | Open | Save | Print | Play | Stop | Log | Startcenter | Connect | Updates | Settings | Help

File | Execute | View | Extras

Startcenter | AES Cipher (Te... | Keccak Cipher | Padding Oracle... | Heartbleed Test | CrypCloudMan... | Simple AES Chat | Simple Video a...

CLIENT
The client wants to send a message to the server. AES in CBC mode is used to encrypt the message. The message consists of two 16 byte long blocks: The first block is only used as initialization vector, while the second block contains the secret information.

Text...
49 4E 49 54 76 45 43 54 EA 33
10 8C 92 07 06 05 02 03 04 05
06 07 08 09 01 02 03 04 05 06
07 08
95 characters, 1 line
0 %
Plaintext

St... 0 %
String Decode

AES 0 %
Encrypt

Text...
1234567812345678
16 characters, 1 line
0 %
Key

St... 0 %
String Decode

Padding Oracle Attack

PHASE 1 | PHASE 2 | PHASE 3

Input
C1 | C2
Response from the Padding Oracle

Attack Logic
D2
C1
O
P2
Currently Viewing Bytes 1..8

<< | < | Next | Auto search | Decrypt completely

Output
Oracle Requests: 0
C1 | C2

0 %
Padding Oracle Attack

SERVER
Upon receipt of an encrypted message, the server decrypts it in CBC mode (C2 is decrypted and then XORed with C1). Afterwards the padding is validated. The result of the validation is then returned as True/False Response to the attacker.

AES 0 %
Decrypt

Padding Oracle
P2
Padding:
Currently Viewing Bytes 1..8
0 %
Padding Oracle

LEGEND
C1: The first block of the client message (encrypted).
C2: The second block of the client message (encrypted).
D2: The decrypted block C2.
O: The Overlay, which represents the manipulation to C1.
P2: The resulting plaintext message. During the attack, P2 represents the padding. At the end of the attack, P2 represents the second block of the original plaintext message.

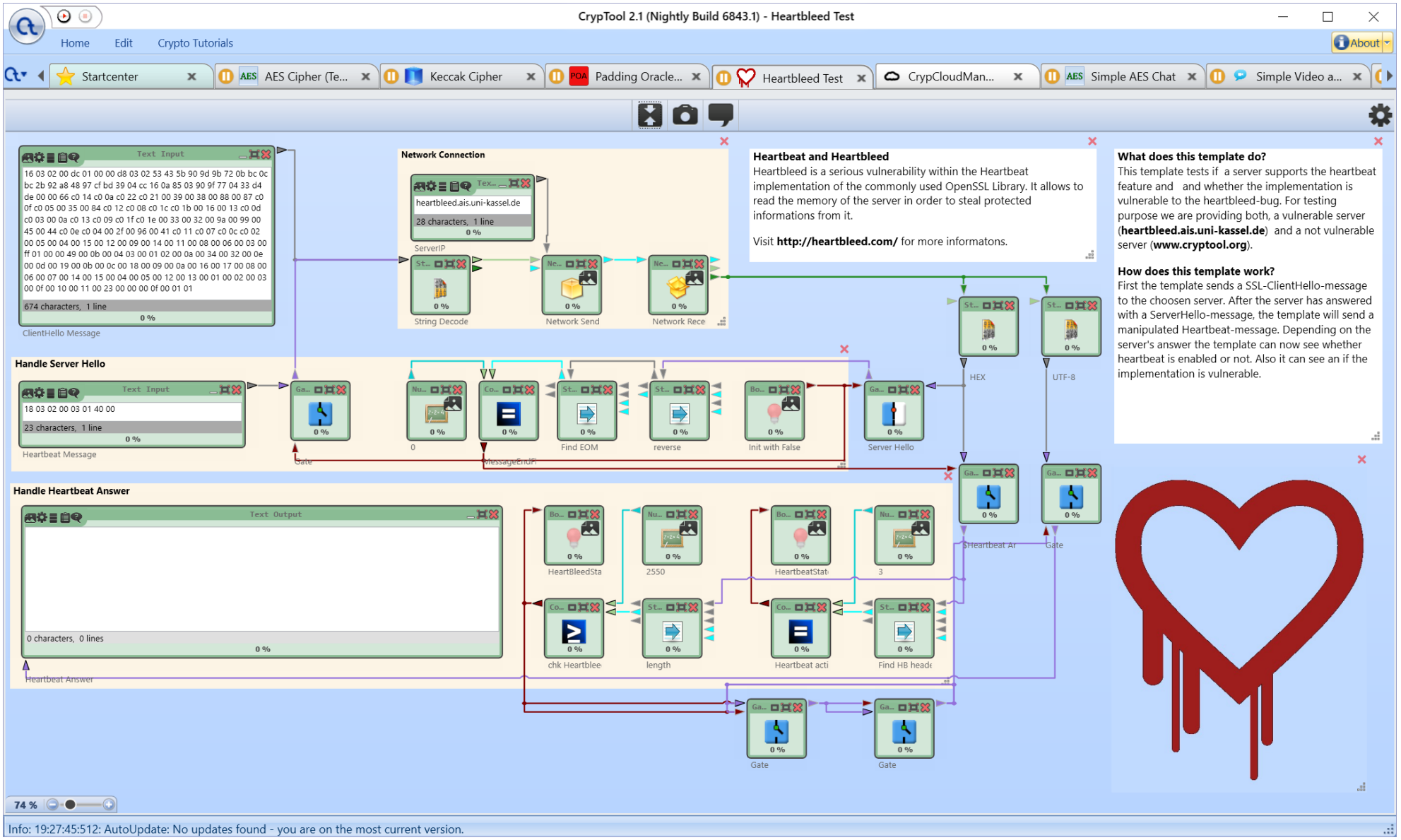
This plugin can display only 8 bytes per block at one time. While having a block length greater than 8, the scroll bar can be used to view all bytes.

ATTACKER
The attacker is between client and server and intercepts every message. His/her goal is to decipher the secret information of the message. The message consists of two ciphertext blocks C1 and C2 and the secret information is in block C2. The decipherment is performed by modifying the first ciphertext block C1, sending messages to the server and interpreting the server responses. The modifications are performed by computing the XOR of the original block C1 and a so called "Overlay" O. The result of this computation is the new, corrupt block C1.

The attack consists of three phases:
1. Find a valid padding.
2. Determine the padding length by finding the first padding byte.
3. Bytewise decryption of the message.

88 %

Info: 18:27:45:511: AutoUpdate: No updates found - you are on the most current version.


CrypTool 2.1 (Nightly Build 6843.1) - Heartbleed Test
Home
Edit
Crypto Tutorials
About
Startcenter
AES Cipher (Te...
Keccak Cipher
Padding Oracle...
Heartbleed Test
CrypCloudMan...
Simple AES Chat
Simple Video a...
Text Input
16 03 02 00 dc 01 00 00 d8 03 02 53 43 5b 90 9d 9b 72 0b bc 0c
674 characters, 1 line
ClientHello Message
Network Connection
heartbleed.ais.uni-kassel.de
28 characters, 1 line
ServerIP
String Decode
Network Send
Network Rece
Heartbeat and Heartbleed
Heartbleed is a serious vulnerability within the Heartbeat implementation of the commonly used OpenSSL Library. It allows to read the memory of the server in order to steal protected informations from it.
Visit http://heartbleed.com/ for more informatons.
What does this template do?
This template tests if a server supports the heartbeat feature and and whether the implementation is vulnerable to the heartbleed-bug. For testing purpose we are providing both, a vulnerable server (heartbleed.ais.uni-kassel.de) and a not vulnerable server (www.cryptool.org).
How does this template work?
First the template sends a SSL-ClientHello-message to the choosen server. After the server has answered with a ServerHello-message, the template will send a manipulated Heartbeat-message. Depending on the server's answer the template can now see whether heartbeat is enabled or not. Also it can see an if the implementation is vulnerable.
Handle Server Hello
Text Input
18 03 02 00 03 01 40 00
23 characters, 1 line
Heartbeat Message
Gate
0
Find EOM
reverse
Init with False
Server Hello
HEX
UTF-8
Handle Heartbeat Answer
Text Output
0 characters, 0 lines
Heartbeat Answer
HeartBleedSta
2550
HeartbeatStat
3
chk Heartblee
length
Heartbeat acti
Find HB heade
Gate
74 %
Info: 19:27:45:512: AutoUpdate: No updates found - you are on the most current version.

CrypTool 2.1 (Nightly Build 6843.1) - CrypCloudManager

Home Edit Crypto Tutorials About

AES Cipher (Te... | Keccak Cipher | Padding Oracle... | Heartbleed Test | CrypCloudMan... | Simple AES Chat | Simple Video a... | Enigma Cipher...

Current Jobs

Logged in as: betest

Logout

Click on one of the open buttons to participate in a job.

Date	Job ID	Job name	Creator	Progress	
1/26/2016 1:22 PM	963E3	testjob@konze	asdasd7	0 / 128	Open
~	C9598	DES 56bit Distributed Known-Plaintext Attack	kopal	/ 17.179.869.1	Open
1/26/2016 5:00 PM	879BA	DES 56bit Distributed Known-Plaintext Attack	kopal	0 / 536.870.912	Open
1/26/2016 5:03 PM	14041	DES 56bit Distributed Known-Plaintext Attack	kopal	.533 / 536.870.	Open
1/27/2016 4:36 AM	C1B03	DES keysearch for plaintext 40000000	Andrew T	0 / 2.048	Open
1/27/2016 4:42 AM	704AD	DES keysearch for plaintext 40000000	Andrew T	0 / 16.777.216	Open
1/27/2016 5:18 AM	6F3E8	DES keysearch for plaintext 40000000	Andrew T	0 / 131.072	Open
1/27/2016 7:41 PM	C54A0	Key search for 40000000	Andrew T	.85 / 67.108.86	Open
~	0A523	Fast Cloud Job	kopal	0 / 524.288	Load
~	E4C03	Big Cloud Test Job	kopal	0 / 65.536	Load
~	514CE	Windows 8 TestPC Big Cloud Test Job	kopal	0 / 65.536	Load
~	68FC6	New Big Cloud Test Job	kopal	0 / 65.536	Load

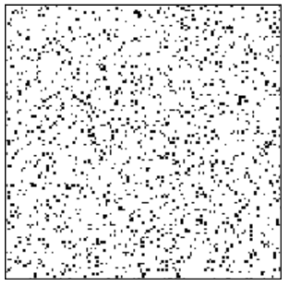
Job ID: 140417552B6A936AE49600E40120B4AE

Job name: DES 56bit Distributed Known-Plaintext Attack

Epoch:

52/32768

Bitmask:



Description:

This cloud job performs an exhaustive keysearching attack on a full DES 56bit keyspace.

Used cost function is RegEx
We know that the plaintext begins with "CrypTool"

Refresh

Add

Info: 19:54:18:372: Ausführungsmaschine erfolgreich gestoppt

CrypTool 2.1 (Nightly Build 6843.1) - SATSolverTextInput.cwm

Home Edit Crypto Tutorials About

Enigma Cipher... | Keccak Cipher | Padding Oracle... | Heartbleed Test | CrypCloudMan... | Simple AES Chat | Simple Video a... | SAT Solver (T...

Text Input

```
c this is the cnf related to the example in the online help
c consisting of 4 variables and 3 clauses
p cnf 4 3
1 2 0
-2 3 0
1 3 -4 0
```

140 characters, 6 lines

0 %

Text Input

SAT Solver

0 %

String Encode

Text Output

```
============================[ Problem Statistics ]=============================
|                                                                             |
| Number of variables:          4                                             |
| Number of clauses:            3                                             |
| Parse time:                 0.00 s                                          |
| Eliminated clauses:         0.00 Mb                                         |
| Simplification time:        0.00 s                                          |
|                                                                             |
============================[ Search Statistics ]==============================
| Conflicts |          ORIGINAL          |          LEARNT          | Progress |
|           |    Vars  Clauses Literals  |    Limit  Clauses Lit/Cl |          |
===============================================================================
===============================================================================
restarts              : 1
conflicts             : 0          (0 /sec)
decisions             : 1          (0.00 % random) (250 / sec)
propagations          : 0          (0 /sec)
conflict literals     : 0          (-1.#J % deleted)
CPU time              : 0.004 s

SATISFIABLE
```

1,363 characters, 21 lines

0 %

Text Output

0 %

String Encode

Text Output

```
SAT1 -2 3 -4 0
```

14 characters, 1 line

0 %

Text Output

77 %

Info: 20:07:25:100: Ausführungsmaschine erfolgreich gestoppt

Parameter

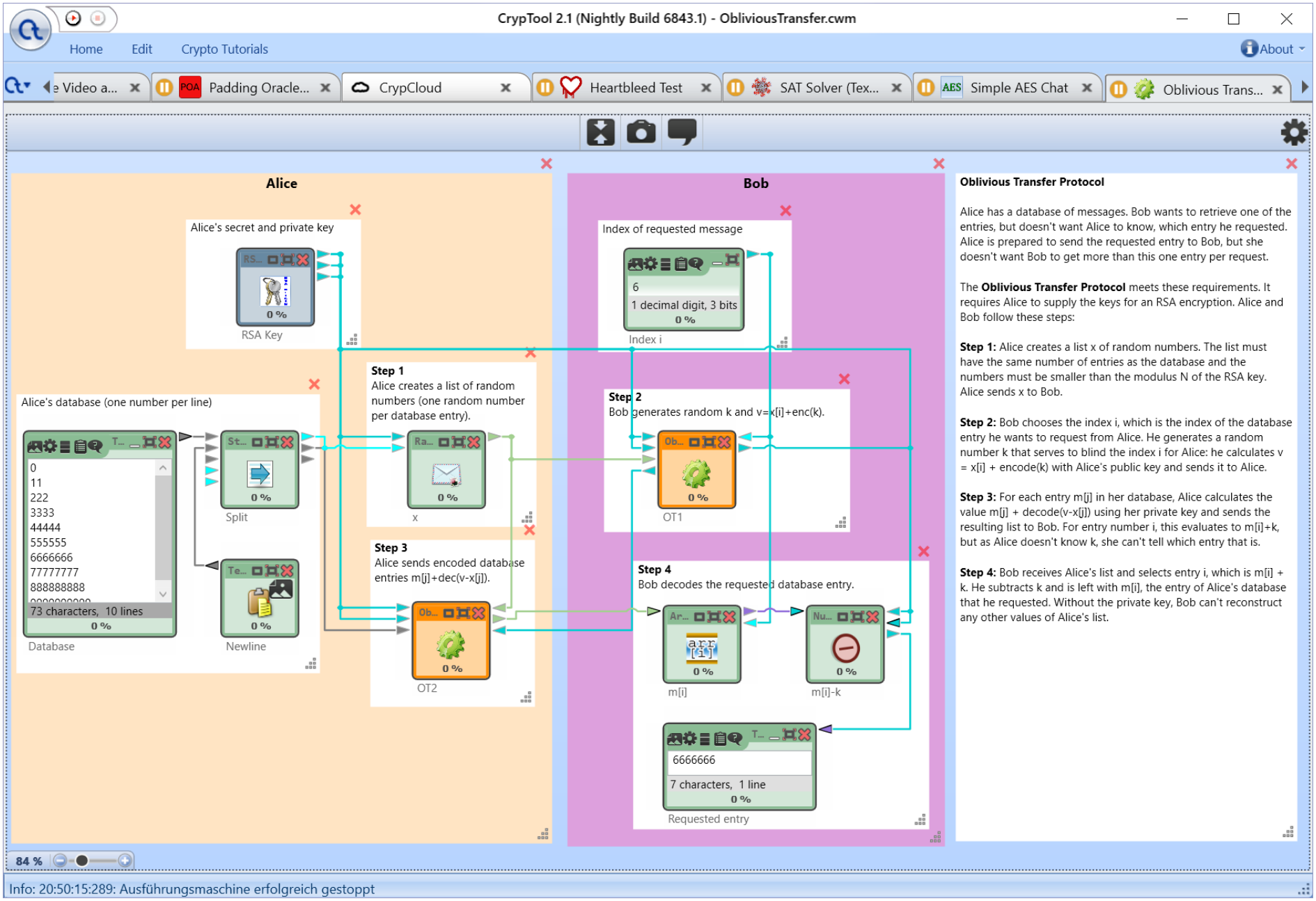
SAT Solver
SAT Solver

Output options

Clear output: Yes
Verbosity: Normal
Dimacs: Off

Solver options

Base restart interval: 100
Restart interval increase factor: 2
Level of phase saving: Full
Conflict clause minimization: Deep
Variable activity decay factor: 0.95
Clause activity decay factor: 0.999
Random frequency: 0
Randomize initial activity: Off
Luby sequence


CrypTool 2.1 (Nightly Build 6843.1) - ObliviousTransfer.cwm
Home
Edit
Crypto Tutorials
About
Video a...
Padding Oracle...
CrypCloud
Heartbleed Test
SAT Solver (Tex...
Simple AES Chat
Oblivious Trans...
Alice
Alice's secret and private key
RSA Key
Alice's database (one number per line)
0
11
222
3333
44444
555555
6666666
77777777
888888888
73 characters, 10 lines
Database
Split
Newline
Step 1
Alice creates a list of random numbers (one random number per database entry).
x
Step 3
Alice sends encoded database entries m[j]+dec(v-x[j]).
OT2
Bob
Index of requested message
6
1 decimal digit, 3 bits
Index i
Step 2
Bob generates random k and v=x[i]+enc(k).
OT1
Step 4
Bob decodes the requested database entry.
m[i]
m[i]-k
6666666
7 characters, 1 line
Requested entry
Oblivious Transfer Protocol
Alice has a database of messages. Bob wants to retrieve one of the entries, but doesn't want Alice to know, which entry he requested. Alice is prepared to send the requested entry to Bob, but she doesn't want Bob to get more than this one entry per request.
The Oblivious Transfer Protocol meets these requirements. It requires Alice to supply the keys for an RSA encryption. Alice and Bob follow these steps:
Step 1: Alice creates a list x of random numbers. The list must have the same number of entries as the database and the numbers must be smaller than the modulus N of the RSA key. Alice sends x to Bob.
Step 2: Bob chooses the index i, which is the index of the database entry he wants to request from Alice. He generates a random number k that serves to blind the index i for Alice: he calculates v = x[i] + encode(k) with Alice's public key and sends it to Alice.
Step 3: For each entry m[j] in her database, Alice calculates the value m[j] + decode(v-x[j]) using her private key and sends the resulting list to Bob. For entry number i, this evaluates to m[i]+k, but as Alice doesn't know k, she can't tell which entry that is.
Step 4: Bob receives Alice's list and selects entry i, which is m[i] + k. He subtracts k and is left with m[i], the entry of Alice's database that he requested. Without the private key, Bob can't reconstruct any other values of Alice's list.
84 %
Info: 20:50:15:289: Ausführungsmaschine erfolgreich gestoppt

Agenda

CT Future and Wishes

- Consistency and completeness
- Development assistance (programming, layout, translation, testing)
  - Mainly for the new projects (preferred):
    - C# project: “CrypTool 2“ = CT2
    - Java project: “JCrypTool“ = JCT
    - Browser project: “CrypTool-Online“ = CTO
  - CT1 will be maintained, but new features will be added only to CT2 and JCT
- Don’t hesitate to contact us if you wish to contribute.
  Some open tasks can also be found in the wiki.
- Users who make a significant contribution are referenced on request by name in the online help, the readme file, the about dialog, and/or on the CrypTool website
- Download numbers per month from the CrypTool website: circa 10,000
  - A bit more than half of these downloads are of the English version
  - CT1 is currently downloaded around 4,000 times
  - CT2 and JCT are downloaded around 2,000 times a month each

CrypTool Needs

- Feedback, criticism, suggestions, and ideas
  (e.g. add privacy stuff, and more modern theory)
- Integration of additional algorithms, protocols, analysis for CT2, JCT, and CTO
- Developers, testers, translators, people who commit to take care for a while
- Administrators for the websites (e.g. Joomla upgrade) and the development environments
- Especially a JS developer for CTO, and a Java developer for JCT
- In particular, university faculties that use CrypTool for educational purposes are invited to contribute to the further development of CrypTool

Wishes to you today:

- Offer your students seminars, projects, and theses to enhance CT2, JCT, and CTO
- Create challenges for MTC3
- Use it yourself in your exercises, your lectures or as research framework
- Spread the word

bernhard.esslinger@uni-siegen.de
bernhard.esslinger@gmail.com

Thanks for your attention!

www.cryptool.org

(also see: https://en.wikipedia.org/wiki/CrypTool)