Cryptology with JCrypTool (JCT)

A Practical Introduction to Cryptography and Cryptanalysis

Prof Bernhard Esslinger and the CrypTool team

Nov 24th, 2020

JCrypTool 1.0

Introduction to the e-learning software JCrypTool

Applications within JCT – a selection

How to participate

2

87

Introduction to the software JCrypTool (JCT)

Overview

JCrypTool – A cryptographic e-learning platform	Page 4
What is cryptology?	Page 5
The Default Perspective of JCT	Page 6
Typical usage of JCT in the Default Perspective	Page 7
The Algorithm Perspective of JCT	Page 9
The Crypto Explorer	Page 10
Algorithms in the Crypto Explorer view	Page 11
The Analysis tools	Page 13
Visuals & Games	Page 14
General operation instructions	Page 15
User settings	Page 20
Command line parameters	Page 21

JCrypTool – A cryptographic e-learning platform The project

Overview

- JCrypTool abbreviated as JCT is a free e-learning software for classical and modern cryptology.
- JCT is platform independent, i.e. it is executable on Windows, MacOS and Linux. It has a modern pure-plugin architecture.
- JCT is developed within the open-source project CrypTool (<u>www.cryptool.org</u>).
- The CrypTool project aims to explain and visualize cryptography and cryptanalysis in an easy and understandable way while still being correct from a scientific point of view.
- The target audience of JCT are mainly:
 - Pupils and students
 - Teachers and lecturers/professors
 - Employees in awareness campaigns
 - People interested in cryptology.
- As JCT is open-source software, everyone is capable of implementing his own plugins. Already developed components can be easily reused.
- JCT was built by more than 100 contributors from different countries.



JCT Splash Screen

The meaning of cryptology

- From Greek: "kryptós" ("hidden, secret") and "lógos" ("writing", however in this context "lógos" means "study").
- Cryptology is about techniques and protocols making information available only for authorized persons. Cryptology consists of two parts (fields).

The field cryptography

- Science of encryption systems guaranteeing secure and confidential storage and exchange of information (e.g. between computers).
- Nowadays, other important tasks are a secure exchange of the encryption keys and integrity checking, e.g. for online banking, for electronic elections, or for electronic money.
- Most of the methods used in this field are based on (unsolved/difficult) mathematical problems.

The field cryptanalysis

- Cryptanalysis is the counter part to cryptography and studies theories and techniques for testing and breaking cryptographic methods.
- It tries e.g. to derive information about the original plaintext or the used encryption key by investigating a ciphertext (the result of an encryption process).
- Therefore, maths and computer science are used (e.g. statistical tests, entropy, analysis of frequency and structure, complexity considerations, brute-force algorithms and much more).

The Default Perspective of JCT

... focuses on documents (document-oriented)

00	JCrypTool	
] 🗟 • 🚑 🔛 🕼 🖆 📮 🕒] 🧿		😭 🔙 Default 🔥 Algorithm
🎩 File Explorer 🕱 🧔 Aktionen 🏻 🚹 📄 🤣 🌄 🗖	🗑 unsaved001.txt 🕱 🗖 🗖	🕉 Crypto Explorer 🕱 🛛 🗸 🖓 🖓
	This is the JCrypTool sample file.	Q type filter text
Applications Beautzerinformationen	You can use this file for a fast start with JCrypTool, e.g. by	Symmetric 🗠
▶ 🗋 bin	encrypting or digitally signing it using the 'Algorithms' menu,	🐕 ElGamal
Cores Cores	'Analysis' menu.	🐕 RSA
▶ 🛅 data	All another another and additionally around in the	🗎 Hash 🛛 🗠
▶ atc	'Crypto Explorer' view on the right side. A double click on the	R MD5
🛅 home	selected entry on the 'Algorithms' tab launches a wizard which	SHA
▶ 🔁 Library	guides you step by step through the encryption process. Decrypting the file at a later time works the same way.	ISI SHA3-Candidates
Network		Buie
▶ 🛅 opt	All offered algorithms as well as all analysis always reauire an opened file in one of the JCrypTool editors.	
▶ 🛅 private	Visualizations and games on the other hand are normally	100 нмасмоз
▶	independent of any opened file. You can either use this sample	Candom Number Generator
System	The original file remains untouched all the time, every	🐕 SHA1
▶ 🛅 tmp	cryptographic operation creates its own working file.	🔁 Signature 🛷
▶ 🛅 Users	The filter field on top of the 'Crypto Explorer' view can help you	🐕 DSA
▶ usr	searching a particular algorithm (as well as an analysis,	Symmetric 🛛
🕼 Help 🕱 🔶 ♥ ╹ 🗖	active tab on matching results.	R AES
陷 Contents 💖 Search 📽 Related Topics 💷 Bookmarks	More information on learning, using and extending [[rvnToo] is	J⊆i Dragon
🛗 Index	available in the extensive online help, which can be accessed	🔞 IDEA
Algorithm	via the menu 'Help'> 'Help Contents'.	JSi LFSR
The Algorithms view is the heart of the FlexiProvider perspective.		10 RC6
by their functionality (like block cipher, signature schemes etc.).		The security the s
An algorithm can be selected via a double click on its name. In the		Si XML Canonicalization (maintain comments)
algorithm, like the block cipher mode.		🔄 XML Canonicalization (remove comments)
Selecting a function (algorithm) is the first step. All following steps		J도 XML Decryption
are then shown in the Operations view window.		JST XML Encryption
More results:		S XML Signature
Search for Algorithms view		S XML Verification
		Algorithms Analysis Visuals Games
] □ [◆] Search for Algorithms view		

Typical usage of JCT in the Default Perspective

... selecting a method from the main menu "Algorithms"

File Edit Algorithms Analysis Visuals Ga	mes Window Help	
🗟 🔻 / 🗐 Asymmetric		😰 🔛 Default 🔀 Algorithm
🗏 File Exp 🛛 Classic 🔹 🕨	🗑 unsaved001.txt 🕱 🛛 🗖 🗖	🚿 Crypto Explorer 🕱 🛛 🖇 🗖 🗖
√ि / Hash ►	This is the JCrypTool sample file.	🔍 type filter text
▶ È bin MAC ►	You can use this file for a fast start with JCrypTool, e.g. by	Asymmetric
Random Number Generator	encrypting or digitally signing it using the 'Algorithms' menu,	🔁 Classic 👳
Gr Signature	'Analysis' menu	4 ADFGVX
Symmetric	🔤 AES	🖾 Autokey-Vigenère
For the Keystore	ARC4/Spritz operations are additionally arranged in the view on the right side. A double click on the	🔄 Bifid
▼ ■ nome	Dragon the 'Algorithms' tab launches a wizard which	🖾 Caesar
	BIDEA step through the encryption process.	🖾 Double Box
▶ ☐ data		🔤 Playfair
	RC6	5 Substitution
	Visualizations and games on the other hand are normally	🔄 Transposition
Downloads	independent of any opened file. You can either use this sample	🔄 Vigenère
to Help ☎ ↔ ♀ ♀ □	file or open any of your own files wherever one is required. The original file remains untouched all the time, every	5 XOR
Contents & Search Related Topics	cryptographic operation creates its own working file.	🔁 Hash
💷 Bookmarks 🗮 Index	The filter field on top of the 'Crypto Explorer' view can help you	MAC
	searching a particular algorithm (as well as an analysis,	Random Number Generator
Crypto Explorer View	visualization or game). Using this field filters the currently	🛅 Signature
cryptographic operations in JCrypTool		Symmetric 👳
ordered by category. A file must be opened	More information on learning, using and extending JCrypTool is	See AES
in one of the JCrypTool editors in order to execute an Algorithm . This is the case for	available in the extensive online help, which can be accessed	🖾 ARC4/Spritz
views in the category Analysis too.		🖼 Dragon
Visualizations and Games generally do not		10EA
require an opened rite.		🖾 LFSR
Use the menu to switch between the tree		10 RC6
and palette presentation. The filter field enables you to filter for the name of an		Algorithms Analysis Visuals Games

Typical usage of JCT in the Default Perspective

... selecting a method from the main menu "Visuals"

File Edit Algorithms Analysis	Visuals Games Window Help					
S - 2 : C - E = 7 0 Q ½ -	Android Unlock Pattern (AUP)		😰 🔛 Default 😘 Algorithm			
😹 File Explorer 🛿 🛛 🚹 🖪	Ant Colony Optimization (ACO)	- 8	🚿 Crypto Explorer 🕱 🔋 🗖 🗍			
▼```)/	ARC4 / Spritz		🔍 type filter text			
▶ 🛅 bin	Certificate Verification	with JCrypTool, e.g. by	🛱 Visuals 👳			
▶ 🛅 boot	🔲 Chinese Remainder Theorem (CRT)	g the 'Algorithms' menu,	🗖 Diffie-Hellman Kêy Exchange (EC)			
🗀 cdrom	🔲 Diffie-Hellman Key Exchange (EC)	red in the	🗖 ElGamal Cryptosystem			
▶ 🖻 dev	🔲 ElGamal Cryptosystem		Elliptic Curve Calculations			
▶ 🖻 etc	Elliptic Curve Calculations	onally arranged in the	🗖 Extended Euclidean / Reciprocal			
▼ in home	Extended Euclidean / Reciprocal Subtraction	launches a wizard which	Subtraction			
▼ in snuc	Extended RSA Cryptosystem	ryption process.	Extended RSA Cryptosystem			
▶ 🖨from_sp3	🗖 Grille	ks the same way.	🗖 Grille			
data	Hash Sensitivity	nalysis always	Hash Sensitivity			
Desktop	Homomorphic Encryption (HE)	rypTool editors.	Homomorphic Encryption (HE)			
Documents	Huffman Coding	and are normally either use this sample	🗖 Huffman Coding			
ि Help ध	Inner States of the Data Encryption Standard (DES)	ever one is required. the time, every	Inner States of the Data Encryption Standard (DES)			
🖻 Contents 🖉 Search 📽 Relat	Kleptography	working file.	Kleptography			
Bookmarks 🗒 Index	McEliece Cryptosystem	Explorer' view can beln you	McEliece Cryptosystem			
	Merkle Signatures (XMSS^MT)	ell as an analysis,	Merkle Signatures (XMSS^MT)			
Crypto Explorer View	Merkle-Hellman Knapsack Cryptosystem	d filters the currently	Merkle-Hellman Knapsack			
The Crypto Explorer view lists all cryptographic operations in ICry	Multipartite Key Exchange (BD II)		Cryptosystem			
ordered by category. A file must l	Multivariate cryptography	extending JCrypTool is	🗖 Multipartite Key Exchange (BD II)			
in one of the JCrypTool editors in	RSA Cryptosystem	which can be accessed	Multivariate cryptography			
views in the category Analysis to	Shamir's Secret Sharing		🗖 RSA Cryptosystem			
Visualizations and Games genera	Shanks Babystep-Giantstep		SPHINCS Signature			
require an opened file.	Signature Demonstration		SPHINCS+ Signature			
Use the menu to switch between I	Signature Verification		SSL/TLS Handshake			
and palette presentation. The filt	Simple Power Analysis / Square and Multiply		Shamir's Secret Sharing			
enables you to filter for the name	SPHINCS Signature		Algorithms Analysis Visuals Games			
	SPHINCS+ Signature	Insert 7:1				

The Algorithm Perspective of JCT

... focuses on functions (function-oriented)

Image: Construction of the second	1.3.1.3.4.1)
Image: Construction of the second state of the second s	1.3.1.3.4.1)
JCrypTool Keystore	1.3.1.3.4.1)
 RC6 (key strength: 128) He original rite remains untourned all the time, every cryptographic operation creates its own working file. He 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. Hor entropy tographic prevation on learning, using and extending JCrypTool is overlable in the extensive on line help, which can be accessed via the menu 'Help'> 'Help Contents'. Hor entropy tographic prevation on learning, using and extending JCrypTool is overlable in the extensive on line help, which can be accessed via the menu 'Help'> 'Help Contents'. Operations & Operations witw, the corresponding entry will apart in the output destination a file. Via drag and drop from the keystore view, akey can be added to a specific operation. It is necessary to add a matching key before executig an algorithm. To execute an algorithm, after all parameters are set correctly, click on the execute icon in the upper right or the for the operations view. More results: Search for Keystore view Search for Keystore view Operation: <not specified=""></not> Correct String and gorithm in the apperiation of the cycle on the cycle on	549.1.1.1) 549.1.1.7) .2) 1.4.1.8301.3.1.3.4.2.1) 1.6.1.4.1.8301.3.1.3.4.2.3) 3.6.1.4.1.8301.3.1.3.4.2.2)
L	n

The Crypto Explorer In the Default Perspective of JCT

Functionality

- On the right side in the Default Perspective of JCT you can find the tab "Crypto Explorer". In this explorer the functions of JCT are shown.
- All functions shown in the explorer can be found in the menus as well.
- In the same manner as the menus, the explorer is clustered into
 - Algorithms
 - Analysis
 - Visuals
 - Games
- Usually algorithms and analyses are applied to the active document in the editor; the calculated output is shown in a new editor window.
- Visuals and games are independent from the document shown in the editor.

🚿 Crypto Explorer 🕱	~
Q type filter text	8)
asymmetric	
🛅 Classic	⇔
JG ADFGVX	
ा Autokey–Vigenère	
05 Bifid	
🕞 Caesar	
🔄 Double Box	
🔄 Playfair	
Substitution	
🛅 Hash	
MAC	
Random Number Generator	
📄 Signature	
🛅 Symmetric	\Leftrightarrow
🔛 AES	
🔄 Dragon	
10EA	
JST LFSR	
强 RC6	
XML Security	
Algorithms nalysis Visuals Ga	mes

Algorithms in the Crypto Explorer view Clustering 1/2

Classic methods

 This category gathers methods, which were used to encrypt messages roughly until World War I. Many of them are breakable by analyzing frequencies. Most of these methods are nowadays insecure.

Symmetric methods

- Modern methods, where sender and receiver need to have the same key.
- A main problem of symmetric methods is: The key must be shared safely between the relevant participants of the communication.

Asymmetric methods

- Modern methods, where each participant has a pair of keys a private and a public one.
- The sender encrypts his message with the public key of the receiver, while only the receiver can decrypt the message with his own private key.



Algorithms in the Crypto Explorer view Clustering 2/2

Hash & MAC

- Hash functions map data of arbitrary length to a hash value. This
 hash value is associated with the data in a preferably unique way and
 has a fixed bounded length which is normally much smaller than the
 length of the referred data (comparable to a fingerprint).
- Hash values are used to check for changes in documents (integrity). A widely used second application is to check passwords. Therefore the hash value (instead of the plain password) is stored in the database.

Signatures

- Signature algorithms are used to sign messages and documents.
- With a signature one can check the integrity of documents the property that a document is unchanged.

Random number generators

 In cryptography random numbers play a major role. Therefore functions for generating (pseudo-random) sequences of numbers are implemented in JCT as well.

🐝 Crypto Explorer 🛚	~ - 0
Q type filter text	8
asymmetric	
🛅 Classic	
🛅 Hash	0
105 MD5	
🐕 SHA	
🔄 SHA3-Candidates	
MAC	
andom Number Generator	0
强 SHA1	
🛅 Signature	⇔
🐕 DSA	
-	
Symmetric	
C XML Security	

The Analysis tools

In the Crypto Explorer

Analysis algorithms

- In this tab of the Crypto Explorer analysis tools are listed. These tools allow the user to analyze a given cipher text, to find possible regularities (patterns) to derive the plain text or the password (key) of the encryption.
- The algorithms are also applied to the document which is currently opened in the editor.
- Different kinds of analyses are possible. E.g. a *transposition analysis*: a ciphertext which was transposed column-wise or row-wise might be rearranged to its original plaintext.
- With an analysis of frequencies the frequencies of characters or pairs of characters can be determined. As characters appear with variant frequency in each natural language, patterns or recurrences can be found and first ideas of the plain text can be deviated.



Algorithm Analysis Visuals Games

Visuals & Games In the Crypto Explorer

Visuals

- Visuals can be found in the tab "Visuals" in the Crypto Explorer or in the menu "Visuals".
- More than 20 visuals of cryptographic problems, circumstances and algorithms shall help the user to understand cryptography in a descriptive and playful way.
- To understand cryptology, a basic knowledge of mathematics and informatics is necessary. Therefore the visuals explain the appropriate knowledge as well.

Games

- In the section "Games", games can be played and strategies developed to solve apparently easy problems.
- Some games (e.g. the number shark) provide extensive theories and possible strategies.





... 1/5 (Quick Access: Search over all parts in JCT)

Tips and tricks

- With the key combination "Ctrl+3", the quick access window can be opened.
- Algorithms, visualizations and other content from JCrypTool can be found here and directly opened.
- This is the quickest way to search something in the Default Perspective or in the online help, if you don't know where to look for in the menus or in the Crypto Explorer.
- Clicking at a shown entry makes JCT to jump straight there.
- Remark: Elements from the Algorithm Perspective are currently not shown in the quick access window.

Views	Entropy Analysis (Analysis)
	Frequency Ana lysis (Analysis)
	Friedman Test (Ana lysis)
	🔟 Grille Ana lysis (Analysis)
	Simple Power Analysis / Square and Multiply (Visuals)
	Substitution Analysis (Analysis)
	Transposition Analysis (Analysis)
	Vigenère Breaker (Analysis)
	Viterbi Analysis (Analysis)
Commands	Entropy Analysis
	Frequency Analysis
	Grille Analysis
	Show In (Entropy Analysis)
	Show In (Frequency Analysis)
	Show In (Grille Analysis)
	Show In (Simple Power Analysis / Square and Multiply)
	Show In (Substitution Analysis)
	Show In (Transposition Analysis)
	Show In (Viterbi Analysis)
	Show View (Entropy Analysis) - Shows a particular view
	Show View (Frequency Analysis) - Shows a particular view
	Show View (Grille Analysis) - Shows a particular view
	Show View (Simple Power Analysis / Square and Multiply) - Sh
	Show View (Substitution Analysis) - Shows a particular view
	 Show View (Transposition Analysis) - Shows a particular view
Help	察 Search ' ana ' in Help

Quick Access window

... 2/5 (open the exhaustive online help within a new browser tab)

Tips and tricks

 With the question mark symbol 3 in the toolbar, an exhaustive online help can be opened in a new browser tab.





JCrypTool 1.0

... 3/5 (open the docked context help)

Tips and tricks

The function key F1 can be used (under Linux and Windows) to open the context-sensitive help at any time.

The context help contains detailed information and references for the current view.

 Alternatively, you can use the big blue question mark symbol to show (or hide) the contexthelp window. This big question mark can be found in all description headers of visual plugins.

😰 Help 🕱 🔅 🗘 🖓 🖓 🖓 🖓	🗖 Verifiable Secret Sharing 🛱								5-0
陷 Contents 💖 Search 📽 Related Topics	Verifiable Secret Sharing Reconstruction G	Chart							
Bookmarks IIII Index Verifiable Secret Sharing Verifiable Secret Sharing is a secret sharing algorithm for sharing a secret. The secret is distributed in a way that every person receives a unique part of the secret. For the reconstruction of the secret are just a few parts	Verifiable Secret Sharing (VSS) Verifiable Secret Sharing is a variation of th reconstruct the secret, while less than t pla improvement to the simple secret sharing s When the mouse focus is set on a container	ne simple secret sharing sch nyers cannot. However, in th scheme by adding a functio r, a corresponding detailed	eme. As with the simp e simple scheme one p n 'Check' by which eacl explanation is display	e scheme layer can player ca ed in the b	e, the secret is split into n shares covertly sabotage a reconstructi an verify the correctness of all sh pox below.	and distributed to n pla on by modifying his sha ares ('Verifiability').	ayers, so that any number t of I are. Verifiable secret sharing ir	the players atroduces a	
necessary. The secret is reconstructed with the	Parameters		Coefficients		Commitments	Shares			Reconstruction-
Sharing it is possible to verify if all the participants provide a correct part of the secret.	Number of players n 2	- +	ao = s 0	- +	Coefficient Commitment Ye	Player i	Share Ni [mod q]		Player 1 🔲 Player 2 💭
See also: Verifiable Secret Sharing 	Number of players t for reconstruction 2	- +	a1 1	- +	a1	Player 1		Check	
More results: Ø Search for Verifiable Secret Sharing view	Secret s							CHECK	

Context help for the visuals plugin "Verifiable Secret Sharing"

In dialog windows: Clicking F1 or pressing the small question mark on the left shows a contexthelp window docked at the right side of the dialog.

?	<u>F</u> inish	Cancel

... 4/5 (minimize, maximize and resize)

Tips and tricks

- The size of each area in JCT can be adjusted via the buttons on the upper right corner of the area.
 - Maximize current area
 - Minimize area 🗖
- Once minimized, an area is represented as a small bar at the left or right border. The tabs contained are displayed as small icons inside the bar.
 - An area can be reset to its old size by clicking on
 - The other icons represent the tabs inside an area. Clicking on one of these, the appropriate tab will be shown as an overlapping window which will be automatically hidden again.
- Reset of a view or of the perspective:
 - Menu "Window" \ "Reset Perspective"
 - Double-click on the plugin's tab changes the view between full and reduced place for the plugin (typical behavior in Eclipse)

Eile <u>E</u> dit Algorithr	ms Analysis							
9 - 6 8 6 6 1	🗉 🥹 👔 🕐						😰 🔛 Default 🐕 Al	gorithm
Certificate 🖾	🖻 Verifiable	🖻 Winternitz 📄	Huffman 📄 Hom	nomo 🖹 ElGamal 📄	Multipartit 🖻 Merkle	e Hel 🖪	Number S 🗖 Android 📄 Hash Sen	
Cantificate	Manifianti							~
In this plug-in y	vou can load t	on hree certificates ar	d adiust their valic	dity periods with six sli	ders. The signature an	nd verif	fication time can be adjusted with two	
additional slide	rs. Three valid	ity models are ava	ilable to validate th	his certificates and the	validity periods.			1
	1	Not Valid Befor	e	Not Valid	After		Load Root CA certificate	
	2004		2034 200	04	2034		Load CA certificate	
Root CA							Load User certificate	
CA						Lo	g;	
		••••••••••••••••••••••••••••••••••••••				#	## 000 ###	~
User						Ri th	oot CA: valid from: 01.07.2010, valid rru: 01.03.2032	
Signature date		11		11		C	A: valid from: 01.01.2013, valid thru:	
Verification date) 					U	ser: valid from: 01.07.2015, valid thru:	
	2004				2034	0	1.11.2023 ignature date: 01.11.2018	
Signature date						V	erification date: 01.07.2020	
Verification date						D	ates based on selection SUCCESSFULLY alidated with Shell model	
Details to the c	ertificates (adju	ust month and yea	r with the sliders a	bove; day can be set h	nere)			
	Root CA	CA	User	Signature date	Verification date			
valid from:	1 /Jul/10	1 /Jan/13	1 /Jul/15	1 /Nov/18	1 /Jul/20			
valid thru:	1 /Mar/32	1 /Sep/29	1 /Nov/23					4
JCrypTool 1.0.0	20190831 (w	() : ekly)	5hell model 🔾 Mo	odified Shell model) Chain model		Validate	×
JCrypTool 1.0.0 jle Edit Algorithr	20190831 (w ms Analysis	• : eekly) Visuals Games ۱	5hell model () Mo	odified Shell model 🔾) Chain model		Validate (×
	20190831 (w ms Analysis I I I I I I I I I I I I I I I I I I I	eekly) Visuals Games <u>}</u>	Shell model () Mo	odified Shell model 🤇) Chain model		Validate) (lgorithr
JCrypTool 1.0.0 ile Edit Algorithr · · · · · · · · · · · · · · · · · · ·	20190831 (w ms Analysis) @ 1 ⑦ n (Read-Only) rypTool sa	eekly) Visuals Games) ای unsaved001. ۳ple file.	Shell model () Mo	odified Shell model 🤇) Chain model		Validate (- - - - - - - - - - - - -	k k lgorithr
JCrypTool 1.0.0 ile Edit Algorithr · · · · · · · · · · · · · · · · · · ·	20190831 (w ms Analysis	● zekly) Visuals Games \ ■ unsaved001. mple file. r a fast star	Shell model O Mo Mindow Help txt کا t with JCrvpT	odified Shell model () Chain model	····	Validate (- - - - - - - - - - - - -	× Igorith
JCrypTool 1.0.0 ile Edit Algorithm ••••••••••••••••••••••••••••••••••••	20190831 (w ms Analysis (Read-Only) rypTool sa is file fo digitally	•) sekly) Visuals Games) Punsaved001. mple file. r a fast star signing it us oralumit us	Shell model O Mo	odified Shell model ○ ool, e.g. by rithms' menu,) Chain model		Crypto Explorer 22 Crypto Explorer 23 Default 24 A Symmetric Classic	v × Igorith
JCrypTool 1.0.0 ile Edit Algorithn e e in a in a line ile keystream002.bir his is the JC ou can use th ncrypting or by applying Analysis' men	20190831 (ww ms Analysis) @ ? ? (Read-Only) rypTool sa is file fo digitally one of th u.	ekky) Visuals Games) @ unsaved001. mple file. r a fast star signing it us e analysis of	Shell model O Mo Mindow Help Mindow Help twith JCrypT ing the 'Algo fered in the	odified Shell model ○) Chain model		Crypto Explorer 22 Crypto Explorer 22 Crypto Explorer 23 Classic 24 Aprice Vizione	v × gorith
JCrypTool 1.0.0 ile Edit Algorithm Edit Algorithm Resystream002.bir his is the JC oou can use th ncrypting or by applying Analysis' men 11 cryptograp	20190831 (wm ms Analysis) @ ? 7 n (Read-Only) rypTool sa is file fo digitally one of th u. hic operat	ekky) Visuals Games) Sunsaved001. mple file. r a fast star signing it us e analysis of ions are addi	Mindow Help	<pre>iool, e.g. by rithms' menu, nged in the</pre>) Chain model		Crypto Explore 22 Crypto Explore 22 Generative Classic Asymmetric Classic 24 ADFCVX 26 Autokey-Vigenère 26 Bidd	y X
JCrypTool 1.0.0 ile Edit Algorithn ile keystream002.bin his is the JC ou can use th ncrypting or r by applying Analysis' men 11 cryptograp Crypto Explor	20190831 (w ms Analysis)	eekly) Visuals Games \ Wisuals Comes \ Wisu	Mindow Help	cool, e.g. by rithms' menu, nged in the click on the wizard which) Chain model		Crypto Explore 20 Crypto Explore 20 Generative Classic Active-Vigenère 20 Bidd Cracer	v ×
JCrypTool 1.0.0 lie Edit Algorithr le keystream002.bin his is the JC ou can use th ncrypting or by applying Analysis' men ll cryptograp Crypto Explor Elected entry uides you ste erryuting the	20190831 (w ms Analysis a (Read-Only) rypTool sa is file fo digitally one of th u. hic operat er' view o on the 'A p by step file at -	eekly) Visuals Games \ Winsaved001. mple file. r a fast star signing it us e analysis of ions are addi n the right s loorithms' ta through the e later + ime	Mindow Help	cool, e.g. by rithms' menu, nged in the click on the wizard which ccess.	=		Validate (×
JCrypTool 1.0.0 lie Edit Algorithr keystream002.bin his is the JC ou can use th ncrypting or by applying Analysis' men ll cryptograp Crypto Explor elected entry uides you ste ecrypting the	20190831 (w ms Analysis a to only n(Read-Only) rypTool sa is file fo digitally one of th u. hic operat er' view o on the 'A p by step file at a casiti-	eekly) Visuals Games \ Wunsaved001. mple file. r a fast star signing it us e analysis of ions are addi n the right s liporithms' ta through the e later time w	Mindow Help	cool, e.g. by rithms' menu, nged in the click on the wizard which ccess. ₩ay.	=		Validate (Validate) Validate) Valida	× k lgorithi
JCrypTool 1.0.0 lie Edit Algorithr keystream002.bir his is the JC ou can use th ncrypting or by applying Analysis' men ll cryptograp Crypto Explor elected entry uides you ste ecrypting the electred and entry uides you ste ecrypting the	20190831 (wm ms Analysis Part of the second second in (Read-Only) rypTool sas is file fo digitally one of th u. hic operat file at a gorithms a ned file i	<pre>eekly) Visuals Games } @ unsaved001. mple file. r a fast star signing it us e analysis of ions are addi n the right s liporithms' ta through the e later time w s well as all n one of the</pre>	Mindow Help Mindow Help twith JCrypT ing the 'Algo fered in the tionally arra ide. A double to launches a ncryption pro orks the same analysis alw JCrypTool edi	oodified Shell model oool, e.g. by rithms' menu, nged in the click on the wizard which cess. way. way. ays tors.) Chain model		Validate (Validate) Validate (Validate) Validate) Valida	× lgoritht
JCrypTool 1.0.0 JCrypTool 1.0.0 Je Edit Algorith Comparison of the second Analysis' men ll cryptograp Crypto Explor to Explor Analysis' men ll cryptograp Crypto Explor ecrypting the ecrypting the ecrypting the ecrypting the leted entry uides you ste ecrypting the ll offered al equire an ope isualizations	20190831 (wm ms Analysis	eekky) Visuals Games } Wunsaved001. mple file. r a fast star signing it us e analysis of ions are addi n the right s later time w later time w s well as all n one of the on the other. You could be a start of file. You could be a start on the other the other the other the other of file. You could be a start one of the other the othe	Mindow Help Mindow Help twith JCrypT ing the 'Algo fered in the tionally arra ide. A double to launches a ncryption pro orks the same analysis alw JCrypTool edi hand are nor an either use	<pre>oolified Shell model ○ ool, e.g. by rithms' menu, nged in the elick on the vizard which cess. way. ays tors. mally this sample</pre>) Chain model		Validate (Validate) Validate (Validate) Validate) Valida	× k k k k k k k k k k k k k k k k k k k
❑ JCrypTool 1.0.0 JCrypTool 1.0.0 Jle Edit Algorithm A egale and a set of the set	20190831 (wm rs Analysis Page 1 a 1 a 1 (Read-Only) rypTool sa is file fo digitally one of th u. hic operat er' view o on the 'A p by step file at a gorithms a ned file i and games any opene hy for your ile remain	eekly) Visuals Games } Winsaved001. mple file. r a fast star signing it us e analysis of ions are addi n the right s lacer time w s well as all n one of the on the other of file. You c own files wh s untouchad a	Mindow Help Mindow Help twith JCrypT ing the 'Algo fered in the tionally arra ide. A double tionally arra ide. A double analysis alw analysis alw analysis alw analysis alw an either use erever one is lithe + ime	<pre>ool, e.g. by rithms' menu, nged in the elick on the wizard which cess. way. ays ays this sample required. everv</pre>) Chain model		Validate (Validate Valid	× Ilgorithm
CrypTool 1.0.0 GrypTool 1.0.0 Grading Edit Algorithm a v d a v o a v b v o a v o a v b v o a v o v crypto r by applying Analysis' men 11 cryptograp Analysis' men 11 cryptograp crypto Explor crypto Explor elected entry uides you ste tecrypting the b cryptographic independent o ropen a he original f ryptographic	20190831 (wm ns Analysis (Read-Only) rypTool sa is file fo digitally one of th u. hic operat er' view o on the 'A p by step file at a gorithms a ned file i and games any opene ned file i eremain of your ile remain operation	eekly) Visuals Games } Winsaved001. mple file. r a fast star signing it us e analysis of ions are addi n the right s later time w later time w s well as all on one of the on the other of file. You c own files wh s untouched a creates its o	Mindow Help Mindow Help twith JCrypT ing the 'Algo fered in the tionally arra ide. A double b launches a ncryption pro orks the same analysis alw analysis alw analysis alw analysis alw analysis alw ne ther use erever one is li the time, wn working fi	oodified Shell model ○ ool, e.g. by rithms' menu, nged in the click on the wizard which cess. way. way. ays support way. mally this sample required. every le.) Chain model		Validate (Validate Valid	k algorithm
■ JCrypTool 1.0.0 like Edit Algorithm keystream002.bir his is the JC ou can use th norrypting or r by applying Analysis' men 11 cryptograp crypto Explor elected entry crypto Explor ecrypting the litoffered al equire an ope isualizations a he original f ryptographic	20190831 (wm ns Analysis (Read-Only) rypTool sa is file fo digitally one of th u. hic operat er' view o on the 'A p by step file at a gorithms a ned file i and games any opene ned file i eremain of your ile remain operation	<pre>eekly) Visuals Games } @ unsaved001. mple file. r a fast star signing it us e analysis of ions are addi n the right s later time w s well as all n one of the on the other of file. You c own files wh s untouched a creates its o</pre>	Mindow Help Mindow Help twith JCrypT ing the 'Algo fered in the tionally arra ide. A double tionally arra ide. A double tionally arra ide. A double b launches a ncryption pro orks the same analysis alw hand are nor an either use erever one is li the time, wn working fi	ool, e.g. by rithms' menu, nged in the click on the wizard which cess. way. ays tos. mally this sample required. every le.) Chain model		Validate (Validate V	v algorithm
■ JCrypTool 1.0.0 like Edit Algorithm a keystream002.bir his is the JC ou can use th norxypting or r by applying Analysis' men 11 cryptograp analysis' men 11 cryptograp elected entry cryptographic tile or open a he original f ile original	20190831 (wm ns Analysis Analysis Point (Read-Only) rypTool sa is file fo digitally one of thu u. hic operat r' view o on the 'A po by step file at a gorithms a and games any opene ny of your ile remain operation ' ' ' ' ' ' '''''''''''''''''''''	<pre>eekly) Visuals Games } @ unsaved001. mple file. r a fast star signing it us e analysis of ions are addi n the right s later time w s well as all n one of the on the other on the other of file. You c own files wh s untouched a creates its o</pre>	Mindow Help Mindow Help twith JCrypT ing the 'Algo fered in the tionally arra ide. A double tionally arra ide. A double to launches a ncryption pro orks the same analysis alw hand are nor an either use erever one is wn working fi 	ool, e.g. by ool, e.g. by rithms' menu, nged in the relick on the wizard which cess. way. ays. mally this sample required. every le. interference. the sample required. every le. interference. every le. omor) EEGamal C (⊂) Chain model		Validate Validate Va	×
■ JCrypTool 1.0.0 The Edit Algorithm A set a	20190831 (wm ms Analysis	<pre>eekly) Visuals Games } @ unsaved001. mple file. r a fast star signing it us e analysis of ions are addi n the right s later time w later time w s well as all n one of the on the other of file. You c own files wh file. You c own files wh s untouched a creates its o Winternitz</pre>	Mindow Help twith JCrypT ing the 'Algo fered in the tionally arra ide. A double to launches a ncryption pro orks the same analysis alw analysis alw analysis alw ihand are nor an either use erever one is li the time, wn working fi for the same analysis alw the model of the same analysis alw the same the same th	ool, e.g. by rithms' menu, nged in the click on the wizard which cess. way. ays tots. mally this sample required. every le.) Chain model		Validate Validate Va	× lgoritht v =
JCrypTool 1.0.0 lie Edit Algorithn → → → → → → → → → → → → → → → → → → →	20190831 (w ns Analysis a 1 a 1 o (Read-Only) rypTool sa is file fo digitally one of th u. hic operat er' view o on the 'A p by step file at a and games and game	<pre>eekly) Visuals Games 1 @ unsaved001. mple file. r a fast star signing it us e analysis of ions are addi n the right s later time w s well as all n one of the on the other on the other othe</pre>	Mindow Help Mindow Help twith JCrypT ing the 'Algo fered in the tionally arra ide. A double tionally arra ide. A double to launches a ncryption pro- orks the same analysis alw JCrypTool edi JCrypTool edi the time, wn working fi wn working fi Mindow Help the time, mn working fi Mindow Help the time, mn working fi Mindow Help the time, mn working fi Homw Mindow Help the time, the time, mn working fi Homw Mindow Help the time, mn working fi Homw	<pre>ool, e.g. by rithms' menu, nged in the click on the wizard which ccess. way. ays tors. mally this sample required. every le. i</pre>	Multipartit C Merkl	Alle Hell	Validate (Validate V	V Igorith V V V V V V V V V V V V V V V V V V V
■ JCrypTool 1.0.0 lie Edit Algorithh a ← a ← a ← a ← a ← a ← a ← a ← a ← a ←	20190831 (w ns Analysis 20190831 (w ns Analysis 2019 (Read-Only) rypTool sa is file fo digitally one of th u. hic operat er' view of p by step file at a and games and games	<pre>eekby) Visuals Games) @ unsaved001. mple file. r a fast star signing it us e analysis of ions are addi ions are addi ions are addi ions are addi ions are addi in the right s lgorithms' ta through the e swell as all n one of the other of the s well as all n one of the other of the s untouched a creates its o Wintermitz</pre>	Mindow Help Mindow Help twith JCrypT ing the 'Algo fered in the tionally arra ide. A double tionally arra ide. A double b launches a ncryption pro- orks the same analysis alw JCrypTool edi bad are nor an either use erever one is li the time, wn working fi Unthe time, wn working fi Momention Momen	ool, e.g. by rithms' menu, nged in the click on the wizard which cess. way. ays tors. mally this sample required. every le. i periods with six slided and the validity perio	> Chain model	Alae Hell	Validate (Validate V	V Igorithh V I Ional
Certificate ≥ Certificate	20190831 (w ms Analysis a 1 a 1 o (Read-Only) rypTool sa is file fo digitally one of th u. hic operat er' view o on the 'A gorithms a and games and gam	<pre>eekly) Visuals Games 1 @ unsaved001. mple file. r a fast star signing it us e analysis of ions are addi ions are addi ions are addi ions are addi ions are addi ions are addi ions are addi later time w s well as all n one of the on the other of the Vouce own files wh s untouched a creates its o if the Vouce Winternitz</pre>	Mindow Help Mindow Help twith JCrypT ing the 'Algo fered in the tionally arra ide. A double tionally arra ide. A double tionally arra ide. A double orks the same analysis alw JCrypTool edi orks the same analysis alw JCrypTool edi erever one is ll the time, wn working fi 	ool, e.g. by rithms' menu, nged in the click on the wizard which cess. way. ays tors. mally this sample required. every le. in the click and the stand required. every le. in the required. every le. Mot Valid A	Chain model	Verifica	Validate (Validate V	A second seco
Certificate № Certificate №	20190831 (w ms Analysis a 1 a 1 c (Read-Only) rypTool sa is file fo digitally one of th u. hic operat en' view of p by step file at a ang games ang gam	<pre>eekly) Visuals Games 1 @ unsaved001. mple file. r a fast star signing it us e analysis of ions are addi ions are addi available to addi winternitz entry e certificates and a available to valid t Valid Before</pre>	Mindow Help Mindow Help twith JCrypT ing the 'Algo fered in the tionally arra ide. A double tionally arra ide. A double tionally arra ide. A double analysis alw JCrypTool edi orks the same analysis alw JCrypTool edi m working fi I the time, wn working fi I the time, wn working fi Homw dust their validity ate this certificates 2034 2004	ool, e.g. by rithms' menu, nged in the click on the wizard which ccess. way. ays tors. mally this sample required. every le. i	Multipartit C Merkl	Verifica	Validate (Validate V	V X Igorithr V V V
Certificate № Certificate V Certificate V Cont CA	20190831 (wms Analysis Analysis Part of the second secon	<pre>eekly) Visuals Games 1 @ unsaved001. mple file. r a fast star signing it us e analysis of ions are addi n the right s later time w s well as all n one of the on the other on the other ot</pre>	Mindow Help Mindow Help tt with JCrypT ing the 'Algo fered in the tionally arra ide. A double tionally arra ide. A double analysis alw JCrypTool edi orks the same analysis alw JCrypTool edi analysis alw JCrypTool edi analysis alw JCrypTool edi the time, wn working fi Content of the same analysis alw Algo the same algo the	ool, e.g. by rithms' menu, nged in the click on the wizard which ccess. way. aays tors. mally this sample required. every le.	Multipartit C Merka s. The signature and ds.	Alle Hell	Validate Validate	× Igorithm

ICrypTool 1.0.0 20190831 (weekly)

JCrypTool 1.0

... 5/5 (restart within an opened visual plugin; open an editor)

Tips and tricks

JCrypTool 1.0

 Each visual can be reset to its initial settings by clicking on the button "Restart". The button is located in the upper toolbar of the plugin window.

- Quick creation of an editor window
 - On the far left of the toolbar is the editor icon.
 If you click on it, a text editor with the sample file is opened.
 - Directly to the right of it is the arrow.
 With the arrow you can choose which type of editor (and whether empty or filled) should be opened in the middle of the Default Perspective or the middle of the Algorithm Perspective.







User settings ...the global preferences of JCT

More settings

 The global settings of JCT can be found in the preferences: See the menu paths: on Windows and Linux: "Window \ Preferences"; and under MacOS: "JCrypTool \ Preferences".

The most important JCT specific settings are:

Concerning cryptography

- Alphabets: Manage alphabets which are used for many of the classic encryption methods.
- Keystore: Here you can manage the files in which the keys of the JCT keystore are saved.

A newly generated keystore can then be used in the perspective "Algorithm".

	Preferences			• •
type filter text	Alphabets			
Algorithms Cryptography	Available alphabets			
 Alphabets Crypto Provider Editors General Help Install/Update JCT General 	Name Printable ASCII Upper and lower Latin (A-Z,a-; Upper Latin (A-Z) Lower Latin (a-z) Playfair/alike alphabet (25cha ADFGVX Alphabet XOR Alphabet with 32 charact XOR Alphabet with 64 charact	Built-in Yes Yes Yes Yes Yes Yes Yes	Default Yes	Add Edit Remove Set default
	Options Characters. Defines the filtering of non-alphene This option defines the default characters that are not part of algorithms. If non-alphabet characters are cipher unencrypted.	abet charac handling fo the plainte: not filtered Resto	ters In the encrypti In the encrypti In the encryption In the encr	on of r classic taken into the Apply
? <u>``</u>		Car	ncel	oply and Close

Command line parameters

...setting the language and the data directory

Default

- If JCrypTool is started without parameters its language will be inferred from the operating system, if the operating system is in English or German. Otherwise, JCrypTool defaults to English.
- Settings of the last JCrypTool session as well as JCT-specific files are stored in a directory named 'workstation'. This directory is created when JCrypTool is started for the first time, below the JCrypTool directory.

Parameters to control the JCT application from the command line when starting

- Language
 -nl [de, en]
- Data directory (with -data you can change the directory 'workstation') -data [directory]
- Sample: JCrypTool -nl de -clean -data USERDIRECTORY/jct-de JCrypTool -nl en -clean -data USERDIRECTORY/jct-en

With -nl you control, whether the German or English JCT appears.

With -data you control, where JCT stores its data. It's up to you to specify a meaningful value for USERDIRECTORY (under Windows for instance %LOCALAPPDATA%).

Introduction to the e-learning software JCrypTool

Applications within JCT – a selection

How to participate

87

22

Applications within JCT – Overview

The ant colony optimization (ACO)	Page 25
Viterbi analysis	Page 30
Verifiable Secret Sharing	Page 35
Signature demonstration	Page 40
Extended RSA cryptosystem	Page 45
SETUP attack on the RSA key generation (Kleptography)	Page 50
Zero-knowledge protocol: Fiat Shamir	Page 55
Android Unlock Pattern (AUP)	Page 60
Cascades in the Actions window	Page 64
Variable alphabets for classic algorithms	Page 70
JCrypTool console for classic methods	Page 74
The perspective "Algorithm"	Page 79

The ant colony optimization (ACO) The idea

Abstract

The implementation of the ant colony optimization in JCT is a visualization which allows the user to decrypt a cipher text which was encrypted by a transposition cipher.

Functionality

- The ant colony algorithm is an efficient algorithm for solving combinatorial problems.
- E.g. it can be used to find the shortest path from A to B in a graph.
- The algorithm appreciates the way of ants quickly finding their path to a desired location.
- In the algorithm an ant chooses its path based on local information (e.g. information stored in the edges of the graph) and depending on decisions of preceding ants.
- The more ants choose a certain way, the more ants follow.
 This behavior is called swarm intelligence.
- In principle, this algorithm is based on statistical evaluations.



The ant colony optimization

The implementation in JCT

In the menu

"Visuals" \ "Ant Colony Optimization"

The algorithm in application

- Ciphertexts encrypted with a simple column transposition cipher can be decrypted with the ant colony optimization.
- To do so, the key length n is needed and the ciphertext is written row wise in n columns. These columns will then be arranged as a graph.
- Different pairs of characters arise by concatenating the columns in different orders. In each language, theses combination of characters appear with differing frequencies. Weights on edges in the graph are calculated based on theses probabilities, and frequency of an ant following a preceding ant.
- In each iteration a possible plain text is generated from a different ordering of the columns. The resulting text is then compared and rated with a given list of words of a language.
- The rating influences the pheromone matrix. The decisions of following ants is based on this pheromone matrix and hopefully theses ants will find the right solution.



The ant colony optimization Application sample 1/2

Try to decrypt the following text:

CCUSFSSEALLUOCTNYNOLRCEDITPYPONO

- Paste this sequence of characters into the text field next to "Insert ciphertext directly" and choose 4 as the key length^[1].
- Press on "Start analysis".

Configuration	
Generate cipher text	
or	
insert cipher text directly	
CCUSFSEALLUOCTNYNOLRCEDIT	PYPC
Length of cipher text: 32	
Estimated key length	
4	\$
Language of text	
English	÷
Start analysis	

[1] The length of the key can be estimated with statistic evaluations.Additionally, here the length of the ciphertext has to be a multiple of the length of the key.

The ant colony optimization Application sample 2/2

Now the two frames "Analysis" and "Visual" are activated.

There you have the following parameters:

Alpha & Beta:

 Theses parameters influence the probability of an ant to choose a certain edge. The higher the value of alpha, the more often an ant chooses a path a preceding ant had already chosen. The higher beta, the more important are

bigrams of characters.

Evaporation:

- A high evaporation lets the pheromone dropped by an ant on its way – evaporated much faster.
 So following ants will find a less intensive pheromone trace and will be influenced less.
- The pheromone matrix is calculated by these three parameters and indirectly controls the ants.
 More precise information can be found in the help.

Ant controller:

With the buttons in this sub-frame the ants can be steered on their path through the graph.

Analysis	Visualization
Level of detail	• Visualization as graph
 Knot-by-knot 	Show pheromone matrix
O Multiple iterations	
S Animation	
Ant controller	U S
To next knot	3 S 4 E T V N
Until last knot	i i
Place new ant	
Algorithm settings	
Alpha: 0.8	
-	2 S 1 F
Beta: 0.8	
-	N Y
Evaporation: 0.9	

The ant colony optimization Educational objective

Result

- Did you manage to decrypt the text given on page 20?
- As plaintext (after approx. 25 iterations^[1] with alpha=0.8, beta=0.8, evaporation=0.9) you should get:

SUCCESSFULLANTCOLONYDECRYPTIONOP^[2]

Best known decryption	
Best found plaintext	
SUCCESSFULLANTCOLONYDECRYPTIONOP	0
Kay of descention	
Key of decryption	
(4,3,2,1)	

Conclusion

- The permutation cipher is not a secure encryption method.
- The ant colony algorithm is an efficient algorithm to solve different combinatorial optimization problems. Not only in the field of cryptanalysis.
- For many problems nature has already found a solution, its just necessary to detect, understand and abstract this solution.

- [1] The number of iterations diverges a lot. It can happen, that a solution has not yet been found after 50 or more iterations. Then it pays off to restart the plugin and start from scratch.
- [2] Pad character which where appended to the ciphertext such that its length is divisible by 4.

Viterbi analysis

The idea

The problem

- Given is a running key ciphertext resulting of two plaintexts which were combined by either an XOR or a modular addition.
- Is it possible to regain the original two plaintexts?

Indeed, it is possible. The Viterbi algorithm is designed to solve such a problem.

Functionality

- The Viterbi algorithm is a recursive algorithm which uses the method of dynamic programming.
- The algorithm analyses probabilities of hidden Markov chains in a given input sequence.
- Beside cryptanalysis, the algorithm is also used in a broad range of other fields, e.g. in voice recognition or analysis of DNA structure. It is also used for the reduction of errors in transmissions.
- See http://en.wikipedia.org/wiki/Viterbi_algorithm

In the menu

"Visuals" \ "Viterbi"

The algorithm in the cryptanalytical application

- The basic concept of the algorithm is statistical evaluation of the probability of N-grams combined with the usage of a dictionary of the language in which the ciphertext is written.
- The model of the the analysis is set up with the knowledge that the ciphertext was originally constructed by modular addition or by XOR.
- The ciphertext is iterated letter by letter and possible letters for the plaintext are calculated.
 Surrounding letters build N-grams and their probabilities in the given language will be included in the reconstruction.
- The different possible letters at each position form different paths for different possible plaintexts.
 For each path a probability is generated and more unlikely paths won't be considered anymore.

Viterbi analysis

Application sample 1/2

First, we have to generate a ciphertext which can be handled by the Viterbi analysis.

The plugin therefore comes with a special generator for texts.

- Type in two plaintexts or load plaintexts from files.
- You can decide in which way (XOR or modular addition) the plaintexts are combined – letter by letter.
- By clicking on "Calculate ciphertext" a ciphertext is calculated from the given plaintexts.
- Press on "Next tab (analysis)".

	Running-Key-Encrypt	ion Viterbi-Analysis	
Running-Key-Encryption his plug-in generates a runr he cipher will be analyzed ir	ing key cipher. A running the next tab.	g key is the result of the combination of two pla	intexts
Plaintext 1	We t The com	test the functionality of the Viterbi algorithm. refore we need two plain texts which will be bined.	
Plaintext 2		will the decrypted plain texts look like in the er possible to decrpyt the cipher?	nd?
Ciphertext Select way of • XOR • Modular addition	9f 0. 9f 9i 9f 9i 0a 1 81 4 1b c 00 c 4b	a d7 d4 12 9a 18 cc d4 9c 8d c5 c6 11 8b 00 00 a 84 88 c9 84 95 41 06 88 00 00 8d 1d d4 a5 c dd 09 c9 cc 88 87 82 cf 1b 87 d4 9c 05 4b 00 b dd c5 af 9c d2 0c d4 d7 95 cf 1d 96 0c 06 cc 11 5 50 88 84 0a 9c 50 8d 11 d8 00 1b c5 d7 8b 0 c5 05 56 24 03 d7 42 12 c9 0f 03 4d 96 81 8b c	5 90 9 18 1 06 d7 0 93 5 00
Calculate ciphe	rtext		

Viterbi analysis

Application sample 2/2

In the next step, the "Viterbi analysis", the Viterbi algorithm is applied to the ciphertext. The algorithm tries to gain information about the two original plaintexts.

- Choose the language you guess the plaintexts are written in.
- Possibly, adjust the size of the N-grams and the depth of search and click on "Start analysis".
- In the lower two text areas the results are being calculated now. One can observe how the sequences of letters are being generated dynamically.

This can take some seconds.

The best way to watch the whole dialog is full screen.

How does the size of the N-grams and the search depth affect the algorithm?

	Running-Key-Encryption	Viterbi-Analysis	
/iterbi-Analysis asically the Viterbi algori	ithm is like a frequency analysis. I	epending on the probabilities of the	single
he function can be sumr . A character will be disa lf you use XOR, you are The modular addition i . The probability of this o consider longer charac To get a proper solutio . The probabilities for cc twestigated further.	narized as follows (this is just a sh seembled in all possible combinat inverting this process with anoth s inverted with a modular subtrac combinations is calculated. For th ter sequences. n you also have to take the prece imbinations are stored in an order	ort explanation for the algorithm): ions. er XOR. ion. s calculations you need a complex lau ing characters into account. ed manner. Unlikely combinations are	nguage model e not
Input			
Ciphertext	9f 0a d7 d4 12 9a 18 cc d4 9c	8d c5 c6 11 8b 00 06 90 9f 9a 84 88	c9 84 95
Load text file	41 06 88 00 00 8d 1d d4 ab C9 18 0a 1d 09 C9 cC 88 87 82 cr 1b 87 d4 9C 05 4b 00 b1 06 81 4d c5 af 9c d2 0c d4 d7 95 cf 1d 96 0c 06 cc 11 d7 1b cf 50 88 84 0a 9c 50 8d 11 d8 00 1b c5 d7 8b 00 93 00 c5 05 56 24 03 d7 42		d7 1b cf 3 d7 42
Display	12 c9 0f 03 4d 96 81 8b c5 00	4b	
Hexadecimal As text			
Parameters for the analy	rsis		
Language o N-Gr O German O English	ram size 5 ÷	Cancel	
Result			
Possible plaintext 1	he test the functuonality of tex plain at the cipher?How will the	ts termeliavorithm. Therefore we nee d	d two
Possible plaintext 2	wow will the decrepted plain th decrplexts which will be combi	e Vilood, aft in the end? Is it possible ned	to

Conclusion

- Plaintexts which were encrypted by modular addition or XOR can be decrypted with the help of the Viterbi algorithm.
- A disadvantage of the algorithm: The beginning of the revealed plaintexts is often not decrypted correctly. Surrounding N-grams are missing and paths of probabilities are not yet calculated.
- Long words are seldom decrypted in the right way.
- The used dictionary plays an important role for the quality of the resulting text, because it is the source of words for the algorithm.
- Only N-grams which are contained in the dictionary are found. The length of N-grams is limited in the plugin by N=5. As otherwise the dictionary needs to contain all words with length N. For N=7 there are already a lot of words more.
- The variation of the size of N-grams and the depth of search directly influences the result.
 - The size of N-grams determines which words from the dictionary are used.
 - The parameter depth of search determines how many candidates for plaintext pairs (paths) are used for the analysis of the next character (the algorithm discards after each character unlikely paths).

Verifiable Secret Sharing

The idea

The problem

- Verifiable Secret Sharing (VSS) is an enhanced variant of Secret Sharing.
- Secret Sharing is about sharing a common secret between multiple persons or players. Each player receives a so called "share".
- A small number of players, but not all, is needed to reconstruct the common secret.
- A single share or less than the predefined minimal amount of needed shares shall be useless.

The enhancement "Verifiable"

- VSS is more secure than normal Secret Sharing. Before sharing the secret, one person, the "dealer" needs to know the secret to share it. Before handing out the shares, he can easily modify the shares and so make them useless.
- To resolve this problem, the dealer hands out "commitments" to each player. With a commitment, each player is able to test whether his share is right or not.

Verifiable Secret Sharing

The implementation in JCT

In the menu

"Visuals" \ "Verifiable Secret Sharing"

The algorithm applied

- The secret is represented by a number (instead of a secret in form of a text).
 So a transformation between the text and the number is necessary.
- Each of the n players receive a share. For reconstruction of the secret any t shares (1 < t <= n) shall suffice.
- In mathematics, a polynomial of degree (t-1) can be reconstructed by the knowledge of t points which lie on the graph. This can be done with the so called Lagrange interpolation.
- This mathematical knowledge is used in a clever way by VSS.
- The secret is stored in the absolute term of a polynomial. Therefore the secret is simply the evaluation of the polynomial at the point 0.

Verifiable Secret Sharing

Application sample 1/2

First step

- Choose the number of players n and the minimal number of players t which is needed to reconstruct the secret.
- Determine the secret in form of a number.

The numbers "Safe prime", "Prime factor" und "Generator" are calculated automatically, if possible.

Click on "Determine coefficients".

Second step

The polynomial can now be specified. As a dealer, here you can influence the polynomial from which the shares are calculated. The commits are as well generated from the polynomial.

- The initial polynomial gives player 1 too much information. So you should generate random coefficients via the button "Generate".
- Press "Commit" to calculate the commits.

If you change the polynomial now, and later check the shares with the previously generated commits, then the shares are marked as invalid.

Click on "Calculate shares".

Number of players n	6
Number of players t for reconstruction	5
Secret s	10
Safe prime p (p>2s)	23
Prime factor q (2q=p-1)	11
Generator g	2
Next step: Determine coefficie	nts


Verifiable Secret Sharing

Application sample 2/2

Step of reconstruction

The secret is shared between the players.

- The shares can be checked for validity by clicking on "Check".
- In our example on the right, the polynomial was changed after generating the commits.
 Therefore the shares are invalid

and the dealer should not be seen as trustworthy.

Commitme	nts	Shares				Reconstrue	ction
Coefficient	Commitment Y					Player 1	
a ₀	4	Player 1	14 =	4	Check	Player 2	
aı	3	Player 2	70 =	0	Check	Player 3	
a2	5	Player 3				Player 4	
a3	4	Thayer 5	232 =	2	Check	Player 5	
a4	3	Player 4	584 =	4	Check	Player 6	
		Player 5	1234 =	4	Check		
		Player 6	2314 =	4	Check	Recons	struct

Interesting is the fact, that one share was marked as valid even though the polynomial was changed. So it is necessary to check multiple shares for validity.

- To reconstruct the secret, the players whose shares shall be used can be selected (see screenshot on the right).
- For our example we selected five players which is as well the minimal number of shares needed.
 As we have t = 5.
- Now the secret can be reconstructed by clicking on "Reconstruct" (invalid shares don't necessarily deliver a wrong secret).

Verifiable Secret Sharing

Educational objective

Conclusion

- A secret can be split between multiple persons such that it can be decrypted only in the group.
- E.g. multiple ambassadors can transmit volatile data without them knowing the important secret.
- A tolerance can be implemented such that not each of the ambassadors is needed for reconstruction.
- In the VSS, again a mathematical model, the Lagrange interpolation, can be used in an important application.



Signature demonstration

The problem

- 1. The author of electronic documents cannot be checked à priori. An attribute to verify the author is needed. This can be a signature.
- 2. Having only an electronic document one hardly can notice a belated change.

To solve this problem, an author can digitally sign his document.

Functionality

- The author generates a hash value from the document (see slide <u>41</u>).
- The hash value is encrypted with the private key of the author (if using RSA, signing is equivalent to encrypting with the private key).
- The encrypted hash value and the used hash function are made available to the public or to the receiver next to the document.
- A person who is interested in the integrity of the document, can use the public key of the author to decrypt the hash value of the document.
- By calculating the hash value of the received document and comparing it to the decrypted hash value, it is easy to ensure that the document was finally changed by the named author.

Signature demonstration

The implementation in JCT

In the menu

"Visuals" \ "Signature Demonstration"

The algorithm applied

- The plugin is capable of digitally signing a document, such as a file or an arbitrary text typed in.
- As hash method one can choose between MD5, SHA-1, and SHA-2 (SHA-256, SHA-384, and SHA-512).
- Finally, depending on the chosen hash function it is possible to choose between DSA, RSA, ECDSA, or RSA with MFG1 as signature method.
- Below that you can choose the subject (key owner) who owns an according key for the chosen signature method.*

Sig	jnature methods
\bigcirc	DSA
۲	RSA
\odot	ECDSA
\odot	RSA and MGF1
Sele	ct a key:
	*
Erik	a Mustermann - 1024Bit - de.flexiprovider.core.rsa.RSAPrivateCrtKey
Alic	e Whitehat - 1024Bit - de.flexiprovider.core.rsa.RSAPrivateCrtKey
el I	Mustermann - 1024Bit - de.flexiprovider.core.rsa.RSAPrivateCrtKey
A. P	rism - 1024Bit - de.flexiprovider.core.rsa.RSAPrivateCrtKey
ROP	whitehat - 1024Bit - deflexiprovider.core.rsa.RSAPrivateCrtKey

* There are two ways to create according keys for the subjects (key owners, users):

- a) within the Algorithm Perspective.
- b) with the visualization plugin "Public-Key Infrastructure" (JCT-PKI).

Signature demonstration

Application sample 1/2

Signing a document is neither hard nor elaborate. And can be done in two simple steps.

First step: Create hash value

- Via "Choose input" the document to be signed can be chosen.
- A dialog appears and either a text can be typed in directly or an arbitrary document can be opened with "From file".
- Next a "Hash function" must be selected.
- The hash value is then be generated and is shown in the lower left area. This hash value is an electronic fingerprint of the document.



Signature demonstration Application sample 2/2

Second step: Create signature

- Clicking "Signature function", an encryption algorithm for the hash value can be selected.
- We choose "ECDSA" as signature method.
 Below we choose from the JCT keystore a key of the signing person (here: "Alice Whitehead").
- By clicking on "Finish", the signature is generated and can finally be displayed via "Show generated signature".

	5	
Owner of	signature: -	
Used key,	curve: ANSI X9.62 prime256v	(256 bits)
Signature	method: SHA384withECDSA	
Signature		
Address	Hex	Ascii
00000	30 44 02 20 02 E9 72 99 19 80 A4 A1 4B 5E	0D ér¤jK^
0000E	14 2A 6F 8D CB 16 9F 74 71 52 73 39 5A 3D	*oËtqRs9Z=
0001C	B4 C4 CF C0 2D 5F 3B 15 02 20 6D B0 1C 44	´ĂĬÀ; m°D
0002A	63 38 CA 91 F8 C8 0D 40 58 ED 79 A2 0F AD	c8ÊøÈ
00038	73 78 BE 92 12 95 C0 E4 CA 58 58 5F 75 1B	sx¾ÀäÊXX_u
• Hex	dump (hex and ascii) Octal ODeci	mal 🔿 Hex
• Hex	dump (hex and ascii) Octal ODeci	mal 🔘 Hex
• Hex Signed me	dump (hex and ascii) Octal Occi essage Hex	mal O Hex
Hex Signed m Address 00000	dump (hex and ascii) Octal Deci essage Hex 53 6F 6D 65 20 61 72 62 69 74 72 61 72 79	mal O Hex Ascii Some arbitrary
Hex Signed mathematical Address 00000 0000E	dump (hex and ascii) Octal Deci essage Hex 53 6F 6D 65 20 61 72 62 69 74 72 61 72 79 20 74 65 78 74 20 79 6F 75 20 68 61 76 65	Ascii Some arbitrary text you have
Hex Signed ma Address 00000 0000E 0001C	dump (hex and ascii) Octal Deci essage Hex 53 6F 6D 65 20 61 72 62 69 74 72 61 72 79 20 74 65 78 74 20 79 6F 75 20 68 61 76 65 20 74 6F 20 74 79 70 65 20 69 6E 20 68 65	Mal Hex Ascii Some arbitrary text you have to type in he
Hex Signed ma Address 00000 0000E 0001C 0002A	dump (hex and ascii) Octal Deci essage Hex 53 6F 6D 65 20 61 72 62 69 74 72 61 72 79 20 74 65 78 74 20 79 6F 75 20 68 61 76 65 20 74 6F 20 74 79 70 65 20 69 6E 20 68 65 72 65 2E 20 49 74 20 77 69 6C 6C 20 62 65	Ascii Some arbitrary text you have to type in he re. It will be
Hex Signed m Address 00000 0001C 0002A 00038	dump (hex and ascii) Octal Deci essage Hex 53 6F 6D 65 20 61 72 62 69 74 72 61 72 79 20 74 65 78 74 20 79 6F 75 20 68 61 76 65 20 74 6F 20 74 79 70 65 20 69 6E 20 68 65 72 65 2E 20 49 74 20 77 69 6C 6C 20 62 65 20 74 68 65 6E 20 73 69 67 6E 65 64 2E	Ascii Some arbitrary text you have to type in he re. It will be then signed.
Hex Signed m Address 00000 0000E 0001C 0002A 00038	dump (hex and ascii) Octal Deci essage Hex 53 6F 6D 65 20 61 72 62 69 74 72 61 72 79 20 74 65 78 74 20 79 6F 75 20 68 61 76 65 20 74 6F 20 74 79 70 65 20 69 6E 20 68 65 72 65 2E 20 49 74 20 77 69 6C 6C 20 62 65 20 74 68 65 6E 20 73 69 67 6E 65 64 2E	Ascii Some arbitrary text you have to type in he re. It will be then signed.
Hex Signed m Address 00000 0000E 0001C 0002A 00038	dump (hex and ascii) Octal Deci essage Hex 53 6F 6D 65 20 61 72 62 69 74 72 61 72 79 20 74 65 78 74 20 79 6F 75 20 68 61 76 65 20 74 6F 20 74 79 70 65 20 69 6E 20 68 65 72 65 2E 20 49 74 20 77 69 6C 6C 20 62 65 20 74 68 65 6E 20 73 69 67 6E 65 64 2E	Ascii Some arbitrary text you have to type in he re. It will be then signed.
Hex Signed m Address 00000 0000E 0001C 0002A 00038	dump (hex and ascii) Octal Deci essage Hex 53 6F 6D 65 20 61 72 62 69 74 72 61 72 79 20 74 65 78 74 20 79 6F 75 20 68 61 76 65 20 74 6F 20 74 79 70 65 20 69 6E 20 68 65 72 65 2E 20 49 74 20 77 69 6C 6C 20 62 65 20 74 68 65 6E 20 73 69 67 6E 65 64 2E signed message: 552 Bits	Mal Hex Ascii Some arbitrary text you have to type in he re. It will be then signed.
Hex Signed m Address 00000 0000E 0001C 0002A 00038 Length of	dump (hex and ascii) Octal Deci essage Hex 53 6F 6D 65 20 61 72 62 69 74 72 61 72 79 20 74 65 78 74 20 79 6F 75 20 68 61 76 65 20 74 6F 20 74 79 70 65 20 69 6E 20 68 65 72 65 2E 20 49 74 20 77 69 6C 6C 20 62 65 20 74 68 65 6E 20 73 69 67 6E 65 64 2E signed message: 552 Bits	Mal Hex Ascii Some arbitrary text you have to type in he re. It will be then signed.
Hex Signed m Address 00000 0000E 0001C 0002A 00038 Length of To show with the l	dump (hex and ascii) Octal Deci essage Hex 53 6F 6D 65 20 61 72 62 69 74 72 61 72 79 20 74 65 78 74 20 79 6F 75 20 68 61 76 65 20 74 6F 20 74 79 70 65 20 69 6E 20 68 65 72 65 2E 20 49 74 20 77 69 6C 6C 20 62 65 20 74 68 65 6E 20 73 69 67 6E 65 64 2E signed message: 552 Bits the signed file and the signature, click on "Save" tex editor from JCT.	Ascii Some arbitrary text you have to type in he re. It will be then signed. and then open the saved file
Hex Signed m Address 00000 0000E 0001C 0002A 00038 Length of To show with the l	dump (hex and ascii) Octal Deci essage Hex 53 6F 6D 65 20 61 72 62 69 74 72 61 72 79 20 74 65 78 74 20 79 6F 75 20 68 61 76 65 20 74 6F 20 74 79 70 65 20 69 6E 20 68 65 72 65 2E 20 49 74 20 77 69 6C 6C 20 62 65 20 74 68 65 6E 20 73 69 67 6E 65 64 2E signed message: 552 Bits the signed file and the signature, click on "Save"	mal Hex Ascii Some arbitrary text you have to type in he re. It will be then signed. and then open the saved file
Hex Signed m Address 00000 0000E 0001C 0002A 00038 Length of To show with the l	dump (hex and ascii) Octal Deci essage Hex 53 6F 6D 65 20 61 72 62 69 74 72 61 72 79 20 74 65 78 74 20 79 6F 75 20 68 61 76 65 20 74 6F 20 74 79 70 65 20 69 6E 20 68 65 72 65 2E 20 49 74 20 77 69 6C 6C 20 62 65 20 74 68 65 6E 20 73 69 67 6E 65 64 2E signed message: 552 Bits the signed file and the signature, click on "Save" tex editor from JCT.	Mal Hex Ascii Some arbitrary text you have to type in he re. It will be then signed. and then open the saved file Save Close

Signature demonstration Educational objective

Conclusion

- The integrity of electronic documents can be checked with the help of electronic signatures.
- Cryptographic algorithms help to verify the author and the integrity of the document.
- If a document was changed in any way, the hash value changes.
- To make sure the document was created by the named author, its author signs it with his private key.
 Only with the "right" public key (the one from the signing person) it is possible, to validate the original hash value (to verify the integrity of the document).

So the hash value can be publicly accessible, without being endangered to be changed.

Extended RSA cryptosystem

Which encryption ciphers are used nowadays, which guarantee security?

- For data which is transmitted over public channels, an encryption method should be used. One possible cipher for such tasks is the RSA cryptosystem (if used with the correct parameters).
- The RSA cipher is an asymmetric method. To communicate each participant needs two keys, a private and a public key. These two keys have to be generated first.
- Data which was encrypted with the public key of one participant can only be decrypted with the corresponding private key.
- To communicate in an encrypted manner with another person, one has to have his public key.
 Therefore, the public keys have to be exchanged preliminarily.
 - A "Certificate Authority" (PKI) is often used to simplify the process. This "authority" saves, manages and verifies the public keys of the possible communicators and generates certificates.
 - → See the visualization plugin "Public-Key Infrastructure" (JCT-PKI) which visualizes the processes within a PKI with its instances User, RA and CA.

Extended RSA cryptosystem

The implementation in JCT

In the menu

"Visuals" \ "Extended RSA Cryptosystem"

Functionality

- This plugin implemented in JCT helps managing identities and their associated keys, and it offers
 a complete independent communication platform to send and receive messages.
- Further on, it is possible to attack the system via attacking the key. Therefore a brute-force method is used to factor the modulus "n" into its primes.
- The user can experiment and find security holes of the RSA cryptosystem.

Extended RSA cryptosystem Application sample 1/2

Generation of primes

- First, we generate a key which can then be attacked.
- Therefore the plugin provides the option "Manage keys". We choose primes p and q and a random e.
- Finally, the key has to be saved in a keystore using a password, which we enter in the lower right.
- Now we have created a key for the identity "Alice Whitehat". Next we try to attack the keys. Using the RSA crypto-system this means solving the factorization of the modulus n = p*q.

Actions:	Action window
Encrypt and send message	Create new key Create new key (extended) My keys Select two different prime numbers p and q, and an exponent e:
Receive and decrypt message	• RSA
Manage keys	p: 991 v q: 971 v
Attack public key	e: 61121 v Pick random 'e'
	O Multi-prime RSA
	Number of primes (3–5): 3 0
	p: v q: v r: v
	e: v Pick random 'e'
	Enter password:
	Repeat password:
	Create key

 As Alice knows the keys and will not attack her own keys, we switch the tab to "Bob Whitehat". (Alice and Bob are default identities in JCT.)

Extended RSA cryptosystem Application sample 2/2

The attack

- Bob Whitehat is now able to attack the public key of Alice.
- So, you click the button "Attack public key" on behalf of Bob and choose the according key of Alice. The previously generated key can be recognized at its bit length (here 20 bit).
- Due to the short bit length, the key can be attacked via the button "Attack key". So the key
 generated by Alice can be factorized without knowing the primes p and q.
- Here, the factorizing is done only via a brute-force attack.
- A bit length of only 20 bit for the modulus does by far not offer appropriate security for the RSA cipher.

	Action window	/	
Actions:	Alice Whiteha	t – 1024Bit – RSAPublicKev – KevID: 9	
and and an end of the second o	Alice Whiteha	t – 20Bit – RSAPublicKev – KevID: 1	
rypt and send message	Erika Mustern	nann – 1024Bit – RSAPublicKev – KevID: 6	
	Alice Whiteha	it – 20Bit – RSAPublicKey – KeylD: 5	Reconstruct key
ve and decrypt message	Erika Mustern	nann – 1024Bit – RSAPublicKey – KeylD: 8	
	Alice Whiteha	t – 1024Bit – RSAPublicKev – KevID: 7	
Manage keys	Alice Whiteha	t - 1024Bit - RSAPublicKey - KeylD: 10 e modulus N of Alice Whitehat:	
	Erika Mustern	nann – 1024Bit – RSAPublicKev – KevID: 2	
Attack public key			
	N. 502201		
-			
т	The factorizatio	on was successful. The following values were reconstructed in 0.022 seconds:	
Т	The factorizatio	on was successful. The following values were reconstructed in 0.022 seconds:	
T	The factorizatio	on was successful. The following values were reconstructed in 0.022 seconds:	
T P	The factorizatio Parameter P	on was successful. The following values were reconstructed in 0.022 seconds: Value 971	
T F C	The factorizatio Parameter p q	on was successful. The following values were reconstructed in 0.022 seconds: Value 971 991	
T F C C C	The factorizatio Parameter 9 9	Value 971 991 733	
T P c c c c	The factorizatio Parameter p q e d	Value 971 991 733 233197	
T P C C C C C C C C C C C C C C C C C C	The factorizatic Parameter p q e d	on was successful. The following values were reconstructed in 0.022 seconds: Value 971 991 733 233197	
T F C C C C C C C C C C C C C C C C C C	The factorizatic Parameter p q e d	on was successful. The following values were reconstructed in 0.022 seconds: Value 971 991 733 233197	
T P C C C C C C C C	The factorizatic Parameter p q e d	Value 971 991 733 233197	
T F G G G G G G	The factorizatic Parameter p q e d	Value 971 991 733 233197	
T F G G d d	The factorizatio Parameter p q e d	on was successful. The following values were reconstructed in 0.022 seconds: Value 971 991 733 233197	
T P c c c c c c	The factorizatio Parameter p q e d	on was successful. The following values were reconstructed in 0.022 seconds: Value 971 991 733 233197	
T P c c c c c c c c c c c c c c c c c c	The factorizatic Parameter p q e d	Value 971 991 733 233197	

Extended RSA cryptosystem Educational objective

Conclusion

- The factorization methods allows us to factorize numbers with a short bit length in almost no time. Given a modulus n with only 64 bit (binary representation of the number has 64 digits, which is around 20 decimal digits, like the number 2^64-15) for instance can be factorized with a current notebook (Intel Core i7 2,4GHz) in less than a second.
- Once an attacker can find a factorization of the modulus n, the messages which are sent from the associated identity can be decrypted by the attacker.
- Nowadays, bit length of 2048 bit are rated as secure.

And more ...

• The plugin offers the possibility to send and attack messages encrypted with the RSA cipher.

SETUP attack on the RSA key generation (Kleptography) The idea

Problem

- There are some "backdoor" attacks, which make the RSA cipher insecure.
- Most of these attacks start by modifying the key generation. The user needs to rely on the random generation of the primes this is not always possible.
- The SETUP ("secretly embedded trapdoor with universal protection") attack is such an attack where the generation of the key is modified.

A short description of the attack:

Functionality

- Some extra values and keys are injected into the system.
- The public keys, which are needed by the RSA method, are modified such that information needed for decryption can easily be extracted by the attacker. However, without knowing the implementation of the key generation, one can hardly detect that it is not really random.

SETUP attack on RSA

The implementation in JCT

In the menu

"Visuals" \ "Kleptography"

Functionality in detail

- Generally the RSA cipher uses two randomly generated private primes P and Q. Their product, the modulus N = P*Q is published.
- For the attack, initially the prime number P is generated, then this prime is encrypted with the public key of the attacker. Next the prime Q will be chosen such that the first digits of the modulus N represent the encrypted value of P.
- As N is publicly available, the attacker can easily reveal the prime P by decrypting the first digits of the modulus N with his own private key, and the cipher is hacked.
- As only the encrypted prime number P is part of the modulus N and P was randomly chosen, the modulus seems to be random too.

Moreover, as P will be regenerated for each new pair of keys the attack is not detectable without reverse engineering the code of the key generator.

SETUP attack on RSA Application sample 1/2

The attack is divided into two main steps: the generation of the keys and the decryption by the attacker.

Key generation

- In the dropdown menu choose the method "Attack 4: SETUP".
- First, the two keys of the attacker have to be generated. This is done by "Generate new attacker keys".
- Next, the primes P and Q which are used in the ordinary key generation can be generated.
- The prime Q will be chosen such that the modulus N contains the encrypted prime P (marked yellow in the figure).

ey generation		
Settings		
Att	ack 4: SETUP	Binary O Decimal • Hexadecimal
ey bit length: 64	 (in decimal) 	
Additional cryptosystem	values	
	Gen	nerate new attacker keys
ttacker's N	d7ffe043	
ttacker's E	c457ec9d	
ncrypted P	80e4885c	
(temporary composite)	80e4885c84c43eb7	
Standard cryptosystem va	alues	
		Generate all at once
Conorato primos R and	P (prime)	Q (prime)
Generate primes P and	b9cedb01	b19580e1
Calculate N	N = P * Q	
Carculate N	80e4885be4e3fbe1	
	E (public exponent)	Generate random E Bestore default E
	10001	
Calculate D	D (private exponent)	

- Finally, N and D can be generated. Then, in the lower third part of the plugin a plaintext can be encrypted.
- By clicking on the button "Save public key and ciphertext", the user can switch to the tab "SETUP attack" to continue and decrypt the ciphertext.

SETUP attack on RSA Application sample 2/2

The decryption by the attacker

- Switch to the tab "SETUP attack"
- The data known by the attacker is directly shown in the appropriate fields: These are the keys of the attacker, the modulus N and the exponent E. The last two values are public, as the communication partner needs them to encrypt the text.
- Using the four buttons on the left, the text can be decrypted by the attacker.
- First, the encrypted prime P is extracted from the modulus N, and decrypted with the attacker's private key.

	Key ge	neration and encryption SETUP attac			
a attacker's perspective					
goal of an attacker is to use publicly availab	ble information, i.e. the public keys and ci	phertexts, to obtain the information that	t he or she needs to factor the key composite N and thus recalculate th	ne private	
onent D.	, , , , , , , , , , , , , , , , , , , ,		, , , , , , , , , , , , , , , , , , , ,		
	Public law				
Decrypt encrypted P	Public Rey				
Calculate private keys	The public key is by definition public	y accessible and thus visible to an attac	ker. This key was copied directly from the first tab.		
carcolate private keys	N (composite)		E (public exponent)		
Decrypt ciphertexts	80e4885be4e3fbe1		10001		
Back to Key Conception and Ecomotion	Additional data				
Back to key Generation and Encryption	The encrypted P is read from the upp	er bits of the public key composite N, ar	d to decrypt it the attacker will need his or her own private key.		
	Encounted P		Attacker's D (private exponent)		
	80e4885b		2278a965		
			22100303		
	Calculations				
	The attacker will find the prime P enc	rypted with his or her public key stored	in the upper bits of the composite N. Because of a potential carry bit in	division	
	and then D but must also use P' to fin	e attacker must also calculate P by addi id Q' and D'.	ng one to the encrypted P before decrypting. The attacker can use P to	nna Q	
	Decrypted P		Decrypted P ^(P+1)		
arrying out a SETUP attack	Ac3c9a7e		beergheed () by		
ep 7: Extract the encrypted prime P from	Q = N / decrypted P		Q' = N / decrypted P'		
ecrypting this value with the attacker's	1b0d114bd		b19580e1		
ivate key yields the prime P, unless a prow bit was taken in the earlier	D (private exponent)		D' (private exponent)		
vision. Since the attacker can't know if	1f3fc8f3039a1c8d		1698fa96f22a4001		
e encrypted P and that value plus one.	Cinhertayt				
on R: Using P and P' along with the					
ablic key allows the attacker to	The ciphertext is transmitted publicly	, so it is entirely visible to an attacker o	any other party monitoring the communication channel.		
produce the second prime Q and then e private key. The attacker must		6256b6895746b322314795394854a 995646079e65f9ff5203b318032a62	a7402a70a603a6c291c6e9bb98f78 aa37669864c79d31284231696a287		
lculate Q and Q', although only one of		bbb619b4ba340971843f7cd3ff304c	147c32ee7451324a07de4534e48bd		
em will be prime and divide N eveniy.		3608d6422ce84ece7ab281b0f690ec 5bc361b0ef738be0261e066193b89	1525a181d58ed49b43775aa5d25c		
ep 9: You now have two possible keys to		6cfe1dbfcf2			
ould already know which is correct, but					
aintext message and the other	Decrypted texts				
onsense.	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		
		制模机度谱模块钳制。目示+3-2 国久德	is it really possible to decrypt this text with this attack? We will see		
	■個本11回營D種用幹約未用国烈片發防費	≥を愛知掌國國評測公司→♥愛兴♥2♥			

Because of a potential carry bit two different cases have to be analyzed.

SETUP attack on RSA Educational objective

Conclusion

- By cleverly encroaching the key generation, an attacker has the possibility to decrypt the cipher text with the use of his own keys.
- Almost all effective attacks on RSA attack the key generation. Therefore, one has to confide to the key generation, which is often done by a "Certificate Authority" (CA) or within a hardware security module (HSM).
- As the modulus N still appears to be random, as P and Q are chosen differently for each pair of keys, it is hard to detect the attack by just analyzing the output – without applying reverse engineering.
- For this attack, only the public key of the attacker is needed. So, revealing his attack does not cause any insecurity for his communication.

Zero-knowledge protocol: Fiat Shamir

Problem

- A person A wants to convince a second person B that he knows a secret which person B knows as well.
- It is required to do the verification in public without revealing the whole secret. So a possible attack from a third person will disclose the secret.
- A solution for this problem is called a zero-knowledge protocol.
- An important characteristic for such a protocol is its need for honest players. A third person C shall not be able to convince B of knowing the secret, without really knowing it.

In this application sample we present the zero-knowledge protocol from Fiat Shamir. There exist a couple more zero-knowledge protocols, like Feige Fiat Shamir, or a version using an isomorphism for graphs.

Zero-knowledge protocol: Fiat Shamir The implementation in JCT

In the menu

"Visuals" \ "Fiat Shamir"

Functionality

- The Fiat Shamir method relies on the difficulty of the following problem: Given an arbitrary number in the field modulo n, its square root can only be found by factoring of the number n.
- If the modulus n is a product of two unknown primes p and q which are chosen large enough, it is hardly possible to find the factorization of n.
- As the method operates on numbers, the secret s must be given as a number.
- Person A published the number v = s² mod n, generates a random number r < n, and receives another random number b. b is 0 or 1. Person B now receives from person A the number x = r² mod n.
- Person A calculates y = rs^b mod n and sends this number to person B. Person B verifies if the equation y²=xv^b mod n holds. If it does, the secret is verified, due to the fact:

$$y^2 = (rs^b)^2 = r^2s^{2b} = xv^b \mod n$$

Zero-knowledge protocol: Fiat Shamir Application sample 1/2

As prover

- Choose the radio button "Prover".
- First, the two primes p und q have to be generated. Their product is the public modulus n. Additionally, the secret s has to be generated.
- In the section "Action flow" the steps which are required for the verification can be executed.
- All values public and private ones which are calculated during this process are shown in the lower part of the plugin.
- In this example, Alice performs the proof, as she actually knows the secret. Her communication partner will verify that she knows the secret (green hint in the lower left part of the figure).

Si	tuation				
•	Prover Choose two Attacker with gcd(s,	o prime numbers $(n) = 1$ and $v = s^2$	p and q. Th mod n. v is	e digit n = pq is the pub s published.	lic module. Alice knows a digit s < n
In	put for a prime number				
0	139		n : 2!		Generate prime numbers
,	181				Generate secret
1					
	Take prime nu	mbers			
A	ction Flow				
		Generate rand	lom numb	Alice generates a random sends $x = r^2 \mod n$ to be	n digit r < n and ob.
		Generat	te b	Bob generates a random and sends b to Alice.	bit b from {0,1}
		Calculate a	answer	Alice computes an answer b and sends y to Bob.	er y depending on
				$y = rs^b.$	
		Verif	У	Bob verifies Alice's ans whether it's right: y ² = xv^b.	wer. He checks,
(Reset			Rerun	Run several times
	6				
m	Rob		Alice		
	b: 0		Secret		Public
	y²: 1406		r	: 1664	$v = s^2 \mod n 24303$
	x v^b : 1406		S	: 2903	y: 1664
					$x = r^2 \mod n \ 1406$

Zero-knowledge protocol: Fiat Shamir Application sample 2/2

As attacker

- On the other hand, the plugin offers the possibility to act as an "attacker" who pretends to know the secret.
- By cleverly choosing the values x and y it is possible to convince the other person in 50 percent of the cases that one knows the secret.
- This can be done in this scenario.
 By repeating the method multiple times, the probability to detect the attacker is 1-(0,5)ⁿ.
- The more often the test is repeated the higher is the probability to detect the attacker.

Situation							
Prover Cho Attacker Ali	ose two prime nun ce and knew a digit	nbers p and q s with v = s ²	. The digit r mod n. v is	n = pq is the pub published.	olic module	e. Carol affirms si	he was
Input for a prime nu	mber						
211		n: 3!				Generate prir	me numbers
167		_				Cenerate	a secret
						Generati	eserrer
Take pr	ime numbers						
Action Flow							
			Carol gen	erates a random d	ligitr < n ar	ud a hit	
	Generate ra	indom numb	c from {0	,1}. She sends x =	r ² v^-c mod	in to Bob.	
			Bob gener	ates a random bit	b from (0.1	} and	
	Gene	raté b	sends b to	Carol.		,	
			Carol com	putes an answer v	and sends	v to Bob:	
	Calculat	e answer	y = r.	, , , , , , , , , , , , , , , , , , , ,			
	Ve	rifu	Bob verifi	es Carols answe	r. He check	s, whether	
	ve	rity	$y^2 = xv^2$).			
	set		Por			Pup course	ltimes
K	:501		Ker	un		Kull Severa	a unies
Information							
Bob		Caro	I				
b: :	L	Sec	ret	2000	P	ublic	
γ ² :	5711		r:	2969		v = s* mod n	13374
x v^b :	20335		S :	1		у:	2969
Has not been verifie	d		c :	0		$x = r^2 v^{\Lambda} - c$:	5711



Zero-knowledge protocol: Fiat Shamir Educational objective

Conclusion

- Zero-knowledge protocols are methods which are used to convince someone else of owning a secret without handing out the secret.
- The Fiat-Shamir protocol is such a method.
- It is important to know that an attacker can fake the result with a probability of (0,5)ⁿ.
 Here, n is the number of repetitions of the test. The more often the method is repeated the better is the quality of the result.
- Hint: If it is possible to factorize large numbers easily, then this method is not be secure anymore (this means, that then the above described probabilities don't hold any more).

Android Unlock Pattern (AUP) The idea

Problem

Nowadays, smart phones offer – next to writing messages and calling – a lot more functions, e.g. checking mails, creating notes, or online banking.

Using such functions implies storing much sensible data on the phone (or in a cloud).

- People who lost their smartphone often ask themselves, whether it is possible for others to access their data. How secure is the lock of the smartphone? What is the difference between the security of a common PIN and the Android Unlock Pattern which is used by Android devices.
- The Android Unlock Pattern is visualized in JCT, and in its online help the security evaluation is documented and compared with other unlock patterns.

Android Unlock Pattern

The implementation in JCT

In the menu

"Visuals" \ "Android Unlock Pattern (AUP)"

Functionality

- The Android Unlock Pattern can be used on smartphones running on Android to lock the screen. Typically nine points on the screen are arranged as a square. The user can create a pattern by connecting the dots (under certain rules). This pattern has to entered before using the phone.
- In the visual in JCT the user can check different patterns concerning their security. Therefore, a security indicator is provided. The indicator shows the number of different patterns possible with the used number of points of the pattern.

Android Unlock Pattern

Application sample

Set pattern

- The visuals come along with the typical unlock screen of Android.
- First, a pattern can be set by clicking on one of the points and the moving the mouse over the other points. To finish the pattern you click on the last point of the pattern.
- Once created, in the lower right text field the security indicator shows the possible permutations of a pattern with the same amount of points.

For instance, there are 8776 possible combinations for a pattern with five points.

Change pattern, check pattern

 The plugin also provides the possibility to save a pattern, and then draw a second pattern to compare it with the saved one.



The stored pattern can also be changed. Therefore, you either need to know stored one.
 If you forgot the pattern the visual can simply be reset.

Android Unlock Pattern

Educational objective

Conclusion

- For a Android Unlock Pattern the order of the used points is important.
- A pattern for the Android unlock screen has to fulfill some rules. For example, each point can only be visited once.
- Due to this (and some more) rules the possible number of patterns shrinks. In total there are 389,112 different patterns.
- Comparing this AUP pattern to a 4 to 9 digit PIN of the numbers 1 to 9, where each number can be used only once, there are 985,824 different PIN combinations. The Android pattern fulfills the following rule: A connection of two points, where the connection line

crosses an unused point, is not a valid. If this rules was not applied, there would exist as many combinations as for the PIN, where each number can be used only once.

Cascades in the Actions window The idea

Functionality

- In the Actions window, sequences of application of crypto methods (cascades) can be recorded and reapplied. Basically, it's a recorded and player for JCT functions.
- Arbitrary many function calls can be recorded and reapplied in the JCT Default Perspective.
- Cascades of classic crypto methods can also be viewed in the crypto console (see slide <u>73</u>).

Examples of application

- Multiple files can quickly be encrypted or decrypted with the same algorithms, settings and ordering of the algorithm.
- Commutativity, the exchangeability of the order of different encryption algorithms, can easily be investigated with this cascade functionality (see slides <u>66</u> ff).

The Actions window allows to automate and re-run procedures – similar as with batch files on the command prompt.

In some cases, recorded cascades may not yield the exact same result after playback as was recorded.

🕭 File Exp	lorer 🖉 Actions	x		
		L° 😤 🕨	× 🖂 🕹	\bigtriangledown
Action	Algorithm	Filename		
	Bifid	out001.txt]	
	LFSR	out002.bin		
(X:	Vigenère	out003.txt		
Bifid				\wedge
Playfair/	/alike alphabet	(25chars, w/	o "J")	
nullchar	: 0			
key:	WA			
key2:	null			
outputIS	5: null			
transfor	mData:			
upper/lo	wercase=upper	case, filterBlan	ks=on,	
filterUml	auts=on			
filterNo	nAlphaChars:	true		

The implementation in JCT

In the menu

"Window" $\$ "Show view" $\$ "Actions"

Create a recording

- To start recording a cascade press (Section 2).
- All algorithms being executed now are recorded.
- To finish an recording just press (Response) again.

Edit, store and rerun a recording

- In the list below the toolbar, all algorithms are displayed in the order they have been processed.
- By selecting an algorithm, its execution details (e.g. alphabet, key etc.) are shown in the area below.
- Now the recorded cascade can be applied to an opened file in JCT by pressing .
- Use the buttons and and to simply export or import a cascade simply (save as / load from a file).

墨 File Explorer 🖉 Actions 🛛 👘 👘				
	Ľ	🎭 🕨 🗶 🎽	∇	
Action	Algorithm	Filename		
(×	Bifid	out001.txt		
	LFSR	out002.bin		
X	Vigenère	out003.txt		
Bifid Playfair, w/o "J" nullchai key: key2: output! transfor upper/lo filterBlan filterNo	/alike alphab) r: 0 WA null S: null fmData: owercase=uppo ks=on, filterUp nAlphaChars:	et (25chars, ercase, mlauts=on true		

Application sample 1/3

In this example we show, that the order of a Caesar and a transposition cipher can be exchanged in the decryption process (**commutativity**).

A first recording

- Start recording the cascade with 😤.
- Encrypt an arbitrary text with Caesar:
 "Algorithms" \ "Classic" \ "Caesar"
- Add a transposition cipher and encrypt: "Algorithms" \ "Classic " \ "Transposition"
- Apply a transposition **de**cryption which reverts the last encryption: Therefore, simply use the same settings as for the encryption, but just use "Decrypt".
- **De**cryption of the Caesar encryption applied first.
- Stop the recording with
 Stop the recording with

The action window should now look like the figure on the right.

😹 File Explorer 🖉 Actions 🛛 👘						
	🗳	<mark>≀ ▶ ≭ थ</mark> थ	\bigtriangledown			
Action	Algorithm	Filename				
X	Caesar	out006.txt				
	Transposition	out007.txt				
P	Transposition	out008.txt				
×:	Caesar	out009.txt				
Caesar			\mathbf{A}			
Upper a	nd lower Latin	(A-Z,a-z)				
nullcha	r: 0					
key:	К					
key2:	null					
output	S : null					
transfor	transformData					
filterBlan	nks=on, filterUm	lauts=on				
filterNo	nAlphaChars	true				

Application sample 2/3

The cascade we created on the last slide should output a text unchanged, as each ciphertext will directly be decrypted afterwards.

The current sequence of the algorithms

- Now there should be the following sequence of crypto operations, where E stands for encryption and D for decryption.
 - --> E (Caesar)
 --> E (Transposition)
 --> D (Transposition)
 --> D (Caesar)
- Note the different layers of algorithms and its inverse.
 Such a structure will always output the original plaintext.
 So all the functions together form a identity transformation.
- So the question arises: When could we rearrange the order of the calls of the decryption algorithms such that a text will still be "decrypted" to itself?

😹 File Exp	olorer 🖉 Actions	; 🖾 🦳 🖾 ;	3
	LŤ 🤗	<mark>: ▶ ≭ थे थ</mark>	\bigtriangledown
Action	Algorithm	Filename	
E	Caesar	out006.txt	
X	Transposition	out007.txt	
P	Transposition	out008.txt	
×:	Caesar	out009.txt	
Caesar			^
Upper a	nd lower Latin	(A-Z,a-z)	
nullcha	r: 0		
key:	К		
key2:	null		
outputl	S: null		
transfo	rmData:		
filterBlar	nks=on, filterUm	lauts=on	
filterNo	nAlphaChars:	true	

Application sample 3/3

Now we want to reorder our decryption algorithm and observe what happens to the output.

Rearrange a recording

 By right-clicking on a row (e.g. the Caesar decryption), a context menu appears, allowing the user to exchange the position in the call stack ("Move up" / "Move down").

A new ordering

- Rearrange the stack to the following:
 - --> E (Caesar)
 - --> E (Transposition)
 - --> D (Caesar)
 - --> D (Transposition)
- Open a text file in JCT.
- Apply the new stack to the opened file by clicking on

What happens to the plaintext?

Does this also work with other encryption methods?



Educational objective

Conclusion

 The cascade function is perfect for saving and automatically applying different sequences of cryptographic operations to multiple files at once.

Conclusion with a sample

- A text which was encrypted by the Caesar and a transposition cipher can be decrypted in arbitrary order. So these methods are commutative.
- This is possible as the Caesar method shifts each character by a fixed number of characters in the alphabet and the transposition cipher permutes each character in the text. Both methods are applied to the exactly same objects.

It would be same by taking a monoalphabetic substitution instead of Caesar.

Many methods (e.g. ADFGVX and Playfair) use a technique so called "fractioning".
 For instance, pairs of characters are substituted but then single characters are permuted. In this way substitution and transposition are not commutative any longer.

Variable alphabets for classic algorithms The idea

User-defined alphabets

- For most of the classic encryption algorithms (e.g. Vigenère), the ciphertext depends on the alphabet used in the plaintext.
- Frequently used alphabets are upper- or lowercase alphabets (A-Z, a-z) with or without digits (0-9).
- Many cryptographic tools restrict themselves to a fixed set of alphabets or characters, or an alphabet has to be entered manually.
- In order to improve the usability, a user should be able to easily create and test an encryption method with his own alphabets to get a feeling for the importance of encryption alphabets.
- JCT provides the following solution:
 - A custom alphabet can always be created for classic encryption algorithms in the appropriate encryption wizard.
 - Own alphabets can be built by arranging frequently used building blocks.

$\Theta \bigcirc \Theta$	Add a new Alphabet
Alphabet Enter a name and the cha	acters for the new alphabet
1) Enter a name for the a	lphabet:
German Alphabet	
2) Define the alphabet cl	naracters:
• Compose alphabet cl	naracters from 'alphabet blocks'
 Enter alphabet chara 	tters manually
Click on the alphabet	blocks to compose a new alphabet:
A-Z a-z 0	-9 aous AOU [Space] Linebreak (Unix/Windows)!? ASCII beyond latin alphabet
	+ New alphabet block
Selected blocks and re	sulting alphabet:
äöüß ÄÖÜ	A-Z a-z
(Right-click on the selected blo	cks to reorder and (slightly) transform them)
Result (composed alph	abet):
Alphabet characters:	äöüßÄÖÜABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz
Alphabet length:	59

Variable alphabets for classic algorithms

The implementation in JCT 1/2

Generate a custom alphabet

If a methods supports custom alphabets, a user can always provide and create an alphabet for deand encryption on-the-fly.

🚭 Caesar	— 🗆 X	
Caesar		
Select an alphabet and enter a key.		
Operation		
Encrypt	◯ Decrypt	
Alphabet (current length: 52)		
Only alphabet characters will be processed.		
Plain-/Ciphertext alphabet: Select alpha Custom alpha	bet: Upper and lower Latin (A-Z,a-z) habet	
Filter non-alphabet characters from the input text before	e the operation.	
Key		
Enter key using a character: C	or the amount of shift along the alphabet: 2 \sim	
Interpretation of the first alphabet character: \bigcirc Shift = 0	⊖ Shift = 1	
Pre-operation text transformation		
Apply alphabet-fitting text transformations first (see new	rt page)	
JCT command line		

Special characters which are not on the keyboard, can be entered in curly brackets via their ASCII value. E.g. {10} represent a line break.

Inhahas	Add a new Alphabet	
Enter a name a	and the characters for the new alphabet	
1) Enter a na	me for the alphabet:	
German Alph	abet	
2) Define the	alphabet characters:	
Compose	alphabet characters from 'alphabet blocks'	
O Enter alph	habet characters manually	
Click on th	e alphabet blocks to compose a new alphabet:	
A-Z	a-z 0-9 aous AOU [Space] Linebreak (Unix/Windows)	ASCII beyond latin alphat
		New alphabet block
Selected bl	locks and resulting alphabet:	
äöüß	ĀÕÜ A-Z a-z	
(Right-click on	the selected blocks to reorder and (slightly) transform them)	
Result (com	posed alphabet):	
Alphabet c	haracters: aöüßÄÖÜABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz	
Alababab	ength: 59	
Alphabet I		
Alphabet i	Manuel input	
	Manuel input Create custom alphabets	
Alphabet I	Manuel input Create custom alphabets and the characters for the new alphabet	
Alphabet II Iphabet Enter a name	Manuel input Create custom alphabets and the characters for the new alphabet ume for the alphabet:	
Alphabet II Alphabet Enter a name 1) Enter a na A_Z, a-z, 0-	Manuel input Create custom alphabets and the characters for the new alphabet ume for the alphabet: 9, Line Breaks	
Alphabet II Alphabet Enter a name (A_Z, a-z, 0 2) Define the	Manuel input Create custom alphabets and the characters for the new alphabet ume for the alphabet: 9, Line Breaks e alphabet characters:	
Appnabet in Appnabet Enter a name 1) Enter a name 2) Define the Compose	Manuel input Create custom alphabets and the characters for the new alphabet une for the alphabet: 9, Line Breaks] e alphabet characters: e alphabet characters from 'alphabet blocks'	
Appradet in Appradet in Apprabet Enter a name 1) Enter a name A Z, a-z, 0- 2) Define the Compose \odot Enter alp	Manuel input Create custom alphabets and the characters for the new alphabet ame for the alphabet: 9, Line Breaks e alphabet characters: e alphabet characters from 'alphabet blocks' whate characters manually	
Appraded in Appraded in Apprabet Enter a name 1) Enter a name 2) Define the Composi © Enter alp ABCDEF(1)	Manuel input Create custom alphabets and the characters for the new alphabet and the characters for the new alphabet me for the alphabet: 9. Line Breaks] e alphabet characters from 'alphabet blocks' whabet characters manually 30(10)	

Building blocks compilation

Variable alphabets for classic algorithms

The implementation in JCT 2/2

Further hints

- Custom defined alphabets can be stored permanently.
- In a JCT session predefined alphabets can also be reused without storing them for permanent usage.

reate a	custom alphabet
Here you	r can build your own alphabet, and determine whether your alphabet should be permanently saved of not.
Perman	ience of the newly created alphabet
Your ne	wly created alphabet can be used only once, or it can be stored into the standard set of JCT alphabets.
But ever before r	i if you don't make your alphabet permanent, you can still reuse it in this JCrypTool session. All custom alphabets you have create estarting the JCrypTool will appear in this screen.
🗹 Make	e the newly created alphabet permanent
Reuse	custom alphabets
🗌 Use	a custom alphabet from the past
	Name: A_Z, a-z, 0-9, Line Breaks
	Content: ABCDEF\r\n

- The saved alphabets can be managed and later edited in the global settings in JCT:
 - Windows + Linux: "Window" \ "Preferences/Settings"
 - MacOS: "JCrypTool" \ "Preferences"

Variable alphabets for classic algorithms Educational objective

Conclusion

- New alphabets can be created quickly in JCT using existing building blocks.
- As special characters can be included as well, there are no limits for the usage of alphabets.
- As it is easy to understand, and efficient to build a custom alphabet, a user is motivated to try out.
- Most of the common crypto tools use fixed sets of alphabets for classic ciphers. At this juncture, JCT is maximally flexible.
JCrypTool console for classic methods

The implementation in JCT

The console

The classic cryptographic methods can be started from the console as well.

```
Console X
                                                                                                       🕞 🕂 📑 🛃 🔄 ד 📬 ד 🗖 🗖
JCrypTool crypto console
Welcome to the JCrypTool console.
Enter 'help' or 'help <command>' for assistance.
JCrypTool=> help
In the JCrypTool console, you can use cryptographic algorithms from the command line.
Enter 'help -l' to view the list of the console commands (in alphabetical order).
The help command helps you with the syntax of commands and can give you examples of calling commands:
SYNTAX HELP:
For every command, there is a short, and a detailed syntax help. There are many possibilities to reach them:
                        Detailed:
Short:
'help <command>'
                        'HELP <command>'
                        '?? <command>'
'? <command>'
                        '<command> HELP'
'<command> help'
'<command> ?'
                        '<command> ??'
EXAMPLES:
Many commands offer examples of their usage, which can be viewed by entering 'help -x <command>'.
For example, try viewing the examples of the help command now ('help -x help')!
JCrypTool=>
```

- To receive some additional information about the console, simply type the command "help".
- There are help and example pages for each single method.

JCrypTool console for classic methods Application sample 1/2

Example Autokey Vigenère

- From the console, all classic cryptographic methods can be invoked on the current editor's content, a file on the disk or text as an argument in the console.
- The console can be called via the icon bar (below the main menu) via the following icon:



- Example with the Autokey-Vigenère method:
 - Invoke help and examples:

```
JCrypTool=> help autovigenere
The Vigenère cipher, but the key is generated partly from the plaintext.
Syntax:autovigenere [-a <ALPHABET>] -D | -E -ed | -f <FILE_PATH> | -t <TEXT> -k <KEY> [-noFi]
Examples for this command are available under 'help -x autovigenere'.
For a more detailed help, enter 'HELP autovigenere'.
More information for this algorithm is available in the JCrypTool online help.
JCrypTool=> help -x autovigenere
'autovigenere -E -ed -k akey' -> Encrypts the active editor's text with the key "akey"
'autovigenere -D -ed -k akey' -> Decrypts the active editor's text with the key "akey"
'autovigenere -E -a A-Z -t "TEST TEXT" -k AKEY' -> Encrypts the text "TEST TEXT" with the key "AKEY", using only the uppercase alphabet
```

- The upper screen shot shows the command line options described in the console help for a special method (here using the example "HELP autovigenere").

JCrypTool console for classic methods Application sample 2/2

Encryption and decryption with Autokey Vigenère

As sample plaintext we use "ACTIONxCODExDAYYTT ", and as key we use "THEKEY":

```
JCrypTool=> autovigenere -E -a a-zA-Z -t "ACTIONxCODExDAYYTT" -k THEKEY
TJXSSlxEhLSKACmbXQ
```

- The 2nd row shows the generated ciphertext "TJXSSIxEhLSKACmbXQ", generated by the command "autovigenere -E -a a-zA-Z -t "ACTIONXCODEXDAYYTT" -k THEKEY"
- By substituting "-E" with "-D" in the command we can simply revoke the encryption: "autovigenere -D -a a-zA-Z -t "TJXSS1xEhLSKACmbXQ" -k THEKEY"

JCrypTool=> autovigenere -D -a a-zA-Z -t "TJXSSlxEhLSKACmbXQ" -k THEKEY ACTIONxCODExDAYYTT

JCrypTool console for classic methods Educational objective 1/2

Advantages using the console

- The parameters of an operation (such as the alphabet and the key) can be easily inserted and reused via Copy&Paste.
- The more parameter one uses, the more efficient the usage of the console is. There can be much more parameters than the alphabet, the key and the filtering of non-alphabet characters.
- For instance, the transposition encryption method uses a lot of parameters:
 - Each of the following parameters can be configured for the 1st and 2nd round (at all 6 parameters):
 - Direction of read in
 - Direction of read-out
 - Key
 - Alphabet
 - Filtering of characters not in the alphabet
- Once entered in the dialog window the command line contains all parameters.
- The command line can be copied, pasted and easily modified.

⊖ O O Tra	ansposition				
Transposition Please enter transposition key(s), and in-/out-reading modes.					
Operation					
Encrypt	ODecrypt				
Alphabet					
Plain-/Ciphertext alphabet: Anzeigbares ASCII Show alphabet: Custom alphabet Image: Select alphabet characters from the input text before the en-/decryption.					
Transposition(s)					
First transposition 1) Read the text into the transposition table Column-wise Row-wise 2) Transposition – enter the key	 Second (optional) transposition 1) Read the text into the transposition table Column-wise Row-wise 				
C4D3 3 2 4 1	2) Transposition – enter the key 3143 2 1 4 3				
3) Read the ciphertext out of the transposition table Column-wise Row-wise	3) Read the ciphertext out of the transposition table Oclumn-wise ORow-wise				

Appropriate command for the console:

transposition	I -Eeditor -	a "Printable	ASCII"key	CAD4
-t1ReadIn rw	-t1ReadOut cw	key2 RT334	-t2ReadIn rw	-t2ReadOut cw

JCrypTool console for classic methods Educational objective 2/2

Detailed help from the console

Help on the console for the transposition method

```
JCrypTool=> HELP transposition
Transposes characters of the plain (columnar transposition with definable read-in / read-out directions).
Syntax:transposition [-a <ALPHABET>] -D | -E -ed | -f <FILE_PATH> | -t <TEXT> -k <KEY> [-k2 <KEY>] [-noFi] [-t1ReadIn
       <ORDER = 'cw'/'rw'>] [-t1ReadOut <ORDER = 'cw'/'rw'>] [-t2ReadIn <ORDER = 'cw'/'rw'>] [-t2ReadOut <ORDER =</pre>
       'cw'/'rw'>]
Option explanation:
    -a,--currentAlphabet <ALPHABET>
                                                                        One of ASCII, a-zA-Z, A-Z, a-z, Playfair, ADFGVX,
                                                                        Xor32, Xor64, default: ASCII
    -D,--modeDecrypt
                                                                        Decryption
    -E,--modeEncrypt
                                                                        Encryption (Default, if neither en- nor
                                                                        decryption is specified)
    -ed.--editor
                                                                        Use active Editor as Input
    -f,--inputFile <FILE_PATH>
                                                                        File is input
    -k,--key <KEY>
                                                                        Key (only characters from the selected alphabet
                                                                        are allowed)
    -k2,--key2 <KEY>
                                                                        Optional second transposition key, which
                                                                        signalizes that a double columnar transposition
                                                                        has to be executed.
    -noFi,--noFilter
                                                                        Non-alphabetic characters will not be filtered
    -t,--inputText <TEXT>
                                                                       Text as input (as string between "")
    -t1ReadIn,--transposition1ReadInOrder <ORDER = 'cw'/'rw'>
                                                                       ORDER = 'cw' (column by column) / 'rw' (row by
                                                                        row). Read-in direction of plaintext into
                                                                        transposition table (if not defined.row-wise).
                                                                        (applies for the 1st transposition)
    -t1ReadOut,--transposition1ReadOutOrder <ORDER = 'cw'/'rw'>
                                                                        see argument 't1ReadIn's description (if not
                                                                        defined.column-wise).
    -t2ReadIn,--transposition2ReadInOrder <ORDER = 'cw'/'rw'>
                                                                        see argument 't1ReadIn's description (if not
                                                                        defined, row-wise).
    -t2ReadOut,--transposition2ReadOutOrder <ORDER = 'cw'/'rw'>
                                                                        see argument 't1ReadIn's description (if not
                                                                        defined.column-wise).
Examples for this command are available under 'help -x transposition'.
```

```
Aliases for this command are 'transp'.
More information for this algorithm is available in the JCrypTool online help.
```

The implementation in JCT

JCT perspectives

 JCT supports two main user interfaces: the Default Perspective and the Algorithm Perspective.



The **Algorithm Perspective** is separated – next to the editor and the help – in the following 3 windows:

- 1. Keystore
- Allows to save keys and key pairs for later usage.
- 2. Algorithms
- An explorer for algorithms. The algorithms are provided by the crypto libraries FlexiProvider^[1] and BouncyCastle^[2]. In contrast to the Crypto Explorer in the Default Perspective, many different variants of the algorithms are directly listed and selectable. Altogether, the Algorithm Explorer is much more extensive than the Crypto Explorer.

3. Operations

- The algorithm, chosen via double-click in the Algorithm Explorer, is listed here. Then additional settings (e.g. the source of the input, the target for the output, the key and the algorithm's parameter) are outlined here.

[1] http://www.flexiprovider.de



^[2] http://www.bouncycastle.org

The introduction plugin

Algorithm Perspective explained

 On opening the Algorithm Perspective for the first time, a slide show appears in the editor area which explains the basic functionality of that perspective.



File Edit Window Help 월 ▼ 🖉 및 및 🙆 💱 🔃 Q / 선 ▼ 🕅 ▼ 💔 <? 🖹 🔛 Default 🔂 Algorithm - -[Keystore 🖾 > <</p> 🚯 Instruction Algorithm Perspective 🖾 🗟 unsaved001.txt 🗟 Algorithms 🖾 JCrypTool Keystore CrypTool 1.0.0 File Edit Window Help Block Ciphers Alice Whitehat 0 · 0 10 0 0 10 0 0 0 + 0 + 10 🕾 🔛 Default 🔛 Algorithm an ter et = = 0 12 unsaved001.txt = Keystore @ Algorithms Bob Whitehat Hybrid Ciphers JCrypTool Keystone Tutorial for the Asymmetric Block Ciphers Alice Whitehat Block Ciphers [PW: 1234] Alice Whitehead S Certificates (Public Keys) Camella. **Algorithm Perspective** III DESede Key Pairs IDEA (OID: 1.3.6.1.4.1.188,7.1.1) Message Digests - Secreti Keyr MARS. Misty1 using RC6 encryption as an Password-Based Ciphers TRC2 TORCS. Pseudo Random Number Generators example. CI RC6 III Rijndael Signatures SAFER+ ► in id ** Operations SAFER++ Current Entry: «None» Serpent (OID: 1.3.6.1.4.0.11591.13.2) TS Shacal Block Cipher Shacal2 77 Twofish A block cipher divides a message into blocks of a fixed size, e.g. 16 bytes, and then encrypts Hybrid Ciphers each message block individual with the same Message Authentication Code Message Digests D Password-Based Ciphers For decryption the cipher text is again divided D Pseudo Random Number Generator into message block and the decryption metho 0000000000 Signatures is applied to each block individual 🕅 Help 🖾 Contents & Search 📽 Related Topics Bookmarks 🗮 Index Do not show again How to use the Algorithm Perspective ڬ Operations 🖾 ≥ 22 ° 0 This quick introduction explains how to use the Algorithm Perspective. Like in the Current Entry: <None> slideshow, we use the RC6 block cipher as an example. The procedure for other algorithms is similar, so this guide can also be used for

Application sample 1/3: select and customize the AES operation

In this example we encrypt a text from the opened editor with AES and export the result to a file.

Generate a new key and assign it to a contact

- First, we generate a key to be used for encryption.
- With a new symmetric key can be created. As AES is a symmetric crypto system, it does only need a single secret key (instead of a key pair). Alternatively, for asymmetric crypto systems the appropriate key pairs can be generated with
 This is step 1 on slide <u>83</u>.
- In the wizard "New Symmetric Key" we choose "AES, Rijndael (OID 2.16.840.1.101.3.4.1)"^{[1],} select or create our new contact by changing the contacts name and set an arbitrary password.
- Then, the key is stored in the JCT keystore, listed below the chosen contact name (in the example "Max").



[1] OID: Object Identification, a unique identifier for an algorithm Defined by ITU (http://en.wikipedia.org/wiki/Object_identifier).

Application sample 2/3: select and customize the AES operation

Selection of the algorithm for the current operation

- Now, it's time to choose the appropriate algorithm: In the tab "Algorithm" below "Block Ciphers" you find the the AES-Rijndael algorithm.
- Select the algorithm with a **double click** (step 2 on slide <u>83</u>).
- A wizard appears where padding and mode^[1] of the block cipher can be adjusted.
 Additionally, further algorithm specific settings can be adjusted here (e.g. for AES the length of each block in bits).
- Like in the Crypto Explorer, the algorithms in this explorer are also grouped by the kind of the cryptographic method.
- Note about the usage:

Step 2 (selection of the algorithm) can also be performed BEFORE step 1 (generation of a key for its owner in the keystore). So the order of step 1 and step 2 is independent.

Especially if the contact (owner) already has a key for the chosen algorithm, you can directly start with step 2.



[1] The mode of a block cipher is responsible for the mapping of the plain text to the blocks, which will be eventually encrypted. If in the last block some bits are missing, the padding rules how these bits will be filled.

Application sample 3/3: select and customize the AES operation

Settings for input and output of the current operation

- Via the double click (in the Algorithms tab on the right) Rijndael was added to the Operations tab.
- Via drag'n'drop the generated key (from the JCT keystore on the left) can now be copied on the key field of the algorithm (see slide <u>83</u>, step <u>3</u>).
- The option Input/Output offers to select via double click the source and target for the algorithm.
 E.g., performing a double click on Input, you can switch text input from a file or from an active editor window (see slide <u>83</u>, step <u>4</u>).
- To toggle between encryption or decryption, you can adjust the option "Operation".
- After setting all the parameters you can start the operation by clicking the green arrow within the title of the Operations window.



 With asymmetric algorithms, the kind of operation (encrypt/decrypt, sign/verify) is dictated by the type of key. A public key encrypts or verifies, a private key decrypts or signs.

Overview of the 4 steps to select and customize an operation



Result after executing the operation (here both, input and output, are in the JCT editor)

JCrypTool	
<u>F</u> ile <u>E</u> dit <u>W</u> indow Help	
🗟 ▾ 🚑 🔚 🔞 🙆 🔞	😭 🔙 Default 🌇 Algorithm
📔 Keystore 🛛 🕒 🗢 🕿 💐 🏱 🗖 🗖	😰 unsaved001.txt 📓 out001.bin 🕱 🔤 🗖 🖉 🖓 🖓 🖓 🖓
JCrypTool Keystore	Offset 0 1 2 8 4 5 6 7 8 9 A B C D E F A
🔏 A. Prism	000001 59 30 61 85 87 D7 42 91 89 C5 38 41 70 2D 1D CD
🙀 Certificates (Public Keys)	00000 49 7F 42 27 72 CA 08 86 DF 89 D8 AD 3D A5 53 8F
📚 Key Pairs	00000 0 02 H4 D5 H5 77 35 27 70 47 D9 F1 B9 43 18 73 47
🗭 Secret Keys	0000000000000000000000000000000000000
🔒 Alice Whitehat	00000 96 E5 E0 /6 FF F5 F2 5/ 0C /D F6 50 1/ D4 96 F9
🔀 Certificates (Public Keys)	00000 B6 AF 44 60 78 46 B3 95 88 E3 94 9B B9 9E 0F 8F
🎨 Key Pairs	■ RSA_PKCS1_v2_1 (OID: 1.2.840.113549.1
🕞 Secret Keys	Offset: 00000000 h of 0000055Eh (03 Value: 0x59 (hev) = 89 (dec) = 0131 (oct) = 01011001 (hij: Filesize: 1.632 hut
AES, Rijndael (OID: 2.16.840.1.101.3.4.1) (key strength: 128)	Camellia
HmacMD5 (OID: 1.3.6.1.5.5.8.1.1) (key strength: 64)	Dependions & DESede
IDEA (OID: 1.3.6.1.4.1.188.7.1.1) (key strength: 128)	MIDEA (OID: 13.6.1.4.1.188.71.1)
₽ RC6 (key strength: 128)	Current Entry: (Rjindael) no name Created on: Wed Apr 16 15:11:45 CEST 2014
Twofish (key strength: 128)	a 💟 (Rijndael) no name Created on: Wed Apr 16 15:11:45 CEST 2014
🔏 Bob Whitehat	Algorithm: Rijndael
🔏 Erika Mustermann	Mode:\CBC
JCT-PKI Root Certificates - DO NOT DELETE	Belding: PKCSPadding
🔏 el Mustermann	A 🖸 Input/Output
1 Help 🕅 ↔ 🗢 🗖 🗖	Output: <editor> SAFER+ Second Alice Whitehet AES Bijedeel (OID: 2.16 840.1.101.2.4.1) (///// traneth: 1.28)</editor>
Contents 🌮 Search 📽 Related Topics 💷 Bookmarks	Operation: Encount
E Index	Shacal
index .	Shacal2
Operations	/ 💽 Twofish
After selecting an algorithm in the Algorithms view, the	Contraction Contra
corresponding entry will appear in the Operations view. This view	/ 🗎 Message Authentication Codes
allows you to enter the input source (either the text editor or a file) and	/ 🛅 Message Digests
the output destination (a file). Via drag and drop from the Keystore	C Password-Based Ciphers
a matching key before executing an algorithm.	🛅 Pseudo Random Number Generators
	📄 Signatures
To execute an algorithm, after all parameters are set correctly, click on	
the execute icon in the upper right corner of the Operations view.	
Mara reculta	

Further functions in JCrypTool

Further samples what is in JCrypTool

- Tri-partite key agreements (MPKE)
- Visualization of the inner states of DES
- Visualization of calculations on elliptic curves over real and discrete fields
- ElGamal Cryptosystem (for encryption and signing)
- Visualization of the simple power analysis attack against RSA (SPA)
- Quick solver of the number shark game with heuristic methods; solving of Sudoku variants
- Mathematical games: number shark, divider game, zero-knowledge Sudoku (Zudo-Ku)
- Entropy analysis
- Dynamic visualization of Huffman coding trees
- Signature demonstration, signature and certificate verification (shows effect of validity models)
- Visualization of the SSL/TLS handshake (protocol)
- Implementation and visualization of ARC4 and Spritz
- Visualization of post-quantum signature algorithms [SPHINCS+, MerkleTree XMSS-MT, WOTS, McEliece (error-correcting code), multivariate cryptography (rainbow signature scheme)]
- Fast cryptanalysis of the grille cipher

•

Overview about all functions in JCrypTool

Visible within JCT. Alternatives: Listed within the CrypTool Portal or with the JCT admin tool



https://www.cryptool.org/en/ documentation/functionvolu me

Legend:

[A] in Algorithm Perspective

[D] in Default Perspective

Introduction to the e-learning software JCrypTool

Applications within JCT – a selection

How to participate

87

Ζ

22

How to participate – Overview

JCrypTool – Request for participation	Page 89
Participation in JCrypTool	Page 90
Contacts	Page 92

JCrypTool – Request for participation

Arms are wide open for your participation

- Feedback, critique, helpful suggestions and ideas
- Implementation of more algorithms, protocols or techniques for analysis
- Help to ensure consistence and completeness
- Participation in the development (programming, layouting, translation, tests, website development)
 - in the "old" C/C++ project CrypTool 1 and
 - in the new projects (preferred):
 - C# project: "CrypTool 2" = CT2 (<u>https://www.cryptool.org/de/ct2/volunteer</u>)
 - Java project: "JCrypTool" = JCT (<u>https://www.cryptool.org/en/jct/volunteer</u>)
 - Browser project: "CrypTool-Online" = CTO (<u>http://www.cryptool-online.org</u>)
- Especially faculties/chairs who use JCrypTool for educational purposes, are invited to join the development.
- Significant contribution can be mentioned (in the online help, in about dialogs, and on the website).

Participation in JCrypTool

Example ideas for more visuals

- Visualization of the interoperability between S/MIME and OpenPGP formats
- Demonstration of visual cryptography
- Protocol validator
- Cryptanalysis of further algorithms
- Visualization of different methods from post-quantum cryptography
- Visualization of current developments like indistinguishability obfuscation

Further things of high interest

- One place for all the manipulations of frequency tables (creation, exchange, deepness) and of permutations
- Key Storage
- JavaFX support

Open tasks are also mentioned on the developer sites:

JCrypTool: <u>https://github.com/jcryptool/core/wiki/project-Ideas</u>

Participation in JCrypTool

More information for developers

- Wiki: <u>https://github.com/jcryptool/core/wiki</u>
- Style-Guide: <u>https://github.com/jcryptool/doc/raw/master/Guidelines/JCrypTool-GUI-Guidelines.pdf</u>
- Tutorial: <u>https://github.com/jcryptool/core/wiki/Getting-started-as-a-JCrypTool-Developer</u>
- Information for developing plugins is provided in the JCT wiki. The wiki in the internet offers links and information for JCT core developers and crypto plugin developers.
- Plugin developers should not need any projects from the JCT repository. They just need to run JCT as a target platform and develop for it.



Contacts

Prof. Bernhard Esslinger	Simon Leischnig	Thorben Groos
Overall CT lead University of Siegen	JCT project lead	JCT project co-lead
<u>bernhard.esslinger@uni-siegen.de</u> <u>bernhard.esslinger@gmail.com</u>	<u>simonjena@gmail.com</u>	<u>thorben.groos@web.de</u>

Dominik Schadow: former project lead, info@xml-sicherheit.de

www.cryptool.org