CRYPTOLOGY WITH CRYPTOOL v 1.4.30 Beta04

Practical Introduction to Cryptography and Cryptanalysis

Scope, Technology and Future of CrypTool

www.cryptool.org
www.cryptool.com
www.cryptool.de
www.cryptool.es
www.cryptool.pl

Prof. Bernhard Esslinger and CrypTool-Team, June 2009

Content (I)

Content (II)

III. Examples

10. RSA attack using lattice reduction
11. Random analysis with 3-D visualization
12. Secret Sharing using the Chinese Remainder Theorem (CRT) and Shamir
13. Implementation of CRT in astronomy (solving linear modular equation systems)
14. Visualization of symmetric encryption methods using ANIMAL
15. Visualizations of AES
16. Visualization of Enigma encryption
17. Secure E-Mail with S/MIME
18. Generation of a message authentication code (MAC)
19. Hash demonstration
20. Learning tool for number theory and asymmetric encryption
21. Point addition on elliptic curves
22. Password quality mete (PQM) and password entropy
23. Brute-force analysis
24. CrypTool online help

IV. Project / Outlook / Contact

Content

Definition Cryptology and Cryptography

Cryptology *(from the Greek kryptós, "hidden," and lógos, "word") is the science of secure (generally secret) communications. This security obtains from legitimate users, the transmitter and the receiver, being able to transform information into a cipher by virtue of a key -- i.e., a piece of information known only to them. Although the cipher is inscrutable and often unforgeable to anyone without this secret key, the authorized receiver can either decrypt the cipher to recover the hidden information or verify that it was sent in all likelihood by someone possessing the key.*

Cryptography *was concerned initially with providing secrecy for written messages. Its principles apply equally well, however, to securing data flow between computers or to encrypting television signals. ... Today the modern (mathematical) science of cryptology contains not only mechanisms for encryption but also for integrity, electronic signatures, random numbers, secure key exchange, secure containers, electronic voting and electronic money, and has achieved to render a broad range of applications in modern life.*

Source: Britannica (www.britannica.com)

A similar definition can be found on Wikipedia: http://en.wikipedia.org/wiki/Cryptology

Relevance of Cryptography

Examples for Cryptography Usage

- Phone cards, cell phones, remote controls
- Cash machines, money transfer between banks
- Electronic cash, online banking, secure eMail
- Satellite TV, Pay TV
- Immobiliser systems in cars
- Digital Rights Management (DRM)

- Cryptography is no longer limited to agents, diplomats or the military. Cryptography is a modern, mathematically characterised science.
- Breakthrough for cryptography started with the broad use of the Internet
- For companies and governments it is important that systems are secure and

... users (clients, employees) have a certain understanding and awareness for IT security!

Cryptography – Objectives

- Confidentiality
  - Information can practically not be made available or disclosed to unauthorized individuals, entities or processes.
- Authentication
  - Authentication ensures that users are identified and those identities are appropriately verified.
- Integrity
  - Integrity ensures that data has not been altered or destroyed in an unauthorized manner.
- Non-Repudiation
  - The principle that, afterwards, it can be proven that the participants of a transaction did really authorize the transaction and that they have no means to deny their participation.

The CrypTool Project

- Origin in awareness program of a bank (in-firm training)
  → Awareness for employees
- Developed in co-operation with universities (improving education)
  → Media didactic approach and standard oriented

  1998 Project start – effort more than 17 man-years since then
  2000 CrypTool available as freeware
  2002 CrypTool on Citizen-CD-ROM from BSI (German Information Security Agency)
  2003 CrypTool becomes Open-Source – Hosting by University of Darmstadt (Prof. Eckert)
  2007 CrypTool available in German, English, Polish und Spanish
  2008 .NET version and Java version – Hosted by University of Duisburg (Prof. Weis) and SourceForge
- Awards

  2004 TeleTrusT (TTT Förderpreis)
  2004 NRW (IT Security Award NRW)
  2004 RSA Europe (Finalist of European Information Security Award 2004)
  2008 "Selected Landmark" in initiative "Germany – Land of Ideas"

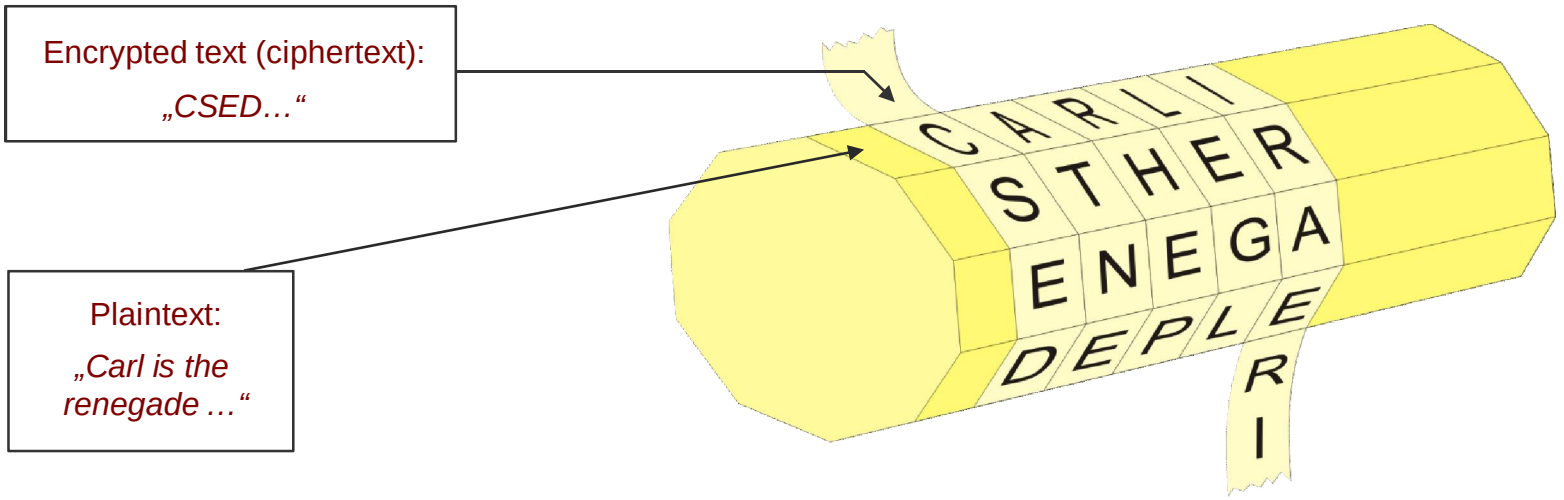
  

- Developers
  – Developed by people from companies and universities in different countries
  → Additional project members or usable sources are always appreciated (currently there are around 40 people working on CrypTool world wide).

Examples of Early Cryptography (1)

Ancient encryption methods

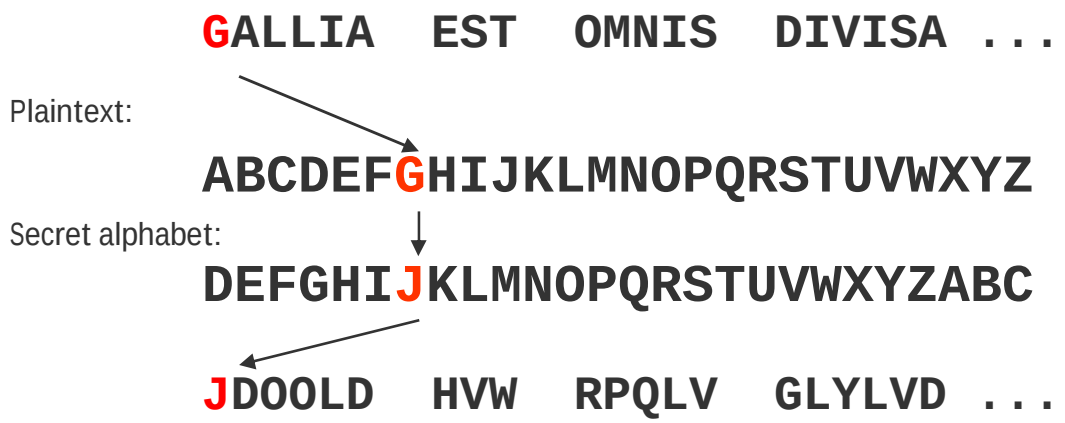
- Tattoo on a slave's head concealed by re-grown hair
- Atbash (around 600 B.C.)
  - Hebrew secret language, reversed alphabet
- Scytale from Sparta (500 B.C.)
  - Described by Greek historian/author Plutarch (45 - 125 B.C.)
  - Two cylinders (wooden rod) with identical diameter
  - Transposition (plaintext characters are re-sorted)

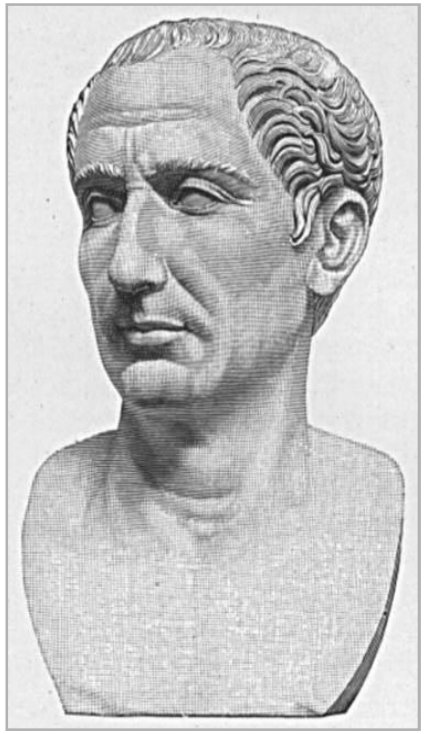


Examples of Early Cryptography (2)

Symmetric Caesar encryption

- Caesar encryption (Julius Caesar, 100 - 44 B.C.)
- Simple substitution cipher





- Attack: Frequency analysis (typical character allocation)

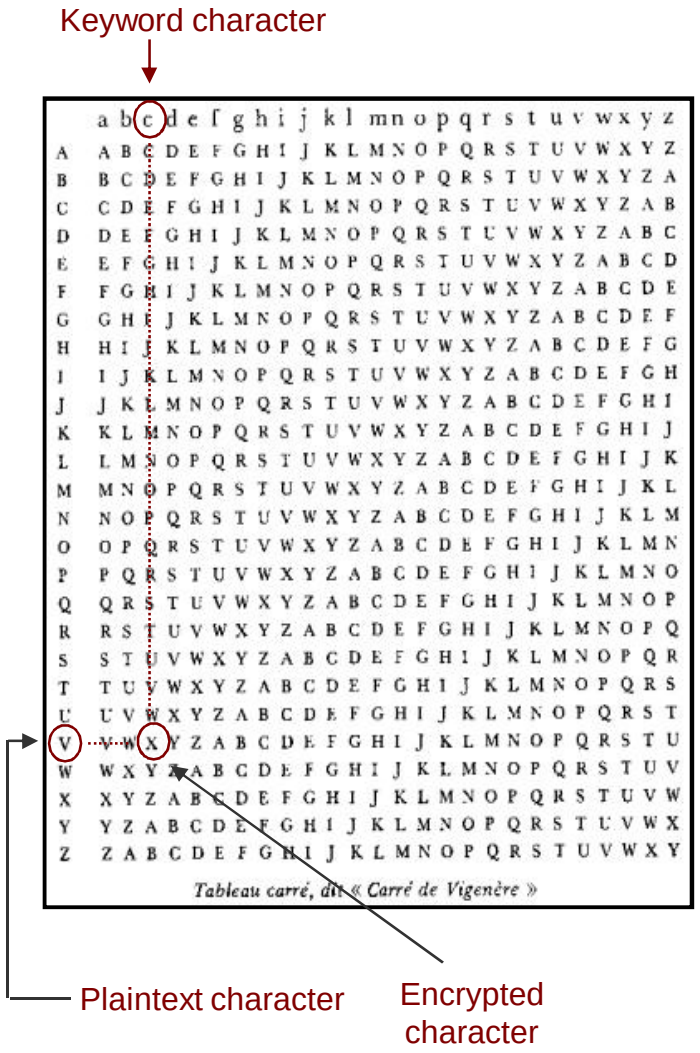
Presentation with CrypTool via the following menus:

- Animation: „Indiv. Procedures“ \ „Visualization of algorithms“ \ „Caesar“
- Implementation: „Crypt/Decrypt“ \ „Symmetric (classic)“ \ „Caesar / Rot-13“

Examples of Early Cryptography (3)

Symmetric Vigenère encryption

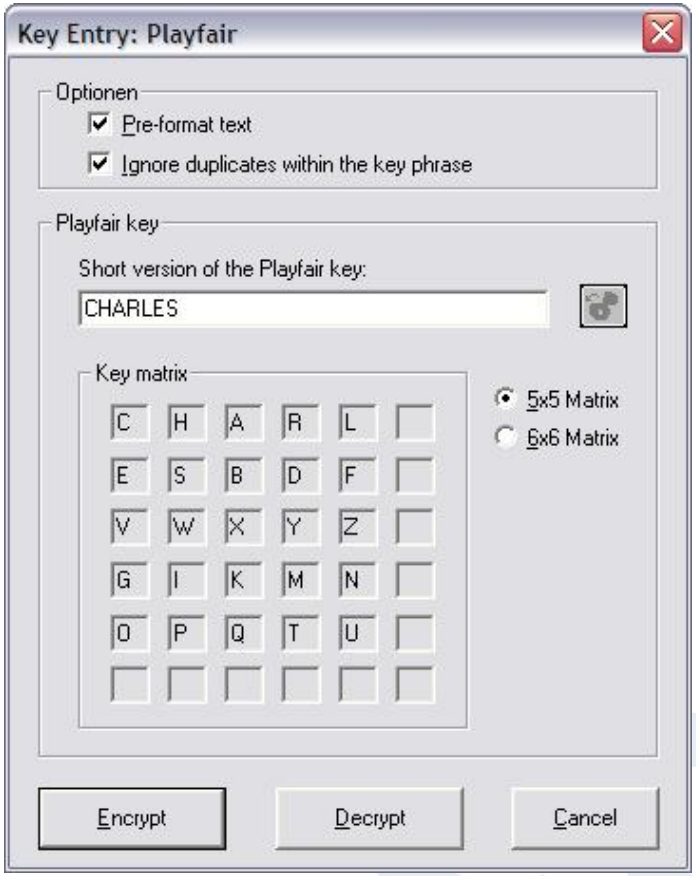
- Vigenère-Encryption (Blaise de Vigenère, 1523-1596)
- Encryption with a key word using a key table
- Example:
  Keyword: **CHIFFRE**
  Encrypting: **VIGENERE** becomes **XPOJSVVG**
- The plaintext character (V) is replaced by the character in the corresponding row and in the column of the first key word character (c). The next plaintext character (I) is replaced by the character in the corresponding row and in the column of the next key word character (h), and so on.
- If all characters of the key word have been used, then the next key word character is the first key character.
- Attack (via Kasiski test): Plaintext combinations with an identical cipher text combination can occur. The distance of these patterns can be used to determine the length of the keyword. An additional frequency analysis can then be used to determine the key.



Examples of Early Cryptography (4)

Other symmetric encryption methods

- Homophone Substitution
- Playfair (invented 1854 by Sir Charles Wheatstone, 1802-1875)
  - Published by Baron Lyon Playfair
  - Substitution of one character pair by another one based on a square-based alphabet array
- Transfer of book pages
  - Adaptation of the One-Time Pad (OTP)
- Turning grille (Fleissner)
- Permutation encryption
  - „Double Dice“ (double column transposition) (Transposition / very effective)



Cryptography in Modern Times

Cryptography developments in the last 100 years till 1970

Classic methods

- are still in use today.
  (since, not everything can be done by a computer…)
- and their principals of transposition and substitution are inputs for the design of modern algorithms: combination of simple operation (a type of multiple encryption, a so called cascades of ciphers), on bit level, block cipher, rounds.

Encryption becomes

- more sophisticated,
- mechanised or computerised and
- remains symmetric.

Examples of the First Half of the 20th Century

Mechanical encryption machines (rotor machines)

Enigma Encryption (Arthur Scherbius, 1878-1929)

- More than 200000 machines have been used in WW2
- The rotating cylinder set causes, that every character of the text becomes encrypted with a new permutation.
- Broken by massive effort of cryptography experts (around 7000 persons in UK) with decryption machines, captured original Enigmas and by intercepting daily status reports (e.g. weather reports).
- Consequences of this successful crypto analysis:
  "In general the successful crypto analysis of the engima encryption has been a strategic advantage, that has played a significant role in winning the war. Some historians assume that the break of the enigma code has shortened the war by several months or even a year."

  (translated from http://de.wikipedia.org/wiki/Enigma_%28Machine%29 - March 6, 2006)

Cryptography – Important Insights (1)

- Kerckhoffs principle (1883)
  - Separation of algorithm (method) and key
    e.g. Caesar encryption:
    Algorithm: "Shift alphabet by a certain number of positions to the left"
    Key: The "certain number of positions" (Caesar for example)
  - Kerckhoffs principle:
    The secret lies within the key and not within the algorithm or „No security through obscurity"
- One-Time Pad – Shannon / Vernam
  - Demonstrably theoretically secure, but not usable in reality (only red phone)
- Shannons concepts: Confusion and Diffusion
  - Relation between M, C and K has to be as complex as possible (M=message, C=cipher, K=key)
  - Every cipher text character should depend on as many plaintext characters and as many character of encryption key
  - „Avalanche effect"(small modification, big impact)
- Trapdoor function (one-way function)
  - Fast in one direction, not in the opposite direction (without secret information)
  - Having the secret the opposite direction works (access to the trapdoor)

Examples for a Breach of the Kerckhoffs Principle

Secret lies within the key and not within the algorithm

- Cell phone encryption penetrated (December 1999)

  „ *Israeli researchers discovered design flaws that allow the descrambling of supposedly private conversations carried by hundreds of millions of wireless phones. Alex Biryukov and Adi Shamir describe in a paper to be published this week how a PC with 128 MB RAM and large hard drives can penetrate the security of a phone call or data transmission in less than one second. The flawed algorithm appears in digital GSM phones made by companies such as Motorola, Ericsson, and Siemens, and used by well over 100 million customers in Europe and the United States." […]*

  "Previously the GSM encryption algorithms have come under fire for being developed in secret away from public scrutiny -- but most experts say high security can only come from published code. Moran said "it wasn't the attitude at the time to publish algorithms" when the A5 ciphers was developed in 1989, but current ones being created will be published for peer review."

  [http://wired.lycos.com/news/politics/0,1283,32900,00.html]

Sample of a One-Time Pad Adaptation

Clothes hanger of a Stasi agent
with a secret one-time pad
(taken from: *Spiegel Spezial 1/1990)*

Key Distribution Problem

Key distribution for symmetric encryption methods

If 2 persons communicate with each other using symmetric encryption, they need one common secret key.

If n persons communicate with each other, then they need $S_n = n * (n-1) / 2$ keys.

That is:

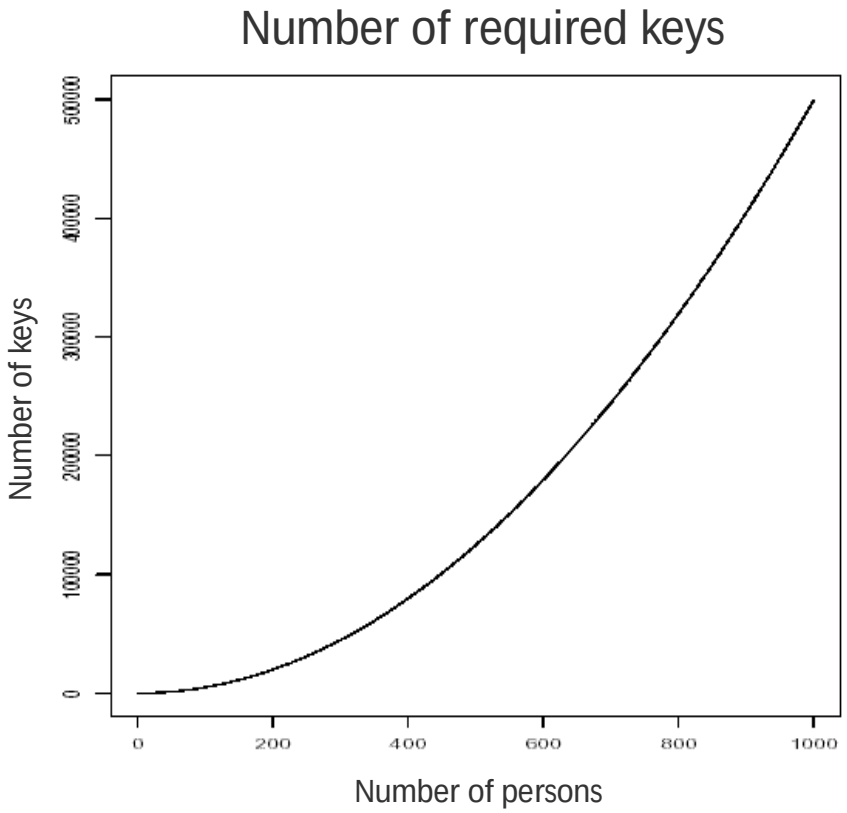
n = 100 persons require

S_{100} = 4.950 keys; and

n = 1.000 persons require

S_{1000} = 499.500 keys.

⇨ factor 10 more persons mean factor 100 more keys



Cryptography – Important Insights (2)

Solving the key distribution problem through asymmetric cryptography

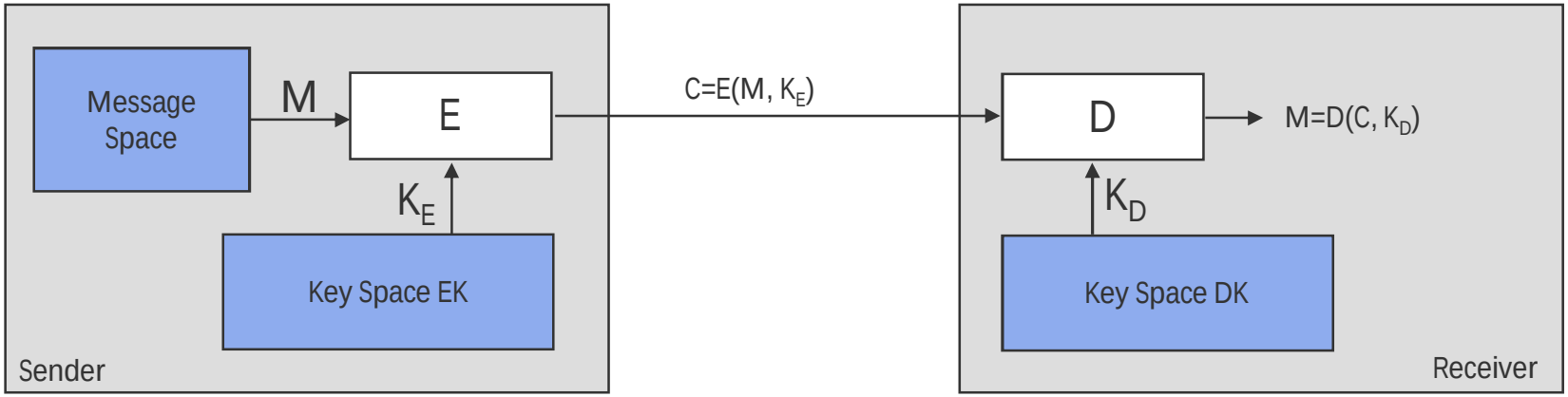
Asymmetric cryptography

- For centuries it was believed that: Sender and receiver need same secret.
- New: Every member needs a key pair (Solution of the key distribution problem)
- Asymmetric encryption
  - „Everyone can lock a padlock or can drop a letter in a mail box.“
  - MIT, 1977: Leonard Adleman, Ron Rivest, Adi Shamir (well known as RSA)
  - GCHQ Cheltenham, 1973: James Ellis, Clifford Cocks (admitted in public December 1997)
- Key distribution
  - Stanford, 1976: Whitfield Diffie, Martin Hellman, Ralph Merkle (Diffie-Hellman key exchange)
  - GCHQ Cheltenham, 1975: Malcolm Williamson

Security in open networks (such as the internet) would be extremely expensive and complex without asymmetric cryptography!

Encryption and Decryption

Symmetric und asymmetric encryption



a) Symmetric Encryption: $K_E = K_D$ (e.g. AES)

b) Asymmetric Encryption: $K_E \neq K_D$ (e.g. RSA)

secret

public

private/secret

Cryptography – Important Insights (3)

Increasing relevance of mathematics and information technology

- Modern cryptography is based on mathematics
  - Still new symmetric encryption methods such as AES (better performance and shorter key length compared to the asymmetric methods purely based on mathematical problems).
- The security of encryption methods heavily depends on the current status of mathematics and information technology (IT)
  - Computation complexity (meaning processing effort in relation to key length, storage demand and data complexity)
    → see RSA: Bernstein, TWIRL device, RSA-160, RSA-200 (CrypTool script, chapter 4.11.3)
  - Very high activity in current research:

    Factorization, non-parallelizable algorithm (because of quantum computing), better understanding of protocol weaknesses and random generators, ...).
- Serious mistake: “Real mathematics has no effects on the war.” (G.H. Hardy, 1940)
- Vendors discover security as an essential purchase criterion.

Demonstration in CrypTool

- *Statistic Analysis*
- *Encrypting twice is not always better:*
  - *Caesar: C + D = G (3 + 4 = 7)*
  - *Vigenère: - CAT + DOG = FOZ [(2,0,19)+(3,14,6)=(5,14,25)]*
    - *"Hund" + "Katze" ="RUGCLENWGYXDATRNHNMH")*
- *Vernam (OTP)*
- *AES (output key, brute-force analysis)*

Content

I. CrypTool and Cryptology – Overview

II. CrypTool Features

III. Examples

IV. Project / Outlook / Contact

CrypTool Features

eLearning

1. What is CrypTool?

- Freeware program with graphical user interface
- Cryptographic methods can be applied *and* analysed
- Comprehensive online help (understandable without deeper cryptography knowledge)
- Contains nearly all state-of-the-art cryptography functions
- Easy entry into modern and classical cryptography
- Not a *"hacker tool"*

2. Why CrypTool?

- Origin in awareness initiative of a financial institute
- Developed in close cooperation with universities
- Improvement of university education and in-firm training

3. Target group

- *Core group*: Students of computer science, business computing and mathematics
- *But also for*: computer users, application developers, employees
- *Prerequisite*: PC knowledge
- *Preferable*: Interest in mathematics and/or programming

Content of the Program Package

German, English, Polish and Spanish

CrypTool program
- All functions integrated in a *single* program with consistent graphical interface
- Runs on Win32
- Cryptography libraries from Secude and OpenSSL
- Long integer arithmetic from Miracl and GMP, Lattice base reduction via NTL (Shoup)

AES-Tool
- Standalone program for AES encryption (and creation of self extracting files)

Educational game
- „Number Shark“ encourages the understanding of factors and prime numbers.

Comprehensive Online Help (HTML-Help)
- Context-sensitive help available via F1 for all program functions (including menus)
- Detailed use cases for a lot of program functions (tutorial)

Script (.pdf file) with background information
- Encryption methods • Prime factorization • Digital signature
- Elliptic curves • Public-key certification • Basic number theory • Crypto 2020 • SAGE

Two short stories related to cryptography by Dr. C. Elsner
- „The Dialogue of the Sisters“ (a RSA variant as key element)
- „The Chinese Labyrinth“ (Numbers theory tasks for Marco Polo)

Learning tool for number theory

Features (1)

Cryptography

Classical cryptography

- Caesar (and ROT-13)
- Monoalphabetic substitution (and Atbash)
- Vigenère
- Hill
- Homophone substitution
- Playfair
- ADFGVX
- Byte Addition
- XOR
- Vernam
- Permutation / Transposition (Rail Fence, Scytale, ...)
- Solitaire

Several options to easily comprehend cryptography samples from literature:

- Selectable alphabet
- Options: handling of blanks, etc.

Cryptanalysis

Attack on classical methods

- Cipher text only
  - Caesar
  - Vigenère
  - Addition
  - XOR
  - Substitution
  - Playfair
- Known-plaintext
  - Hill
  - Single-column transposition
- Manually (supported)
  - Mono alphabetical substitution
  - Playfair, ADFGVX, Solitaire

Supported analysis methods

- Entropy, floating frequency
- Histogram, n-gram analysis
- Autocorrelation
- Periodicity
- Random analysis
- Base64 / UU-Encode

Features (2)

Cryptography

Modern symmetric encryption

- IDEA, RC2, RC4, RC6, DES, 3DES, DESX
- AES candidates of the last selection round (Serpent, Twofish, ...)
- AES (=Rijndael)
- DESL, DESXL

Asymmetric encryption

- RSA with X.509 certificates
- RSA demonstration
  - Understanding of examples
  - Alphabet and block length selectable

Hybrid encryption (RSA + AES)

- Interactive data flow diagram

Cryptanalysis

Brute-force attack on symmetric algorithm

- For all algorithms
- Assumptions:
  - Entropy of plaintext is small or key is partly known or plaintext alphabet is known

Attack on RSA encryption

- Factorization of RSA module
- Lattice-based attacks

Attack on hybrid encryption

- Attack on RSA or
- Attack on AES (side-channel attack)

Features (3)

Cryptography

Digital signature

- RSA with X.509 certificates
  - Signature as data flow diagram
- DSA with X.509 certificates
- Elliptic Curve DSA, Nyberg-Rueppel

Hash functions

- MD2, MD4, MD5
- SHA, SHA-1, SHA-2, RIPEMD-160

Random generators

- Secude
- x^2 mod n
- Linear congruence generator (LCG)
- Inverse congruence generator (ICG)

Cryptanalysis

Attack on RSA signature

- Factorization of the RSA module
- Feasible up to 250 bits or 75 decimal places (on standard desktop PCs)

Attack on hash functions / digital signature

- Generate hash collisions for ASCII based text (birthday paradox) (up to 40 bit in around 5 min)

Analysis of random data

- FIPS-PUB-140-1 test battery
- Periodicity, Vitany, entropy
- Floating frequency, histogram
- n-gram analysis, autocorrelation
- ZIP compression test

Features (4)

Animation / Demos

- Caesar, Vigenère, Nihilist, DES (all with ANIMAL)
- Enigma (Flash)
- Rijdael/AES (twice with Flash, once with Java)
- Hybrid encryption and decryption (AES-RSA and AES-ECC)
- Generation and verification of digital signatures
- Diffie-Hellman key exchange
- Secret sharing (with CRT or Shamir)
- Challenge-response method (authentication)
- Side-channel attack
- Secure e-mail with the S/MIME protocol (with Java and Flash)
- Graphical 3D presentation of (random) data streams
- Sensitivity of hash functions regarding plaintext modifications
- Number theory and RSA crypto system (with Authorware)

Features (5)

Additional functions

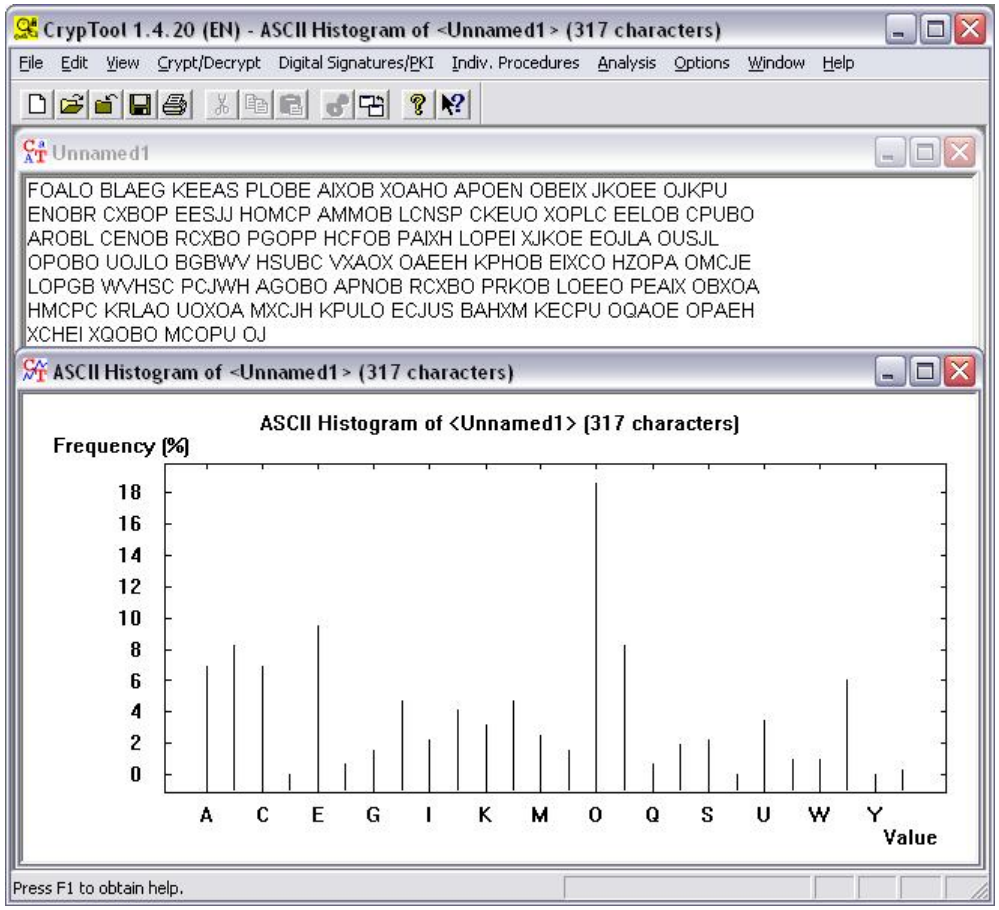
- Homophone and permutation encryption (Double Column Transposition)
- PKCS #12 import and export for PSEs (Personal Security Environment)
- Generate hashes of large files, without loading them
- Flexible brute-force attacks on any modern symmetric algorithm
- ECC demonstration (as Java application)
- Password Quality Meter (PQM) and password entropy
- And a lot more ...

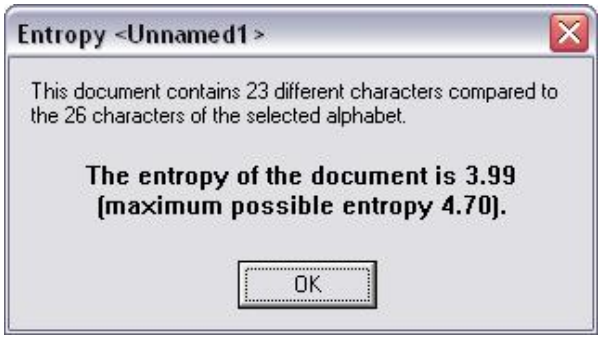
Language Structure Analysis

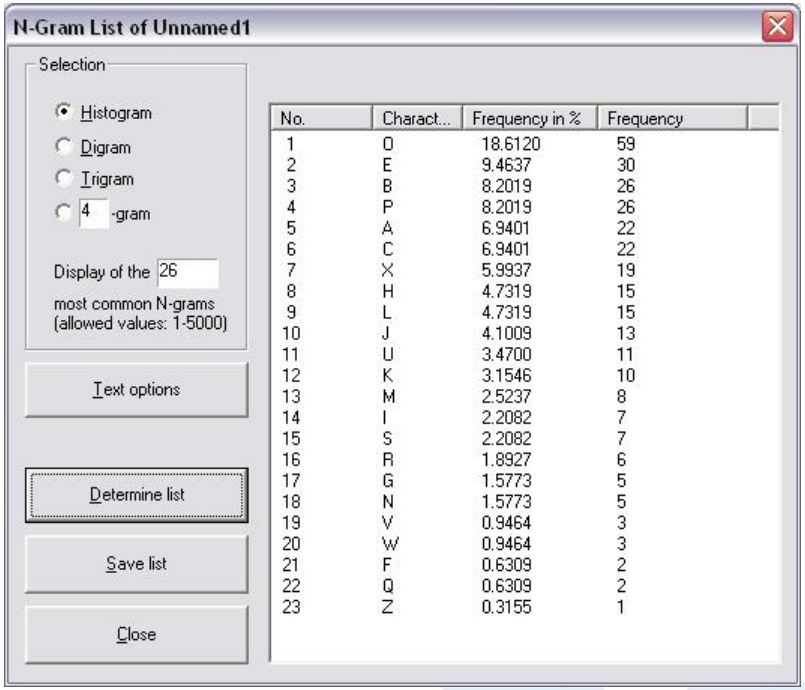
Language analysis options available in CrypTool

Number of characters, n-gram, entropy

- See menu „Analysis“ \ „Tools for Analysis“ \ ...







Demonstration of Interactivity (1)

Vigenère analysis

Demonstration in CrypTool

The result of the Vigenère analysis can be manually reworked (changing the key length):

1. Encrypt starting example with TESTETE
   - *„Crypt/Decrypt“* \ *„Symmetric (classic)“* \ *„Vigenère“*
   - Enter TESTETE ⇨ *„Encrypt“*

   Analysis of the encryption results:
   - *„Analysis“* \ *„Symmetric Encryption (classic)“* \ *„Ciphertext only“* \ *„Vigenère“*
   - Derived key length 7, Derived key TESTETE
2. Encrypt starting example with TEST
   - *„Crypt/Decrypt“* \ *„Symmetric (classic)“* \ *„Vigenère“*
   - Enter TEST ⇨ *„Encrypt“*

   Analysis of the encryption results:
   - *„Analysis“* \ *„Symmetric Encryption (classic)“* \ *„Ciphertext only“* \ *„Vigenère“*
   - Derived key length 8 – not correct
   - Key length automatically set to 4 (can also be adjusted manually)
   - Derived key TEST

Demonstration of Interactivity (2)

Automated factorization

Demonstration in CrypTool

Factorization of a compound number with factorization algorithms

- Menu: „Indiv. Procedures“ \ „RSA Cryptosystem“ \ „Factorization of a Number“
- Some methods are executed in parallel (multi-threaded)
- Methods have specific advantages and disadvantages (e.g. some methods can only determine small factors)

Factorization example 1 :

316775895367314538931177095642205088158145887517

48-digit decimal number

=

3 * 1129 * 6353 * 1159777 * 22383173213963 * 567102977853788110597

Factorization example 2:

2^250 - 1

75-digit decimal number

=

3 * 11 * 31 * 251 * 601 * 1801 * 4051 * 229668251 * 269089806001 * 4710883168879506001 * 5519485418336288303251

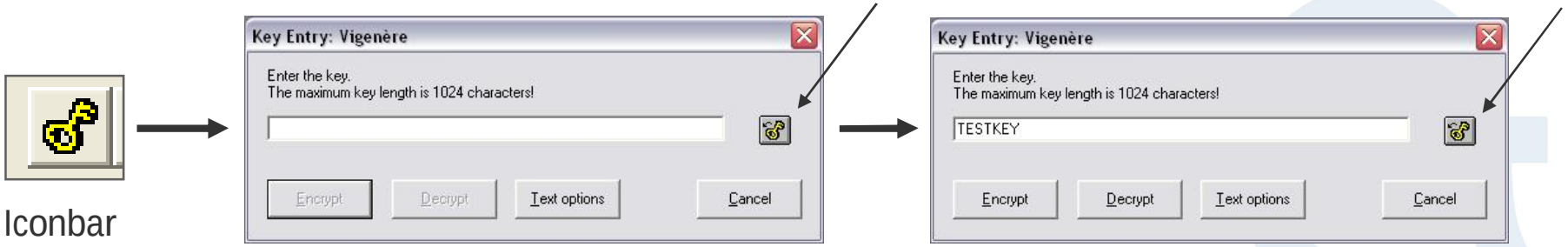
Concepts for a User-Friendly Interface

1. Context sensitive help (F1)

- F1 on a selected menu entry shows information about the algorithm/method.
- F1 in a dialog box explains the usage of the dialog.
- These assistances and the contents of the super ordinate menus are cross linked in the online help.

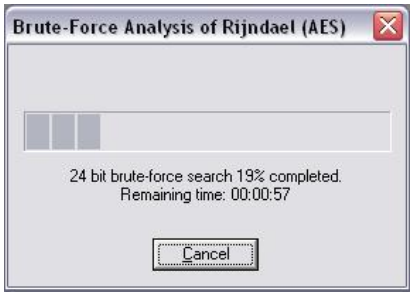
2. Paste of keys in key-input dialog

- CTRL-V can be used to paste contents from the clipboard.
- Used keys can be taken out of cipher text windows via an icon in the icon bar. A corresponding icon in the key-input dialog can be used to paste the key into the key field. A CrypTool-internal memory which is available for every method is used (helpful for large „specific“ keys – e.g. homophone encryption).



Challenges for Developers (Examples)

1. Different functions running in parallel
   - Factorization runs with multi-threaded algorithms
2. High performance
   - Locate hash collisions (birthday paradox) or perform brute-force analysis
3. Consider memory limits
   - Floyd algorithm (mappings to locate hash collisions) or factorization with quadratic sieve
4. Time measurement and estimates
   - Display of elapsed time while using brute-force
5. Reusability / Integration
   - Forms for prime number generation
   - RSA cryptosystem (switches the view after successful attack from public key user to private key owner)
6. Partly automate the consistency of functions, GUI and online help (including different languages)



Content

CrypTool Examples

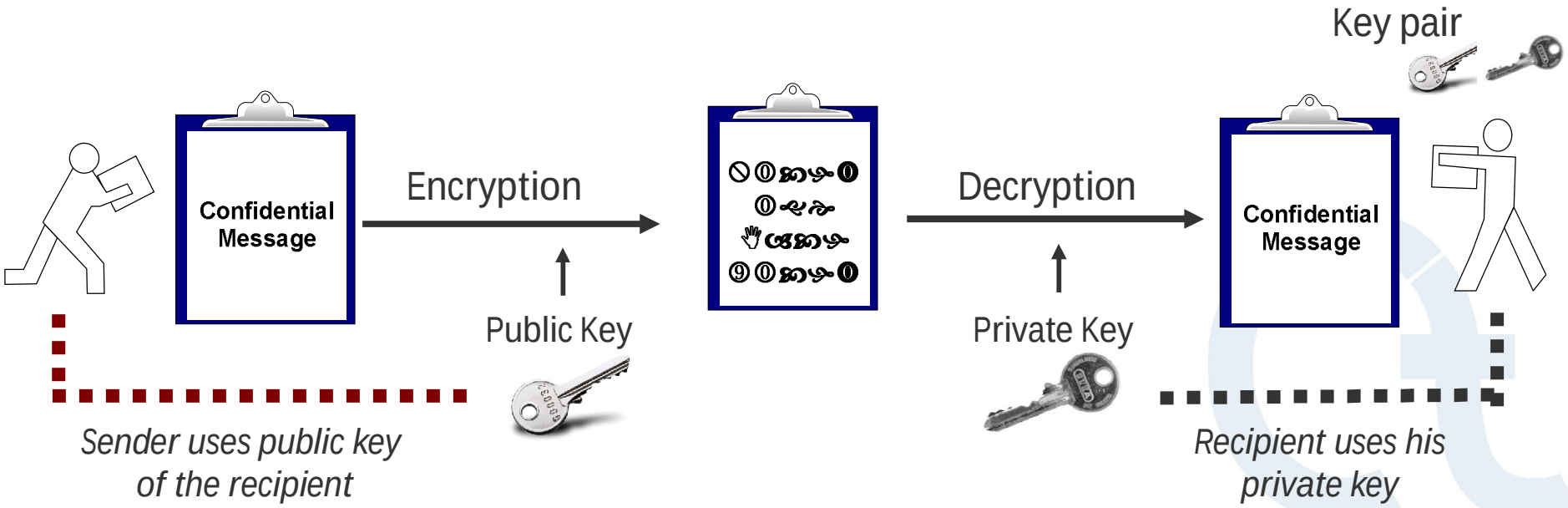
Overview of examples

1. Encryption with RSA / Prime number tests / Hybrid encryption and digital certificates / SSL
2. Digital signature visualized
3. Attack on RSA encryption (modul N too short)
4. Analysis of encryption in PSION 5
5. Weak DES keys
6. Locating key material (“NSA key”)
7. Attack on digital signature through hash collision search
8. Authentication in a client-server environment
9. Demonstration of a side-channel attack (on hybrid encryption protocol)
10. Attack on RSA using lattice reduction
11. Random analysis with 3-D visualization
12. Secret Sharing using the Chinese Remainder Theorem (CRT) and Shamir
13. Implementation of CRT in astronomy (solving linear modular equation systems)
14. Visualization of symmetric encryption methods using ANIMAL
15. Visualizations of AES
16. Visualization of Enigma encryption
17. Secure E-Mail with S/MIME
18. Generation of a message authentication code (MAC)
19. Hash demonstration
20. Learning tool for number theory and asymmetric encryption
21. Point addition on elliptic curves
22. Password quality meter (PQM) and password entropy
23. Brute-force analysis
24. CrypTool online help

Examples (1)

Encryption with RSA (in reality mostly hybrid encryption)

- Basis for e.g. SSL protocol (access to protected web sites)
- Asymmetric encryption using RSA
  - Every user has a key pair – one public and one private key
  - Sender encrypts with public key of the recipient
  - Recipient decrypts with his private key
- Implemented usually in a combination with symmetric methods (transfer of the symmetric key through RSA asymmetric encryption/decryption)



Examples (1)

Encryption using RSA – Mathematical background / algorithm

- Public key: (n, e)
- Private key: (d)

where:

p, q large, randomly chosen prime numbers with $n = p*q$;

d is calculated under the constraints $\gcd[\varphi(n),e] = 1$; $e*d \equiv 1 \bmod \varphi(n)$.

Encryption and decryption operation: $(m^e)^d \equiv m \bmod n$

- n is the module, which length in bits is referred to as RSA key length.
- gcd = greatest common divisor.
- $\varphi(n)$ is the Euler phi function.

Procedure :

- Transformation of message in binary representation
- Encrypt message $m = m_1,...,m_k$ block wise, with for all m_j:
  $0 \leq m_j < n$; maximum block size r, so that: $2^r \leq n$ $(2^r-1 < n)$

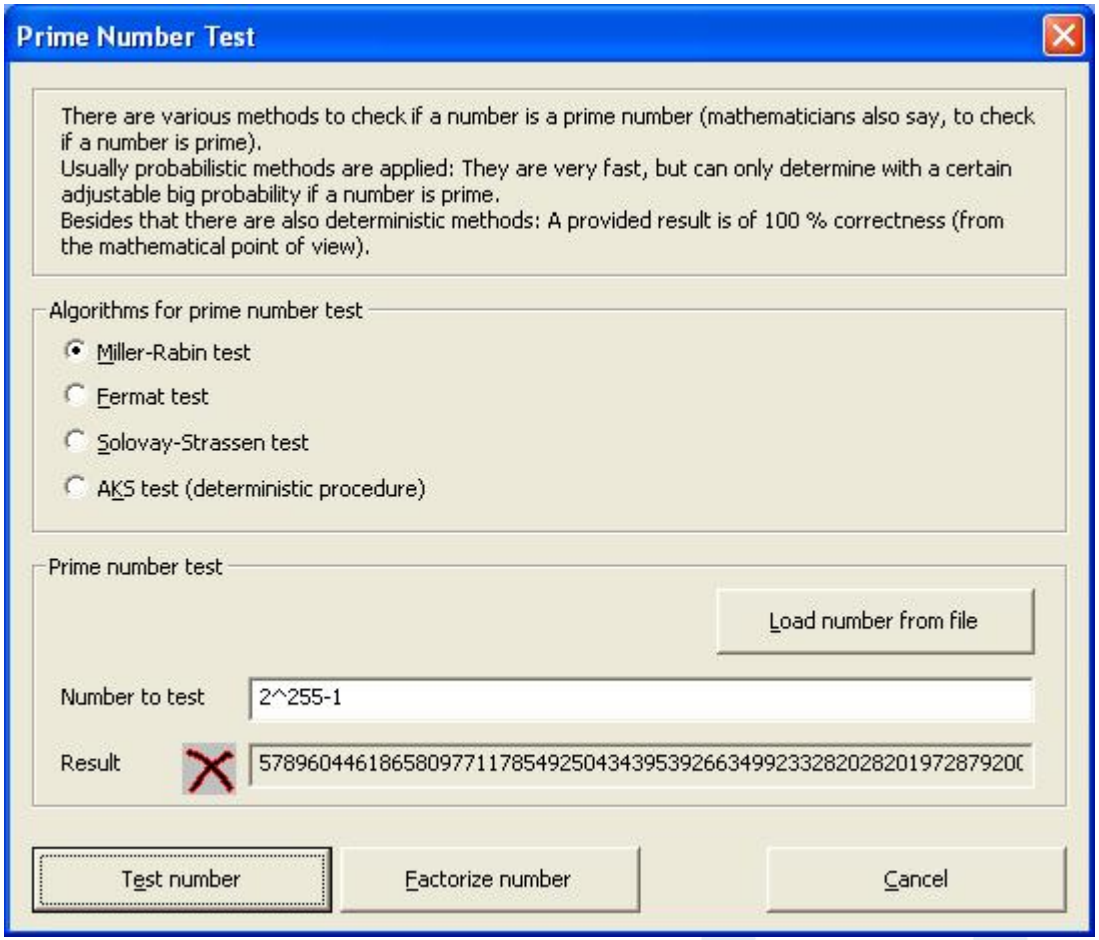
Examples (1)

Prime number tests – For RSA huge primes are needed

- Fast probabilistic tests
- Deterministic tests

The prime number test methods can test much faster whether a big number is prime, than the known factorization methods can divide a number of a similar size in its prime factors.

For the AKS test the GMP library (**G**NU **M**ultiple **P**recision Arithmetic Library) was integrated into CrypTool.



Menu: „Indiv. Procedure“ \ „RSA Cryptosystem“ \ „Prime Number Test “

Remark: 2^255 - 1 = 7 * 31 * 103 * 151 * 2143 * 11119 * 106591 * 131071 * 949111 * 9520972806333758431 * 57024515776397755458386431 51

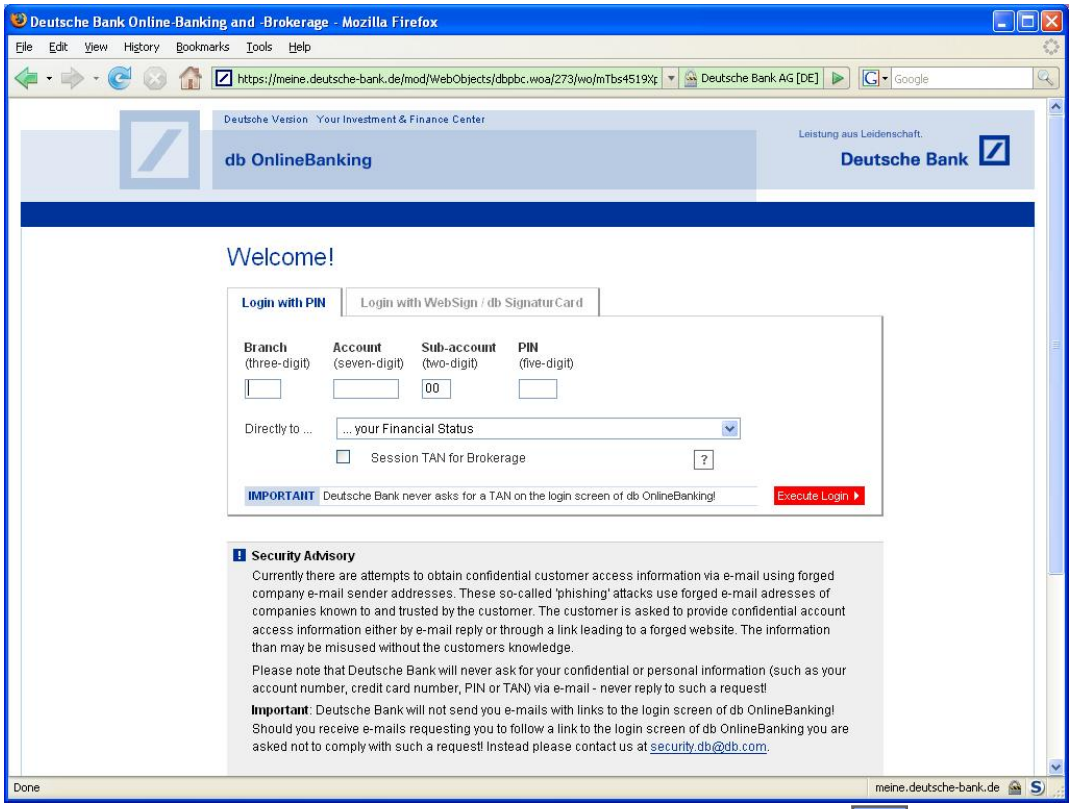
Examples (1)

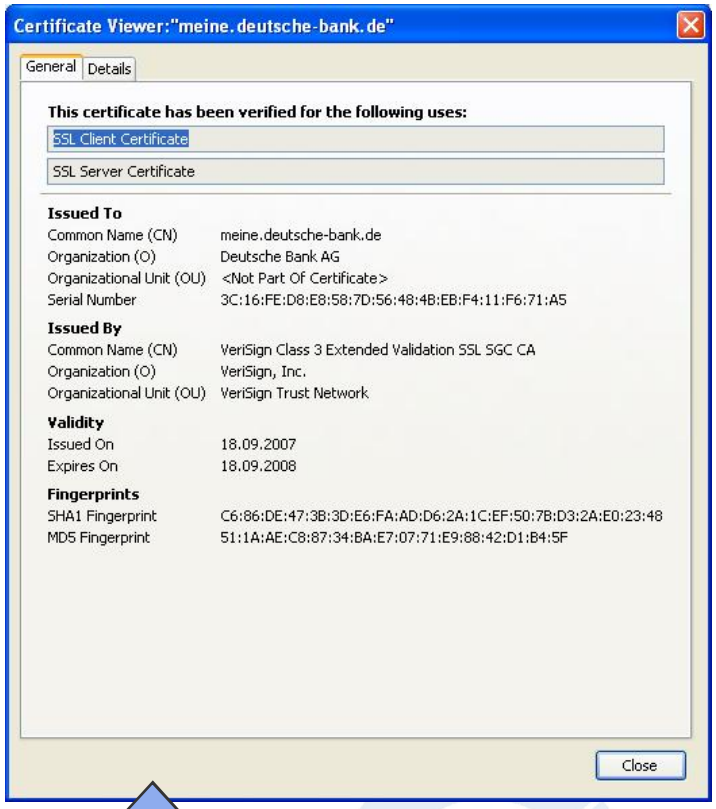
Hybrid encryption and digital certificates

- Hybrid encryption – Combination of asymmetric and symmetric encryption
  1. Generation of a random symmetric key (session key)
  2. Session key is transferred – protected by asymmetric key
  3. Message is transferred – protected by session key
- Problem: Man-in-the-middle attacks – does the public key of the recipient really belong to the recipient?
- Solution: Digital certificates – A central instance (e.g. Telesec, VeriSign, Deutsche Bank PKI), that is being trusted by all users, ensures the authenticity of the certificate and the contained public key (similar to a passport issued by the state).
- Hybrid encryption based on digital certificates is the foundation for all secured electronic communication:
  - Internet Shopping and Online Banking
  - Secure eMail

Examples (1)

Secured online connection using SSL and certificates

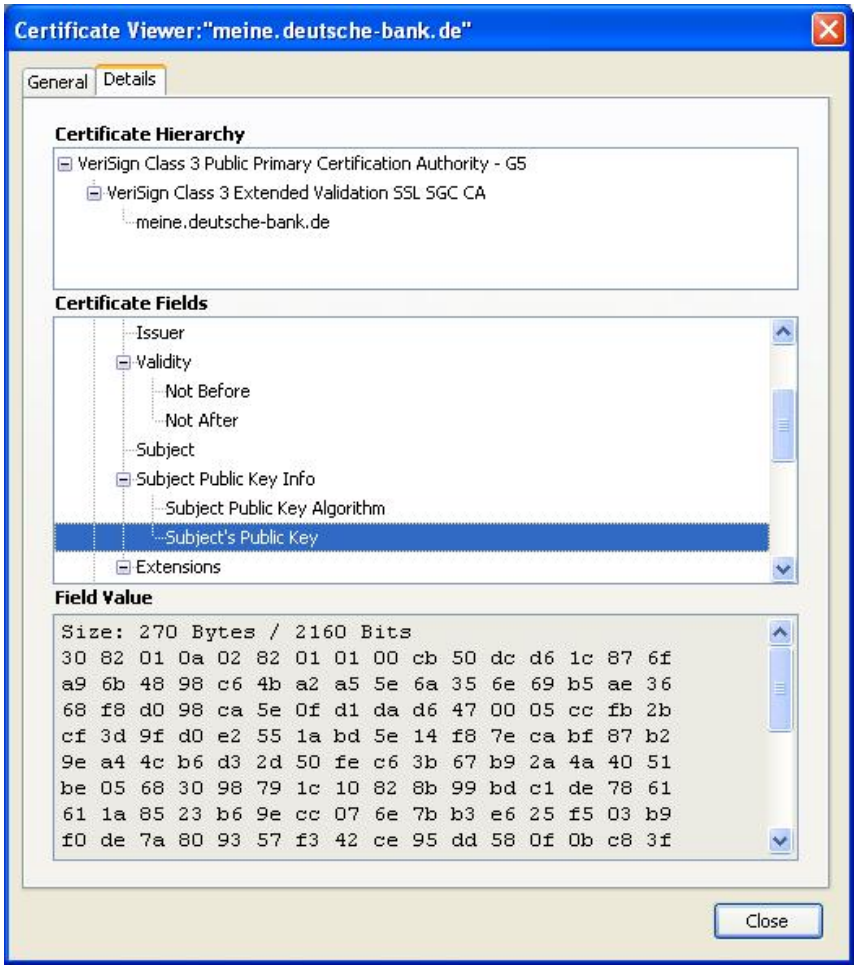




This means, that the connection is authenticated (at least at one side) and that the transferred data is strongly encrypted.

Examples (1)

Attributes or fields of a certificate



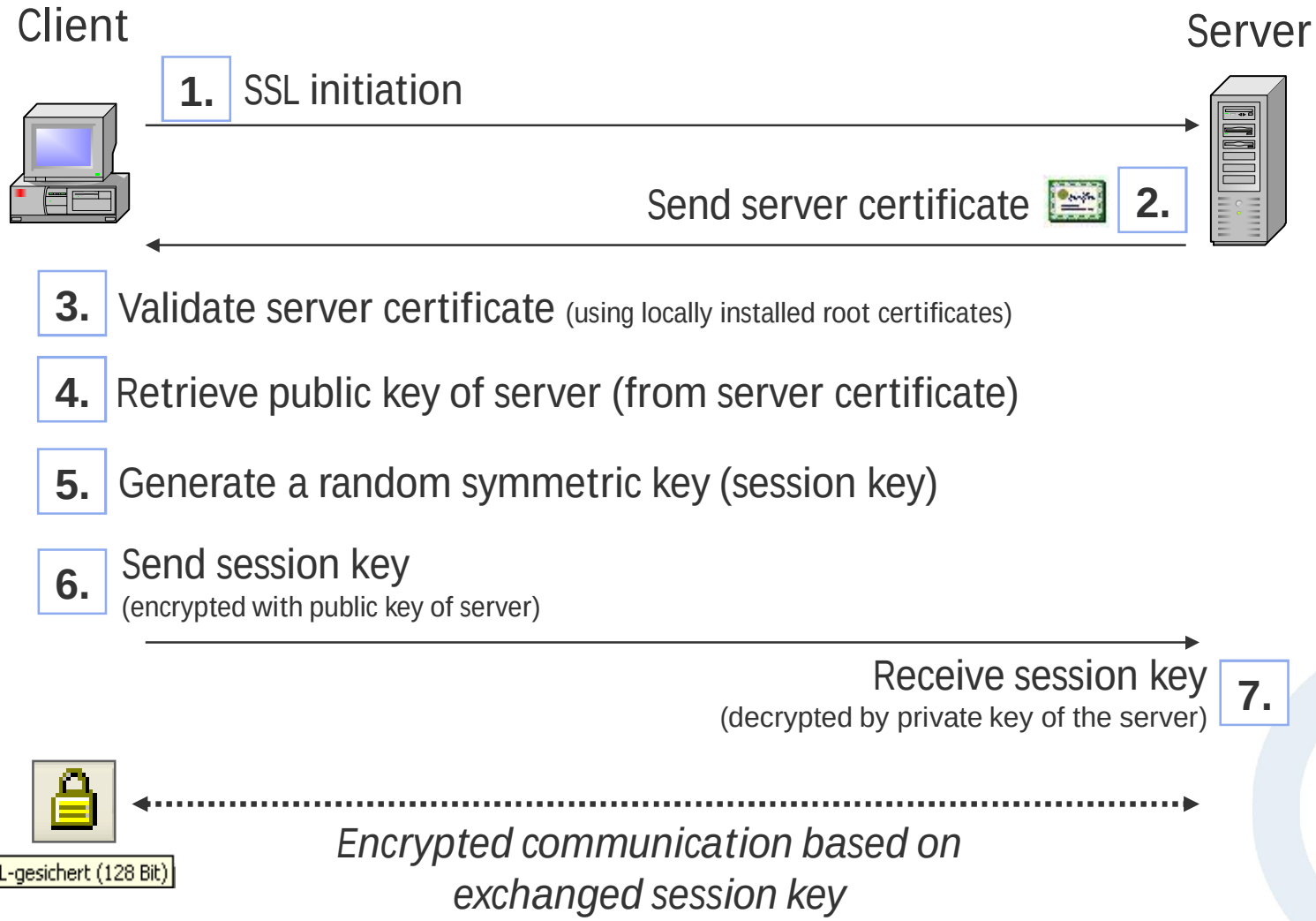
General attributes / fields

- Issuer (e.g. VeriSign)
- Requestor
- Validity period
- Serial number
- Certificate type / Version (X.509v3)
- Signature algorithm
- Public key (and method)

Public Key

Examples (1)

Establishing a secure SSL connection (server authentication)



Examples (1)

Establishing a secure SSL connection (server authentication)

General

- The example shows the typical SSL connection establishment in order to transfer sensitive data over the internet (e.g. online shopping).
- During SSL connection establishment only the server is authenticated using the digital certificate (authentication of the user usually occurs through user name and password after the SSL connection has been established).
- SSL also offers the option for client authentication based on digital certificates.

Comments to the SSL connection establishment

- ad (1): SSL Initiation – during this phase the characteristics of the session key (e.g. bit size) as well as the symmetric encryption algorithm (e.g. 3DES, AES) are negotiated.
- ad (2): In case of a multi-level certificate hierarchy the required intermediate certificates are being passed to the client, too.
- ad (3): In this phase the root certificates installed in the browser‘s certificate store are used to validate the server certificate.
- ad (5): The session key is based on the negotiated characteristics (see 1).

Examples (2)

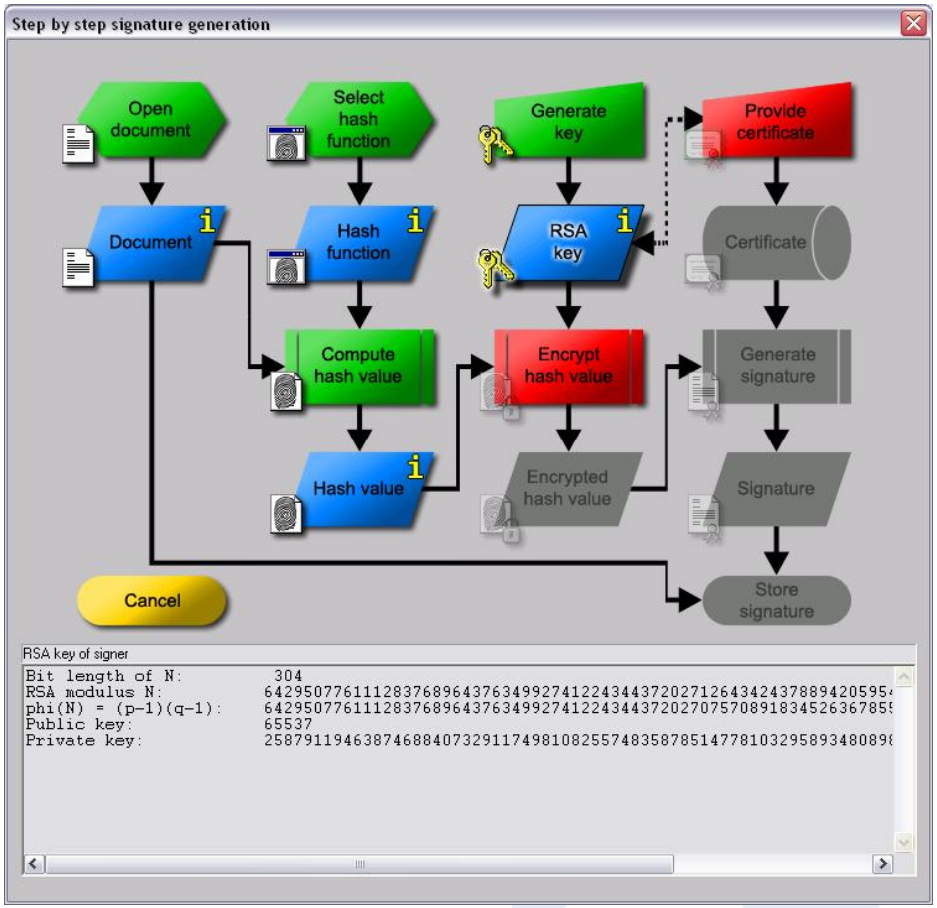
Digital signature visualized

Digital signature

- Increasingly important
  - equivalence with manual signature (digital signature law)
  - increasingly used by industry,
  - government and consumers
- Few people know how it works exactly

Visualization in CrypTool

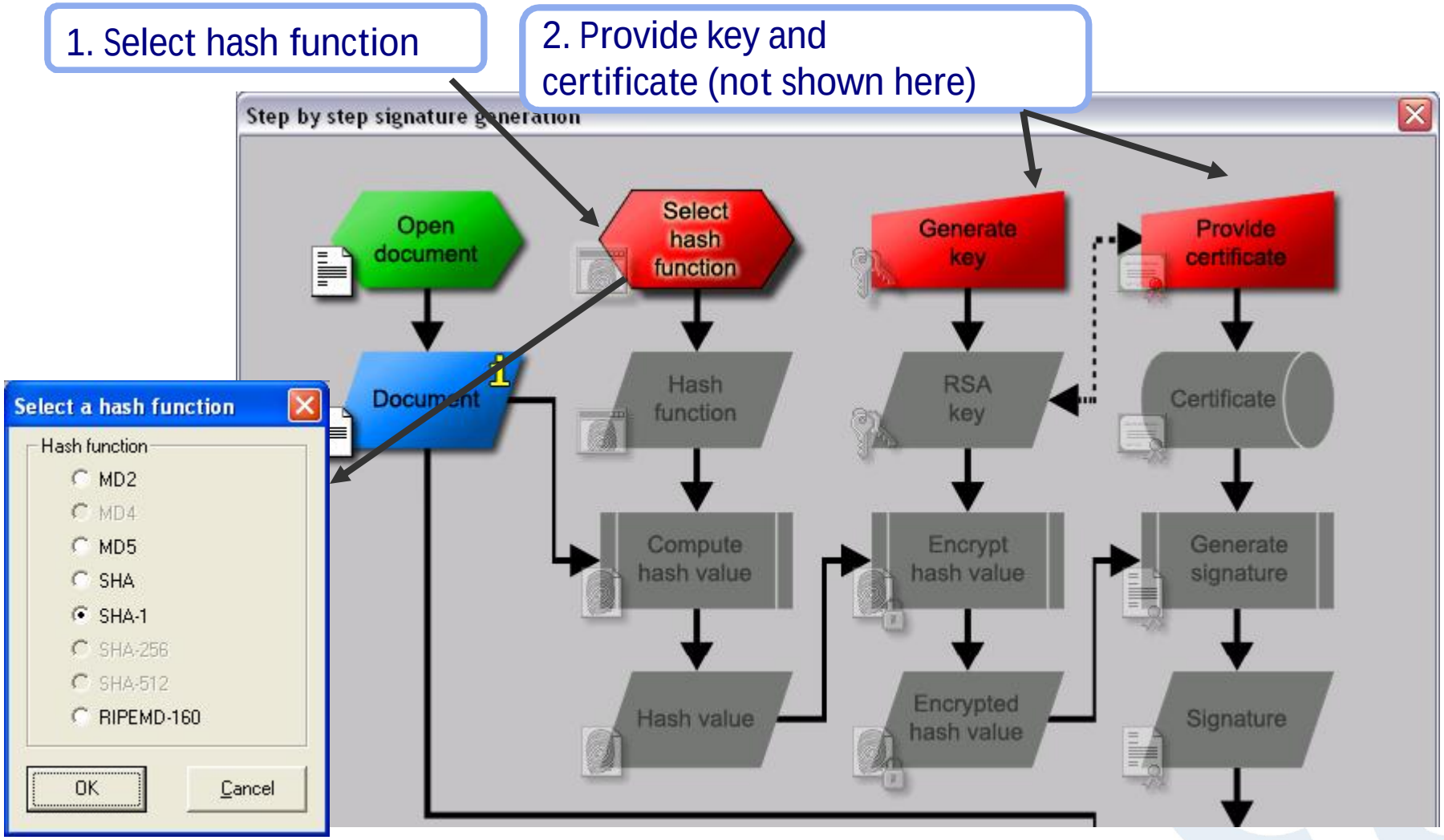
- Interactive data flow diagram
- Similar to the visualization of hybrid encryption



Menu: „Digital Signatures/PKI“ \
„Signature Demonstration (Signature Generation)“

Examples (2)

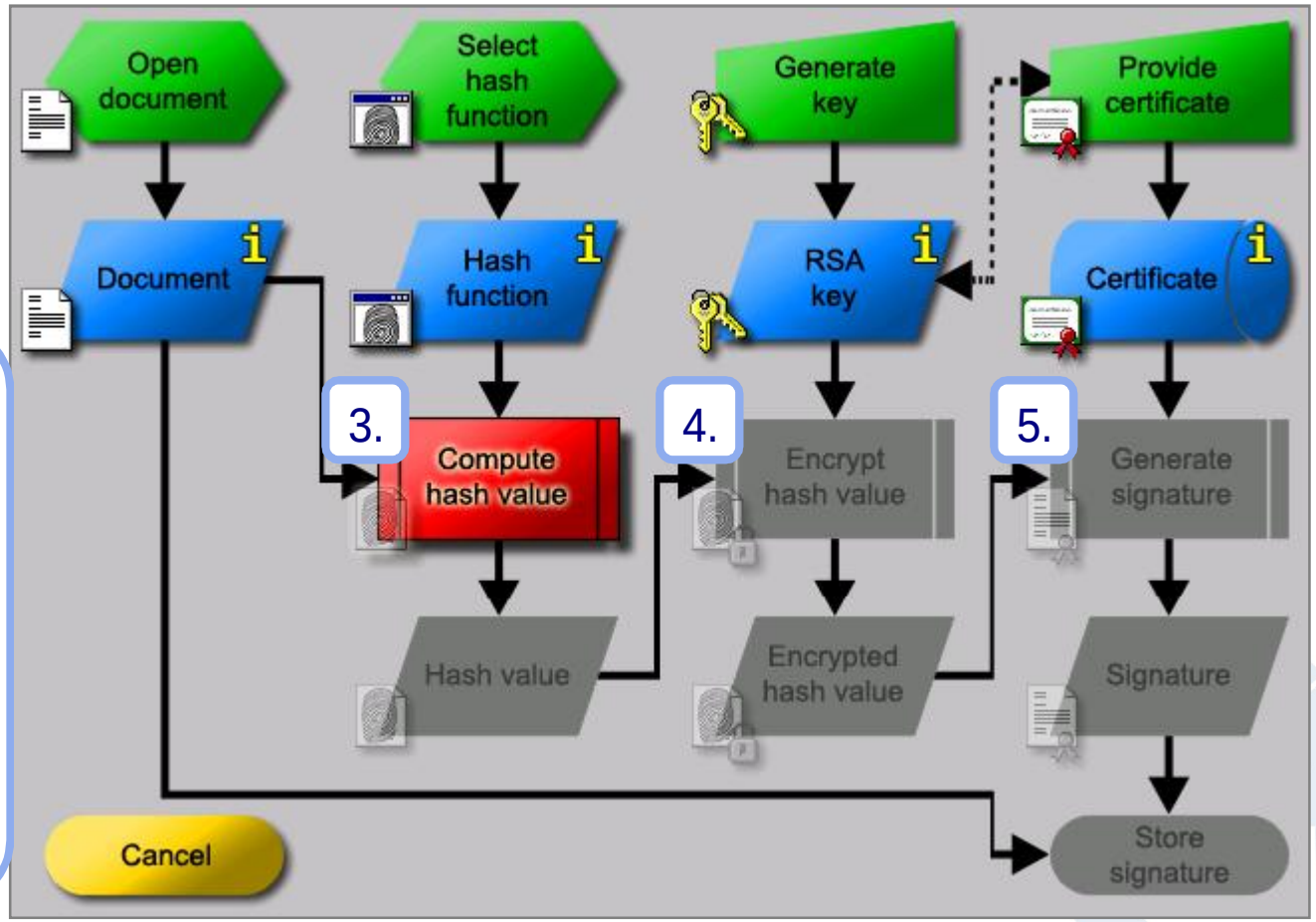
Digital signature visualized: a) Preparation



Examples (2)

Digital signature visualized: b) Cryptography

3. Calculate hash value
4. Encrypt hash value with private key (sign)
5. Generate signature

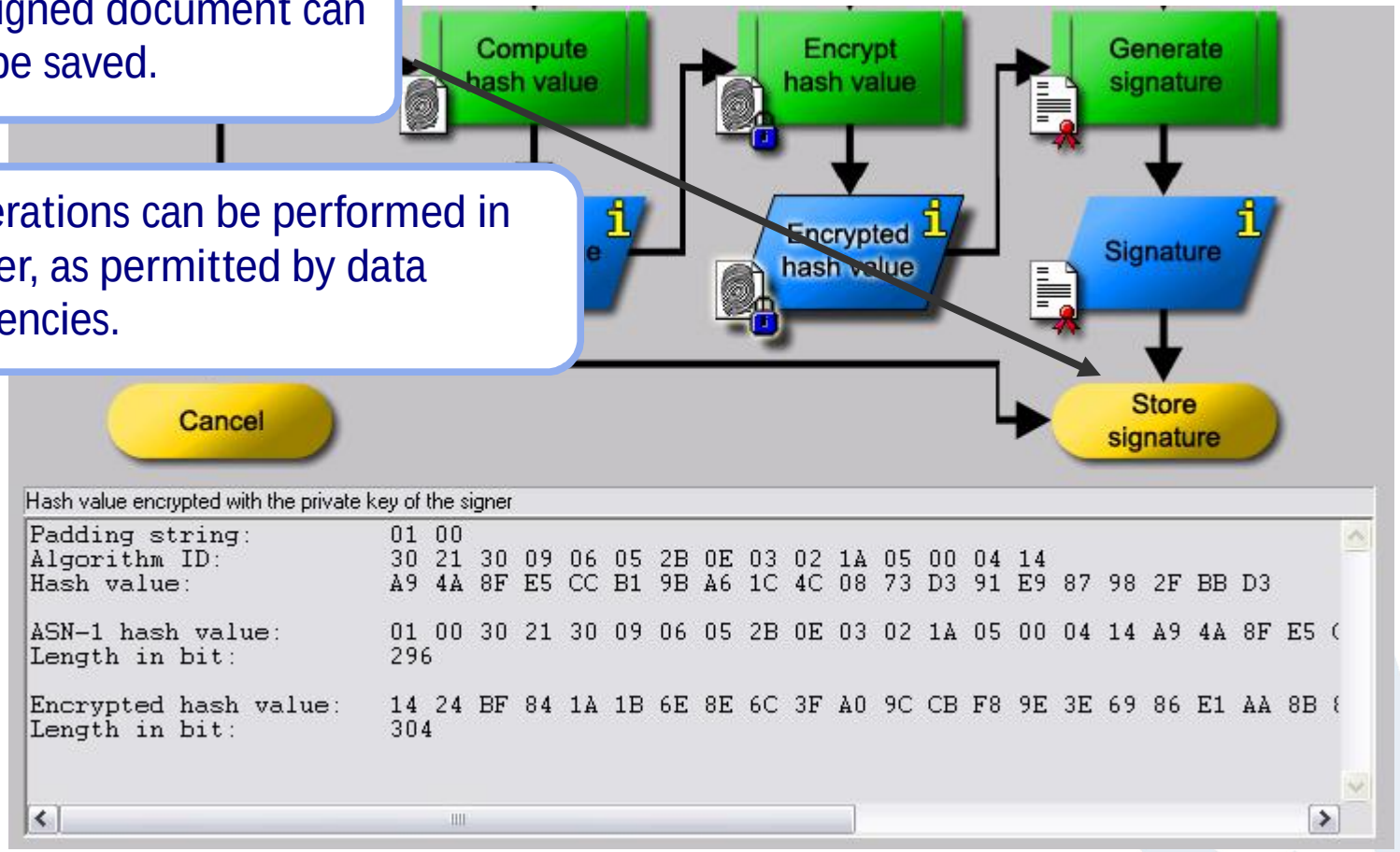


Examples (2)

Digital signature visualized: c) Result

6. The signed document can now be saved.

The operations can be performed in any order, as permitted by data dependencies.



Examples (3)

Attack on RSA encryption with short RSA modulus

Example from *Song Y. Yan*, Number Theory for Computing, Springer, 2000

- Public key
  - RSA modulus N = 639784868795271438588314150 41 (95 bit, 29 decimal digits)
  - public exponent e = 17579
- Ciphertext (block length = 8):

  C_1 = 45411667895024938209259253423,
  C_2 = 16597091621432020076311552201,
  C_3 = 46468979279750354732637631044,
  C_4 = 32870167545903741339819671379
- The text shall be deciphered!

The ciphertext is not necessary for the actual cryptanalysis (locating the private key) !

Solution using CrypTool (more detailed in online help examples section)

- Enter public parameters into “RSA cryptosystem” (menu: „Indiv. Procedures”)
- Button “Factorize the RSA modulus” yields the two prime factors pq = N
- Based on that information the private exponent $d=e^{-1}$ mod (p-1)(q-1) is determined
- Decrypt the cipher text with d: $M_i = C_i^d$ mod N

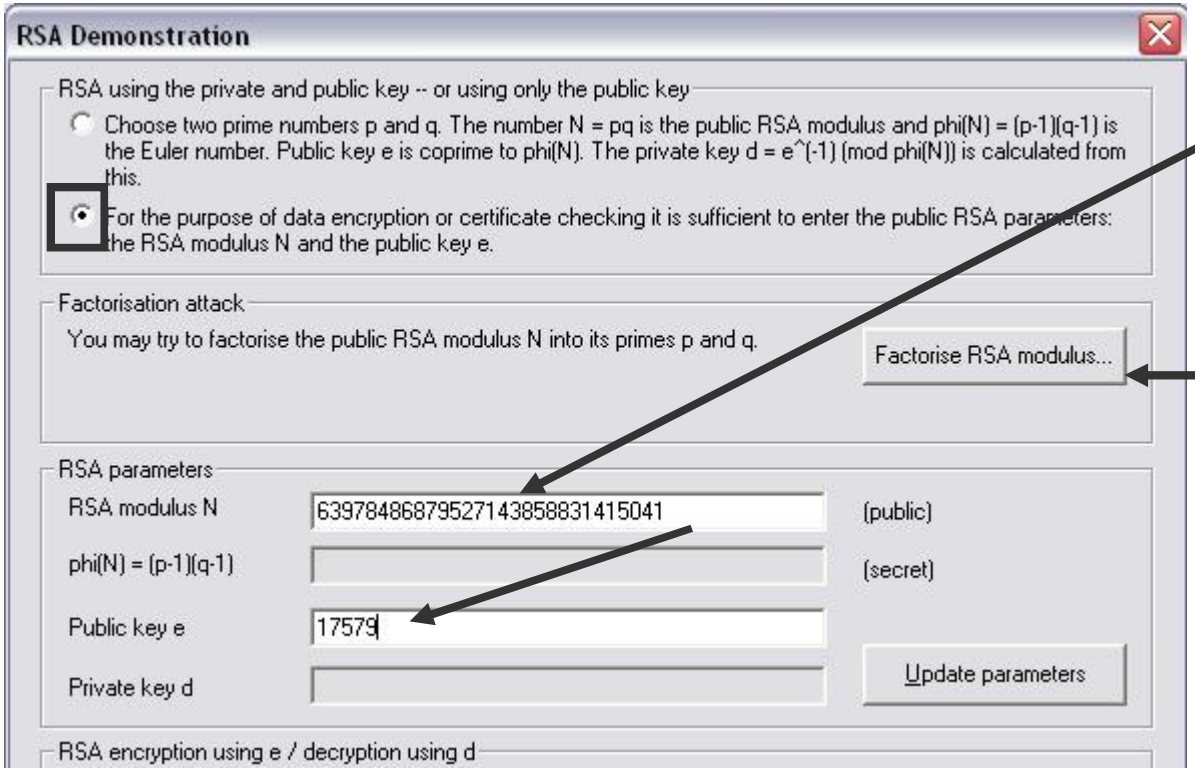
The attack with CrypTool works for RSA moduli up to 250 bit.

Then you could digitally sign for someone else !

Examples (3)

Short RSA modulus: enter public RSA parameters

Menu: „Indiv. Procedures“ \ „RSA Cryptosystem” \ „RSA Demonstration ...”

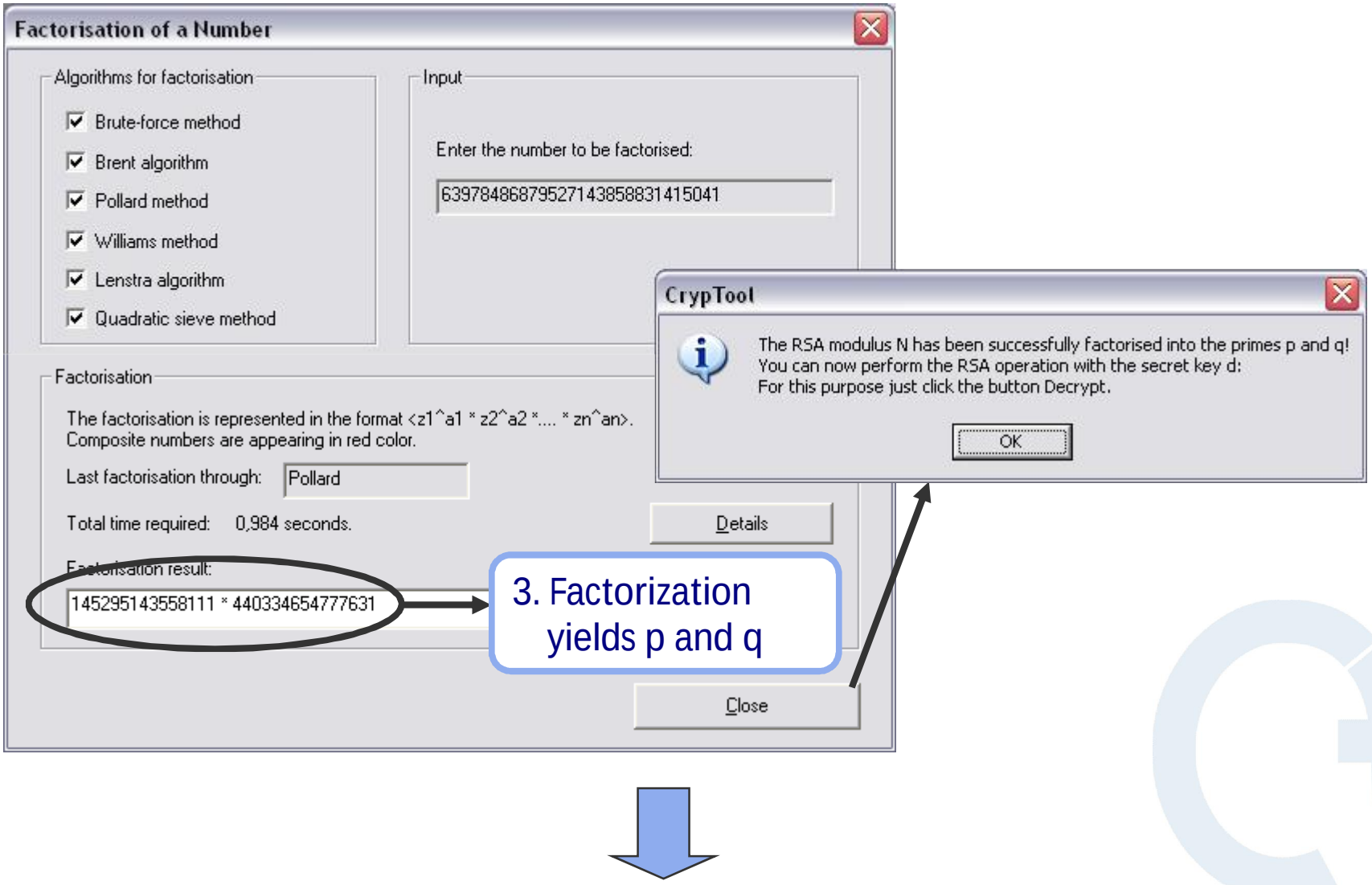


1. Enter RSA parameters N and e

2. Factorize

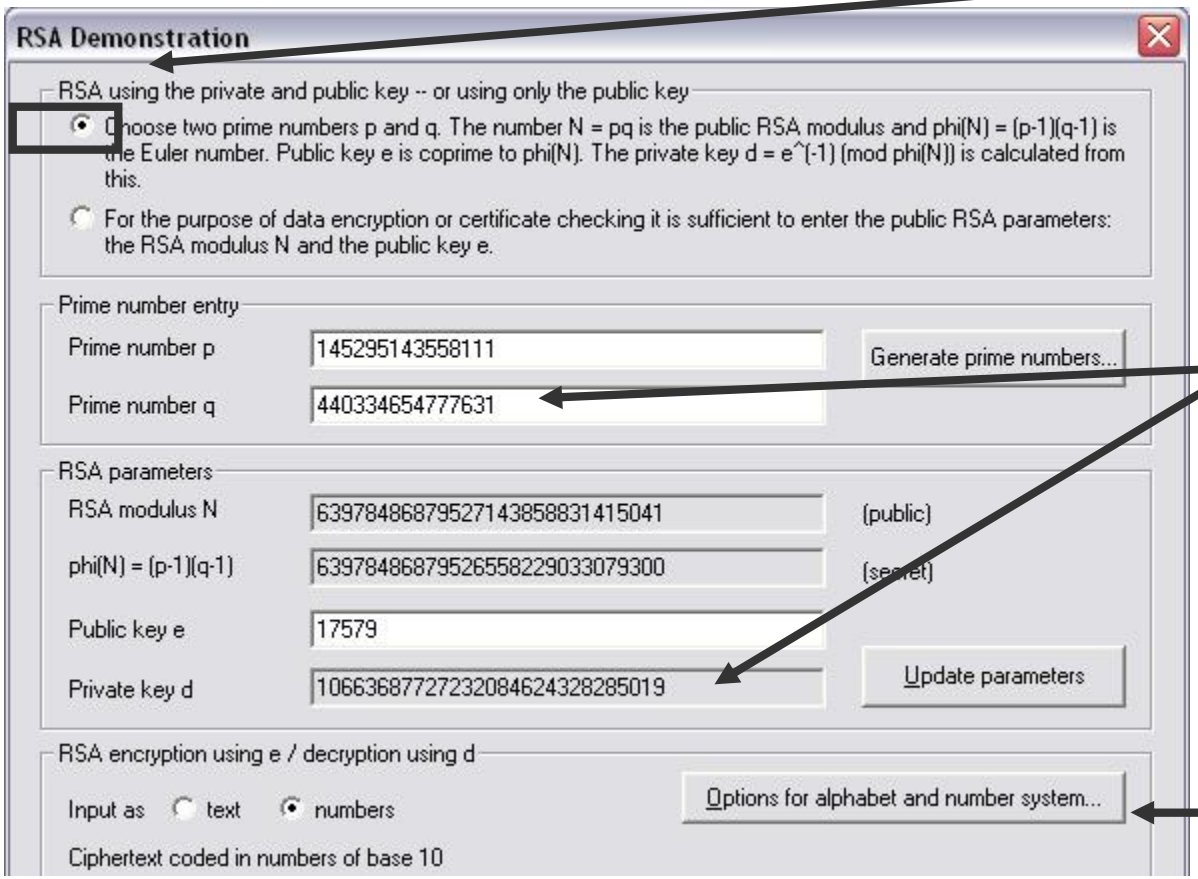
Examples (3)

Short RSA modulus: factorize RSA modulus



Examples (3)

Short RSA modulus: determine private key d



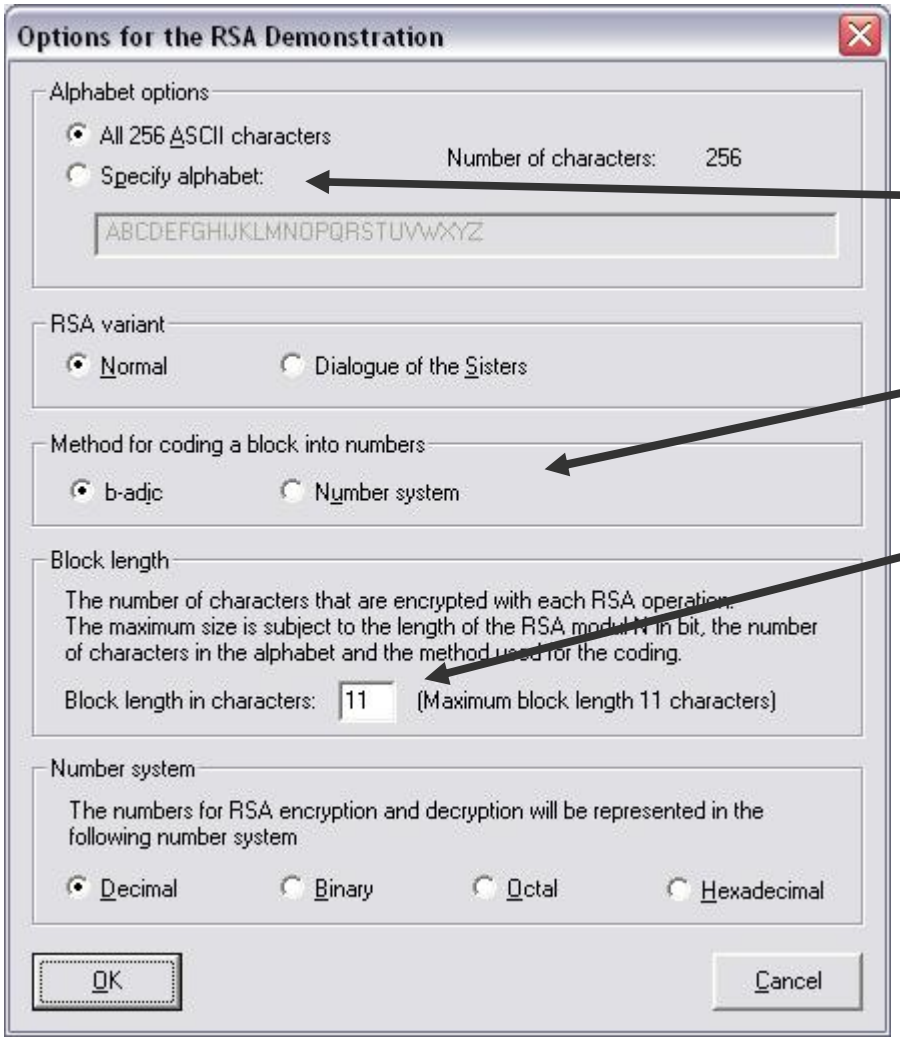
Change the view to the owner of the secret key

4. p and q have been entered automatically and secret key d has been calculated

5. Adjust options

Examples (3)

Short RSA modulus: adjust options



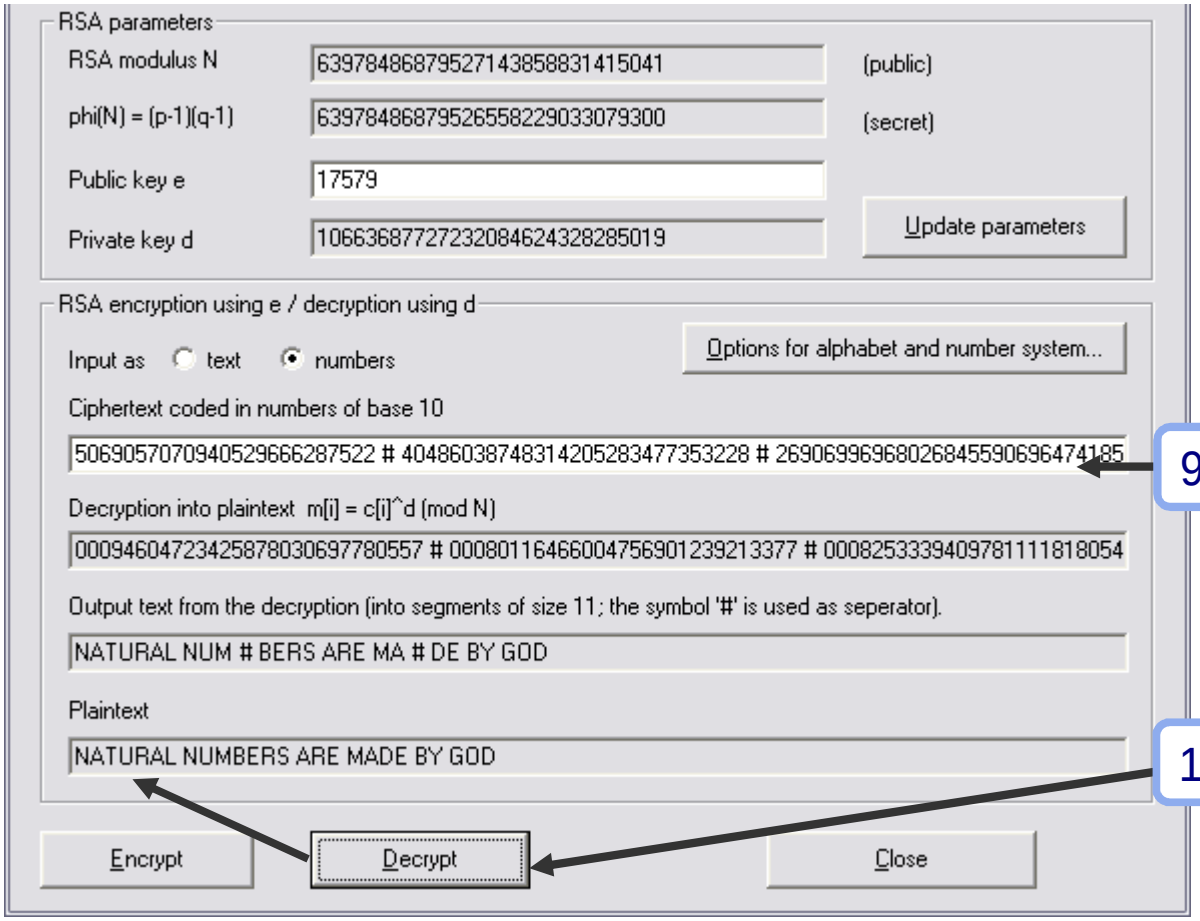
6. Select alphabet

7. Select coding method

8. Select block length

Examples (3)

Short RSA modulus: decrypt cipher text



9. Enter cipher text

10. Decrypt

Examples (4)

Analysis of encryption used in the PSION 5

Practical application of cryptanalysis:

Attack on the encryption option in the PSION 5 PDA word processing application

Starting point: an encrypted file on the PSION

Requirements

- Encrypted English or German text
- Depending on method and key length, 100 bytes up to several kB of text

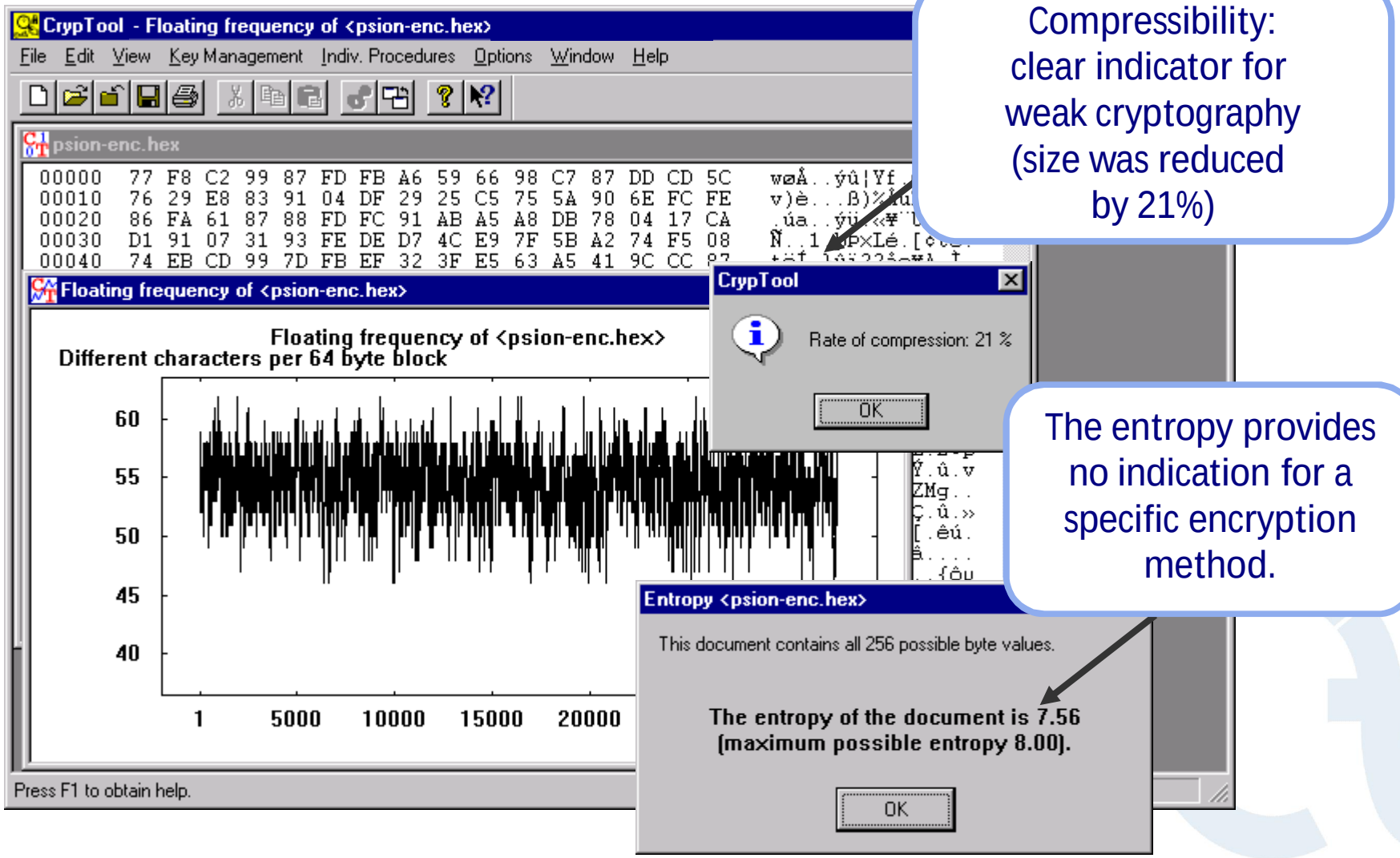
Procedure

- Pre-analysis
  - entropy
  - floating entropy
  - compression test

  } *probably classical encryption algorithm*
- Auto-correlation
- Try out automatic analysis with classical methods

Examples (4)

PSION 5 PDA – determine entropy, compression test

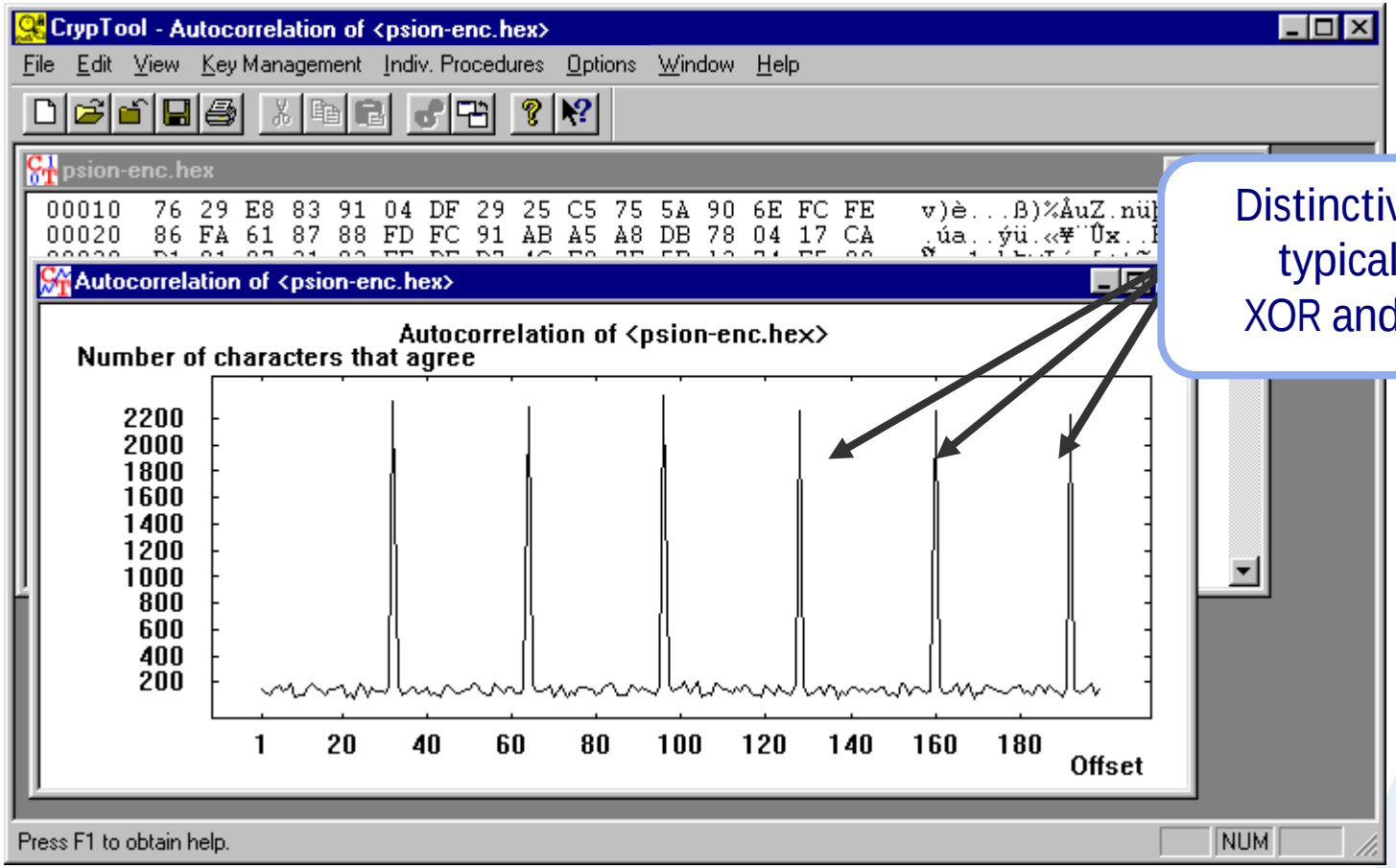


Compressibility: clear indicator for weak cryptography (size was reduced by 21%)

The entropy provides no indication for a specific encryption method.

Examples (4)

PSION 5 PDA – determine auto-correlation



Distinctive comb pattern:
typical for Vigenère,
XOR and binary addition

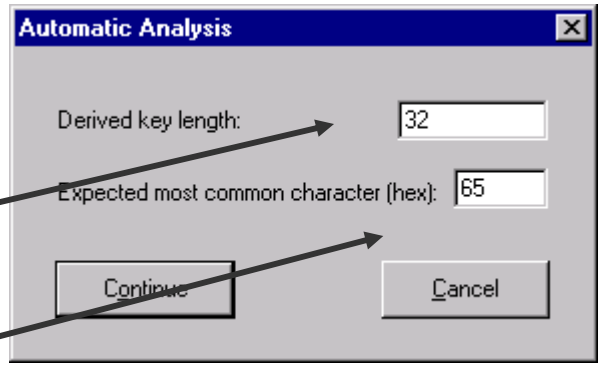
* The encrypted file is available with CrypTool (see CrypTool\examples\psion-enc.hex)

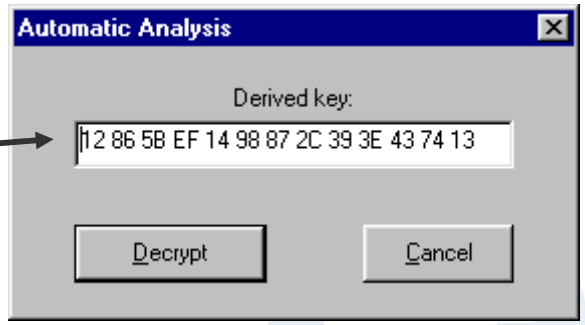
Examples (4)

PSION 5 PDA – automatic analysis

Automatic analysis using

- Vigenère: no success
- XOR: no success
- binary addition
  - CrypTool calculates the key length using auto-correlation: 32 bytes
  - The user can choose which character is expected to occur most frequently: “e” = 0x65 (ASCII code)
  - Analysis calculates the most likely key (based on the assumptions about distribution)
  - Result: good, but not perfect



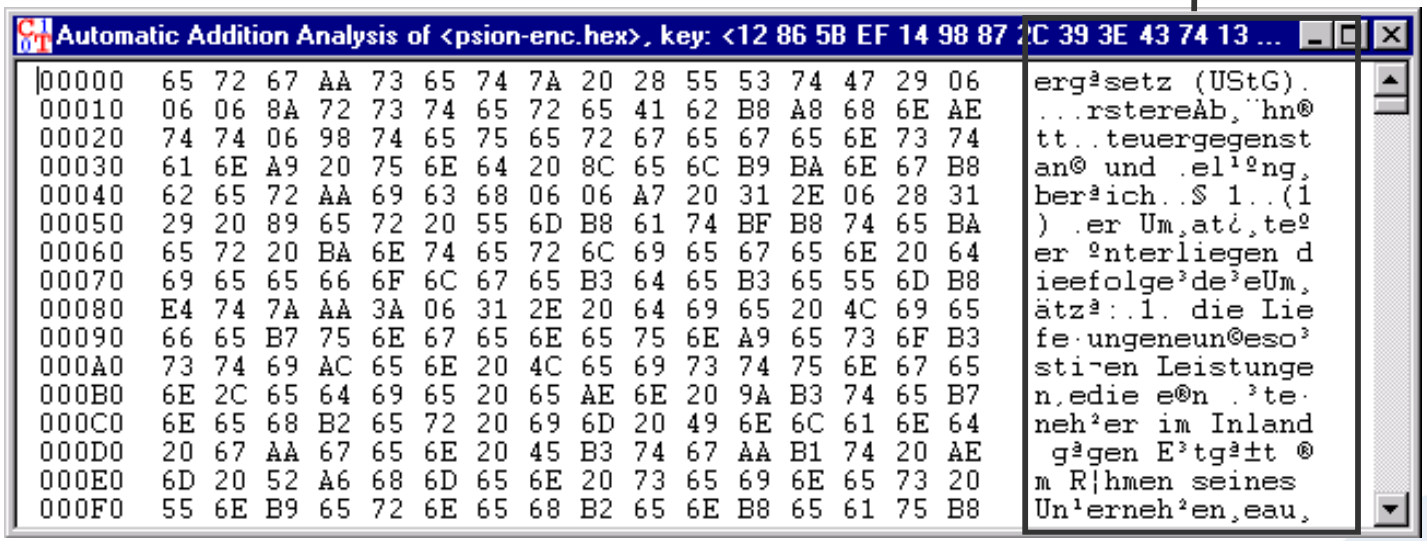


Examples (4)

PSION 5 PDA – results of automatic analysis

Results of automatic analysis with assumption “binary addition”:

- Result is good, but not perfect: 24 out of 32 key bytes correct.
- The key length 32 was correctly determined.



- The password entered was not 32 bytes long.
  → PSION Word derives the actual key from the password.
- Manual post-processing produces the encrypted text (not shown).

Examples (4)

PSION 5 PDA – determining the remaining key bytes

Copy key to clipboard during automatic analysis

In automatic analysis hex dump,

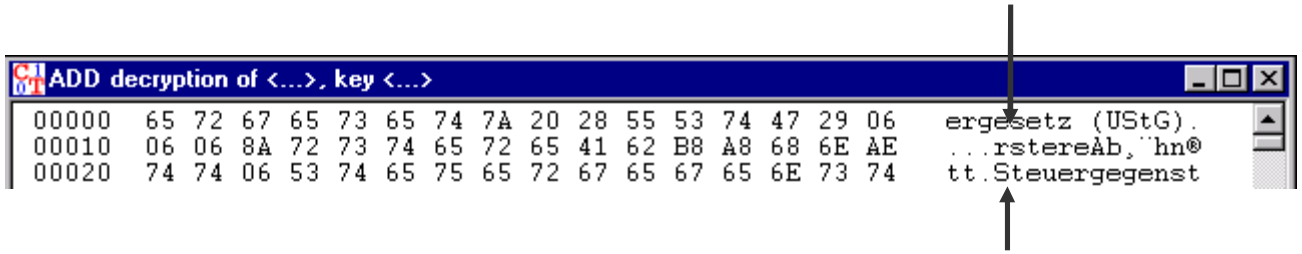
- Determine incorrect byte positions, e.g. 0xAA at position 3
- Guess and write down corresponding correct bytes: „e“ = 0x65

In encrypted initial file hex dump,

- Determine initial bytes from the calculated byte positions: 0x99
- Calculate correct key bytes with CALC.EXE: 0x99 - 0x65 = 0x34

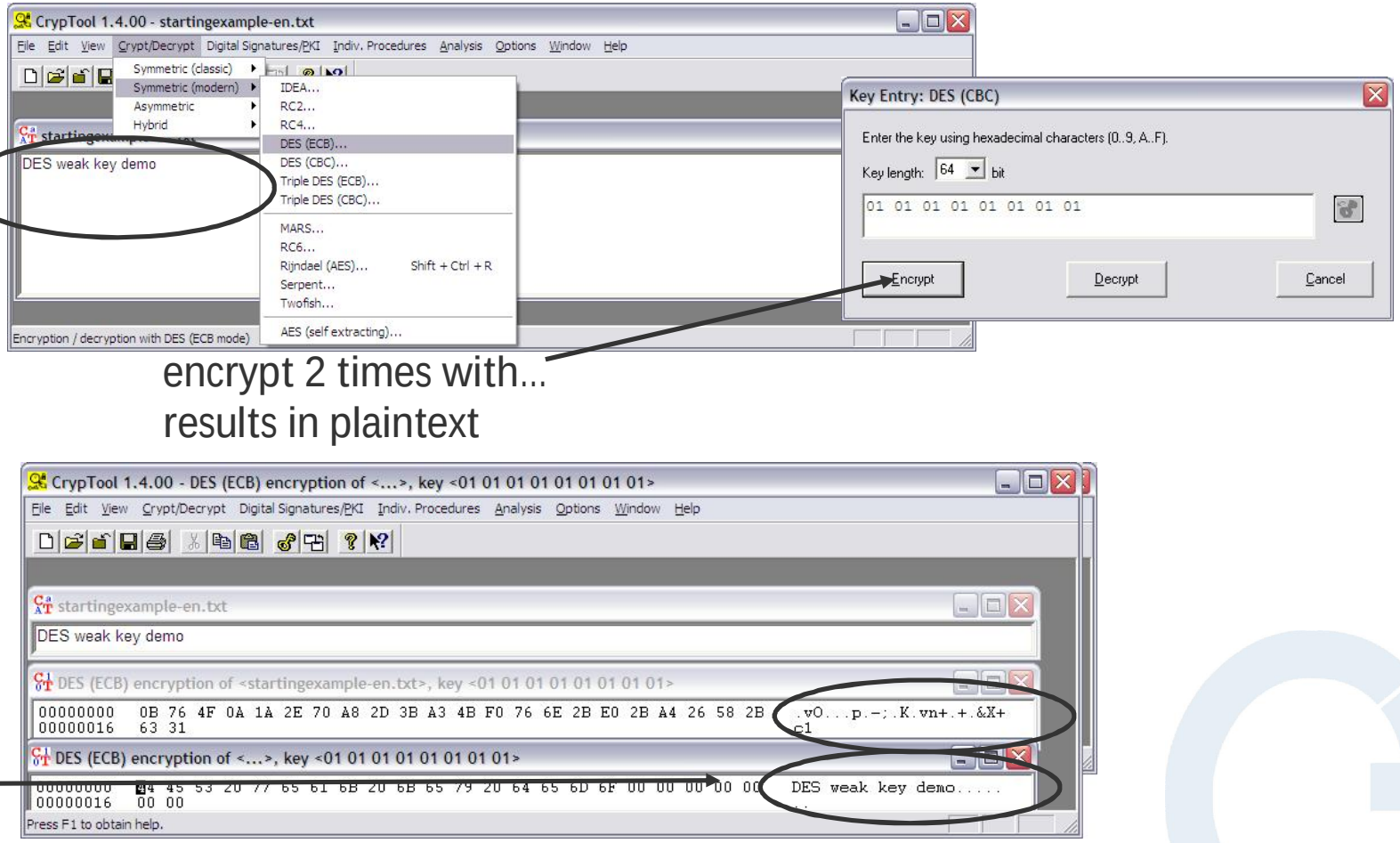
Key from the clipboard

- Correct 12865B341498872C393E43741396A45670235E111E907AB7C0841...
- Decrypt encrypted initial document using binary addition
- Bytes at position 3, 3+32, 3+2*32, ... are now correct



Examples (5)

Weak DES key



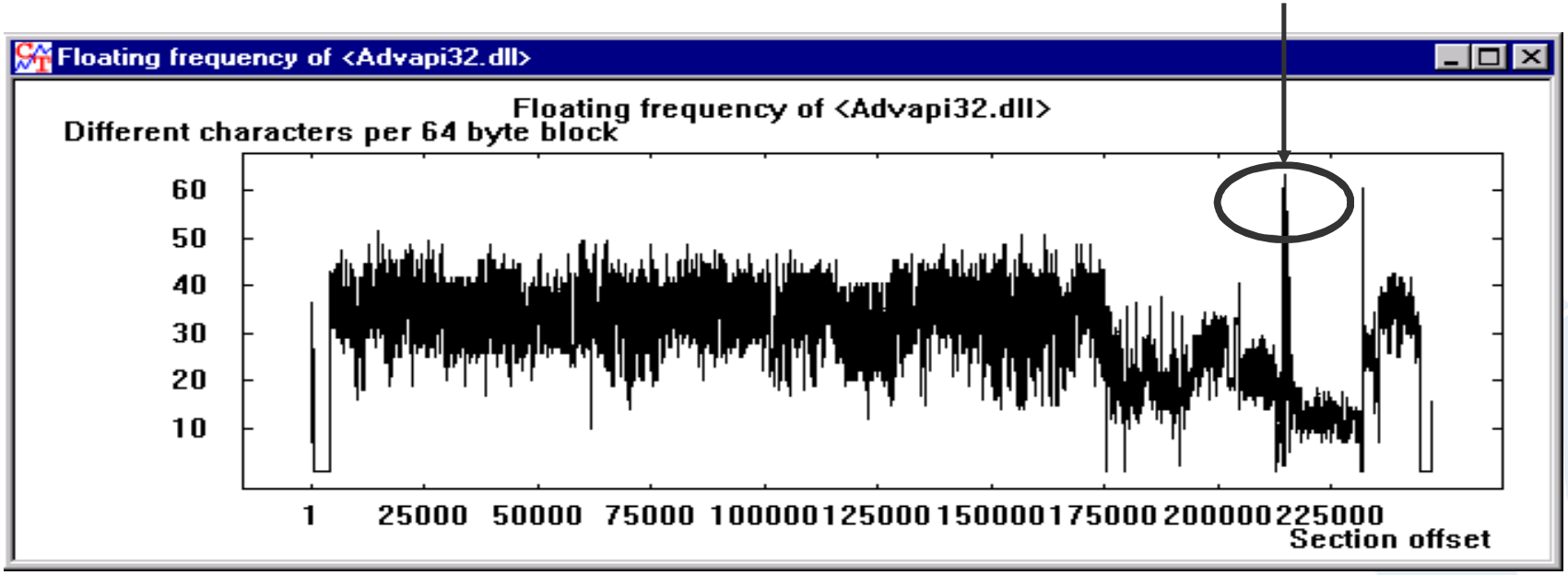
Examples (6)

Locate key material

The function “Floating frequency” is suitable for locating key material and encrypted areas in files.

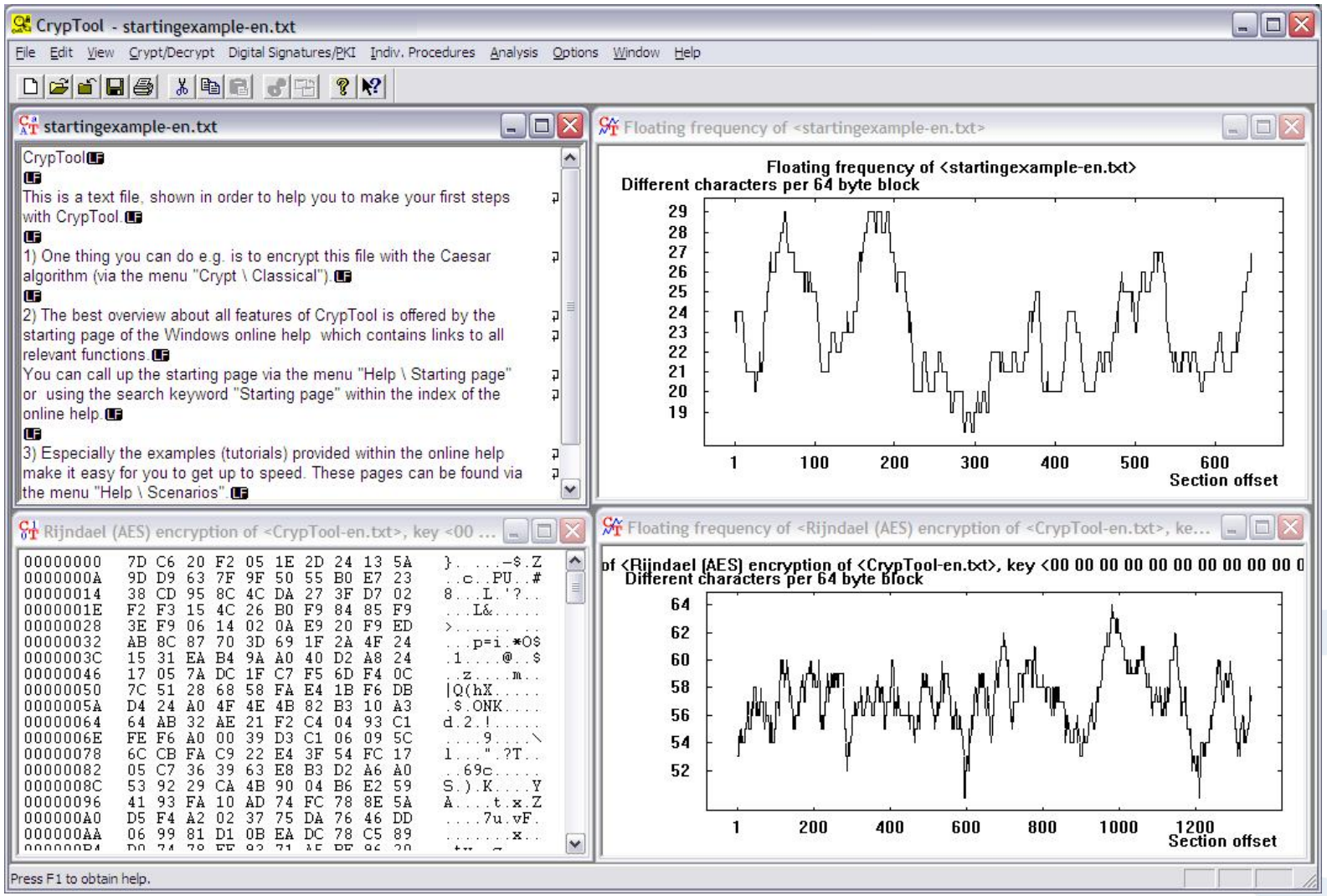
Background:

- Key data is “more random” than text or program code
- Can be recognized as peaks in the “floating frequency”
- Example: the “NSA key” in advapi32.dll (Windows NT)



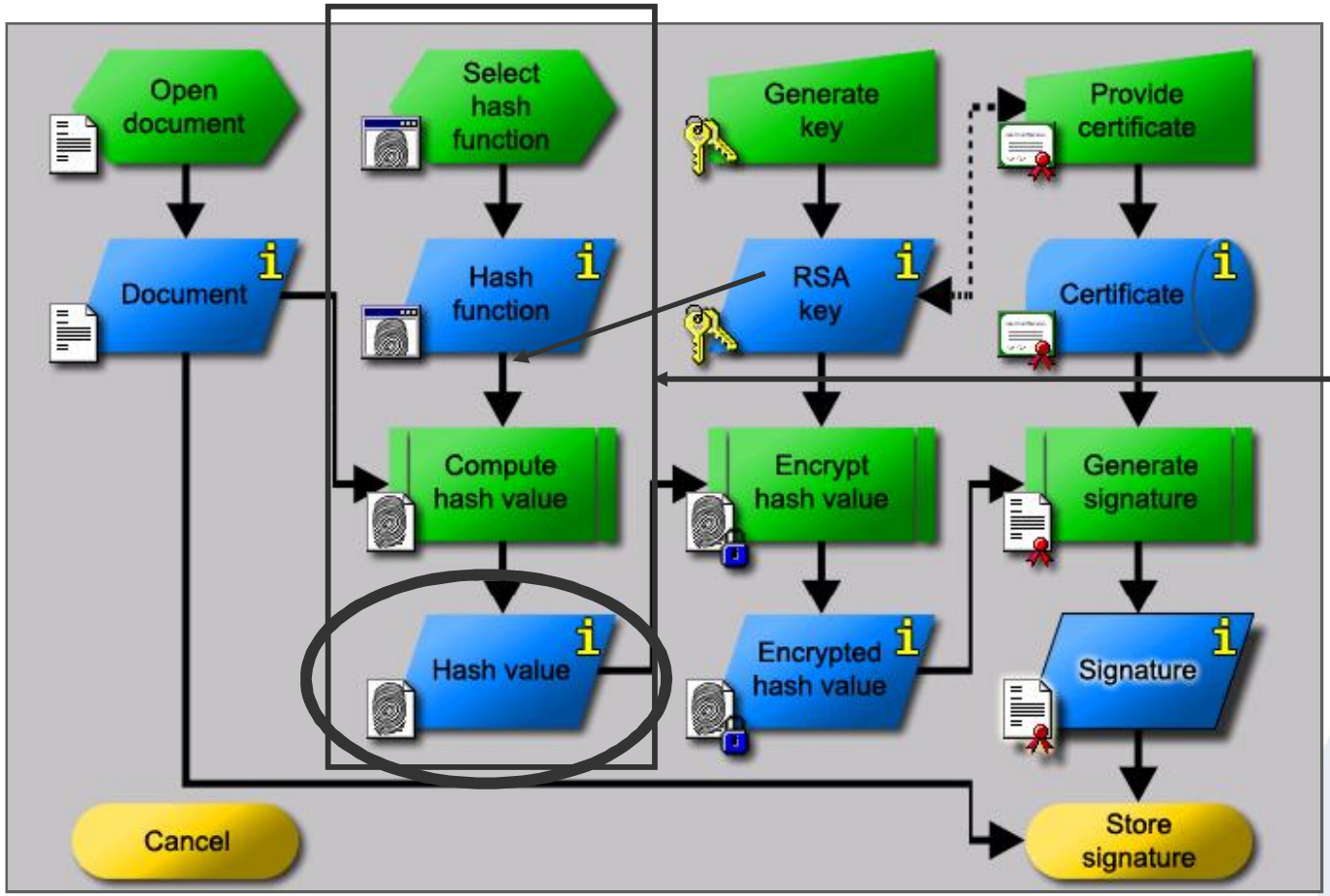
Examples (6)

Comparison on floating frequency with other files



Examples (7)

Attack on digital signature



Attack:

Find two messages with the same hash value !

Menu: „Analysis“ \ „Hash“ \ „Attack on the Hash Value of the Digital Signature“

Examples (7)

Attack on digital signature – idea (I)

Attack on the digital signature of an ASCII text based on hash collision search.

Idea:

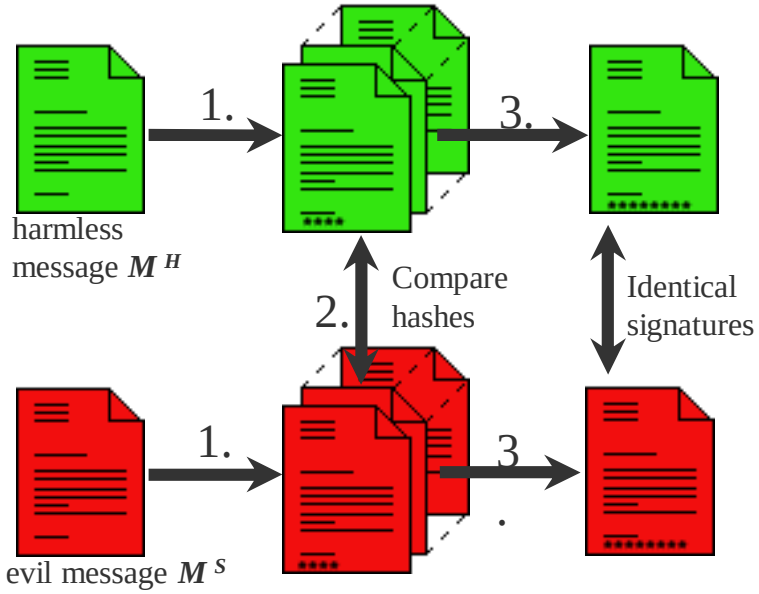
- ASCII texts can be modified by changing/inserting non-printable characters, without changing the visible content
- Modify two texts in parallel until a hash collision is found
- Exploit the birthday paradox (birthday attack)
- Generic attack applicable to all hash functions
- Can be run in parallel on many machines (not implemented)
- Implemented in CrypTool as part of the bachelor thesis "*Methods and Tools for Attacks on Digital Signatures"* (German), 2003.

Concepts:

- Mappings
- Modified Floyd algorithm (constant memory consumption)

Examples (7)

Attack on digital signature – idea (II)



1. Modification: starting from a message M create N different messages $M_1, ..., M_N$ with the same "content" as M.
2. Search: find modified messages M_i^H and M_j^S with the same hash value.
3. Attack: the signatures of those two documents M_i^H and M_j^S are the same.

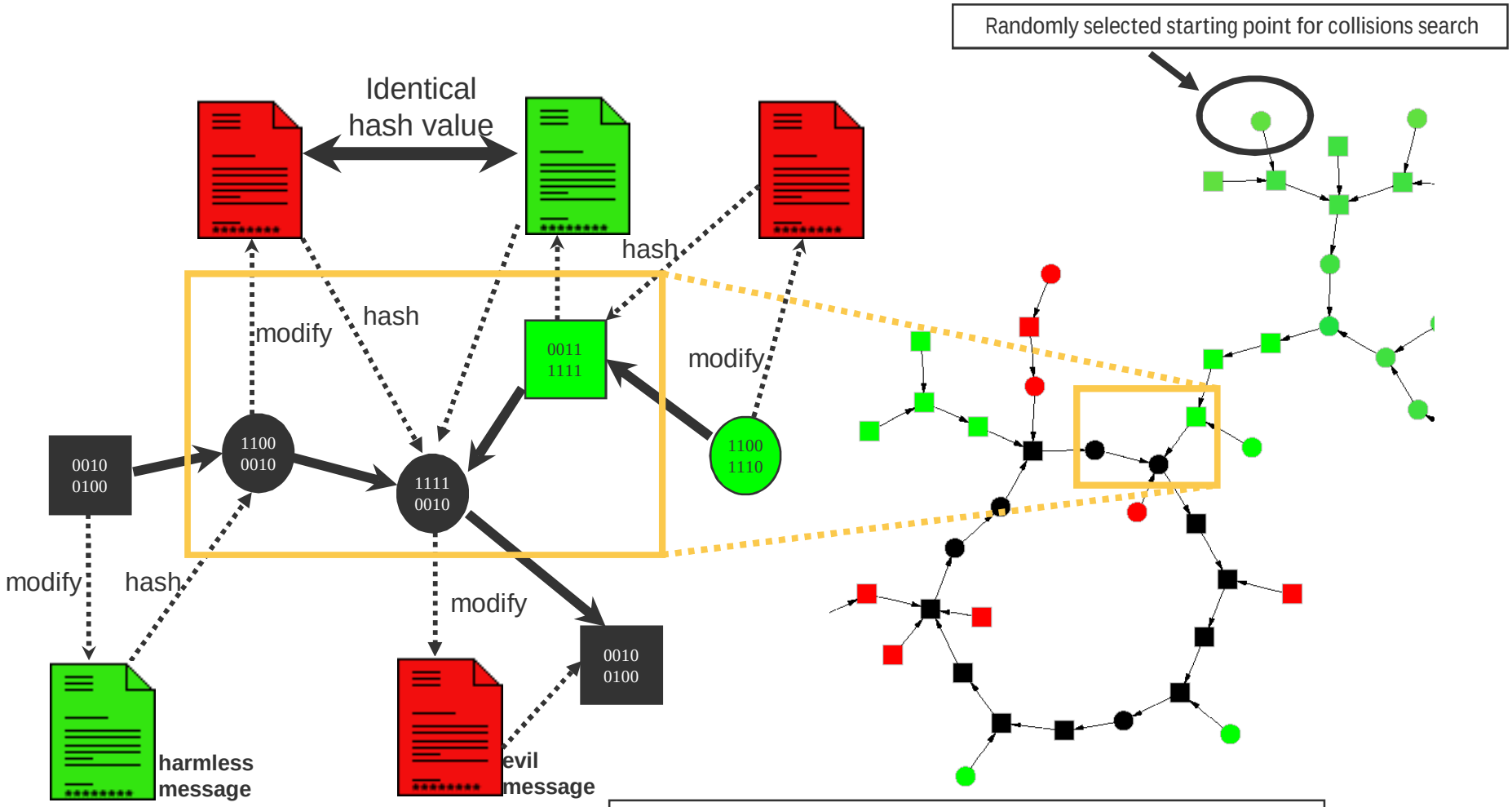
We know from the birthday paradox that for hash values of bit length n:

- search collision between M^H and $M_1^S, ..., M_N^S$: $N \approx 2^n$
- search collision between $M_1^H, ..., M_N^H$ and $M_1^S, ..., M_N^S$: $N \approx 2^{n/2}$

Estimated number of generated messages in order to find a hash collision.

Locate Hash Collisions (1)

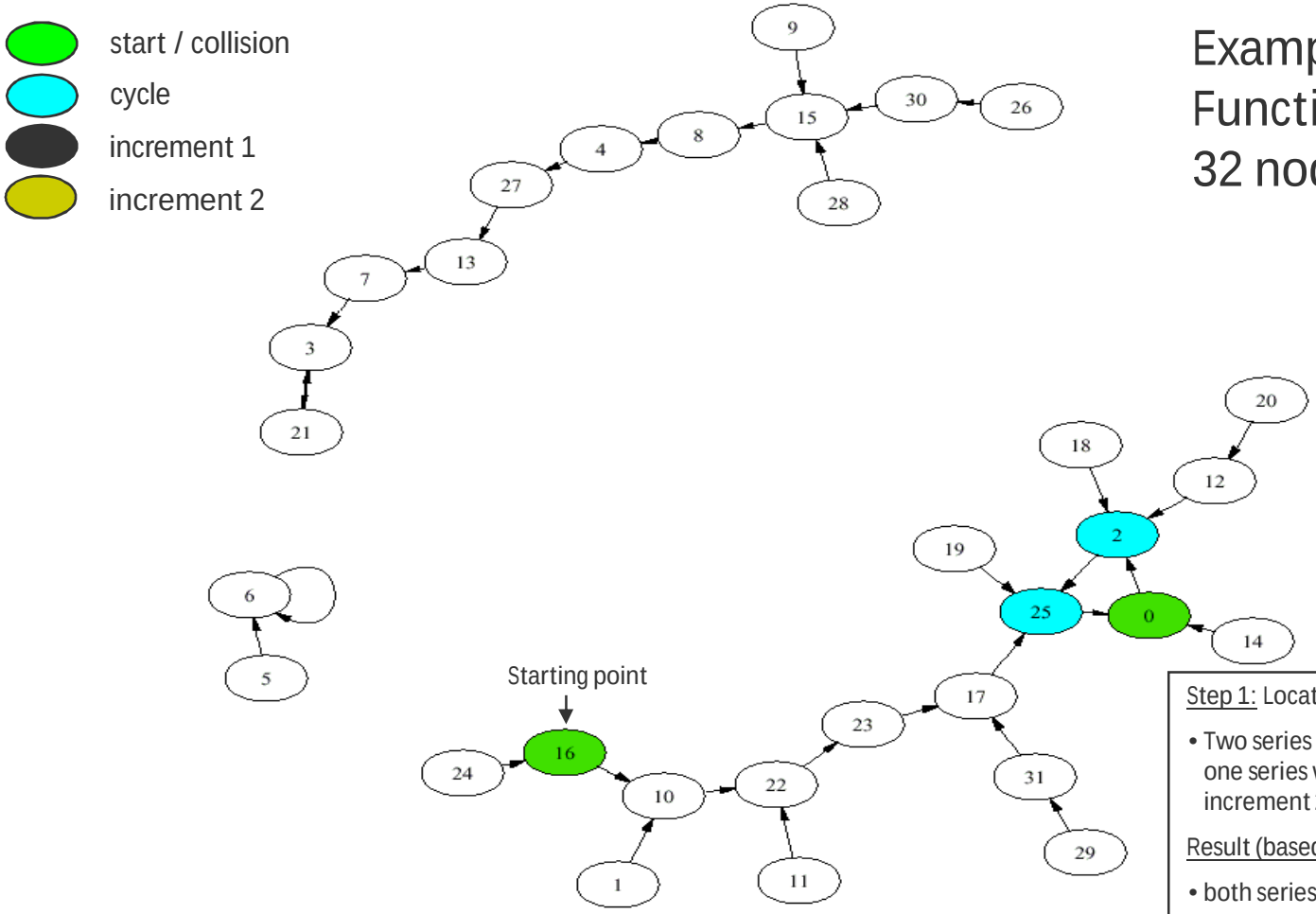
Mapping via text modifications



- green / red: path from a tree to the cycle – this can lead to a useful or useless collision.
- square / round: hash value has even / odd parity
- black: all nodes within the cycle

Locate Hash Collisions (2)

Floyd Algorithm: meet within the cycle



Example:
Function graph with
32 nodes

Step 1: Locate matching point within cycle:

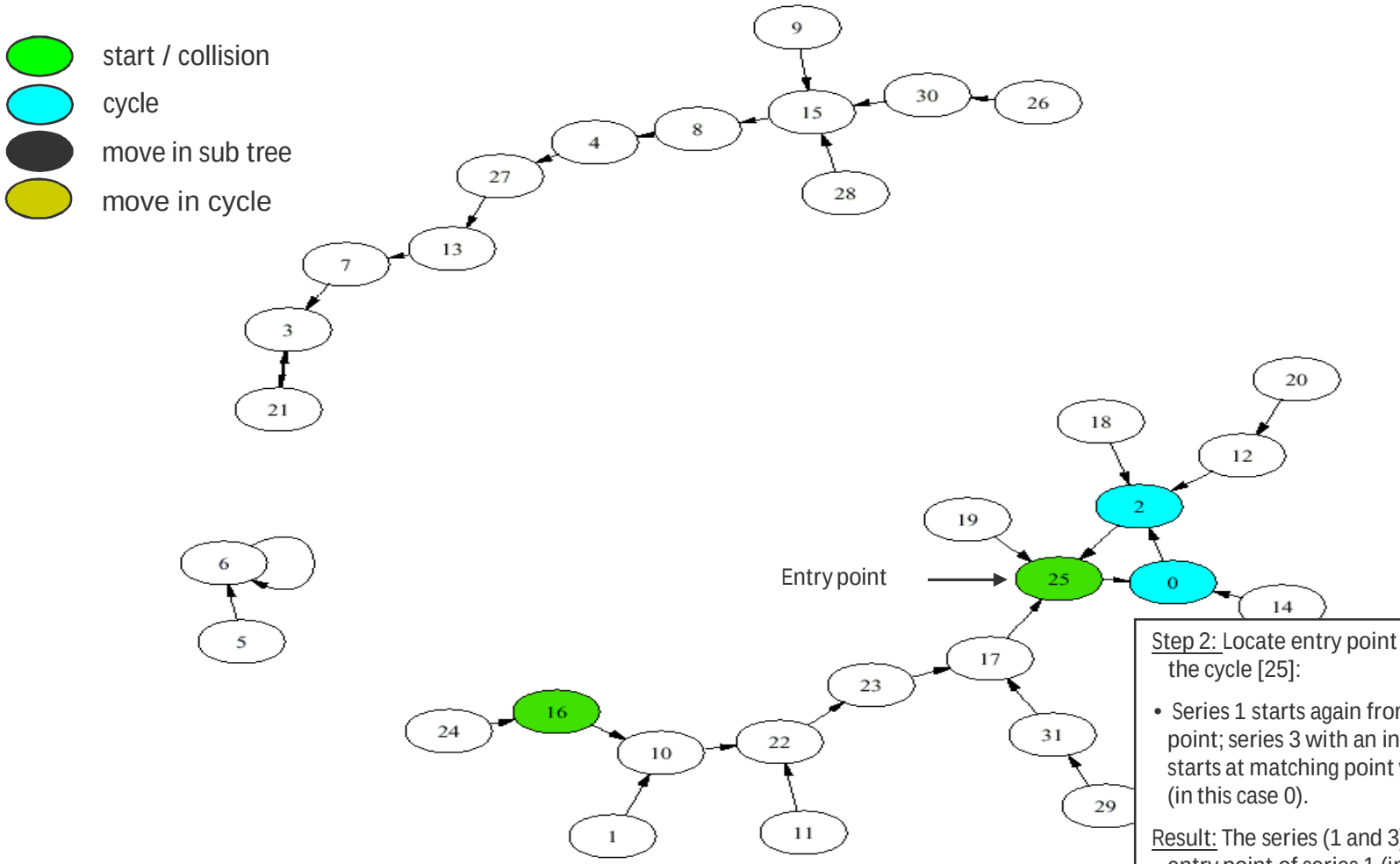
- Two series with identical starting point [16]: one series with increment 1, the other with increment 2.

Result (based on graph theory):

- both series always end up in a cycle.
- both series match in a node within the cycle (in this case 0).

Locate Hash Collisions (3)

Step into cycle (Extension of Floyd): find entry point



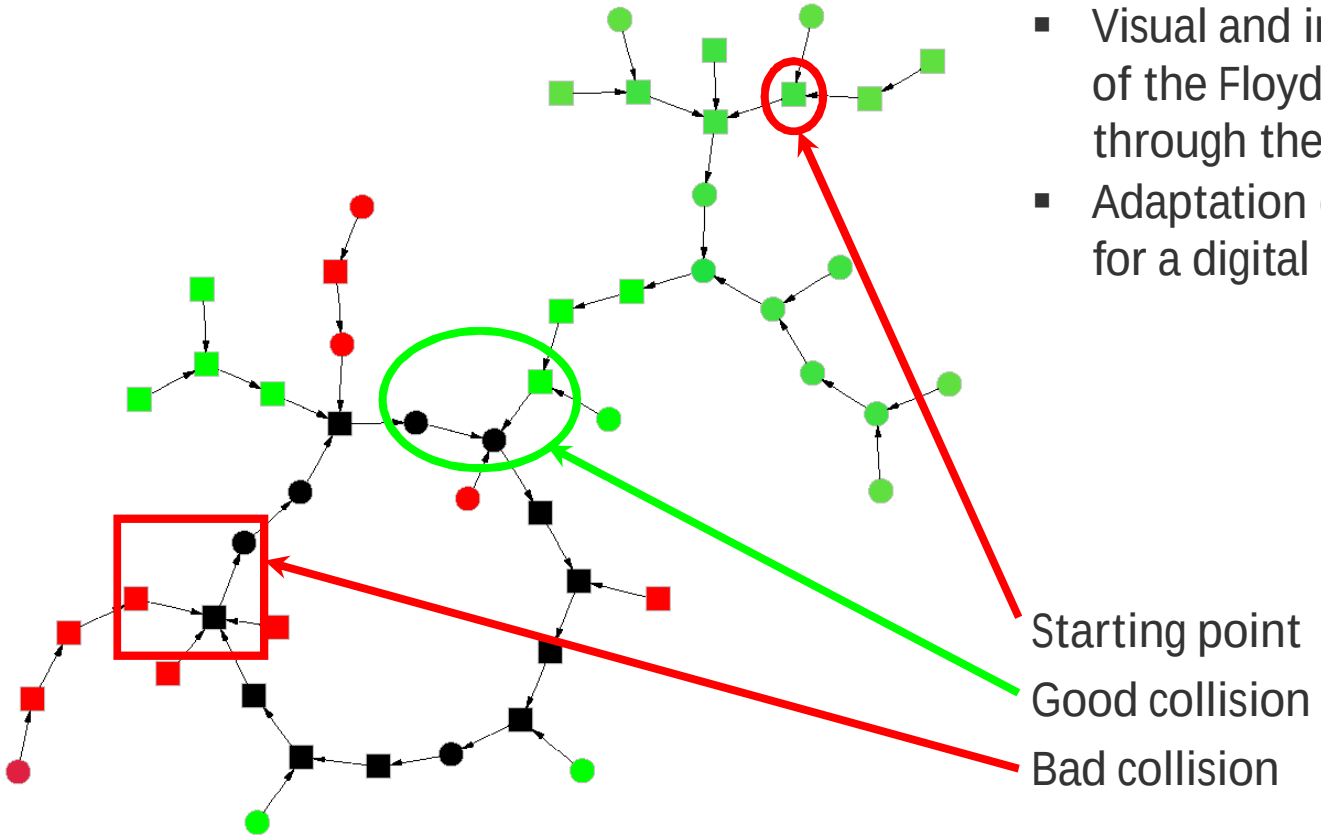
Step 2: Locate entry point of series 1 in the cycle [25]:

- Series 1 starts again from starting point; series 3 with an increment of 1 starts at matching point within the cycle (in this case 0).

Result: The series (1 and 3) match in cycle entry point of series 1 (in this case 25)

- The predecessors (in this case 17 and 2) result in a hash collision.

Birthday Paradox Attack on Digital Signature



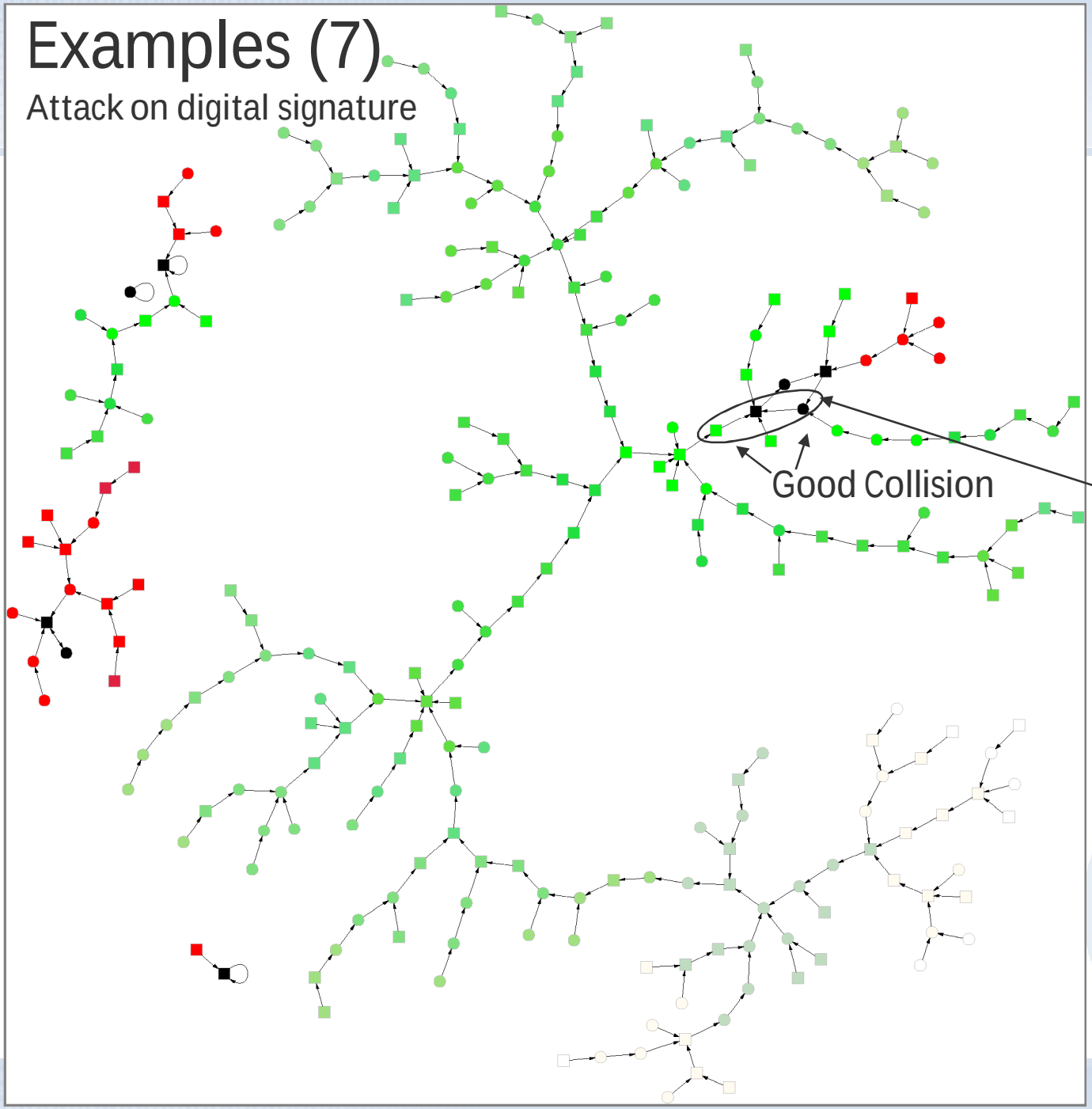
Examination of Floyd algorithm

- Visual and interactive presentation of the Floyd algorithm („Moving through the mapping" into a cycle).
- Adaptation of the Floyd algorithm for a digital signature attack.

*The Floyd algorithm is implemented in CrypTool, but the visualization of the algorithm is not yet implemented.

Examples (7)

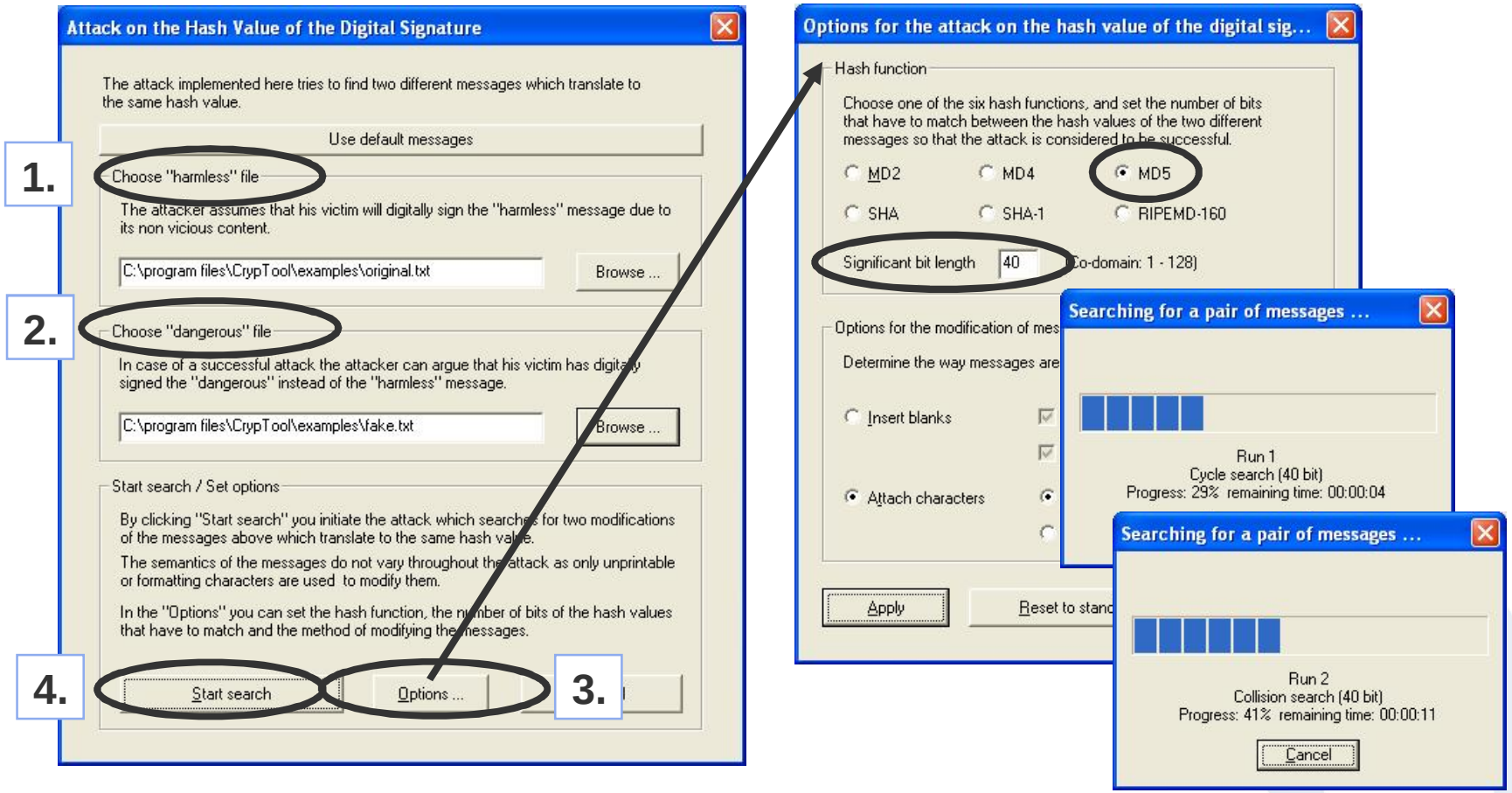
Attack on digital signature



An example for a “good” Mapping (nearly all nodes are green).
In this graph almost all nodes belong to a big tree, which leads into the cycle with an even hash value and where the entry point predecessor within the cycle is odd.
That means that the attacker finds a useful collision for nearly all starting points.

Examples (7)

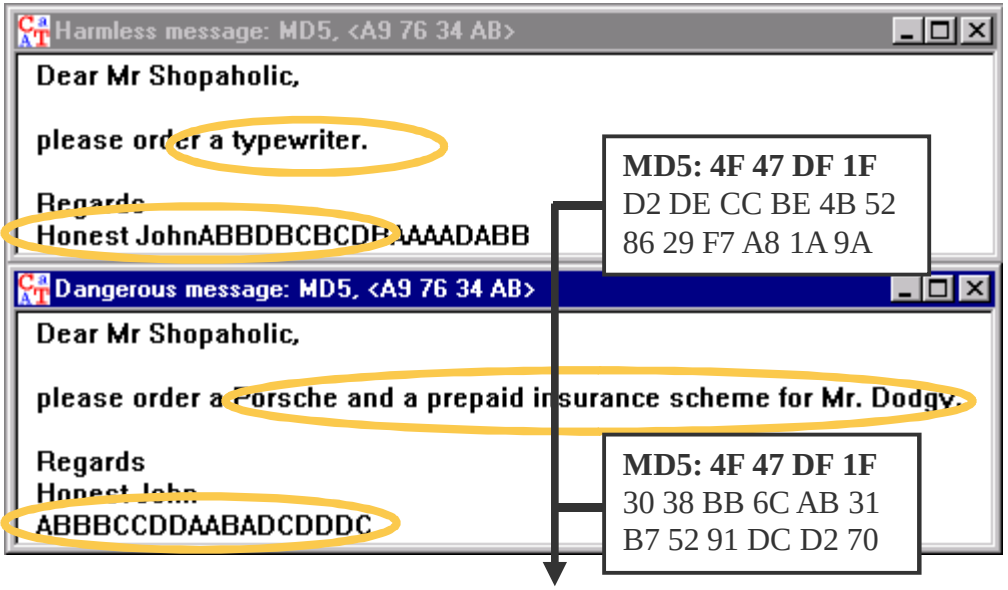
Attack on digital signature: attack



Menu: „Analysis“ \ „Hash“ \ „Attack on the Hash Value of the Digital Signature“

Examples (7)

Attack on digital signature: results



The first 32 bits of the hash values are identical.

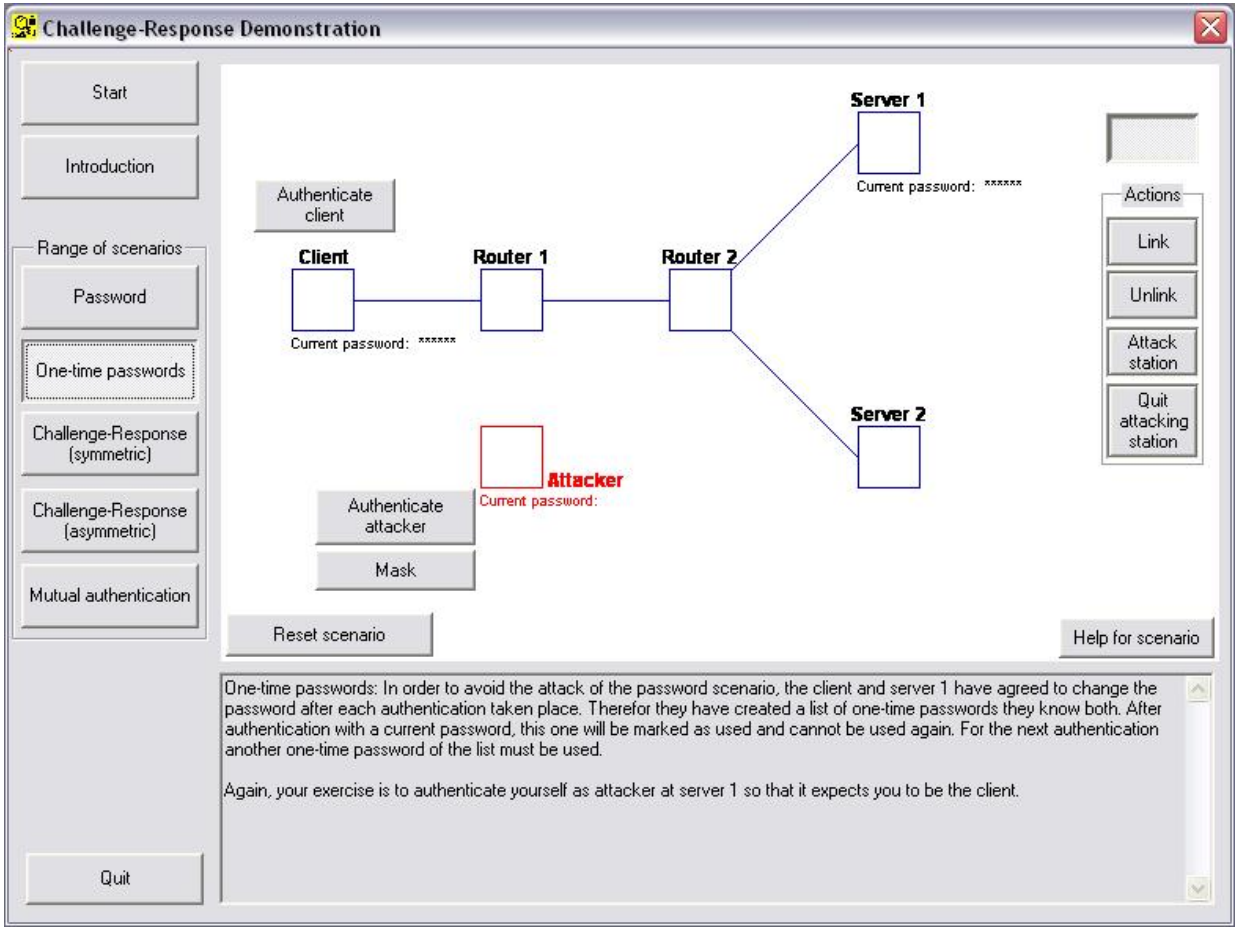
Experimental results

- 72 Bit *partial collision* (equality of the first 72 hash value bits) were found in a couple of days on a single PC.
- Signatures using hash values of up to 128 bit can be attacked today using massive parallel search!
- Use hash values of at least 160 bit length.

In addition to the interactive handling:
Automated offline feature in CrypTool: Execute and log the results for entire sets of parameter configurations. Available through command line execution of CrypTool.

Examples (8)

Authentication in a client-server environment

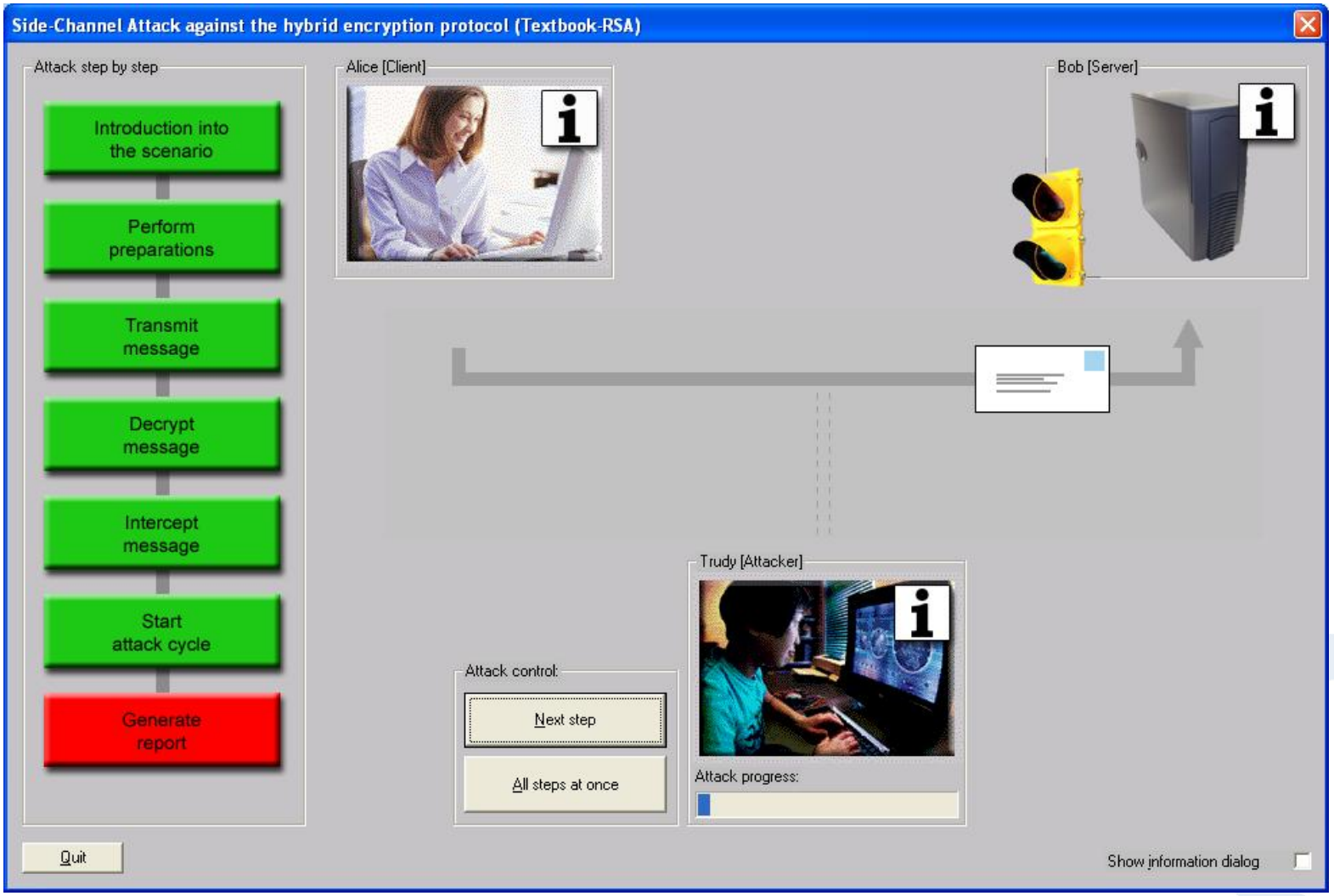


Menu: „Indiv. Procedures“ \ „Protocols“ \ „Network Authentication“

- Interactive demo for different authentication methods.
- Defined opportunities of the attacker.
- You can play the role of an attacker.
- Learning effect: Only mutual authentication is secure.

Examples (9)

Demonstration of a side-channel attack (on a hybrid encryption protocol)



Menu: „Analysis“ \ „Asymmetric Encryption” \ „Side-Channel Attack on Textbook-RSA”

Examples (9)

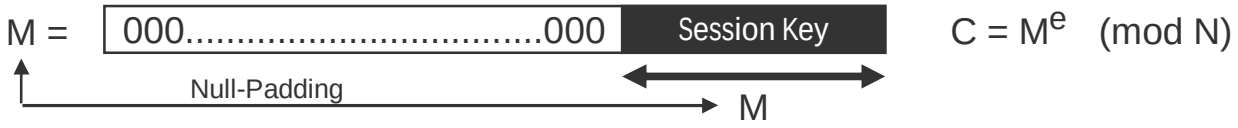
Idea for this side channel attack

Ulrich Kühn "*Side-channel attacks on textbook RSA and ElGamal encryption*", 2003

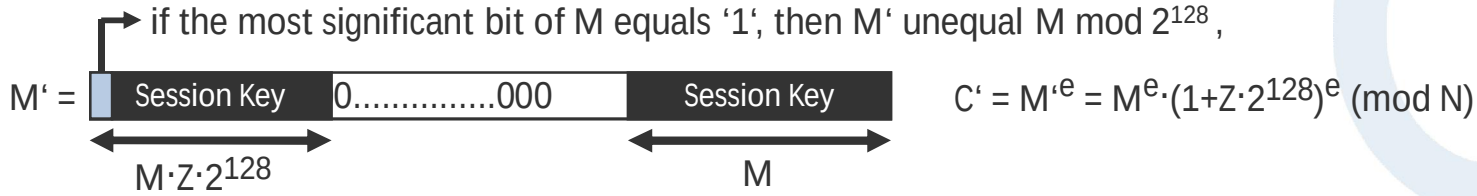
Prerequisites:

- RSA encryption: $C = M^e \pmod N$ and decryption: $M = C^d \bmod N$.
- 128-Bit session keys (in M) are „word book encoded" (null padding).
- The server knows the secret key d and
  - uses after decryption the 128 least significant bits only (no validation of zero padding bits) (that means the server does not recognize if there is something other than zero).
  - Prompts an error message, if the encryption attempt results in a wrong session key (decrypted text can not be interpreted by the server). In all other cases there will be no message.

Idea for attack: Approximation for Z out of the equation $N = M * Z$ per $M = ||N/Z||$

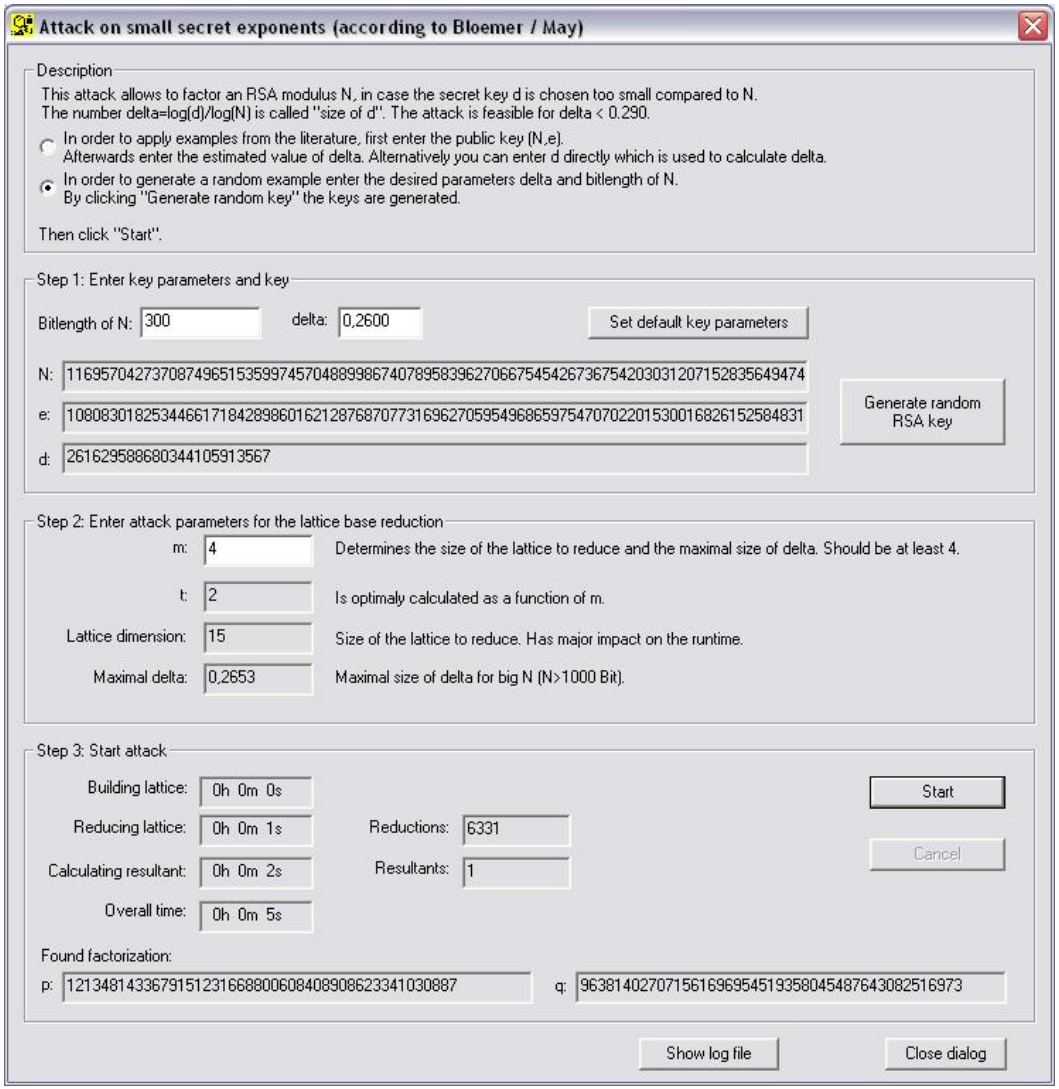


All bit positions for Z are successively calculated: For every step one gets 1 further bit. The attacker modifies C to C' (see below). If a bit overflow occurs while calculating M' on the server (recipient), the server sends an error message. Based on this information the attacker gets a bit for Z.



Examples (10)

Mathematics: Attacks on RSA using lattice reduction



- Shows how the parameters of the RSA method have to be chosen, so that the algorithm resists the lattice reduction attacks described in current literature.
- 3 variants
  1. The secret exponent d is too small in comparison to N.
  2. One of the factors of N is partially known.
  3. A part of the plaintext is known.
- These assumptions are realistic

Menu: „Analysis“ \
„Asymmetric Encryption” \
„Lattice Based Attacks on RSA“ \ ...

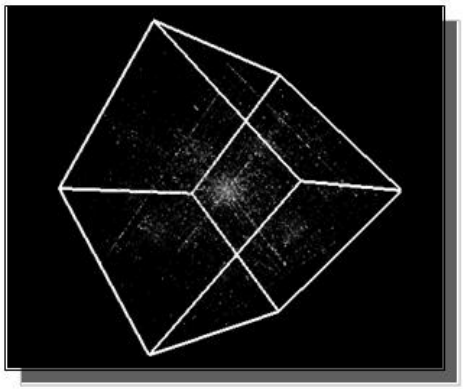
Examples (11)

Random analysis with 3-D visualization

3-D visualization for random analysis

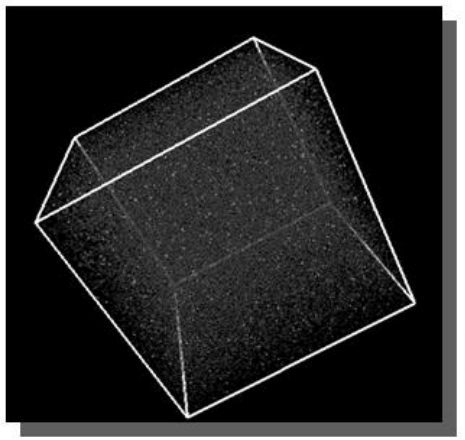
Example 1

- Open an arbitrary file (e.g. report in Word or PowerPoint presentation)
- It is recommended to select a file with at least 100 kB
- 3-D analysis using menu: „Analysis“ \ „Analyse Randomness“ \ „3-D Visualization“
- Result: structures are easily recognisable



Example 2

- Generation of random numbers: „Indiv. Procedures“ \ “Tools” \ „Generate Random Numbers“
- It is recommended to generate at least 100,000 random bytes
- 3-D analysis using menu: „Analysis“ \ „Analyse Randomness“ \ „3-D Visualization”
- Result: uniform distribution (no structures are recognisable)

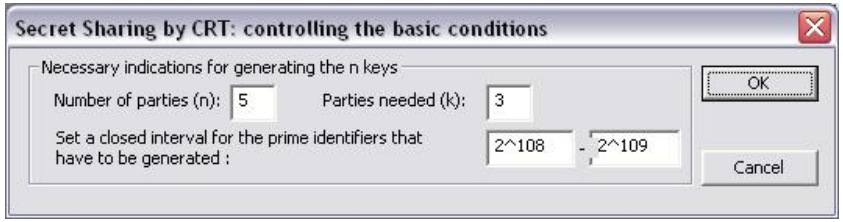


Examples (12)

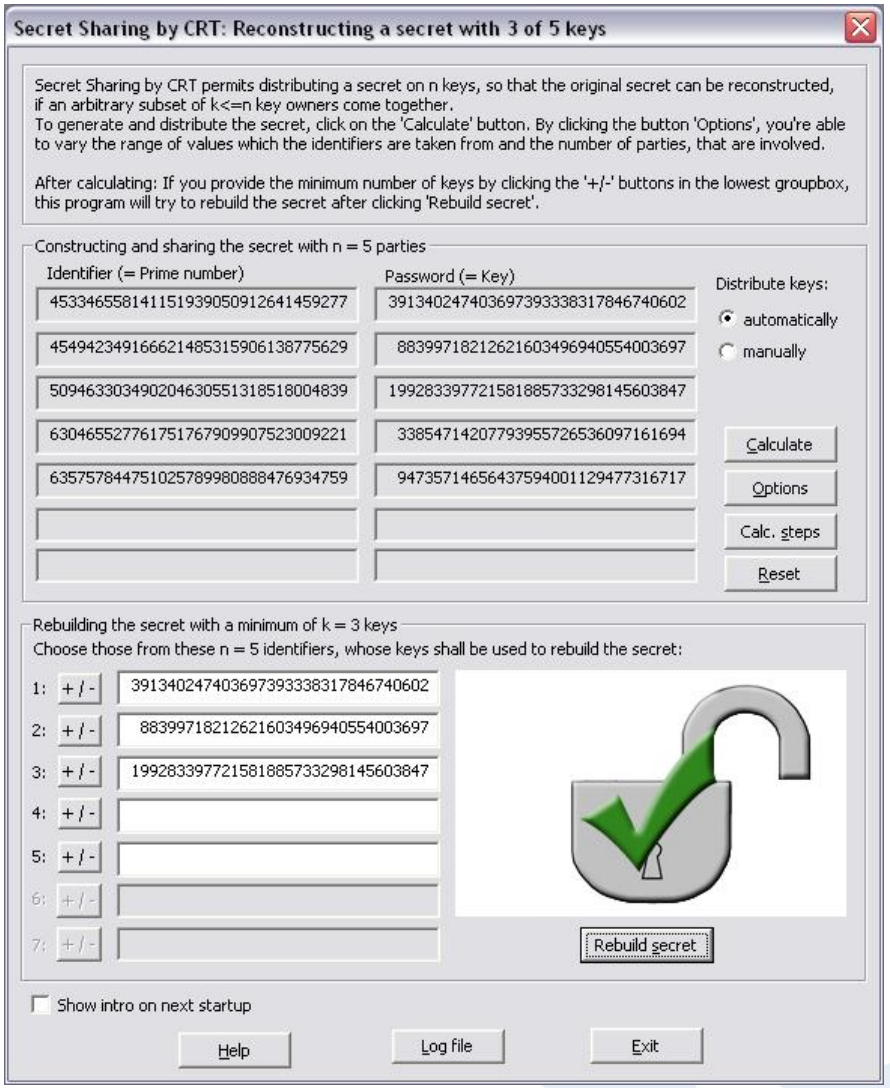
Secret sharing with CRT – implementation of the Chinese remainder theorem (CRT)

Secret sharing example (1):

- Problem:
  - 5 people get a single key
  - To gain access at least 3 of the 5 people have to be present
- Menu: „Indiv. Procedures“ \ „Chinese Remainder Theorem Applications“ \ „Secret Sharing by CRT“
- „Options“ allows to configure more details of the method.



- „Calc. steps“ shows all steps to generate the key.

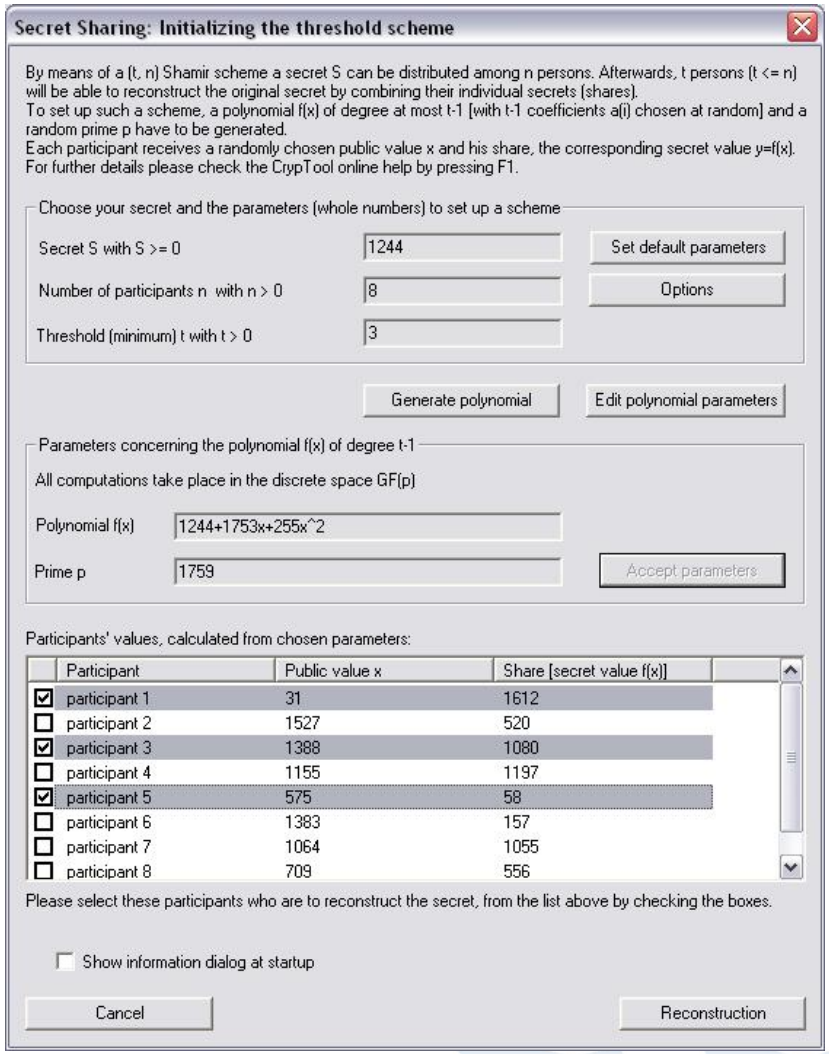


Examples (12)

Shamir secret sharing

Secret sharing example (2)

- Problem
  - A secret value should be split for n people.
  - t out of n people are required to restore the secret value K.
  - (t, n) threshold scheme
- Menu: „Indiv. Procedures“ \ „Secret Sharing Demonstration (Shamir)“
  1. Enter the secret K, number of persons n and threshold t
  2. Generate polynom
  3. Use parameters
- Using „Reconstruction“ the secret can be restored

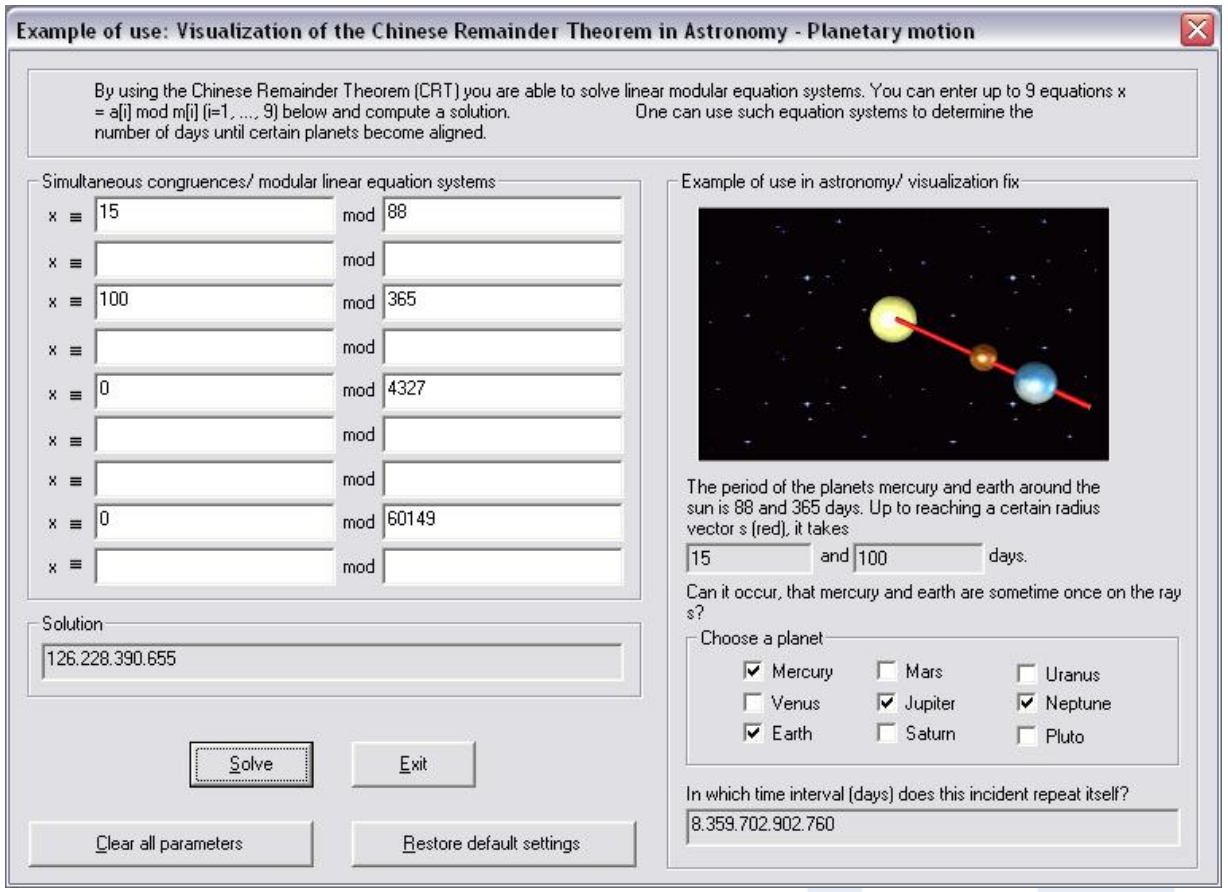


Examples (13)

Implementation of CRT to solve linear modular equation systems

Scenario in astronomy

- How long does it take until a given number of planets (with different rotation times) to become aligned?
- The result is a linear modular equation system, that can be solved with the Chinese remainder theorem (CRT).
- In this demo you can enter up to 9 equations and compute a solution using the CRT.



Menu: „ Indiv. Procedures“ \ „Chinese Remainder Theorem Applications“ \ „Astronomy and Planetary Motion“

Examples (14)

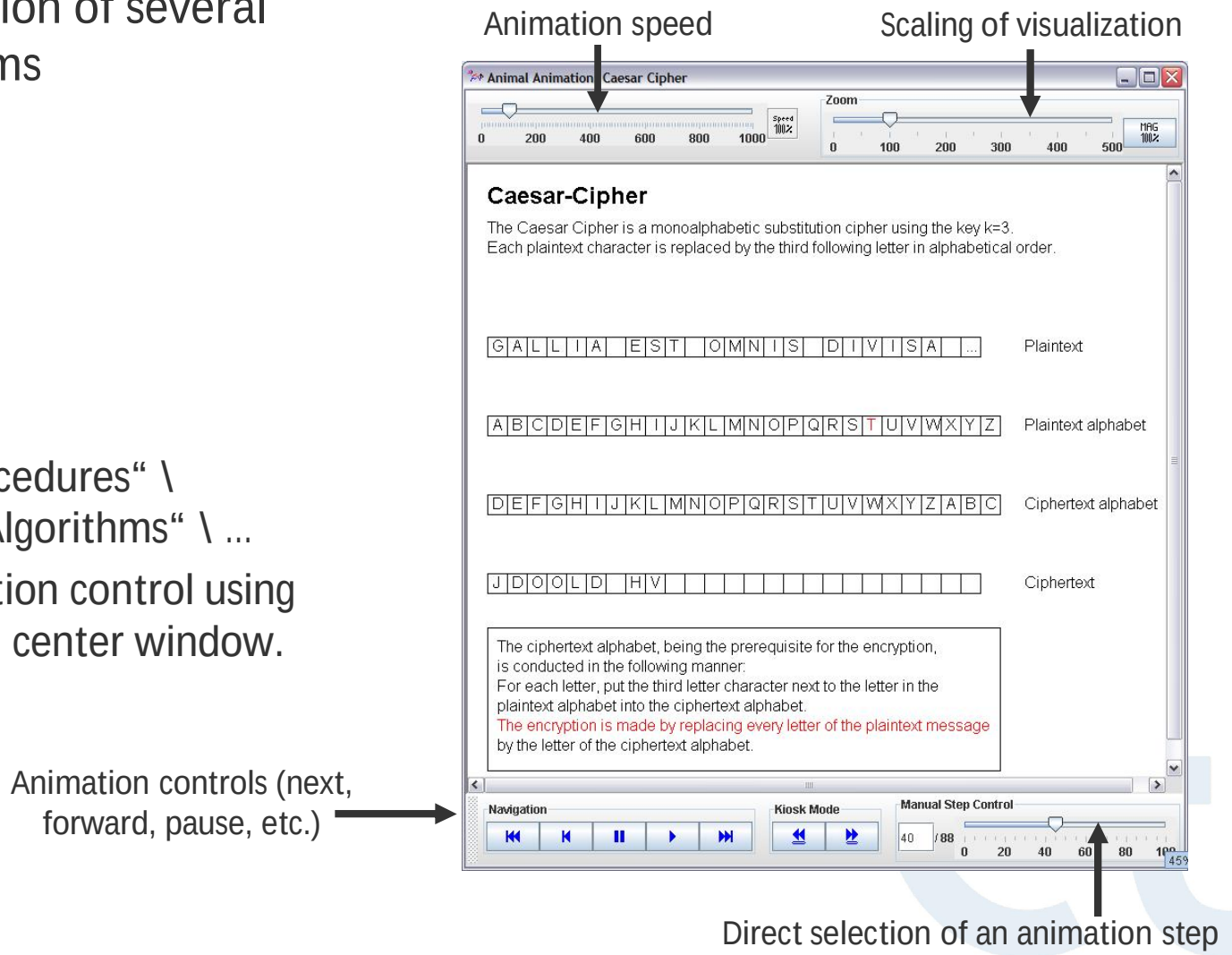
Visualization of symmetric encryption methods using ANIMAL (1)

Animated visualization of several symmetric algorithms

- Caesar
- Vigenère
- Nihilist
- DES

CrypTool

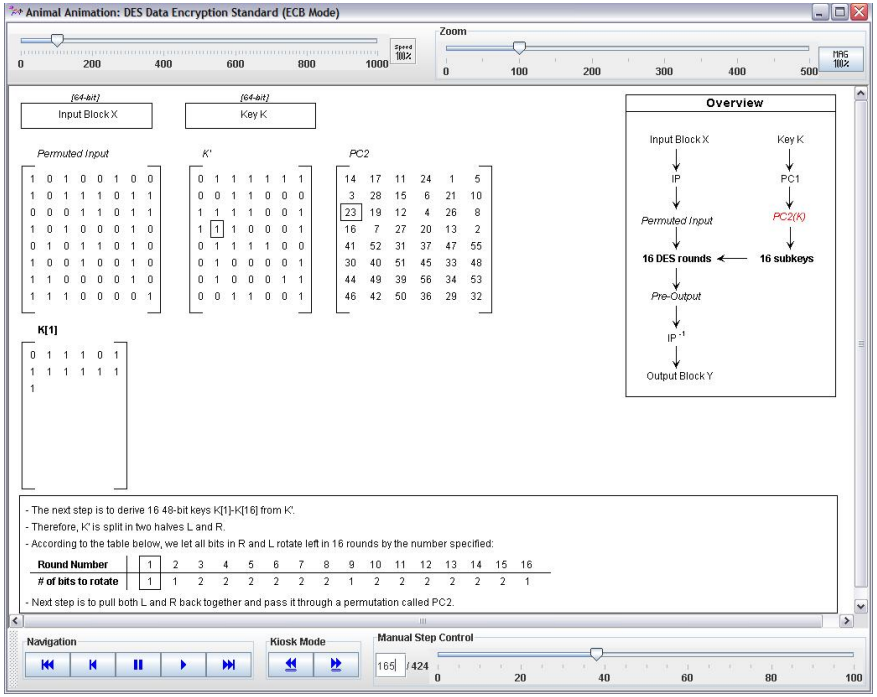
- Menu: „Indiv. Procedures“ \ „Visualization of Algorithms“ \ ...
- Interactive animation control using integrated control center window.



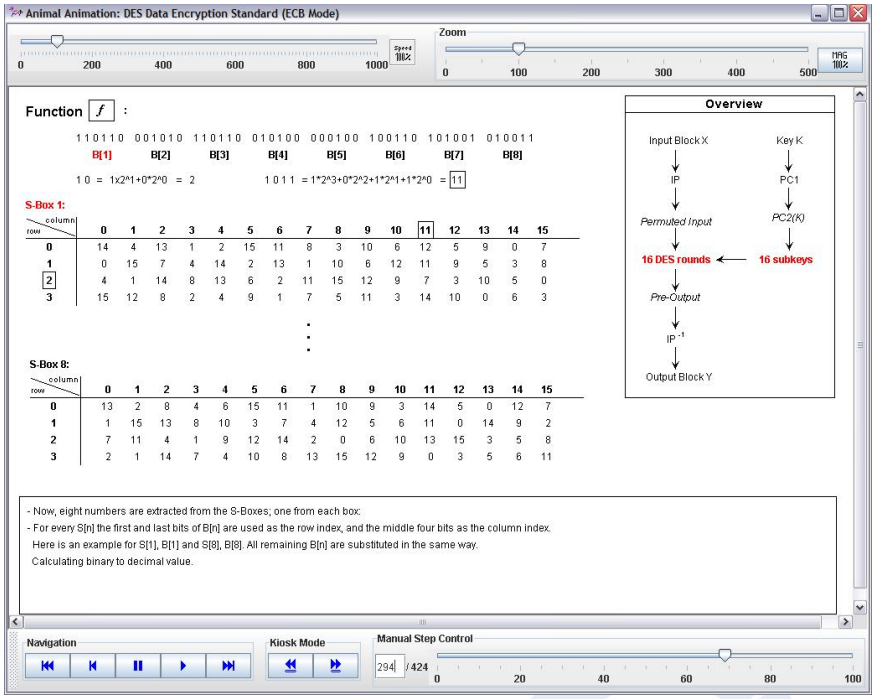
Examples (14)

Visualization of symmetric encryption methods using ANIMAL (2)

Visualization of DES encryption



After the permutation of the input block using the initialisation vector IV the key K is being permuted with PC1 and PC2.



The core function *f* of DES, which links the right half of the block R_{i-1} with the partial key K_i.

Examples (15)

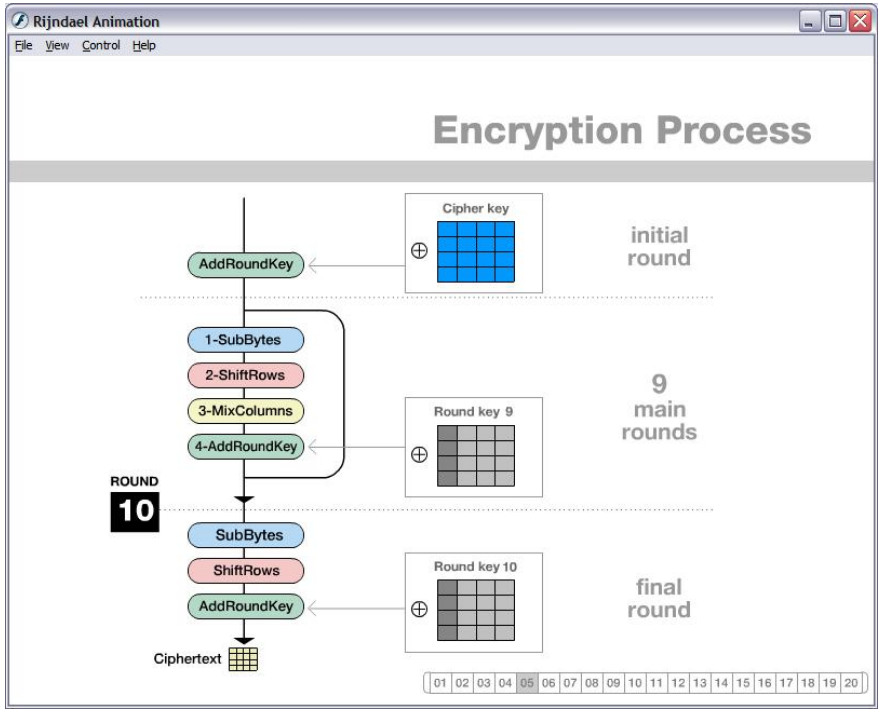
Visualizations of AES (Rijndael cipher) – using Flash

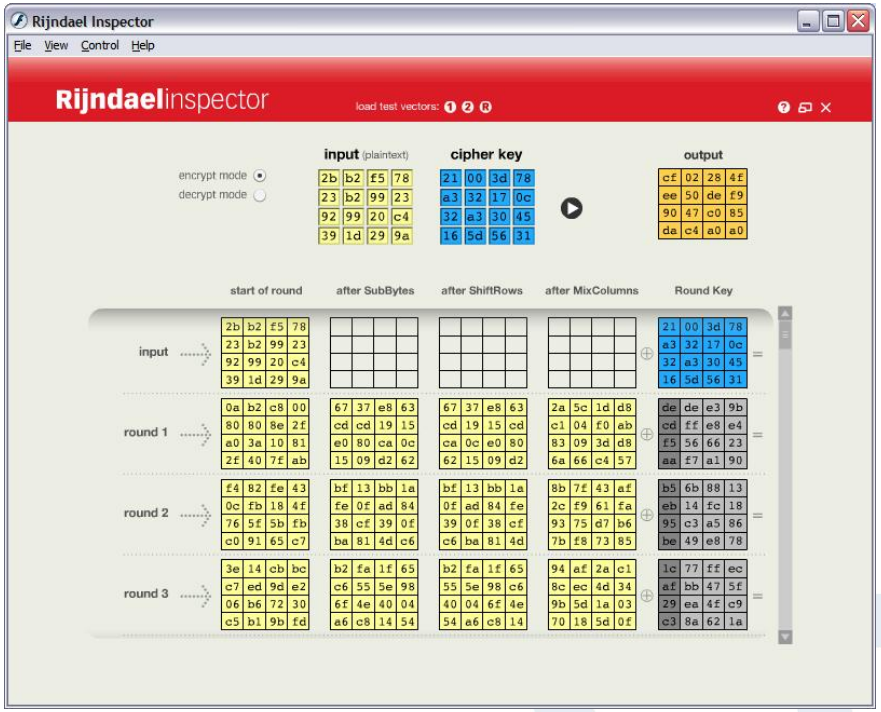
Rijndael Animation (the Rijndael cipher was the winner of the AES submission)

- Shows the encryption processes of each round (using fixed initial data)

Rijndael Inspector

- Encryption process for testing (using your own data, showing each step's matrices)





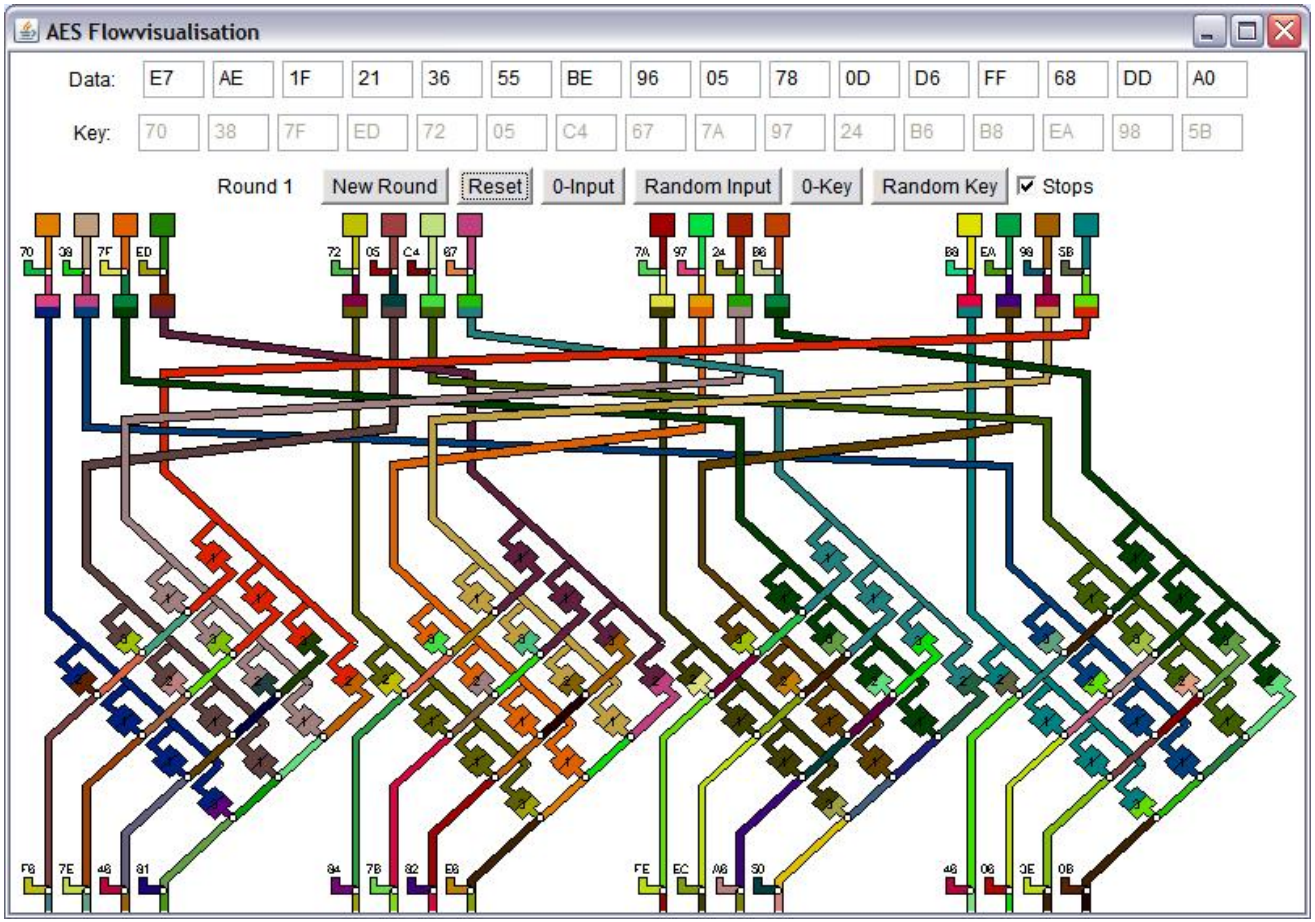
Menu: "Indiv. Procedures" \ "Visualization of Algorithms" \ "AES" \ "Rijndael Animation" or "Rijndael Inspector"

Examples (15)

Flow visualization of AES (Rijndael Cipher) – using Java

Rijndael Flow Visualization

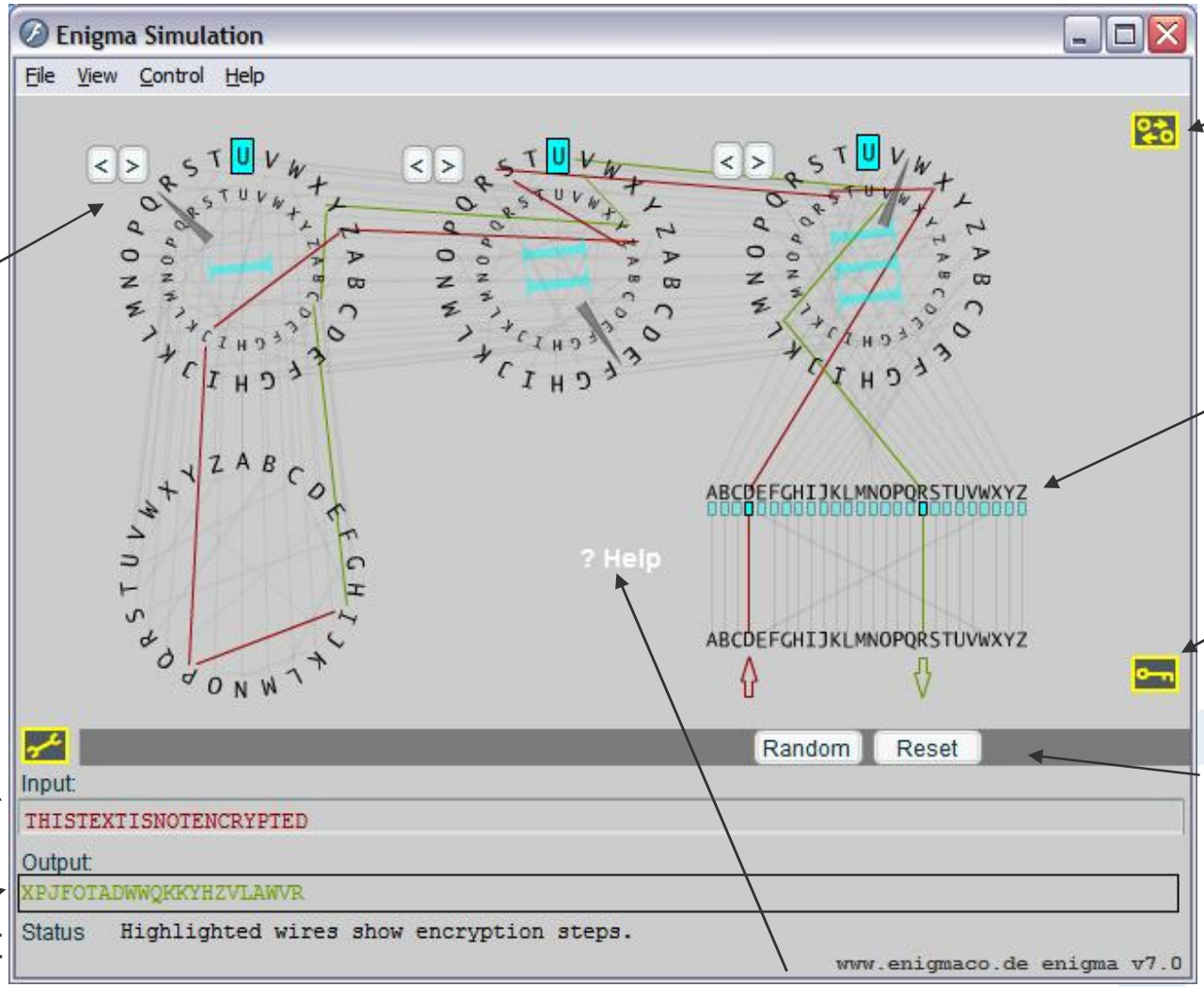
- Visualization of data changes per round using colour gradient



Menu: “Indiv. Procedures” \ “Visualization of Algorithms” \ “AES” \ “Rijndael Flow Visualization...”

Examples (16)

Visualization of the Enigma encryption



Select rotors

Change rotor setting

Change plugs

Show settings

Input of plaintext

Output of encrypted text

Reset Enigma to initial state or random state

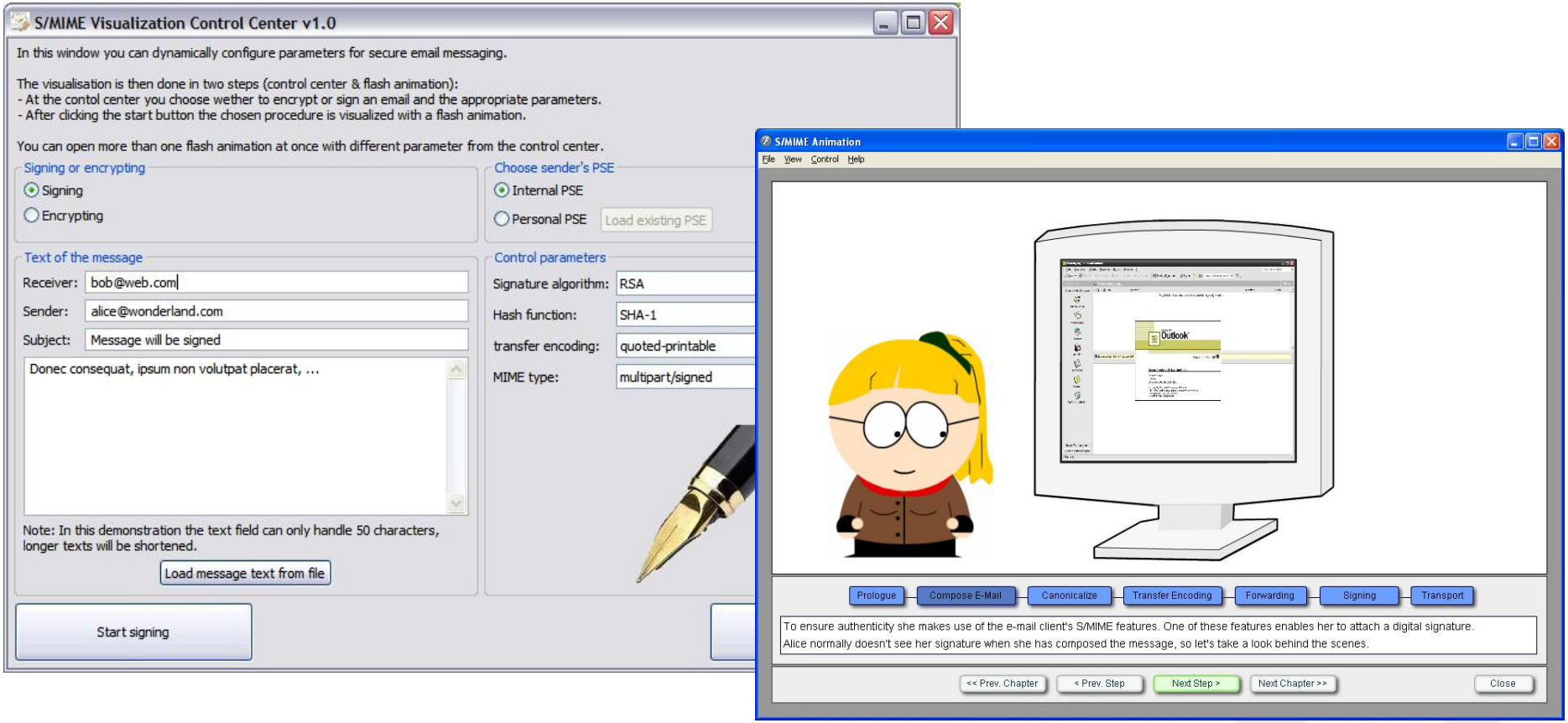
Additional HTML online help

Examples (17)

Visualization of secure E-Mail using S/MIME

S/MIME Visualization

- Control Center: Sign/Encrypt messages with different parameters
- Animation: From the creation at the sender to the reading at the receiver



Menu: „Indiv. Procedures“ \ „Protocols“ \ „Secure E-Mail with S/MIME...“

Examples (18)

Generation of a message authentication code (MAC)

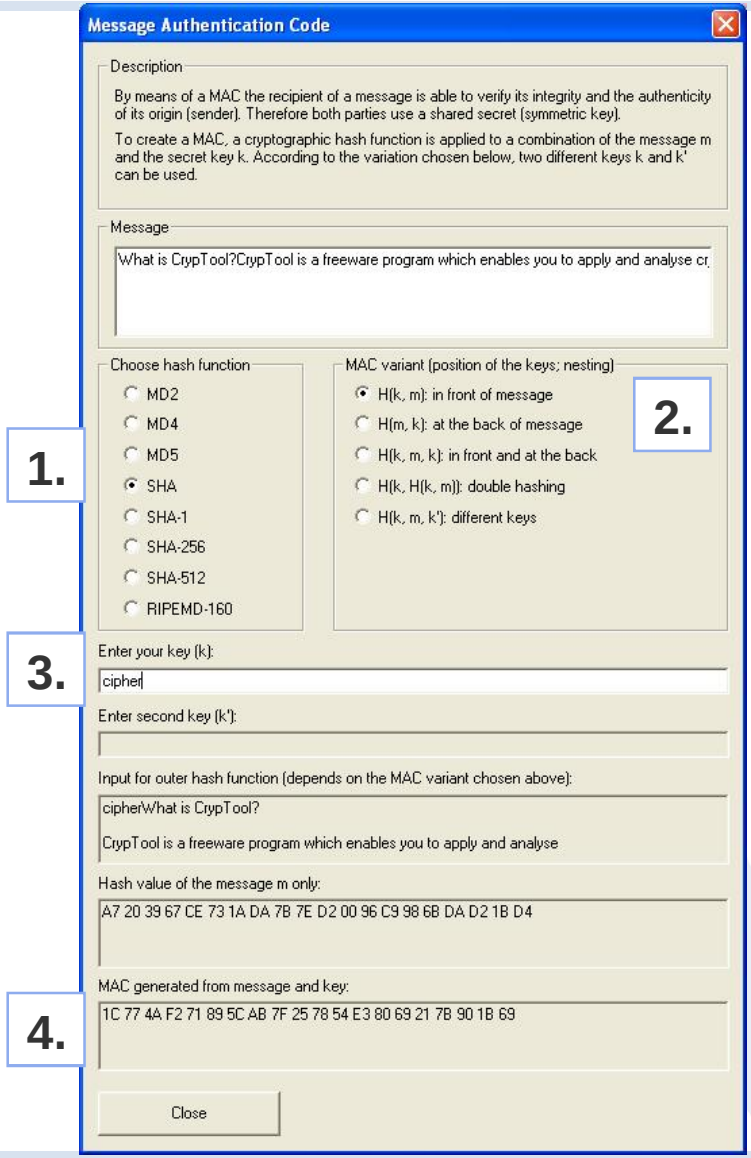
Message Authentication Code (MAC)

- Ensures integrity of a message
- Authentication of the message
- Basis: a common key

Generation of a MAC in CrypTool

1. Choose a hash function
2. Select MAC variant
3. Enter a key (depending on MAC variant also two keys)
4. Generation of the MAC (automatic)

Menu: „Indiv. Procedures" \ „Hash" \ „Generation of MACs"



Examples (19)

Hash demonstration

Sensitivity of hash functions to plaintext modifications

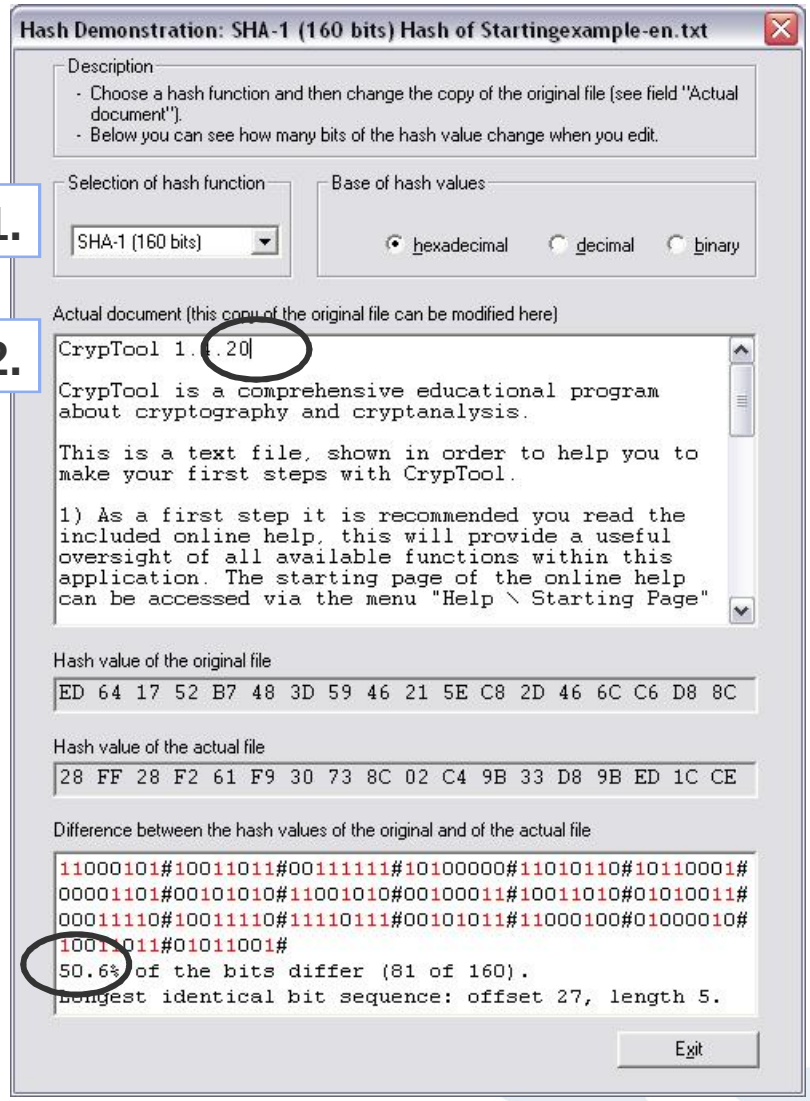
1. Select a hash function
2. Modification of characters in plaintext

Example:

Entering a blank after „CrypTool“ in the example text results in a 50.6 % change of the bits of the generated hash value.

A good hash function should react sensitive to even the smallest change within the plaintext – „Avalanche effect“ (small change, big impact).

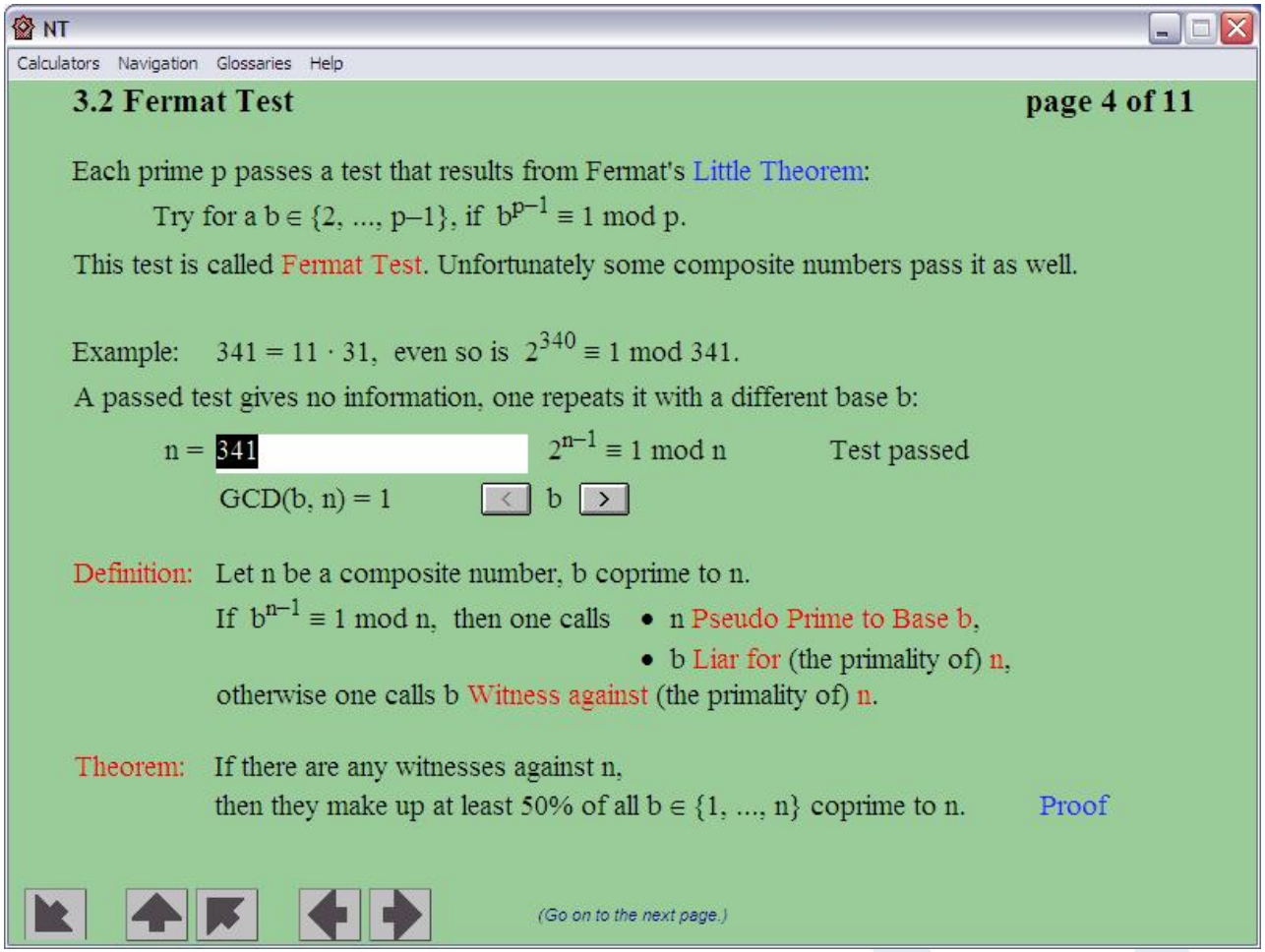
Menu: „Indiv. Procedures” \ „Hash“ \ „Hash Demonstration”



Examples (20)

Learning tool for number theory

- Number theory supported by graphical elements and tools to try-out
- Topics:
  1. Integers
  2. Residue classes
  3. Prime generation
  4. Public-key cryptography
  5. Factorization
  6. Discrete logarithms



Menu: „Indiv. Procedures" \ „Number Theory - Interactive" \ „Learning tool for number theory"

Examples (21)

Point addition on elliptic curves

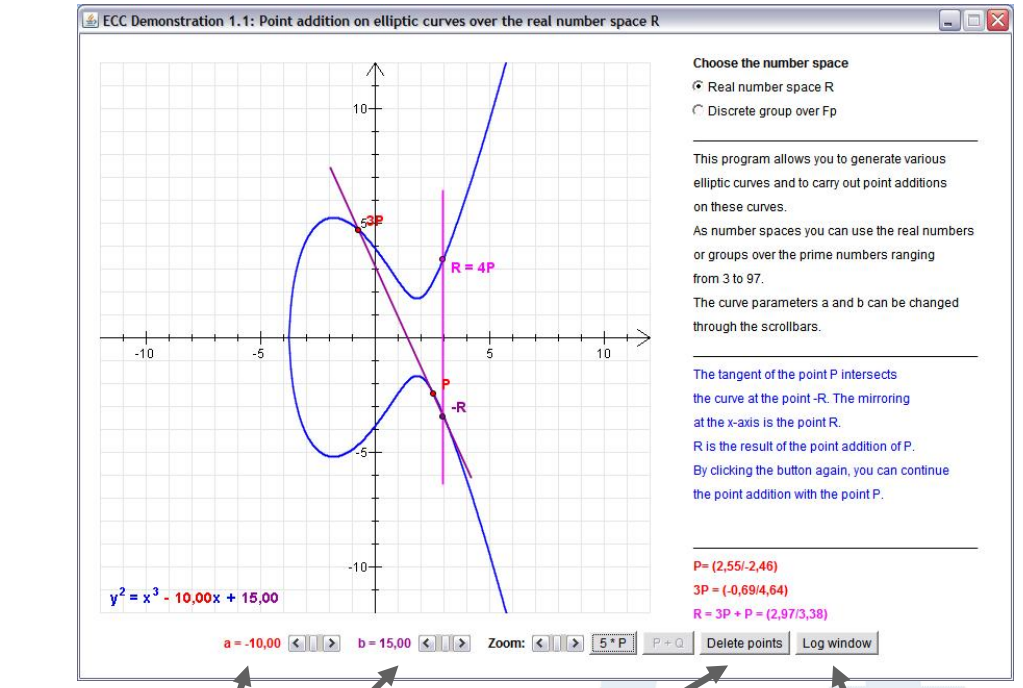
- Visualization of point addition on elliptic curves
- Foundation of elliptic curve cryptography (ECC)

Example 1

- Mark point P on the curve
- Mark point Q on the curve
- Press button „P+Q“: The straight line through P and Q intersects the curve in point -R
- Mirroring on the X-axis results in point R

Example 2

- Mark point P on the curve
- Press button „2*P“: The tangent of point P intersects the curve in point -R
- Mirroring on the X-axis results in point R



Menu: „Indiv. Procedures“ \ „Number Theory - Interactive“ \ „Point Addition on Elliptic Curves“

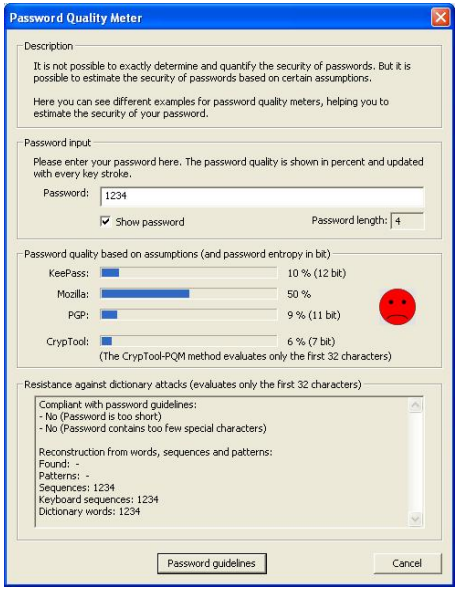
Examples (22)

Password Quality Meter (PQM) 1

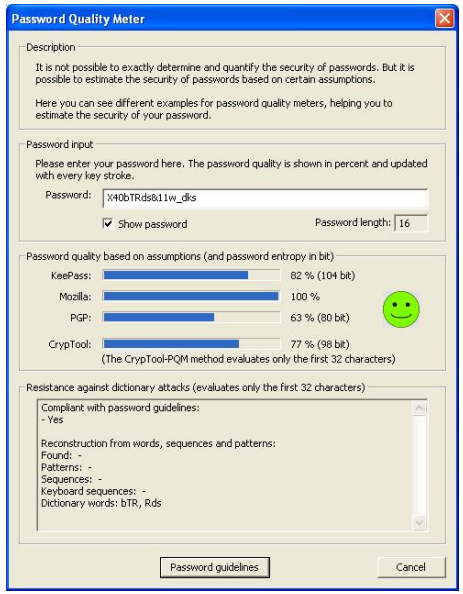
Functions

- Measuring the quality of passwords
- Compare with PQMs in other applications: KeePass, Mozilla und PGP
- Experimental measuring through CrypTool algorithm
- Example: Input of a password (while showing the password)

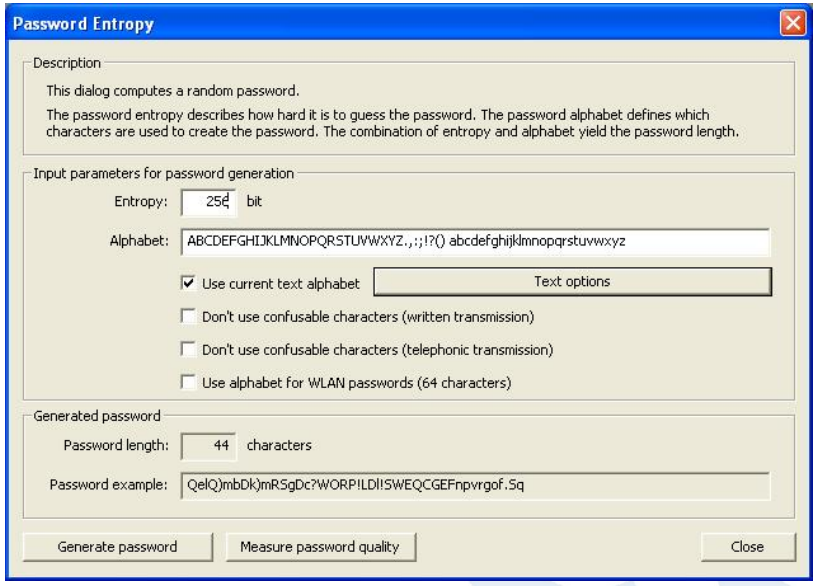
Password: ***1234***



Password: ***X40bTRds&11w_dks***



Menu: “Indiv. Procedures“ “Tools“ \ “Password Quality Meter“



Menu: “Indiv. Procedures“ \ “Tools“ \ “Password Entropy“

Examples (22)

Password Quality Meter (PQM) 2

Findings of the Password Quality Meter

- Password quality depends primarily on the length of the password.
- A higher quality of the password can be achieved by using different types of characters: upper/lower case, numbers and special characters (password space)
- Password entropy as indicator of the randomness of password characters of the password space (higher password entropy results in improved password quality)
- Passwords should not exist in a dictionary (remark: a dictionary check is not yet implemented in CrypTool).

Quality of a password from an attacker's perspective

- Attack on a password (if any number of attempts are possible):
  1. Classical dictionary attack
  2. Dictionary attack with variants (e.g. 4-digit number combinations: Summer2007)
  3. Brute-force attack by testing all combinations (with additional parameters such as limitations on the types of character sets)

⇨ A good password should be chosen so that attack 1. and 2. do not compromise the password. Regarding brute-force attacks the length of the password (at least 8 characters) as well as the used character sets are important.

Examples (23)

Brute-force analysis 1

Brute-force analysis

Optimised brute-force analysis under the assumption that the key is partly known.

Example – Analysis with DES (ECB)

Attempt to find the remainder of the key in order to decrypt an encrypted text (Assumption: The plaintext is a block of 8 ASCII characters).

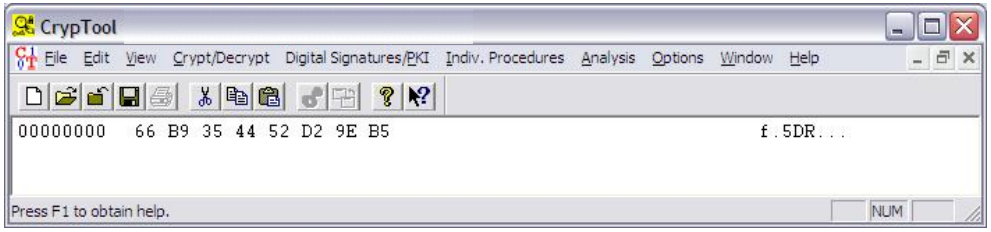
Key (Hex)	Encrypted text (Hex)
68ac78dd40bbefd*	66b9354452d29eb5
0123456789ab****	1f0dd05d8ed51583
98765432106*****	bcf9ebd1979ead6a
0000000000******	8cf42d40e004a1d4
000000000000****	0ed33fed7f46c585
abacadaba*******	d6d8641bc4fb2478
ddddddddddd******	a2e66d852e175f5c

Examples (23)

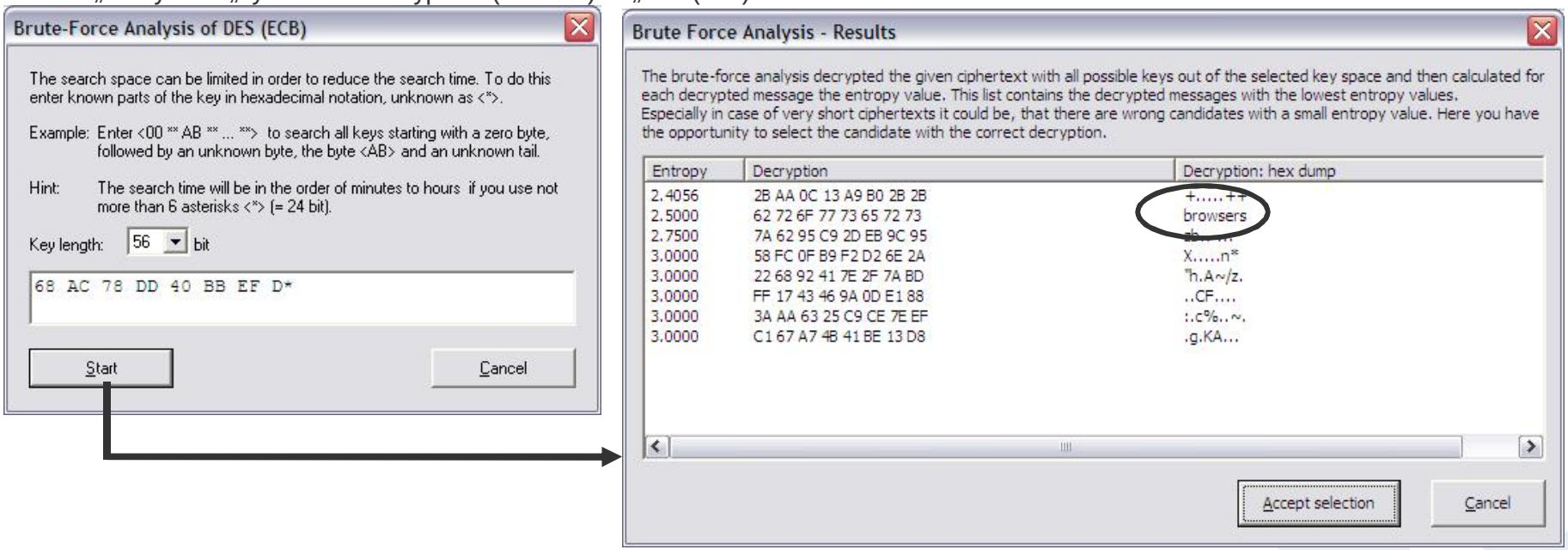
Brute-force analysis 2

1. Input of encrypted text
2. Use brute-force analysis
3. Input partly known key
4. Start brute-force analysis
5. Analysis of the results: Low entropy as evidence of a possible decryption. However, because a very short plaintext has been used in this example, the correct result does not have the lowest entropy.

Use „View“ \ „Show as HexDump“

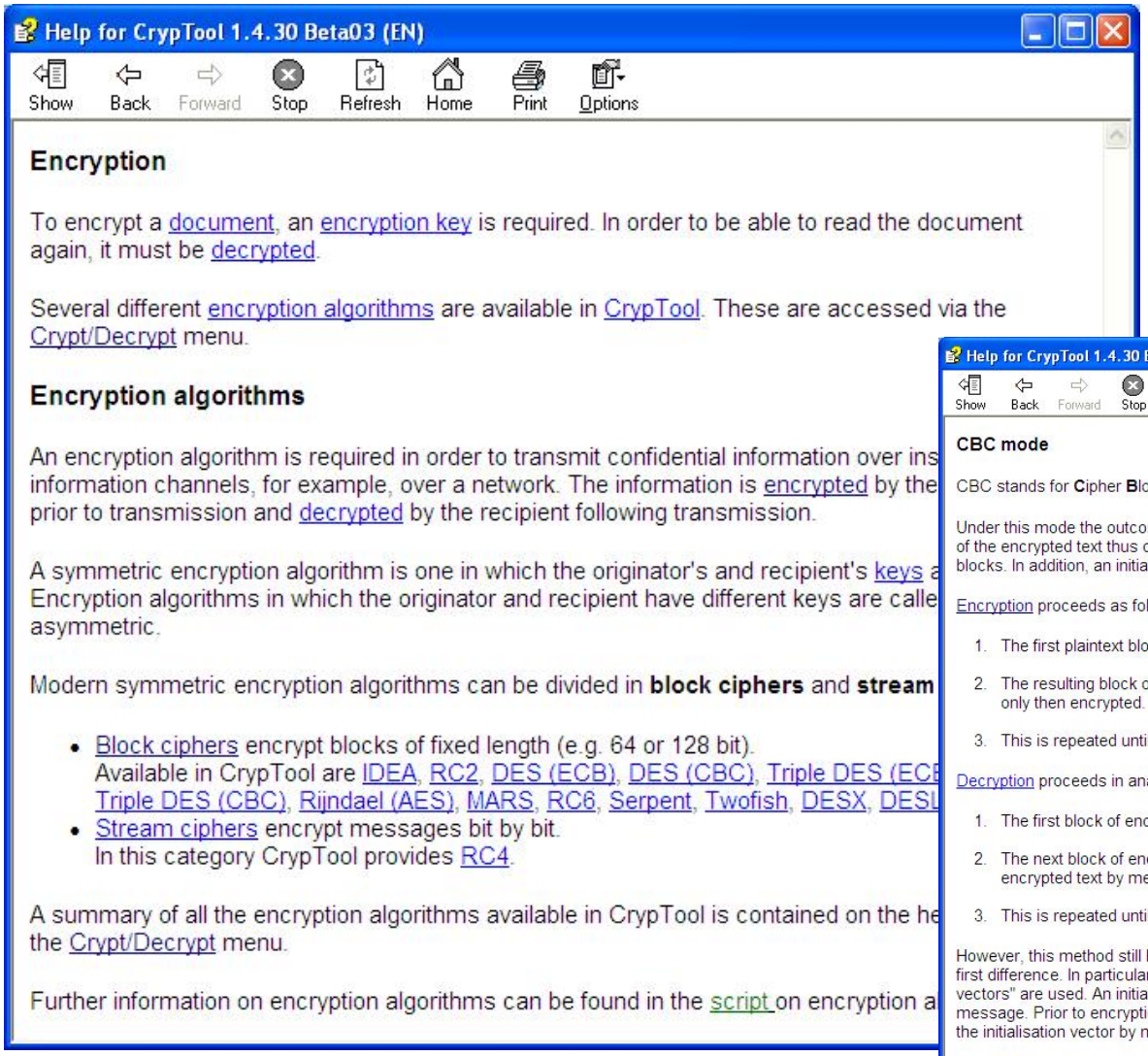


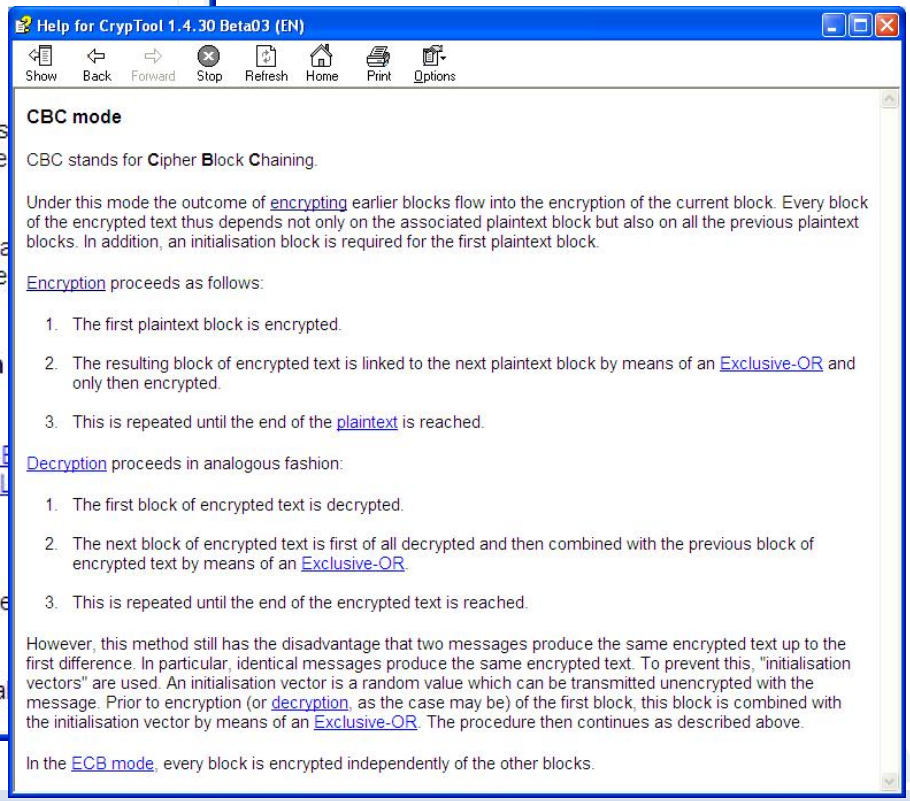
Menu: „Analysis“ \ „Symmetric Encryption (modern)“ \ „DES (ECB)“



Examples (24)

CrypTool online help 1

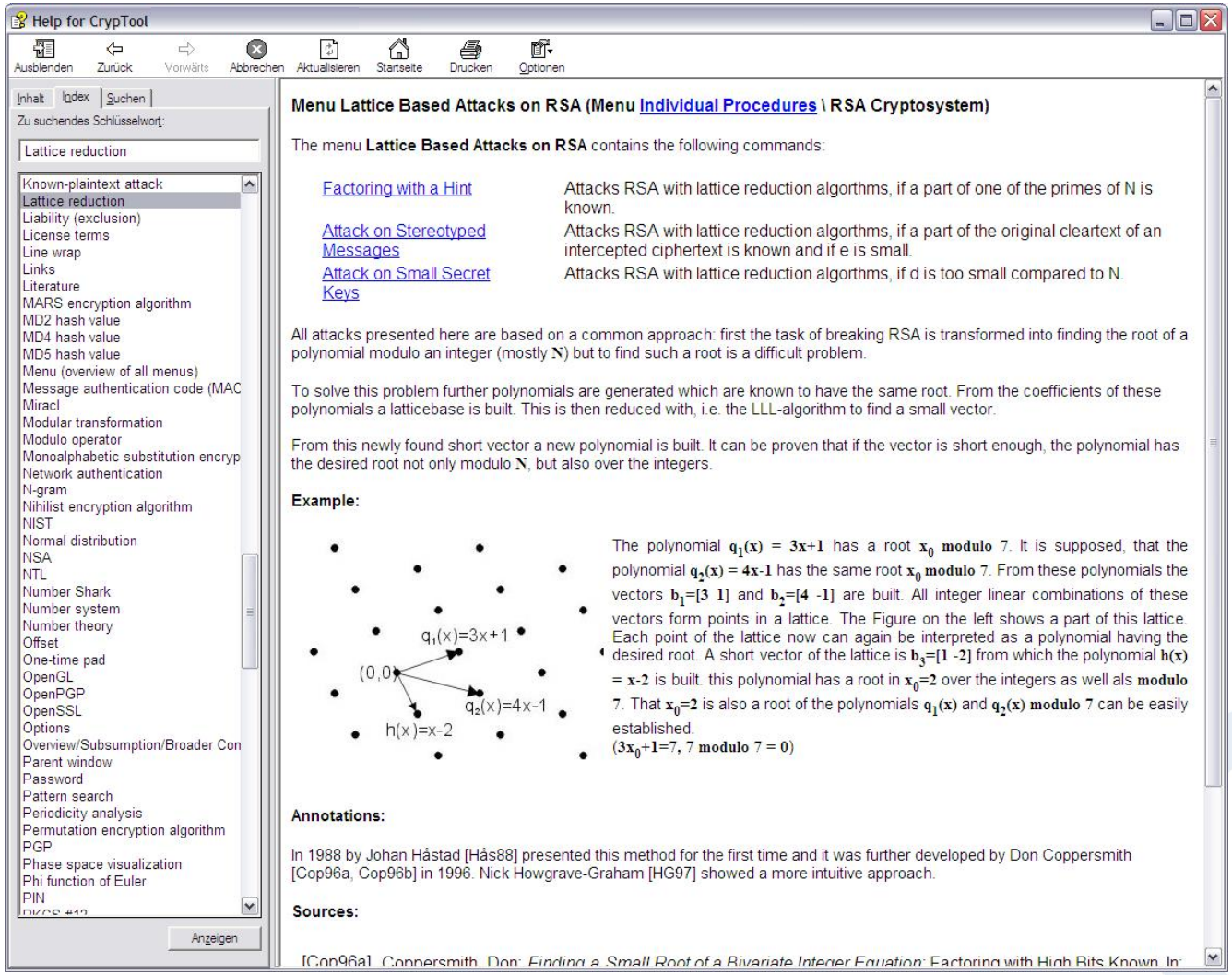




Menu: „Help“ \ „Starting Page“

Examples (24)

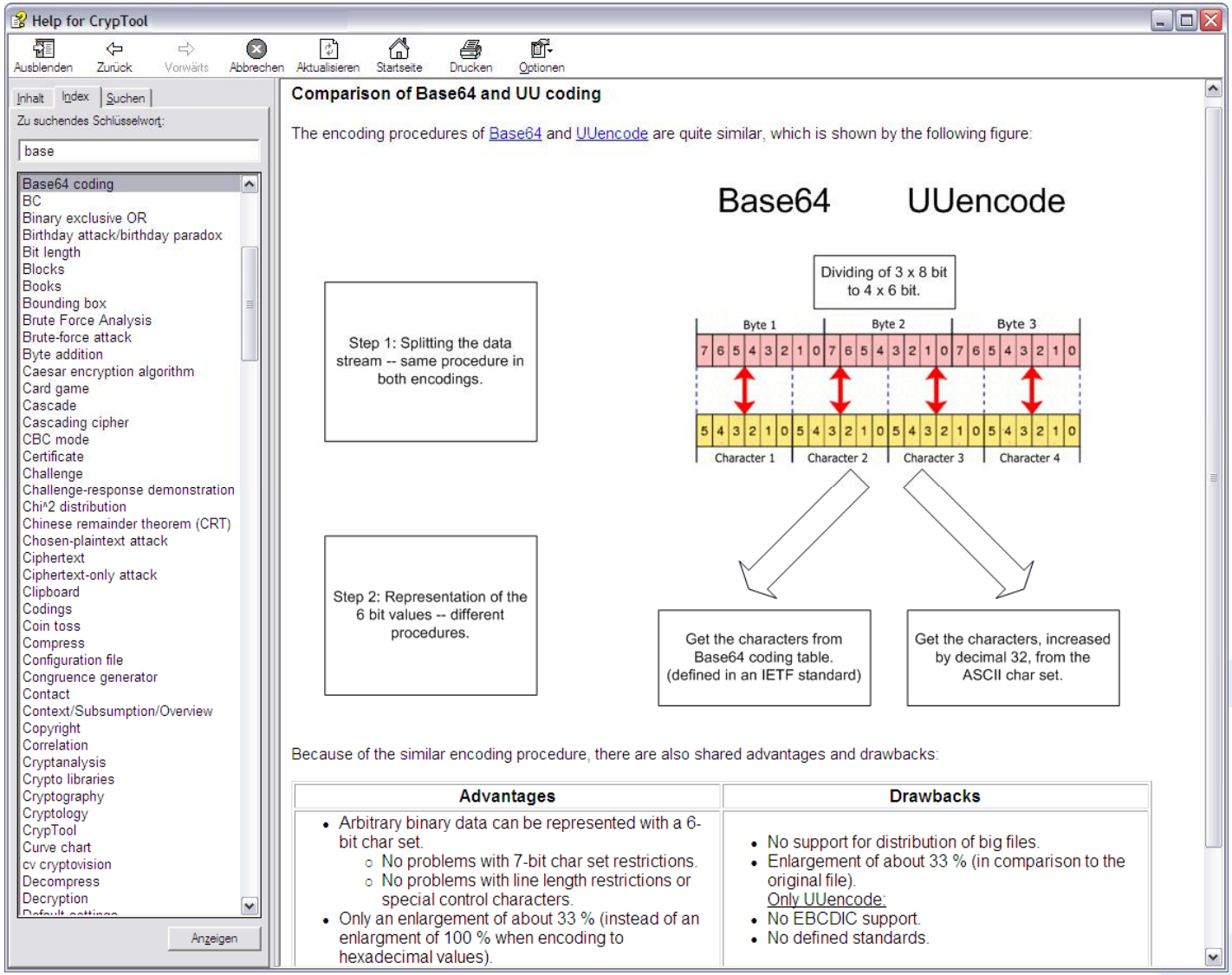
CrypTool online help 2



Menu: „Help“ \ „Starting Page“

Examples (24)

CrypTool online help 3



Content

Future CrypTool Development (1)

CT1 = CrypTool 1.x
CT2 = CrypTool 2.0
JCT = JCrypTool

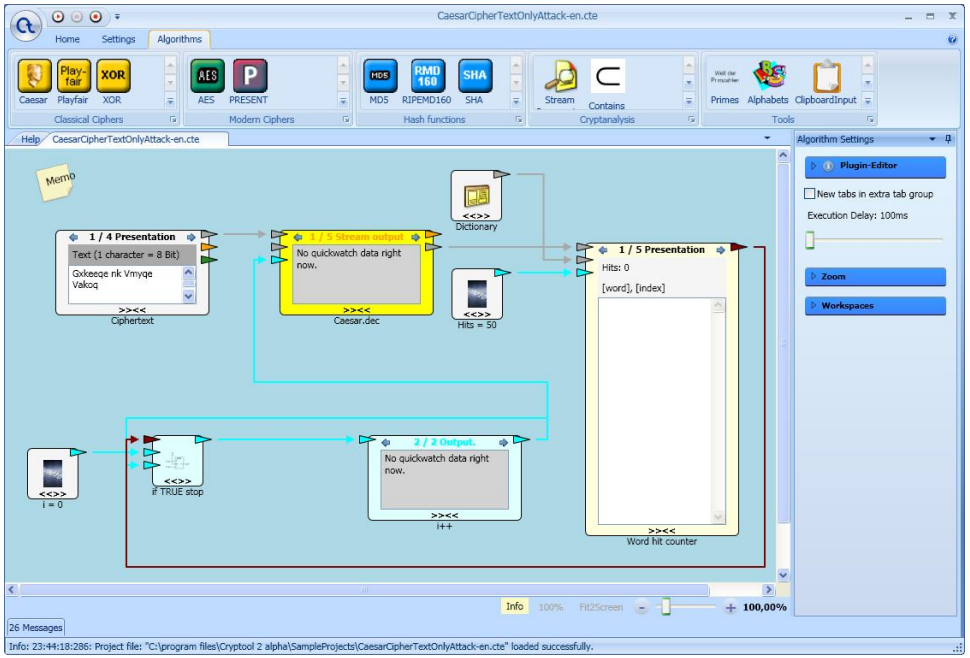
Planned after release CrypTool 1.4.30 (see readme file)

CT1	Sigaba, FIPS test to investigate more than 2500 bytes, ...
JCT	Tri-partite key agreements
JCT	Visualization of the interoperability between S/MIME and OpenPGP formats
JCT	Analysis of entropy
JCT	Grille, Autokey Vigenère, interactive cryptanalysis of classic ciphers
JCT	Analysis of transposition ciphers using the ACO algorithm
JCT	Visualization of zero-knowledge proofs
JCT	Cube attack (I. Dinur and A. Shamir, "Cube Attacks on Tweakable Black Box Polynomials", 2008)
JCT	Action-History with the additional feature, to create any product cipher
CT2	Comprehensive visualization on the topic of prime numbers
CT2	Demonstration of Bleichenbacher's RSA signature forgery
CT2	Demonstration of virtual credit card numbers (approach against credit card abuse)
CT2	WEP encryption and WEP analysis
CT2	Mass pattern search
CT2	Demonstration of SOA security (SOAP messages according WS Security between the participants)
CT2	Framework to create and analyze LFSR stream ciphers
CT2	Graphical design oriented mode for beginners plus expert mode
CT2/JCT	Creation of a command line version for batch processing
CT2/JCT	Modern pure plugin architecture with loading of plugins
All	Further parameterization / Increasing the flexibility of present algorithms
Idea	Visualization of the SSL protocol
Idea	Demonstration of visual cryptography
Idea	Integration of crypto library crypto++ from Wei Dai

Future CrypTool Development (2)

In Progress (see readme file)

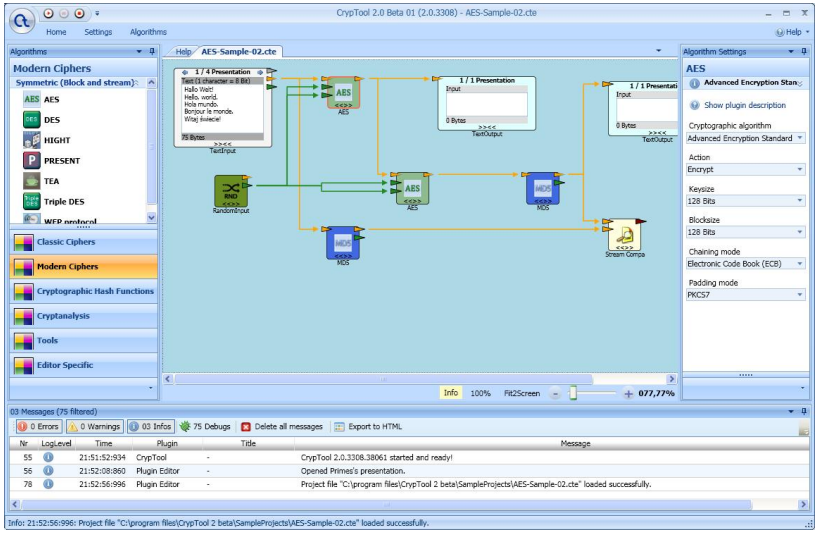
1. JCT: Port and redesign of CrypTool in Java / SWT / Eclipse 3.5 / RPC
   - see: http://jcryptool.sourceforge.net
   - Milestone 4a available for users and developers (February 2009) // M5: planned for August 2009
2. CT2: Port and redesign of the C++ version with C# / WPF / VS2008 / .NET 3.5
   - direct successor of current releases: allows visual programming, ...
   - Beta available for users and developers (since July 2008, permanently updated)
3. C2L: Direct port of the C++ version to Linux with Qt4 (very slow progress)
   - see: http://www.cryptoolinux.net

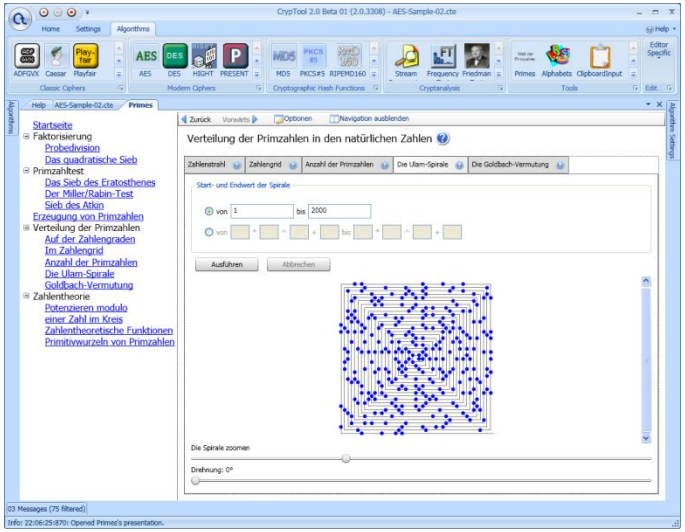


CrypTool 2 (CT2)

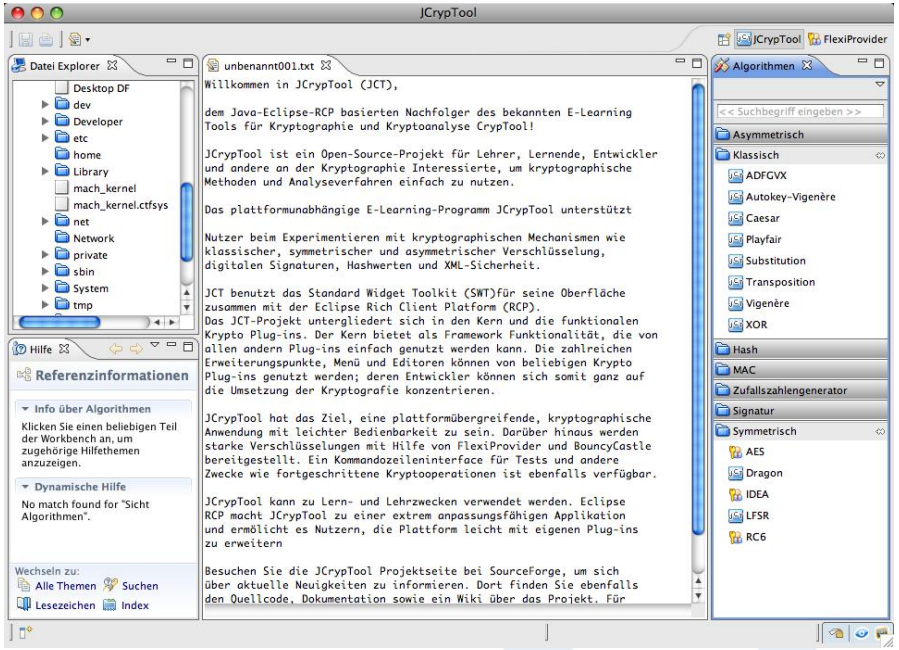
JCrypTool (JCT)

Future CrypTool Development (3)





CrypTool 2 (CT2)



JCrypTool (JCT)

CrypTool as a Framework

Proposal

- Re-use the comprehensive set of algorithms, included libraries and interface elements as foundation
- Free of charge training in Frankfurt, how to start with CrypTool development
- Advantage: Your own code does not „disappear“, but will be maintained

Current development environment for CT1: Microsoft Visual Studio C++ , Perl, Subversion Source-Code Management

- CrypTool 1.4.30: Visual C++ .net (= VC++ 9.0)(= Visual Studio 2008 Standard)
- Description for developers: see readme-source.txt
- Download: Sources and binaries of releases.
  To get sources of current betas, please see subversion repository.

Development environments for CT2 and JCT:

- CT2 – C# version: .NET with Visual Studio 2008 Express Edition (free), WPF und Perl
- Java – Java version: Eclipse 3.5, RCP, SWT (free)

CrypTool – Request for Contribution

Every contribution to the project is highly appreciated

- Feedback, criticism, suggestions and ideas
- Integration of additional algorithms, protocols, analysis (consistency and completeness)
- Development assistance (programming, layout, translation, test)
  - For the current C/C++ project
  - For the new projects (preferred):
    - C# project: „CrypTool 2.0“ = CT2
    - Java project: „JCrypTool“ = JCT
  - Especially University faculties using CrypTool for educational purposes are invited to contribute to the further development of CrypTool.
- Significant contributions can be referenced by name (in help, readme, about dialog and on the CrypTool web site).
- Currently CrypTool is being downloaded more than 4000 times a month (with a little bit more than 50 % getting the English version).

CrypTool – Summary

- *THE* e-learning program for cryptology
- Over more than 10 years a successful open source project
- More than 200,000 downloads
- International utilisation in schools, universities as well as companies and government agencies
- Extensive online help and documentation
- Available for free and multi-language support

Contact

Prof. Bernhard Esslinger

University of Siegen
Faculty 5, Economics and Business Computing

Deutsche Bank AG
Director, IT Security Manager

esslinger@fb5.uni-siegen.de

www.cryptool.org
www.cryptool.com
www.cryptool.de
www.cryptool.es
www.cryptool.pl

Additional contacts: See readme within the CrypTool folder

Additional Literature

As introduction to cryptology and more

- Klaus Schmeh, "*Codeknacker gegen Codemacher. Die faszinierende Geschichte der Verschlüsselung*", 2nd edition, 2007, W3L [German]
- Simon Singh, *"The Codebook"*, 1999, Doubleday
- Udo Ulfkotte, "Wirtschaftsspionage", 2001, Goldmann [German]
- Johannes Buchmann, "*Introduction to Cryptography*", 2nd edition, 2004, Springer
- Claudia Eckert, *"IT-Sicherheit", 5th* edition, 2008, Oldenbourg [German]
- A. Beutelspacher / J. Schwenk / K.-D. Wolfenstetter, *"Moderne Verfahren der Kryptographie"*, 5th edition, 2004, Vieweg [German]
- [HAC] Menezes / van Oorschot / Vanstone, *"Handbook of Applied Cryptography"*, 1996, CRC Press
- van Oorschot / Wiener, *"Parallel Collision Search with Application to Hash Functions and Discrete Logarithms"*, 1994, ACM
- Additional cryptography literature – see also the links at the CrypTool web page and the literature in the CrypTool online help (e.g. by Wätjen, Salomaa, Brands, Schneier, Shoup, Stamp/Low, ...)
- Importance of cryptography in the broader context of IT security and risk management
  - See e.g. Kenneth C. Laudon / Jane P. Laudon / Detlef Schoder, "Wirtschaftsinformatik", 2009, Pearson, chapter 14 [German]
  - Wikipedia: http://en.wikipedia.org/wiki/Risk_management
  - CrypTool Site: http://cryptool.com/index.php/en/cryptool-for-awareness-aboutmenu-74.html

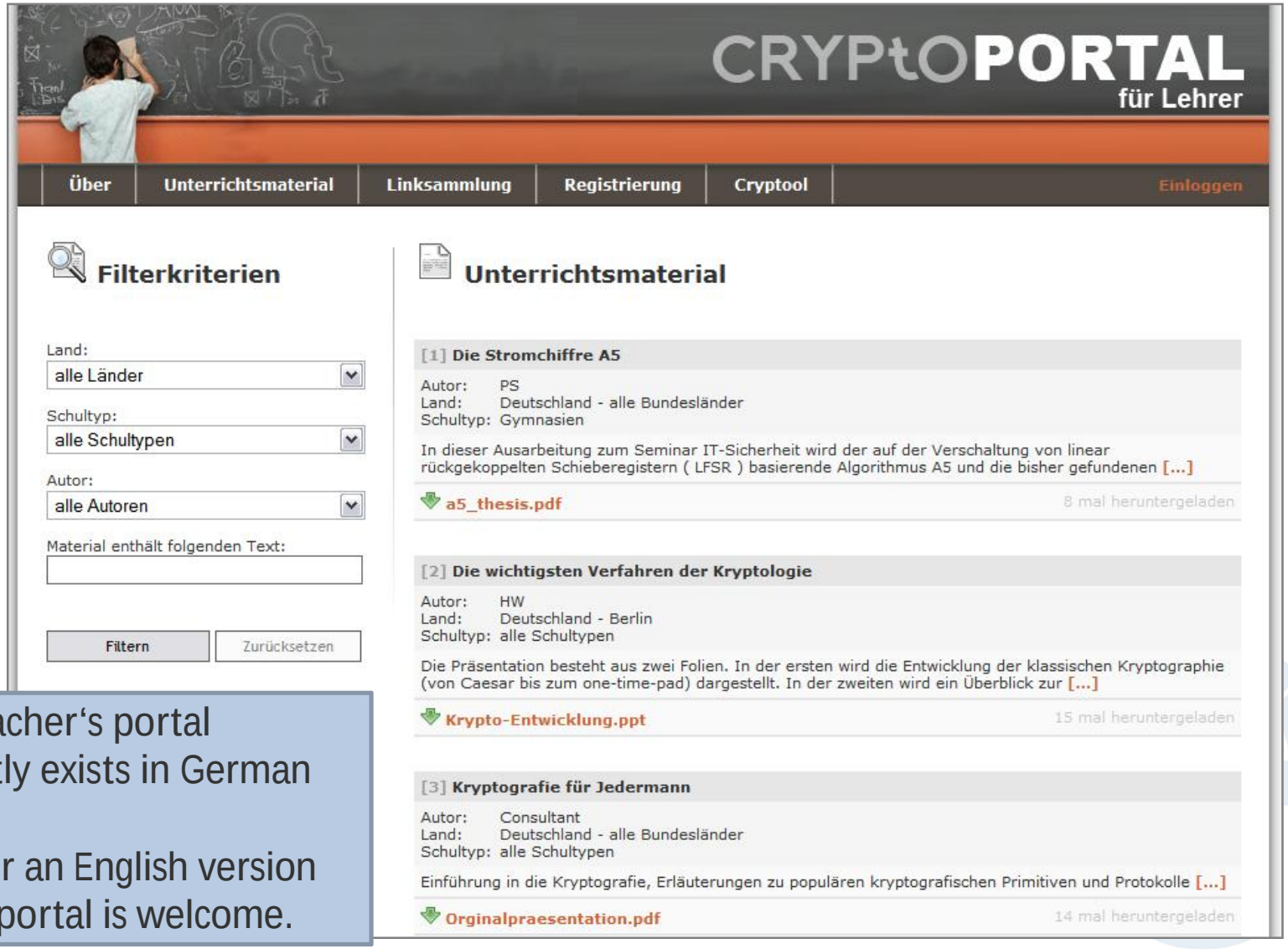
www.cryptool.org / .com / .de / .es / .pl

www.cryptool-online.org

The newest member in the family of CrypTool related websites:

- CrypTool site (CT1)
- CT2 developer site
- JCT developer site
- CryptoPortal for teachers (currently only in German)
- CrypTool-Online (try cryptographic methods within your browser).

www.cryptoportal.org



The teacher‘s portal currently exists in German only.
Help for an English version of this portal is welcome.

Download Software and CrypTool Script