CrypTool

A free software program

- **for creating awareness of IT security issues**
- **for learning about and obtaining experience of cryptography**
- **for demonstrating encryption algorithms and analysis procedures**

www.cryptool.de
www.cryptool.com
www.cryptool.org

Claudia Eckert / Thorsten Clausius
Bernd Esslinger / Jörg Schneider / Henrik Koy

TECHNISCHE UNIVERSITÄT DARMSTADT

Contents

Introduction

1. What is CrypTool?

- a freeware Program with graphical user interface
- a tool for applying and analysing cryptographic algorithms
- with extensive online help, understandable without deep crypto knowledge
- contains nearly all state of the art crypto algorithms
- “playful” introduction to modern and classical cryptography
- not a “hacker tool”

2. Why CrypTool?

- origin in Deutsche Bank’s IT security awareness program
- developed in co-operation with universities
- improve IT security related courses in universities and companies

3. Audience

- target group: students of computer science, commercial IT and mathematics
- also aimed at: interested computer users and application developers
- prerequisites: secondary school mathematics or programming skills

CrypTool Overview
1. Features

Cryptography

Classical algorithms

- Caesar
- Vigenère
- Hill
- Monoalphabetic substitution
- Homophonic substitution
- Playfair
- Permutation
- Addition
- XOR
- Vernam

To facilitate performing text book examples with CrypTool

- alphabet can be configured
- treatment of white space etc. configurable

Cryptoanalysis

Attacks on classical algorithms

- ciphertext only
  - Caesar
  - Vigenère
  - Addition
  - XOR
- known plaintext
  - Hill
  - Playfair
- manual
  - mono-alphabetic substitution

Supporting analysis procedures

- entropy, floating frequency
- histogram, n-gram analysis
- auto-correlation
- ZIP compression test

TECHNISCHE UNIVERSITÄT DARMSTADT

CrypTool Overview
1. Features

Cryptography

Modern symmetric algorithms

- IDEA, RC2, RC4, DES, 3DES
- last round AES candidates
- AES (=Rijndael)

Asymmetric algorithms

- RSA with X.509 certificates
- RSA demonstration
  - to facilitate performing text book examples with CrypTool
  - alphabet and block length configurable

Hybrid encryption

- RSA combined with AES encryption
- visualised by an interactive data flow diagram

Cryptoanalysis

Brute-force attack on symmetric algorithms

- implemented for all algorithms
- assumption:
  entropy of the plain text is small

Attack on RSA encryption

- factor RSA modulus
- workable for bit lengths <= 250

Attack on hybrid encryption

- attack on RSA (see below) or
- attack on AES (see above)

CrypTool Overview
1. Features

Cryptography

Digital Signature
- RSA with X.509 certificates
  - signature procedure visualised by an interactive data flow diagram
- DSA with X.509 certificates
- Elliptic curve DSA, Nyberg-Rueppel

Hash functions
- MD2, MD4, MD5
- SHA, SHA-1, RIPEMD-160

Random generators
- SECUDE
- X^2 modulo N
- Linear Congruence Generator (LCG)
- Inverse Congruence Generator (ICG)

Cryptoanalysis

Attack on RSA Signature
- RSA modulus factorisation
- workable up to approx. 250 bit

Attack on hash function/digital signature
- Generation of hash collisions to ASCII texts

Random data analysis
- FIPS-PUB-140-1 test battery
- periodicity, Vitany, entropy
- histogram, n-gram analysis
- auto-correlation
- ZIP compression test

TECHNISCHE UNIVERSITÄT DARMSTADT

CrypTool Overview
2. Software package contents

completely bilingual
English
German

CrypTool program

- all functions integrated in *one* program with uniform graphical user interface
- platforms: Win32 and Linux with WINE emulator
- cryptography based on Secude library (www.secude.com)
- arbitrary precision arithmetic: Miracl library (http://indigo.ie/~mscott/)

AES-Tool

- standalone program for AES encryption (self extracting)

Extensive online help (Winhelp)

- context sensitive online help for *all* program functions and all menu items
- detailed examples of usage for many program features

Script (PDF) with background information on

- encryption algorithms • prime numbers • digital signature
- elliptic curves • public key certification • elementary number theory

Short story “Dialogue of the Sisters” by Dr. C. Elsner

CrypTool Overview
3. New in release 1.3.xx

Most important changes (details: see ReadMe-en.txt):

Release 1.3.00 published January 2002

- completely bilingual English/German
- improved dialog box consistency and comprehensibility
- Windows 9x file size limit removed
- homophonic and permutation encryption
- random generators, random data analysis (FIPS-140-1, periodicity, n-gram)
- AES-Tool: create self-decrypting files (AES)
- demonstration: number theory and RSA crypto system (further improved in 1.3.02)
- PKCS#12 export/import for PSEs

Release 1.3.02 published June 2002

- visualisation of hybrid encryption and decryption
- visualisation of signature creation and verification
- hash value calculation of large files (without loading them into memory)
- visualisation of the sensitivity of hash functions to changes in the hashed data
- short story “Dialogue of the Sisters” by Dr. C. Elsner included

TECHNISCHE UNIVERSITÄT DARMSTADT

CrypTool Overview
3. New in release 1.3.xx

Release 1.3.04 published June 2003

- visualisation of Diffie-Hellman key exchange
- attack on digital signature using hash-collisions (birthday paradox)
- brute-force attack on symmetric ciphers improved
- script updated (primes, factorization) and extended (hash functions, ECC, CrypTool menu tree)
- many small improvements (especially online help) and bug fixes

TECHNISCHE UNIVERSITÄT DARMSTADT

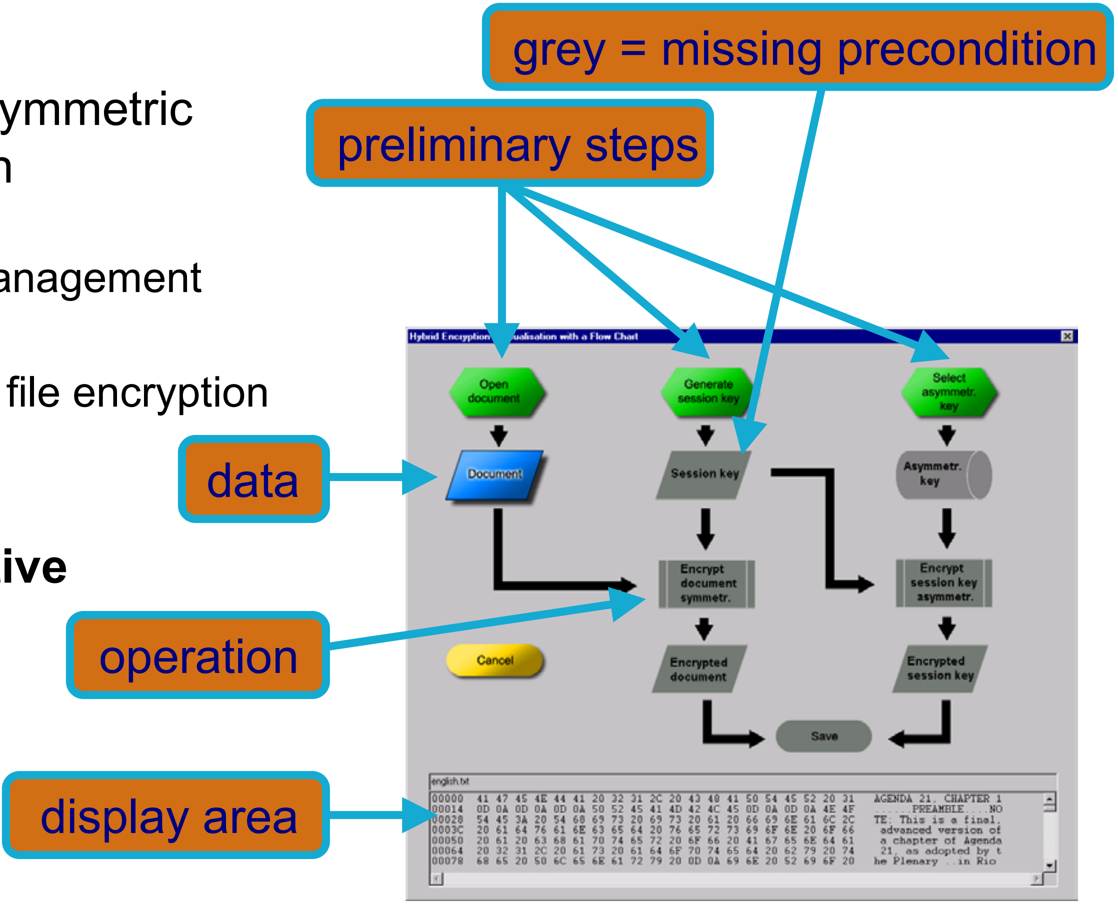
Examples of use
1. Hybrid encryption visualised

Hybrid encryption

- combines advantages of symmetric and asymmetric encryption
  - speed
  - simple and scalable key management
- widely used in practice
  - e-mail (S/MIME, PGP) and file encryption
  - SSL (https)

Visualisation by an interactive data flow diagram

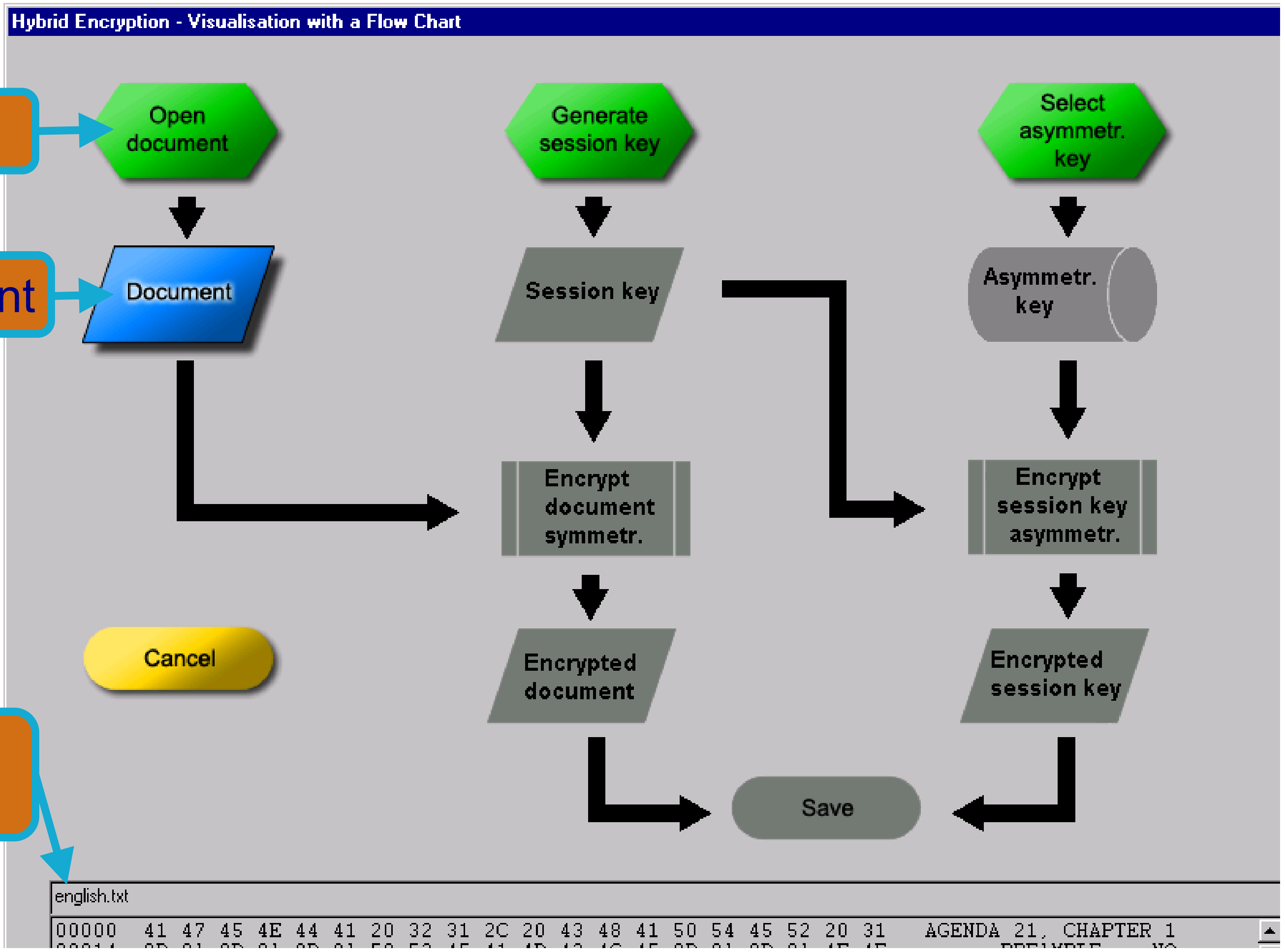
- playful learning leads to deeper understanding



```
00000  41 47 45 4E 44 41 20 32 31 2C 20 43 48 41 50 54 45 52 20 31   AGENDA 21, CHAPTER 1
00014  0D 0A 0D 0A 0D 0A 50 52 45 41 4D 42 4C 45 0D 0A 0D 0A 4E 4F   ......PREAMBLE....NO
00028  54 45 3A 20 54 68 69 73 20 69 73 20 61 20 66 69 6E 61 6C 2C   TE: This is a final,
0003C  20 61 64 76 61 6E 63 65 64 20 76 65 72 73 69 6F 6E 20 6F 66    advanced version of
00050  20 61 20 63 68 61 70 74 65 72 20 6F 66 20 41 67 65 6E 64 61    a chapter of Agenda
00064  20 32 31 2C 20 61 73 20 61 64 6F 70 74 65 64 20 62 79 20 74    21, as adopted by t
00078  68 65 20 50 6C 65 6E 61 72 79 20 0D 0A 69 6E 20 52 69 6F 20   he Plenary ..in Rio
```

TECHNISCHE UNIVERSITÄT DARMSTADT

Examples of use
1. Hybrid encryption visualised: Preparation



1. select file

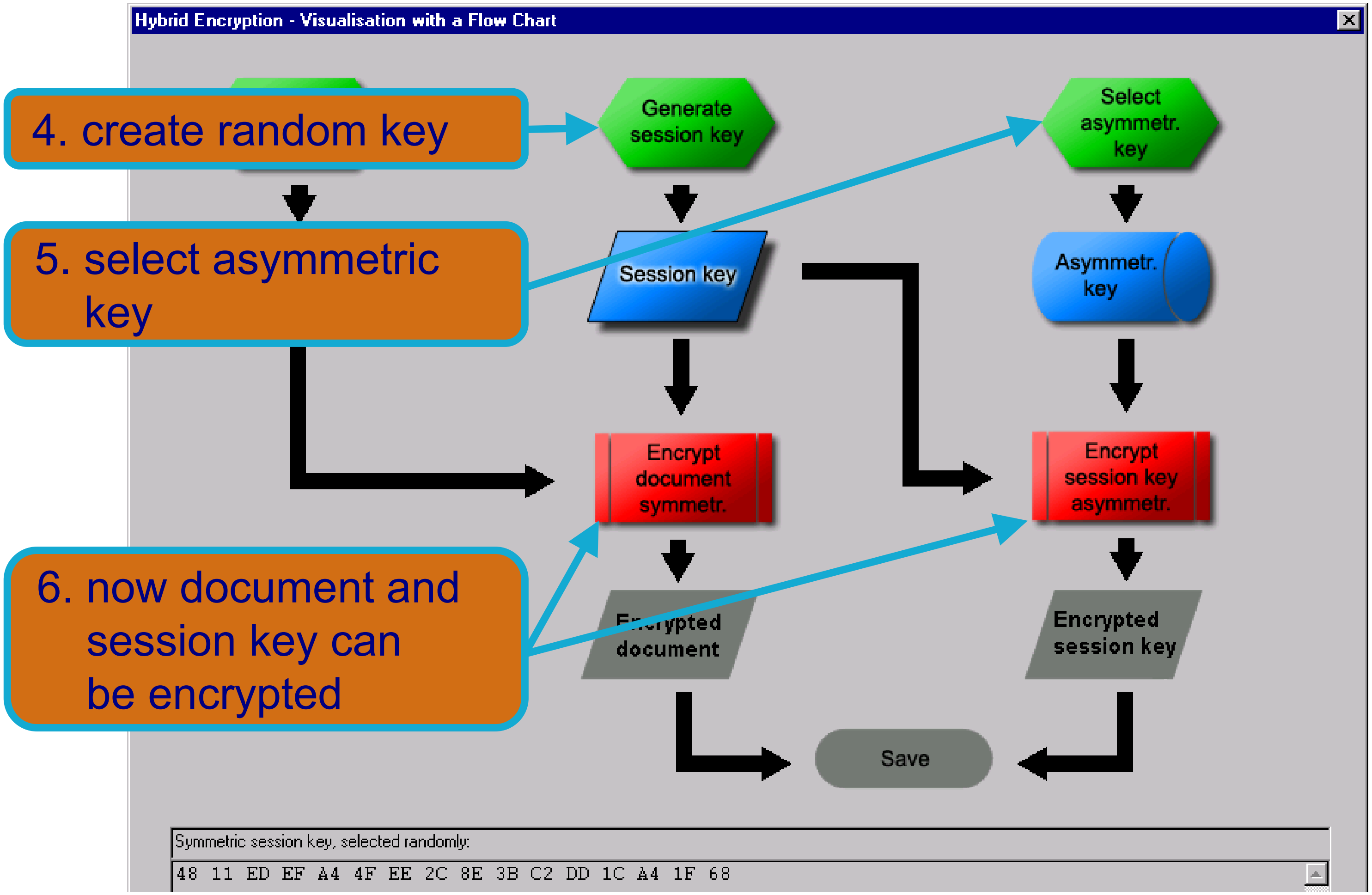
2. click on document

3. contents displayed here

```
english.txt
00000  41 47 45 4E 44 41 20 32 31 2C 20 43 48 41 50 54 45 52 20 31   AGENDA 21, CHAPTER 1
```

TECHNISCHE UNIVERSITÄT DARMSTADT

Examples of use
1. Hybrid encryption visualised: Cryptography

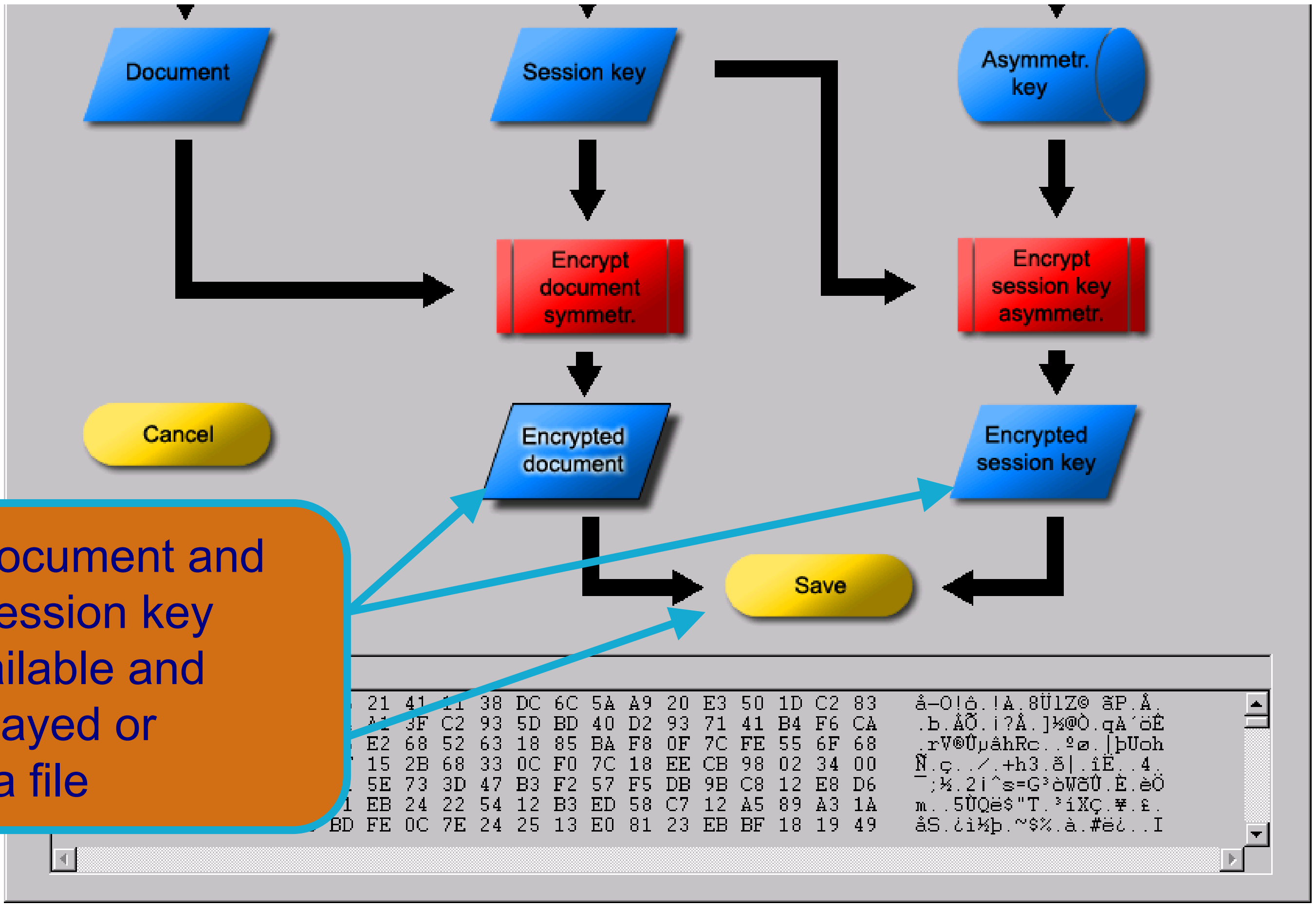

Hybrid Encryption - Visualisation with a Flow Chart
4. create random key
5. select asymmetric key
6. now document and session key can be encrypted
Generate session key
Select asymmetr. key
Session key
Asymmetr. key
Encrypt document symmetr.
Encrypt session key asymmetr.
Encrypted document
Encrypted session key
Save

Symmetric session key, selected randomly:

```
48 11 ED EF A4 4F EE 2C 8E 3B C2 DD 1C A4 1F 68
```

TECHNISCHE UNIVERSITÄT DARMSTADT

Examples of use
1. Hybrid encryption visualised: Result



```
   21 41 11 38 DC 6C 5A A9 20 E3 50 1D C2 83    å—O!ô.!A.8ÜlZ© ãP.Å.
   A1 3F C2 93 5D BD 40 D2 93 71 41 B4 F6 CA    .b.ÅÕ.¡?Å.]¼@Ò.qA´öÊ
   E2 68 52 63 18 85 BA F8 0F 7C FE 55 6F 68    .rV®ÛµâhRc..º ø.|þUoh
   15 2B 68 33 0C F0 7C 18 EE CB 98 02 34 00    Ñ.ç../.+h3.ð|.îË..4.
   5E 73 3D 47 B3 F2 57 F5 DB 9B C8 12 E8 D6    ¯;¼.2¡^s=G³òWõÛ.È.èÖ
   EB 24 22 54 12 B3 ED 58 C7 12 A5 89 A3 1A    m..5ÛQë$"T.³íXÇ.¥.£.
BD FE 0C 7E 24 25 13 E0 81 23 EB BF 18 19 49    åS.¿i½þ.~$%.à.#ë¿..I
```

7. encrypted document and encrypted session key are now available and can be displayed or written into a file

TECHNISCHE UNIVERSITÄT DARMSTADT

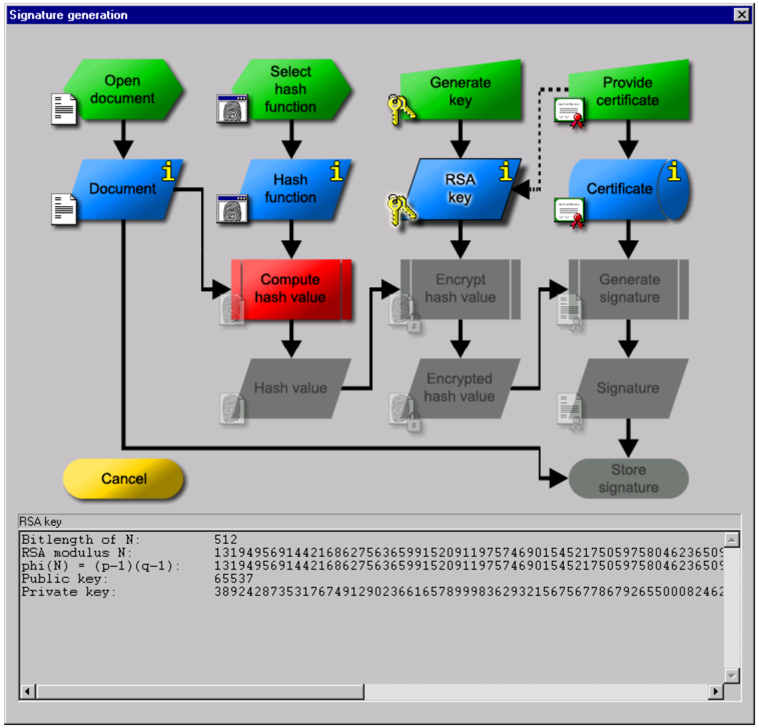
Examples of use
2. Digital signature visualised

Digital signature

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

Visualisation in CrypTool

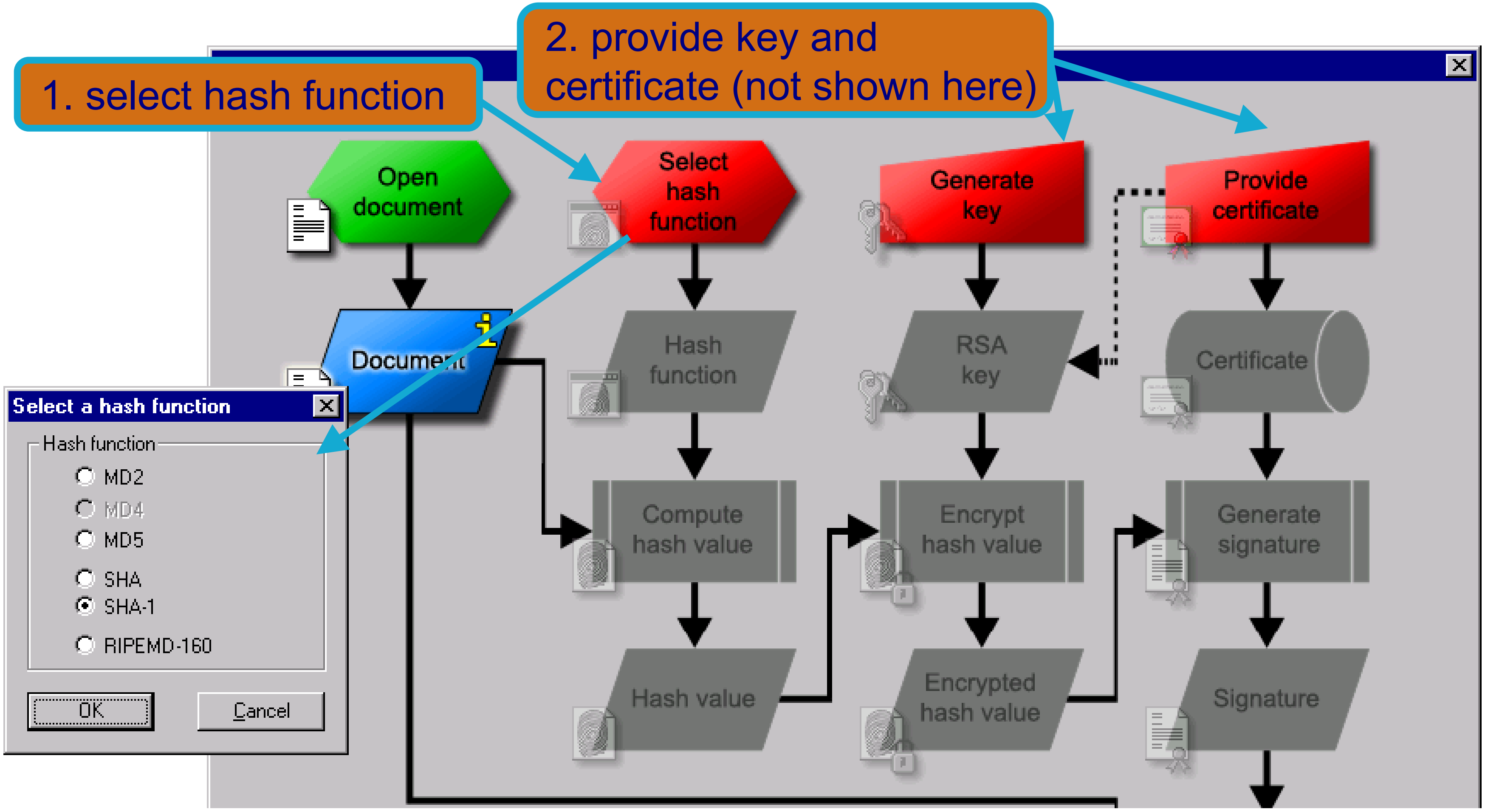
- interactive data flow diagram
- similar to the visualisation of hybrid encryption



```
Bitlength of N:           512
RSA modulus N:            131949569144216862756365991520911975746901545217505975804623650
phi(N) = (p-1)(q-1):      131949569144216862756365991520911975746901545217505975804623650
Public key:               65537
Private key:              389242873531767491290236616578999836293215675677867926550008246
```

TECHNISCHE UNIVERSITÄT DARMSTADT

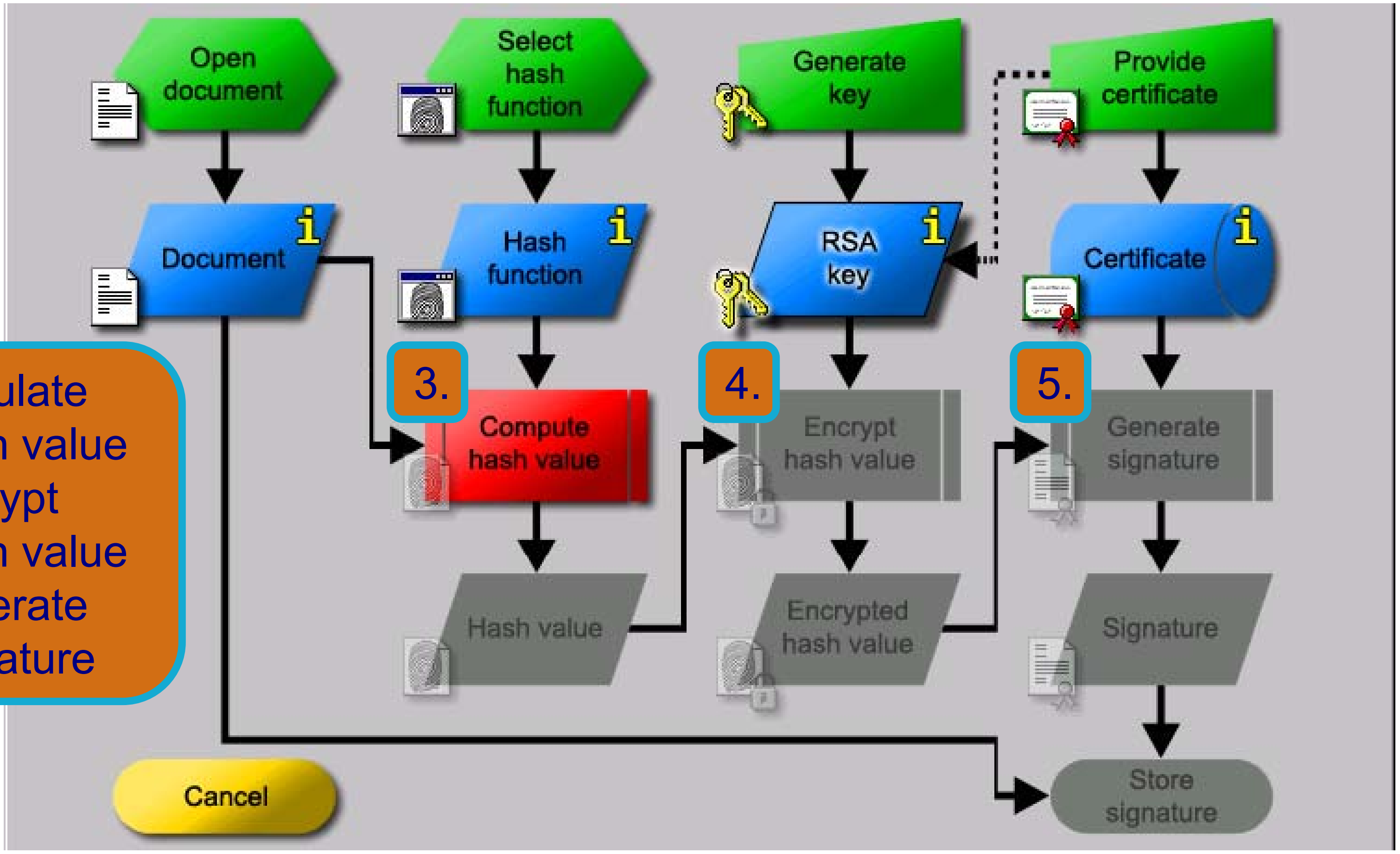
Examples of use
2. Digital signature visualised: Preparation



Hash function:
- MD2
- MD4
- MD5
- SHA
- [✓ SHA-1]
- RIPEMD-160

TECHNISCHE UNIVERSITÄT DARMSTADT

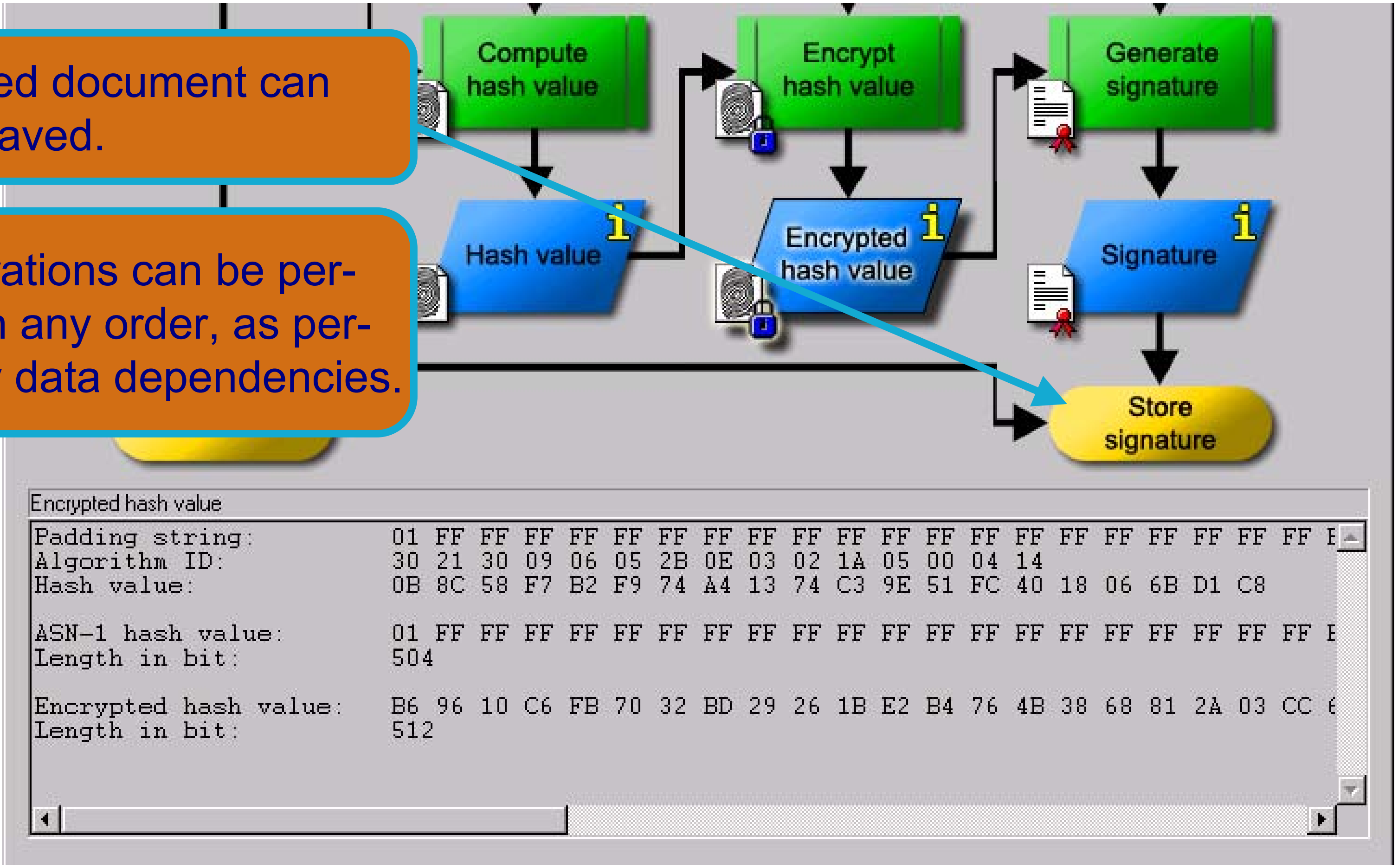
Examples of use
2. Digital signature visualised: Cryptography



3. calculate hash value
4. encrypt hash value
5. generate signature

TECHNISCHE UNIVERSITÄT DARMSTADT

Examples of use
2. Digital signature visualised: Result



Encrypted hash value

```
Padding string:       01 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF F
Algorithm ID:         30 21 30 09 06 05 2B 0E 03 02 1A 05 00 04 14
Hash value:           0B 8C 58 F7 B2 F9 74 A4 13 74 C3 9E 51 FC 40 18 06 6B D1 C8

ASN-1 hash value:     01 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF F
Length in bit:        504

Encrypted hash value: B6 96 10 C6 FB 70 32 BD 29 26 1B E2 B4 76 4B 38 68 81 2A 03 CC 6
Length in bit:        512
```

The signed document can now be saved.

The operations can be per-formed in any order, as per-mitted by data dependencies.

TECHNISCHE UNIVERSITÄT DARMSTADT

Examples of use
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 = 63978486879527143858831415041 (95 bit, 29 decimal digits)
  - public exponent e = 17579
- cipher text (block length = 14):
  - C_1 = 45411667895024938209259253423,
    C_2 = 16597091621432020076311552201,
    C_3 = 46468979279750354732637631044,
    C_4 = 32870167545903741339819671379
- the text shall be deciphered!

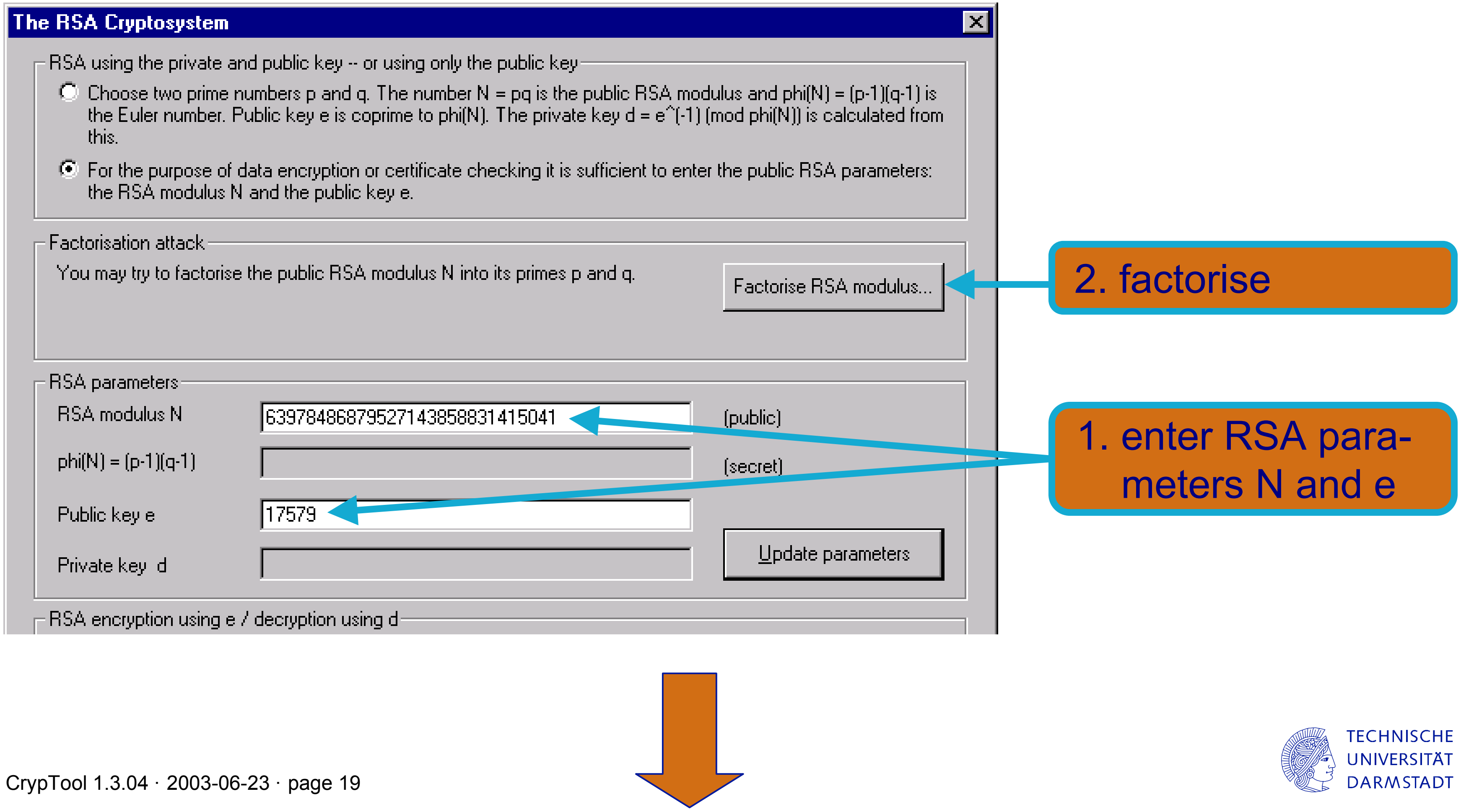
Solution using CrypTool (more detailed in online help examples section):

- enter public parameters into “RSA cryptosystem” (menu indiv. procedures)
- button “factorise the RSA modulus” yields prime factors pq = N
- based on that information 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 is workable for RSA moduli up to 250 bit

TECHNISCHE UNIVERSITÄT DARMSTADT

Examples of use:
3. Short RSA modulus: enter public RSA parameters



The RSA Cryptosystem

RSA using the private and public key -- or using only the public key

( ) Choose two prime numbers p and q. The number N = pq is the public RSA modulus and phi(N) = (p-1)(q-1) is the Euler number. Public key e is coprime to phi(N). The private key d = e^(-1) (mod phi(N)) is calculated from this.

(●) For the purpose of data encryption or certificate checking it is sufficient to enter the public RSA parameters: the RSA modulus N and the public key e.

Factorisation attack

You may try to factorise the public RSA modulus N into its primes p and q.

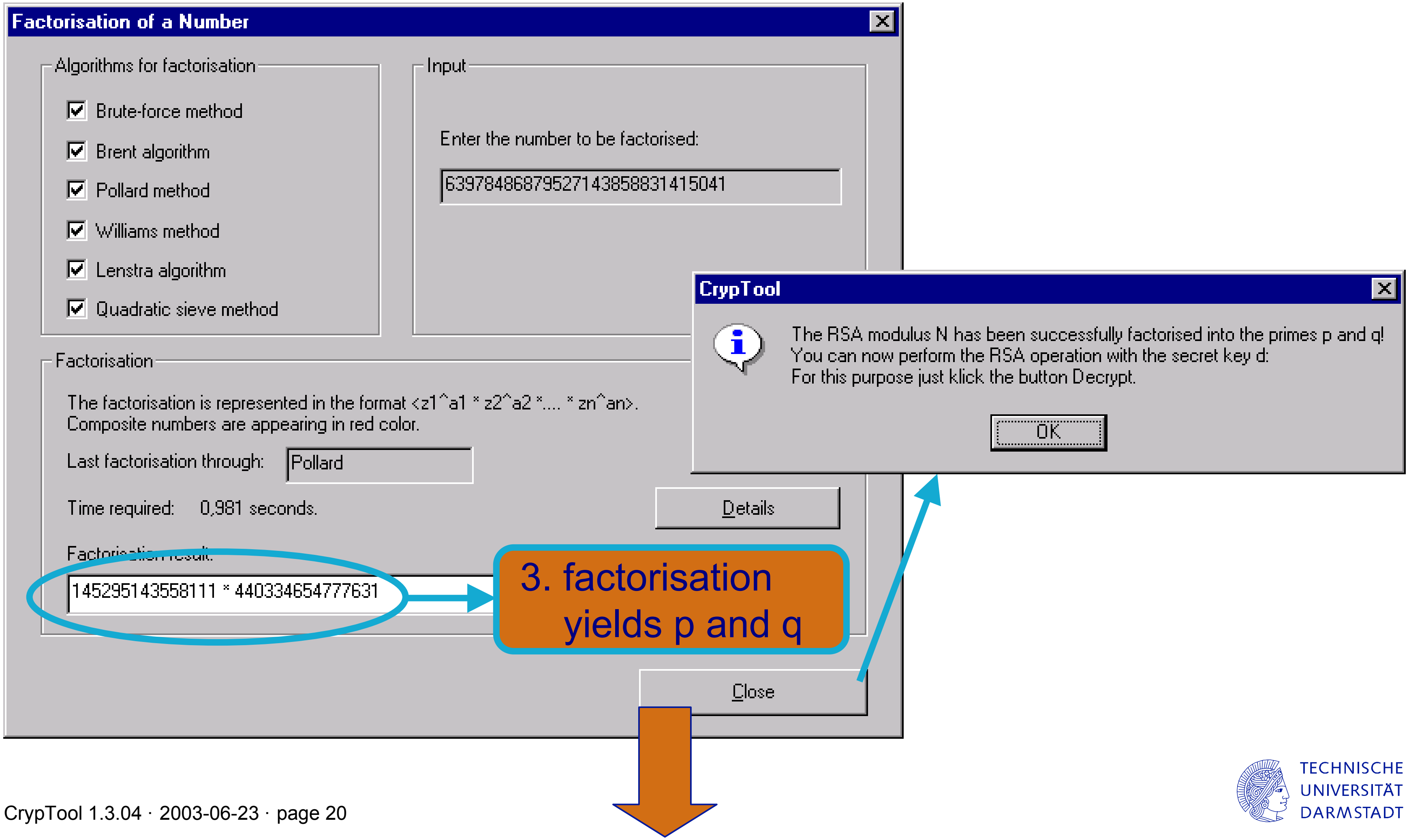
RSA parameters

RSA modulus N	63978486879527143858831415041	(public)
phi(N) = (p-1)(q-1)		(secret)
Public key e	17579	
Private key d		

RSA encryption using e / decryption using d

TECHNISCHE UNIVERSITÄT DARMSTADT

Examples of use: 3. Short RSA modulus: factorise RSA modulus



Factorisation of a Number

Algorithms for factorisation
- ☑ Brute-force method
- ☑ Brent algorithm
- ☑ Pollard method
- ☑ Williams method
- ☑ Lenstra algorithm
- ☑ Quadratic sieve method

Input

Enter the number to be factorised: 63978486879527143858831415041

Factorisation

The factorisation is represented in the format <z1^a1 * z2^a2 *.... * zn^an>.
Composite numbers are appearing in red color.

Last factorisation through: Pollard

Time required: 0,981 seconds.

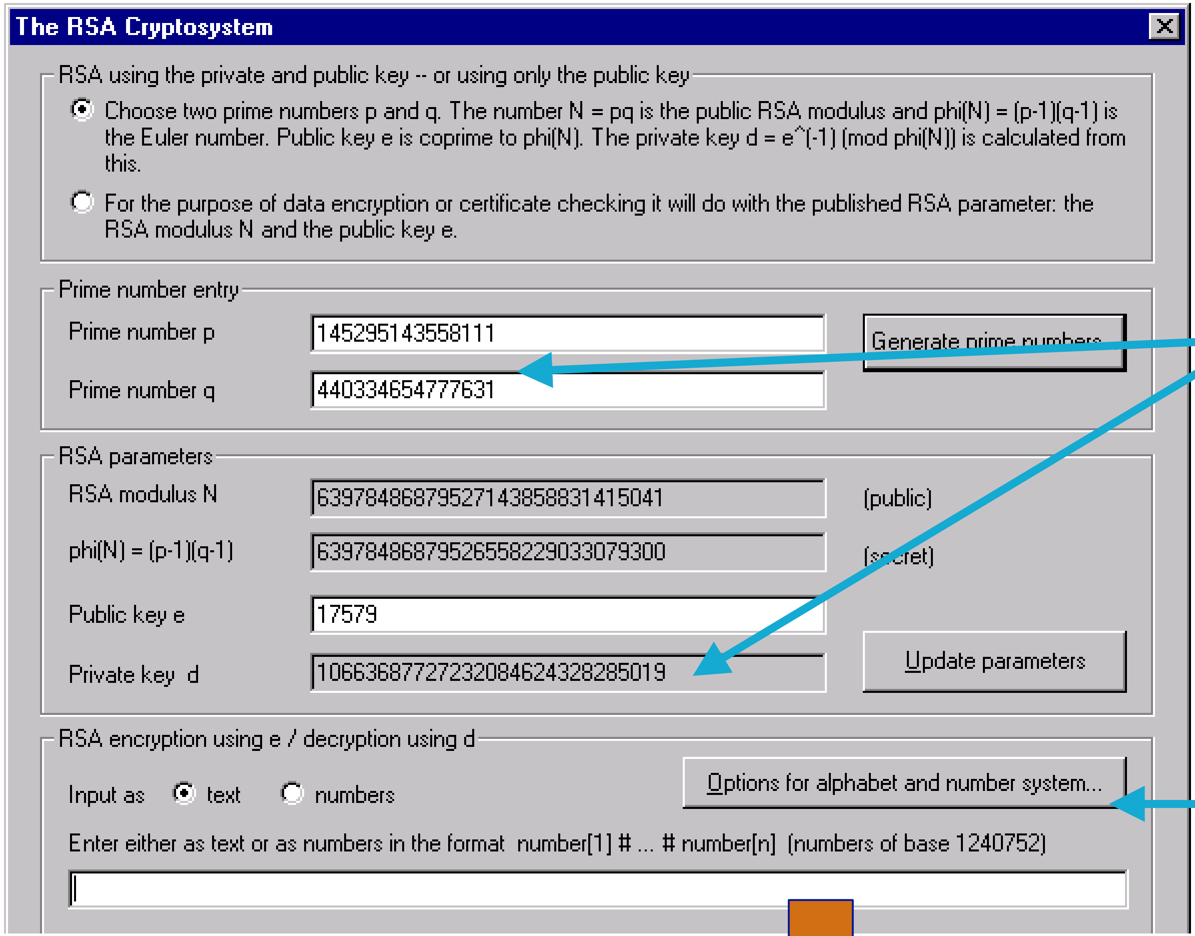
Factorisation result: 145295143558111 * 440334654777631

CrypTool

The RSA modulus N has been successfully factorised into the primes p and q!
You can now perform the RSA operation with the secret key d:
For this purpose just klick the button Decrypt.

TECHNISCHE UNIVERSITÄT DARMSTADT

Examples of use:
3. Short RSA modulus: determine private key d



The RSA Cryptosystem

RSA using the private and public key -- or using only the public key

[✓ Choose two prime numbers p and q. The number N = pq is the public RSA modulus and phi(N) = (p-1)(q-1) is the Euler number. Public key e is coprime to phi(N). The private key d = e^(-1) (mod phi(N)) is calculated from this.]

For the purpose of data encryption or certificate checking it will do with the published RSA parameter: the RSA modulus N and the public key e.

Prime number entry

Prime number p	145295143558111
Prime number q	440334654777631

RSA parameters

RSA modulus N	63978486879527143858831415041	(public)
phi(N) = (p-1)(q-1)	63978486879526558229033079300	(secret)
Public key e	17579	
Private key d	10663687727232084624328285019	

RSA encryption using e / decryption using d

Input as [✓ text] numbers

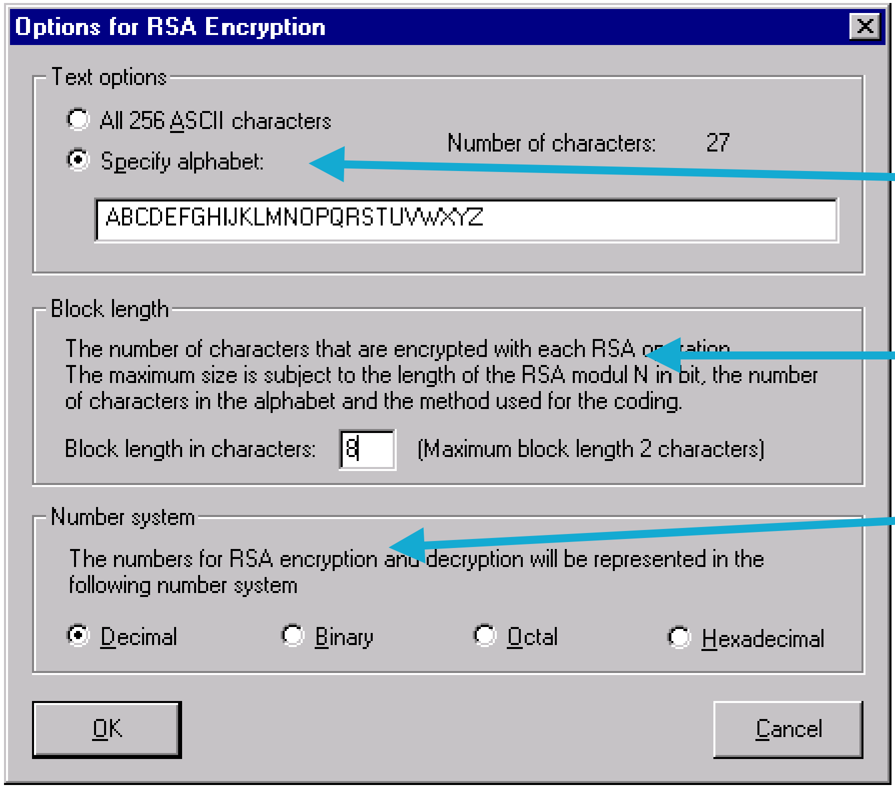
Enter either as text or as numbers in the format number[1] # ... # number[n] (numbers of base 1240752)

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

5. adjust options

TECHNISCHE UNIVERSITÄT DARMSTADT

Examples of use:
3. Short RSA modulus: adjust options



Options for RSA Encryption

Text options

( ) All 256 ASCII characters

(●) Specify alphabet:

Number of characters: 27

ABCDEFGHIJKLMNOPQRSTUVWXYZ

Block length

The number of characters that are encrypted with each RSA operation.
The maximum size is subject to the length of the RSA modul N in bit, the number of characters in the alphabet and the method used for the coding.

Block length in characters: 8 (Maximum block length 2 characters)

Number system

The numbers for RSA encryption and decryption will be represented in the following number system

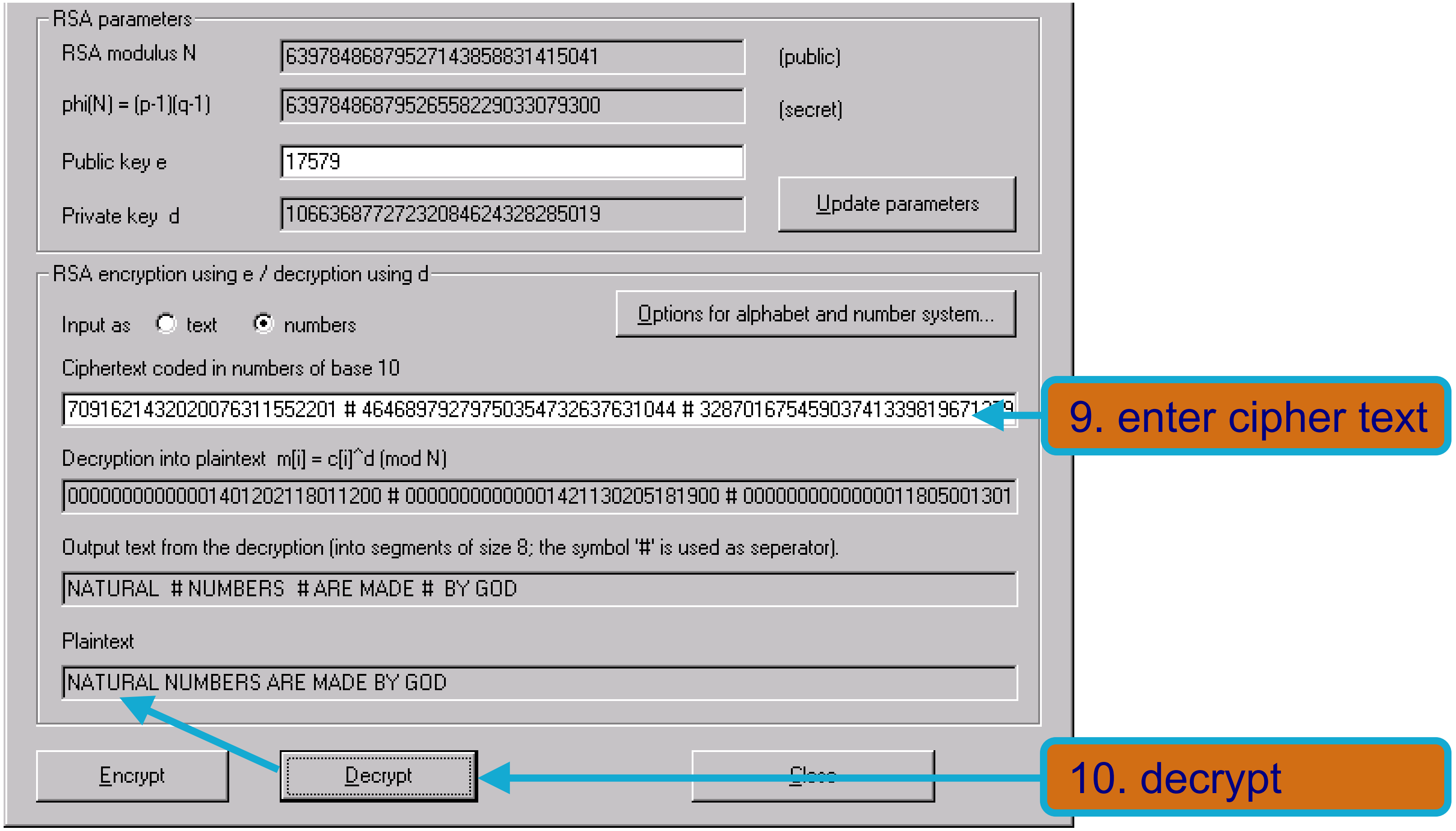
[✓ Decimal] Binary Octal Hexadecimal

6. select alphabet

7. select coding method

8. select block length

Examples of use:
3. Short RSA modulus: decrypt cipher text


RSA parameters

RSA modulus N: 63978486879527143858831415041 (public)

phi(N) = (p-1)(q-1): 63978486879526558229033079300 (secret)

Public key e: 17579

Private key d: 10663687727232084624328285019

RSA encryption using e / decryption using d

Input as ( ) text [✓ numbers]

Ciphertext coded in numbers of base 10

70916214320200763115522201 # 46468979279750354732637631044 # 32870167545903741339819671[illegible]

Decryption into plaintext m[i] = c[i]^d (mod N)

0000000000001401202118011200 # 0000000000001421130205181900 # 0000000000000011805001301

Output text from the decryption (into segments of size 8; the symbol '#' is used as seperator).

NATURAL # NUMBERS # ARE MADE # BY GOD

Plaintext

NATURAL NUMBERS ARE MADE BY GOD

TECHNISCHE UNIVERSITÄT DARMSTADT

Examples of use
4. Analysis of encryption used in the PSION 5 PDA

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

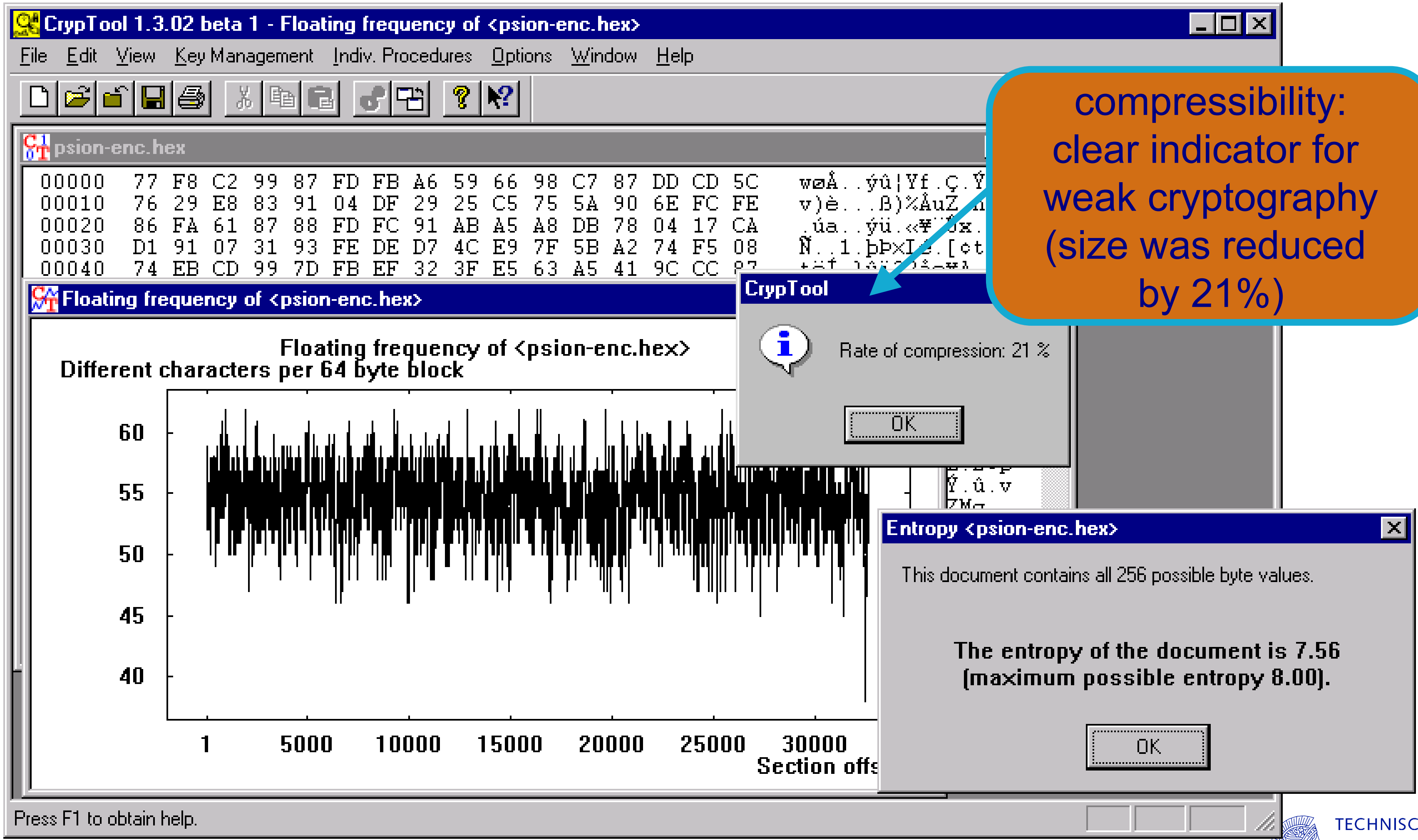
Procedure

- pre-analysis
  - entropy
  - floating entropy
  - compression test

  $\Big\} \Rightarrow$ probably classical encryption algorithm
- auto-correlation
- try out automatic analysis with classical methods

TECHNISCHE UNIVERSITÄT DARMSTADT

Examples of use
4. PSION PDA: determine entropy, compression test



CrypTool 1.3.02 beta 1 - Floating frequency of <psion-enc.hex>
File Edit View Key Management Indiv. Procedures Options Window Help
psion-enc.hex
Floating frequency of <psion-enc.hex>
Different characters per 64 byte block
60
55
50
45
40
1
5000
10000
15000
20000
25000
30000
Section offs
CrypTool
OK
Entropy <psion-enc.hex>
OK
Press F1 to obtain help.

```
00000  77 F8 C2 99 87 FD FB A6 59 66 98 C7 87 DD CD 5C
00010  76 29 E8 83 91 04 DF 29 25 C5 75 5A 90 6E FC FE
00020  86 FA 61 87 88 FD FC 91 AB A5 A8 DB 78 04 17 CA
00030  D1 91 07 31 93 FE DE D7 4C E9 7F 5B A2 74 F5 08
00040  74 EB CD 99 7D FB EF 32 3F E5 63 A5 41 9C CC 87
```

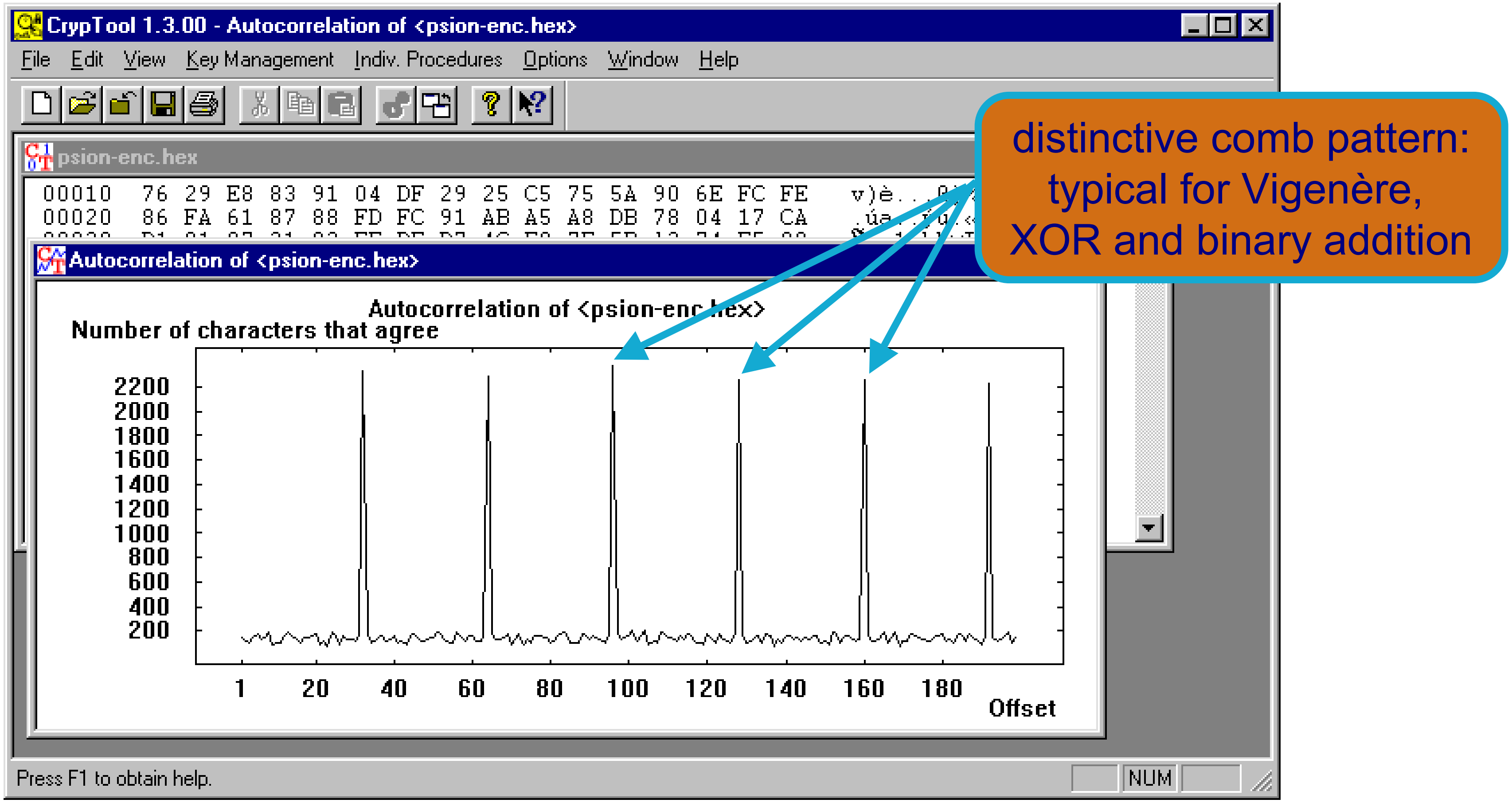
Rate of compression: 21 %

This document contains all 256 possible byte values.

The entropy of the document is 7.56 (maximum possible entropy 8.00).

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

Examples of use
4. PSION PDA: determine auto-correlation


CrypTool 1.3.00 - Autocorrelation of <psion-enc.hex>
File Edit View Key Management Indiv. Procedures Options Window Help
psion-enc.hex
Autocorrelation of <psion-enc.hex>
Autocorrelation of <psion-enc.hex>
Number of characters that agree
2200
2000
1800
1600
1400
1200
1000
800
600
400
200
1
20
40
60
80
100
120
140
160
180
Offset
Press F1 to obtain help.
NUM
distinctive comb pattern: typical for Vigenère, XOR and binary addition

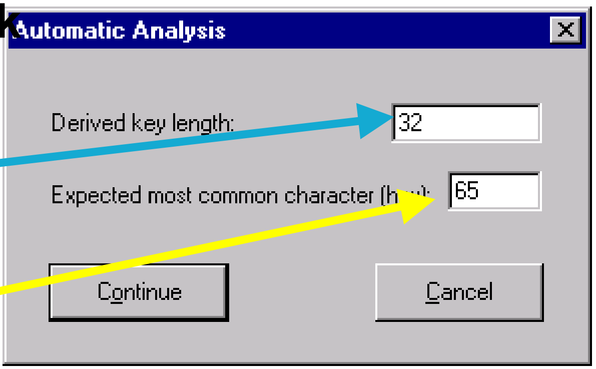
```
00010   76 29 E8 83 91 04 DF 29 25 C5 75 5A 90 6E FC FE   v)è...
00020   86 FA 61 87 88 FD FC 91 AB A5 A8 DB 78 04 17 CA   .úa...
```

Examples of use
4. PSION PDA: automatic analysis

Automatic analysis using XOR: does not work

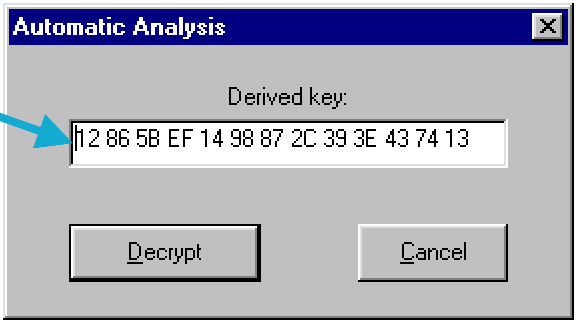
Automatic analysis using Binary Addition:

- 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)
- Results: good, but not perfect


Automatic Analysis

Derived key length: 32

Expected most common character (hex): 65


Automatic Analysis

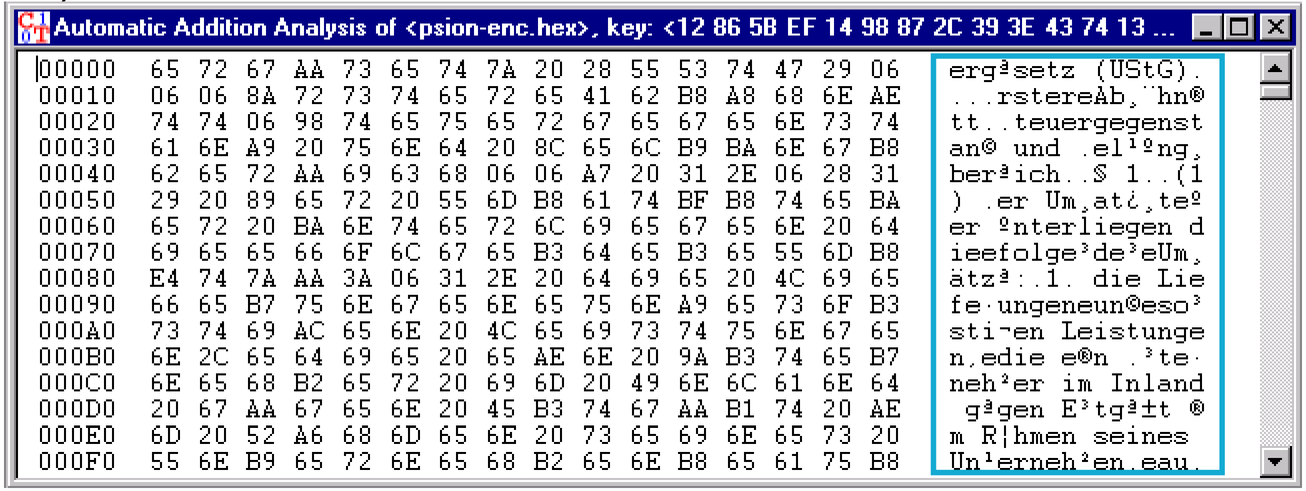
Derived key: 12 86 5B EF 14 98 87 2C 39 3E 43 74 13

TECHNISCHE UNIVERSITÄT DARMSTADT

Examples of use
4. PSION PDA: results of automatic analysis

Results of automatic analysis with assumption “binary addition”:

- results good, but not perfect: 24 out of 32 key bytes correct.
- the key length 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)


Automatic Addition Analysis of <psion-enc.hex>, key: <12 86 5B EF 14 98 87 2C 39 3E 43 74 13 ...

```
00000   65 72 67 AA 73 65 74 7A 20 28 55 53 74 47 29 06   ergªsetz (UStG).
00010   06 06 8A 72 73 74 65 72 65 41 62 B8 A8 68 6E AE   ...rstereAb¸¨hn®
00020   74 74 06 98 74 65 75 65 72 67 65 67 65 6E 73 74   tt..teuergegenst
00030   61 6E A9 20 75 6E 64 20 8C 65 6C B9 BA 6E 67 B8   an© und .el¹ºng¸
00040   62 65 72 AA 69 63 68 06 06 A7 20 31 2E 06 28 31   berªich..§ 1..(1
00050   29 20 89 65 72 20 55 6D B8 61 74 BF B8 74 65 BA   ) .er Um¸at¿¸teº
00060   65 72 20 BA 6E 74 65 72 6C 69 65 67 65 6E 20 64   er ºnterliegen d
00070   69 65 65 66 6F 6C 67 65 B3 64 65 B3 65 55 6D B8   ieefolge³de³eUm¸
00080   E4 74 7A AA 3A 06 31 2E 20 64 69 65 20 4C 69 65   ätzª:.1. die Lie
00090   66 65 B7 75 6E 67 65 6E 65 75 6E A9 65 73 6F B3   fe·ungeneun©eso³
000A0   73 74 69 AC 65 6E 20 4C 65 69 73 74 75 6E 67 65   sti¬en Leistunge
000B0   6E 2C 65 64 69 65 20 65 AE 6E 20 9A B3 74 65 B7   n,edie e®n .³te·
000C0   6E 65 68 B2 65 72 20 69 6D 20 49 6E 6C 61 6E 64   neh²er im Inland
000D0   20 67 AA 67 65 6E 20 45 B3 74 67 AA B1 74 20 AE    gªgen E³tgª±t ®
000E0   6D 20 52 A6 68 6D 65 6E 20 73 65 69 6E 65 73 20   m R¦hmen seines 
000F0   55 6E B9 65 72 6E 65 68 B2 65 6E B8 65 61 75 B8   Un¹erneh²en¸eau¸
```

TECHNISCHE UNIVERSITÄT DARMSTADT

Examples of use
4. PSION PDA: determining the remaining key bytes

Copy key to clipboard during automatic analysis

In automatic analysis hexdump,

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

In encrypted initial file hexdump,

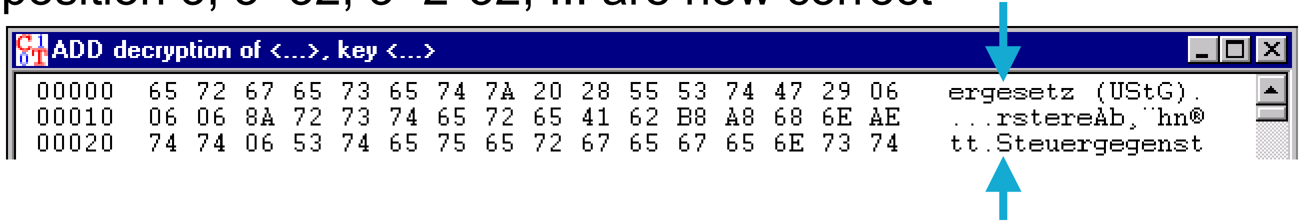
- determine initial bytes from the calculated byte positions: 0x99
- calculate correct key bytes with CALC.EXE: 0x99 - 0x65 = 0x34

Correct key from the clipboard

- 12865B341498872C393E43741396A45670235E111E907AB7C0841...

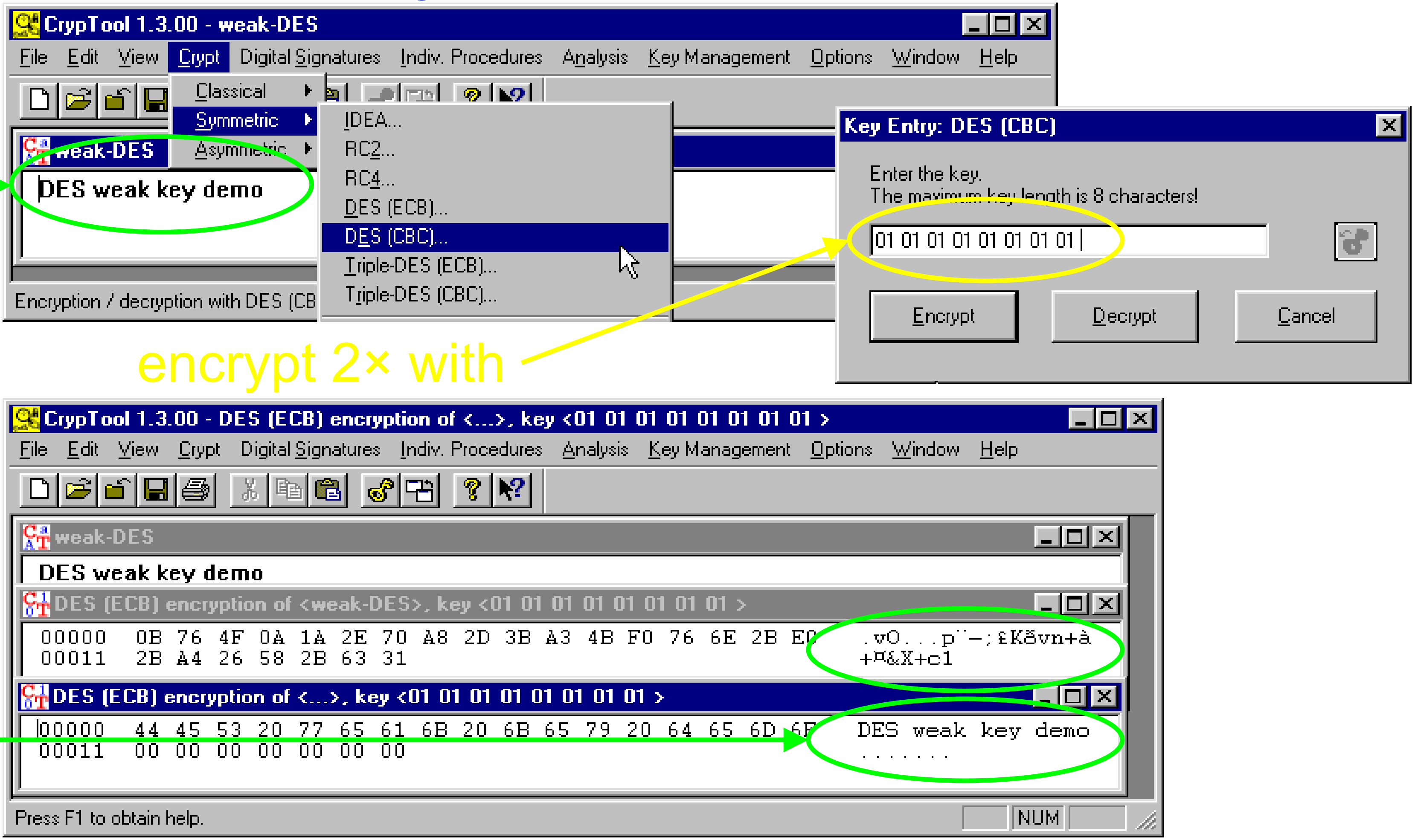
Decrypt encrypted initial document using binary addition

- bytes at position 3, 3+32, 3+2*32, ... are now correct


ADD decryption of <...>, key <...>

```
00000  65 72 67 65 73 65 74 7A 20 28 55 53 74 47 29 06   ergesetz (UStG).
00010  06 06 8A 72 73 74 65 72 65 41 62 B8 A8 68 6E AE   ...rstereAb¸¨hn®
00020  74 74 06 53 74 65 75 65 72 67 65 67 65 6E 73 74   tt.Steuergegenst
```

Examples of use
5. Weak DES keys


CrypTool 1.3.00 - weak-DES
File Edit View Crypt Digital Signatures Indiv. Procedures Analysis Key Management Options Window Help
Classical
Symmetric
Asymmetric
IDEA...
RC2...
RC4...
DES (ECB)...
DES (CBC)...
Triple-DES (ECB)...
Triple-DES (CBC)...
weak-DES
Encryption / decryption with DES (CB
Key Entry: DES (CBC)
Encrypt
Decrypt
Cancel
encrypt 2× with
CrypTool 1.3.00 - DES (ECB) encryption of <...>, key <01 01 01 01 01 01 01 01 >
weak-DES
DES (ECB) encryption of <weak-DES>, key <01 01 01 01 01 01 01 01 >
DES (ECB) encryption of <...>, key <01 01 01 01 01 01 01 01 >
Press F1 to obtain help.
NUM

DES weak key demo

Enter the key.
The maximum key length is 8 characters!

01 01 01 01 01 01 01 01

DES weak key demo

```
00000  0B 76 4F 0A 1A 2E 70 A8 2D 3B A3 4B F0 76 6E 2B E0    .vO...p¨–;£Kövn+à
00011  2B A4 26 58 2B 63 31                                   +¤&X+c1
```

```
00000  44 45 53 20 77 65 61 6B 20 6B 65 79 20 64 65 6D 6F    DES weak key demo
00011  00 00 00 00 00 00 00                                   .......
```

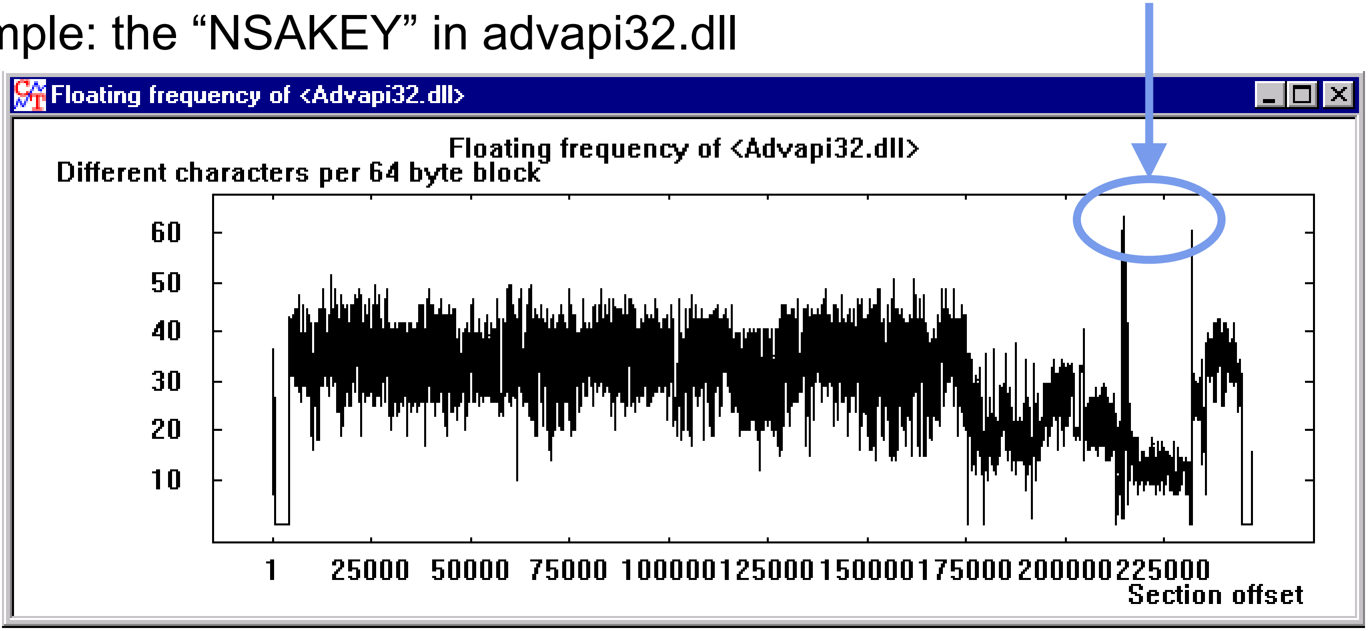
TECHNISCHE UNIVERSITÄT DARMSTADT

Examples of use
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 recognised as peaks in the "floating frequency"
- example: the "NSAKEY" in advapi32.dll



TECHNISCHE UNIVERSITÄT DARMSTADT

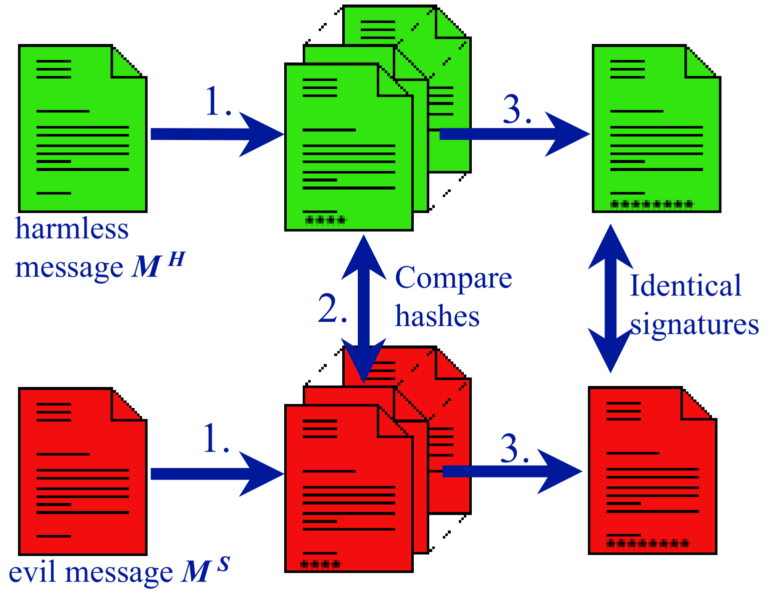
Examples of use
7. Attack on digital signature: idea

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 by Jan Blumenstein as part of his bachelor thesis "*Methods and tools for attacks on digital signatures*" (German), 2003.

Examples of use
7. Attack on digital signature: idea (2)



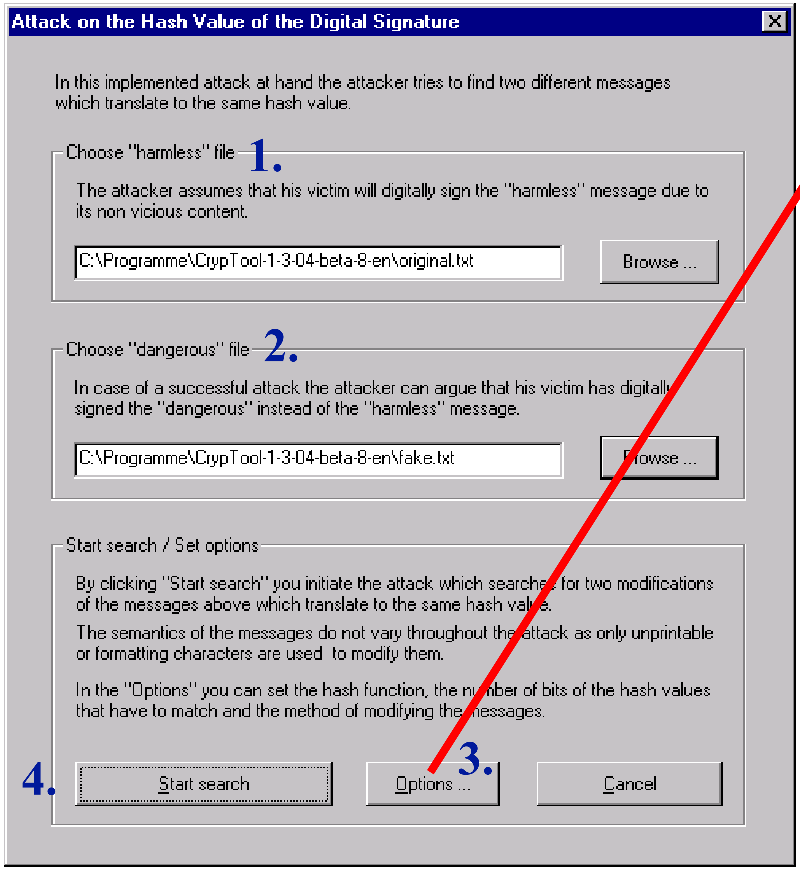
1. **Modification:** starting from a message M create N different messages M_1, ..., M_N with the same “content” as M.
2. **Search:** find modified messages M_i^H und M_j^S with the same hash value.
3. **Attack:** the signatures of those two documents M_i^H und M_j^S are the same.

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 : $N \approx 2^{n/2}$

TECHNISCHE UNIVERSITÄT DARMSTADT

Examples of use
7. Attack on digital signature: attack



Attack on the Hash Value of the Digital Signature

In this implemented attack at hand the attacker tries to find two different messages which translate to the same hash value.

Choose "harmless" file **1.**

The attacker assumes that his victim will digitally sign the "harmless" message due to its non vicious content.

C:\Programme\CrypTool-1-3-04-beta-8-en\original.txt

Choose "dangerous" file **2.**

In case of a successful attack the attacker can argue that his victim has digitally signed the "dangerous" instead of the "harmless" message.

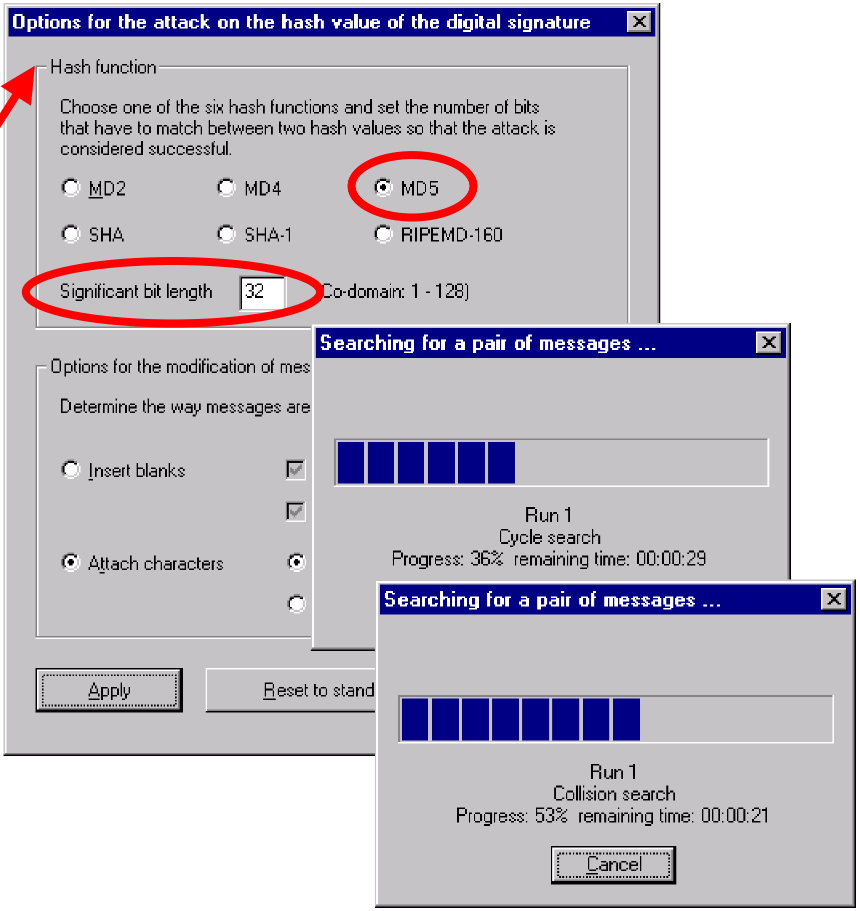
C:\Programme\CrypTool-1-3-04-beta-8-en\fake.txt

Start search / Set options

By clicking "Start search" you initiate the attack which searches for two modifications of the messages above which translate to the same hash value.

The semantics of the messages do not vary throughout the attack as only unprintable or formatting characters are used to modify them.

In the "Options" you can set the hash function, the number of bits of the hash values that have to match and the method of modifying the messages.



Options for the attack on the hash value of the digital signature

Hash function

Choose one of the six hash functions and set the number of bits that have to match between two hash values so that the attack is considered successful.

MD2 MD4 [✓ MD5]

SHA SHA-1 RIPEMD-160

Significant bit length: 32 (Co-domain: 1 - 128)

Options for the modification of mes

Determine the way messages are

Insert blanks

[✓ Attach characters]

Searching for a pair of messages ...

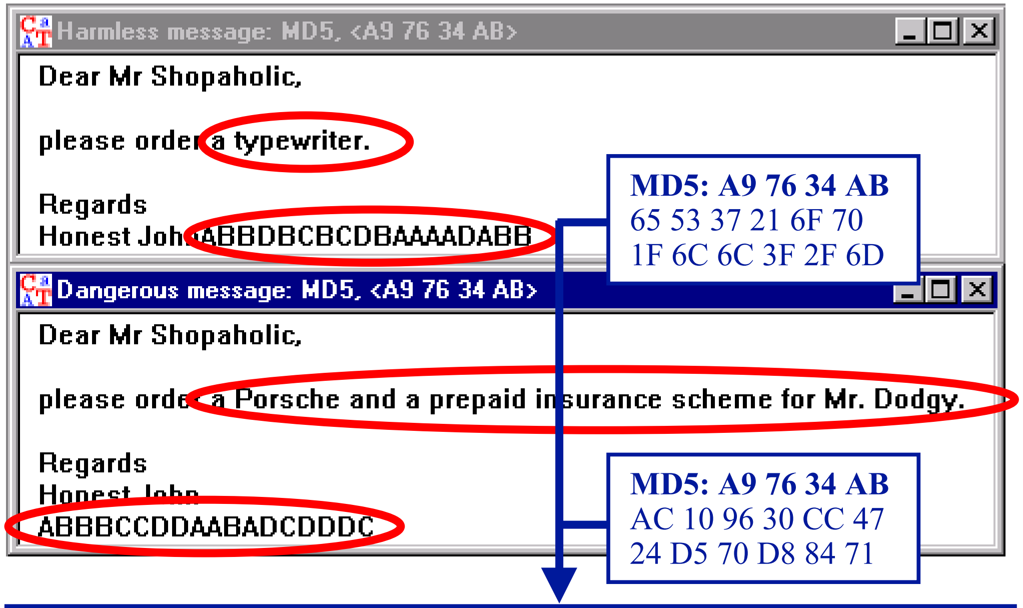
Run 1
Cycle search
Progress: 36% remaining time: 00:00:29

Searching for a pair of messages ...

Run 1
Collision search
Progress: 53% remaining time: 00:00:21

TECHNISCHE UNIVERSITÄT DARMSTADT

Examples of use
7. Attack on digital signature: results



Harmless message: MD5, <A9 76 34 AB>

Dear Mr Shopaholic,

please order a typewriter.

Regards
Honest JohnABBDBCBCDBAAAADABB

MD5: A9 76 34 AB
65 53 37 21 6F 70
1F 6C 6C 3F 2F 6D

Dangerous message: MD5, <A9 76 34 AB>

Dear Mr Shopaholic,

please order a Porsche and a prepaid insurance scheme for Mr. Dodgy.

Regards
Honest John
ABBBCCDDAABADCDDDC

MD5: A9 76 34 AB
AC 10 96 30 CC 47
24 D5 70 D8 84 71

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!

TECHNISCHE UNIVERSITÄT DARMSTADT

Further development

Work in process

- visualisation of challenge-response authentication
- attack on single-sided authentication with CR and weak encryption
- mass pattern search

Planned for near future

- visualisation of SSL protocol
- visualisation of Man-in-the-Middle attack
- demonstration of a side-channel attack

Planned for remote future

- visualisation of different security protocols (e.g. Kerberos)
- visualisation of attacks on these different security protocols
- port to Linux or Java
- many more ideas can be found in the readme file, chapter 6

Contact addresses

Prof. Dr. Claudia Eckert
Technical University Darmstadt
Faculty of Computer Science
IT Security
Wilhelminenstr. 7
64283 Darmstadt, Germany
claudia.eckert@
sec.informatik.tu-darmstadt.de

Bernhard Esslinger
- University of Siegen
  Faculty of Economics
- Deutsche Bank AG
  Head of Information Security

bernhard.esslinger@db.com
besslinger@web.de

Thorsten Clausius
Technical University Darmstadt
thorsten.clausius@
sec.informatik.tu-darmstadt.de

Jörg Cornelius Schneider
Deutsche Bank AG
joerg-cornelius.schneider@db.com
js@joergschneider.com

www.cryptool.de
www.cryptool.org
www.cryptool.com
Mailing list: cryptool-list@sec.informatik.tu-darmstadt.de