Module 4: Cryptology for IoT

Cryptology for IoT

Modules M4, M7, M9
Session of 26th April, 2022.

M4. Introduction to the modules
M4.1 Introduction to the Cryptology
M4.2 Introduction to Cryptool CT2

Prof.: Guillermo Botella

Sec

Module 4: Cryptology for IoT

Cryptology for IoT

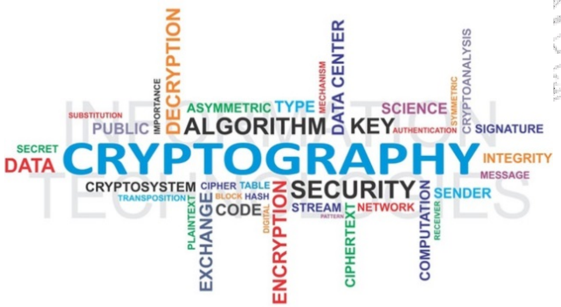
Modules M4, M7, M9
Session of 26th April, 2022.

M4. Introduction to the modules
M4.1 Introduction to the Cryptology
M4.2 Introduction to Cryptool CT2

Prof.: Guillermo Botella

Sec

Objetives (M4, M7, M9)



Objectives:

- Learn why cryptology is important
- Study how crypto primitives' work
- Study how to use them correctly and reason about security in general and particularly in the IoT context

Recommendations:

- Stop and rethink about the concepts
- Complete the exercises and assignments. Answer the questions

Scheduling of the M4, M7, M9 modules

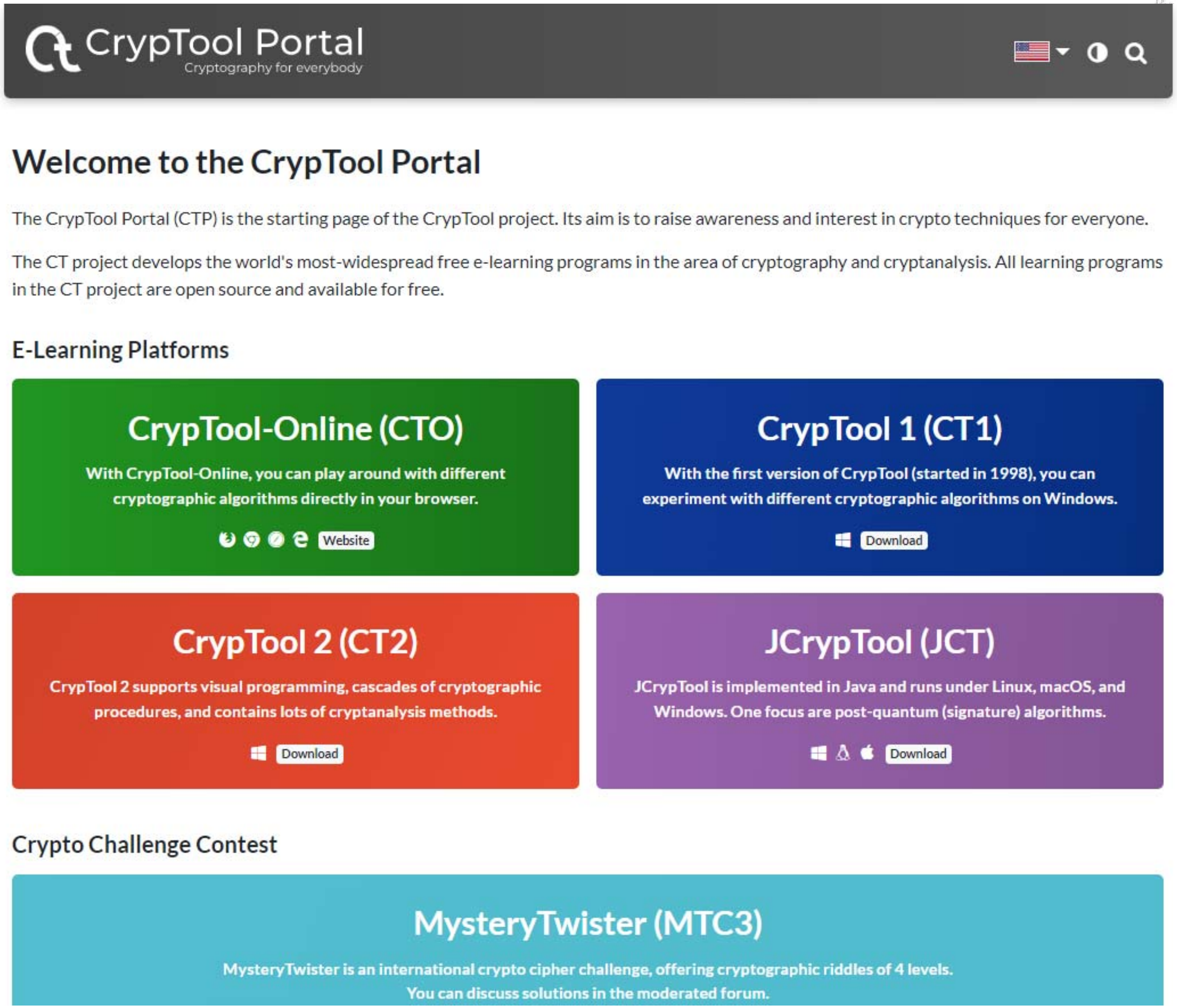
Tentative Scheduling (subject to modifications according to the progress of the module)

- M4 Module (2nd week of Security Module): Introduction & Classic Cryptology
- M7 Module (4th week of Security Module): Modern Cryptology
- M9 Module (6th week of Security Module): Important topics and Applications

Organization of the M4, M7, M9 modules

- 1- Briefing at the beginning of the session
- 2- Theoretical content will be explained with slides and additional material (basic and supplementary material will be placed at this link:)
  - https://www.dropbox.com/sh/wskrslivhe55nx1/AABKnnIhQumuuJ5BYRy6Ciwka
- 3- Practice using Cryptool lab to complete the assignments
  - – Same groups for the whole Security Module
- 4- A test/quiz/battery of questions will be carried out frequently at the end of the session (individual)
  - – It is important to fill tests/quizzes introducing just **individual email** at the beginning!

Laboratoraries for Crypto



CrypTool Portal

Cryptography for everybody

Welcome to the CrypTool Portal

The CrypTool Portal (CTP) is the starting page of the CrypTool project. Its aim is to raise awareness and interest in crypto techniques for everyone.

The CT project develops the world's most-widespread free e-learning programs in the area of cryptography and cryptanalysis. All learning programs in the CT project are open source and available for free.

E-Learning Platforms

CrypTool-Online (CTO)

With CrypTool-Online, you can play around with different cryptographic algorithms directly in your browser.

CrypTool 1 (CT1)

With the first version of CrypTool (started in 1998), you can experiment with different cryptographic algorithms on Windows.

CrypTool 2 (CT2)

CrypTool 2 supports visual programming, cascades of cryptographic procedures, and contains lots of cryptanalysis methods.

JCrypTool (JCT)

JCrypTool is implemented in Java and runs under Linux, macOS, and Windows. One focus are post-quantum (signature) algorithms.

Crypto Challenge Contest

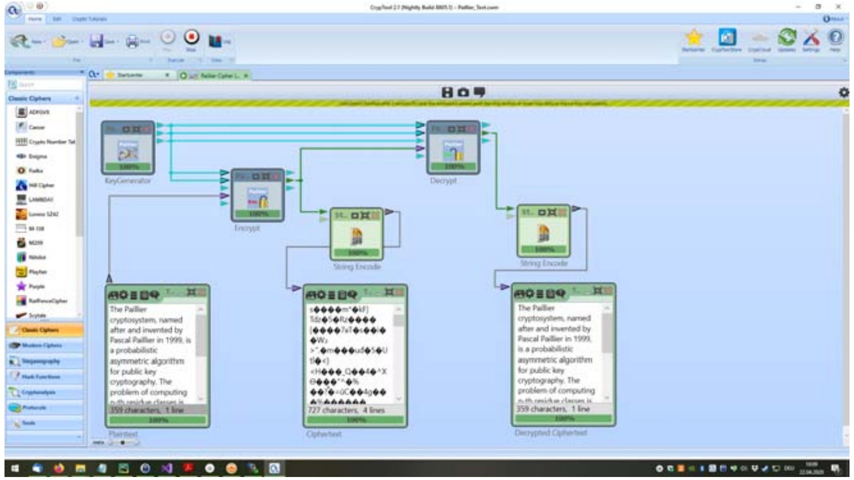
MysteryTwister (MTC3)

MysteryTwister is an international crypto cipher challenge, offering cryptographic riddles of 4 levels. You can discuss solutions in the moderated forum.

Module 4: Cryptology for IoT

Laboratoraries for Crypto

- Using mainly Cryptool CT2 https://www.cryptool.org/en/ct2-download/
- CT2: Port and redesign of the C++ version with C# / WPF / Visual Studio / .NET
- Allows visual programming and distributed calculations (CrypCloud)
- Multilingual environment. English and Chinese language allowed! ☺



The Paillier cryptosystem, named after and invented by Pascal Paillier in 1999, is a probabilistic asymmetric algorithm for public key cryptography. The problem of computing n-th residue classes is

359 characters, 1 line

727 characters, 4 lines

The Paillier cryptosystem, named after and invented by Pascal Paillier in 1999, is a probabilistic asymmetric algorithm for public key cryptography. The problem of computing n-th residue classes is

359 characters, 1 line

Sec

Tests/quizzes/Activities

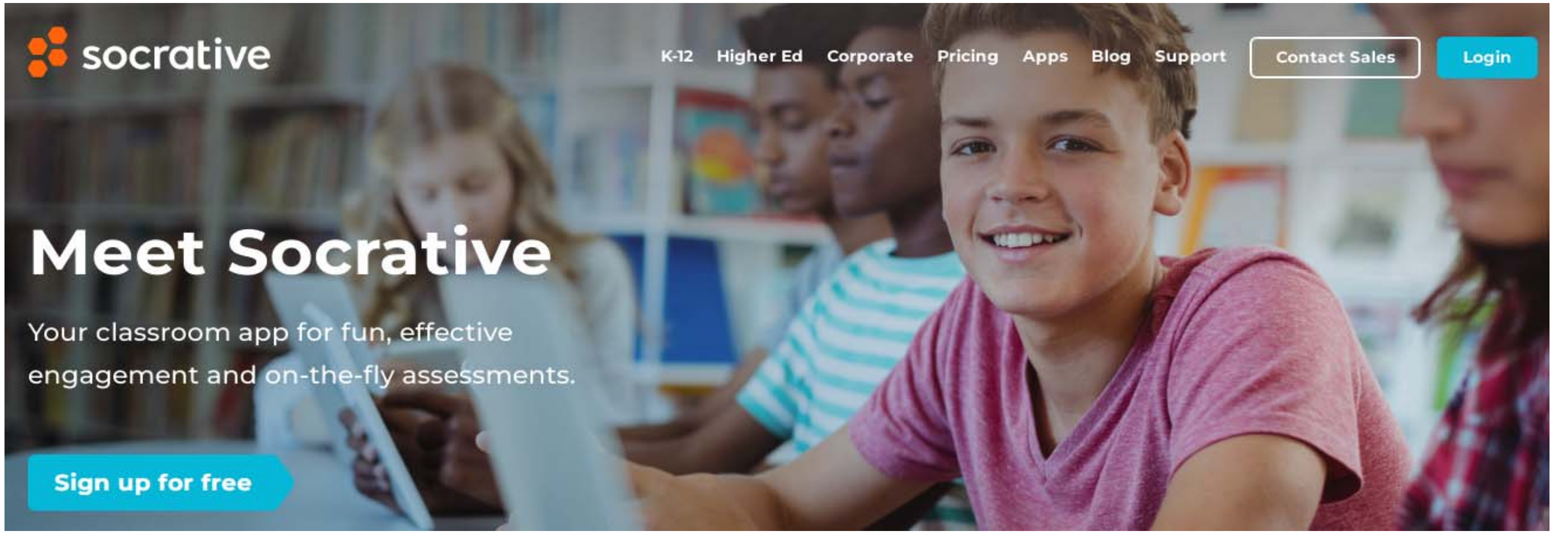
– Socrative student login → https://b.socrative.com/login/student/
  - ROOM KEY: *It will provided when we start*

– It will take around 15-20 mins to be completed

– No need to download the App/register

– It is important to enter at the beginning of each test the personal email that you use in this course

– Honor code: Individual Assignments!



Meet Socrative

Your classroom app for fun, effective engagement and on-the-fly assessments.

Cryptology for IoT

Modules M4, M7, M9
Session of 26th April, 2022.

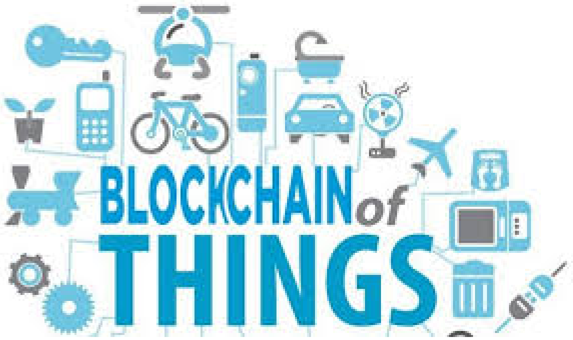
M4. Introduction to the modules

M4.1 Introduction to the Cryptology

M4.2 Introduction to Cryptool CT2

Prof.: Guillermo Botella

Where is Cryptography?



Secure communication:

- web traffic: HTTPS
- wireless traffic: WPA2, 4G/5G, Bluetooth

Encrypting files on disk: EFS, TrueCrypt

Content protection (e.g. DVD, Blu-ray): CSS, AACS

User authentication

… and much much more (Crypto is in everywhere)

Terminology and Background
Threats to Messages

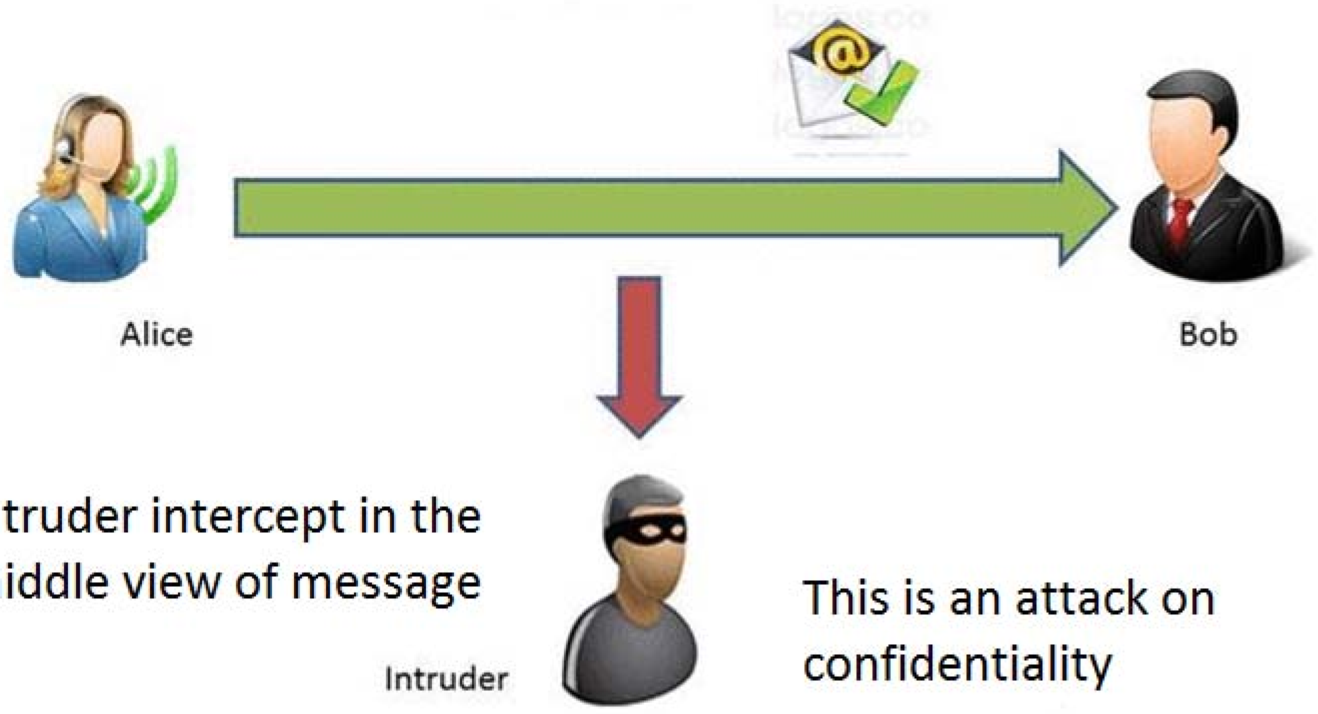
- Interception
- Interruption
  - Blocking msgs
- Modification
- Fabrication

“A threat is blocked by control of a vulnerability”

[Pfleeger & Pfleeger]

Terminology and Background
Threats to Messages

- Interception



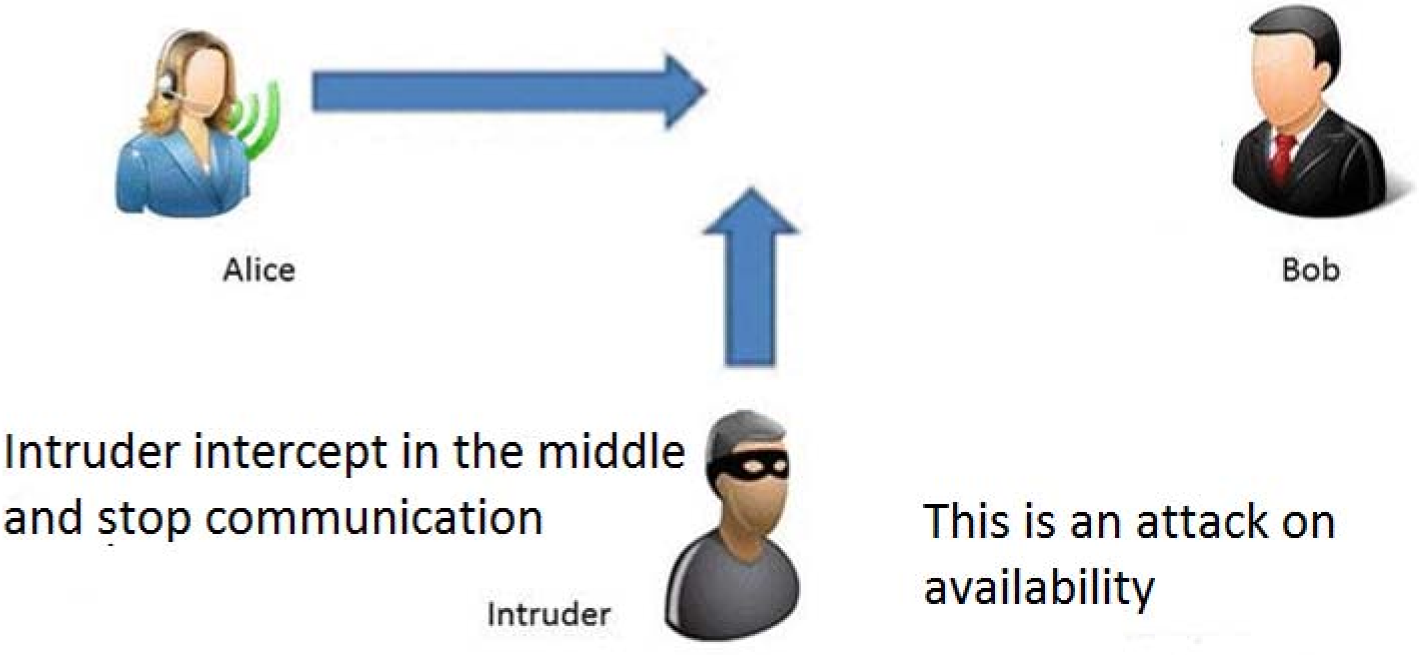
Intruder intercept in the middle view of message

This is an attack on confidentiality

A passive intruder

Terminology and Background
Threats to Messages

- Interruption



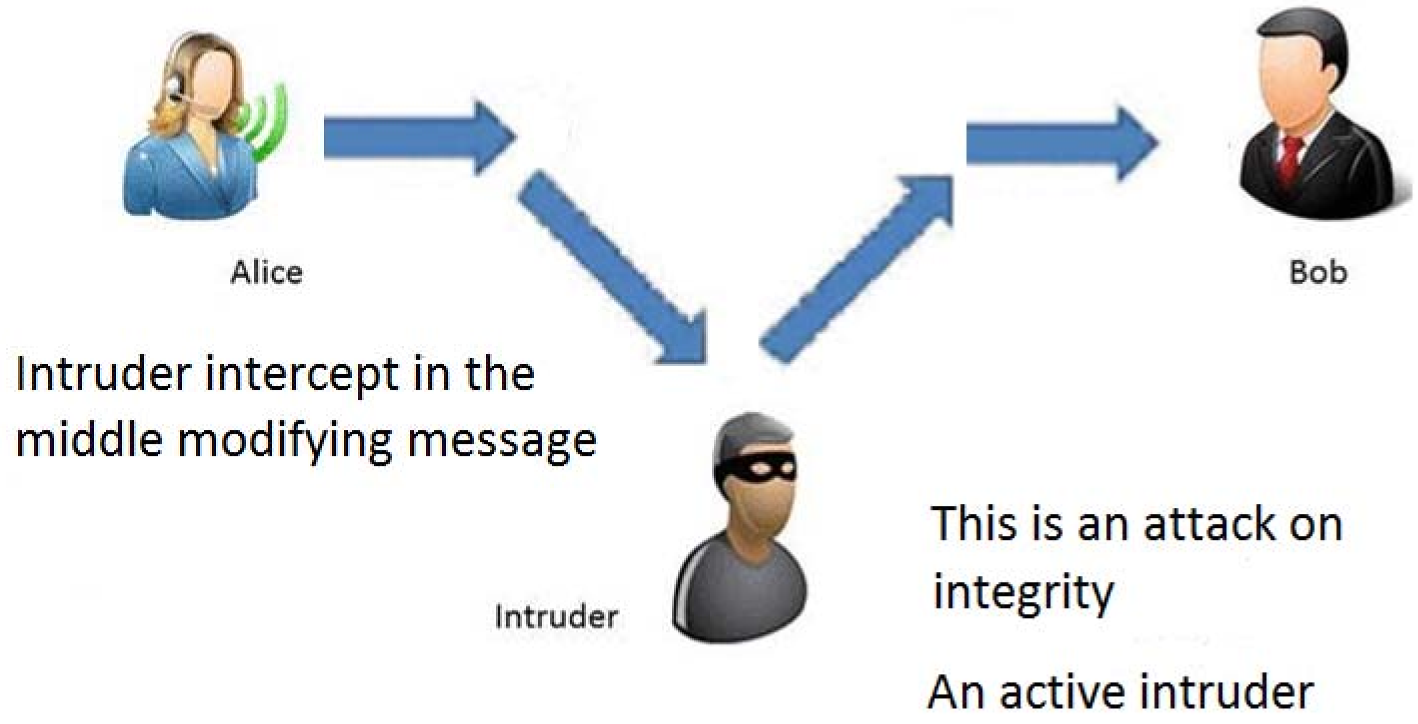
Intruder intercept in the middle and stop communication

This is an attack on availability

An active intruder

Terminology and Background
Threats to Messages

- Modification



Intruder intercept in the middle modifying message

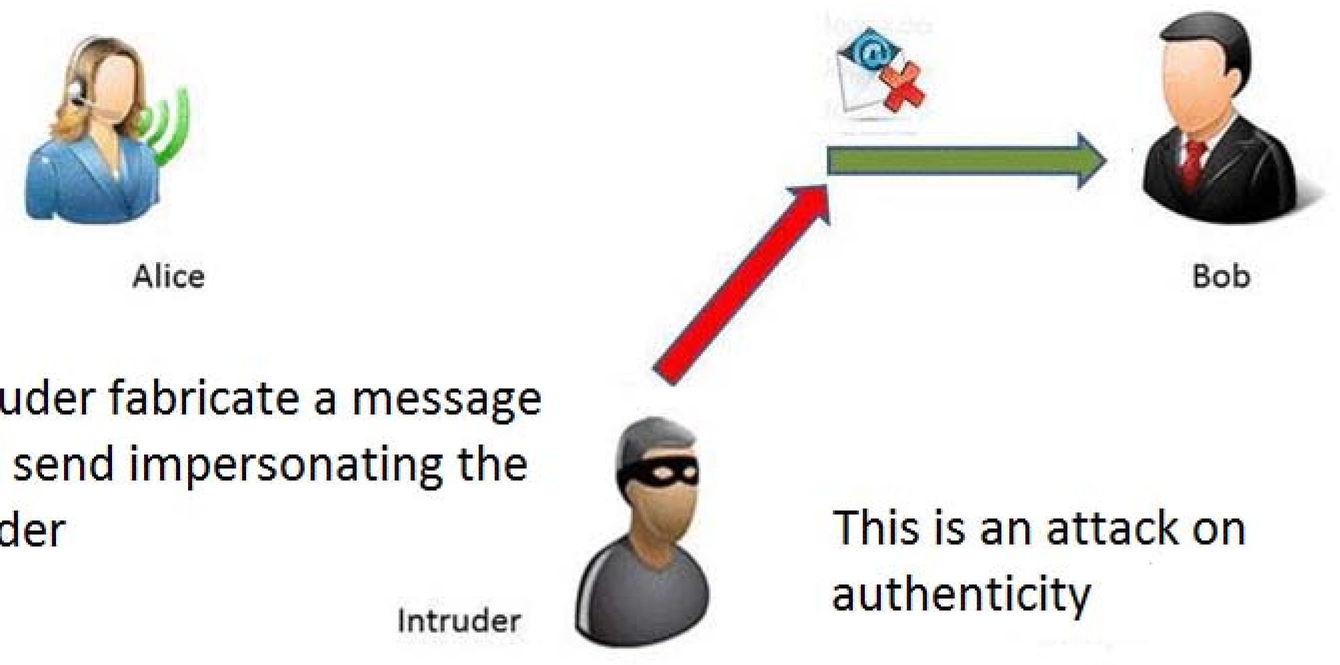
This is an attack on integrity

An active intruder

Module 4: Cryptology for IoT

Terminology and Background
Threats to Messages

- Fabrication



Intruder fabricate a message and send impersonating the sender

This is an attack on authenticity

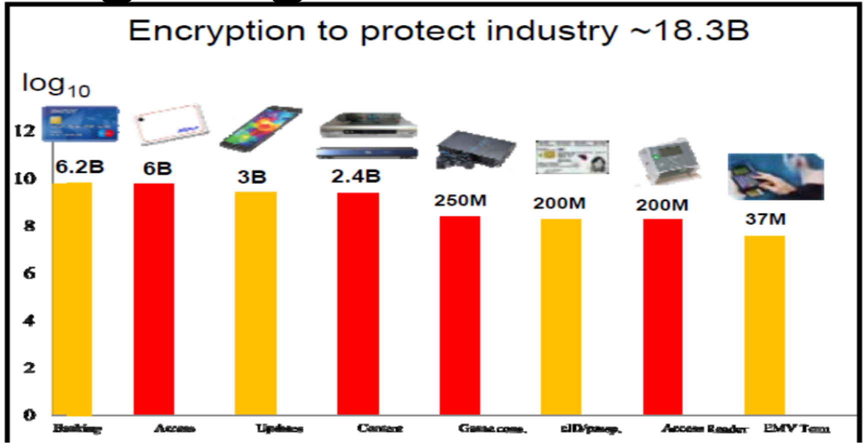
An active intruder

Sec

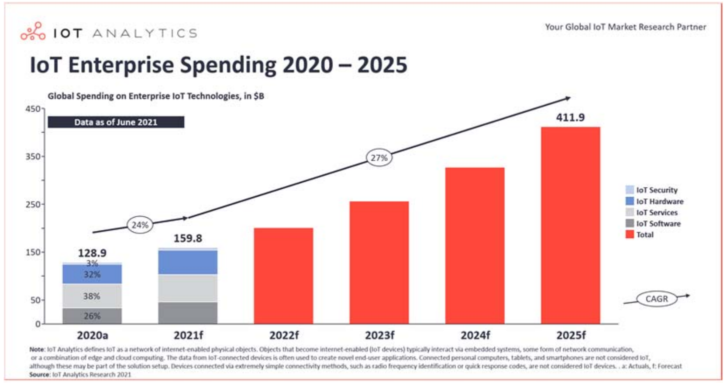
Crypto plays an increasingly important role

Crypto principles see growing usage in information protection

A locking approach



Encryption to protect industry ~18.3B



IoT Enterprise Spending 2020 – 2025

Note: IoT Analytics defines IoT as a network of internet-enabled physical objects. Objects that become internet-enabled (IoT devices) typically interact via embedded systems, some form of network communication, or a combination of edge and cloud computing. The data from IoT-connected devices is often used to create novel end-user applications. Connected personal computers, tablets, and smartphones are not considered IoT, although these may be part of the solution setup. Devices connected via extremely simple connectivity methods, such as radio frequency identification or quick response codes, are not considered IoT devices. . a: Actuals, f: Forecast
Source: IoT Analytics Research 2021

Cryptographic algorithms protects critical infrastructure and assets!

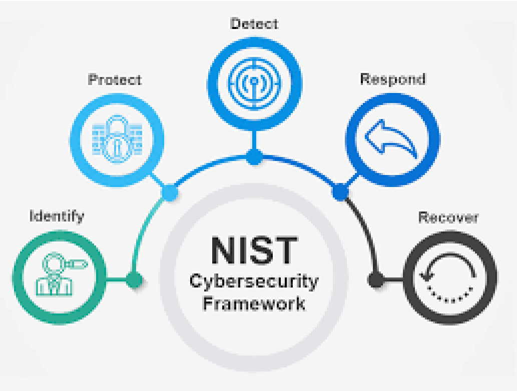
Module 4: Cryptology for IoT

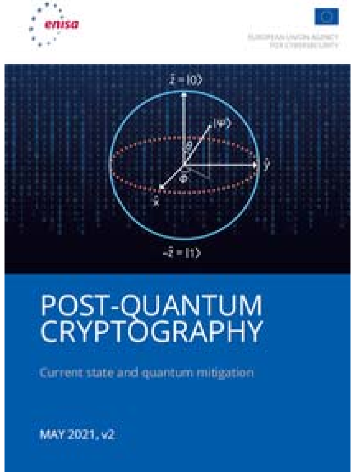
Crypto plays an increasingly important role

- **New cryptographic regulations**

NIST
National Institute of Standards and Technology
U.S. Department of Commerce

ISO
International Organization for Standardization




enisa

EUROPEAN UNION AGENCY FOR CYBERSECURITY

POST-QUANTUM CRYPTOGRAPHY

Current state and quantum mitigation

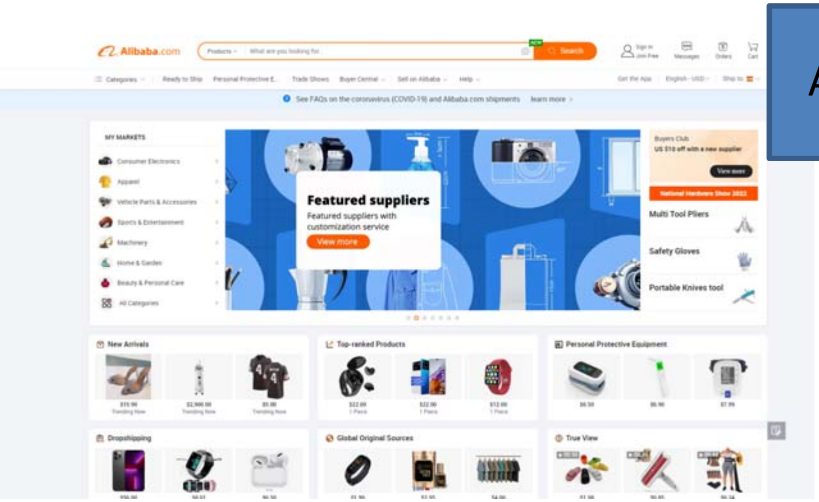
MAY 2021, v2

Sec

Crypto plays an increasingly important role

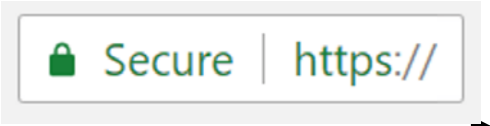
- **The core elements that make the cryptographic layers safe include:**
  - Algorithms,
  - Keys
  - Libraries
  - Certificates

Secure communication



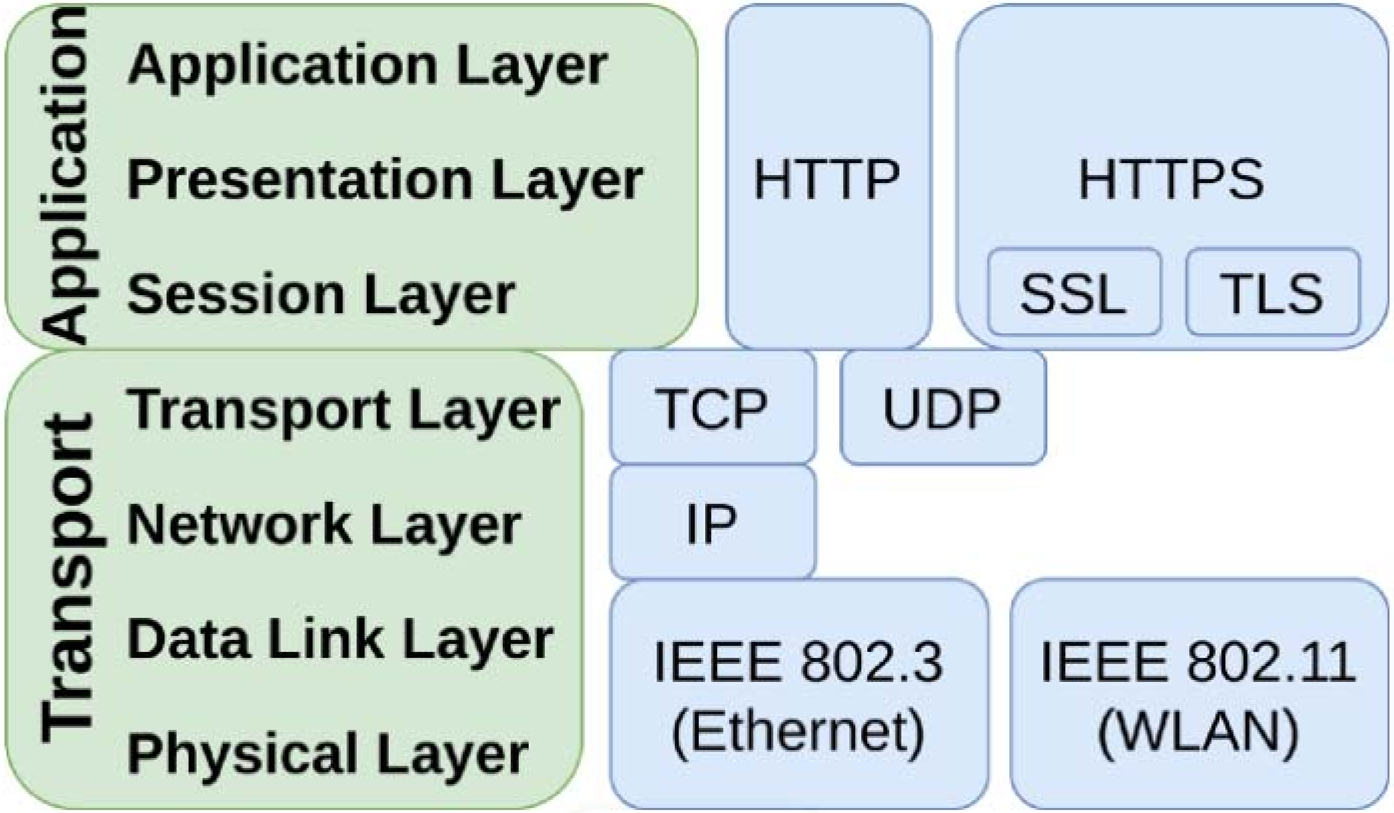
A

B

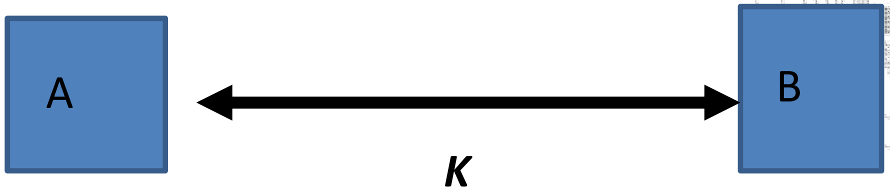


no eavesdropping
no tampering

Secure Sockets Layer / TLS



Secure Sockets Layer / TLS



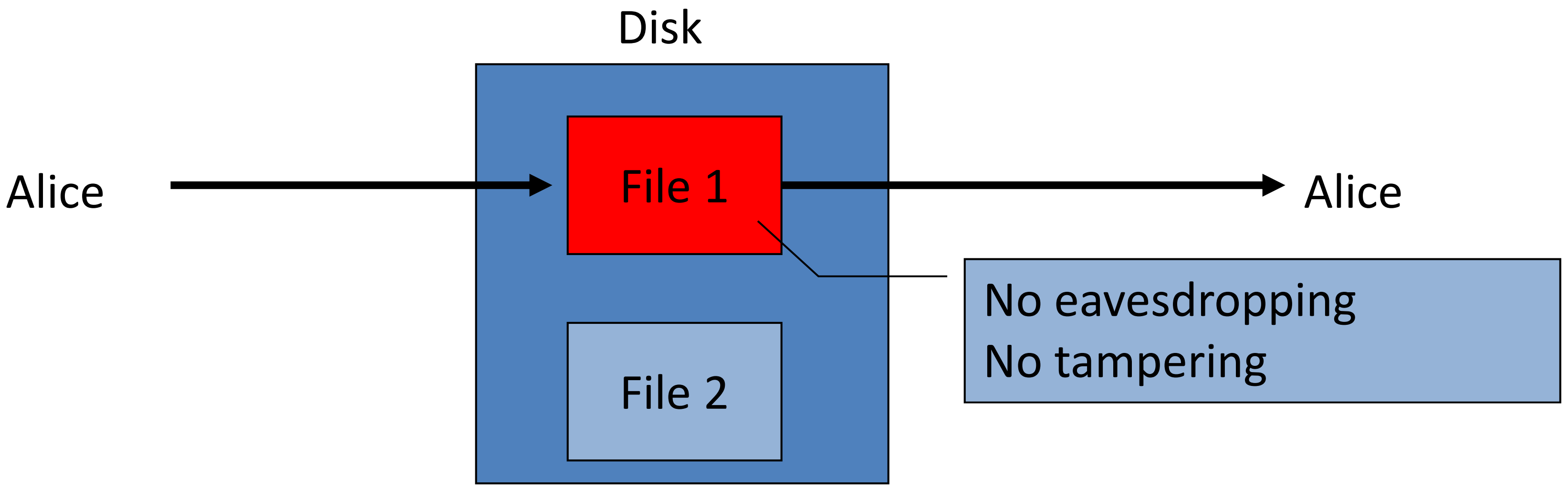
Two main parts

1. Handshake Protocol: **Establish shared secret key using public-key cryptography** (Last part of Crypto Module)

2. Record Layer: **Transmit data using shared secret (private) key**

   Ensure confidentiality and integrity (First part of Crypto Module)

Protected files on disk



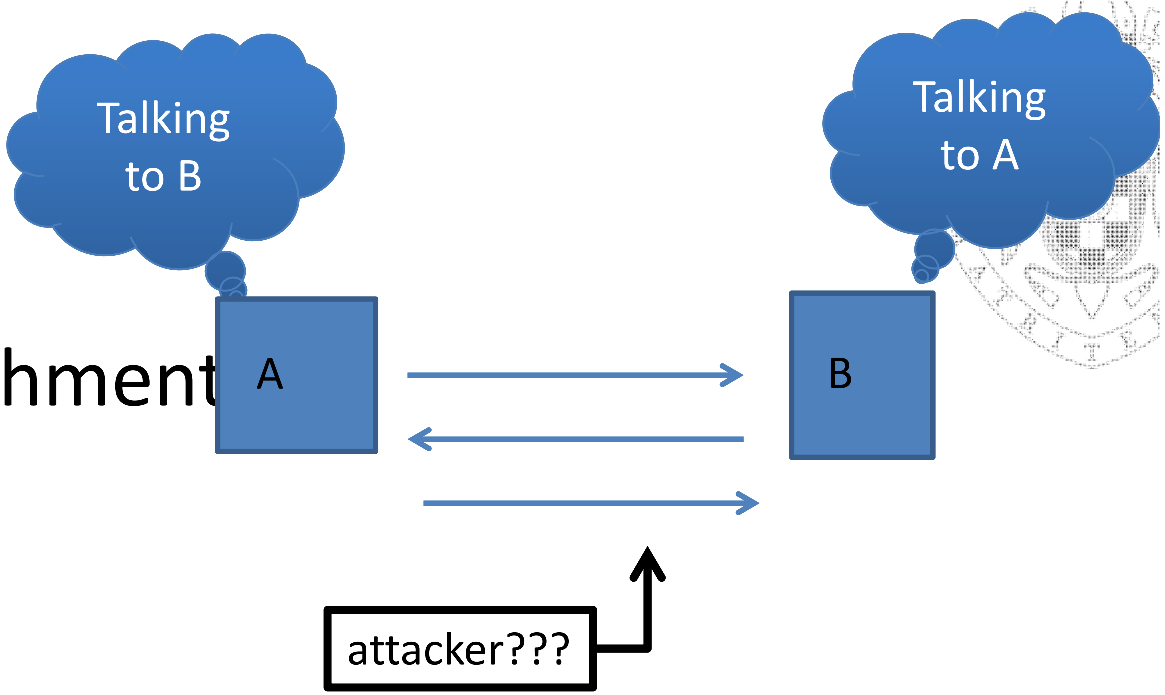
Analogous to secure communication:

Alice today sends a message to Alice tomorrow

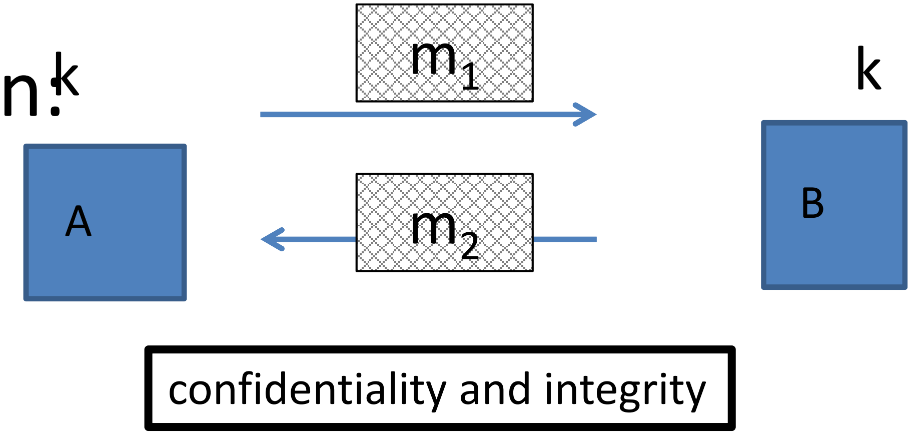
Module 4: Cryptology for IoT

Crypto core

Secret key establishment



Secure communication:



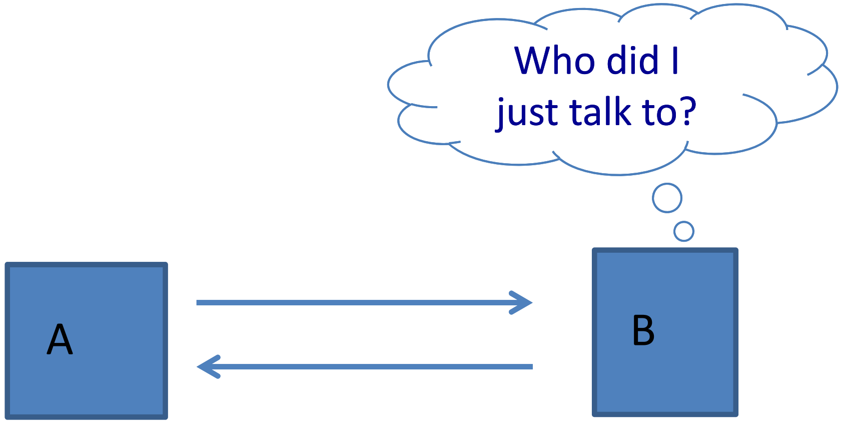
Sec

What else can crypto do?

- Digital signatures

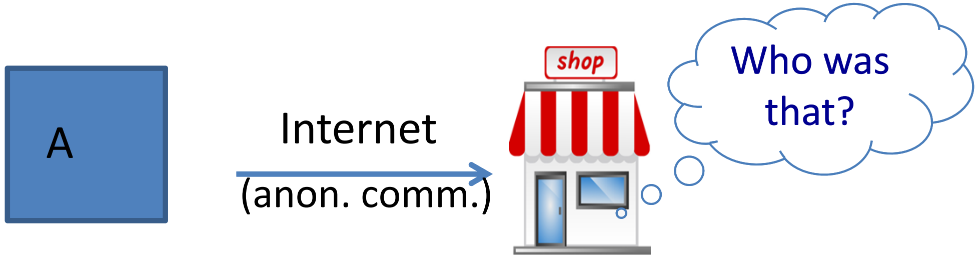
A signature

- Anonymous communication



But crypto can do much more

- Digital signatures
- Anonymous communication
- Anonymous **digital** cash
  - Can I spend a “digital coin” without anyone knowing who I am?
  - How to prevent double spending?



Module 4: Cryptology for IoT

Protocols

- Elections
- Private auctions

Sec

Protocols

- Elections
- Private auctions

trusted authority

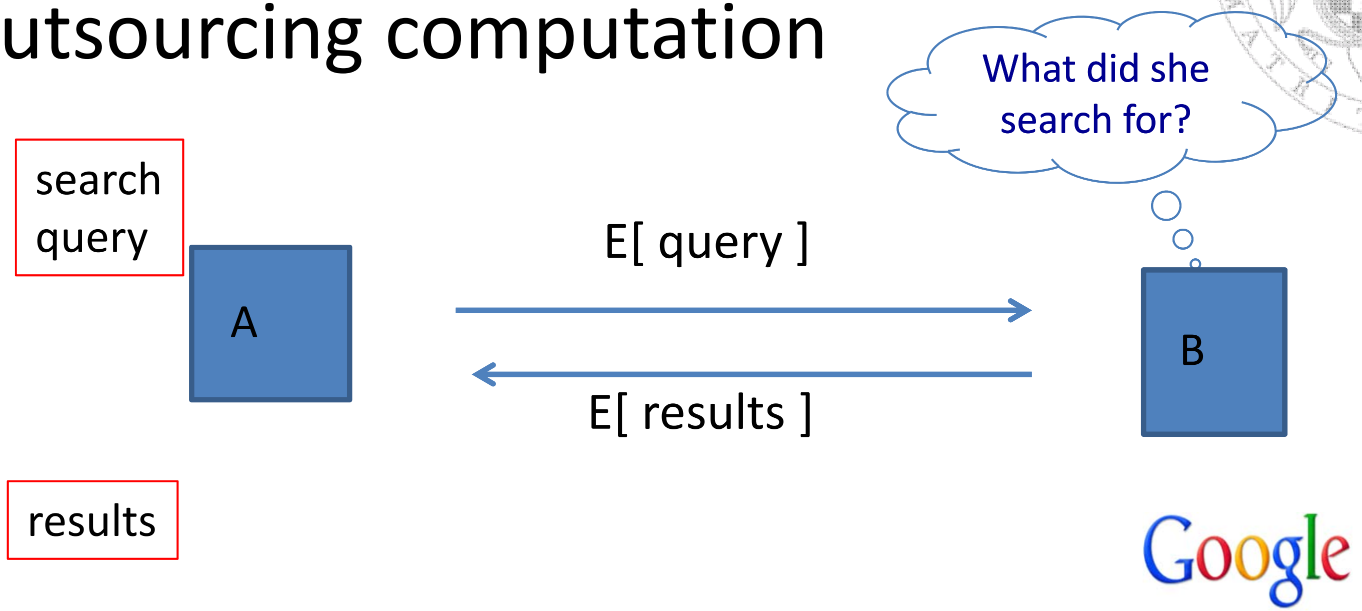
Goal: compute $f(x_1, x_2, x_3, x_4)$

“Thm:” anything the can done with trusted auth. can also be done without

- Secure multi-party computation
  - – E-voting without fraud.

Crypto magic

- Privately outsourcing computation



search query

A

E[query]

E[results]

What did she search for?

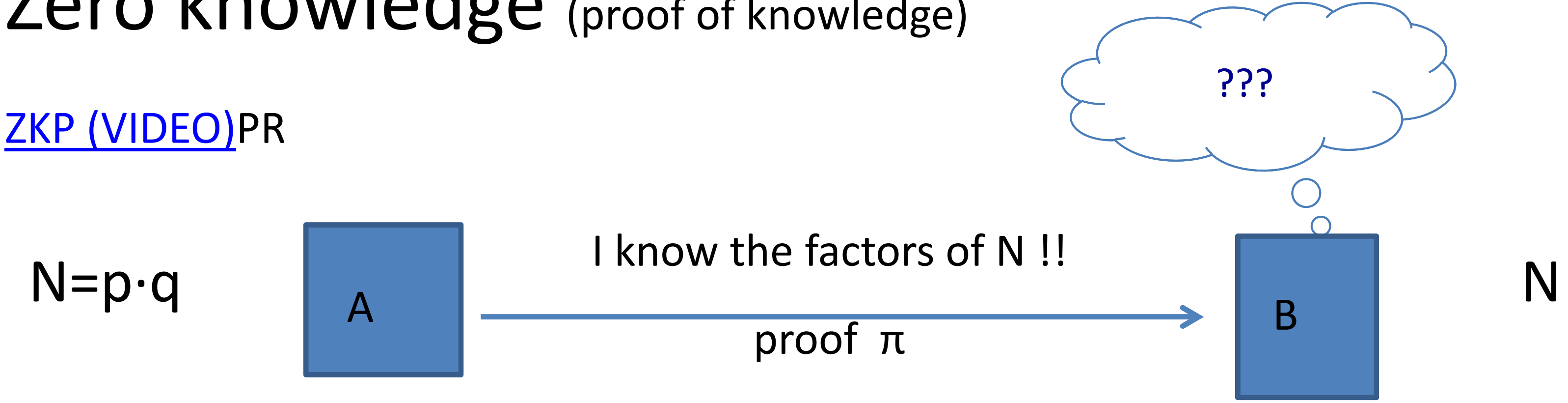
B

results

Google

- Zero knowledge (proof of knowledge)

- ZKP (VIDEO)PR



???

N=p·q

A

I know the factors of N !!

proof π

B

N

A rigorous science

The three steps in cryptography:

- Precisely specify threat model
- Propose a construction
- Prove that breaking construction under threat mode will solve an underlying hard problem

Terminology and Background
Threats to Messages and Crypto solutions

- Interception → Confidentiality
- Interruption
  - Blocking msgs
- Modification → Integrity
- Fabrication →Authentication

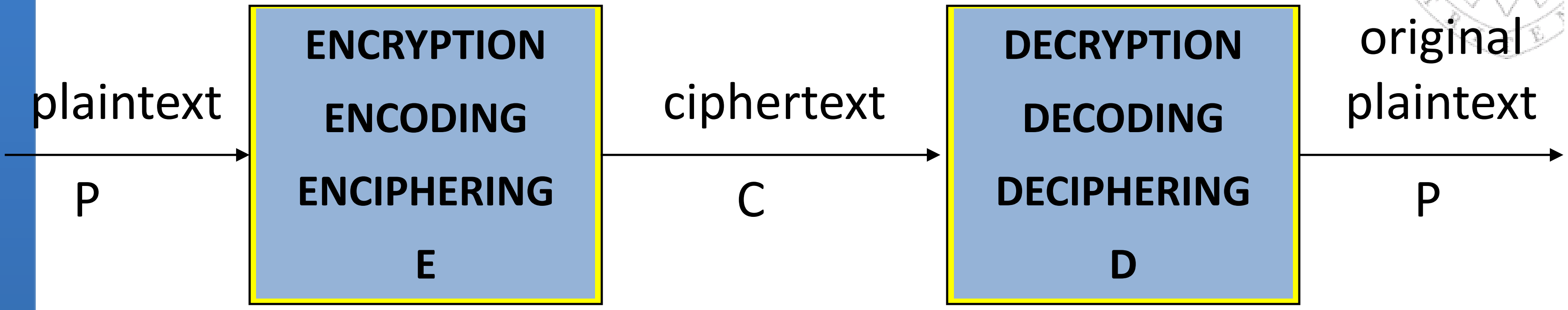
“A threat is blocked by control of a vulnerability”

[Pfleeger & Pfleeger]

Basic Terminology & Notation

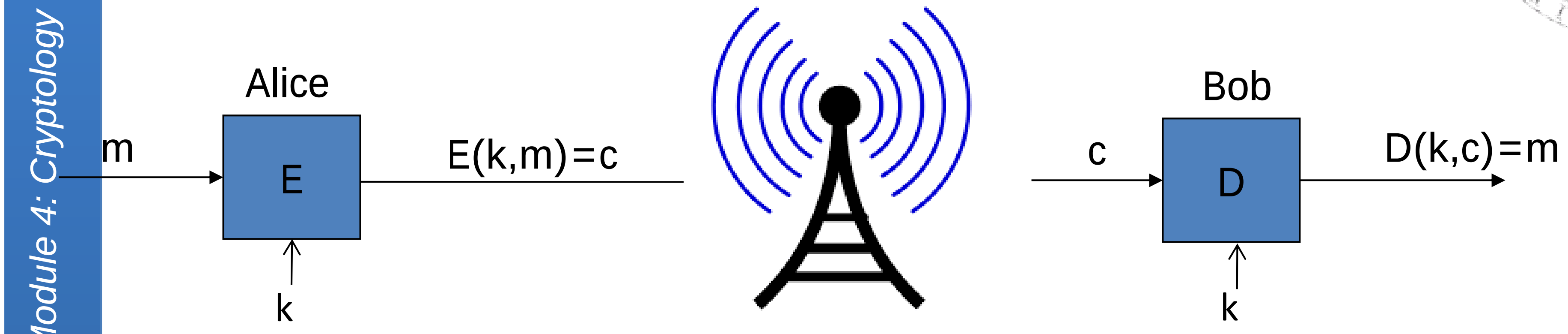
- **Cryptology:**
  - – cryptography + cryptanalysis
- **Cryptography:**
  - – art/science of keeping message secure
- **Cryptanalysis:**
  - – art/science of breaking ciphertext
    - *Enigma* in world war II
      - – Read the real story – not fabrications!

Basic Cryptographic Scheme



- $P = \langle p_1, p_2, \ldots, p_n \rangle$ p_i = i-th char of P
  - P = “DO NOT TELL ANYBODY” p_1 =”D”, p_2 = ”O”, etc.
  - By convention, cleartext in uppercase
- $C = \langle c_1, c_2, \ldots, c_n \rangle$ c_i = i-th char of C
  - C = “ep opu ufmm bozcpez” c_1 =”e”, c_2 =”p”, etc.
  - By convention, ciphertext in lowercase

Building block: sym. encryption



Alice

m

E

k

E(k,m)=c

Bob

c

D

k

D(k,c)=m

E, D: cipher k: secret key (e.g. 128 bits)

m, c: plaintext, ciphertext

Encryption algorithm is publicly known

- Never use a proprietary cipher

Use Cases

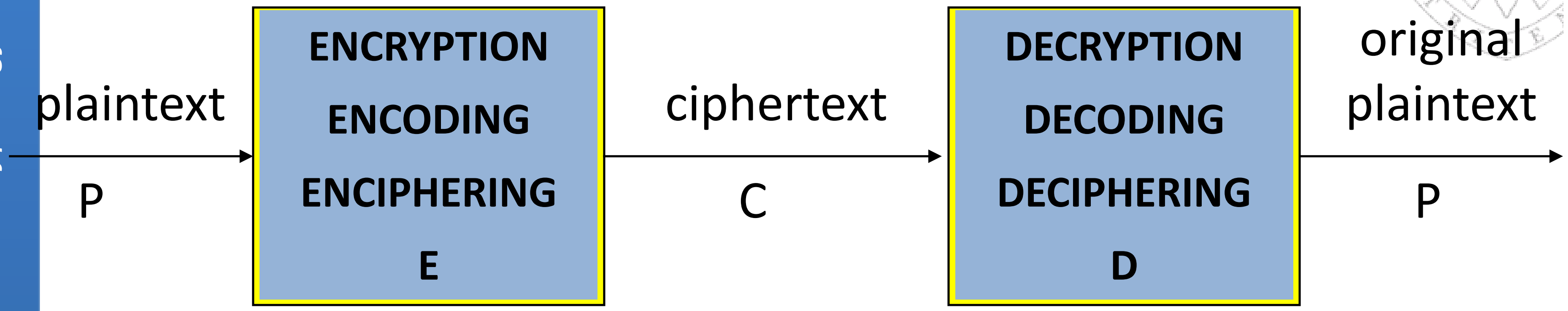
Single use key: (one time key)

- Key is only used to encrypt one message
  - encrypted email: new key generated for every email

Multi use key: (many time key)

- Key used to encrypt multiple messages
  - encrypted files: same key used to encrypt many files
- Need more machinery than for one-time key

Formal Notation



plaintext

P

ENCRYPTION

ENCODING

ENCIPHERING

E

ciphertext

C

DECRYPTION

DECODING

DECIPHERING

D

original plaintext

P

- C = E(P)
- P = D(C)

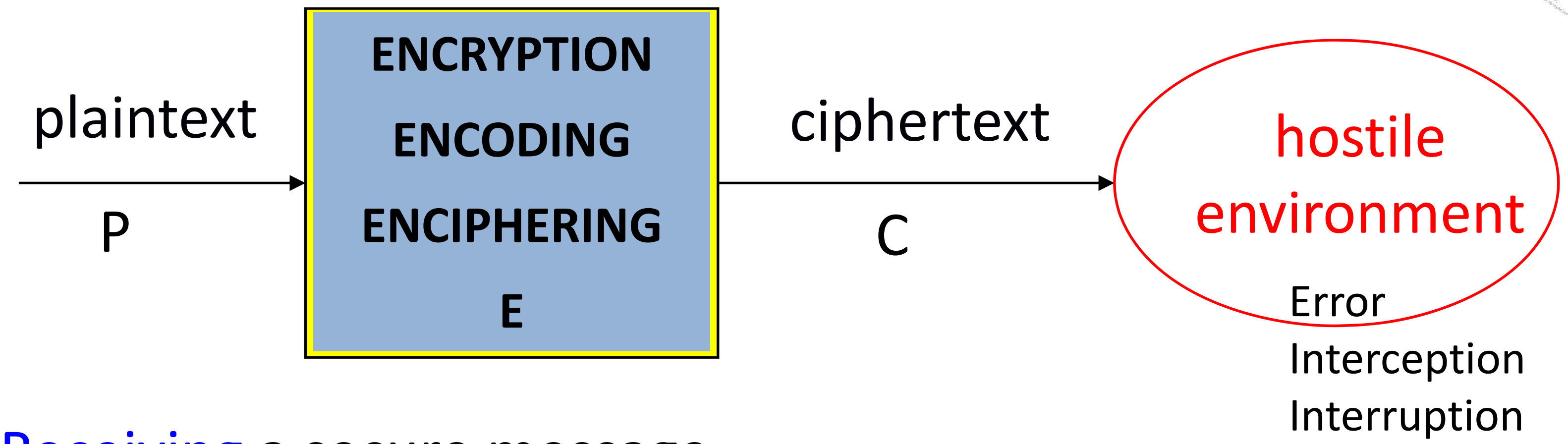
E – encryption rule/algorithm

D – decryption rule/algorithm

- We need a cryptosystem, where:
  - P = D(C)= D(E(P))
    - i.e., able to get the original message back

Cryptography in Practice

- Sending a secure message



plaintext

P

ENCRYPTION

ENCODING

ENCIPHERING

E

ciphertext

C

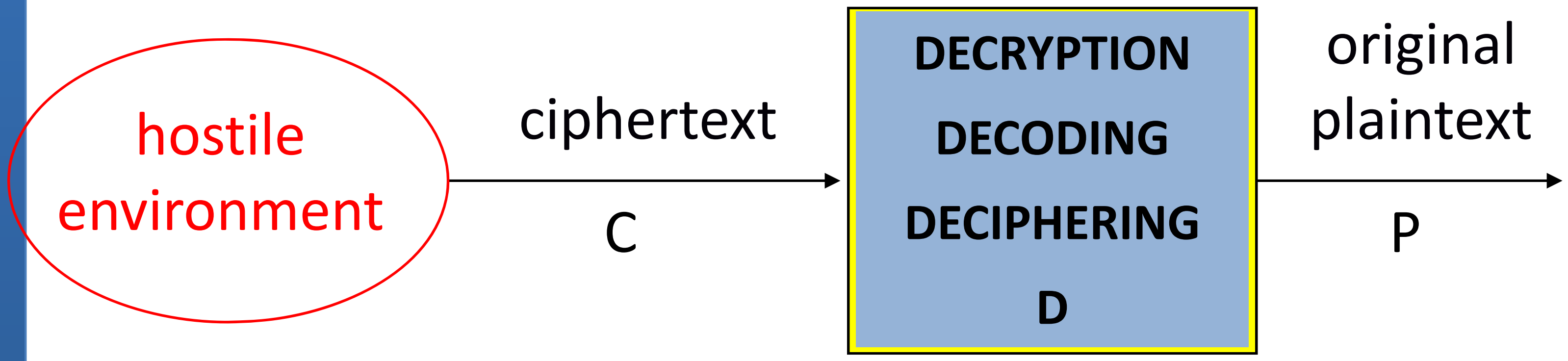
hostile environment

Error

Interception

Interruption

- Receiving a secure message



hostile environment

ciphertext

C

DECRYPTION

DECODING

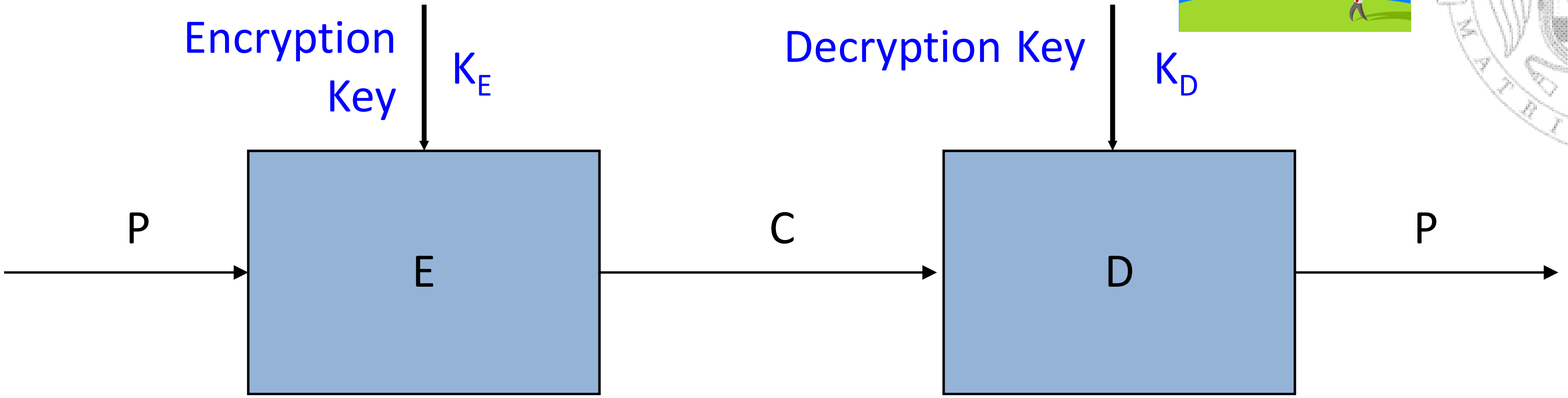
DECIPHERING

D

original plaintext

P

Crypto System with Keys



- $C = E(K_E, P)$
  - E = *set* of encryption algorithms / K_E selects $E_i \in E$
- $P = D(K_D, C)$
  - D = *set* of decryption algorithms / K_D selects $D_j \in D$
- Crypto algorithms and keys are like door locks and keys
- We need: $P = D(K_D, E(K_E, P))$

Classification of Cryptosystems w.r.t. Keys

- Keyless cryptosystems exist (e.g., Caesar's cipher)
  - Less secure
- Symmetric cryptosystems: $K_E = K_D$
  - Classic
  - Encipher and decipher using the same key
    - Or one key is easily derived from other
- Asymmetric cryptosystems: $K_E \neq K_D$
  - Public key system
  - Encipher and decipher using different keys
    - Computationally infeasible to derive one from other

Cryptanalysis (1)

- **Cryptanalysts goals:**
  - Break a single message
  - Recognize patterns in encrypted messages, to be able to break the subsequent ones
  - Infer meaning without breaking encryption
    - Unusual volume of messages between enemy troops may indicate a coming attack
    - Busiest node may be enemy headquarters
  - Deduce the key, to facilitate breaking subsequent messages
  - Find vulnerabilities in implementation or environment of an encryption algorithm
  - Find a general weakness in an encryption algorithm

Cryptanalysis (2)

- **Information for cryptanalysts:**
  - Intercepted encrypted messages
  - Known encryption algorithms
  - Intercepted plaintext
  - Data known or suspected to be ciphertext
  - Math or statistical tools and techniques
  - Properties of natural languages
    - Esp. adversary's natural language
      - To confuse the enemy, Americans used Navajo language in WW2
  - Propertiers of computer systems
- Role of ingenuity / luck
- There are *no* rules!!!

Breakable Encryption (1)

- **Breakable encryption**
  - *Theoretically*, it is possible to devise unbreakable cryptosystems
  - *Practical* cryptosystems almost always are breakable, given adequate time and computing power
  - The trick is to make breaking a cryptosystem hard enough for the intruder

Breakable Encryption (2)

- Example: Breakability of an encryption algorithm

  Message with just 25 characters

  - 26^{25} possible decryptions ~ 10^{35} decryptions
  - Only one is the right one
  - Brute force approach to find the right one:
    - At 10^{10} (10 bln) decryption/sec => 10^{35} / 10^{10} = 10^{16} sec = 10 bln yrs !
    - Infeasible with current technology

- Be smarter – use ingenuity

  – Could reduce 26^{25} to, say, 10^{15} decryptions to check

  At 10^{10} decr./sec => 10^{15} / 10^{10} = 10^{5} sec = ~ 1 day

Requirements for Crypto Protocols

- Messages should get to destination
- Only the recipient should get it
- Only the recipient should see it
- Proof of the sender’s identity
- Message shouldn’t be corrupted in transit
- Message should be sent/received once
- Proofs that message was sent/received (non-repudiation)

Benefits and things to remember

Cryptography is:

- A tremendous tool which minimizes problems
- The basis for many security mechanisms
- Adds an envelope (encoding) to an open postcard (plaintext or cleartext)

Cryptography is not:

- The solution to all security problems
- Reliable unless implemented and used properly
- Something you should try to invent yourself
  - many many examples of broken ad-hoc designs

Bibliography consulted

- Book “A Graduate Course in Applied Cryptography” Dan Boneh and Victor Shoup. 2020. U. Standford
- https://crypto.stanford.edu/~dabo/cryptobook/BonehShoup_0_5.pdf
- https://crypto.stanford.edu/~dabo/courses/OnlineCrypto/
- Introduction to Cryptography. Prof. Leszek T. Lilien from Wmich: http://www.cs.wmich.edu/~llilien/

Cryptology for IoT

Modules M4, M7, M9
Session of 26th April, 2022.

M4. Introduction to the modules
M4.1 Introduction to the Cryptology
M4.2 Introduction to Cryptool CT2

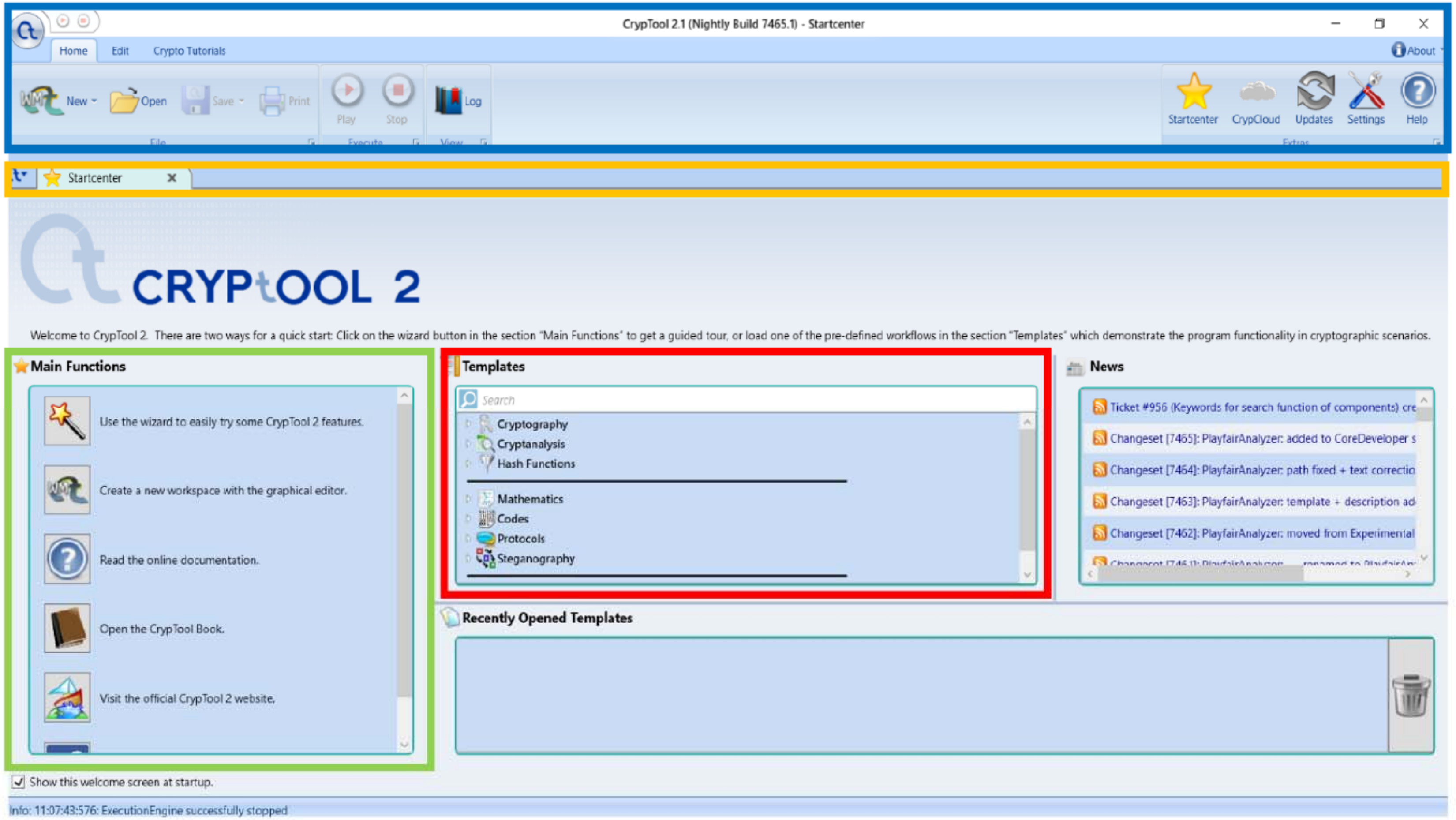
Prof.: Guillermo Botella

Overview

- Startcenter
- Wizard
- Workspace Manager
- Online Help
- Templates
- CrypCloud

Startcenter

- Startcenter is what you see every time you start the CT2 application.



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

Main Functions

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

Templates

News

- Ticket #956 (Keywords for search function of components) cre
- Changeset [7465]: PlayfairAnalyzer: added to CoreDeveloper s
- Changeset [7464]: PlayfairAnalyzer: path fixed + text correctio
- Changeset [7463]: PlayfairAnalyzer: template + description ad
- Changeset [7462]: PlayfairAnalyzer: moved from Experimental

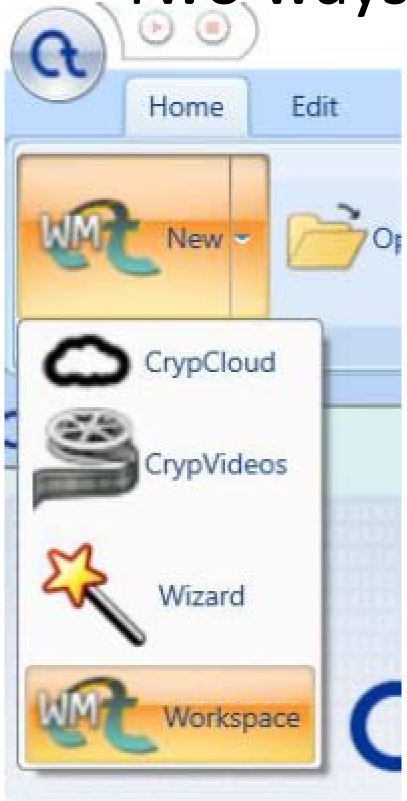
Recently Opened Templates

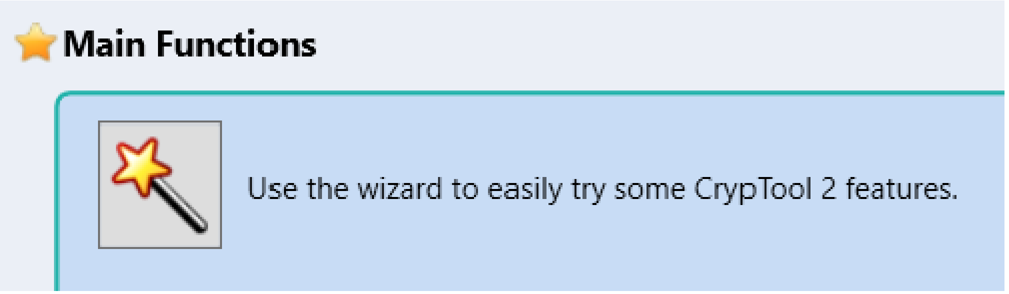
☑ Show this welcome screen at startup.

Info: 11:07:43:576: ExecutionEngine successfully stopped

Wizard

- The Wizard guides beginners through different topics of cryptology.
  - Two ways to access



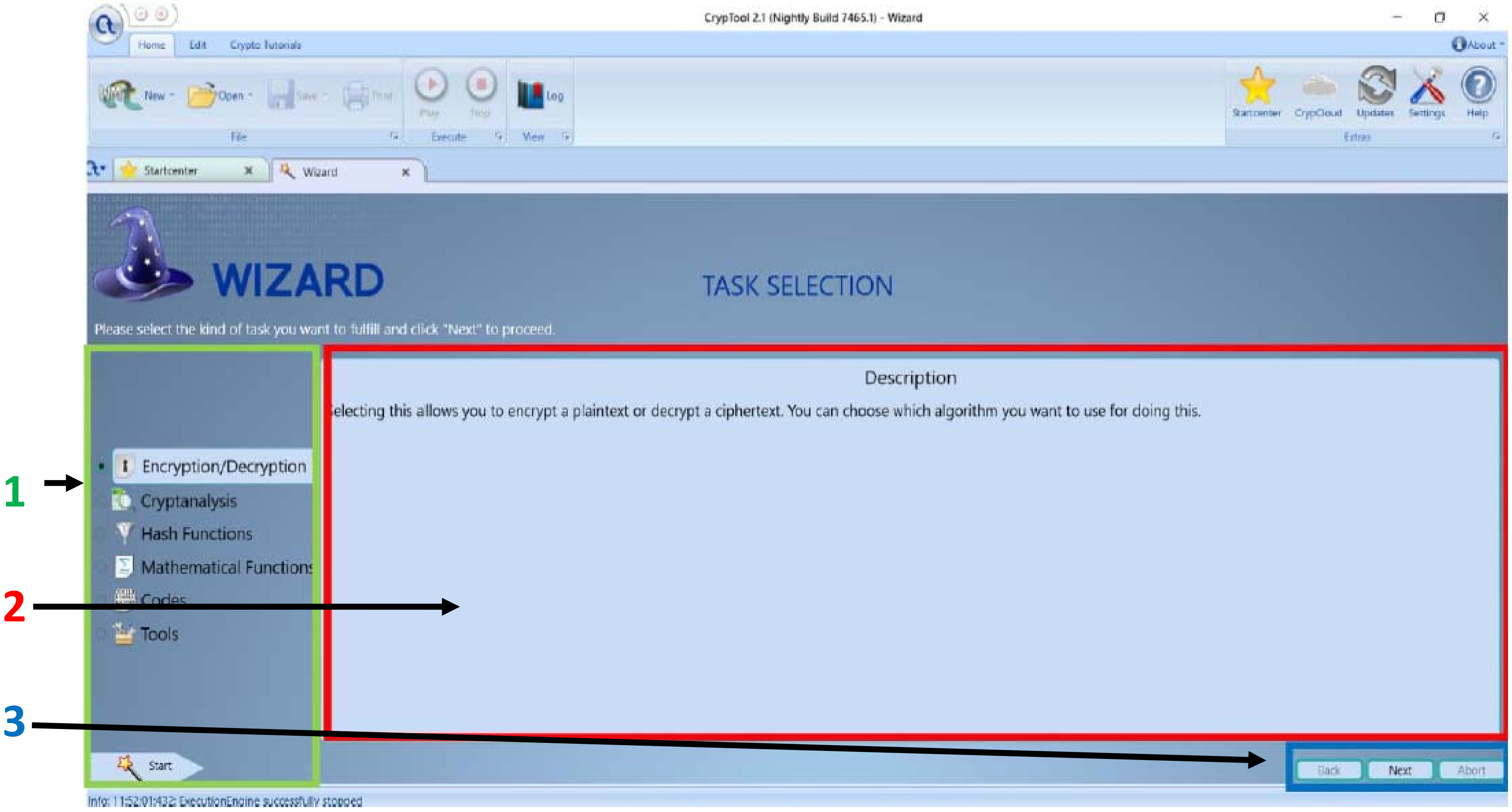


Main Functions

Use the wizard to easily try some CrypTool 2 features.

Wizard

- It consists on three main areas:



Please select the kind of task you want to fulfill and click "Next" to proceed.

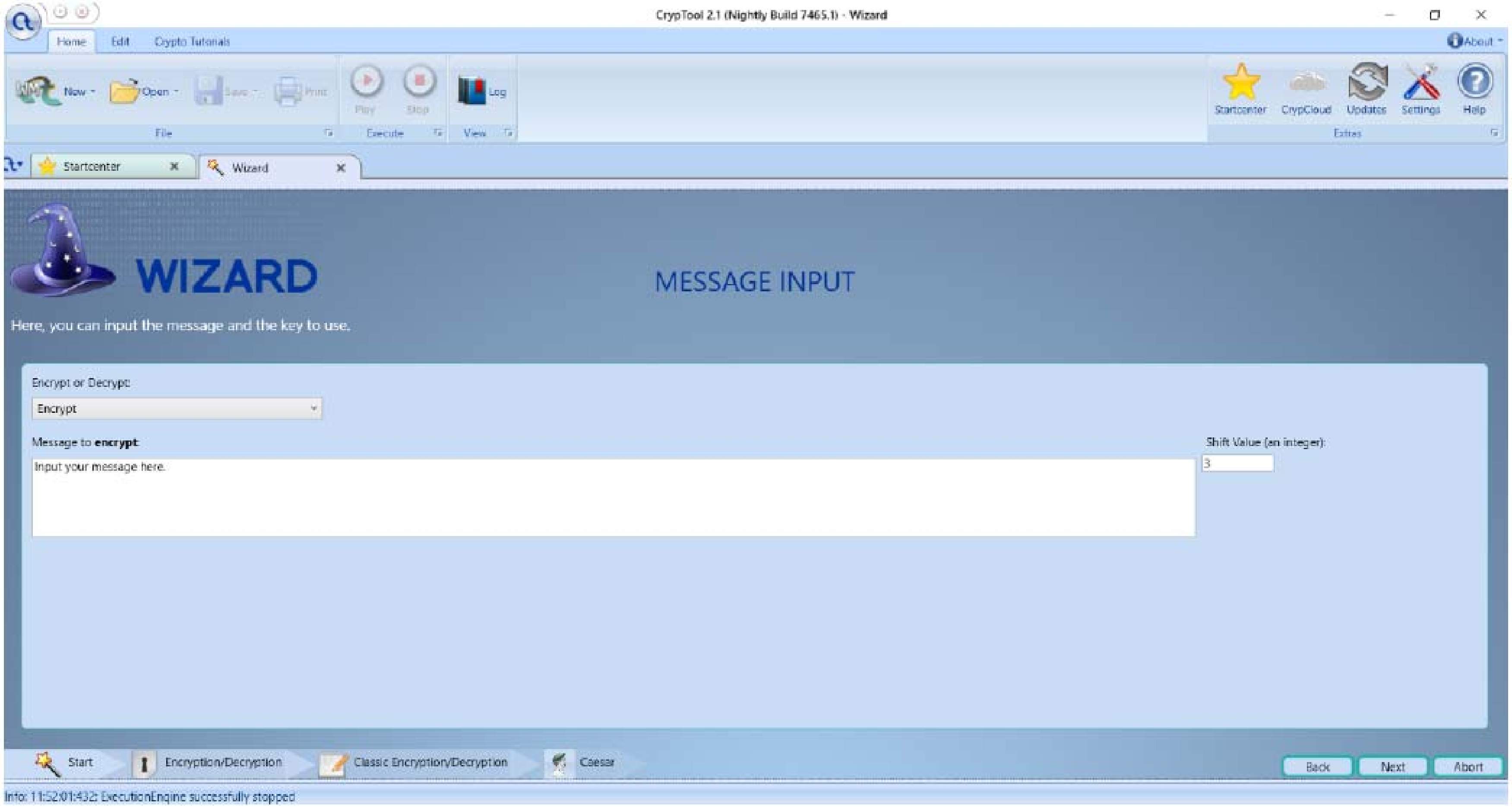
Description

Selecting this allows you to encrypt a plaintext or decrypt a ciphertext. You can choose which algorithm you want to use for doing this.

Info: 11:52:01:432: ExecutionEngine successfully stopped

Wizard

- Use case:



Here, you can input the message and the key to use.

Encrypt or Decrypt: Encrypt

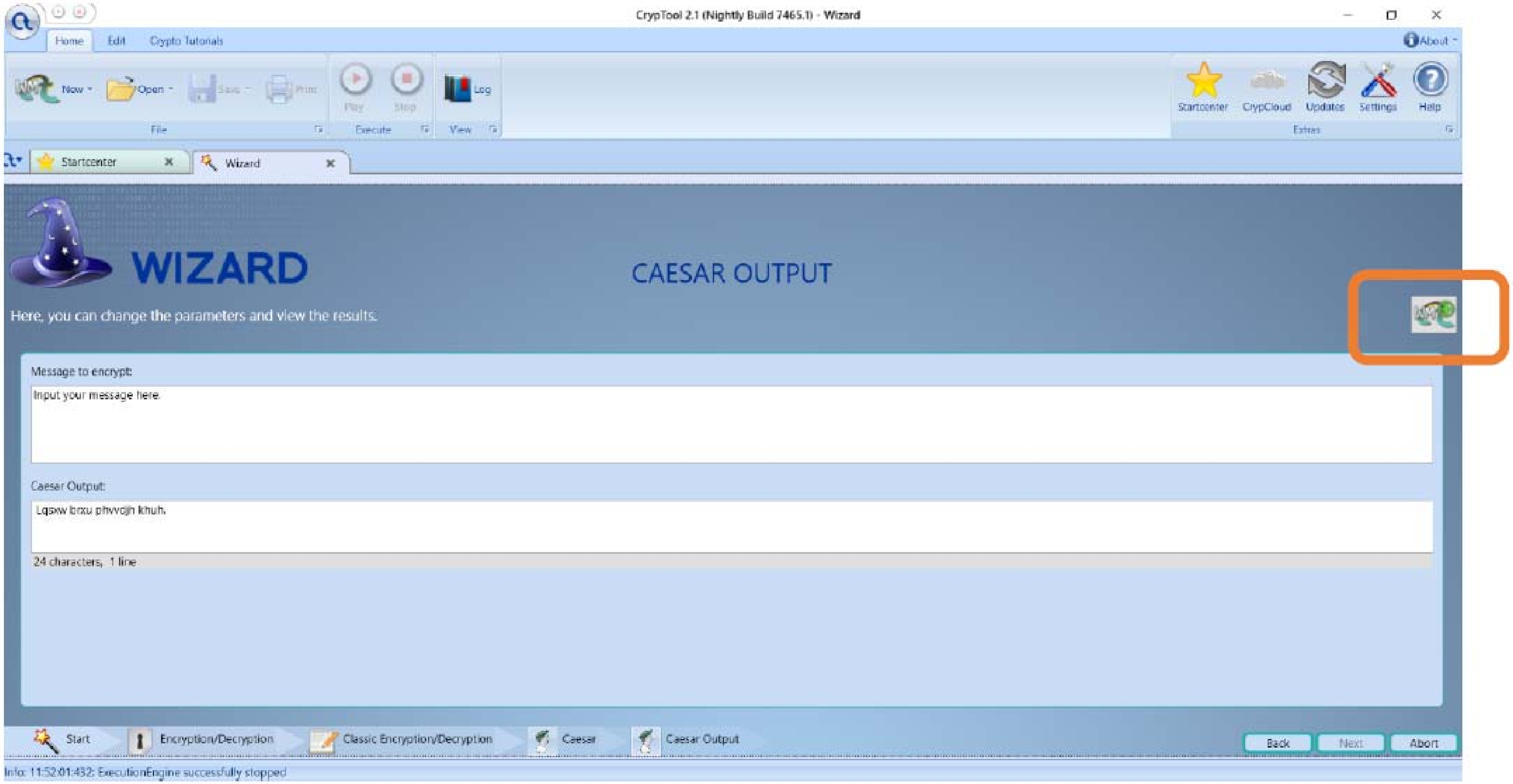
Message to **encrypt**: Input your message here.

Shift Value (an integer): 3

Info: 11:52:01:432: ExecutionEngine successfully stopped

Wizard

- Use case:



Here, you can change the parameters and view the results.

Message to encrypt:

Input your message here.

Caesar Output:

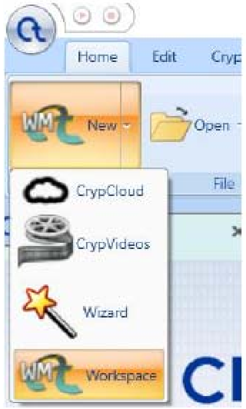
Lqsxw brxu phvvdjh khuh.

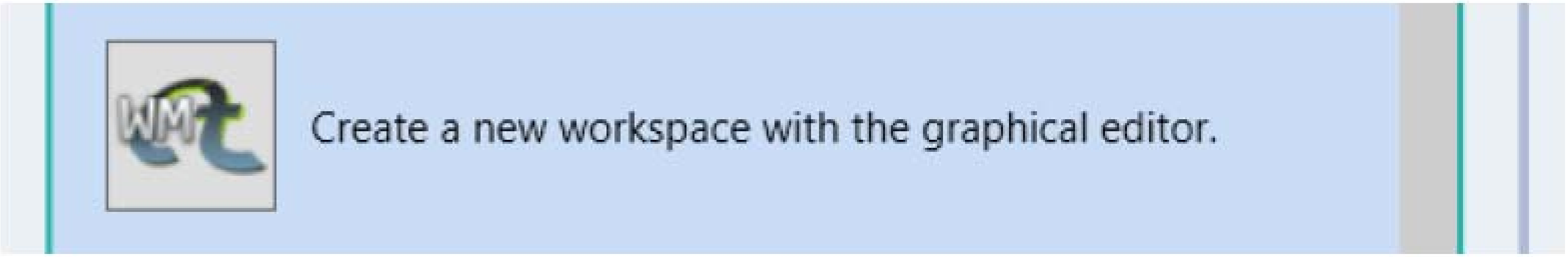
24 characters, 1 line

```
Info: 11:52:01:432: ExecutionEngine successfully stopped
```

Workspace manager

- It implements the graphical programming language of CT2.
- There are two ways to start the workspace manager

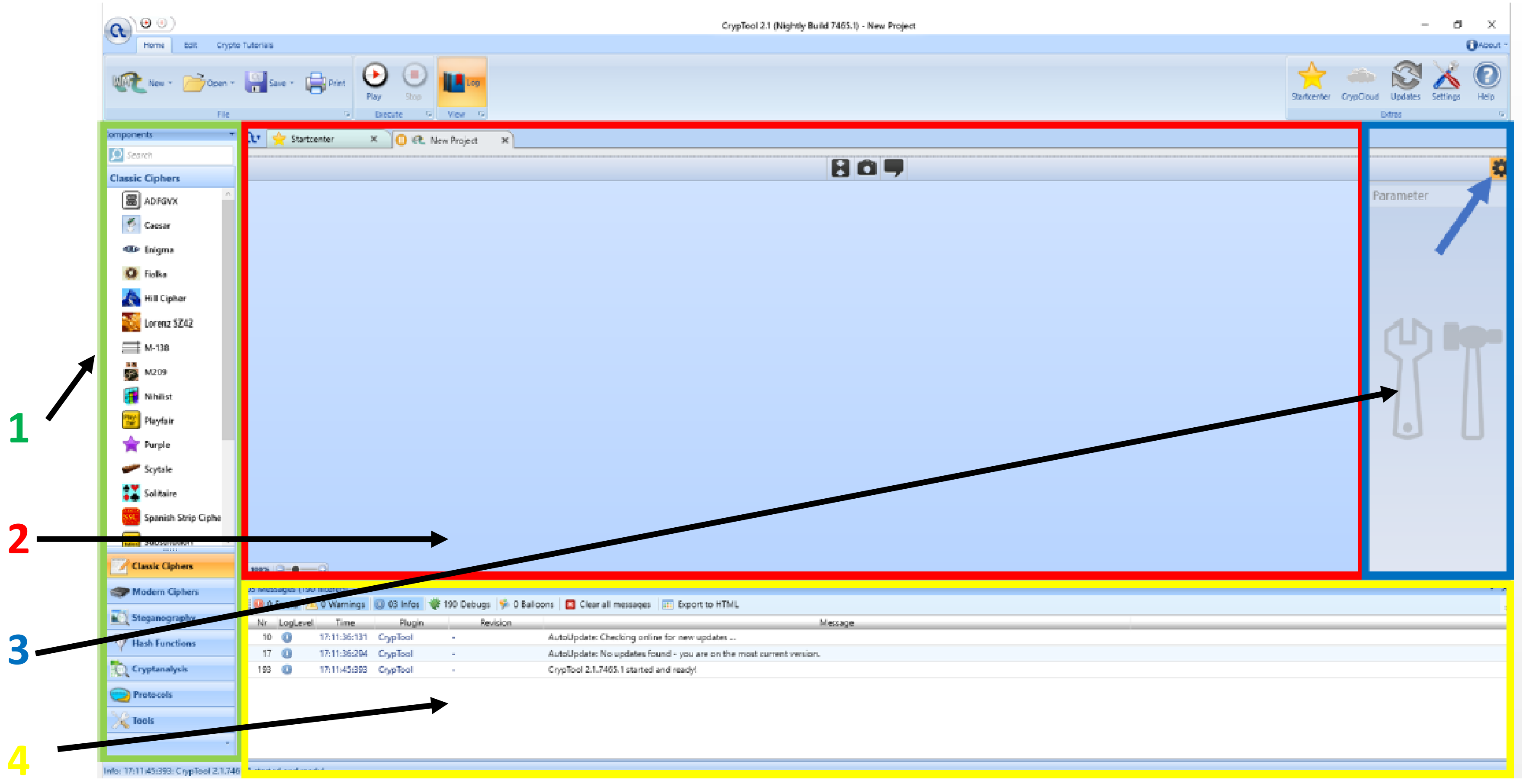




Create a new workspace with the graphical editor.

Workspace manager

- It consists on four main areas:

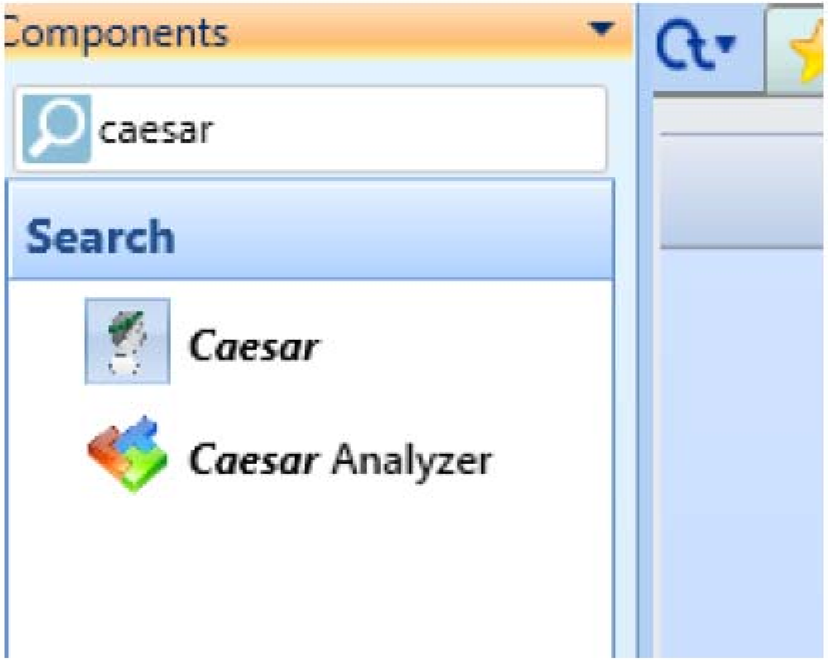


```
Nr   LogLevel  Time           Plugin    Revision  Message
10             17:11:36:131   CrypTool  -         AutoUpdate: Checking online for new updates ...
17             17:11:36:294   CrypTool  -         AutoUpdate: No updates found - you are on the most current version.
193            17:11:45:393   CrypTool  -         CrypTool 2.1.7465.1 started and ready!
```

Info: 17:11:45:393: CrypTool 2.1.7465.1 started and ready!

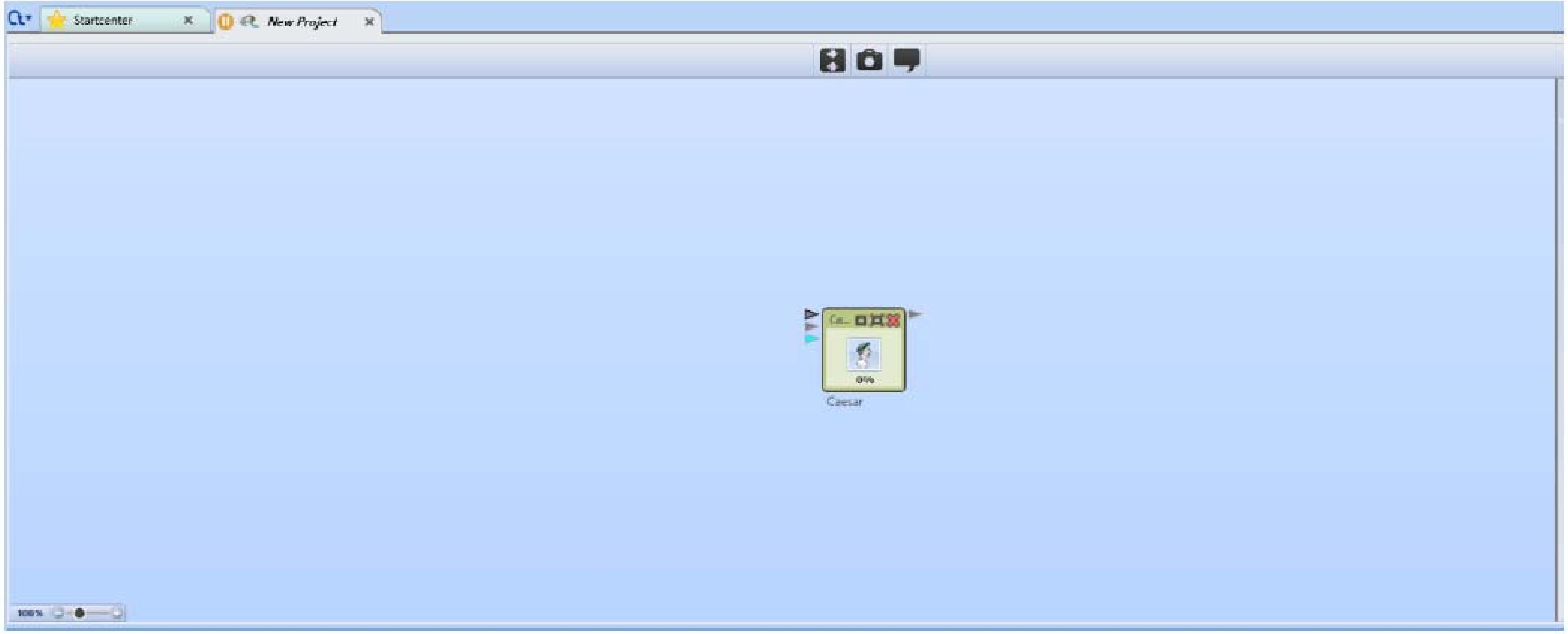
Example

- Building a Workflow for the Caesar Cipher
  - – Search the cipher in the component list



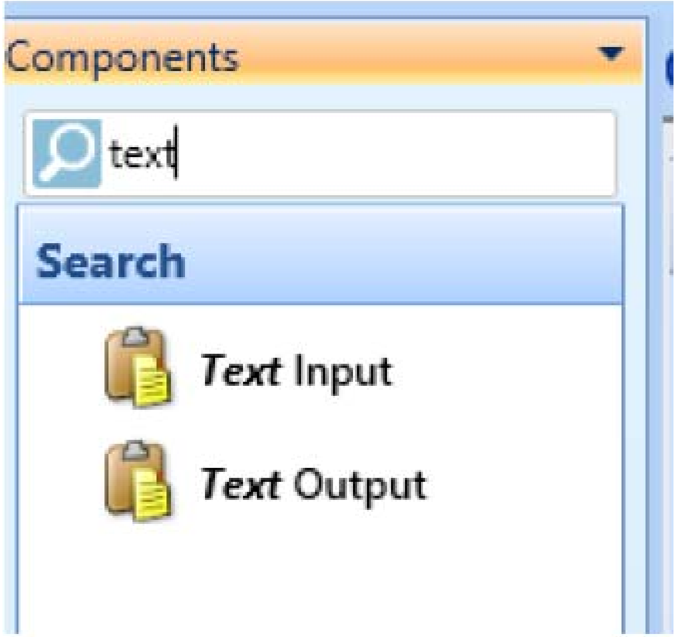
Example

- Building a Workflow for the Caesar Cipher
  - – Drag the component to the canvas



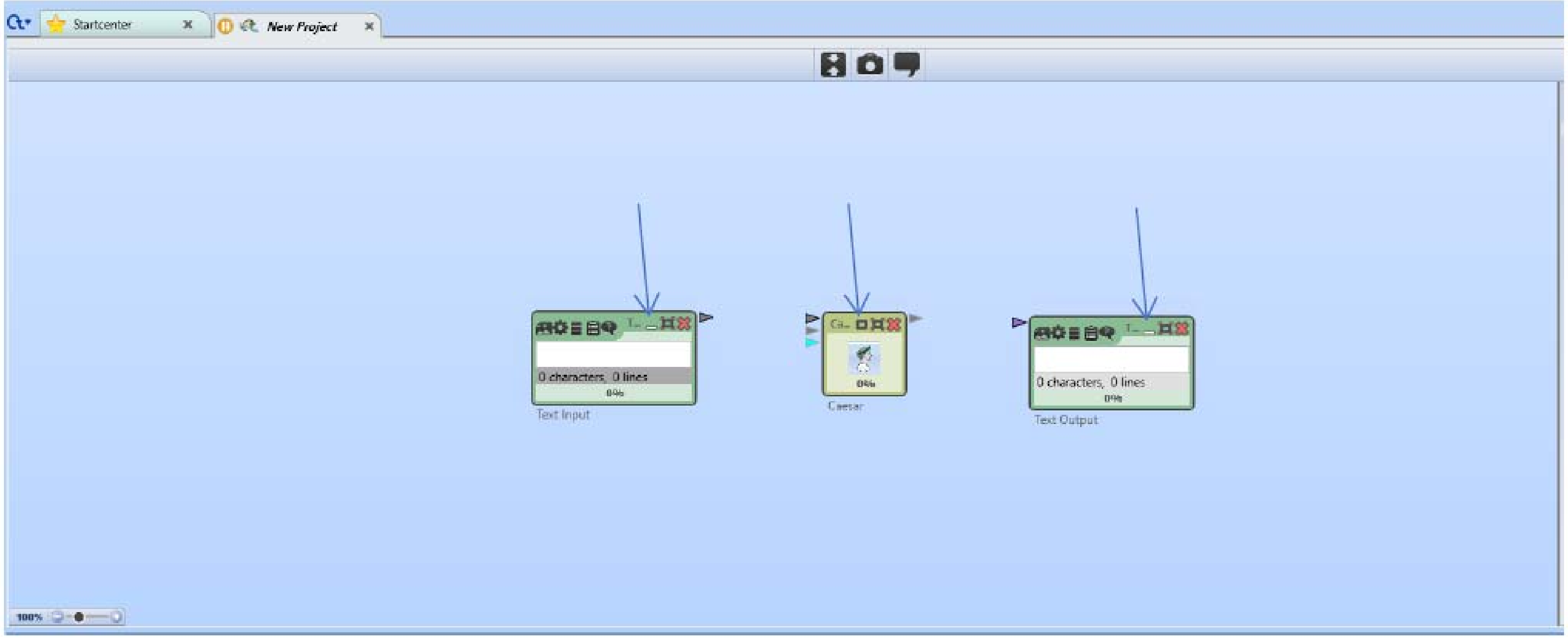
Example

- Building a Workflow for the Caesar Cipher
  - – Search for the text component in the list



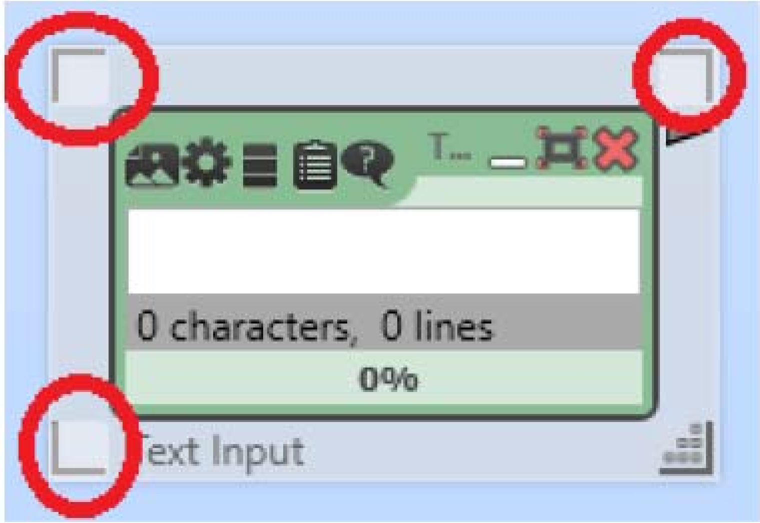
Example

- Building a Workflow for the Caesar Cipher
  - Drag the text component to the canvas
  - Do the same for output component



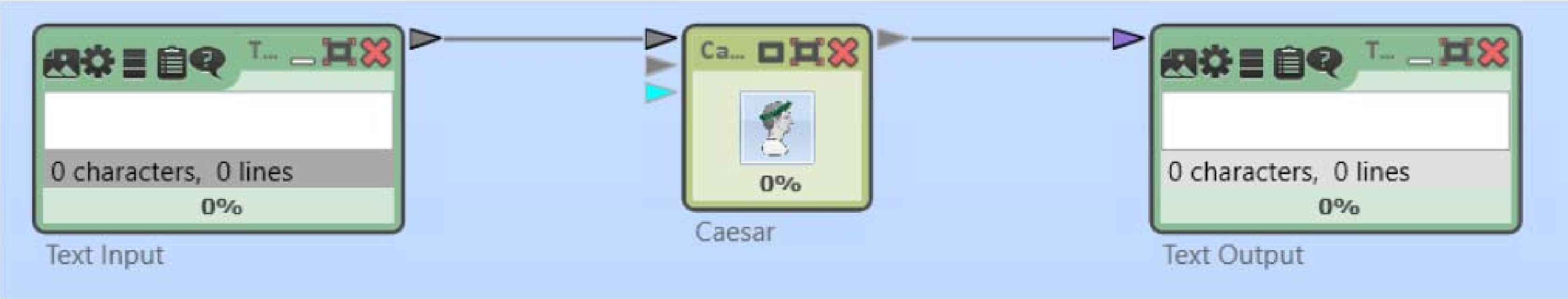
Example

- Building a Workflow for the Caesar Cipher
  - – Resize the components if needed



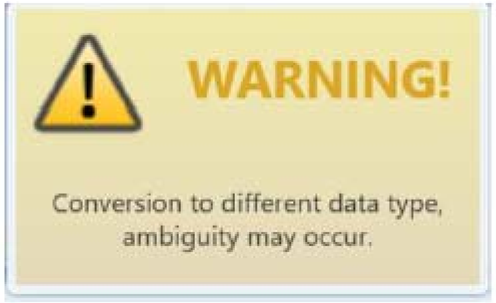
Example

- Building a Workflow for the Caesar Cipher
  - – Connect the components



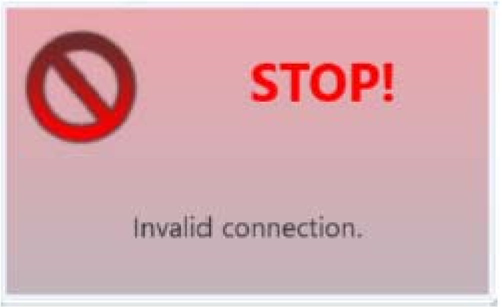
Example

- Building a Workflow for the Caesar Cipher
  - – Connection info



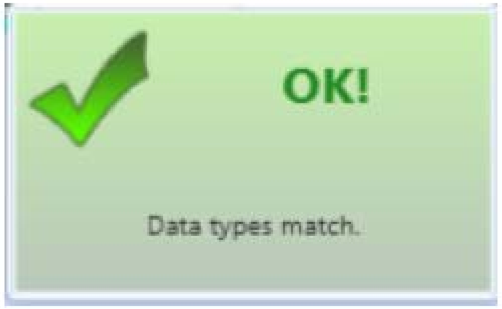
WARNING!

Conversion to different data type, ambiguity may occur.



STOP!

Invalid connection.

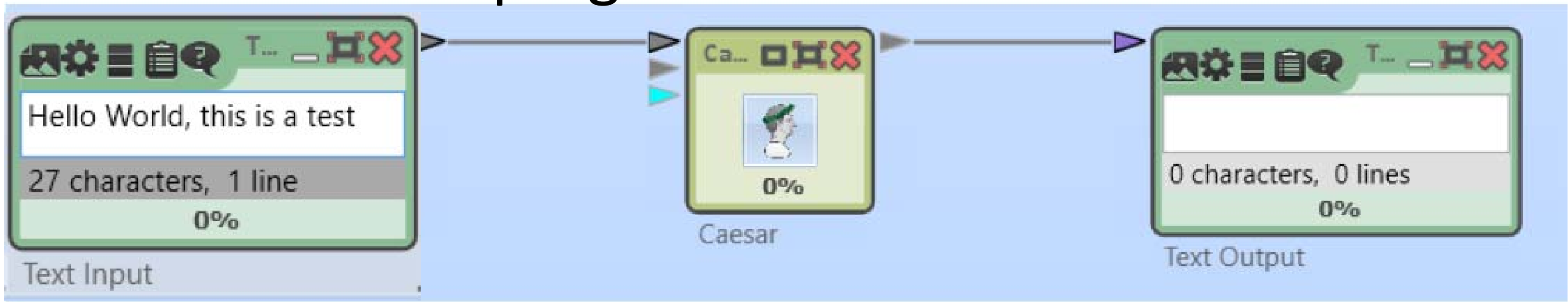


OK!

Data types match.

Example

- Building a Workflow for the Caesar Cipher
  - Execute the program



Hello World, this is a test

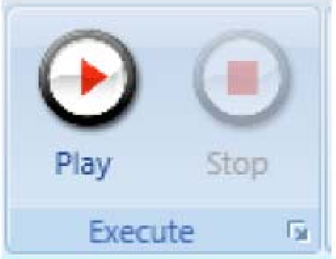
27 characters, 1 line

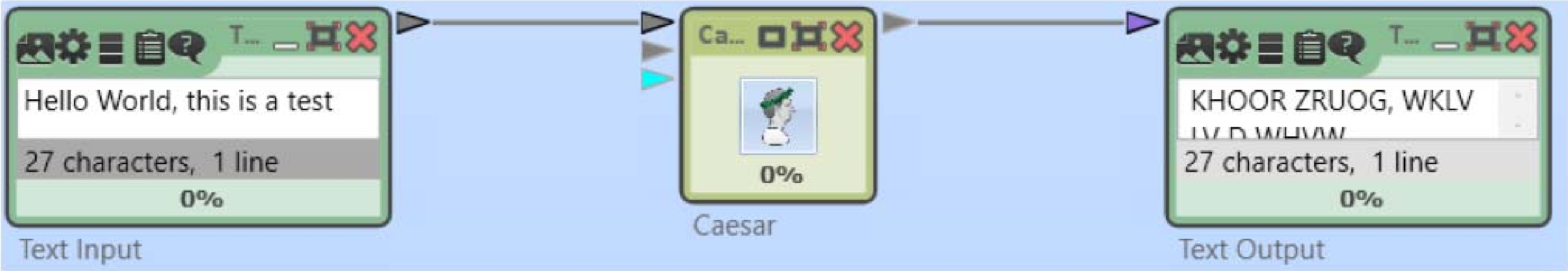
0%

0%

0 characters, 0 lines

0%





Hello World, this is a test

27 characters, 1 line

0%

0%

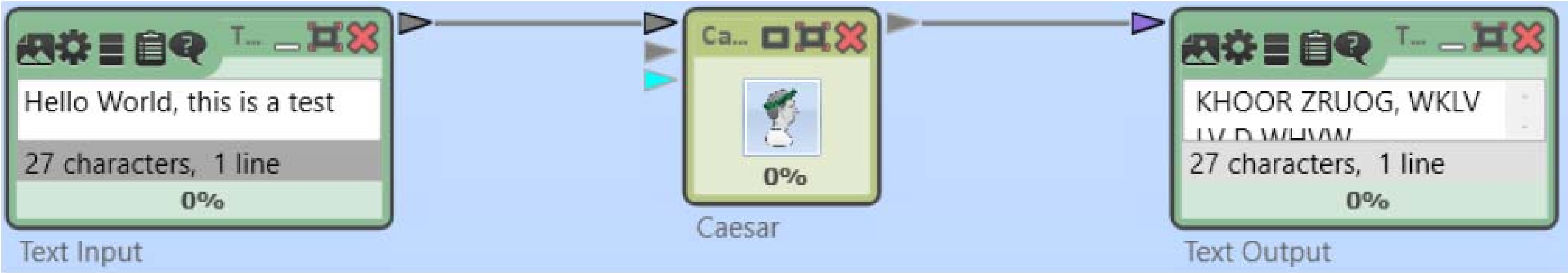
KHOOR ZRUOG, WKLV
LV D WHVW

27 characters, 1 line

0%

Example

- Building a Workflow for the Caesar Cipher
  - Update input/output on the fly

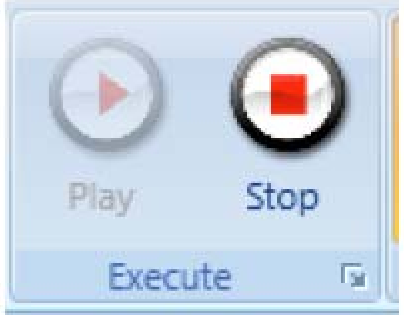


Hello World, this is a test

27 characters, 1 line

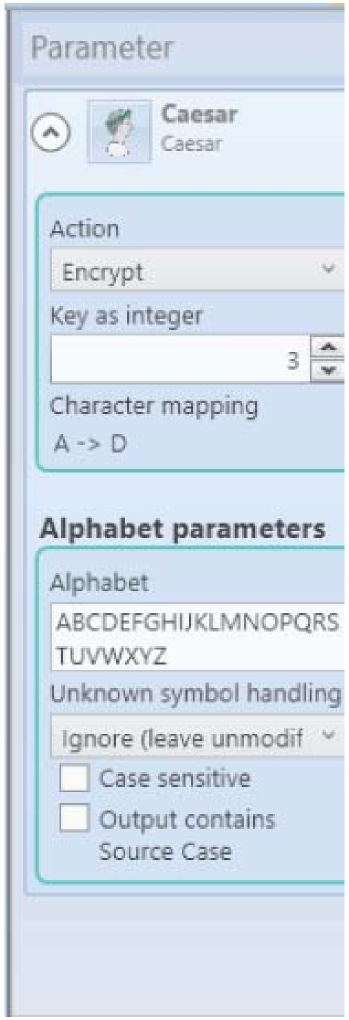
KHOOR ZRUOG, WKLV LV D WHVW

27 characters, 1 line



Example

- Building a Workflow for the Caesar Cipher
  - Change significant data



Parameter

Caesar
Caesar

Action: Encrypt

Key as integer: 3

Character mapping: A -> D

Alphabet parameters

Alphabet: ABCDEFGHIJKLMNOPQRSTUVWXYZ

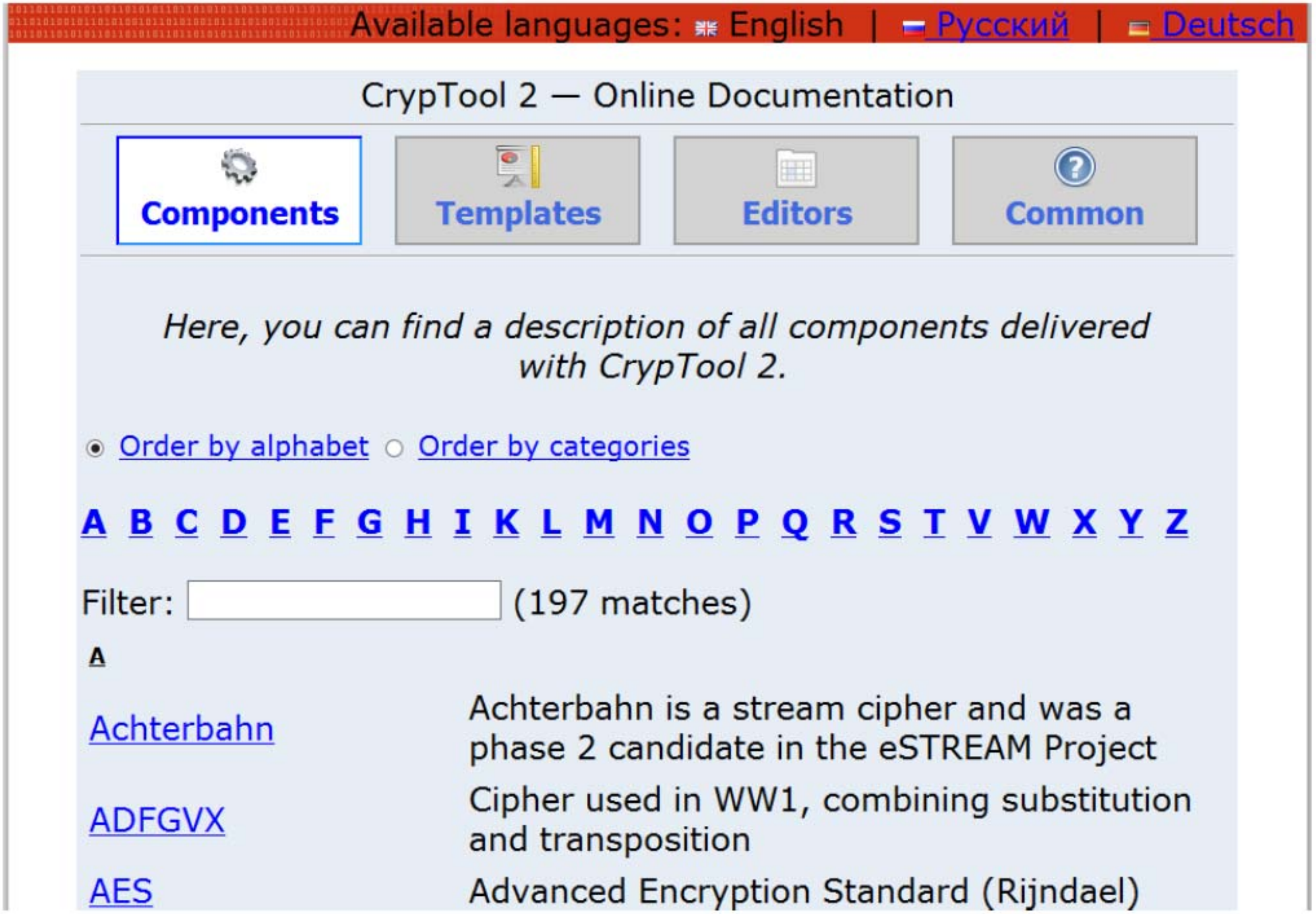
Unknown symbol handling: Ignore (leave unmodif

☐ Case sensitive

☐ Output contains Source Case

Online help

- Get information of each component



CrypTool 2 — Online Documentation

Here, you can find a description of all components delivered with CrypTool 2.

● Order by alphabet ○ Order by categories

A B C D E F G H I K L M N O P Q R S T V W X Y Z

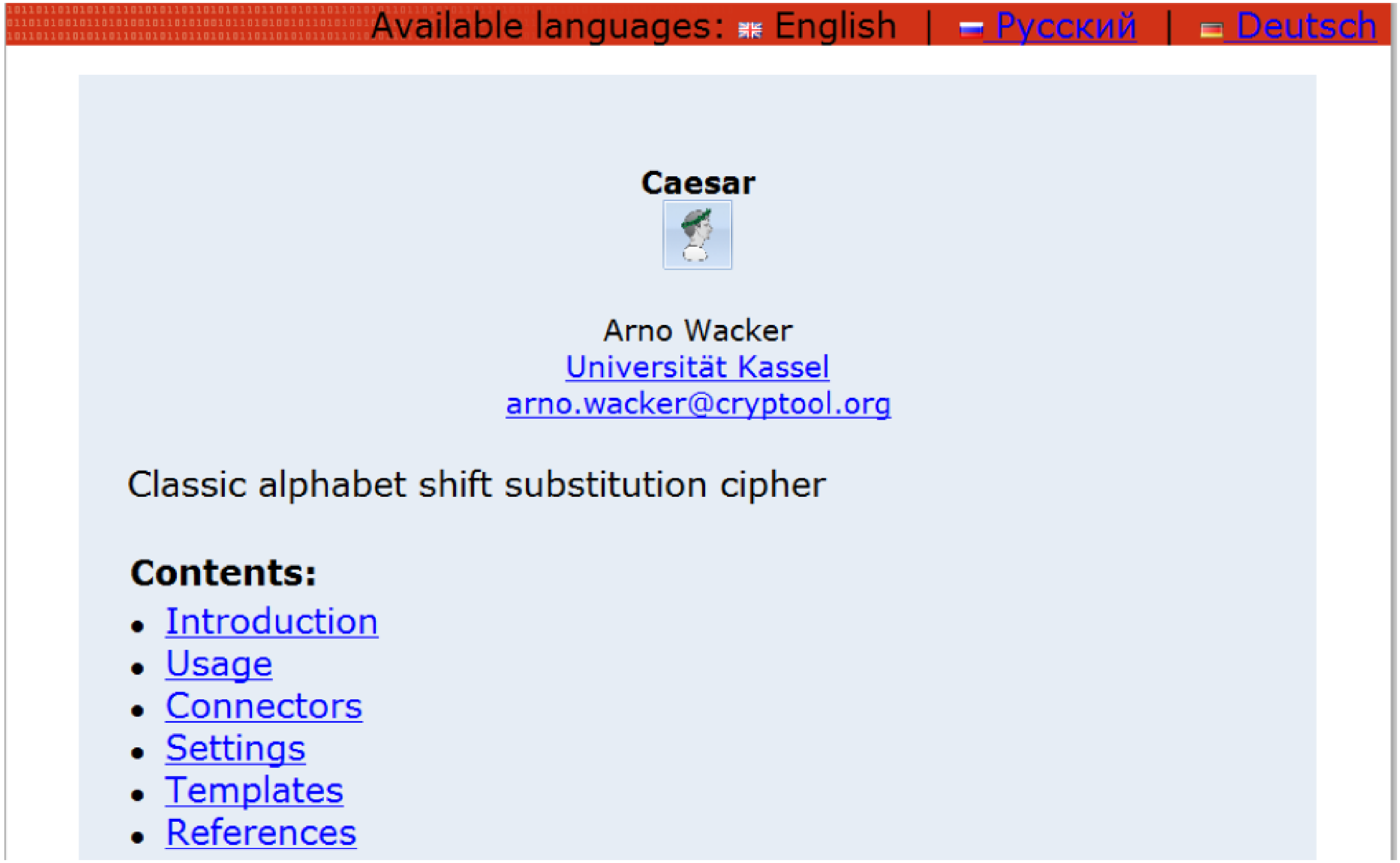
Filter: (197 matches)

A

Achterbahn	Achterbahn is a stream cipher and was a phase 2 candidate in the eSTREAM Project
ADFGVX	Cipher used in WW1, combining substitution and transposition
AES	Advanced Encryption Standard (Rijndael)

Online help

- Look for information of a specific component



Available languages: English | Русский | Deutsch

Caesar

Arno Wacker
Universität Kassel
arno.wacker@cryptool.org

Classic alphabet shift substitution cipher

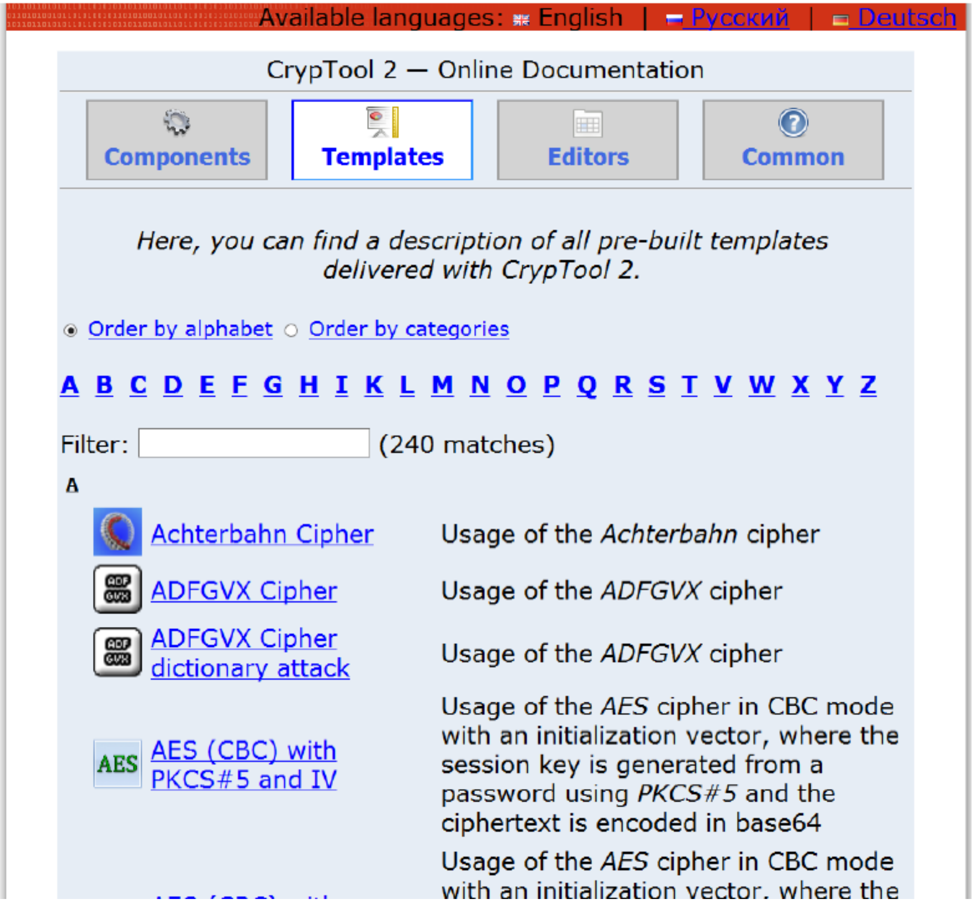
Contents:

- Introduction
- Usage
- Connectors
- Settings
- Templates
- References

Module 4: Cryptology for IoT

Online help

- Get information of each template



Available languages: English | Русский | Deutsch

CrypTool 2 — Online Documentation

Here, you can find a description of all pre-built templates delivered with CrypTool 2.

● Order by alphabet ○ Order by categories

A B C D E F G H I K L M N O P Q R S T V W X Y Z

Filter: (240 matches)

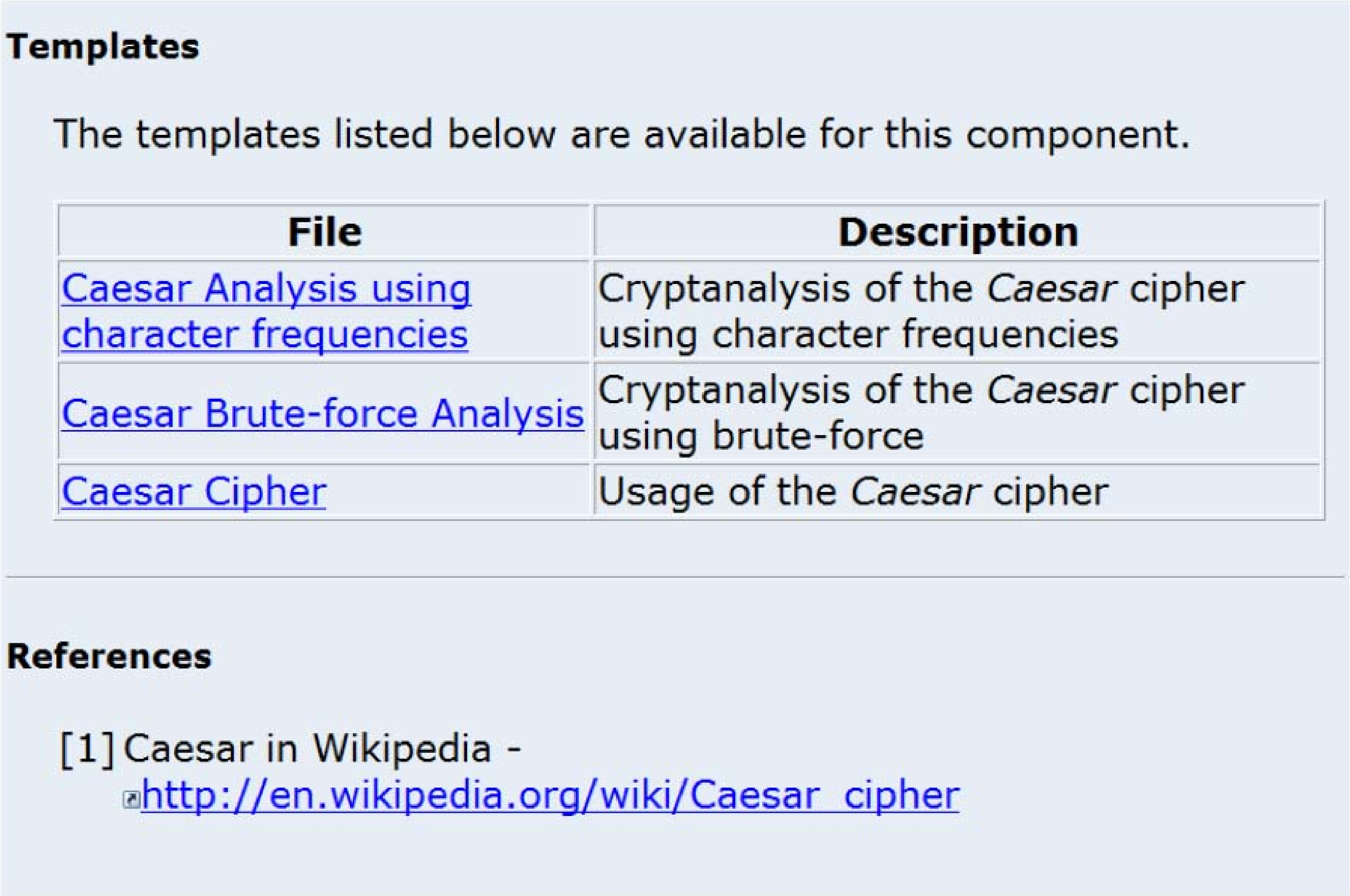
A

Achterbahn Cipher	Usage of the *Achterbahn* cipher
ADFGVX Cipher	Usage of the *ADFGVX* cipher
ADFGVX Cipher dictionary attack	Usage of the *ADFGVX* cipher
AES (CBC) with PKCS#5 and IV	Usage of the *AES* cipher in CBC mode with an initialization vector, where the session key is generated from a password using *PKCS#5* and the ciphertext is encoded in base64
AES (CBC) with	Usage of the *AES* cipher in CBC mode with an initialization vector, where the

Sec

Online help

- Get information of a specific template



Templates

The templates listed below are available for this component.

File	Description
Caesar Analysis using character frequencies	Cryptanalysis of the *Caesar* cipher using character frequencies
Caesar Brute-force Analysis	Cryptanalysis of the *Caesar* cipher using brute-force
Caesar Cipher	Usage of the *Caesar* cipher

References

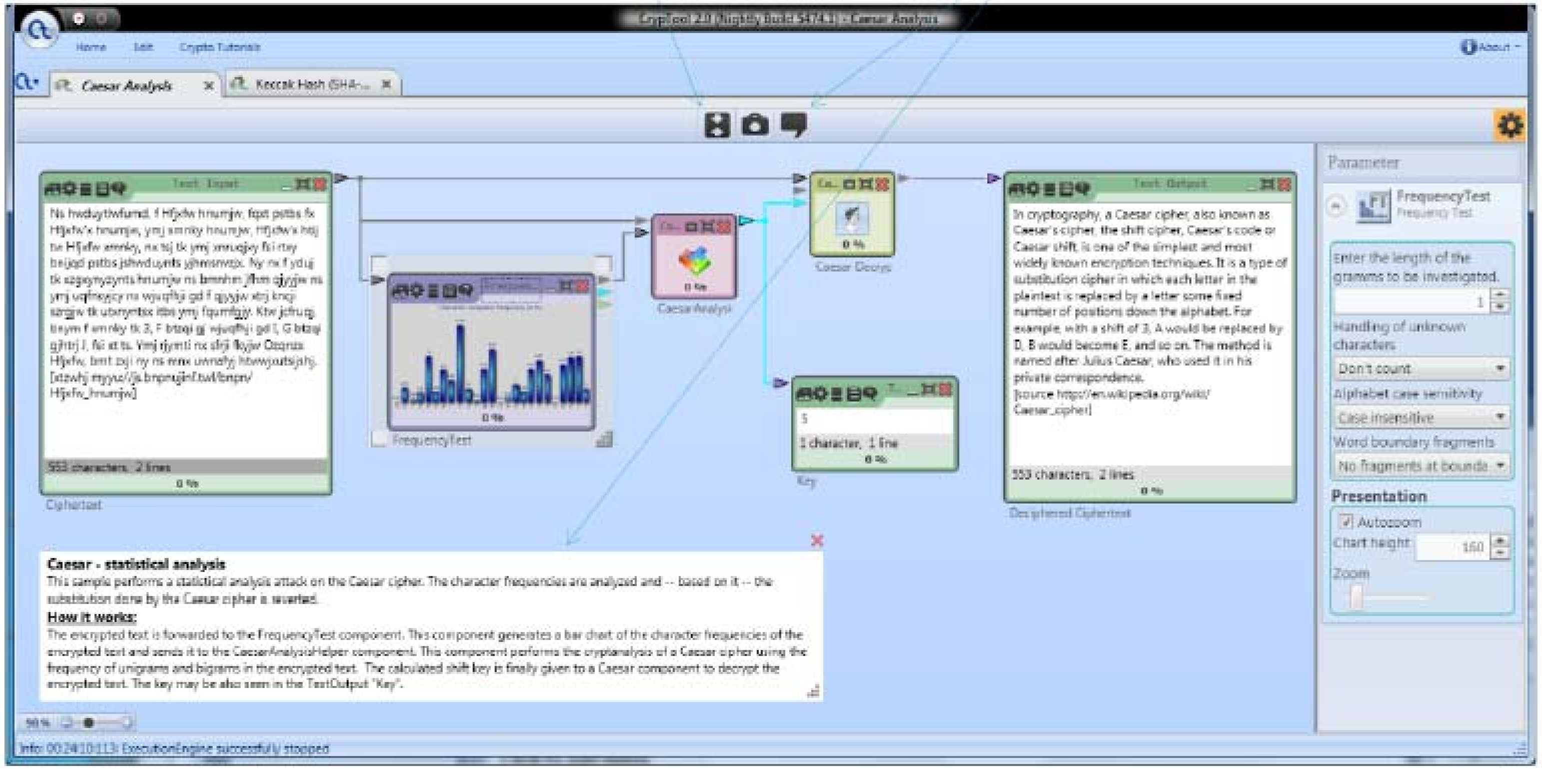
[1] Caesar in Wikipedia - http://en.wikipedia.org/wiki/Caesar_cipher

Remarks

- Quickly adapt the layout to you needs

Fit with one click
to workspace size

Add text field (memo)
to workspace



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

553 characters, 2 lines

5

1 character, 1 line

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

553 characters, 2 lines

Parameter

FrequencyTest
Frequency Test

Enter the length of the grams to be investigated. 1

Handling of unknown characters: Don't count

Alphabet case sensitivity: Case insensitive

Word boundary fragments: No fragments at bounds

Presentation

☑ Autozoom

Chart height 160

Zoom

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

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

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

Some links

- Download the tool:

https://www.cryptool.org/en/ct2-download

- CrypTool 2 Wiki:

https://www.cryptool.org/trac/CrypTool2/

- CrypTool Project / CrypTool Portal:

https://www.cryptool.org/

- CrypTool Project at Wikipedia:

https://en.wikipedia.org/wiki/CrypTool

- Cryptool Video (Short Introduction)
- Book:

https://www.cryptool.org/en/ctp-documentation/ctbook

Appendix

Extra readings:

- Nils Kopal: Solving Classical Ciphers with CrypTool 2, 2018, http://www.ep.liu.se/ecp/149/010/ecp18149010.pdf
- G. Lasry, N. Kopal, A. Wacker: Solving the Double Transposition Challenge with a Divide-and-Conquer Approach. In: Cryptologia, 38, 3 (2014), 197–214
- G. Lasry, N. Kopal, A. Wacker: Ciphertext-only Cryptanalysis of Hagelin M-209 Pins and Lugs. In: Cryptologia, 40, 2 (2016), 141–176
- G. Lasry, N. Kopal, A. Wacker: Cryptanalysis of Columnar Transposition Cipher with Long Keys. In: Cryptologia, 40, 4 (2016), 374–398
- G. Lasry: A Methodology for the Cryptanalysis of Classical Ciphers with Search Metaheuristics. kassel university press GmbH (2018), http://www.upress.uni-kassel.de/katalog/abstract.php?978-3-7376-0458-1
- G. Lasry, I. Niebel, N. Kopal, A. Wacker: Deciphering ADFGVX Messages from the Eastern Front of World War I. In: Cryptologia, 41, 2 (2017), 101–136
- An overview about the whole CrypTool project including more modern algorithms (like post-quantum signatures in JCT): http://fg-krypto.gi.de/fileadmin/fg-krypto/CrypTool-Project_Crypto_Day_Walldorf_2016-09_v09.pdf

Cryptology for IoT

Modules M4, M7, M9
Session of 26th April, 2022.

M4. Introduction to the modules
M4.1 Introduction to the Cryptology
M4.2 Introduction to Cryptool CT2

Prof.: Guillermo Botella