VILNIAUS GEDIMINO
TECHNIKOS UNIVERSITETAS

VILNIAUS GEDIMINO
TECHNIKOS UNIVERSITETAS
FUNDAMENTINIŲ MOKSLŲ FAKULTETAS

VILNIUS
TECH
Faculty of Fundamental
Sciences

Cryptography and Information Security

FMISB18500

PRACTICE, LABORATORY WORKS, HOME WORKS, RESEARCH ESSAY

Lect. P. Narkevičius

Vilnius Tech
Vilnius, 2024

In total: 96 Pages.

Summary

- Following material is prepared for FMISB18500 module.
- All necessary information for practice of Cryptography and Information Security in 2024-2025 Semester.

Contents

4 Introduction

12 Ancient cipher, its performance and security

21 ¤ Task for Ancient cipher, its performance and security

22 Public key cryptography usage, PGP application

31 ¤ Task for Public key cryptography, PGP app

32 Check digit security, Hash function application

37 ¤ Task for Check digit security, Hash f-tion app

38 Web system security testing and prevention

40 ¤ Task for Web system security testing and prevention

41 Caesar and Vigenère cipher, its implementation principles

43 ¤ Task for Caesar and Vigenère cipher, its implementation principles

44 Brute-force attacks and its complexity

50 ¤ Task for Brute-force attacks and its complexity

51 RSA algorithm usage for data encoding and digital signature (Simplified)

63 ¤ Task for RSA algorithm usage for data encoding and digital signature (Simplified)

64 Hash function usage for password security level increase

67 ¤ Task for Hash function usage for password security level increase

68 Steganography usage for data hiding

73 ¤ Task for Steganography usage for data hiding

74 Bots and ways to prevent it

80 ¤ Task for Bots and ways to prevent it

81 Web page security threats and protections against it

84 ¤ Task for Web page security threats and protections against it

85 Vulnerability testing

92 ¤ Task for Vulnerability testing

93 Research Essay (incl. Task)

02/09/2024

Introduction

Assessment of the Student Works

PRACTICE
- no less than four (4) Tasks.

LABORATORY WORKS
- no less than eight (8) Tasks.

RESEARCH ESSAY
- Task to write one Ressearch Essay which could be selected from Available Bibliography from Cryptography and Information Security in the University.

HOME WORK
- to organize your Results into single Document which include your completed PRACTICE, LABORATORY WORKS and RESEARCH ESSAY. Try to implement security measure for this Document, i.e. Encryption.

WARNING

+ if all PRACTICE, LABORATORY WORKS, RESEARCH ESSAY, HOME WORKS will not be finished during the Semester then additional Task will be given.

Evaluation (1)

About Evaluations of these Tasks for 2024-2025 Semester: PRACTICE, LABORATORY WORKS, RESEARCH ESSAY, HOME WORKS.

Lecturer evaluates Student works according Performance assessment table:

Performance level	Grade	Description
Outstanding	10/9	Exceptional/Strong
Typical	8/7	Above average/avg.
Threshold	6/5	Below avg/minimum
Failed	0-4	Not acceptible & / repeat required

Procedure Description for Individual's Studying Performance Assessment and Earning Credits at Vilnius Gediminas Technical University approved by Resolution of Vilnius Gediminas Technical University Senate No. 10.141-38 on 27/06/2023

Source

Evaluation (2)

Full-time studies

FA = E * 0.30 + SE * 0.20 + HW * 0.30 + RE * 0.20
FA - Final Assessment
E - Exam (full credit)
SE - Semi-exam
HW - Homework (avg 3 interim credits)
RE - Research essay

Part-time distance studies

FA = E * 0.50 + HW * 0.30 + RE * 0.20
FA - Final Assessment
E - Exam
HW - Homework (avg 3 interim credits)
RE - Research essay

PRACTICE

Tasks for PRACTICE:

1. Ancient cipher, its performance and security.

2. Public key cryptography usage, PGP application.

3. Check digit security, Hash function application.

4. Web system security testing and prevention.

LABORATORY WORKS

Tasks for LABORATORY WORKS:

5. Caesar and Vigenère cipher, its implementation principles.

6. Brute-force attacks and its complexity.

7. RSA algorithm usage for data encoding and digital signature (Simplified).

8. Hash function usage for password security level increase.

9. Steganography usage for data hiding.

10. Bots and ways to prevent it.

11. Web page security threats and protections against it.

12. Vulnerability testing.

RESEARCH ESSAY

References for RESEARCH ESSAY:

A. Reading List, which provided in mano.vilniustech.lt

B1. Literature, which are Available in Vilnius Tech Library (and related to Cryptography & Information Security)

B2. Literature, which are Available in Vilnius Tech Reading Room (and related to Cryptography & Information Security)

C. Literature, which are Available in Vilnius Tech Virtual Library (and related to Cryptography & Information Security)

HOME WORK

Join Your PRACTICE Results and LABORATORY WORKS Results and RESEARCH ESSAY Results into One-Single Document. Provide this Document to the Lecturer.

01/09/2023

Ancient cipher [1/13]p

Prep. Modular arithmetic (1)

• Purpose - let us align ancient ciphers with modern computing and let begin from the Modular arithmetics.

• It's system of arithmetic for integers, which operations are made using modulus n.

• Given any positive integer n and any non-negative integer a, if we divide a by n, we get an integer quotient q and an integer remainder r that obey following condition:

$$a = q \times n + r \qquad (1)$$

$$0 \leq r < n$$

$$q = \lfloor a \div n \rfloor$$

• If a is integer and n — positive integer, then, we can define this condition:

$$(a \bmod n) = r \qquad (2)$$

$$a = \lfloor a \div n \rfloor \times n + (a \bmod n)$$

Prep. Modular arithmetic (2)

•Fill out the answers from modular arithmetic equations:

3 mod 8 = … 3
13 mod 8 = … 5

-9 mod 8 = …-1
-19 mod 8 = … -3

Prep. Caesar cipher - Math expr

- $\underline{c} = E(p, k)$
- $p = D(\underline{c}, k)$

Plaintext message – p
Encrypted message – $\underline{c}$
Secret key – k
Encryption algorithm – E
Decryption algorithm – D

- Set of variables is used where for each value are given mathematical numerical value which increases by ascending order starting from zero.

In example:
set consist from 26 (twenty six) values ABCDEFGHIJKLMNOPQRSTUVWXYZ, then numerical values would start from A - 0 (zero) and end with Z - 25 (twenty five)

- Math epression for encryption:

$\underline{c} = E(p,k) = (p+k) \bmod (n)$ (3)

- Math expression for decryption:

$p = D(\underline{c},k) = (\underline{c}-k) \bmod (n)$ (4)

Prep. Caesar cipher

• And this how common Caesar Cipher works — using secret key (k) as value 3 (three) and count number (n) of all items in set of variables by starting to count from zero.

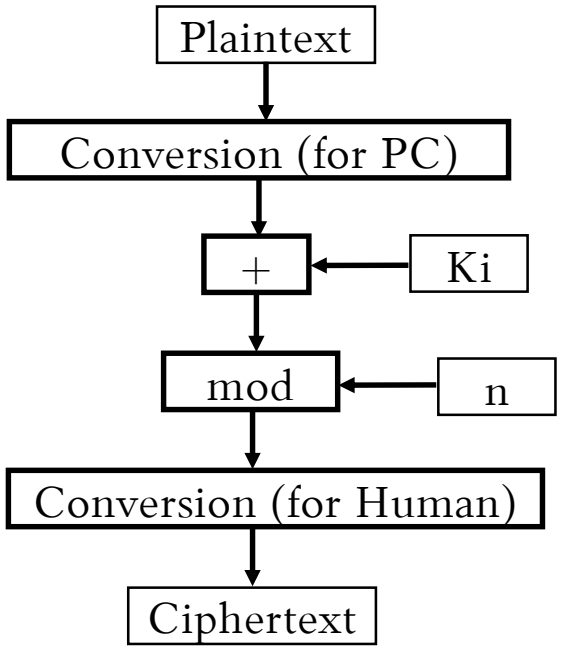
• Sample:

```
  H E  L  L  O
  7  4 11 11 14
+ 3  3  3  3  3
---------------
(10  7 14 14 17) mod 26
 10  7 14 14 17
---------------
  K  H  O  O  R
```

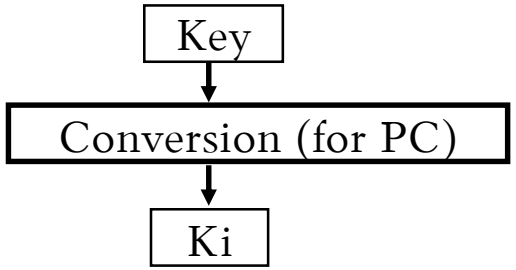
```
  K  H  O  O  R
  10 7 14 14 17
- 3  3  3  3  3
---------------
( 7  4 11 11 14) mod 26
  7  4 11 11 14
---------------
  H  E  L  L  O
```

Caesar Cipher

Function

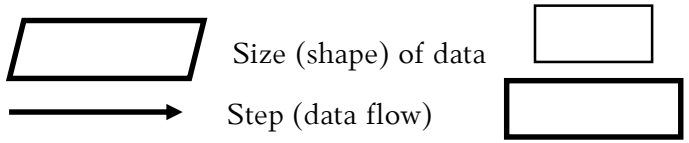


Key



where
$i \in Z^{0+}$
$i \in [0, n-1]$

Legenda:



Mandatory to know Modular Arithmetic and Latin or Roman Alphabet

Here displayed Encryption Part of the Classical (Historical) Cipher Caesar

Values displayed in (Key):
n – length of the Alphabet
i – one selected value depending by size of n
Conversion (for PC) – preparation for computation with device like PC

Values displayed in (Function):
+ – addition of two values
mod – function for modular arithmetic, in the output only positive value up to modulus
Conversion (for PC) – preparing all the values into computation feasible shape for devices like PC
Conversion (for Human) – preparing all the values into view for people to read.
p.s. Conversion is Optional when we talk about Computer-Computer or Human-Human interaction.

Prep. Vigenère cipher - Math expr

•ci = Ek(pi,k) = (pi + ki) mod (n) (5)
•pi = Dk(ci,k) = (ci – ki) mod (n) (6)
•Ek – Vigenère encryption function using secret key k.
•Dk – Vigenère decryption function using secret key k.
•p = p1 ...pn – plaintext.
•c = c1 .....cn – encrypted text.
•k = k1 ...kn – secret key, which we get by repeating key n/m times, where m are secret key length.
•n = count number of all items in set of variables by starting to count from zero, i.e. ABCDEFGHIJKLMNOPQRSTUVWXYZ would be 26.

Prep. Vigenère cipher (1)

• Let us build cheat-sheet table for Vigenère cipher, if we will use the set of variables as shown in previous example: this will be 26x26 table, zero horizontal line - plaintext header where the set of variables is located, 1st horizontal line - repeatition of set of variable with secret key zero, 2nd horizontal line - set of variables which are shifted by one variable to the left where first variable from the start is going to the end of the set, 3rd-26th horizontal line - for each line repeating these shifts by one.

• Horizontal Header values - Plaintext.

• Vertical Header values - Key.

• Inside the Vigenère table - Ciphertext.

• Encryption: Fill secret key value by repeating and concatinating it with previous stack until it will be with the same lenght or bigger as Plaintext. Align Key position with Plaintext position. Start encrypting plaintext one by one by drawing line from both Headers and where they intersect there gonna be one value of Ciphertext.

• Decryption: Fill secret key value by repeating and concatinating it with previous stack until it will be with the same lenght or bigger as Ciphertext. Align the Key with Ciphertext value.

Prep. Vigenère cipher (2)

• Vigenère tableu:

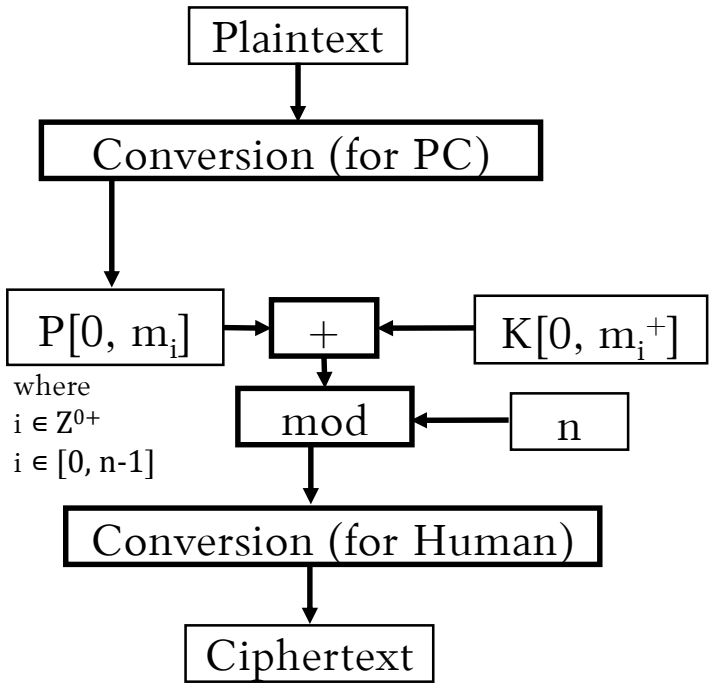
| ABCDEFGHIJKLMNOPQRSTUVWXYZ

A | ABCDEFGHIJKLMNOPQRSTUVWXYZ
B | BCDEFGHIJKLMNOPQRSTUVWXYZA
C | CDEFGHIJKLMNOPQRSTUVWXYZAB
D | DEFGHIJKLMNOPQRSTUVWXYZABC
E | EFGHIJKLMNOPQRSTUVWXYZABCD
F | FGHIJKLMNOPQRSTUVWXYZABCDE
G | GHIJKLMNOPQRSTUVWXYZABCDEF
H | HIJKLMNOPQRSTUVWXYZABCDEFG
I | IJKLMNOPQRSTUVWXYZABCDEFGH
J | JKLMNOPQRSTUVWXYZABCDEFGHI
K | KLMNOPQRSTUVWXYZABCDEFGHIJ
L | LMNOPQRSTUVWXYZABCDEFGHIJK
M | MNOPQRSTUVWXYZABCDEFGHIJKL
N | NOPQRSTUVWXYZABCDEFGHIJKLM
O | OPQRSTUVWXYZABCDEFGHIJKLMN
P | PQRSTUVWXYZABCDEFGHIJKLMNO
Q | QRSTUVWXYZABCDEFGHIJKLMNOP
R | RSTUVWXYZABCDEFGHIJKLMNOPQ
S | STUVWXYZABCDEFGHIJKLMNOPQR
T | TUVWXYZABCDEFGHIJKLMNOPQRS
U | UVWXYZABCDEFGHIJKLMNOPQRST
V | VWXYZABCDEFGHIJKLMNOPQRSTU
W | WXYZABCDEFGHIJKLMNOPQRSTUV
X | XYZABCDEFGHIJKLMNOPQRSTUVW
Y | YZABCDEFGHIJKLMNOPQRSTUVWX
Z | ZABCDEFGHIJKLMNOPQRSTUVWXY

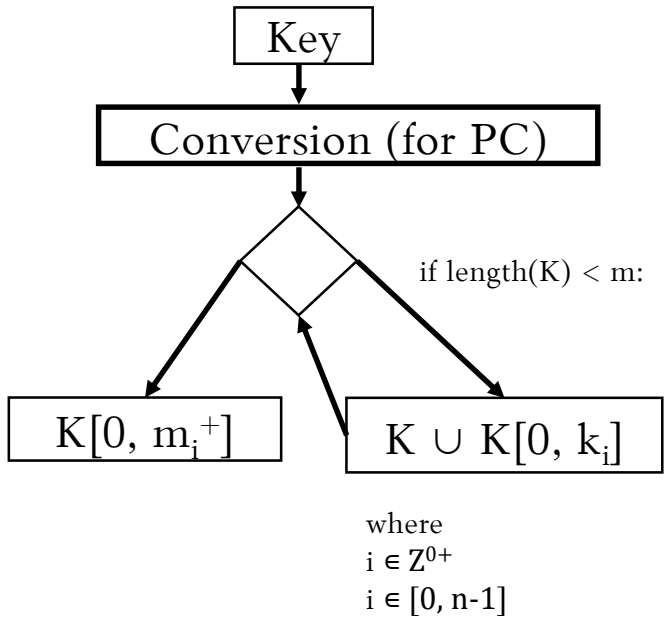
Does Plaintext value of
WE ARE DISCOVERED SAVE
YOURSELF with Secret Key VGTU
would produce Ciphertext of:
RKTLZJBMXUOYMKWMVBXSJAKMZRY
?

Vigenère Cipher

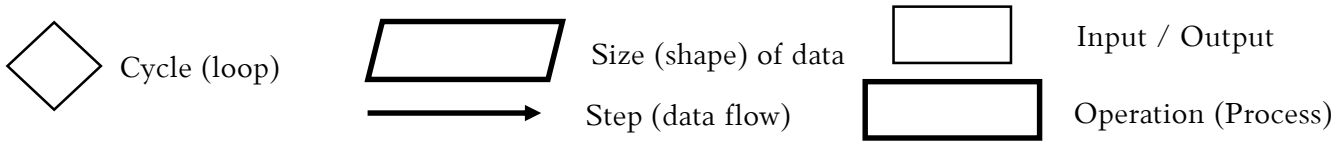
Function



Key



Legenda:



Mandatory to know Modular Arithmetic and Latin or Roman Alphabet

Here displayed Encryption Part of the Classical (Historical) Vigenère Cipher

Values displayed in (Key):
n – length of the Alphabet
k – length of the Original Key
m – length of the Plaintext (Cleartext)
i – value from zero to n
∪ — Union operation
Conversion (for PC) – preparation for computation with device like PC

Values displayed in (Function):
+ – addition of two values
mod – function for modular arithmetic, in the output only positive value up to modulus
Conversion (for PC) – preparing all the values into computation feasible shape for devices like PC
Conversion (for Human) – preparing all the values into view for people to read.
p.s. Conversion is Optional when we talk about Computer-Computer or Human-Human interaction.

Prep. Miscellaneous

Ancient and Modern Ciphers could be expressed in Symbols, Heroglyphs, Letters, Numbers and Paintings.

Computer Systems expresses many writtings in Fonts, which contain excluding the Paintings all the previously mentioned Elements. In other Hand, the Fonts could contain Paintings in Form of Emojis, which are very popular in nowadays.

The Caesar web Application is Available. This web App encrypts and decrypts Messages using Lithuanian Letters, so it's extended non-original Caesar alike Cipher web App. This web App shows Results in Custom "Palemonas" Font, which were created by the Lithuanian Academics. Link to this web App:

https://info.issauga.lt/Caesar-cipher/index.html

P1 Ancient cipher

• Task:

1. Choose Message & Secret Key and encrypt Plaintext with Caesar and Vigenère Algorithms.
2. Encrypt Plaintext w/o any Application(s) and with Encryption Process of the both Ciphers.
3. Decrypt Ciphertext w/o any Application(s) and with Decryption Process of the both Ciphers.
4. Fill up Your Results.

• (Optional) Task(s):

I. Compare Performance of the Ancient and Modern Ciphers.
II. Compare Security of the Ancient and Modern Ciphers.

• Ref.

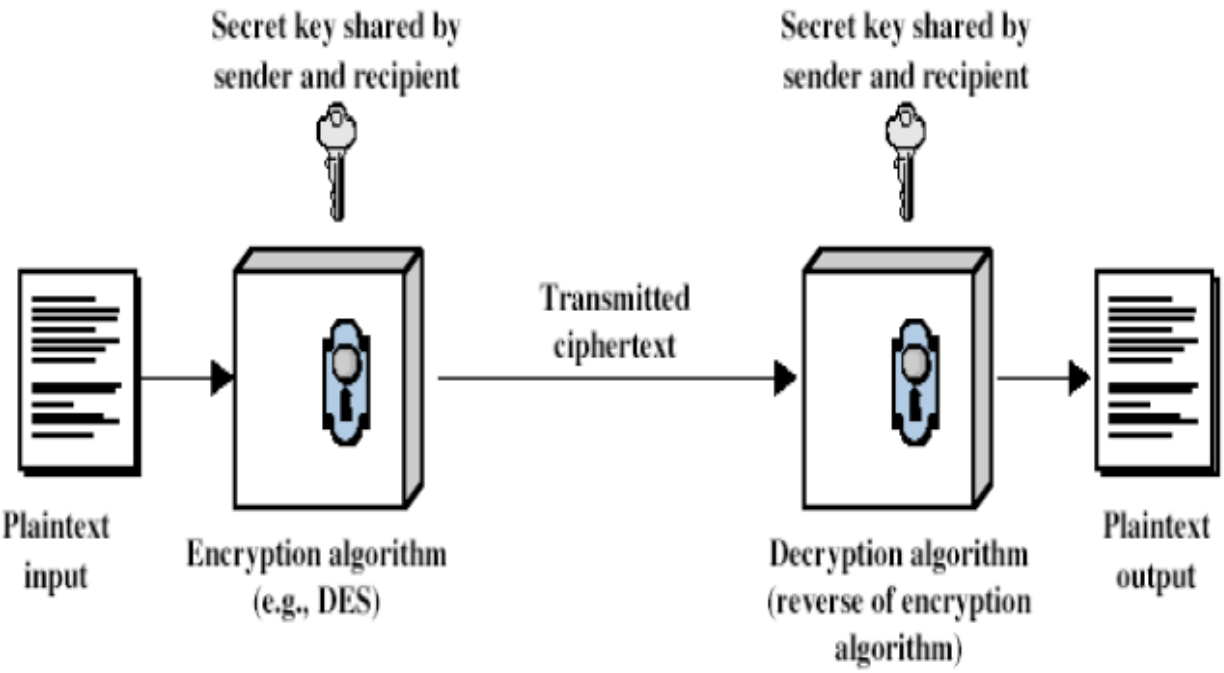
Outstanding — by successfully finishing Task No. 1-4.
Typical — by completing Encryption or Decryption Process with only using Single Cipher Algorithm(s).
Threshold — by completing Task(s) with help of the App or by lack of Evidence (i.e. only Ciphertext is given).

01/09/2023

Public key Cryptography, PGP [2/13]p

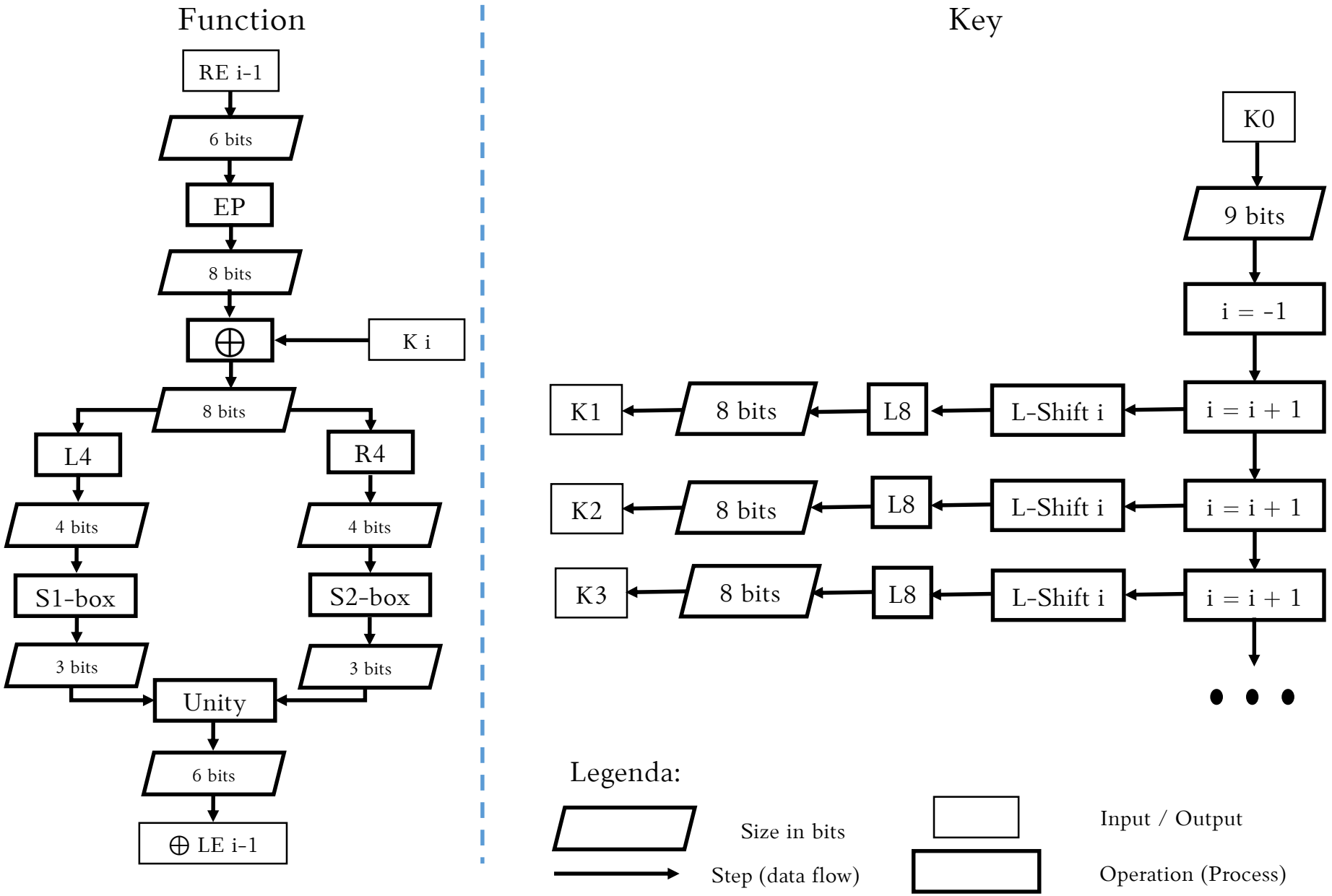
Prep. Symmetrical approach

• Purpose - let us distinguish symmetrical and asymmetrical approaches of Cryptography Ciphers.
• The same Secret Key used which has to be shared with all parties with who needs to communicate.
• The Secret Key has to be securely trasfered (problem).
• The Secret Key requires to be stored securely (problem).
• The algorithm has to be well known and secure with proper parameters of Secret Key (problem).



© http://www.slideshare.net/lineking/classical-encryption-techniques-in-network-security

SDES 2002



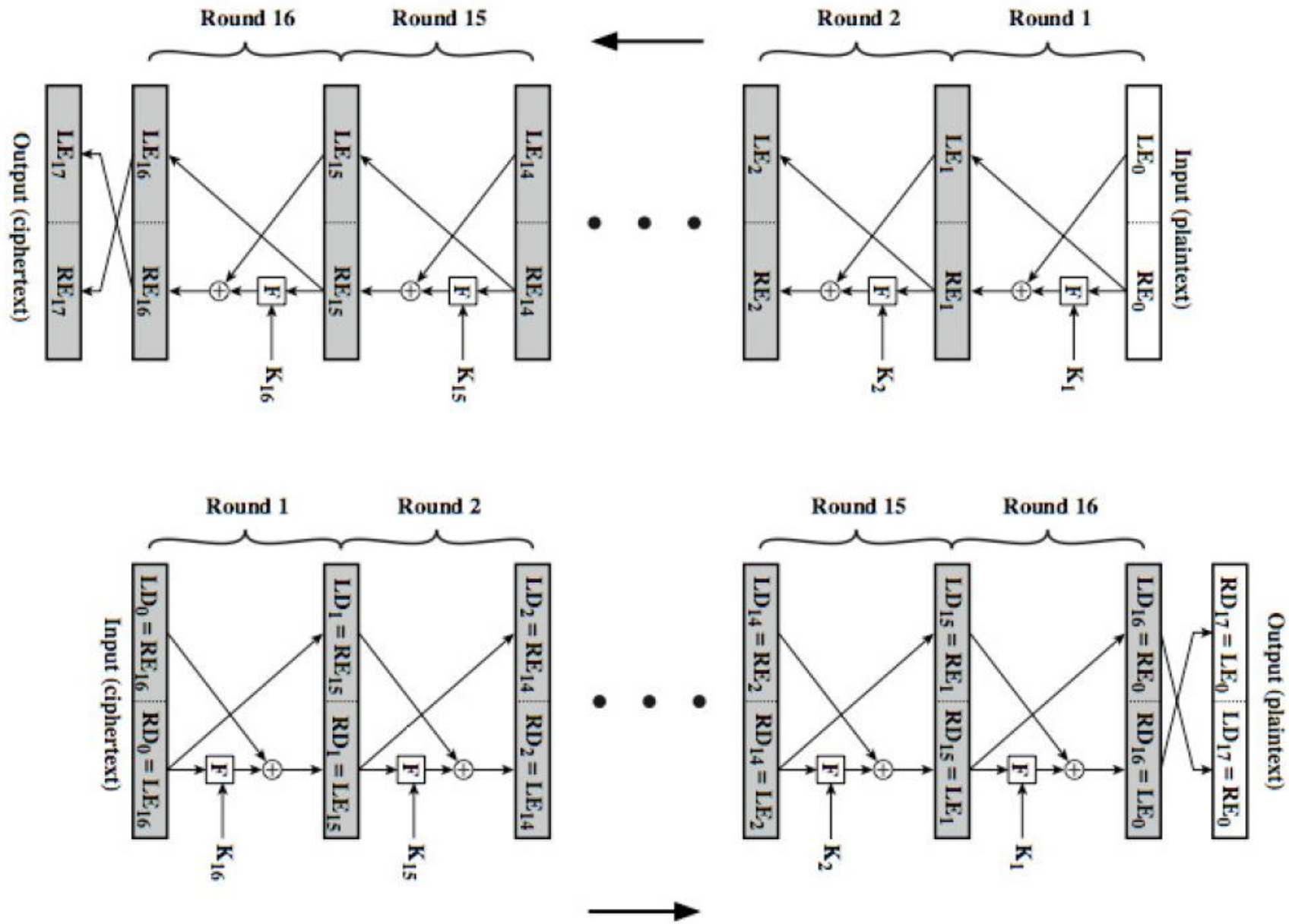
Optional to know about Feistel-like Encryption and Decryption

Here displayed Encryption Part of the SDES 2002, when we own Key – 9 bit‘s and Plaintext – 12 bit‘s.

Values displayed in (Key):
K – Key
L-Shift – sliding values to the Left, who goes out of scope, add them from the Right
L8 – 8 bit‘s from the Left

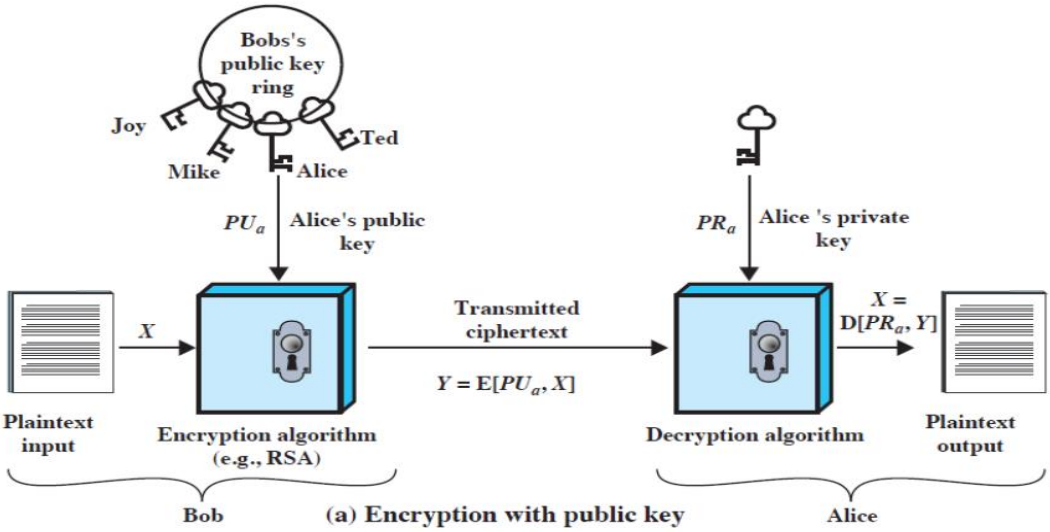
Values displayed in (Function):
RE – Right Encryption
EP – Expansion Permutation
⊕ – XOR‘ing
L4 – 4 bit‘s from the Left
R4 – 4 bit‘s fromt the Right
S1-box – first-bit is a row and last bit‘s – column
S2-box – first-bit is a row and last bit‘s – column
Unity – joining Left bit‘s with Right bit‘s

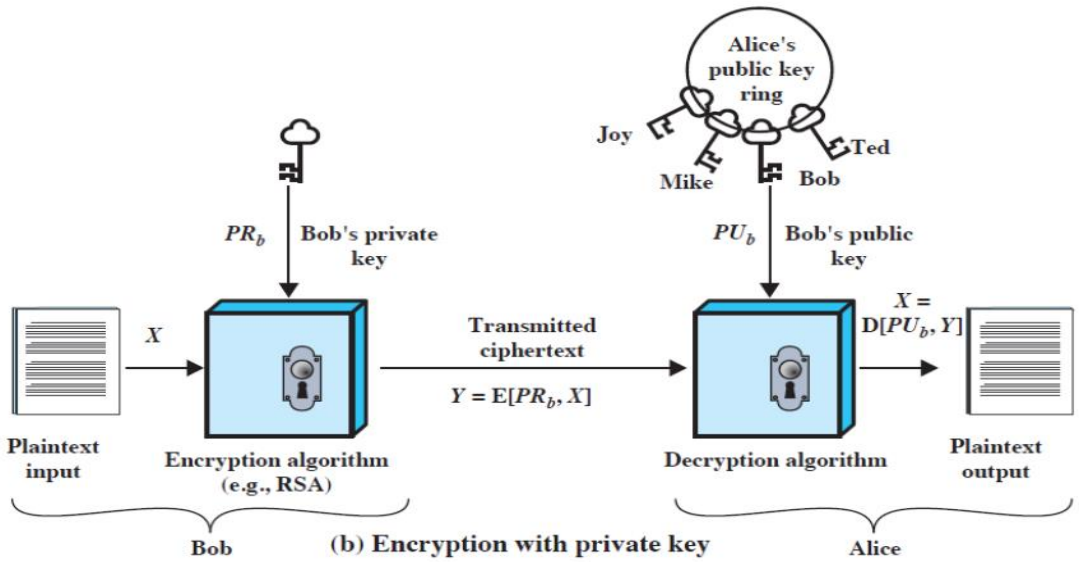
Feistel-like Encryption and Decryption



Prep. Asymmetrical approach

• More than one Key is used for this approach where one-part has be kept securely or private Key and other-part which is shared with the intended party for proper communication or public Key.





Prep. Rabin Algorithm

Rabin Algorithm/Cryptosystem were created by Michael O. Rabin.

This type of algorithm uses multiplication of two huge prime numbers and both of those numbers has to be not identical.

Public key - result of multiplication of primes.
Private key - each of prime values who were used in constructing Public key.

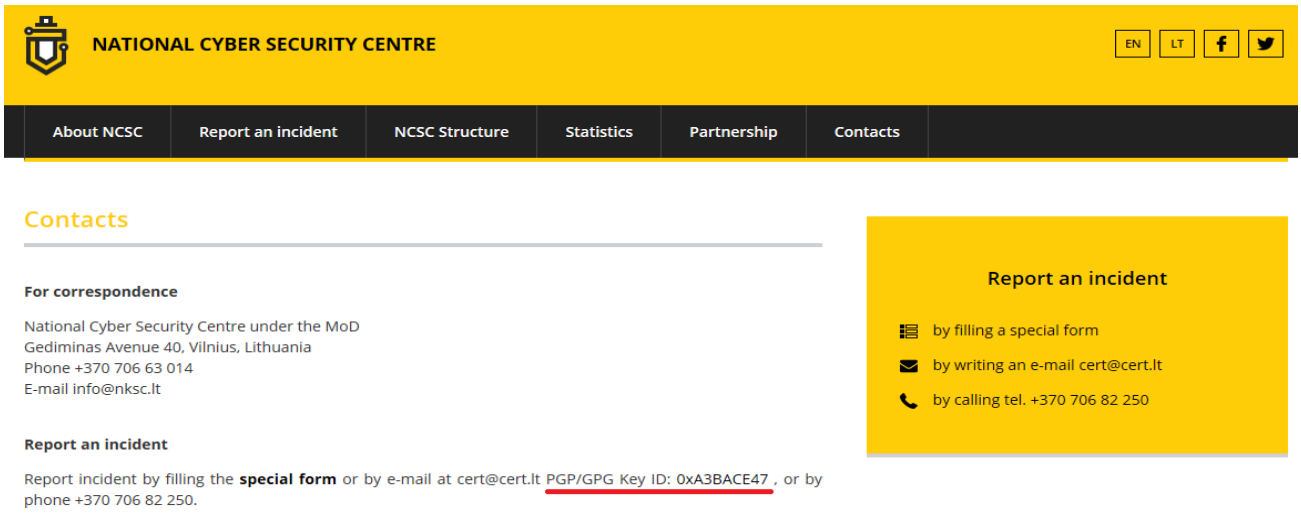
This algorithm is pretty simple at first:
• Choosing Message with values who is less than Public key value.
• For encryption is used modular arithmetic and power operation (squared) by two to encrypt this plaintext Message.
• For decryption is used modular arithmetic and square root of ciphertext Message.

Prep. Rabin Sample

Link to the Sample of Rabin algorithm:
https://info.issauga.lt/Rabin-algorithm/matr.bandy.html#One

Prep. Pretty Good Privacy

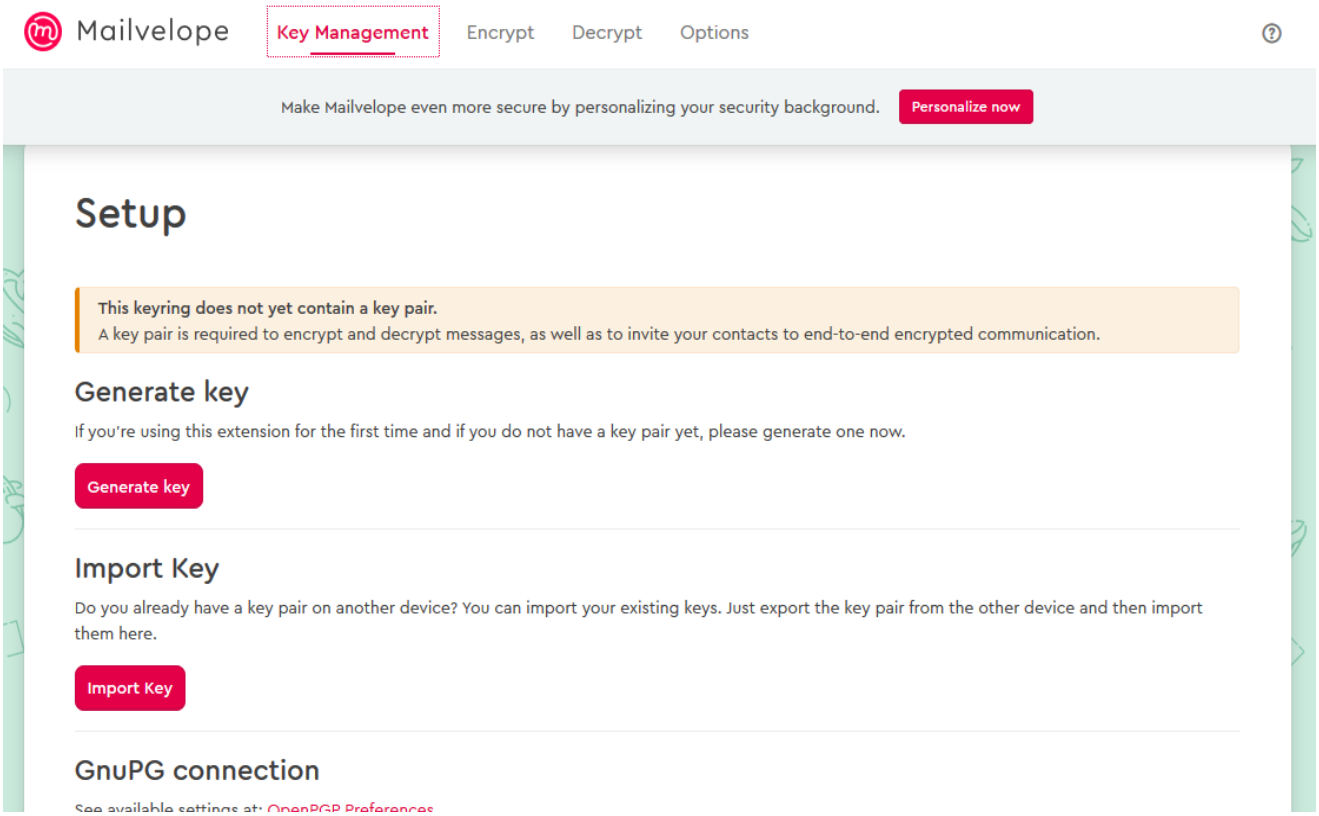
- De facto secure email
- Phil Zimmermann — the author of the Pretty Good Privacy (PGP) — Source: Why I Wrote PGP
- Available in various platforms
- Confidentiality & Authentication procedures is used on the same message:
    - signature is attached to the message
    - both of them are being encrypted
    - attached encrypted session key, e.g. RSA
- Used in incident reporting, i.e. https://www.nksc.lt/en/contacts.html



Prep. Mailvelope (1)

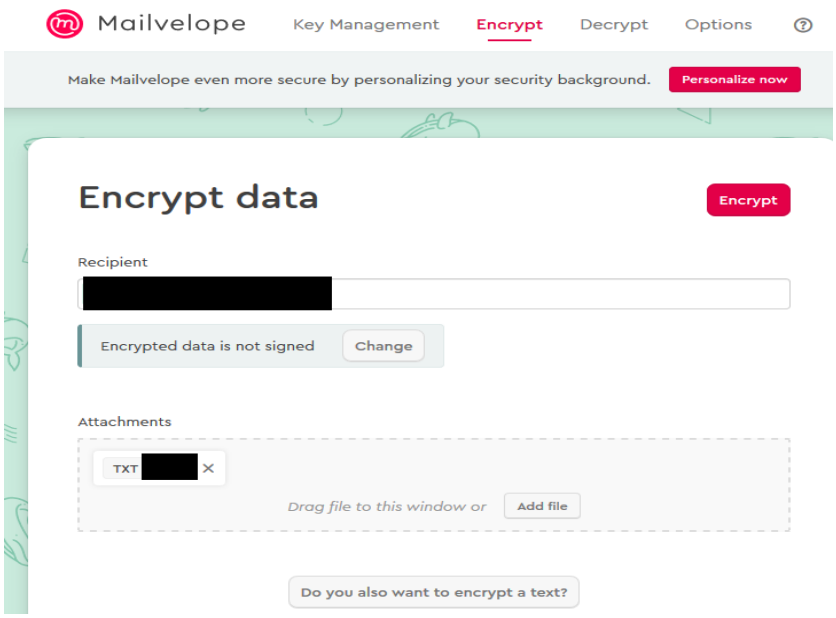
• Mailvelope — browser add-on, which allows using Chrome, Edge and Firefox browsers to extend information security of electronic mail by doing encryption with PGP using selected webmail services.
• You should be able to access it in https://www.mailvelope.com

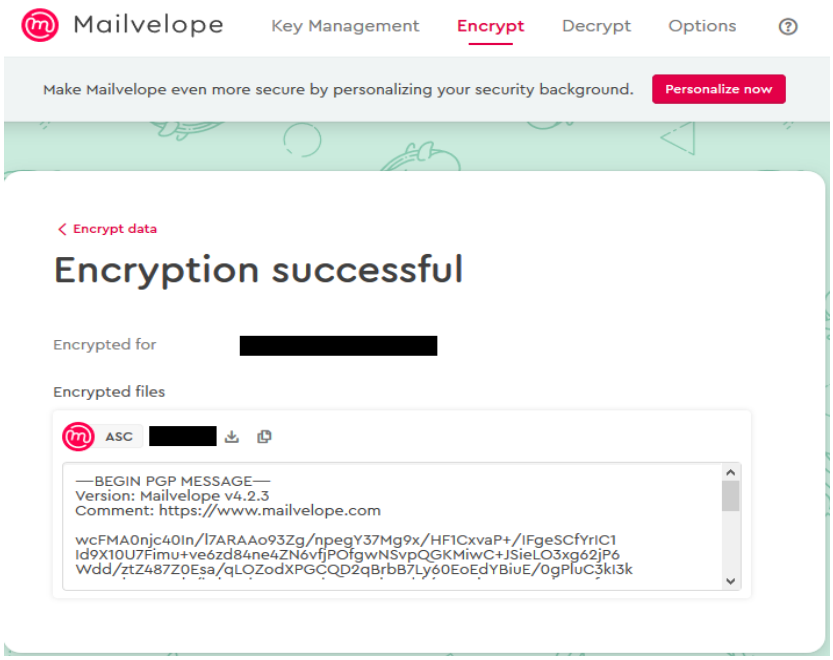
• Important Notes:
• Generate Keys



Prep. Mailvelope (2)

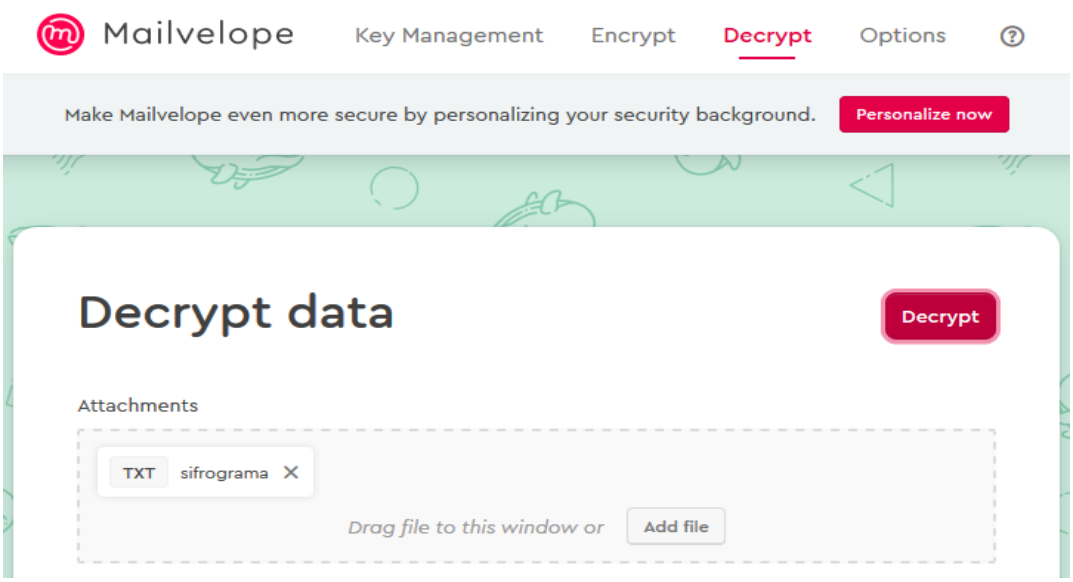
- Encrypt data (and sign it with Your Private Key)

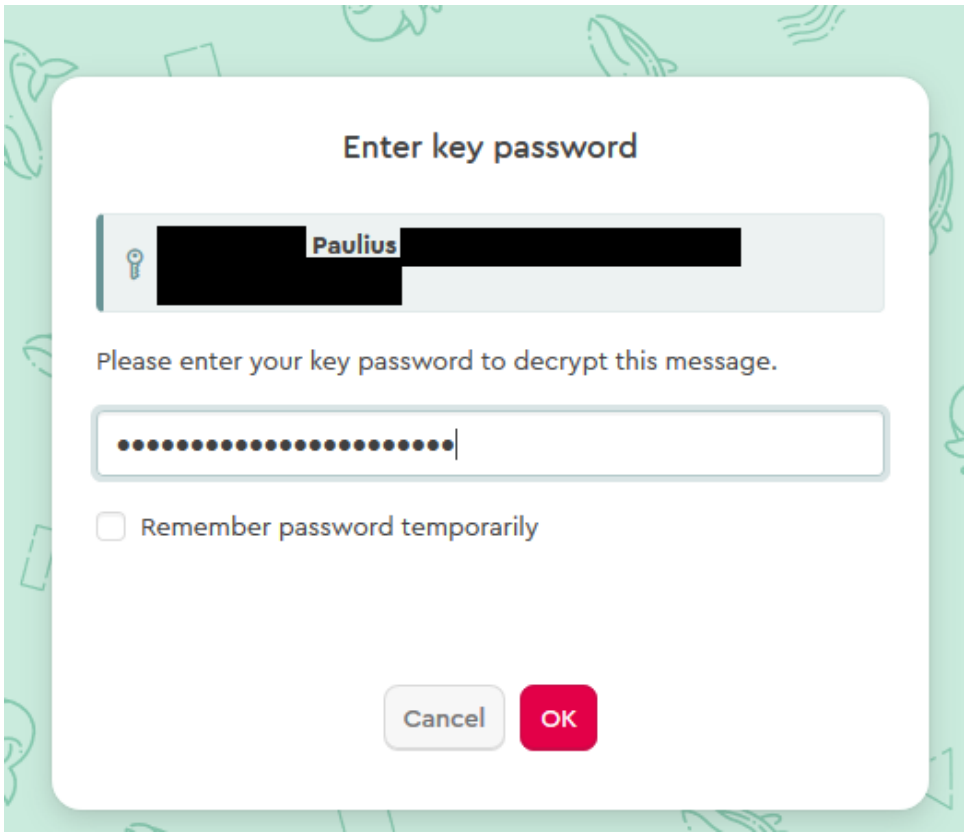




Prep. Mailvelope (3)

- Decrypt data (decryption requires Your Private Key)





P2 Public key Cryptography, PGP

• Task:

1. choose Only One Task out of Two: (i.) the Rabin Algorithm, (ii.) the PGP Application.

1.1. with the Rabin Algorithm do the following:

1.2. choose a Message and make transformation using selected algorithm from first practice work.

1.3. calculate Your Public and Secret Key Pairs.

1.4. encrypt transformed Text with Rabin algorithm from the perspective of Person A.

1.5. decrypt Ciphertext with Rabin algorithm from the perspective of Person B.

1.6. detransform Text to it's original form.

1.7. fill up Your Results.

2.1. for the PGP Application do the following:

2.2. create Your own Key Pair.

2.3. share Public Key with Communication Subject.

2.4. encrypt using Public Key of the Subject and between each other securely exchange encrypted Message(s).

2.5. decrypt encrypted messages with Your Private Key.

2.6. fill up Your Results.

• Ref.

Outstanding — successfully finishing Tasks No. 1.1.-1.7. or Tasks No. 2.1.-2.6.

Typical — only Single Encryption or Single Decryption Process completed for an given Algorithm.

Threshold — Results providen without Explanation and (or) with only Pictures.

01/09/2023

Check-digit security, Hash App [3/13]p

Prep. Check-digit security

Various important Line of the Code Data could be secured with Check-digit Security Feature. Check-digit allows to detect Errors, if the Line of the Code Data were incorrectly provided to the Application or to the Information System. It prevents Erroneous Data to be Submited in the First Place.

ISO/IEC 7064 International Standard gives Guidelines for Security Techniques like Check-digit Security. For stronger Security let us do implementation of 97-10 MOD Check Character System with Two digit Numbers instead of single Check-digit.

For succesful 97-10 MOD implementation the choice of Polynomial Method is expected r^(i-1)*(mod M). To make it in working Condition in each separate Row let us keep the Data about:

1. Character Positions (according each VIN Code Characters);
2. Weights;
3. Original Characters (VIN Code Characters);
4. Products - Multiplications;
5. Product Sums;
6. Obtaining Check-digit.

Prep. VIN code (1)

Every Vehicle has a unique identifier code called a VIN. This number contains vital information about the Vehicle, such as its manufacturer, year of production, the plant it was produced in, type of engine, model and more.
For instance, if someone wants to buy a Vehicle, it is possible to check the VIN number in online database to be more informed that the Vehicle was not stolen, damaged or illegally modified.

The VIN number has a specific format that is pretty widely recognized (ISO 3779:2009 “Road vehicles — Vehicle identification number (VIN) — Content and structure” applies to motor vehicles, towed vehicles, motorcycles and mopeds as defined in ISO 3833.
ISO 3779:2009 requires ISO 3780:2009 Road vehicles — World manufacturer identifier (WMI) code).
Every Vehicle manufacturer is obliged to mark all its Vehicles in widely recognised special format like one provided above.

Prep. VIN code (2)

• According Standard only Arabic numerals and Roman letters, excluding three characters I O Q, has to be used.

• (97-10 MOD) Check-digit is located in the End of the Code (in this case VIN) Data.

• Informative Annexes in the Standard shows Total Number of VIN Characters is 17 (Seventeen) and provide VIN Sample: 1CP H423GA 4G102745

Obtaining Check-digit:

Original VIN Code (LTS ISO 3779:2010): 1CPH423GA4G102745

19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	Position
89	38	62	45	53	15	50	05	49	34	81	76	27	90	09	30	03	10	01	Weight
1	C	P	H	4	2	3	G	A	4	G	1	0	2	7	4	5			
01	12	25	17	04	02	03	16	10	04	16	01	00	02	07	04	05	00	00	Characters
89	456	1550	765	212	30	150	80	490	136	1296	76	00	180	63	120	15	00	00	Product

5708 Sum of Products

5708 MOD 97 = 82 Check-digit Numbers

It means that Final VIN Code (97-10 MOD) - 1CPH423GA4G10274582

Check-digit - 82

Testing the Check-digit:

During VIN Code processing it's necessary to conduct the same calculations as above with small exception in Beginning. We start with nullified VIN Code Places in Both Positions.

After completion of calculations the initial Check-digit Numbers has to be compared with calculated Check-digit Numbers. If these Numbers are the same, then VIN Code is Correct and could be processed further.

Prep. Argon2 Algorithm

Choose Technology:

• JavaScript and TypeScript with a Node.js implementation, which is Available over the Internet URL(s):
https://github.com/ranisalt/node-argon2
https://npm.runkit.com/argon2

• Perl implementation:
https://metacpan.org/pod/PHP::Functions::Password

• PHP implementation:
https://www.php.net/manual/en/function.password-hash

• Python implementation:
https://pypi.org/project/argon2/

• Standalone argon2 library Application over Debian-based Linux System is Available via Command Line:
sudo apt-get install argon2

P3 check-digit security, Hash App

• Task:

1. choose Only One Task out of Two: (i.) Invent the Check-digit for securing VIN Code or (ii.) try out the Hash Application Argon2.

1.1. for inventing the Check-digit security do the following:

1.2. use ISO/IEC 7064:2003 Standard and define Character Positions, Weights, VIN Code Characters, Multiplications and Sums, the Check-digit and Check-digit calculation.

1.3. prove that Your Invention is Correct.

1.4. fill up the Results.

2.1. for trying out the Hash Application Argon2 do the following:

2.2. install and (or) use the App, and explain how the Argon2 could be used in securing IT Assets.

2.3. list out the Features, which makes Argon2 a Modern Algorithm(s).

2.4. fill up the Results.

• Ref.

Outstanding — successfully finishing Tasks No. 1.1.-1.4. or Tasks No. 2.1.-2.4.
Typical — for completing the first part.
Threshold — if Results Documentation is Off.

02/09/2024

Web security testing [4/13]p

Web security

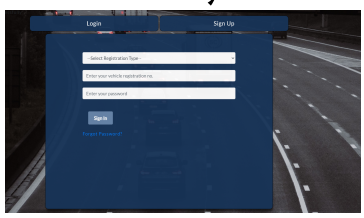
• Purpose - You have to practice to track and detect all security features of a web system.

• Web Security is very important. Small and Medium Enterprises and Organisations own(s) web Site's. Web Site's are great help in fulfilling their purpose and making business as usual.

• You gonna use custom web system for this practice work - Accident Reporting System.

• This system is Available online. Link to the System:

https://studies.issauga.lt/Login.php



• You have all opportunities to try it out on Your Hardware. Link to the Github Repository:

https://github.com/PolVilniusTech/Accident-Reporting-System

Available Log In Data for testing it out:

User type - Username - Password
Driver - H125111 - qwerty
Insurance Company - 24555 - nothingma77er5

P4 Web security

• Task:

1. In available & online Web Site https://studies.issauga.lt/Login.php (Permission Granted for the Site above) You have to track and detect important functionality for Web security, i.e. for protecting SQL-type Database use Prepared Statements (Review of Source Code).
2. Make identification of security Measures and describe how they prevent negative impact from perspective of information security and cryptography, i.e. Prepared Statements protect from „SQL Injection“ attacks.
3. Fill up the Results.

• Useful Guide for the Web Testing to ensure Safe and Secure Life-Cycle of the Information System: https://owasp.org/www-project-web-security-testing-guide/stable/

• Ref.

Outstanding — successfully finishing Tasks No. 1-3. Provided knowledge is Exceptional.
Typical — completing all Tasks and provided advanced Knowledge.
Threshold — completing all Tasks and provided basic Knowledge.

01/09/2023

Caesar and Vigenère [5/13]l

Technological instruments

Choose any of them:

[Interactive]
- https://www.dcode.fr/caesar-cipher
- https://cryptii.com/pipes/caesar-cipher

[Interactive]
- https://www.dcode.fr/vigenere-cipher
- https://cryptii.com/pipes/vigenere-cipher

[Interactive/Install]
- https://gchq.github.io/CyberChef/
- https://github.com/gchq/CyberChef

L5 Caesar and Vigenère cipher

• Task:

1. Take Random Plaintext which would include Name “VGTU”. Take Secret Key: “VILNIUSTECH”. Choose one from provided Variable Sets:
ABCDEFGHIJKLMNOPQRSTUVWXYZ
AĄBCČDEĘĖFGHIĮYJKLMNOPRSŠTUŲŪVZŽ
AĄBCČDEĘĖFGHIĮYJKLMNOPQRSŠTUŲŪVWXZŽ
2. Encrypt Plaintext with Caesar and Vigenère ciphers.
3. Decrypt Ciphertext which You got in 2nd Task.
4. Fill up Your Results.

• (Optional) Task(s):

I. Build Your own Custom Caesar and (or) Vigenère Instrument(s).

• Ref.

Outstanding — successfully finishing Tasks No. 1-4.
Typical — only Single Encryption or Decryption Process completed for given Cipher Algorithms.
Threshold — only Single Cipher Algorithm is used.

01/09/2023

Brute-force attacks [6/13]l

Prep. Testing env. (1)

01. Get or Download Elementary OS from https://elementary.io
02. Verify download by calculating Checksum, if Available
03. Launch Oracle VM VirtualBox pre-installed:
03.01 Click New
03.02 Type desired Name and choose Linux, Ubuntu (64-bit)
03.03 at least 4096 MB Memory Size
03.04 Create a Virtual Hard Disk now
03.05 VirtualBox Disk Image
03.06 Dynamically allocated
03.07 At least 32.00 GB File Size
03.08 At Storage attach Optical Drive with ISO File of Elementary OS
03.09 Click Start
04. Launch Your named VM:
04.01 English
04.02 United States
04.03 English (US)
04.04 Default

Prep. Testing env. (2)

04.05 Erase Disk and Install Elementary OS 7.1 Horus
04.06 Next
04.07 Select a Drive
04.08 Installation without Drive encryption
05. After the Installation launch the VM:
05.01 English
05.02 United States
05.03 English (US)
05.04 Default
05.05 Type Credentials:
ete_adm
eteadm
vilniustech2023pass!@
05.06 Login with these Credentials and choose System Settings
05.07 User Accounts and Unlock
05.08 Create (Standard) User Account with help of Plus Sign
Standard User
ete_usr
eteusr
ete2023pass!

Prep. Testing env. (3)

05.09 Choose ete_usr Account and make it Log in automatically.
05.10 Reboot the System
06. Installing and Configuring HTTP Server from Terminal Window:
06.01 Get Privileges of the Admin Account -
su eteadm
06.02 Update informational Data
sudo apt-get update
06.03 Install HTTP Server -
sudo apt install apache2 apache2-doc apache2-utils lynx lynx-common -y
06.04 Activate apache2 service -
sudo service apache2 start
06.05 Check status of the apache2 service -
service apache2 status
06.06 Not Necessary.
06.07 Not Necessary.

Prep. Testing env. (4)

06.08 Not Necessary.

06.09 Stop HTTP Server -

sudo systemctl stop apache2

06.10 Not Necessary.

06.11 Edit /etc/apache2/apache2.conf and create Folder for Basic Authentication -

“

<Directory /var/www/>
AuthType Basic
AuthName “Draudžiama”
Options SymLinksIfOwnerMatch
AllowOverride None
AuthBasicProvider file
AuthUserFile /etc/apache2/.htpasswd
Require valid-user
</Directory>

“

06.12 Add new User with htpasswd Command -

htpasswd -c /etc/apache2/.htpasswd Student

Prep. Testing env. (5)

06.13 Check Apache2 Syntax and start the Server -
/usr/sbin/apachectl -t
06.14 Start the Server -
sudo systemctl start apache2
07. With Browser test out these Cases:
(a) try access the Folder, which is protected by Basic Authentication and enter the Username and Password
(b) try access the Folder directly via URL in the following schema -
http://username:password@hostname/
08. For Security Assessment of the Basic Authentication -
08.01 Python Script, which repeatedly tries to connect into the Site is Available
08.02 You may build Your own Brute-force Instrument
08.03 You may use Publicly Available Instruments, which usage is not Forbidden by the Jurisdiction

L6 Brute-force attacks

• Task:

1. Deploy Your Authentication Mechanism.
2. Test out Brute-force Attack on Your Authentication Mechanism.
3. Fill up the Results.

• Ref.

Outstanding — successfully finishing Tasks No. 1-3.
Typical — successfully finishing Task No. 1 and Task No. 3.
Threshold — in the Results are not providen any Conclusions about Security Options of the Authentication, which could Stop the Brute-force attack.

02/09/2024

RSA algorithm usage (Simplified) [7/13]l

Prep. Number formats (1)

Letter C in ASCII table would have code of 67 (decimal)

Decimal to decimal conversion:

67 = 6 * (10^1) + 7 * (10^0)

Decimal to hexdecimal conversion:

67 dividing by 16 would get us the quotient 4 and remainder 3;
4 dividing by 16 is not needed, because it's first number of the hexdecimal representation of the selected numerical format.
Hexdecimal representation of decimal 67 is 0x43.

Prep. Number formats (2)

Decimal to binary conversion in similar fashion:

67 dividing by 2 — the quotient 33 and the remainder 1;

33 dividing by 2 — the quotient 16 and the remainder 1;

16 dividing by 2 — the quotient 8 and the remainder 0;

8 dividing by 2 — the quotient 4 and the remainder 0;

4 dividing by 2 — the quotient 2 and the remainder 0;

2 dividing by 2 — the quotient 1 and the remainder 0;

1 dividing by 2 is not needed, because it's first number of the binary representation of the selected numerical format.

Binary representation of decimal 67 is 1000011.

Prep. RSA algorithm (1)

The RSA Algorithm belongs to the Public Key Cryptography Algorithms. It means that Person gonna have pair of Keys from which one is Public and other one - Private or Secret.

1. In the Beginning it is Required to choose two Prime Keys (p) (q)
2. Multiplication Result of these Two Primes will be included in both Public and Private Key's (n=p*q)
3. To find out the Rest of the Public and Private Key Parts we have to find ph Value, which is equal to (ph = (p-1)*(q-1))
4. After completing Equation and finding out ph Value we have to choose Key Part of the Public Key called as "encryption" (e). This Value e could be chosen in random Way and would be equal to the Value, which is between Value One and Value ph. Result of the Greatest Common Divisor between the e and ph should be - One.
5. Key Part of the Private Key called as "decryption" (d) could be found from the Equation of (e * d (mod ph) = 1)

It's important to use the Modular Arithmetic.

Prep. RSA algorithm (2)

6. In this Step we should have two Keys: Public (e, n) and Private (d, n). This means that with Public Key anyone could encrypt Message(s) and afterwards - give Message(s) to You. Private Key, which You have to keep in Secret, is used to decrypt the Message(s).
7. If someone want to encrypt something to You, they use Equation (c(i) = m(i)^e*(mod n)), where i - Character's Position in the Ciphertext or the Message.
8. If You want to decrypt Message(s), then You use Equation (m(i) = c(i)^d*(mod n)), where i - Character's Position in the Message or the Ciphertext.

In Computer Science it is important to understand - Data Encoding.

The RSA Algorithm could be extended for allowing Message(s) signing.

In the Beginning let us practice RSA Algorithm and after that let's dig into Data Encoding and Message signing.

RSA with smaller Requirements to the Key

Sample

p = 7 q = 13

n = (p * q) = (7 * 13)

n = 91

ph = (p-1)(q-1) = (6 * 12)

ph = 72

e = 1<e<ph gcd(e, ph) = 1

e = 5

e * d (mod ph) = 1 5d (mod 72) = 1

d = ([try]*ph+1)/e

d = 29

mod — Modular Arithmetic

/ — Division Operation, [try] — Positive Integer

Public Key (5, 91) Private Key (29, 91)

Current Plaintext = “algorithm”

a |l |g |o |r |i |t |h | m

RSA accepts Numbers, so change the Values in numerical Form:

m = [01|12|07|15|18|09|20|08|13]

Encryption: c(i) = m(i)^e*(mod n)

When e=5:

(1)^5 (12)^5 (7)^5 (15)^5 (18)^5 (09)^5 (20)^5 (8)^5 (13)^5 mod 91

→ → →

01 38 63 71 44 81 76 08 13

c = [01|38|63|71|44|81|76|08|13]
i represent a Position of the Value in Ciphertext

Decryption: m(i) = c(i)^d*(mod n)
When d=29

(1)^29 (38)^29 (63)^29 (71)^29 (44)^29 (81)^29 (76)^29 (8)^29 (13)^29

→ → →

01 12 7 15 18 09 20 08 13

Prep. Message signing

To get Message signing Functionality the RSA algorithm Process (see Prep. RSA algorithm (1) and (2)) has to be extended.

In this case Person, who want to sign the Message, has to calculate Hash of the Message (m) using a hash function (h) and then to encrypt it with his Private Key (d, n). The Signature (s = h(m)^d mod n) has to be sended alongside the Ciphertext (c) to the somebody.

This somebody will be able to check the Authenticity of the Message (m), if he could decrypt the Ciphertext (c) with his own Private Key (Not the Person's Key). Afterwards the Message would go through the same hashing process to obtain (h(m)).

Then this somebody takes (s) Value and decrypts it using Person's Public Key (e, n) and get the Message Hash (H(m) = s^e mod n), which has to be compared with obtained (h(m)) Value.

If H(m) is equal to h(m), then the Signature is Good.
If H(m) is not equal to h(m), then the Signature is Wrong.

Prep. Available Web App

You have chance to practice and try out RSA web App. Link to RSA web App:

https://info.issauga.lt/RSA-algorithm/matr.ciphe.html#Three

Any critic/critique about RSA web App functionality is appreciated. Assessment of the Web Browser cappabilities could be taken as Research Essay.

L7 RSA algorithm usage (Simplified)

• Task:

1. Create Entity A and Entity B Key pairs.
2. Encrypt Message for Entity B (RSA).
3. Sign Message with Entity A Private Key (RSA).
4. Send data to Entity B.
5. Entity B receives the data.
6. Decrypt Message for Entity B (RSA).
7. Check the Signature with Entity A Public Key (RSA).
8. Fill up Your Results.

• Ref.

Outstanding — successfully finishing Tasks No. 1-8.
Typical — finishing all Tasks with less Mistakes.
Threshold — successfully finishing all Tasks with Mistakes.

02/09/2024

Hash function password security [8/13]l

Prep. Commands

• Purpose - to know ways how to ensure data Integrity.

• You may generate a Number of one-way Functions with Help of OpenSSL Library.
• Available openssl algorithms -
openssl list -digest-commands

• Sample of calculating Default Digest Hash Value using the File Format and OpenSSL Library:
cd /tmp
touch file && echo “Lorem Ipsum” > file
openssl dgst file

• Sample of storing Output of the Default Digest Hash Value of the File Data into Different File:
openssl dgst file > checksum

Prep. Utilities for Checksum

• Try out core utilities style for checking checksums of a single file with design (checksum (*)file):

1: choose file
2.1.1: md5sum this file
2.1.2: copy received contents into integrity_checks_new_file.txt
2.2: md5sum file > integrity_checks_new_file.txt
2.3: openssl dgst -md5 -r -hex file > integrity_checks_new_file.txt
3.1: md5sum -c integrity_checks_new_file.txt
Response “OK” approves the integrity of the given file.
Response “FAILED” deny an alignment of the file content with given code.

• Try out BSD style checksums of the single file with design (ALGORITHM (file) = checksum):

1: choose file
2.1.1: md5sum --tag this file
2.1.2: copy received contents into integrity_checks_new_file.txt
2.2: md5sum --tag this file > integrity_checks_new_file.txt
2.3: openssl dgst -md5 -hex file > integrity_checks_new_file.txt
3.1-3.2: md5sum -c integrity_checks_new_file.txt
Response “OK” approves the integrity of the given file.
Response “FAILED” deny an alignment of the file content with given code.

Prep. Utilities for Checksum

• Try out checksums of separate files in a Folder with design (checksum paths + file):

1:create Shell Script " script.sh" (Linux) and place it to directory for checks:

```
#!/bin/bash
stty -echo

PATHWAY="$(pwd)"
ESCPATHWAY="$(pwd | sed "s/\//\\\\\\//g")"
ARRAY="$(find "${PATHWAY}" -type f -exec
md5sum {} \; | sed "s/$ESCPATHWAY/./g")"
for entry in "${ARRAY}"

do

        printf "${entry}\n"

done
```

2: /full/path/to/script.sh > integrity_checks_new_file.txt
3: md5sum -c integrity_checks_new_file.txt
Response “OK” approves the integrity of the given file.
Response “FAILED” deny an alignment of the file content with given code.

L8 Hash function password security

• Task:

VIII-1. Check if " Santrauka" checksums of the " Eve 4.99" App is correct (Link: https://info.issauga.lt/).

VIII-2. Try out various (one-way) Algorithms and (or) Functions on specific Data and compare an Outputs.

VIII-3. Try out various Algorithms for Password Security and compare an Outputs.

VIII-4. Fill out Your Results.

• Ref.

Outstanding — successfully finishing Tasks from No. VIII-1 to No. VIII-4.

Typical — completing all Tasks and provided advanced Knowledge.

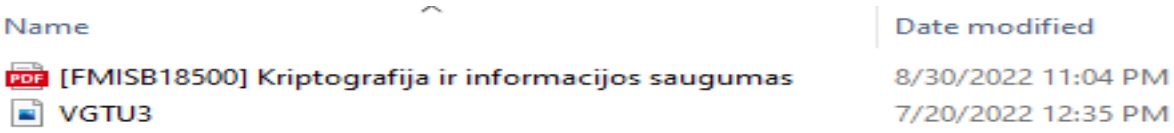
Threshold — completing all Tasks and provided basic Knowledge.

01/09/2023

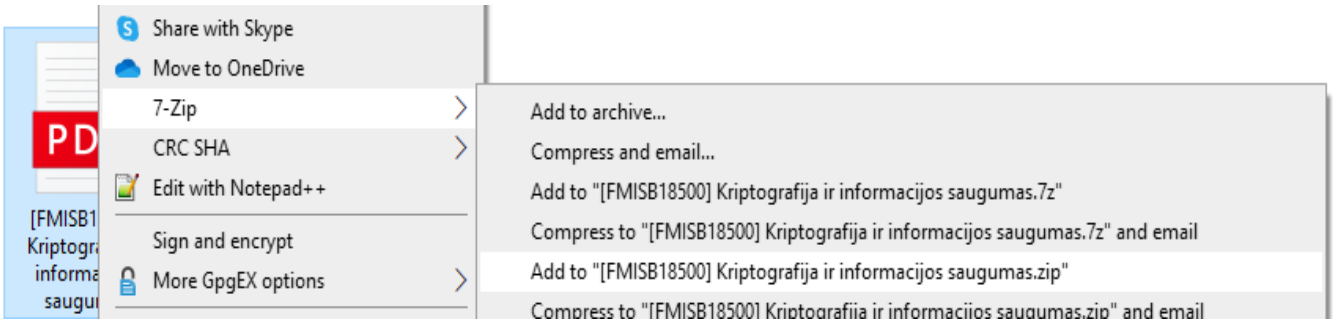
Steganography
[9/13]l

Prep. Compress

Take out some Data and place this Data into the File.
Find some interesting Picture.



Compress this File.



Open simple Command Prompt / Terminal and do following:

- move active Directory to the Place where compressed File is Located;
- type following Instructions:

1. for Microsoft:

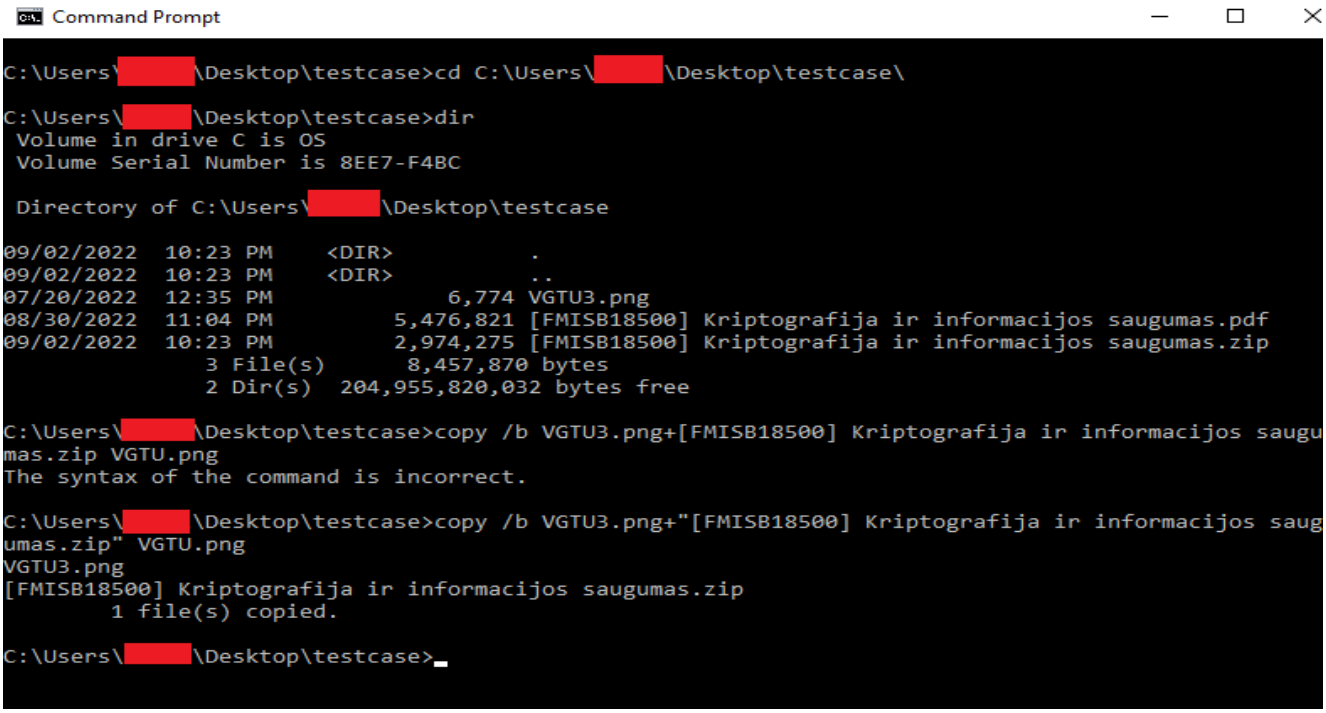
copy /b FILE.PICTURE_EXT+FILE.ARCHIVE_EXT NEW_FILE.PICTURE_EXT

2. for Linux:

cat FILE.PICTURE_EXT FILE.ARCHIVE_EXT > NEW_FILE.PICTURE_EXT

Prep. Windows (1)

Sample from Command Prompt:

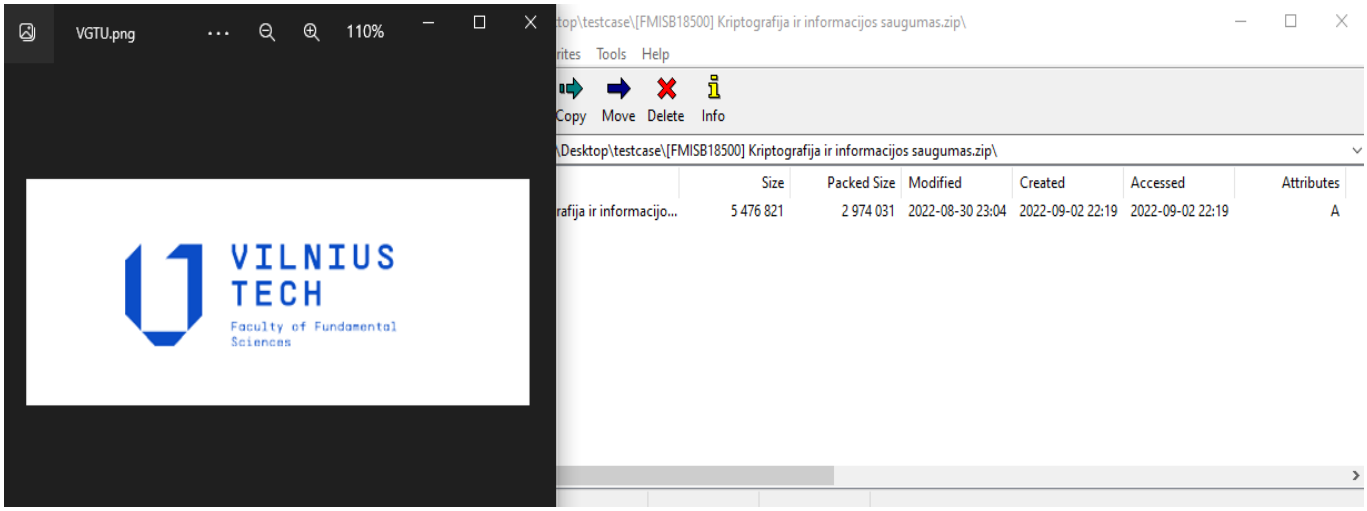


List of Files in the Directory and Metadata, which include Name, Date, Type, Size and Tags:

Name	Date	Type	Size	Tags
[FMISB18500] Kriptografija ir informacijos saugumas	8/30/2022 11:04 PM	Microsoft Edge PDF Document	5,349 KB	
[FMISB18500] Kriptografija ir informacijos saugumas	9/2/2022 10:23 PM	Compressed (zipped) Folder	2,905 KB	
VGTU	9/2/2022 10:31 PM	PNG File	2,912 KB	
VGTU3	7/20/2022 12:35 PM	PNG File	7 KB	

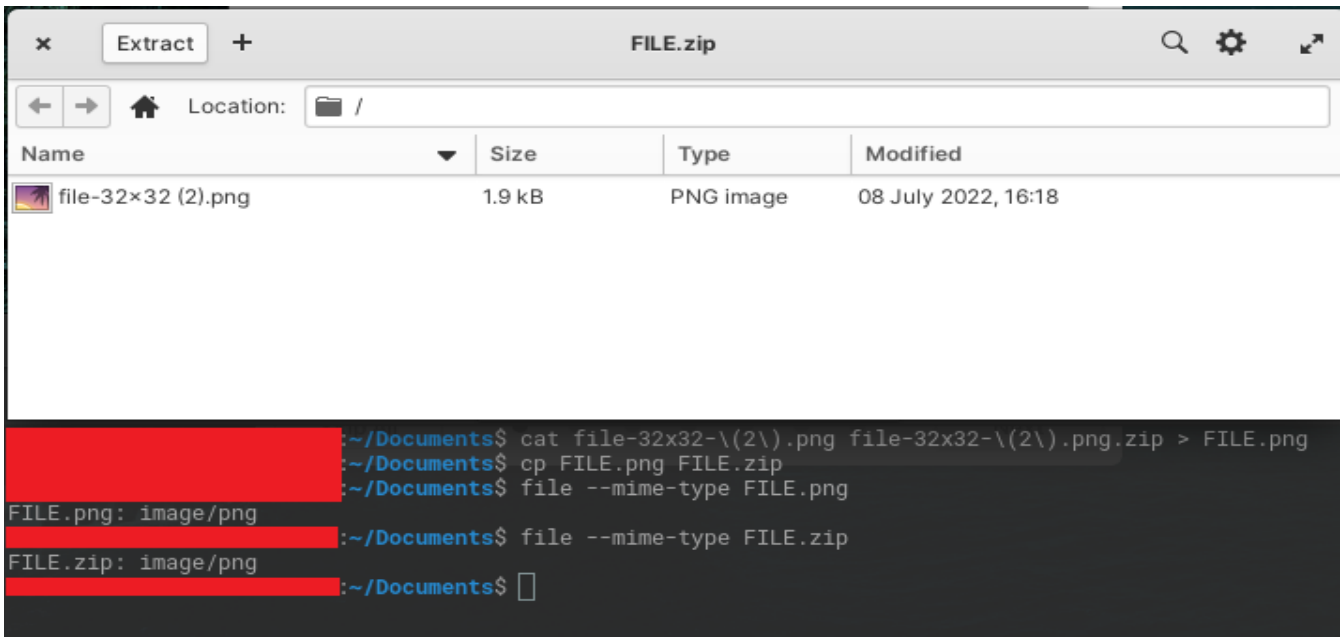
Prep. Windows (2)

Final Result: Picture File which could be viewed by Picture Viewing Software and opened by Compression Instruments.

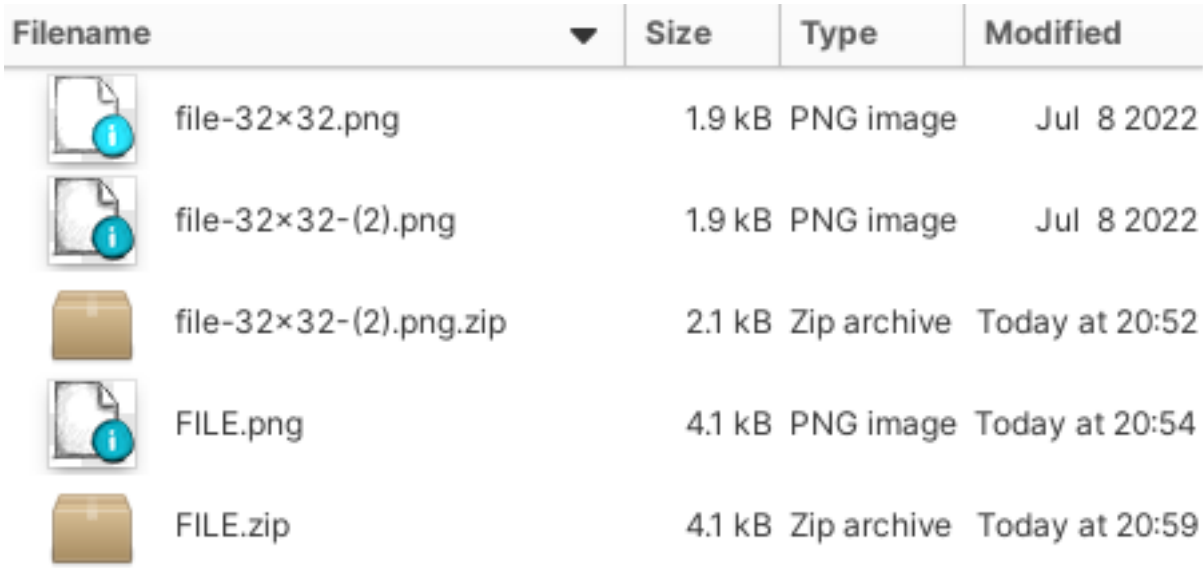


Prep. Linux

Following Two Figures show inconsistency between Mime Type and the Metadata (FILE.png and FILE.zip):



List of Files in the Directory and Metadata of Filename, Size, Type, Modified Date and Time:



L9 Steganography

• Task:

IX-1 Hide the File in the selected File System.
IX-2 Find this hidden File and it's Contents.
IX-3 Fill up the Results.

• (Optional) Task(s):

Ans. Question:
Into which Science Section You would place both Cryptography and Steganography? Provide Explanation of Your Choice.

• Ref.

Outstanding — successfully finishing Tasks from No. IX-1 to No. IX-3 & providing Extensive Explanation.
Typical — successfully finishing Tasks from No. IX-1 to No. IX-3 & providing only Average Explanation.
Threshold — successfully finishing Tasks from No. IX-1 to No. IX-3 & providing only Small Amount of the Explanation.

02/09/2024

Bots
[10/13]l

Prep. Introduction to Bots

• Depending by their purpose the Bot(s) could be used for good or evil Purposes.
• Bot — Application, Script or just functionality of a Technology(-ies), which allows to do various semi-automated or automated Tasks.
• The Term of " Bot" is more wide.
• Other Term of " Robot" is used in the World Wide Web, when we talk about some (Internet) Service like Search Engine.
• Search Engine is a Robot, which is more widely known as a Crawler. Ethical Crawler, i.e. functionality which looks for the Links in the HTML and if [allowed] access, scan allowed Internet resource(s). For non-ethical - vica versa could be applied.
• Other Robot Type - Spider. Spider represents functionality for brute-forcing any URL for gaining more Knowledge about the Target Site.
• Additional Bot/Robot Type - Scraper. Scraper usualy scraps the Data from Internet Resources and uses them for their Goals and Aim's. I.e. App which are using Open Data.
• Modern Search Engine has not only Crawler features, but Spider, Scraper too. For this Reason exists common administrative File for controling access to Your Site, which is called the robots.txt

Prep. Introduction to Bots

Sample of a Scraper:
https://www.heliumscraper.com/blog/introducingcommon-crawler/

According British Broadcasting C orporation (BBC) " large language models (LLMs) " learn" by analysing a massive amount of data often sourced online" .

This show that Generative Pretrained Transformer's require permission to the web Sources if they want to collect information. Automation of indexing of Content in the web orginally were created for Search Engine(s).

- Search Engine(s) may present with API which allows to create the robots.txt file for Site Administrators and to do the Site Administration.

Prep. Search Engines

Some well-known Search Engines:

• BAIDU (HQ Capital of the People's Republic of China)

• BING (Jurisdiction HQ Capital of the United States of America)

• GOOGLE (HQ in California of the United States (America))

• NAVER (HQ in Seoul Capital Area of the South Korea)

• YAHOO (HQ in California of the United States (America))

• YANDEX (HQ in Capital of the Russian Federation)

Search Engines use Crawlers for Conducting the Campaigns of Site Content scanning over the World Wide Web.

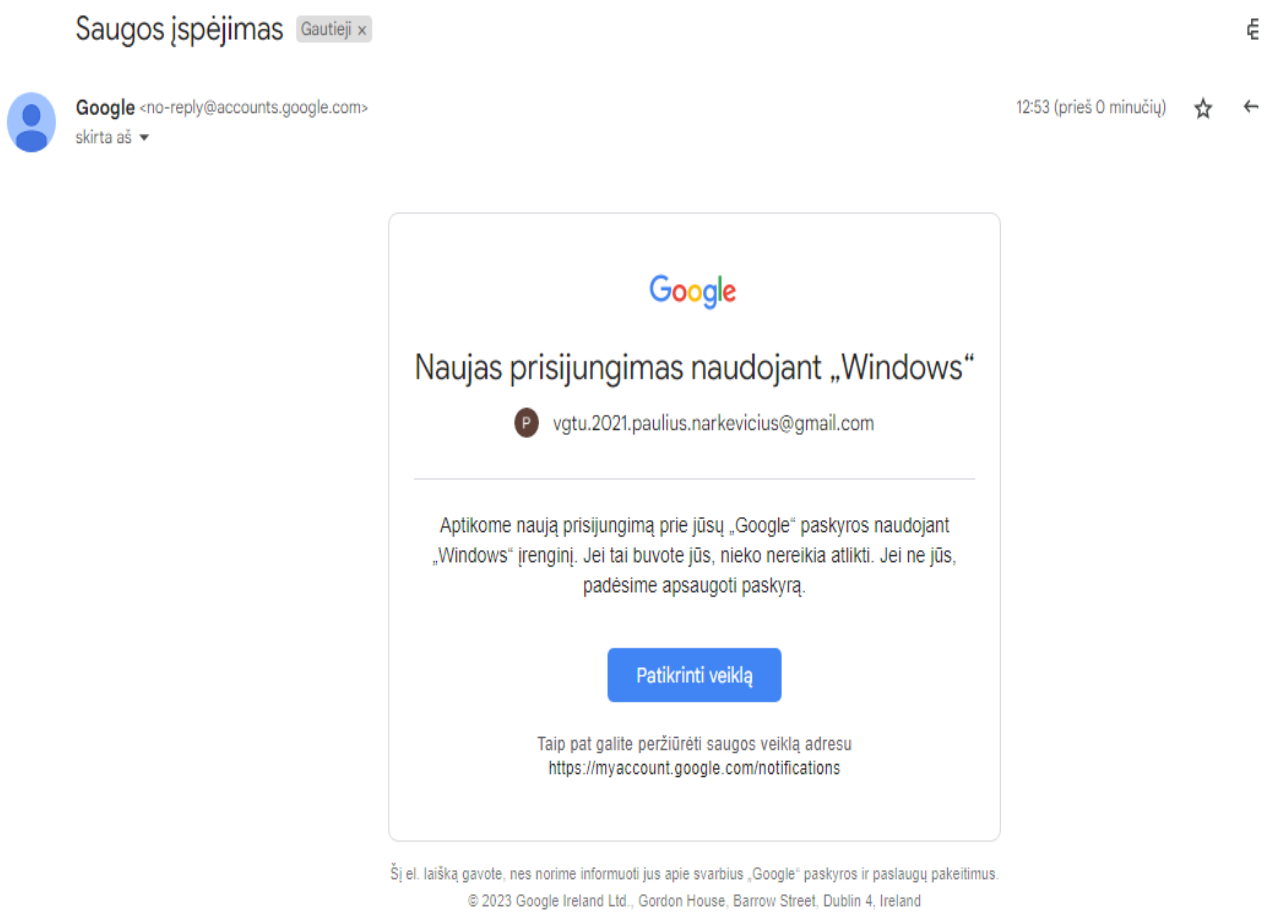
Topics for a Discussion:

• Jurisdiction has a chance to deploy Robots Exclusion Standard to Limit Access for those Bot's;

• Bot's, which does not include Registration Form for Scanning by Request, could include Web Services, which is not meant for Online Searches and is used in Individual-basis.

Prep. Google Services

Google Service, i.e. Gmail, provide Notifications about Security related Events, which are Logged and could be Rewieved at any Time.



Professional Services also could link Site Related Issues, which could be sended into some Type of Dashboard, Posted in a Forum or just formed as E-mail Message and sended it in pre-defined E-mail Address.

Prep. Robots Exclusion Standard (1)

Minimal Functionality includes:

• Defining Access Rules, which include denying, permitting or delaying Access for the Bot to access Document Root and Available Web Resources of the Site (like Files, Folders);

• Defining Bot Names, which are used by Bots in their Web Header, i.e. “User-Agent”;

• Linking Access Rules to the Bots;

• robots.txt is not only place where Robots Exclusion Standard could be used. HTML has specific Tags, which allow to extend the Robot Exclusion from accessing Web Resources.

Prep. Robots Exclusion Standard (2)

Take in consideration that exists Number of Risks, which are related to the Robots Exclusion Standard:

• Un-Ethical Bot don’t take Access Rules Seriously;

• Some Security related Micro-Medium Tools over Internet don’t know how to do Ethical Security Scans and don’t take Access Rules Seriously;

• Lack of Functionality to do administration of the Site, which are linked to the Robots Exclusion Standard.

• Access Rules in the robots.txt is Open to View for Everyone.

L10 Bots

• Task:

X-1. Get used to Robot Exclusion Standard, which gives Rules in robots.txt file for allowing Robot scans of a Web Site Resources.

X-2. Using Eve Application test out Robot Exclusion Standard on a Web Page and try explaining received robots.txt file(s).

X-3. Fill out and explain Your Results.

• (Optional) Task:

Build Your own Ethical Bot and try to list some Risks, which could lead in Un-Ethical use of Your Bot.

• Ref.

Outstanding — successfully finishing the Tasks from No. X-1 to No. X-3.

Typical — successfully finishing the Tasks from No. X-1 to No. X-3. Can't give enough explanation & understanding from the Results of the Eve App.

Threshold — successfully finishing the Task No. X-1 and No. X-3.

01/09/2023

Web page security [11/13]l

Prep. Web Security Threat Management

One Sample of Ethical Web Security Threat Management in Lithuanian Municipality - https://vilnius.lt/lt/vilnius-2in/kibernetinis-saugumas/

- Before beginning is required to Register and Accept the Terms of Service.
- Service has Limitations - Boundaries.
- Each Participant has to Accept Confidentiality Terms. Has his own Rights and Responsibilities.
- Exists Web Form where is possible to Post all Findings.

Prep. Available Resources

Lecturer is providing You with a different Site - https://issauga.lt

Lecturer is giving You all the Code of the Site, which are Available in this Site - https://github.com/PolVilniusTech/bricks

This is enough for conducting White-Box Security Testing for the Available Resource of the issauga.lt

Prep. Instruments

For Manual and Automated Web (Security) Scanning You may use these Scanners:

• Markup Validity of the Web Documents like HTML: https://validator.w3.org

• Internet Link Availability checking for the Web Sites: https://validator.w3.org/checklink

• HTTPS Service Evaluation of the Site and Security Stance of the SSL Configuration: https://www.ssllabs.com/ssltest/

• Mozilla Observatory for numerious Informational and Security Issues: https://observatory.mozilla.org/terms/

• Jurisdictional Law in Republic of Lithuania include Minimal Requirements for e.Services of the Public Sector, which include Requirements to the Software Developers where they should ensure at least Minimal Security Level before giving out their Products to the Client(s). In the Web Space this includes testing out from Web Application Security Risks like: https://owasp.org/www-project-top-ten/

L11 Web page security

• Task:

XI-1. Use Available Online & Internet Scanners.
XI-2. Try out those Scanners. For Ethical Scans Available Resouce - https://issauga.lt
XI-3. Fill up Results and explain Several found Security Issues.

• (Optional) Task(s):

Pre-build existing or write Your own (Information) Security checking Software (App) and try it out on the given Source (XI-2).

• Ref.

Outstanding — successfully finishing Tasks from No. XI-1 to No. XI-3. Explanation given for no less than Several Security Issues.
Typical — successfully finishing Tasks from No. XI-1 to No. XI-2. Explanation given for less than Few Security Issues.
Threshold — successfully finishing Tasks from No. XI-1 to No. XI-2. Explanation given for less than Three Security Issues.

01/09/2023

Vulnerability testing [12/13]l

Prep. Tenable

• Tenable Nessus - Cybersecurity Company Tenable Inc. which most known of the vulnerability scanning software Nessus.
• Tenable has more than Thousand Employees.
• Tenable provide Cross-Platform Cybersecurity related Software Products.

Documentation
https://docs.tenable.com/nessus/Content/GettingStarted.htm

Downloads
https://www.tenable.com/downloads/nessus

• Common Install
• Docker
• Virtual Machine

Trial
https://community.tenable.com/s/trials

Prep. Oracle VirtualBox

Sample:

• Attach host VM for a Network Scan

i. In Oracle Virtual Box
Tools > Create Host-only Network (Ctrl + Shift + C)
Ipv4 address: 192.168.56.1
Ipv4 netmask: 255.255.255.0
Ipv6 address: fe80:68bf:eaac:27d8:22b8
Ipv6 netmask length: 64

ii. In VM's of the Oracle Virtual Box
Settings > Network > 2nd Adapter
Check the Checkbox to activate Adapter
Select Adapter to connect Host Computer Only
"VirtualBox Hoxt-Only Ethernet Adapter" activates automatically
Ok

Prep. Minimum Requirements

For the Linux Type Test Environment(s):

- Linux - Raspberry Pi OS - armhf
  61.3 MB - Raspberry Pi OS (32-bit)

  Site provide Checksums in MD5 and SHA256 algorithms.

  Minimum Req - RPI4 with 8GB RAM.

- Linux - Ubuntu - amd64
  65.2 MB - Ubuntu 14.04 - 20.04 amd64

  Site provide Checksums in MD5 and SHA256 algorithms.

Prep. Debian System

- sudo dpkg -i Nessus-8.15.9-ubuntu1404_amd64.deb
- apt-cache search firefox
- sudo apt-get install firefox firefox-locale-lt
- /bin/systemctl start nessusd.service
- via Browser open something as https://virtualboxd3f775ef:8834 (when using the Oracle VirtualBox)
- Nessus Professional
- Trial Code
- Admin Credentials
- Notification

Prep. Overview

- Settings >

About
Advanced
Proxy Server
Remote Link
SMTP Server
Custom CA
Upgrade Assistant
Password Mgmt
Scanner Health
Debug Logs
Notifications

- Accounts >

My Account
Account Settings
API Keys

Prep. Scans

• My Scans > Create New Scan > Host Discovery (Discovery Section)

• My Scans > Create New Scan > Basic Network Scan (Vulnerabilities Section)

Vulnerabilities

• It is possible to conduct Vulnerability Search, which i.e. include Name, Family, Severity of the Vulnerability, Amount Score of the CVSS. In Auditor Mode there is possible to modify Severity of each Scan Result by increasing or decreasing the Level.
• This Instrument also looks for the INFO Type Results and lists them out, i.e. “Nessus detected 4 installs of OpenSSL”. References in the “See Also” - https://openssl.org

L12 Vulnerability testing

• (Individual Work) Tasks:

XII-1. Register, download and include Tenable Nessus (Trial Version) in a Testing Environment for the Vulnerability Scan.

XII-2. Via Browser Launch the Vulnerability Scan of Your Testing Environment. (if You will use non-latest Computer System Environment beware from the Risks over the Internet)

XII-3. Inspect Locked Functionality of the Tenable Nessus (Trial Version).

XII-4. Fill out a Results.

• (Optional) Task(s):

Build up a List of Vulnerability Scanners and make Assesment. Assess Trustworthiness, Releability and Jurisdictional Acceptance of those Scanners.

• Ref.

Outstanding — successfully finishing Tasks up to No. XII-4.

Typical — successfully finishing Tasks up to No. XII-3.

Threshold — successfully finishing Tasks with a different non-Online only & non-Internet only Scanner.

01/09/2023

Research essay [13/13]r

Limits

• Practical Equations, i.e. Sinuses, Cosinos, Square Root Equations, and calculations gives more Weight to Your Research Essay.
• Practical (and Official) Cryptography Instruments and Knowledge how to use them to Guide yourself through Your Research Essay gives more Weight too.
• It's perfect, if Research Essay could consist for 40-60 or 60-40 Theory-Practice or Practice-Theory.
• Size of Research Essay shall consist from not more than 9 percent size of this Document.
• Word limitations are from 2000 (Two Thousand) to 4000 (Four Thousand) Words for those who are studying in Full-time studies.
• Word limitations are from 1000 (One Thousand) to 2000 (Two Thousand) Words for those who are studying in Part-time distance studies.

R13 Research essay

• Task:

XIII-1. Choose Area of Research Essay.

XIII-2. Provide Reference or Document.

XIII-3. Fill up the results and provide them in Unique Moodle Module Space for the Evaluation.

• Ref.

Outstanding — successfully finishing Tasks up to No. XIII-3.

Typical — successfully finishing Tasks up to No. XIII-3 with no consideration of Limits which were provided in this Document.

Threshold — trying to finishing Tasks up to No. XIII-3 with final practical result — NULL (Academical View).

Consulting

Contacts

Lecturer Paulius Narkevičius
Electronic Mail paulius.narkevicius@vilniustech.lt
Mobile Phone No. +370 684 833 75
Location Saulėtekio al. 11, First Laboratory Corps, Room No. 604
Consultation Hours every Week in Vilnius Tech

This Page was intentionally left Blank.