Information Security Set 1

Emma Seeger, S417302
Muhammed Abdul Majeed Ammeen, S4171861
Ilse Pubben, s3152383

September 2019

Exercise 1

The link to the English acceptable use policy:
`https://www.rug.nl/society-business/centre-for-information-technology/security/aup/?lang=en`

- System managers have the same rights and duties as normal users. However they have the following additional duties:
  1. make sure that the softfware required by the users for their normal work, is provided.
  2. in cooperation with the security manager: installation maintenance of required available software.
  3. maintain the integrity of 'RUGnet' by monitoring in- and outbound traffic
  4. destroy this information after 6 months except if it is required for a legal case
  5. Treat the information as confidential. In special cases (e.g. an account is being abused), systems managers may be ordered to inspect the contents of information stored in, going to or leaving that account.
- The ground rule on which the AUP is based is: That user's may not endanger the system, nor hinder other users.
- Four advices for users of RuGnet are:
  1. Change your password at least once a year
  2. Keep your access information secret
  3. Do not use personal information about yourself, friends nor relatives as password
  4. use a variation of characters for your password (e.g. lower/uppercase, digits, punctuation characters
- Four things that are prohibited according to the AUP:
  1. Violating software / copyright licences that are applicable to software / data on the university computer systems.
  2. Altering identifying data of the university computer systems, like ip-adresses
  3. Harassing or hindering other users of the university computer systems
  4. Modifying or removing hard- or software from the university computer systems without prior consent from proper authorities
- The following sanctions can be applied to you when you violate the rules:
  1. The board of the faculty or department responsible for you is informed of the situation
  2. Your access rights may be restricted or suspended until the investigation is complete
  3. Your data files and media are investigated
- where you can go to, to challenge a sanction that has been applied to you
  You may file an objection to the restrictions or suspension with the chair of your department

Exercise 2

decrypted text

```
Nine common security awareness mistakes (and how to fix them)

To err is human, but to err in cyber security can cause serious and major
damage.

Perfect security may be hard or impossible to achieve, but major improvements
are possible, just by being aware of some of the most common mistakes and their
consequences. By being aware of these mistakes allows you to avoid them.

Here are the nine common security awareness mistakes, don't make 'em yourself:

1. Falling for phishing: One of the most common mistakes.
2. Unauthorized application or cloud use, known as shadow IT.
3. Weak or misused passwords
4. Remote insecurity: This is the common practice of transferring files
   between work and personal computers
5. Disabling security controls: This is usually done by users with
   administrative privileges, to make things easier for employees
6. Clueless social networking
7. Poor mobile security
8. Too many privileges
9. Failure to update or patch software

See also http://www.csoonline.com/article/2877259/security-awareness/
            nine-common-security-awareness-mistakes-and-how-to-fix-them.html
```

source code exercise 2

```python
import argparse
import sys

# Function to encrypt/decrypt an alphabet
def transform(ch, mapping, alpha):
    # return chth elemant of mapping
    i = alpha.find(ch)
    # print("Mapped ", ch, i, mapping[i])
    return mapping[i]

# Function to check if the key provided is valid
def check_mapping(mapping, alpha):
    string = ''.join(sorted(list(mapping)))
    return string == alpha

# Function to encrypt/decrypt a string
# Mapping - key for encryption
# alpha - the set of aplphabets in language
# mode 0 when -o not present , mode 1 when -o present
def string_transform(string, mapping, alpha, mode):
    string1 = ''
    if mode == 1:
    # -o is present
        for ch in string:
            if(ch.isalpha()):
```

```
                # print(ch)
                ech = ch.lower()
                dch = transform(ech, mapping, alpha)
                if ch.isupper():
                    string1 = string1 + dch.upper()
                else:
                    string1 = string1 + dch
            else:
                string1 = string1 + ch

        return string1
    elif mode == 0:
        # -o is absent
        for ch in string:
            if(ch.isalpha()):
                string1 = string1 + transform(ch.lower(), mapping, alpha)
        return string1
    else:
        print("Invalid Mode")
        return ''

def main():

    # checking if all required parameters present
    parser = argparse.ArgumentParser(description="where:")
    parser.add_argument("mapping", help = "26 letter char-mappig \n or an int-value")
    parser.add_argument("-o", help = "keep non-letters as is, honor letter casing", action =
    parser.add_argument("-d", help = "decrypt", action = "store_true")
    try:
        args = parser.parse_args()
    except:
        parser.print_help()
        sys.exit(0)

    # english alphabets
    alpha = 'abcdefghijklmnopqrstuvwxyz'
    # mapping = "yxzwvtsqpnrmlkjgfdchbuoaie"
    # string = "QwertyuioP1234"
    mapping = args.mapping

    # check if key is a valid mapping
    if check_mapping(mapping, alpha) is False:
        print("Key Invalid")
        sys.exit(0)

    # setting mode -o
    if args.o is True:
        mode_o = 1
    else:
        mode_o = 0

    # setting decryption mode
    if args.d is True:
        mode_d = 1
    else:
        mode_d = 0
```

```
    # opening the file to be decrypted
    with open('2019.enc', 'r') as myfile:
        data = myfile.read()

    if mode_d == 0:
        print(string_transform(data, mapping, alpha, mode_o))
    elif mode_d == 1:
        print(string_transform(data, alpha, mapping, mode_o))


if __name__ == "__main__":
    main()
```

Exercise 3

purpose of this exercise: learn to break a simple substitution cipher text

shift y = 19 to solve, originally encrypted with x = 7
Decrypted text

```
welcometothecourseaboutinformationsecuritythiscour
seisaboutsecuringinformationinthiscontextwethinkfo
rexampleabouthowtopreventtheunauthorizedreadingofi
nformationorabouthowtopreventtheunauthorizedmodifi
cationofinformationencryptionisthebasictoolforsecu
ringinformationmanyencryptionmethodsexistsomealrea
dythousandsyearsoldinitiallywellfocusonsimplemetho
dstoencryptinformationfollowingthiswellusecharacte
risticvaluesidentifyinginformationmakingitdifficul
ttomodifyinformationunnotifiedlaterinthiscoursewel
lintroducepersonalencryptionwaystocontrolaccesstoi
nformationandhowtokeepyourprivacywehopeyouwillenjo
ythiscourseaboutinformationsecurity
```

Human readable decrypted text

```
welcome to the course about information security this course
is about securing information in this context we think for example
about how to prevent the unauthorized reading of information or about
how to prevent the unauthorized modification of information encryption
is the basic tool for securing information many encryption methods exist
some already thousands years old initially well focus on simple methods
to encrypt information following this well use characteristic values
identifying information making it difficult to modify information
unnotified later in this course well introduce personal encryption ways
to control access to information and how to keep your privacy we hope you
will enjoy this course about information security
```

To the plain text a shift of `x=7` was applied according `c = (p + x) % 26`
To get the original text back the smallest positive substitution cipher shift is `y=19`.

Exercise 4

1. decrypted text

i came to security from cryptography and thought of the problem in a military like fashion most writings about security come from this perspective and it can be summed up pretty easily security threats are to be avoided using preventive counter measures this is how encryption works the threat is eavesdropping and encryption provides the prophylactic this could all be explained with block diagrams a lice is communicating with bob both are identified by boxes and there is a line between them signifying the communication eve is the eavesdroppers he also is a box and has a dotted line attached to the communications lines he is able to intercept the communication the only way to prevent eve from learning what alice and bob are talking about is through a preventive countermeasure encryption theres no detection theres no response theres no risk management you have to avoid the threat for decades we have used this approach to computer security we draw boxes around the different players and lines between them we define different attackers eavesdroppers impersonators thieves and their capabilities we use preventive countermeasures like encryption and access control to avoid different threats if we can avoid the threats weve won if we cant weve lost imagine my surprise when i learned that the world doesnt work this way some history from the vigenere wikipage the first well documented description of a polyalphabetic cipher was formulated by leon battista alberti around and used a metal cipher disc to switch between cipher alphabets albert is system only switched alphabets after several words and switches were indicated by writing the letter of the corresponding alphabet in the cipher text later in johannestrithemius in his work poligraphi a invented the tabula recta a critical component of the vigenre cipher the trithemius cipher however only provided a progressive rigid and predictable system for switching between cipher alphabets what is now known as the vigenere cipher was originally described by giovan batti stabellaso in his book lacifradelsig giovan battista bellaso he built upon the tabula recta of trithemius but added a repeating counter sign a key to switch cipher alphabets every letter whereas alberti and trithemius used a fixed pattern of substitutions bellasos scheme meant the pattern of substitutions could be easily changed simply by selecting a new key keys were typically single words or short phrases known to both parties in advance or transmitted out of band along with the message bellasos method thus required strong security for only the key as it is relatively easy to secure a short key phrase say by a previous private conversation bell a os system was considerably more secure blaised evigenere published his description of a similar but stronger auto key cipher before the court of henry iii of france in later in the th century the invention of bellasos cipher was misattribut to vigenere david kahn in his book the codebreakers lamented the misattribution by saying that history had ignored this important contribution and instead named a regressiv and elementary cipher for him vigenere though he had nothing to do with it the vigenere cipher gained a reputation for being exceptionally strong noted author and mathematician charles lutwidge dodgson lewis carroll called the vigenere cipher unbreakable in his piece the alphabet cipher in a childrens magazine in scientific american described the vigenre cipher as impossible of translation this reputation was not deserved charles babbage is known to have broken a variant of the cipher as early as however he didnt publish his work kasiski entirely broke the cipher and published the technique in the th century even before this though some skilled cryptanalysts could occasionally break the cipher in the th century cryptographics lide rule use dasa calculation aid by the swiss army between and the vigener ecipher is simple enough to be a field cipher if it is used in conjunction with cipher disks the confederate states of america for example used abrass cipher diskto implement the vigenere cipher during the american civil war the confederacys messages were far from secret and the union regularly cracked their messages through out the war the confederate leadership primarily relied upon three key phrases manchester bluff complete victory and as the war came to a close come retribution gilbert vernam tried to repair the broken cipher creating the vernam vigenere cipher in but nomatter what he did the cipher was still vulnerable to crypt analysis vernams work however eventually led to the one time pada provably unbreakable cipher

2. table of sums of standard deviations of probed key sizes:

```
sum of  5 std. devs:  61.97449880918521
sum of  6 std. devs:  87.21045320359933
sum of  7 std. devs:  63.678583080274834
sum of  8 std. devs:  67.50494266678481
sum of  9 std. devs:  132.13070367198557
sum of  10 std. devs:  66.58228385309064
sum of  11 std. devs:  68.00623336187606
sum of  12 std. devs:  92.55576499798589
sum of  13 std. devs:  69.6527864360467
sum of  14 std. devs:  71.30776962521261
sum of  15 std. devs:  94.74553988067163
```

3. initial suggested keywords:

```
most likely         i   n   t   e   g   r   i   t   y
second              m   r   i   i   v   g   x   p   n
third               x   c   g   a   k   v   m   i   u
```

4. most frequently occurring character in generalized vignere cipher:
   The most frequently occuring character would be the spacebar ' '.