Homework 1

Track A

# **Problem 1**: (10 points) CrypTool

## Download CryptTool (v1.4.42) and install it on your computer.

### Screenshot(s)

A screenshot of a computer

Description automatically generated

## Encrypt the following message using Caesar Cipher (Shift-3,) and submit your ciphertext

The art of war teaches us to rely not on the likelihood of the enemy's not coming,

but on our own readiness to receive him; not on the chance of his not attacking,

but rather on the fact that we have made our position unassailable.

—The Art of War, Sun Tzu

* + Wkh duw ri zdu whdfkhv xv wr uhob qrw rq wkh olnholkrrg ri wkh hqhpb'v qrw frplqj, exw rq rxu rzq uhdglqhvv wr uhfhlyh klp; qrw rq wkh fkdqfh ri klv qrw dwwdfnlqj, exw udwkhu rq wkh idfw wkdw zh kdyh pdgh rxu srvlwlrq xqdvvdlodeoh. —Wkh Duw ri Zdu, Vxq Wcx

### Screenshots

A screenshot of a computer

Description automatically generated

A screenshot of a chat

Description automatically generated

# **Problem 2**: (15 points) List and briefly define the six security services as defined in the OSI security architecture

* Authentication: The ability to confirm the identity of a sender is from its claimed source
  + Peer entity authentication: The ability to verify the identities of connected parties
  + Data origin authentication: The ability to verify the claimed source of received data
* Access Control: The ability to restrict access to resources by defining and enforcing policies to manage user permissions and privileges
* Data Confidentiality: The ability of a system to ensure that transmitted data is protected from passive attacks and viewed only by authorized parties
* Data Integrity: The ability of a system to ensure that data is modified only by authorized parties and are received as sent, with no “duplication, destruction, insertion, modification, reordering, or replays” [1]
* Nonrepudiation: The ability of a system to ensure that neither the sender nor the receiver can deny a message that has been transmitted
* Availability Service: The ability of a system to ensure that data can be accessed by any authorized parties on demand

# **Problem 3**: (15 points) Decrypt the provided ciphertext using CrypTool

Nbkyrpsrws jifx. Jir kjqqbofr mssmne tiarp syrqr nbpntgqsminrq bq syr optsr-cjpnr

mkkpjmny jc spxbih mff kjqqbofr erxq. Bc syr erx qkmnr bq urpx fmphr, sybq ornjgrq

bgkpmnsbnmf. Sytq, syr jkkjiris gtqs prfx ji mi mimfxqbq jc syr nbkyrpsrws bsqrfc,

hrirpmffx mkkfxbih umpbjtq qsmsbqsbnmf srqsq sj bs. Eijvi kfmbisrws. Syr mimfxqs gmx

or mofr sj nmkstpr jir jp gjpr kfmbisrws grqqmhrq mq vrff mq syrbp rinpxksbjiq.

Vbsy sybq eijvfrahr, syr mimfxqs gmx or mofr sj aratnr syr erx ji syr omqbq jc syr

vmx bi vybny syr eijvi kfmbisrws bq spmiqcjpgra. Nyjqri kfmbisrws. Bc syr mimfxqs

bq mofr sj nyjjqr syr grqqmhrq sj rinpxks, syr mimfxqs gmx arfborpmsrfx kbne

kmssrpiq syms nmi or rwkrnsra sj prurmf syr qsptnstpr jc syr erx.

## Plaintext Solution

Ciphertext only. One possible attack under these circumstances is the brute-force approach of trying all possible keys. If the key space is very large, this becomes impractical. Thus, the opponent must rely on an analysis of the ciphertext itself, generally applying various statistical tests to it. Known plaintext. The analyst may be able to capture one or more plaintext messages as well as their encryptions. With this knowledge, the analyst may be able to deduce the key on the basis of the way in which the known plaintext is transformed. Chosen plaintext. If the analyst is able to choose the messages to encrypt, the analyst may deliberately pick patterns that can be expected to reveal the structure of the key.

## Walkthrough

First we begin by analyzing n-grams. By computing and comparing the ciphertext’s Histogram and Diagram frequency, we were able to determine a place to start in manually decrypting this substitution cipher.

A screenshot of a computer

Description automatically generated

As shown in our n-grams, our 3 most frequent characters are ‘R’ (13.82%), ‘S’(11.94%), and ‘M’ (8.36%) and our three most frequent digrams are ‘SY’ (4.73%), 'YR' (4.08%), and 'SR' (2.36%)

If we compare these values to the top n-grams in the English language, as noted by Practical Cryptography [2], we see that the three most frequent characters are ‘E’ (12.1%), ‘T’(8.94%), and ‘A’ (8.55%) and the three most frequent digrams are 'TH' (2.71), 'HE' (2.33), and 'IN' (2.03). These lists are shown below.

A screenshot of a computer screen

Description automatically generated

If we take a closer look at the bigrams in our cipher, we can determine a pattern – Our top three bigrams contain only 3 letters and are in a specific order:

Therefore, to begin out manual analysis, we need to find three common bigrams in the English language that match this pattern. If we start with the most popular bigram in the English language, we can quickly find the remaining bigrams in the list that match this pattern.

A screenshot of a computer

Description automatically generated

|  |  |
| --- | --- |
| Ciphertext | Plaintext |
|  |  |
|  |  |
|  |  |

This also makes sense as our top two most frequent characters ‘R’ and ‘S’, align with the two most frequent characters in the English language are ‘E’ and ‘T’.

If we plug these values into the manual processing screen in CrypTool, we can see that we are most likely correct as many of the three letter words are converted from SYR to THE, which is a valid English stop word.

A screenshot of a computer screen

Description automatically generated

Now that we have a starting point, we can begin to fill in the remaining letters.

Note: From this point on, ciphertext letters will be in lowercase and plaintext will be in uppercase

After plugging in our known letters, we can see the word THEqE. From this, we can deduce that q is most likely an R or S. By plugging each of these into the manual analyzer, we find that q is most likely S. We can come to this conclusion because if we use R, the text TEqTq becomes TERTR which is not and English word. However, if we use S, it becomes TESTS.

Screens screenshot of a computer

Description automatically generated

Now that we have found S, we can see there are multiple locations that suggest b represents I, such as in bS and THbS which become IS and THIS.

A screenshot of a computer

Description automatically generated

Next, we see the word STmTISTInmf which is most likely the word STATISTICAL. Given this, we can plug in A for m, C for n, and L for f.

A screenshot of a computer

Description automatically generated

From here we can begin filling in some of the common stop words, such as vHICH (v becomes W), THEIp (p becomes R), WAx (x becomes Y since S and R have already been found)

A screenshot of a computer screen

Description automatically generated

Next we can see the phrase tiaER THESE CIRCtgSTAiCES, which can be decrypted as ‘UNDER THESE CIRCUMSTANCES.’ With this, we are able to map the following letters t->U, i->N, a->D, and g->M

A screenshot of a computer screen

Description automatically generated

Next we see ENCRYkTIjNS, which we can deduce is ENCRYPTION. Given this, the following mappings take place: k->P and j->O

A screenshot of a computer screen

Description automatically generated

Now that we have a majority of the letters decrypted, we can begin doing longer phrases at a time. For example, oRUTEcORCE APPROACH Oc TRYINh ALL POSSIoLE eEYS becomes BRUTEFORCE APPROACH OF TRYING ALL POSSIBLE KEYS, such that o->B, c->F, h->G, e->K

A screenshot of a computer

Description automatically generated

This leaves the final letters that appear in the cipher, w and v. From the words CIPHERTEwT and PLAINTEwT, we can deduce that w becomes X and from the word uARIOUS we can deduce that u becomes V.

A screenshot of a computer

Description automatically generated

This concludes our manual analysis and produces the final plainext:

Ciphertext only. One possible attack under these circumstances is the brute-force approach of trying all possible keys. If the key space is very large, this becomes impractical. Thus, the opponent must rely on an analysis of the ciphertext itself, generally applying various statistical tests to it. Known plaintext. The analyst may be able to capture one or more plaintext messages as well as their encryptions. With this knowledge, the analyst may be able to deduce the key on the basis of the way in which the known plaintext is transformed. Chosen plaintext. If the analyst is able to choose the messages to encrypt, the analyst may deliberately pick patterns that can be expected to reveal the structure of the key.

# **Problem 4**: (15 points) Answer the questions below for a 5x5 matrix for the Playfair cipher

## Calculate the possible keys the Playfair cipher can have (ignore identical encryption results). Express your answer as an approximate power of 2.

In a Playfair Cipher key, each letter of the alphabet, excluding 'J' which is usually combined with 'I', is placed in the matrix exactly once. As such, the first letter can be chosen from 25 possibilities, the second letter from 24, etc. As a result, the total number of unique keys that can be generated for the cipher is or . The steps to convert is to are shown below.

|  |  |
| --- | --- |
|  | *logarithmic function* |
|  | *cipher possibilities as a factorial* |
|  | *substitute values* |
|  | *convert to exponent form* |

## Consider identical encryption results. How many effectively unique keys does the Playfair cipher have?

In the Playfair Cipher, certain keys produce identical encryption results due to the symmetry of the matrix. In other words, shifting rows or columns does not change the encryption result because the relative letter positions, and thus, the letter pairs, within the matrix remain constant. For example, the following row shifts would produce the same result:

A close-up of a chart

Description automatically generated

For each of these 5 row shifts, there are 5 equivalent column shifts. For example, the matrix at position can undergo the following row shifts:

A close-up of a chart

Description automatically generated

Since there are 5 rows and 5 columns in a 5x5 matrix, there are equivalent matrices. Therefore, we can solve for the number of effectively unique keys () that the Playfair Cipher has by doing the following:

|  |  |
| --- | --- |
|  | *formula* |
|  | *substitute values* |
|  | *cancel like terms* |
|  | *solve* |

As such, there are , or , effectively unique keys in the Playfair cipher.

# **Problem 5**: (15 points) PT-109 Message Decryption

When the PT-109 American patrol boat, commanded by Lieutenant John F. Kennedy, was sunk by a Japanese destroyer, an encrypted message was received at an Australian wireless station in Playfair code. The message was encrypted using the key *royal new zealand navy*.

KXJEY UREBE ZWEHE WRYTU HEYFS

KREHE GOYFI WTTTU OLKSY CAJPO

BOTEI ZONTX BYBNT GONEY CUZWR

GDSON SXBOU YWRHE BAAHY USEDQ

## Decrypt the message using Cryptool (*remember to translate TT into tt*)

After decrypting the Playfair ciphertext using the key '*royal new zealand navy*' in CrypTool, the resulting plaintext was:

“PT BOAT ONE OWE NINE LOST IN ACTION IN BLACKESUSU STRAIT TWO MILES SW MERESU COVE X CREW OF TWELVE X REQUEST ANY INFORMATION X”

A screenshot of a computer

Description automatically generated

However, since the directions state to convert 'TT' to 'tt', we adjusted the highlighted section to read:  
“PT BOAT ONE OWE NINE LOST IN ACTION IN BLACKETT STRAIT TWO MILES SW MERESU COVE X CREW OF TWELVE X REQUEST ANY INFORMATION X”

# **Problem 6**: (15 points) Encrypt the word "explanation" using the key "leg"

pbvwetlxozr

A screenshot of a computer

Description automatically generated

# **Problem 7**: (15 points) Choose either the Meltdown or Spectre attack, study the paper posted on the website, and answer the following questions:

## Briefly describe the attack and the hardware vulnerabilities that make the attack possible.

The Meltdown attack is a hardware vulnerability that attempts to gain access to the kernel by exploiting out-of-order execution, a modern technique that improves CPU performance by executing instructions non-sequentially “as soon as all required resources are available” [3]. This is possible due to ‘Speculative Execution,’ which the CPU uses to maximize resources by predicting upcoming instructions and assigning them to idle execution units. The vulnerability arises during specific fetch instructions from the privileged memory address. For example, when the CPU accesses data from memory, it stores a copy of it in the cache for faster access in the future. During Speculative Execution, the CPU loads this sensitive data into this cache even before the correct privileges can be verified. In other words, even if access would violate privilege rules, the data is still loaded into the cache during this time. This interaction between out-of-order memory lookups and the cache creates a vulnerability that can be exploited through a cache side-channel attack such as Flush+Reload; by careful timing and monitoring the cache accesses, an attacker can leak the contents and access sensitive data. Then, by repeating this technique for various points in memory, the attacker is able to extract all data stored in the kernel memory, “including the

entire physical memory” [3].

## Discuss the general impact of the attack on computer security.

The Meltdown attack has significant implications for the future of computer security, as it exploits optimization methods that are well established in the field. While the risks of these methods have been known for decades, the risks have been considered negligible and manageable up until this point. What sets Meltdown apart from previous attacks is the level of granularity with which an attacker can access sensitive information. Unlike previous vulnerabilities that targeted larger data blocks, Meltdown enables attackers to access individual bits. This level of detail and precision presents an unprecedented challenge for traditional defenses, effectively rendering them incapable of mitigating the threat.

## Explain mitigation strategies to mitigate the security risks due to the attack.

Since Meltdown is a hardware vulnerability, even software that is specifically designed to counter similar side-channel attacks remains vulnerable “if the design of the underlying hardware is not taken into account" [3]. This means that regardless of software defenses, the system remains susceptible if the hardware design does not adequately address security concerns. However, this is not as simple as removing out-of-order execution capabilities from CPU’s, as doing so would have “devastating” performance impacts as it would eliminate the advantages of parallel processing that modern CPUs rely on to execute tasks efficiently [3]. Similarly, stalling the memory fetch until privileges can be verified would also introduce a significant overhead, as each fetch would need to pause while it waits for validation.

While there are a few solutions presented, many of the solutions impact performance or are only temporary measure until a permanent hardware fix can be developed. One such example introduced by the authors is to ensure that user space and kernel space reside in distinct and separate memory regions.

susceptibility lies in the fundamental architecture of the hardware, posing persistent challenges for cybersecurity measures.

A more practical solution suggested is the introduction of a hard split between user space and kernel space

[1] book

**[2]** **http://practicalcryptography.com/cryptanalysis/letter-frequencies-various-languages/english-letter-frequencies/**

[3] paper