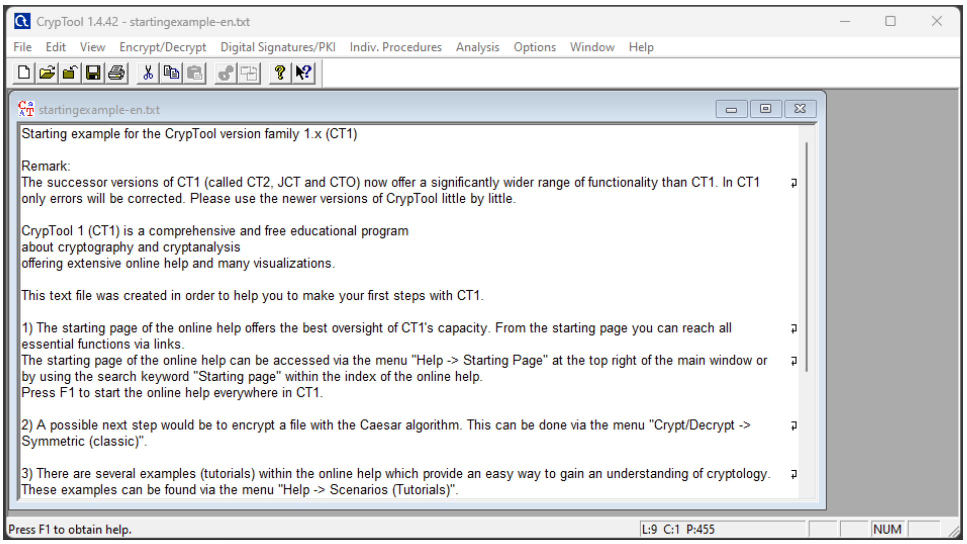
# Homework 1

Track A

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

a) Download CrypTool (v1.4.42) and install it on your computer.



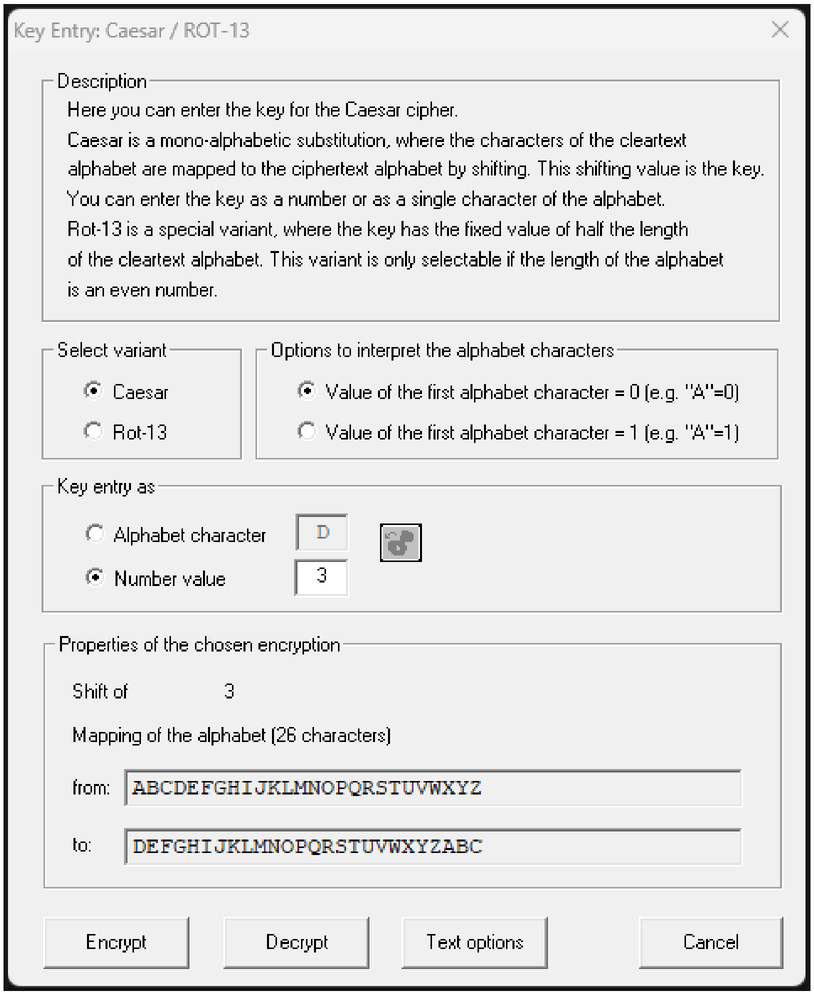

b) Encrypt the following message using Caesar Cipher (Shift-3, $C = (p + 3) mod\ 26$) and submit your ciphertext

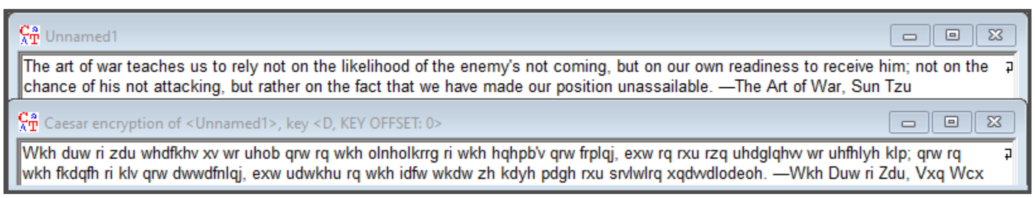
> The art of war teaches us to rely not on the likelihood of the enemy's not coming,
> but on our own readiness to receive him; not on the chance of his not attacking,
> but rather on the fact that we have made our position unassailable.
> —The Art of War, Sun Tzu

- Ciphertext Solution

> 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



**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.

a) 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.

b) Walkthrough

The decryption process begins by analyzing $n$-grams, or sequences of $n$ characters. By computing and comparing the histogram and diagram frequency of the ciphertext, we can pinpoint potential starting points for manually decrypting the substitution cipher.

| No. | Character seq... | Frequency in % | Frequency |
|---|---|---|---|
| 1 | R | 13.8225 | 81 |
| 2 | S | 11.9454 | 70 |
| 3 | M | 8.3618 | 49 |
| 4 | Q | 7.8498 | 46 |
| 5 | B | 5.9727 | 35 |
| 6 | I | 5.9727 | 35 |
| 7 | J | 5.6314 | 33 |
| 8 | F | 5.2901 | 31 |
| 9 | Y | 4.9488 | 29 |
| 10 | P | 4.6075 | 27 |
| 11 | N | 4.0956 | 24 |
| 12 | K | 3.7543 | 22 |
| 13 | X | 3.7543 | 22 |
| 14 | O | 2.0478 | 12 |
| 15 | G | 1.8771 | 11 |
| 16 | T | 1.7065 | 10 |
| 17 | C | 1.5358 | 9 |
| 18 | E | 1.5358 | 9 |
| 19 | A | 1.1945 | 7 |
| 20 | H | 1.1945 | 7 |
| 21 | V | 1.1945 | 7 |
| 22 | W | 1.1945 | 7 |
| 23 | U | 0.5119 | 3 |

| No. | Character seq... | Frequency in % | Frequency |
|---|---|---|---|
| 1 | SY | 4.7312 | 22 |
| 2 | YR | 4.0860 | 19 |
| 3 | SR | 2.3656 | 11 |
| 4 | FX | 2.1505 | 10 |
| 5 | MF | 2.1505 | 10 |
| 6 | QS | 2.1505 | 10 |
| 7 | BQ | 1.9355 | 9 |
| 8 | MI | 1.9355 | 9 |
| 9 | RQ | 1.7204 | 8 |
| 10 | BI | 1.5054 | 7 |
| 11 | JI | 1.5054 | 7 |
| 12 | RP | 1.5054 | 7 |
| 13 | RW | 1.5054 | 7 |
| 14 | FR | 1.2903 | 6 |
| 15 | SJ | 1.2903 | 6 |
| 16 | WS | 1.2903 | 6 |
| 17 | XQ | 1.2903 | 6 |
| 18 | FM | 1.0753 | 5 |
| 19 | HR | 1.0753 | 5 |
| 20 | IM | 1.0753 | 5 |
| 21 | IS | 1.0753 | 5 |
| 22 | KF | 1.0753 | 5 |
| 23 | MS | 1.0753 | 5 |

This analysis identifies the three most frequent characters in the ciphertext as 'R' (13.82%), 'S' (11.94%), and 'M' (8.36%), and the three most frequent digrams as 'SY' (4.73%), 'YR' (4.08%), and 'SR' (2.36%).

These values are important as they can be compared against the English language's frequency distribution, as noted by Practical Cryptography [2]. In English text, the three most frequent characters are 'E' (12.1%), 'T' (8.94%), and 'A' (8.55%), with the three most frequent digrams being 'TH' (2.71%), 'HE' (2.33%), and 'IN' (2.03%). An overview of these $n$-grams and their frequencies are shown below.

| | | |
|---|---|---|
| A : 8.55 | K : 0.81 | U : 2.68 |
| B : 1.60 | L : 4.21 | V : 1.06 |
| C : 3.16 | M : 2.53 | W : 1.83 |
| D : 3.87 | N : 7.17 | X : 0.19 |
| E : 12.10 | O : 7.47 | Y : 1.72 |
| F : 2.18 | P : 2.07 | Z : 0.11 |
| G : 2.09 | Q : 0.10 | |
| H : 4.96 | R : 6.33 | |
| I : 7.33 | S : 6.73 | |
| J : 0.22 | T : 8.94 | |

| | | |
|---|---|---|
| TH : 2.71 | EN : 1.13 | NG : 0.89 |
| HE : 2.33 | AT : 1.12 | AL : 0.88 |
| IN : 2.03 | ED : 1.08 | IT : 0.88 |
| ER : 1.78 | ND : 1.07 | AS : 0.87 |
| AN : 1.61 | TO : 1.07 | IS : 0.86 |
| RE : 1.41 | OR : 1.06 | HA : 0.83 |
| ES : 1.32 | EA : 1.00 | ET : 0.76 |
| ON : 1.32 | TI : 0.99 | SE : 0.73 |
| ST : 1.25 | AR : 0.98 | OU : 0.72 |
| NT : 1.17 | TE : 0.98 | OF : 0.71 |

Upon closer examination of the bigrams in our cipher, a distinct pattern of three-letter sequences emerge:

$$SY$$
$$YR$$
$$SR$$

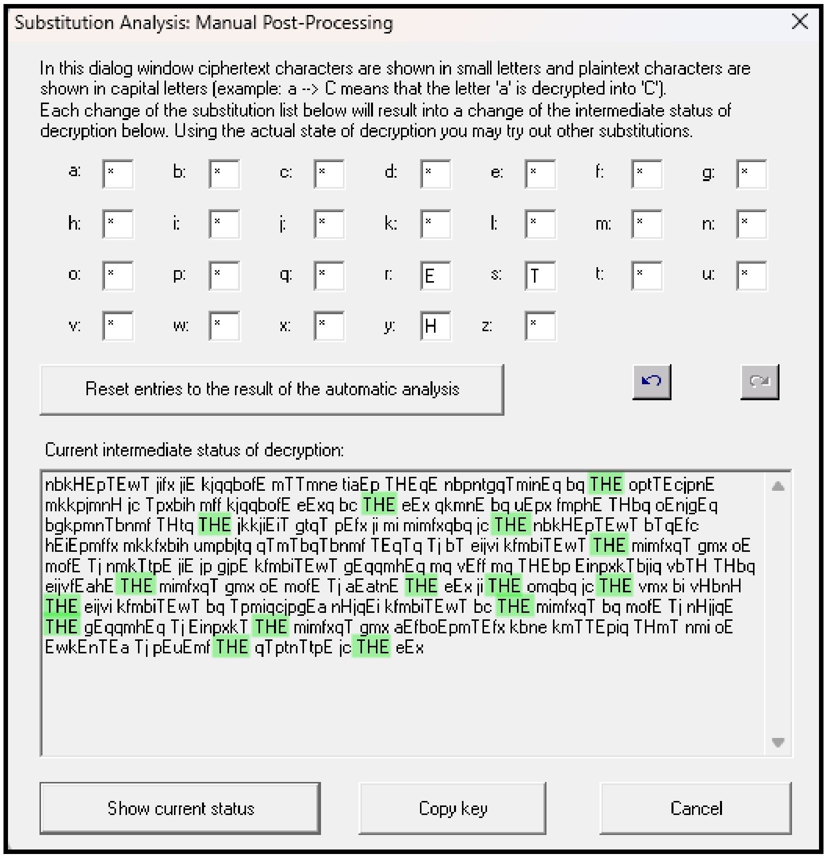
As such, we can identify three common bigrams in the English language that mirror this pattern. Starting with the most popular bigram in English, we can efficiently identify the remaining bigrams that fit this criteria.

| | | |
|---|---|---|
| TH : 2.71 | EN : 1.13 | NG : 0.89 |
| HE : 2.33 | AT : 1.12 | AL : 0.88 |
| IN : 2.03 | ED : 1.08 | IT : 0.88 |
| ER : 1.78 | ND : 1.07 | AS : 0.87 |
| AN : 1.61 | TO : 1.07 | IS : 0.86 |
| RE : 1.41 | OR : 1.06 | HA : 0.83 |
| ES : 1.32 | EA : 1.00 | ET : 0.76 |
| ON : 1.32 | TI : 0.99 | SE : 0.73 |
| ST : 1.25 | AR : 0.98 | OU : 0.72 |
| NT : 1.17 | TE : 0.98 | OF : 0.71 |

| *Ciphertext* | *Plaintext* |
|---|---|
| $SY$ | $TH$ |
| $YR$ | $HE$ |
| $SR$ | $TE$ |

This alignment gains further credibility if we compare the frequency distribution of characters in both the ciphertext and the English language - our top two most frequent characters, 'R' and 'S', correspond closely to the most prevalent characters in English, 'E' and 'T'.

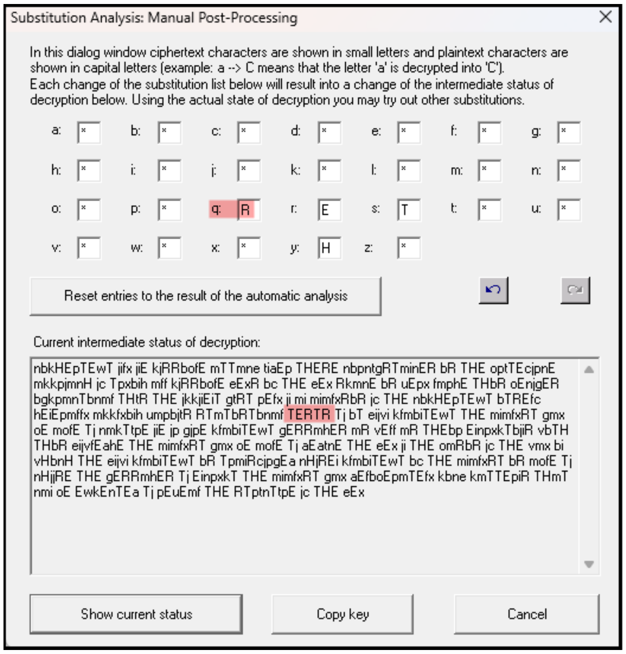
By plugging these values into the manual processing screen in CrypTool, our hypothesis is further validated as multiple three-letter words transform from 'SYR' to 'THE', a valid English stop word.

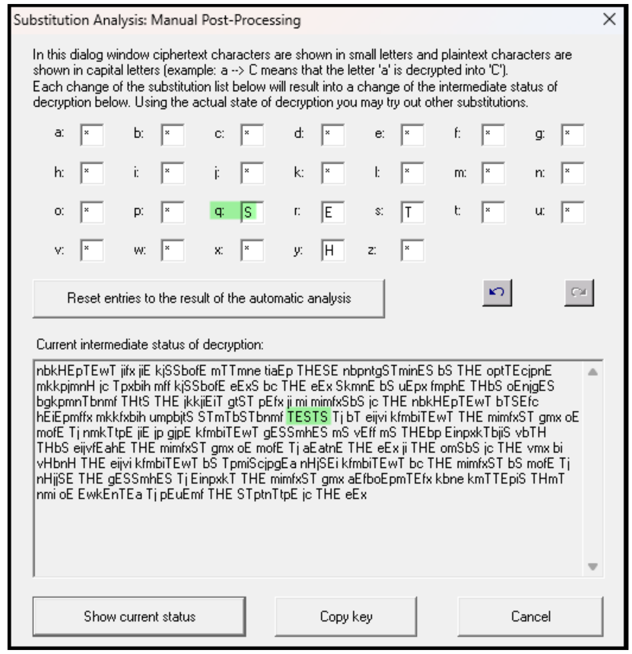



Now that a starting point is established, we can begin to fill in the remaining letters.

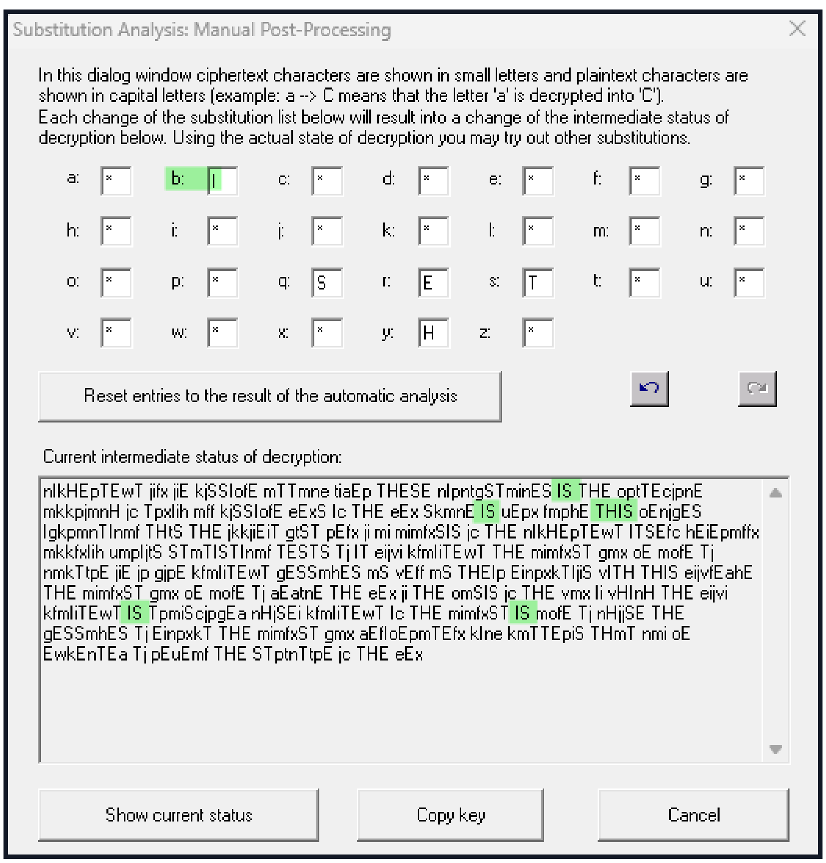
*Note: From this point onward, ciphertext letters will be represented in lowercase, while plaintext letters will be in uppercase.*

After incorporating our known letters, the word 'THEqE' appears. From this, we can deduce that 'q' is likely to represent either 'R' or 'S'. By substituting each possibility into the manual analyzer, we determine that 'q' most closely corresponds to 'S'. This deduction is supported by the transformation of 'TEqTq' to 'TESTS' when 'S' is used, whereas 'TEqTq' becomes 'TERTR' if 'R' is assumed, which is not an English word.

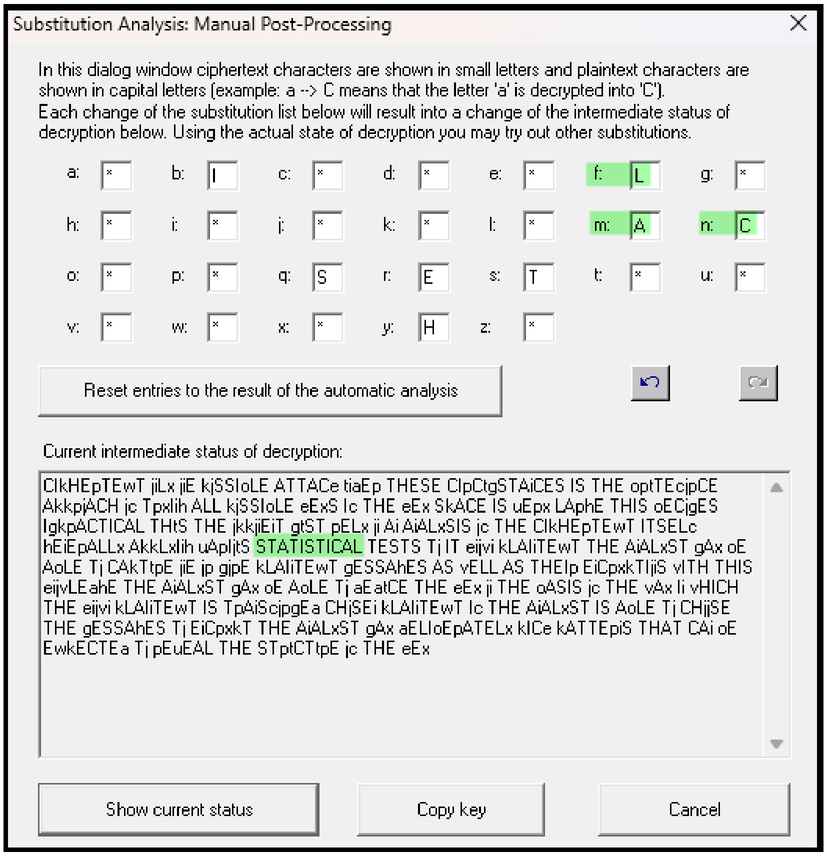



With 'S' identified, several instances suggest that 'b' corresponds to 'I', as evident in 'bS' becoming 'IS' and 'THbS' becoming 'THIS'.



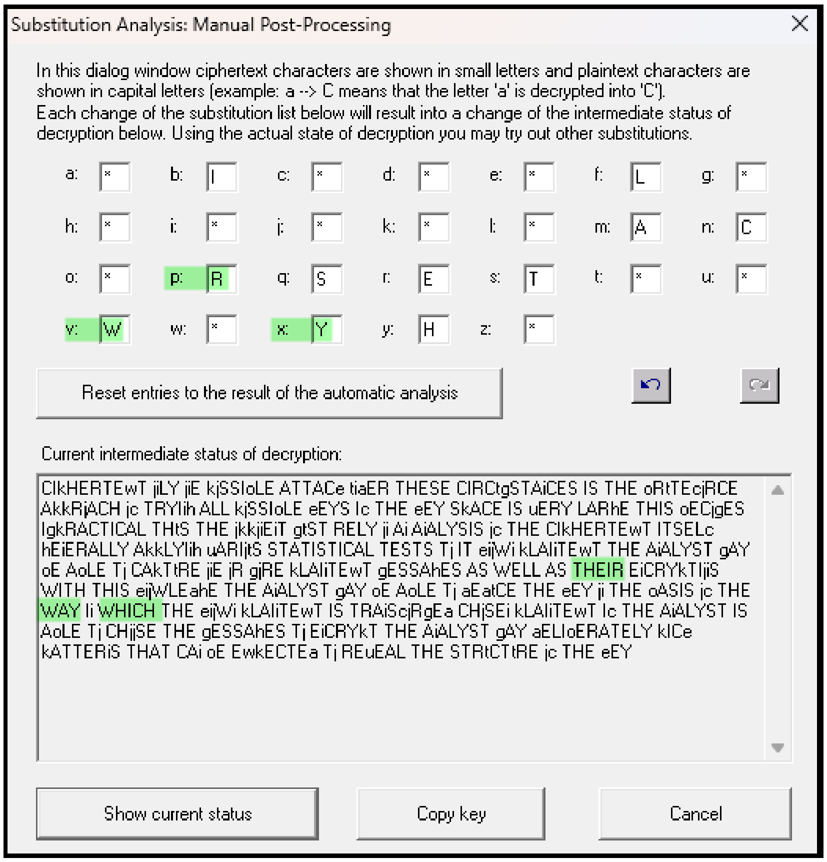
Next, we encounter word STmTISTInmf which is most likely the word STATISTICAL. As such, we can confidently assign 'A' for 'm', 'C' for 'n', and 'L' for 'f'.



Building on this progress, we proceed to fill in common stop words:

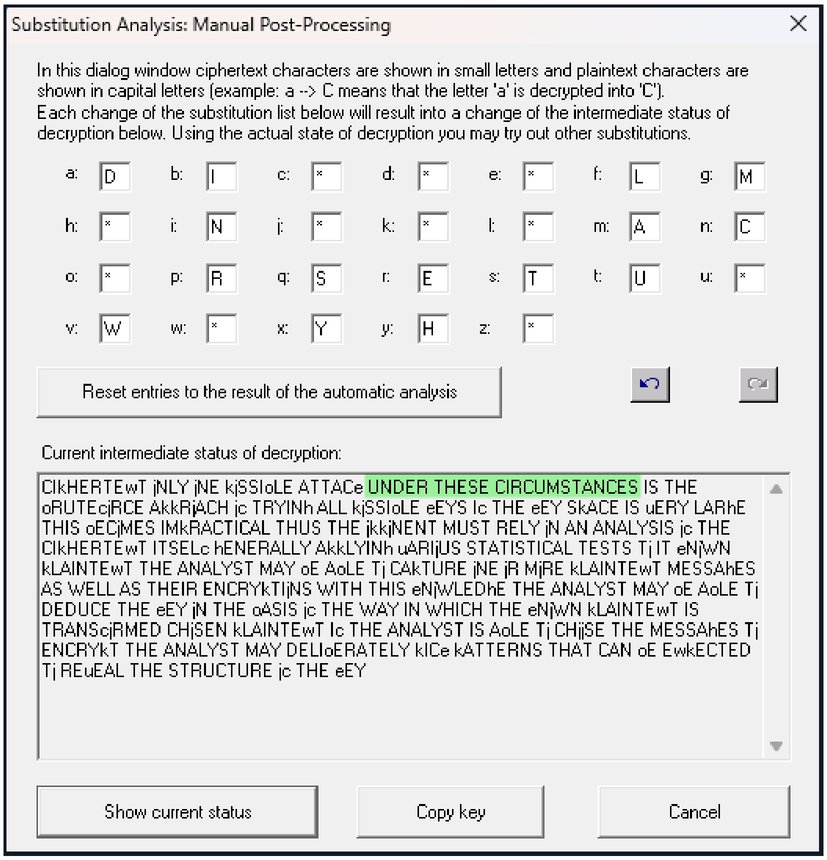
- vHICH v → W
- THEIp p → R
- WAx x → Y *

**We can safely assume this considering that 'S' and 'R' have already been identified*



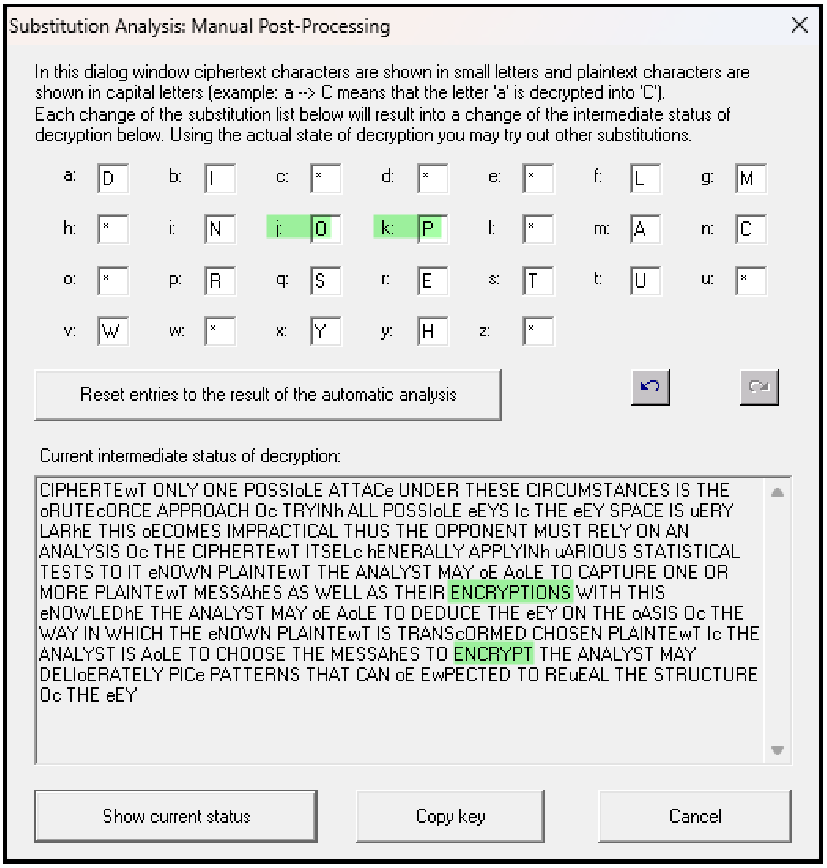
Next, we encounter the phrase 'tiaER THESE CIRCtgSTAiCES,' which decrypts to 'UNDER THESE CIRCUMSTANCES.' This decryption enables us to establish the following mappings:

- t → U
- i → N
- a → D
- g → M



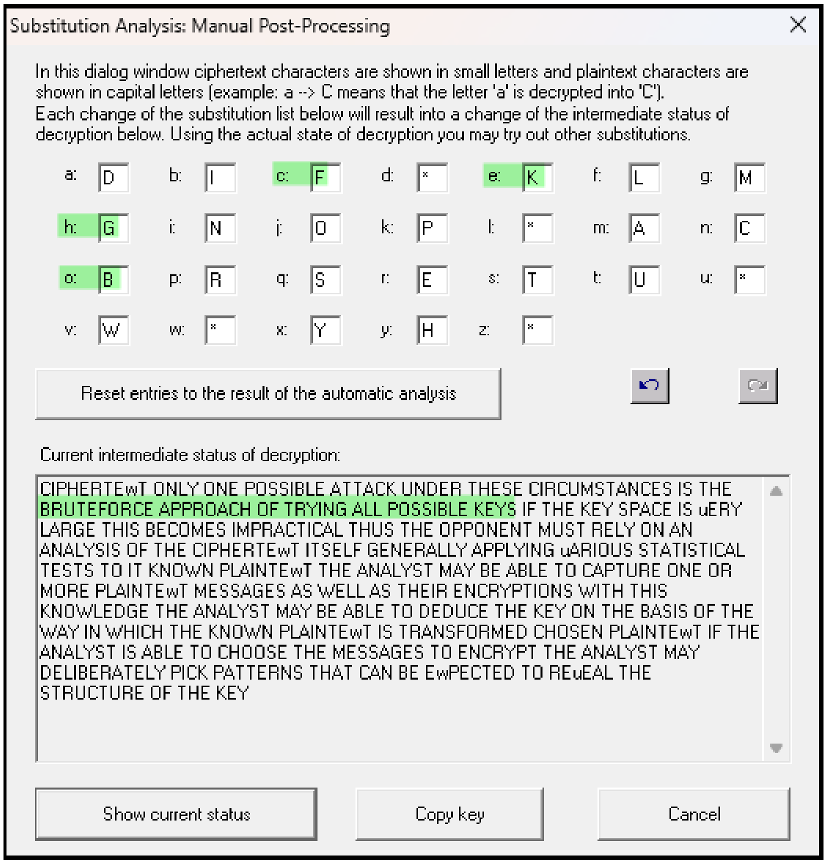
Then we can deduce that the word 'ENCRYkTIjNS,' likely decrypts to 'ENCRYPTION,' producing the following mappings:

- k → P
- j → O

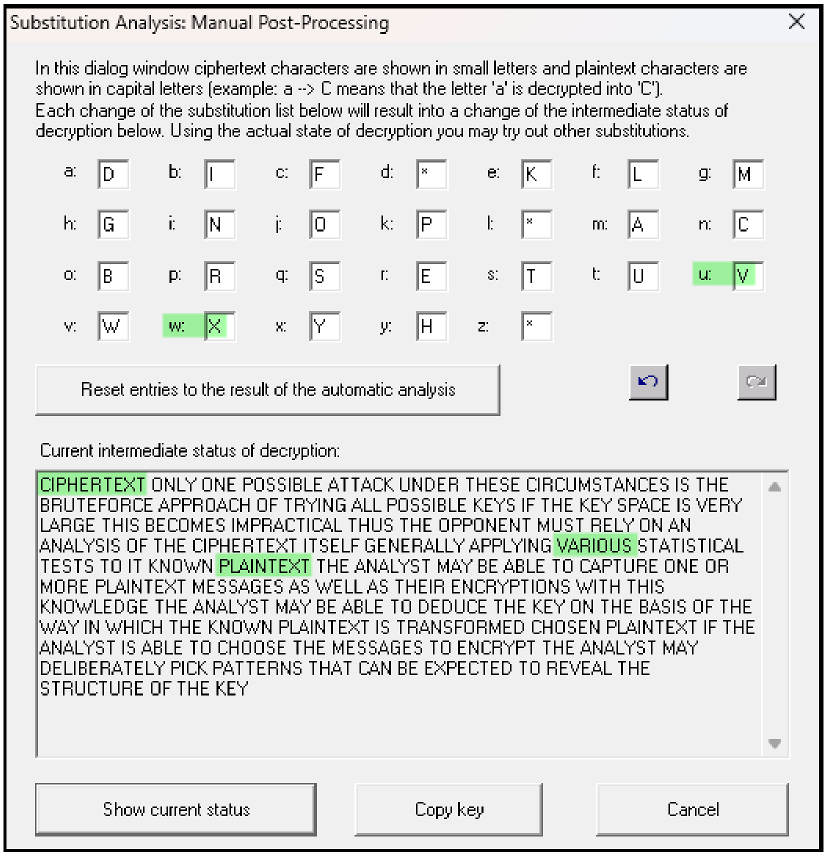


With a majority of the letters decrypted, we can now decipher longer phrases at a time. For instance, 'oRUTEcORCE APPROACH Oc TRYINh ALL POSSIoLE eEYS' decrypts to 'BRUTEFORCE APPROACH OF TRYING ALL POSSIBLE KEYS,' resulting in the following mappings:

- o → B
- c → F
- h → G
- e → K



This leaves the remaining letters that appear in the cipher, w and v. From the words CIPHERTEwT and PLAINTEwT, we can deduce that 'w' corresponds to 'X' and from the word uARIOUS we can deduce that 'u' represents 'V.'




This concludes the manual analysis of the cipher 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

a) 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 25! or $2^{84}$. The steps to convert is 25! to $2^{84}$ are shown below.

| | |
|---|---|
| $\log_b c = a$<br>$b^a = c$ | *logarithmic function* |
| $25! = 1.55 \times 10^{25}$ | *cipher possibilities as a factorial* |
| $\log_2(1.55 \times 10^{25}) = 83.6815$ | *substitute values* |
| $2^{83.6815} = 1.55 \times 10^{25}$<br>$2^{84} \approx 1.55 \times 10^{25}$ | *convert to exponent form* |

b) 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 $x$ 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:

| $x, y$ | $x+1, y$ | $x+2, y$ | $x+3, y$ | $x+4, y$ |
|---|---|---|---|---|
| **A B C D E** | V W X Y Z | Q R S T U | L M N O P | F G H I K |
| F G H I K | **A B C D E** | V W X Y Z | Q R S T U | L M N O P |
| L M N O P | F G H I K | **A B C D E** | V W X Y Z | Q R S T U |
| Q R S T U | L M N O P | F G H I K | **A B C D E** | V W X Y Z |
| V W X Y Z | Q R S T U | L M N O P | F G H I K | **A B C D E** |

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

$x, y$

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

$x, y+1$

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

$x, y+2$

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

$x, y+3$

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

$x, y+4$

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

Since there are 5 rows and 5 columns in a 5x5 matrix, there are $5 \times 5 = 25$ equivalent matrices. Therefore, we can solve for the number of effectively unique keys, $K$, that the Playfair Cipher has by doing the following:

| | |
|---|---|
| $K = \dfrac{total\ number\ of\ unique\ keys}{equivalent\ matrices}$ | *formula* |
| $K = \dfrac{25!}{25}$ | *substitute values* |
| $K = \dfrac{1 \times 2 \times 3 \times \ldots \times 24 \times \cancel{25}}{\cancel{25}}$<br>$K = \dfrac{1 \times 2 \times 3 \times \ldots \times 24}{1}$ | *cancel like terms* |
| $K = 24! = 6.204 \times 10^{23}$ | *solve* |

As such, there are 24!, or $6.204 \times 10^{23}$, effectively unique keys in the Playfair cipher.

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

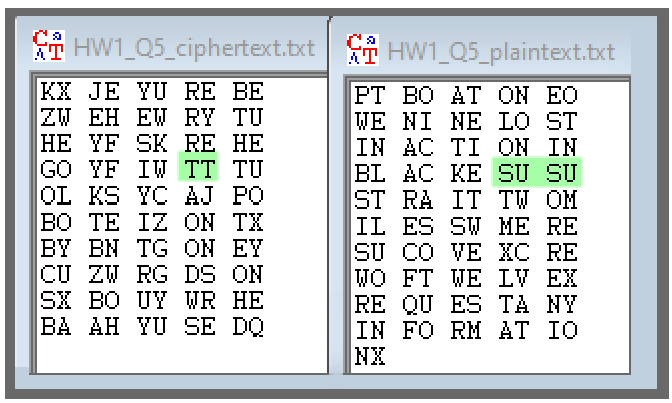
When the PT-109 American patrol boat, commanded by Lieutenant John F. Kennedy, was sunk by a Japanese destroyer, an encrypted message was received at an Australian wireless station in Playfair code. The message was encrypted using the key *royal new zealand navy*.

KXJEY UREBE ZWEHE WRYTU HEYFS
KREHE GOYFI WTTTU OLKSY CAJPO
BOTEI ZONTX BYBNT GONEY CUZWR
GDSON SXBOU YWRHE BAAHY USEDQ

a) 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"

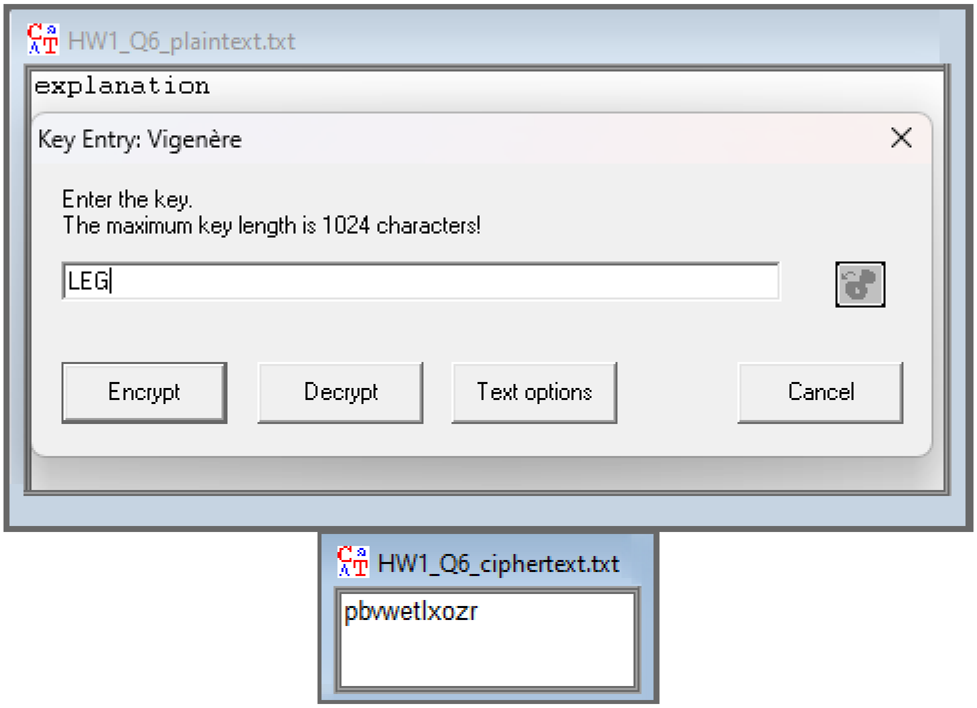



However, since the directions state to convert 'TT' to 'tt', we adjusted the highlighted section to read:

"PT BOAT ONE OWE NINE
LOST IN ACTION IN BLACKE**TT** 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




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

a) 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].

b) 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 and effectively renders them incapable of mitigating the threat.

c) 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.

Another example introduced by the authors is to ensure that user space and kernel space reside in distinct and separate memory regions. This approach is one of the most viable options as it not only prevents Meltdown attacks, but any degradation in performance would be negligible. However, this solution does not address a similar class of attack, Spectre. The authors emphasize the need to develop a solution that is capable of preventing both Meltdown and Spectre attacks simultaneously.

While there are a few solutions presented, many of them impact performance or are only temporary options until a permanent hardware fix can be developed. Until then, the authors suggest Kernel Page-Table Isolation (KPTI), also known as KAISER. KAISER works by preventing user-level processes from directly accessing kernel memory. This separation between user-space and kernel-space memory regions would prevent Meltdown attacks by ensuring that even if a process were to attempt Speculative Execution to access kernel memory, it would not be able to directly read or manipulate that memory because it is not mapped into the user space.

# References

[1] W. Stallings, Cryptography and Network Security: Principles and Practice, 6th ed., Upper Saddle, New Jersey: Prentice Hall Press, 2013.

[2] . J. Lyons, "English Letter Frequencies," 2012. [Online]. Available: http://practicalcryptography.com/cryptanalysis/letter-frequencies-various-languages/english-letter-frequencies/.

[3] M. Lipp, M. Schwarz, D. Gruss, T. Prescher, W. Haas, A. Fogh, J. Horn, S. Mangard, P. Kocher, D. Genkin, Y. Yarom and M. Hamburg, "Meltdown: Reading Kernel Memory from User Space," in *27th USENIX Security Symposium*, Baltimore, 2018.