**Experimentations**

i) How long does a menu need to be before the Stecker based on it produces a correct  
decrypt? What happens if the menu is too short?

Plain: “AIDEGHMC”

Message length: 8

Cipher: “IDEGHMCL”

Menu: [0,1,2,3,4,5,6,7] (length = 8)

Stecker: [('A','X'),('W','I'),('V','D'),('U','E'),('T','G'),('S','H'),('R','M'),('Q','C'),('P','L')]

Steckers Produced from shortening the menu length:

**\*Bombe\_Testing>** findStecker c1 [0,1,2,3,4,5,6,7] [('A','A')] (0,0,0)

Just [('A','X'),('W','I'),('V','D'),('U','E'),('T','G'),('S','H'),('R','M'),('Q','C'),('P','L')]

**\*Bombe\_Testing>** findStecker c1 [0,1,2,3,4,5,6] [('A','A')] (0,0,0)

Just [('A','X'),('W','I'),('V','D'),('U','E'),('T','G'),('S','H'),('R','M'),('Q','C')]

**\*Bombe\_Testing>** findStecker c1 [0,1,2,3,4,5] [('A','A')] (0,0,0)

Just [('A','W'),('X','I'),('F','D'),('T','E'),('U','G'),('C','H'),('Q','M')]

**\*Bombe\_Testing>** findStecker c1 [0,1,2,3,4] [('A','A')] (0,0,0)

Just [('A','O'),('F','I'),('X','D'),('R','E'),('M','G'),('H','H')]

**\*Bombe\_Testing>** findStecker c1 [0,1,2,3] [('A','A')] (0,0,0)

Just [('A','H'),('V','I'),('W','D'),('E','E'),('S','G')]

**\*Bombe\_Testing>** findStecker c1 [0,1,2] [('A','A')] (0,0,0)

Just [('A','G'),('Y','I'),('S','D'),('N','E')]

**\*Bombe\_Testing>** findStecker c1 [0,1] [('A','A')] (0,0,0)

Just [('A','B'),('I','I'),('L','D')]

**\*Bombe\_Testing>** findStecker c1 [0] [('A','A')] (0,0,0)

Just [('A','A'),('D','I')]

**\*Bombe\_Testing>** findStecker c1 [] [('A','A')] (0,0,0)

Just [('A','A')]

|  |  |  |
| --- | --- | --- |
| Menu Length: | Decrypt: | Num letters Wrong: |
| 8 | AIDEGHMC | 0 |
| 7 | AIDEGHMA | 1 |
| 6 | AIDEGHAI | 2 |
| 5 | AIDEGLVD | 3 |
| 4 | AIDEOFIX | 4 |
| 3 | AIDXMFVX | 5 |
| 2 | ALWJMFVX | 7 |
| 1 | XLWJMFVA | 8 |
| 0 | BRWJMFVX | 8 |

Plain: “COMPUTERSCIENCESHEFFIELDUNIVERSITYSTOP”

Message length: 38

Cipher: “RCQRSVHNYQHLVKLELFYSYCCLMKHUFXMVYVREFLHZOLRCBRHWPQDUONZWOGRTYKAUW”

Menu: [35,28,19,15,6,16,22,0,7,12,27,4,30,2] (length = 14)

Stecker: [('T','G'),('C','E'),('K','F'),('D','S'),('R','H'),('O','L'),('N','N'),('B','V'),('A','U'),('P','M'),('Q','Q')]

Steckers produced from reducing the menu length:

**\*Bombe\_Testing>** findStecker crib2 [35,28,19,15,6,16,22,0,7,12,27,4,30,2] [('T','T')] (0,0,5)

Just [('T','G'),('C','E'),('K','F'),('D','S'),('R','H'),('O','L'),('N','N'),('B','V'),('A','U'),('P','M'),('Q','Q')]

**\*Bombe\_Testing>** findStecker crib2 [35,28,19,15,6,16,22,0,7,12,27,4,30] [('T','T')] (0,0,5)

Just [('T','G'),('C','E'),('K','F'),('D','S'),('R','H'),('O','L'),('N','N'),('B','V'),('A','U'),('P','M')]

**\*Bombe\_Testing>** findStecker crib2 [35,28,19,15,6,16,22,0,7,12,27,4] [('T','T')] (0,0,5)

Just [('T','G'),('C','E'),('K','F'),('D','S'),('R','H'),('O','L'),('N','N'),('B','V'),('A','U')]

**\*Bombe\_Testing>** findStecker crib2 [35,28,19,15,6,16,22,0,7,12,27] [('T','T')] (0,0,5)

Just [('T','G'),('C','E'),('K','F'),('D','S'),('R','H'),('O','L'),('N','N'),('B','V'),('A','U')]

**\*Bombe\_Testing>** findStecker crib2 [35,28,19,15,6,16,22,0,7,12] [('T','T')] (0,0,5)

Just [('T','G'),('C','E'),('K','F'),('D','S'),('R','H'),('O','L'),('N','N'),('B','V')]

**\*Bombe\_Testing>** findStecker crib2 [35,28,19,15,6,16,22,0,7] [('T','T')] (0,0,5)

Just [('T','G'),('C','E'),('K','F'),('D','S'),('R','H'),('O','L'),('N','N')]

**\*Bombe\_Testing>** findStecker crib2 [35,28,19,15,6,16,22,0] [('T','T')] (0,0,5)

Just [('T','G'),('C','E'),('K','F'),('D','S'),('R','H'),('O','L')]

**\*Bombe\_Testing>** findStecker crib2 [35,28,19,15,6,16,22] [('T','T')] (0,0,5)

Just [('T','G'),('C','E'),('K','F'),('D','S'),('R','H'),('O','L')]

**\*Bombe\_Testing>** findStecker crib2 [35,28,19,15,6,16] [('T','T')] (0,0,5)

Just [('T','G'),('C','E'),('K','F'),('D','S'),('R','H'),('O','L')]

**\*Bombe\_Testing>** findStecker crib2 [35,28,19,15,6] [('T','T')] (0,0,5)

Just [('T','G'),('C','E'),('K','F'),('D','S'),('R','H')]

**\*Bombe\_Testing>** findStecker crib2 [35,28,19,15] [('T','T')] (0,0,5)

Just [('T','G'),('C','E'),('K','F'),('D','S')]

**\*Bombe\_Testing>** findStecker crib2 [35,28,19] [('T','T')] (0,0,5)

Just [('T','U'),('J','E'),('N','F'),('S','S')]

**\*Bombe\_Testing>** findStecker crib2 [35,28] [('T','T')] (0,0,5)

Just [('T','U'),('J','E'),('N','F')]

**\*Bombe\_Testing>** findStecker crib2 [35] [('T','T')] (0,0,5)

Just [('T','T'),('S','E')]

**\*Bombe\_Testing>** findStecker crib2 [] [('T','T')] (0,0,5)

Just [('T','T')]

|  |  |  |
| --- | --- | --- |
| Menu Length: | Decrypt | Num letters Wrong: |
| 14 | "COMPUTERSCJENCESHEFFJELDUNJVERSJTYSTOPTHJSJSTHETESTTWOMESSAGESTOP" | 4 |
| 13 | "COMPUTERSCJENCESHEFFJELDUNJVERSJTYSTOPTHJSJSTHETESTTWOMESSAGESTOP" | 4 |
| 12 | "COPMUTERSCJENCESHEFFJELDENJVERDJTYSTOMTHJSJSTHETUSTTWOPESSAGESTOM" | 7 |
| 11 | "COPMUTERSCJENCESHEFFJELDENJVERDJTYSTOMTHJSJSTHETUSTTWOPESSAGESTOM" | 7 |
| 10 | "COPMATERSCJENCESHEFFJELDENJGERDJTYSTOMTHJSJSTHETASTGWOPESSUGESWLM" | 9 |
| 9 | "COPMACERSCJEDCESHEFFJELDENJGERDKTUSTOMTHJSJSPHETASTGWOPESSUGESWLM" |  |
| 8 | "COPMACERSCJEDCESHEFFJELDENJGERDKTUSTOMTHJSJSPHETASTGWOPESSUGESWLM |  |
| 7 | "COPMACERSCJEDCESHEFFJELDENJGERDKTUSTOMTHJSJSPHETASTGWOPESSUGESWLM" |  |
| 6 | "COPMACERSCJEDCESHEFFJELDENJGERDKTUSTOMTHJSJSPHETASTGWOPESSUGESWLM" |  |
| 5 | "CLPMACERSCJDDCZSSEFFJEOIENJGERDKTUSTLKTHNVJSPHETASTGMLPEPSUGESWOM" |  |
| 4 | "DLPLACGHSCIDDCZSSEFFJEOIENGGEHDKTUPTLKVRNVSSPHXTASTGMLPEPSOGESWOM" |  |
| 3 | "SVPLZJUYDJISSVZGDQKFEJRICSUTEHSNGTPTPNVRFVDQPHXGADUTMEPCPROKCEWNM" |  |
| 2 | "SVPLZJUYDJISSVZGDQKFEJRICSUTEHSNGTPTPNVRFVDQPHXGADUTMEPCPROKCEWNM" |  |
| 1 | "EVPLTSTHDSIEEVZPDMKPJSRICETTRHEFGUPTBFVRNVDQPHXGADTTMLPCPROGCJWOM" |  |
| 0 | "SVPLZETHDEISSVZMDMKNJERICSTTRHSFGUPPBFVRNVDQPHXGADTTMLPCPROGCJWOM" |  |

From the examples above I can conclude that:

As you shorten the menu, the decrypt gets worse at a rate of around 1 new incorrect character per menu link missing.

It seems that to get a complete decrypt, you need a menu of length = to the length of the word, as demonstrated by the first example where menu length = 8 and word length = 8, produces a correct decrypt.

If the menu is too short, more and more errors occur in the output message. And if the menu length is ½ the length of the word, you end up getting around half the decrypt incorrect.

ii) Does the answer to i) depend on how many pairs are in the stecker?

As shown above, as the menu length decreases, the number of pairs in the stecker decreases too. So yes, the answer to i) does depend on how many pairs there are in the stecker as they are related; a menu of length n produces a stecker of length n+1.