

Cryptography

Tools Used

- RapidTables
 - <https://www.rapidtables.com/web/tools/index.html>
 - Base64
 - Binary
 - Hex String
- Cryptii
 - <https://cryptii.com/>
 - Decode
- Decode
 - <https://www.dcode.fr/>
 - RSA Cipher
- Kali Linux
 - Strings

Challenge 1

Using rapidtables I was able to decode the encoded strings. The first string was a hexadecimal string. I omitted the 0x and pasted the remaining string into the decoder. `0x73636f727069666e`

`scorpion` The decoded string is **scorpion**.

I then proceeded to the next string. The string is a base64 string. I then pasted that into the decoder. `c2NyaWJibGU` `scribble` The decoded string is **scribble**.

The next string was a binary string. `01110011 01100101 01100011 01110101 01110010 01100101 01101100 01111001` `securely` The decoded string is **securely**.

The last string was a double encoded string. Original word was first encoded into a base64 and then that base64 was encoded into a binary string. I first decoded the binary to reveal the base64 and then decoded the base64 to reveal the original string. `01100010 01000111 00111001 01110011`
`01100010 01000111 01101100 01110111 01100010 00110011 01000001 00111101` `bG9sbGlwb3A=` `lollipop`

The decoded string is **lollipop**.

Challenge 2

Using cryptii I took the encrypted string and decoded it using a caesar cipher with a shift of 13. I used the Large Language Model Gemma to aid in my decryption. The model was able to tell me the most logical cipher based on the encoding. `iveghny ynxr` `virtual lake` The decoded string is **virtual lake**.

Challenge 3

Using cryptii and gima I was able to determine that the cipher is an alphabetical substitution using the alphabet in reverse. `hzuvob lyerlfh xzev` `safely obvious cave` The decoded string is **safely obvious cave**.

Challenge 4

Based on the ... and the -- I was able to determine this was morse code. I used cryptii to convert the string. `.... / - . . - . . - / - - - . . . / - . . - . . - . . - . . - / - . . . / ..`
`... / - . . - . . - . . - / ... - . - . . - . . - / ... - . - . . - / - . - - . . . / -`
`. - - - - -` `the secret of getting ahead is getting started sky dkvb 9816` The decoded string is **the secret of getting ahead is getting started sky dkvb 9816**.

Challenge 5

The notes for this challenge is that there were handwritten notes with 3 & 5. Using this information I looked for ciphers on cryptii where the numbers could be inserted and keys. I was able to figure out this was a Rail Cipher. 3 correlated to the first message and 5 the second. `Cair eruSA-0org`

sgaeudrpesr K-II98.ue cn seYQ3 Courage is grace under pressure SKY-AIQI-9380. The decoded first string is **Courage is grace under pressure SKY-AIQI-9380.** F daS-eefn n KZ3eheadty.YI8lta oiwy-Q0. r aI2 Feel the fear and do it anyway. SKY-IQIZ-3802. The decoded second string is **Feel the fear and do it anyway. SKY-IQIZ-3802.**

Challenge 6

The notes for this challenge was that the keyword is **qizkwcgqbs**. The let me to a vigenere cipher because it uses a key. Using cryptii and the keyword I was able to decipher the phrase. Y ln xkv lubj swlzqvkhT, A vmzb pjK bbua we ddgs ILQ-GQYU-8026 I do not fear computers, I fear the lack of them SKY-QIZK-8026 The decrypted string is **I do not fear computers, I fear the lack of them SKY-QIZK-8026.**

Challenge 7

For this challenge I used Kali Linux. The provided file had a string encoded into it. I first tried looking through the metadata and through steghide. The strign was hidden in the strings metadata of the file. Using the following command strings Step1.jpg | grep SKY I was able to pull tjust the string I needed out of the file. SKY-TVJI-2063 The decoded string is **SKY-TVJI-2063.**

Challage 8

This challenge is decoding a string using RSA. I used dcode.fr and their RSA decrypter to decipher the text. The provided values were N, C, and E. I was then able to use that to pull P adn Q from dcode. Then With that I was able to decode the string. C is the encoded string. n = 1079 e = 43 c = 996 894 379 631 894 82 379 852 631 677 677 194 893 p = 13 q = 83 SKY-KRYG-5530 The decoded string is **SKY-KRYG-5530.**

Revision #2

Created 2025-11-25 18:00:49 UTC by David Rizzo

Updated 2025-11-25 21:26:55 UTC by David Rizzo